Though the waystation be dangerous and the destination far, far away,
There is no road that has no end; grieve not!

– Hafez

University of Alberta

DESIGN, HIGH-LEVEL SYNTHESIS, AND DISCRETE OPTIMIZATION OF DIGITAL FILTERS BASED ON PARTICLE SWARM OPTIMIZATION

by

Seyyed Ali Hashemi

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

Master of Science

in

Communications

Department of Electrical and Computer Engineering

©Seyyed Ali Hashemi
Fall 2011
Edmonton, Alberta

Permission is hereby granted to the University of Alberta Libraries to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only. Where the thesis is converted to, or otherwise made available in digital form, the University of Alberta will advise potential users of the thesis of these terms.

The author reserves all other publication and other rights in association with the copyright in the thesis, and except as herein before provided, neither the thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatever without the author's prior written permission.

To my parents
For their unconditional love, support, and encouragement.

Abstract

This thesis is concerned with the development of a novel discrete particle swarm optimization (PSO) technique and its application to the discrete optimization of digital filter frequency response characteristics on the one hand, and the high-level synthesis of bit-parallel digital filter data-paths on the other. Two different techniques are presented for the optimization of sharp-transition band frequency response masking (FRM) digital filters, one of which is based on the conventional finite impulse-response (FIR) digital subfilters, and a new hardware-efficient approach which is based on utilizing infinite impulse-response (IIR) digital subfilters. It is shown that further hardware efficiency can be achieved by realizing the IIR interpolation subfilters by using the bilinear-LDI approach. The corresponding discrete PSO is carried out over the canonical signed digit (CSD) multiplier coefficient space for direct mapping to digital hardware considering simultaneous magnitude and group-delay frequency response characteristics. A powerful encoding scheme is developed for the high-level synthesis of digital filters based on discrete PSO, which preserves the data dependency relationships in the digital filter data-paths. In addition, a constrained discrete PSO is developed to overcome the limitations which would manifest themselves if the conventional PSO were to be used. Several examples are presented to demonstrate the application of discrete PSO to the design, high-level synthesis and optimization of digital filters.

Acknowledgements

I am heartily thankful to my supervisor, Dr. Behrouz Nowrouzian, whose encouragement, supervision and support from the preliminary to the concluding level enabled me to develop an understanding of the subject. This thesis would not have been possible without the patient advice and criticism offered by Dr. Nowrouzian during the period of my M.Sc. studies.

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

Nomenclature

$\Delta\omega_a$	Stopband Frequency Region
$\Delta\omega_p$	Passband Frequency Region
Δ_a	Stopband Attenuation of EMQF Filter
δ_a	Squared Stopband Tolerance of EMQF Filter
Δ_O	Operation Delay
Δ_p	Passband Ripple of EMQF Filter
δ_p	Squared Passband Tolerance of EMQF Filter
$\hat{C}$	Quantized Capacitance
$\hat{g}_{best}$	Integer Global Best Position Element of a Particle
$\hat{L}$	Quantized Inductance
$\hat{m}$	Quantized Multiplier Coefficient
$\hat{v}$	Integer Velocity Element of a Particle
$\hat{v}_{max}$	Integer Maximum Velocity Limit
$\hat{v}_{min}$	Integer Minimum Velocity Limit
$\hat{x}$	Integer Position Element of a Particle
$\hat{x}_{best}$	Integer Best Position Element of a Particle
$\hat{x}_{max}$	Integer Maximum Position Limit
$\hat{x}_{min}$	Integer Minimum Position Limit
λ_M	Displacement for M
$\mathbf{I}$	Identity matrix
$\mathbf{T}$	Transfer Function Matrix
$\mathbf{x}$	Input Vector
$\mathbf{y}$	Output Vector
μ_τ	Average Group-Delay over the Passband Region

ω	Real Frequency Variable
ω_1	Weighting Factor for C_1
ω_2	Weighting Factor for C_2
Ω_A	Analog Frequency Variable
ω_a	Stopband Frequency of Interpolation Digital Subfilter
ω_d	Digital Frequency Variable
ω_p	Passband Frequency of Interpolation Digital Subfilter
ϕ_A	Analog Stopband Frequency
τ	Group-Delay Frequency Response
EvP	Even part of Polynomial P
OdP	Odd Part of Polynomial P
θ_A	Analog Passband Frequency
θ_n	User-Specified Number
Θ_O	Operator Number
$\tilde{\phi}$	Approximate Stopband Edge
$\tilde{\theta}$	Approximate Passband Edge
$\tilde{D}_M$	Distance for M
$\tilde{K}$	Constant Used in the Calculation of G
$\tilde{n}$	Order of Hurwitz Polynomial
ε_a	Maximum Error in the Stopband
ε_p	Maximum Error in the Passband
ς_p	Maximum Group-Delay Distortion
ζ	Constant for Calculating W_{gd}
A	Alternative Digital Subfilter
A_a	FRM Digital Filter Minimum Stopband Loss
A_p	FRM Digital Filter Maximum Passband Ripple
A_{add}	Adder Cost
A_{mult}	Multiplier Cost
A_{mux}	Multiplexor Cost
A_{reg}	Register Cost

B	Alternative Digital Subfilter
b	Order of Signals in an Operator
B_1	Number of Coordinates for Interpolation Digital Subfilters
B_2	Number of Coordinates for Masking Digital Subfilters
C	Overall Cost for Digital Filter Data-Path
C'	Modified Capacitance
C_1	Hardware Requirement Cost
c_1	Correction Factor
C_2	Time-Requirement Cost
c_2	Correction Factor
C_i	i-th Capacitance
D	CSD Digit
f	Number of Digits in the Fractional Part
F_0	Masking Digital Subfilter
F_1	Masking Digital Subfilter
$fitness$	FRM Digital Filter Overall Fitness Value
$fitness_{group-delay}$	FRM Digital Filter Group-Delay Fitness Value
$fitness_{magnitude}$	FRM Digital Filter Magnitude Fitness Value
G	G_0 or G_1
g	Voltage Divider Function
G_0	Odd-Ordered Digital Allpass Network
G_1	Even-Ordered Digital Allpass Network
G_{add}	Number of Gate-Equivalents of a Basic Full Adder
G_{mult}	Number of Gate-Equivalents of a Multiplier
G_{mux}	Number of Gate-Equivalents of a Multiplexor
G_{reg}	Number of Gate-Equivalents of a Register
G_{best}	Global Best Position Vector
g_{best}	Global Best Position Element
H	Overall FRM Digital Filter
H_a	Interpolation Digital Subfilter

H_b	Power Complementary Interpolation Digital Subfilter
H_{hp}	Highpass FRM Digital Filter
H_{lp}	Lowpass FRM Digital Filter
I	Total Number of Operations in a DFG
I_L	Number of Image Lobes
K	Number of Particles
k	Ratio of Analog Passband and Stopband Edges
L	Wordlength
l	Maximum Number of Non-Zero Digits
$L^{'}$	Modified Inductance
L_f	Number of Entries in the Footer of a LUT
L_h	Number of Entries in the Header of a LUT
L_i	i-th Inductance
l_{begin}	Time-Step Where a Signal is First Generated
l_{end}	Time-Step Where a Signal is Last Consumed
M	Interpolation Factor
m	Infinite-Precision Multiplier Coefficient
M_{hp}	Highpass Filter Interpolation Factor
M_{lp}	Lowpass Filter Interpolation Factor
N	Number of Design Variables
N_M	Number of Different Available Functional Units
N_{add}	Number of Digital Adders
N_{mult}	Number of Digital Multipliers
N_{mux}	Number of Multiplexors
N_{reg}	Number of Registers
N_{FIR}	Order of FIR Digital Filter
N_{IIR}	Order of IIR Digital Filter
O	Operation
P	Hurwitz Polynomial
P^1	First Partition of a Particle

P^2	Second Partition of a Particle
R	Fixed Value Radix-Point
$r^{'}$	Modified Resistance
r_0	Resistance
r_1	Random Number
r_2	Random Number
s	Analog Frequency Variable
T	Sampling Period
t_{ALAP}	ALAP Scheduling Time-Step
T_{ASAP}	Total Number of ASAP Scheduling Time-Steps
t_{ASAP}	ASAP Scheduling Time-Step
T_{clk}	Duration of Each Time-Step
t_{DFF}	Propagation Delay through a D-Flip-Flop
t_{diff}	Difference between ALAP and ASAP Scheduling Time-Steps
t_{FA}	Propagation Delay through a Full Adder
t_{G}	Propagation Delay through a Typical Gate
t_{mux}	Propagation Delay through a Two-Input Multiplexor
t_{total}	Total Number of Time-Steps
V	Particle's Velocity Vector
v	Particle's Velocity Element
V_{A_1}	Input Node to g
V_{A_2}	Output Node of g
v_{max}	Maximum Velocity Limit
v_{min}	Minimum Velocity Limit
W	Signal Wordlength
w	Inertia Weight
W_a	Weighting Factor for Stopband Frequency Response
W_p	Weighting Factor for Passband Frequency Response
W_{gd}	Weighting Factor for Group-Delay Frequency Response
X	Particle's Position Vector

x	Particle's Position Element
X_{best}	Particle's Best Position Vector
x_{best}	Particle's Best Position Element
x_{max}	Maximum Position Limit
x_{min}	Minimum Position Limit
Y	Realizable Reactive Admittance
Z	Realizable Reactive Impedance
z	Discrete-Time Complex Frequency Variable
ACO	Ant Colony Optimization
ALAP	As Late As Possible
ASAP	As Soon As Possible
BIBO	Bounded-Input-Bounded-Output
Bilinear-LDI	Bilinear-Lossless-Discrete-Integrator
CSD	Canonical Signed Digit
DCGA	Diversity-Controlled Genetic Algorithm
DE	Differential Evolution
DFG	Data Flow Graph
EMQF	Elliptic Minimum Q-Factor
FIR	Finite Impulse Response
FRM	Frequency Response Masking
GA	Genetic Algorithm
IA	Immune Algorithm
IIR	Infinite Impulse Response
LUT	Look-Up Table
PSO	Particle Swarm Optimization
RTL	Register-Transfer Level
SA	Simulated Annealing
SOA	Seeker Optimization Algorithm
SPT	Signed Power-of-Two
TS	Tabu Search
VLSI	Very-Large-Scale Integration
WD	Wave Digital

Chapter 1

Introduction

Digital filters find wide variety of applications in modern digital signal processing systems [1, 2]. As a result of the recent progress in such systems, there is an ever growing demand for sharp transition band digital filters. These narrow transition bandwidth digital filters are usually designed by using the frequency response masking (FRM) approach [3]. The computational efficiency of the FRM technique makes it suitable for different applications, e.g. in audio signal processing and data compression [4].

Practical design of digital filters is based on optimization for satisfying the given design specifications together with the hardware architecture. However, the optimization may be carried out in terms of fixed configurations but variable multiplier coefficient values. On the other hand, the problem may concern the optimization of the hardware architecture without taking the multiplier coefficient values into consideration.

In order to optimize the given design specifications, the multiplier coefficient values can be determined in infinite precision by using hitherto optimization techniques. However, in an actual hardware implementation of the digital filters, the infinite precision multipliers should be quantized to their finite precision counterparts, but these finite precision multiplier coefficients may no longer satisfy the given design specifications. Consequently, from a hardware implementation point of view, there is a need for finite precision optimization techniques, capable of finding the optimized digital filter rapidly while keeping the computational complexity at a desired level. In principle, there exist two different techniques for the optimization of

digital filters, namely, gradient-based and heuristic optimization approaches.

Gradient-based optimization techniques have been studied widely. In [5], an integer programming technique was developed for the optimization of digital filters over a discrete multiplier coefficient space. In [6], a Remez exchange algorithm was used for the optimization of FRM finite impulse response (FIR) digital filters and it was shown that this algorithm may provide a speed advantage over the linear programming approach. However, both these techniques suffer from sub-optimality problems. In [7], unconstrained weighted least-squares criterion was used to develop another technique for the optimization of digital filters. Convex optimization approaches such as semi-definite programming [8] and second-order cone programming [9] have also been applied to the optimization of digital filters. However, if a large number of constraints are present, these optimization techniques may become computationally inefficient in terms of time consumption and speed.

Heuristic optimization algorithms have emerged as promising candidates for the design and discrete optimization of digital filters, particularly due to the fact that they are capable of automatically finding near-optimum solutions while keeping the computational complexity of the algorithm at moderate levels. Simulated annealing (SA) and genetic algorithms (GAs) were widely used in the design and optimization of digital filters [10–12]. Particle swarm optimization (PSO) and seeker optimization algorithm (SOA) are two newly developed algorithms suitable for the optimization of various digital filters due to their few number of implementation parameters and high speed of convergence [13, 14]. It was shown that SOA has advantages over PSO in terms of the speed of convergence and global search ability [15]. Tabu search (TS) [16], ant colony optimization (ACO) [17], immune algorithm (IA) [18] and differential evolution (DE) [19, 20] are alternative candidates for the optimization of digital filters. All the foregoing techniques allow a robust search of the solution space through a parallel search in all directions without any recourse to gradient information. However, the aforementioned techniques were developed for infinite precision optimization of digital filters which require the user to perform a quantization step for a hardware implementation.

In [21–23], a technique was developed for finite-precision design and opti-

mization of FRM digital filters using GAs. finite-precision optimization of FRM FIR digital filters using PSO was studied in [24, 25] and finite-precision optimization of infinite impulse response based (IIR-based) FRM digital filters was studied in [26, 27]. In this thesis, PSO of FRM FIR and FRM IIR digital filters suitable for direct very-large-scale integration (VLSI) hardware implementation is studied. PSO was originally proposed by Kennedy and Eberhart in 1995 as a new intelligent optimization algorithm which simulates the migration and aggregation of a flock of birds seeking food [28]. It adopts a strategy based on particle swarm and parallel global random search, that may exhibit superior performance to other intelligent algorithms in computational speed and memory. In PSO, a potential candidate solution is represented as a particle in a multidimensional search space, where each dimension represents a distinct optimization variable. The particles in the multidimensional search space are characterized by corresponding fitness values. They make movements in the search space towards regions characterized by high fitness values.

The conventional FRM digital filters incorporate FIR interpolation digital subfilters. These digital subfilters are usually of high orders, rendering the resulting overall FRM digital filters as not economical, since the resulting digital filters occupy large chip areas and consume high amounts of power in their VLSI hardware implementations. In general, the multiplication operation is the most cost-sensitive part in such an implementation. Therefore, there is every incentive to reduce the number of multiplication operations in the digital filter realization. This problem may be circumvented by employing IIR interpolation digital subfilters [29, 30].

There is a vast body of literature available for the design and optimization of digital IIR filters [31–33]. However, all the aforementioned designs are based on the exact transfer function coefficients which leads to an uneconomical hardware realization of such filters. This thesis employs the actual multiplier coefficients for the direct VLSI hardware implementation as design and optimization variables. In order to realize the constituent IIR interpolation digital subfilters on a hardware platform, the bilinear-lossless-discrete-integrator (bilinear-LDI) digital filter design approach is employed [34]. These digital subfilters are realized as a sum/difference of

a pair of bilinear-LDI digital allpass networks. The salient features of the bilinear-LDI digital filters are that they lend themselves to fast two-cycle parallel digital signal processing speeds, while being minimal in the number of digital multiplication operations (and, practically, minimal in number of digital addition and unit-delay operations).

The starting point in the design of FRM digital filters is to find the multiplier coefficients constituent in the FRM digital filter in infinite precision by using the hitherto gradient-based optimization techniques (e.g. Parks-McClellan approach [35] for FIR digital filters) followed by a quantization step. The quantization can be performed by constraining the multiplier coefficients values to conform to certain number systems such as the signed power-of-two (SPT) system. SPT is a computationally efficient number system which can further reduce the hardware complexity of the FRM IIR digital filters. In this number system, each multiplier coefficient is represented with only a few non-zero bits within its wordlength, permitting the decomposition of the multiplication operation into a finite series of shift and add operations. Digital filters incorporating SPT multiplier coefficient representation are commonly referred to as *multiplierless* digital filters [36]. However, the SPT representation of a given number is not unique, resulting in redundancy in the multiplier coefficient representation. This redundancy can adversely affect the corresponding computational complexity due to recourse to compare operations repetitively.

The canonical signed digit (CSD) number system is a special case of the SPT number system which circumvents the above redundancy problem by limiting the number of non-zero bits in the representation of the multiplier coefficients. It is usually used in combination with subexpression sharing and elimination, which in turn results in substantial reduction in the cost of the VLSI hardware implementation of the digital filters [37]. In CSD number system, no two (or more) non-zero bits can appear consecutively in the representation of the multiplier coefficients, reducing the maximum number of non-zero bits by a factor of two in terms of shift and add operations [38].

After multiplier coefficient quantization, the resulting FRM digital filter may no longer satisfy the given target design specifications. Therefore, the next step in the

design of FRM digital filters is to perform a further optimization to make the finite precision FRM digital filter to conform to the design specifications. This can be achieved by resorting to a finite-precision optimization technique such as PSO.

A direct application of the conventional PSO algorithm to the optimization of the above FRM digital filters gives rise to three separate problems:

- The first problem arises because in the course of optimization, the multiplier coefficient update operations lead to values that may no longer conform to the desired CSD wordlength, etc. (due to random nature of velocity and position of particles). This problem is resolved by generating indexed look-up tables (LUTs) of permissible CSD multiplier coefficient values, and by employing the indices of LUTs to represent FRM digital filter multiplier coefficient values.

- The second problem stems from the fact that in case of FRM IIR digital filters, the resulting FRM IIR digital filters may no longer be bounded-input-bounded-output (BIBO) stable. This problem can be resolved by generation and successive augmentation of template LUTs until the BIBO stability constraints remain satisfied [23].

- Finally, the third problem arises because even in case of having indexed LUTs, the particles may go over the boundaries of LUTs in course of optimization (due to the inherent limited search space). This can be resolved by introducing *barren layers*. A barren layer is a region, with a certain width and certain entries, which is added to the problem space such that the particles tend to shy away from such a region. The width of the barren layers is calculated based on a worst case scenario that may happen in the particles movements in the search space. However, the entries of barren layers are different for different problems and depend on the topology of the search space and the fitness function used in the problem.

The field of high-level synthesis has gained a great deal of recognition on the part of digital system designers during the past decades [39–49]. The process of high-level synthesis usually begins with a behavioral description of the required digital system together with a set of user-specified time and/or area constraints.

The goal of high-level synthesis is to produce a corresponding register-transfer level (RTL) implementation that satisfies the given constraints [50–52]. This implementation includes the data-path as well as the hardware to control the constituent data transfers. The resulting data-path itself includes a network of arithmetic functional units, registers, multiplexors, and buses.

The process of high-level synthesis consists of three main steps. The first step concerns the translation of the digital system specifications into a corresponding data flow graph (DFG) . The second step, referred to as scheduling, involves the assignment of the DFG operations to various time-steps. Finally, the third step, referred to as data path synthesis, concerns the binding of the DFG operations to physical arithmetic units (digital adders, digital multipliers, etc.), as well as the subsequent allocation of hardware resources (registers, multiplexors, etc.) to facilitate the data transfer in the DFG.

There are a number of approaches available for the scheduling of digital systems. The as soon as possible (ASAP) and the as late as possible (ALAP) scheduling techniques are the most well known amongst these approaches. In the absence of any constraints on the hardware resources, the ASAP scheduling leads to the fastest possible schedule, generating the minimum number of time-steps. However, the main disadvantage of this scheduling technique is that it leads to high hardware requirements. In [44] and [45], this problem was resolved through a conditional postponement of the operations. In [47], force directed scheduling technique was developed in an attempt to distribute the operations evenly over the various time-steps. The latter technique is global both in the way it selects the next operation to be scheduled and in the way it decides on the time-step to place this operation, although it performs the scheduling and allocation tasks separately. In [43], the scheduling and allocation tasks are carried out simultaneously, where the scheduling is performed based on the critical path in the DFG, and the register allocation is performed by the program proposed in [53]. In [54], a program based on list scheduling is proposed, which employs DFG re-timing to generate the shortest critical path solution.

In [40], a technique was developed for the synthesis of pipelined data-paths,

resulting in an area-optimal, time-optimal or combined area-optimal-cum-time-optimal design. A high-level hardware compiler was developed in [44] to find the parallelism in the behavioral specifications in order to produce a fast-running implementation that meets user-specified constraints. An integer linear programming approach was proposed in [41]. Although this approach could find the global optimum solution, the algorithm was exponential with regard to the computational time, rendering it impractical for large design problems. An iterative algorithm was developed in [49], which could escape from the local minima in a polynomial computational time. Percolation based scheduling [48] starts with an optimal schedule and applies transformation methods to maximize parallelism.

Genetic algorithms have been used for scheduling problems and high-level synthesis of digital filters [55–58]. As discussed before, these techniques permit a fast exploration of the design space in an attempt to find a global optimum. PSO can also be utilized for the application of scheduling [59–61] and high level synthesis of digital filters.

This thesis is concerned with the application of PSO to the development of a new optimization technique for the high-level synthesis of digital filter data paths. This optimization technique is concerned with the minimization of the cost associated with the final digital filter for obtaining global area-optimal, time-optimal, or combined area-optimal-cum-time-optimal data paths subject to user-specified constraints on the number of physical arithmetic functional units employed. The optimization is made computationally effective by encoding the digital filter DFG into particles which preserve the data-dependency relationships in the original digital filter signal flow graph under the operations of addition and subtraction by the underlying PSO algorithm.

This chapter proceeds as follows. Section 1.1 provides a brief overview of the conventional PSO algorithm and explains the recent developments of the PSO algorithm together with the advantages and disadvantages of each of these algorithms. Finally, section 1.2 provides a brief summary of this chapter.

1.1 PSO Algorithm

In this section, the conventional PSO algorithm is studied and the recent developments of PSO algorithm is briefly introduced.

1.1.1 The Conventional PSO Algorithm

Let us consider an optimization problem consisting of N design variables, and let us refer to each solution as a particle. Let us further consider a swarm of K particles in the N-dimensional search space. The position of the k-th particle in the search space can be assigned a N-dimensional position vector $X_k = \{x_{k1}, x_{k2}, \ldots, x_{kN}\}$. In this way, the element x_{kj} (for $j = 1, 2, \ldots, N$) represents the j-th coordinate of the particle X_k.

The PSO optimization fitness function maps each particle X_k in the search space to a fitness value. In addition, the particle X_k is assigned a N-dimensional velocity vector $V_k = \{v_{k1}, v_{k2}, \ldots, v_{kN}\}$. The PSO optimization search is directed towards promising regions by taking into account the velocity vector V_k together with the best previous position of the k-th particle $X_{best_k} = \{x_{best_{k1}}, x_{best_{k2}}, \ldots, x_{best_{kN}}\}$, and the best global position of the swarm $G_{best} = \{g_{best_1}, g_{best_2}, \ldots, g_{best_N}\}$ (i.e. the location of the particle with the best fitness value).

The conventional PSO is initialized by spreading the particles X_k through the search space in a random fashion. Then, the particles make movements through the search space towards regions characterized by high fitness values with corresponding velocities V_k. The movement of each particle is governed by the best previous location of the same particle X_{best_k}, and by the global best location G_{best}. The velocity of particle movement is determined from the previous best location of the particle, the global best location, and the previous velocity.

The velocity and position of each particle in the i-th iteration throughout the

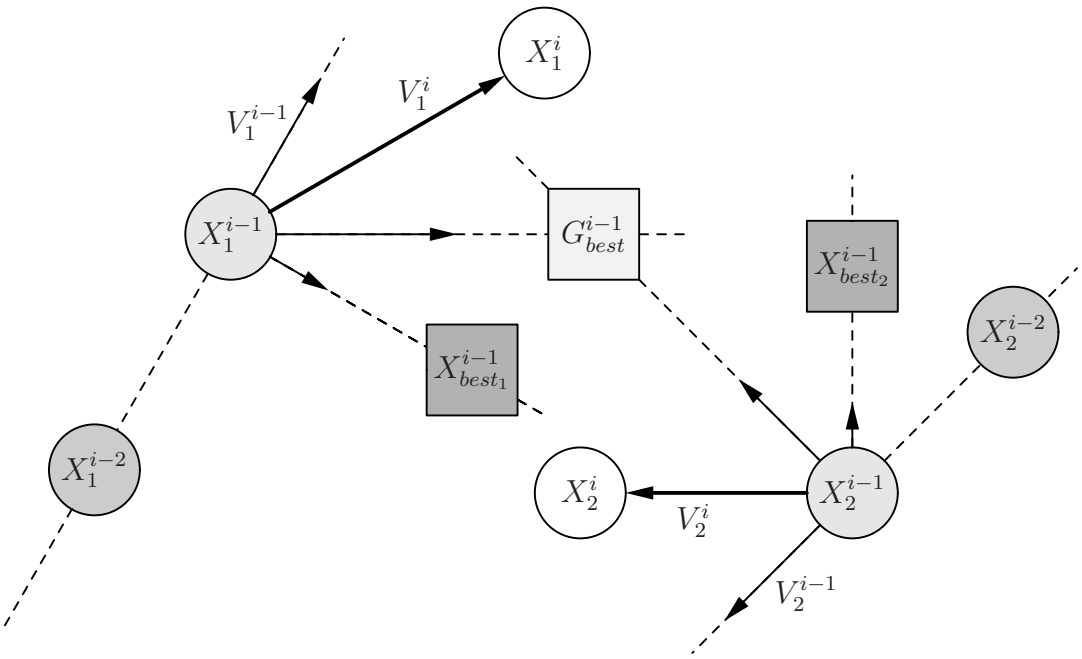

Figure 1.1: Movement of Particles in PSO Algorithm

course of PSO are updated in accordance with the equations:

$$v_{kj}^i = wv_{kj}^{i-1} + c_1 r_1 (x_{best_{kj}}^{i-1} - x_{kj}^{i-1}) + c_2 r_2 (g_{best_j}^{i-1} - x_{kj}^{i-1}) \tag{1.1}$$

$$\text{if} \quad v_{kj}^i < v_{min} \quad ; \quad v_{kj}^i = v_{min}$$

$$\text{if} \quad v_{kj}^i > v_{max} \quad ; \quad v_{kj}^i = v_{max}$$

$$x_{kj}^i = x_{kj}^{i-1} + v_{kj}^i \tag{1.2}$$

The parameter w represents an inertia weight; c_1 and c_2 are the correction (learning) factors, and r_1 and r_2 are random numbers in the interval $[0, 1]$. The velocity is limited between v_{min} and v_{max} to avoid very large particle movements in the search space, where $v_{min} < 0$ and $v_{max} > 0$. Fig. 1.1 illustrates how the particles move in a two-dimensional search space ($N = 2$). In this figure, two particles are present in the swarm, i.e. $K = 2$.

The first term in the right hand side of movement update Eqn. (1.1), weighted by w, signifies the dependence of the current particle velocity on its value in the previous iteration. The second term, weighted by c_1, signifies an attractor to pull the particle towards its previous best position. The third term, weighted by c_2 controls

the movement of the particle towards the global best position.

In addition to the update Eqns. (1.1) and (1.2), one can limit the coordinates in a particle between two user defined values $x_{j_{min}}$ and $x_{j_{max}}$ in order to limit the search space. However, This operation increases the complexity and consumes time.

1.1.2 Literature Survey on PSO

Due to its simple implementation in both software and hardware platforms, the PSO has become a popular optimization technique and has been widely applied to solve many practical problems. Therefore, there is a vast body of literature concerning the performance improvements of the PSO algorithm and many theoretical studies have been made to illustrate the importance and effectiveness of PSO. Convergence analysis and stability studies have been reported in [62–65]. Much research on performance improvements has been reported, including parameter studies, combination with auxiliary operations, and topological structures [66, 67]. There are three basic parameters involved in the conventional PSO algorithm, namely, the inertia weight w and the correction factors c_1 and c_2. The inertia weight w in Eqn. (1.1) was introduced in [68] which was forced to linearly decrease with the iterative generations. In this way, in the first iterations, PSO is more likely to search globally through the entire search space, while as the algorithm proceeds to its final iterations, the PSO searches more precisely in spaces with better fitness values to find optimal solutions. A fuzzy adaptive w was proposed in [69] to dynamically adapt the inertia weight on the population level. A random version setting w was experimented in [70] for dynamic system optimization. The expected value which can be defined for this random w is based on the idea as in Clerc's constriction factor [62, 71]. The constriction factor is equivalent to the inertia weight mathematically, as Eberhart and Shi mentioned in [72].

The correction factors c_1 and c_2 are also important parameters in PSO as shown via experiments in Kennedy's two extreme cases [73]. Kennedy and Eberhart [28] suggested a fixed value of 2, and this configuration has been adopted by many other researchers. In [74], it was shown that using ad hoc values of c_1 and c_2 instead of a fixed value of 2 for different problems could yield better performance. In [75], a

PSO algorithm was proposed with linearly time-varying correction factors, where a larger c_1 and a smaller c_2 were set at the beginning and were gradually reversed during the search. Among these three methods, the last one with linearly time-varying correction factors shows the best overall performance. This may be owing to the time-varying c_1 and c_2 that can balance the global and local search abilities, which implies that adaptation of c_1 and c_2 can be promising in enhancing the PSO performance.

Another active research trend in PSO is hybrid PSO, which combines PSO with other evolutionary paradigms. In [76], a selection operation similar to that in a GA was first introduced into PSO. Hybridization of GA and PSO has been used in [77] for recurrent artificial neural network design. GA operators have been also combined with PSO algorithm to obtain better results. In [76], selection operation was used to improve PSO. In [78], a recombination method with dynamic linkage discovery in GA was employed to obtain the PSO-RDL algorithm. Mutation was also utilized to improve the search ability of PSO [79]. In [80], a breeding operator was incorporated into the PSO algorithm, where breeding occurred inline with the standard velocity and position update rules. In addition, other techniques such as local search [81] and differential evolution [82] have been used to combine with PSO. Self-organizing hierarchical technique [75], cooperative approach [83], deflection, stretching, and repulsion techniques [84] have also been combined with conventional PSO to enhance performance. Inspired by biology, some researchers introduced niche [85], and speciation [86] techniques into PSO. These technique are capable of keeping the particles away from each other in the course of optimization and they can locate as many optimal solutions as possible.

PSO topological structures are also widely studied. The LPSO with a ring topological structure and the Von Neumann topological structure PSO have been proposed in [87, 88] to enhance the performance in solving multi-modal problems. Dynamically changing neighborhood structures have been proposed in [74, 89, 90] to avoid the drawbacks of fixed neighborhoods. Moreover, in the fully informed particle swarm algorithm [91], the information of the entire neighborhood is used to guide the particles. The CLPSO in [92] allows the particle use different previous

best positions to update its flying on different dimensions for improved performance in multi-modal applications.

The remainder of this thesis is organized as follows. Chapter 2 is concerned with the development of a novel discrete PSO to the design and optimization of FRM digital filters. Two different techniques have been employed: FRM FIR digital filters consisting FIR interpolation digital subfilters, and FRM IIR digital filters having bilinear-LDI IIR interpolation digital subfilters. This chapter starts with an introduction to the conventional FRM technique and proceeds with introducing bilinear-LDI design approach. Then, CSD number system is explained, and finally, the design procedure for the optimization of FRM digital filters is proposed.

Chapter 3 discusses a novel PSO algorithm for the high-level synthesis of digital filter data-paths. the optimization is such that the cost function of the digital filter data-paths is minimized. This cost function is derived by taking into account both the time constraints and the hardware requirements associated with the digital filter data-paths. The constrained optimization is carried out in a way that there is no functional unit violation in the course of high-level synthesis of digital filter data-paths using PSO. This chapter starts with a brief introduction to the high-level synthesis of digital filter data-paths. Then, the particle formation in the course of PSO is explained and the constrained PSO for high-level synthesis of digital filter data-paths is proposed. Finally, the cost function associated with the digital filter data-paths is derived by taking into account the time constraints, the number of functional units, and the number of support cells that are required for the implementation of digital filter data-paths.

Chapter 4 presents several examples to illustrate the application of PSO to the design, high-level synthesis and optimization of digital filters. A pair of lowpass FRM digital filters are designed and optimized using the proposed PSO, one of which utilizing the conventional FIR interpolation digital subfilters, and the other having bilinear-LDI IIR interpolation digital subfilters. In addition, the application of PSO to the design and optimization of bandpass FRM FIR and bandpass FRM IIR digital filters is illustrated through two different examples. The optimization is subject to satisfying stringent design specifications. Finally, the proposed PSO is

used for the high-level synthesis of a benchmark fifth order elliptic filter.

Finally, chapter 5 draws the main conclusions of the thesis. A summary of the contributions made in this thesis is presented and suggestions for future work are explained.

1.2 Summary

This chapter has presented an introduction to a very popular optimization technique, called particle swarm optimization. This optimization technique is known to be effective in optimizing various functions which are highly nonlinear, multivariate, and multimodal. Due to independence of any gradient information of the objective function, PSO can find optimal solutions in a given search space fast and easily. PSO is applied to a class of digital filters which are suitable for narrow transition band designs. This optimization is carried out over a CSD number system to reduce the hardware implementation of digital filters. Two categories of such digital filters are introduced, namely, conventional FRM FIR digital filters, and FRM IIR digital filters incorporating IIR digital subfilters realized as a combination of two digital allpass networks that are implemented using bilinear-LDI design approach.

There are three problems in the PSO of FRM digital filters over the CSD multiplier coefficient space. Firstly, due to random nature of PSO operations of addition and subtraction, PSO may result in multiplier coefficient values that are not conformed to the CSD number format. Secondly, in case of FRM IIR digital filter design, PSO may result in candidate FRM IIR digital filters which are not BIBO stable. Thirdly, since the search space is limited, the particles may go outside the boundaries of the search space in the course of PSO. This thesis presents solutions to the above three problems.

In addition, PSO is applied for the high-level synthesis of digital filter data-paths. In this technique, the cost function associated with the digital filter data-path is minimized to obtain a time-optimal, area-optimal, or combined time-cum-area-optimal solution. An encoding scheme is used to ensure that the data-dependency relationships present in the DFG of the digital filter remain satisfied under the op-

erations of addition and subtraction by the underlying PSO algorithm.

Chapter 2

Design and PSO of FRM Digital Filters

This chapter discusses in detail the design, realization and discrete optimization of FRM FIR and FRM IIR digital filters. FRM FIR digital filters consist of FIR masking digital subfilters and FIR digital interpolation subfilters, while FRM IIR digital filters are designed by FIR masking digital subfilters together with IIR interpolation digital subfilters. The FIR filter design is straightforward and can be performed by using hitherto techniques. The IIR digital subfilter design topology consists of a parallel combination of a pair of allpass networks such that its magnitude-frequency 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 coefficient values represented as finite-precision CSD numbers.

The above FRM digital filters are optimized over the discrete multiplier coefficient space, resulting in FRM digital filters which are capable of direct implementation in digital hardware platform without any need for further optimization. A new PSO algorithm is developed to tackle three different problems. In this PSO algorithm, a set of indexed LUTs of permissible CSD multiplier coefficient values is generated to ensure that in the course of optimization, the multiplier coefficient update operations constituent in the underlying PSO algorithm lead to values that are guaranteed to conform to the desired CSD wordlength, etc. In addition, a general set of constraints is derived in terms of multiplier coefficients to guarantee that the IIR bilinear-LDI interpolation digital subfilters automatically remain BIBO stable

throughout the course of PSO algorithm. Moreover, by introducing barren layers, the particles are ensured to automatically remain inside the boundaries of LUTs in course of optimization.

This chapter proceeds as follows. Section 2.1 is concerned with a brief introduction to the conventional FRM design approach. Section 2.2 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 2.3 outlines the special type of elliptic filters used for the design of FRM digital filters. Section 2.4 presents the design methodology for implementing an allpass network using the bilinear-LDI approach. Section 2.5 provides an introduction to CSD number system and the corresponding quantization error. Section 2.6 introduces a set of stability constraints that guarantee the BIBO stability of digital filters described in Section 2.4. Section 2.7 presents a novel PSO algorithm that allows the optimization to search through the CSD multiplier coefficient space automatically, while maintaining BIBO stability of every particle throughout the optimization process. Section 2.8 outlines in detail the design and PSO of FRM FIR digital filters and FRM bilinear-LDI based IIR digital filters. Finally, Section 2.9 provides a summary of this chapter.

2.1 The Conventional FRM Design Approach

2.1.1 Design of Lowpass FRM Digital Filters

The block diagram in Fig. 2.1 shows a conventional FRM digital filter, where $H_a(z)$ represents a FIR interpolation lowpass digital subfilter, and where $H_b(z)$ represents a power complementary counterpart of $H_a(z)$ in accordance with

$$|H_a(e^{j\omega})|^2 + |H_b(e^{j\omega})|^2 = 1 \tag{2.1}$$

Here, z represents the discrete-time complex frequency, and ω represents the corresponding (normalized) real frequency variable. Moreover, $F_0(z)$ and $F_1(z)$ represent FIR masking digital subfilters, while $H_a(z^M)$ and $H_b(z^M)$ represent M-fold interpolated versions of $H_a(z)$ and $H_b(z)$, respectively. In case of FIR digital

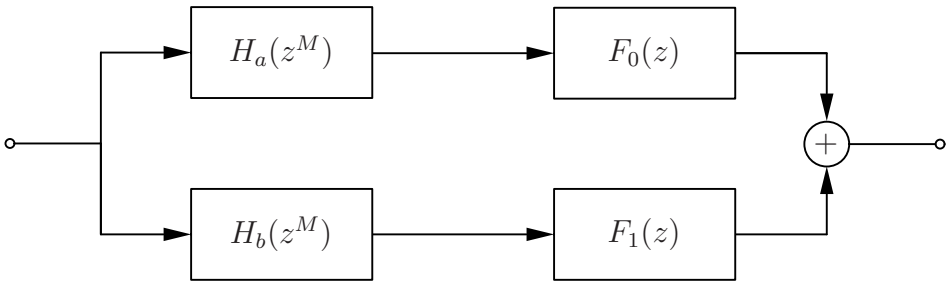

Figure 2.1: FRM Digital Filter Block Diagram

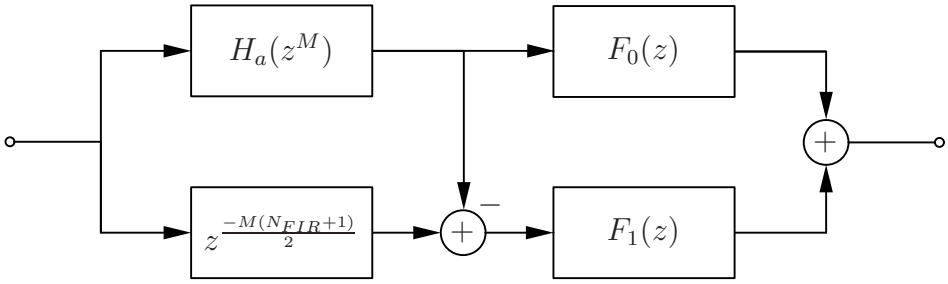

Figure 2.2: Block Diagram Representation of Frequency-Response Masking

interpolation subfilters, for a linear-phase filter $H_a(z)$ of order N_{FIR}, the relationship between $H_b(z)$ and $H_a(z)$ is as follows:

$$H_b(z) = z^{(N_{FIR}+1)/2} - H_a(z) \tag{2.2}$$

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

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

$$H(z) = H_a(z^M)F_0(z) + H_b(z^M)F_1(z) \tag{2.3}$$

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 $H_a(z^M)$ and $H_b(z^M)$. The masking filters are made to have equal order (by zero padding) in order to ensure that their phase characteristics are similar. The corresponding interpolated digital subfilters $H_a(z^M)$ and $H_b(z^M)$ can realize transition bands which are a factor of M sharper than those of $H_a(z)$ and $H_b(z)$, without increasing the number of required non-zero digital multipliers. The magnitude frequency-response of the various subfilters incorporated by the FRM digital filter design approach are shown in Fig. 2.3.

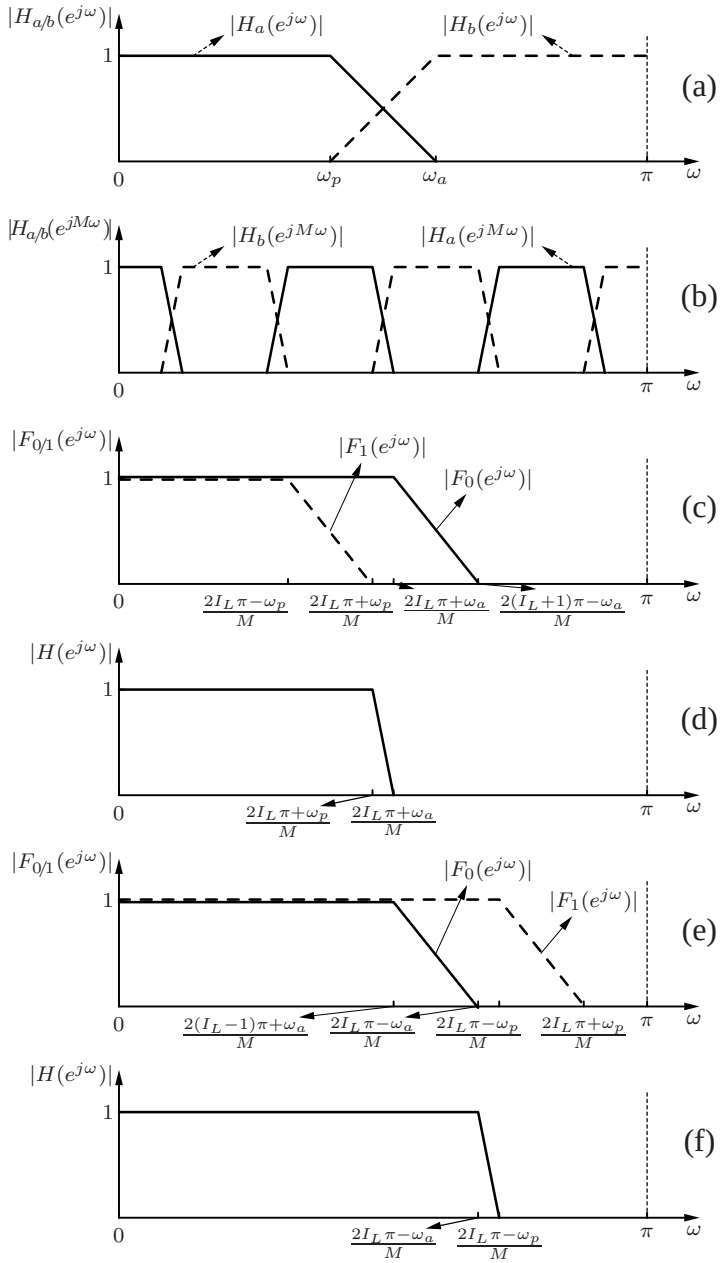

Figure 2.3: Magnitude Frequency-Response of FRM Digital Filter. (a) Magnitude Frequency-Response of the Bandedge-Shaping Digital Subfilters $H_a(z)$ and $H_b(z)$. (b) Magnitude Frequency-Response of the M-Interpolated Complementary Digital Subfilters $H_a(z^M)$ and $H_b(z^M)$. (c) Magnitude Frequency-Response of the Masking Digital Subfilters $F_0(z)$ and $F_1(z)$ for Case I. (d) Magnitude Frequency-Response of the Overall FRM Digital Filter $H(z)$ for Case I. (e) Magnitude Frequency-Response of the Masking Digital Subfilters $F_0(z)$ and $F_1(z)$ for Case II. (f) Magnitude Frequency-Response of the Overall FRM Digital Filter $H(z)$ for Case II [3].

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

	Filter	Passband Edge	Stopband Edge
Case I	$H(z)$	$\frac{2I_L\pi + \omega_p}{M}$	$\frac{2I_L\pi + \omega_a}{M}$
	$F_0(z)$	$\frac{2I_L\pi + \omega_a}{M}$	$\frac{2(I_L+1)\pi - \omega_a}{M}$
	$F_1(z)$	$\frac{2I_L\pi - \omega_p}{M}$	$\frac{2I_L\pi + \omega_p}{M}$
Case II	$H(z)$	$\frac{2I_L\pi - \omega_a}{M}$	$\frac{2I_L\pi - \omega_p}{M}$
	$F_0(z)$	$\frac{2(I_L-1)\pi + \omega_a}{M}$	$\frac{2I_L\pi - \omega_a}{M}$
	$F_1(z)$	$\frac{2I_L\pi - \omega_p}{M}$	$\frac{2I_L\pi + \omega_p}{M}$

Here, Case I design is when the transition band of $H(z)$ is extracted from that of $H_a(z^M)$ and Case II design is when the transition band of $H(z)$ is extracted from that of $H_b(z^M)$. The edge frequencies of the overall digital FRM filter and its constituent subfilters are given in Table 2.1, where I_L represents the number of image lobes to be masked given by:

$$I_L = \begin{cases} \left\lfloor \frac{M\omega_p}{2\pi} \right\rfloor & \text{Case I} \\ \left\lceil \frac{M\omega_a}{2\pi} \right\rceil & \text{Case II} \end{cases} \tag{2.4}$$

where $\lfloor\ \rfloor$ denotes the largest integer from the lower side, and $\lceil\ \rceil$ signifies the smallest integer from the upper side.

2.1.2 Design of Bandpass FRM Digital Filters

In general, it is possible to extend the conventional FRM approach for the design of bandpass or bandstop FRM digital filters. However, the resulting FRM digital filters are constrained to have identical lower and upper transition bandwidths. In [93], this restriction was relaxed by realizing the bandstop FRM FIR digital filter as a

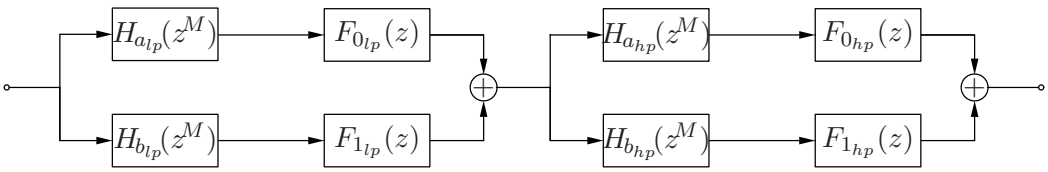

Figure 2.4: Bandpass FRM Digital Filter Block Diagram

parallel combination of a corresponding pair of lowpass and highpass FIR digital filters. The latter lowpass and highpass FRM digital filters were obtained using a variation of the conventional FRM approach.

Let the desired bandpass FRM digital filter $H(z)$ have a lower transition bandwidth which is not identical to its upper transition bandwidth. $H(z)$ can be realized as a cascade combination of a pair of lowpass and highpass FRM digital filters, so that

$$H(z) = H_{lp}(z)H_{hp}(z) \tag{2.5}$$

where $H_{lp}(z)$ represents a lowpass and $H_{hp}(z)$ represents a highpass FRM digital filter. In this way, $H_{lp}(z)$ and $H_{hp}(z)$ can be obtained with the help of Eqn. (2.3) as

$$H_{lp}(z) = H_{a_{lp}}(z^M)F_{0_{lp}}(z) + H_{b_{lp}}(z^M)F_{1_{lp}}(z) \tag{2.6}$$

$$H_{hp}(z) = H_{a_{hp}}(z^M)F_{0_{hp}}(z) + H_{b_{hp}}(z^M)F_{1_{hp}}(z) \tag{2.7}$$

The lower transition bandwidth is governed by the constituent transition bandwidth of the highpass FRM digital filter, while the upper transition bandwidth is governed by the constituent transition bandwidth of the lowpass FRM digital filter. The realization for bandpass FRM digital filter are as shown in Fig. 2.4.

2.2 Design of FRM Digital Filters Incorporating IIR Interpolation Digital Subfilters

In the case of FRM IIR digital filters, $H_a(z)$ and $H_b(z)$ (in section 2.1) act as IIR interpolation digital subfilters. The masking filters $F_0(z)$ and $F_1(z)$ are not changed (i.e. they are still equal order FIR digital filters). Therefore, Eqn. (2.3) is still valid for the FRM IIR digital filter.

The IIR interpolation digital subfilter $H_a(z)$ is chosen to have an odd order N_{IIR}. Odd-ordered elliptic transfer functions can be represented as a sum of or difference between two allpass transfer functions [94]. Therefore, $H_a(z)$ can be realized as the addition of two allpass digital networks $G_0(z)$ and $G_1(z)$ as follows:

$$H_a(z) = \frac{G_0(z) + G_1(z)}{2} \tag{2.8}$$

where $G_0(z)$ is odd-ordered and $G_1(z)$ is even-ordered. The interesting fact is 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 subsequently be used as the power complementary interpolation digital subfilter $H_b(z)$ as in the following:

$$H_b(z) = \frac{G_0(z) - G_1(z)}{2} \tag{2.9}$$

It can be easily verified that $H_a(z)$ and $H_b(z)$ are power complementary digital filters [29], i.e. they satisfy Eqn. (2.1). In addition, it is well known that this structure halves the number of multiplier coefficients required for the implementation of FRM digital filters and therefore is the most economical realization since it requires a total of only N_{IIR} multiplier coefficients to realize both $H_a(z)$ and $H_b(z)$. The overall transfer function of $H(z)$ given by Eqn. (2.3) can be expressed as:

$$H(z) = \frac{G_0(z^M) + G_1(z^M)}{2} F_0(z) + \frac{G_0(z^M) - G_1(z^M)}{2} F_1(z) \tag{2.10}$$

The block diagram in Fig. 2.5 shows the IIR interpolation digital subfilters $H_a(z)$ and $H_b(z)$ realized as a parallel combination of two allpass networks. It should be noted that if $H_a(z)$ is a lowpass filter, $H_b(z)$, which is the power complementary of $H_a(z)$, is a highpass filter. Fig. 2.6 shows an overall FRM IIR digital filter realization.

One may rearrange the structure in Fig. 2.6 by using Eqns. (2.8-2.9). This can be performed by defining two digital subfilters as follows:

$$A(z) = \frac{F_0(z) + F_1(z)}{2} \tag{2.11}$$

$$B(z) = \frac{F_0(z) - F_1(z)}{2} \tag{2.12}$$

Then $H(z)$ in Eqn. (2.10) simplifies to:

$$H(z) = G_0(z^M)A(z) + G_1(z^M)B(z) \tag{2.13}$$

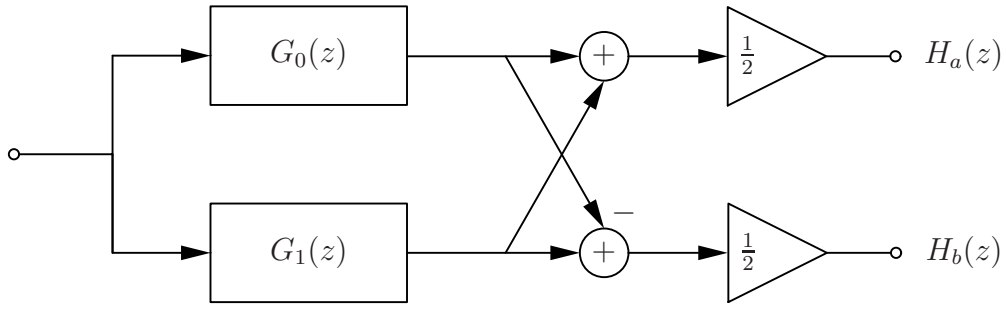

Figure 2.5: Block Diagram of Interpolation and Complementary Filters as a Parallel Combination of Two Allpass Networks

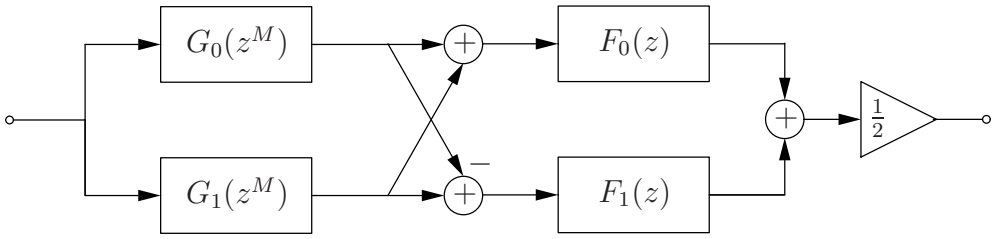

Figure 2.6: FRM Digital Filter Realization in Terms of Allpass Digital Networks $G_0(z)$ and $G_1(z)$

Fig. 2.7 shows the block diagram representing Eqn. (2.13).

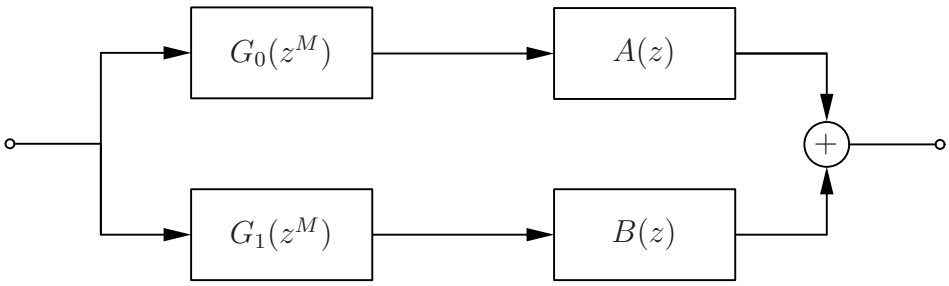

Figure 2.7: Alternative Structure of the Overall FRM IIR Digital Filter

The advantage of realizing the FRM IIR digital filter as shown in Fig. 2.7 is that two adders shown in Fig. 2.6 are removed and they are no longer required. This subsequently simplifies the hardware implementation of the overall FRM IIR digital filter. However, it should be noted that the FIR masking digital subfilters $F_0(z)$ 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. 2.7 is offset by a greater number of non-zero multiplier

coefficients required in the realization of FRM IIR digital filters.

2.3 Realization of IIR Interpolation Digital Subfilters Using Elliptic Filters with Minimum Q-factor (EMQF)

Bilinear-LDI transformation falls into the category of digital filter realization techniques that transform an analog reference filter to its digital counterpart. Therefore, in order to determine the multiplier coefficient values of the IIR interpolation digital subfilters $H_a(z)$ and $H_b(z)$ constituent in the FRM IIR digital filter, a suitable analog reference filter $H_a(s)$ and its power complementary analog filter $H_b(s)$ have to be determined, 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 2.4).

EMQF filters have several advantages for the design of FRM IIR digital filters. The squared ripple in the passband region of $H_a(z)$ and the squared ripple in the stopband region of $H_b(z)$ are equal as indicated by Eqn. (2.1). On the other hand, the squared ripple in the stopband region of $H_a(z)$ and the squared ripple in the passband region of $H_b(z)$ are equal. In addition, depending on whether the design specifications require a Case I or Case II FRM technique, either $H_a(z)$ or $H_b(z)$ could determine the maximum passband and stopband ripple of the overall FRM IIR digital filter $H(z)$. Consequently, the interpolation filter $H_a(z)$ is chosen to have equal passband and stopband squared tolerances. In this way, the resulting $H_b(z)$ also displays equal passband and stopband squared tolerances. These characteristics can be generalized for the analog reference subfilters $H_a(s)$ and $H_b(s)$. Therefore, there is a need for an analog reference filter $H_a(s)$ that together with its power complement $H_b(s)$ can exactly satisfy the passband and stopband relations in the FRM IIR filter. EMQF filters can successfully comply with the specifications present in the FRM IIR filter design. In addition, an EMQF transfer function can be easily designed by using bilinear-LDI transformation technique or any other structure consisting of two digital allpass networks in parallel. Furthermore, filters having EMQF transfer functions are minimally sensitive to component variations.

Despite all the advantages of EMQF filters, they suffer from not being able to independently specify passband and stopband ripples [95], [96] of the filter. Additionally, EMQF filters have exceedingly low passband attenuation.

All the poles of an EMQF transfer function reside on a circle in the s domain rendering them to have equal magnitudes. Given a squared passband and stopband tolerance of δ_p and δ_a, respectively, for an EMQF filter, the passband ripple Δ_p and minimum stopband attenuation Δ_a can be obtained as follows [97]:

$$\Delta_p = -10\log(1-\delta_p) \tag{2.14}$$

$$\Delta_a = -10\log(\delta_a) \tag{2.15}$$

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 vice versa, has to be taken into account in accordance with:

$$\Omega_A = \frac{2}{T}\tan(\frac{\omega_d T}{2}) \tag{2.16}$$

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

Once the transfer function of the analog reference filter $H_a(s)$ is determined, it is represented as a sum of two allpass analog filters $G_0(s)$ and $G_1(s)$. In addition, $H_b(s)$, which is the power complementary of $H_a(s)$ is represented as the difference of $G_0(s)$ and $G_1(s)$. The poles of $G_0(s)$ and $G_1(s)$ are determined by cyclically distributing the poles of the reference filter $H_a(s)$ [97]. In the next section, belinear-LDI design technique is used to transform the two allpass networks $G_0(s)$ and $G_1(s)$ into digital domain.

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

In this section, the design procedure in [34, 98] is briefly explained to design and implement digital filters $G_0(z)$ and $G_1(z)$ using the the bilinear-LDI approach. This

approach transforms analog reference filters $G_0(s)$ and $G_1(s)$ to obtain their digital filter counterparts $G_0(z)$ and $G_1(z)$.

The bilinear frequency transformation maps the analog frequency variable s to its digital domain counterpart z in accordance with:

$$s = \frac{2}{T}\frac{z-1}{z+1} \tag{2.17}$$

where T represents the sampling period, for mapping the transfer function of a prototype reference filter from the analog domain to the digital domain. The bilinear transform maps the left half of the complex s-plane to the interior of the unit circle in the z-plane. Therefore, BIBO stable filters in the s domain are converted to filters in the z domain which preserve that stability. Similarly, if the analog reference filter is minimum-phase, the previous characteristic of bilinear transform guarantees that the resulting digital filter is also minimum-phase. It also preserves the sensitivity properties of the analog reference filter. However, bilinear transform may result in a digital filter that has delay-free loops in its implementation. Unfortunately, delay-free loops prevent the implementation of a digital filter to be realizable in hardware platform.

The LDI frequency transformation ensures the absence of delay-free loops in the digital implementation and is given by

$$s = \frac{1}{T}\left(z^{\frac{1}{2}} - z^{-\frac{1}{2}}\right) \tag{2.18}$$

The LDI frequency transformation maps the hardware implementation of the analog reference filter to digital domain. While the LDI frequency transformation guarantees that there are no delay-free loops in the implementation of the digital filter, it does this to the cost of resulting in a digital filter having poor magnitude-frequency responses. Moreover, it is incapable of preserving the BIBO stability properties of the analog reference filter.

The bilinear-LDI approach is a combination of the two above mentioned realization techniques. In bilinear-LDI transform, a precompensation is performed to the reference analog filter. Then, the conventional LDI design technique is applied to a network resulting from the precompensated analog prototype filter. The precompensation is such that the application of the LDI design technique results in a

filter that exactly matches the bilinear frequency transform of the uncompensated analog prototype filter.

The resulting bilinear-LDI digital filters have several desirable features from a hardware realization point of view. They are minimal in the number of digital multiplication operations. Although they are not minimal in the number of digital adders and unit-delays, the additional adders and the additional unit delay lead to certain advantages when the concept of generalized delay unit is used for the realization of the network [34]. Moreover, The bilinear-LDI digital filters lend themselves to fast two-cycle parallel digital signal processing speeds and they exhibit exceptionally low passband sensitivity to their multiplier coefficient values, resulting in small coefficient wordlengths.

As discussed in Section 2.3, the analog reference filter $H_a(s)$ is decomposed into two allpass analog networks $G_0(s)$ and $G_1(s)$. The digital allpass networks $G_0(z)$ and $G_1(z)$ are obtained from $G_0(s)$ and $G_1(s)$ using the bilinear-LDI design approach.

It should be pointed out that $G_0(s)$ is an odd-ordered allpass function. Therefore, it has a pole on the real axis in the s domain. On the other hand, $G_1(s)$ ends up having an even-ordered allpass function. It is well known that an allpass transfer function can be written in the general form [34]:

$$G(s) = \frac{P(-s)}{P(s)} \tag{2.19}$$

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

$$P(s) = \text{Ev}P(s) + \text{Od}P(s) \tag{2.20}$$

where $\text{Ev}P(s)$ denotes the even and $\text{Od}P(s)$ denotes the odd part of $P(s)$.

By simple manipulation of Eqns. (2.19) and (2.20) one can get

$$G(s) = \tilde{K}\frac{1 - Z(s)}{1 + Z(s)} \tag{2.21}$$

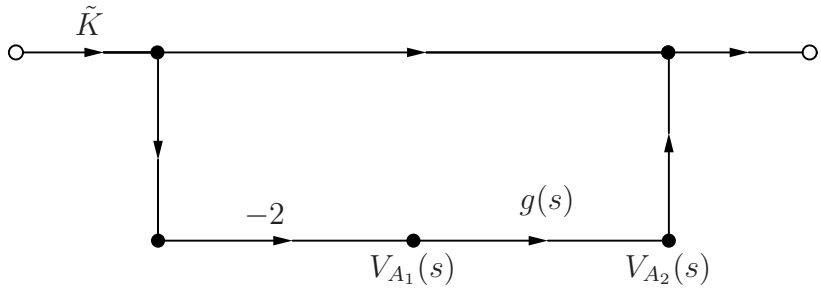

Figure 2.8: Signal Flow Graph of $G(s)$

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

$$Z(s) = \begin{cases} \dfrac{\text{Od}P(s)}{\text{Ev}P(s)} & \text{for even } \tilde{n} \\ \\ \dfrac{\text{Ev}P(s)}{\text{Od}P(s)} & \text{for odd } \tilde{n} \end{cases} \tag{2.22}$$

where $\tilde{n}$ is the order of $G(s)$ (odd when realizing $G_0(s)$ and even when realizing $G_1(s)$). 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 for odd $\tilde{n}$.

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

- The transfer function $G(s)$ is decomposed in the form

$$G(s) = \tilde{K}[1 - 2g(s)] \tag{2.23}$$

where

$$g(s) = \frac{Z(s)}{1 + Z(s)} \tag{2.24}$$

Here, $G(s)$ can be realized as the transfer function of the signal-flow graph in Fig. 2.8. 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. 2.9. 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. 2.10, in accordance with

$$Z(s) = \frac{1}{Y(s)} \tag{2.25}$$

$$Y(s) = sC_1 + \frac{1}{sL_1} + \sum_{i=2}^{m} \frac{sC_i}{s^2 C_i L_i + 1} \tag{2.26}$$

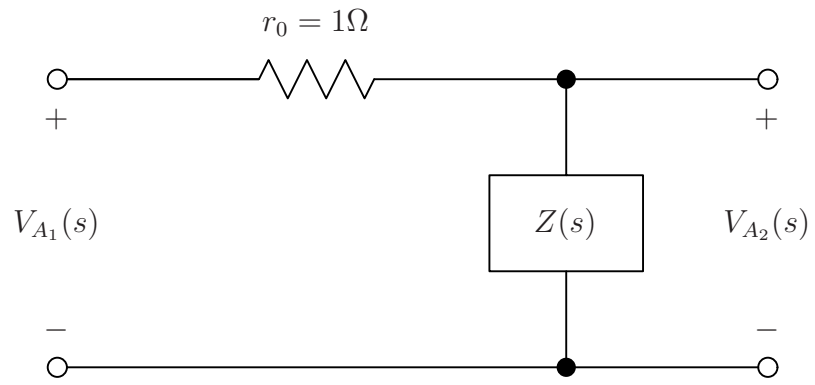

Figure 2.9: Voltage Divider Circuit for $g(s)$

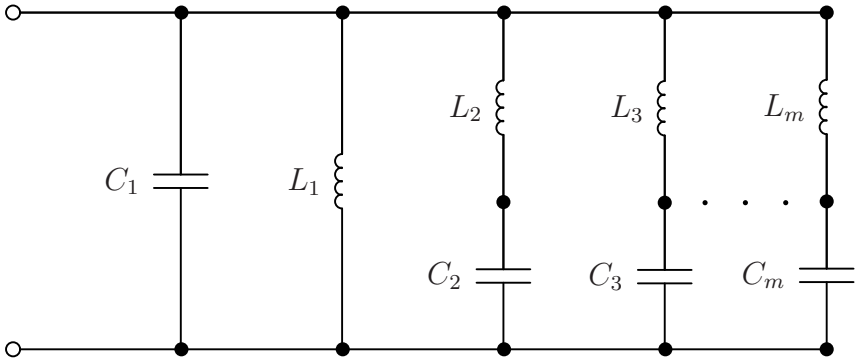

Figure 2.10: Realization of Impedance $Z(s)$

where $m = \tilde{n}/2$ for even $\tilde{n}$ and $m = (\tilde{n}+1)/2$ for odd $\tilde{n}$, and where C_i represent capacitances and L_i represent inductances (for $i = 1, 2, \ldots, m$), and inductor L_1 is only present for even $\tilde{n}$.

- The impedance $Z(s)$ in Fig. 2.10 is substituted into Fig. 2.9 and the precompensation is applied to the resulting network. This amounts to a modification of circuit elements in accordance with:

$$V_{A_1}^{'}(s) = \frac{V_{A_1}(s)}{1 - sT/2} \tag{2.27}$$

The resistance in r_0 in Fig. 2.9 is modified to:

$$r_0^{'} = z^{\frac{1}{2}} r_0 \tag{2.28}$$

and

$$L_1^{'} = L_1 \tag{2.29}$$

$$C_1^{'} = C_1 + \frac{T}{2} + \frac{T^2}{4L_1} + \sum_{i=2}^{m} \frac{C_i \frac{T^2}{4L_i}}{C_i + \frac{T^2}{4L_i}} \tag{2.30}$$

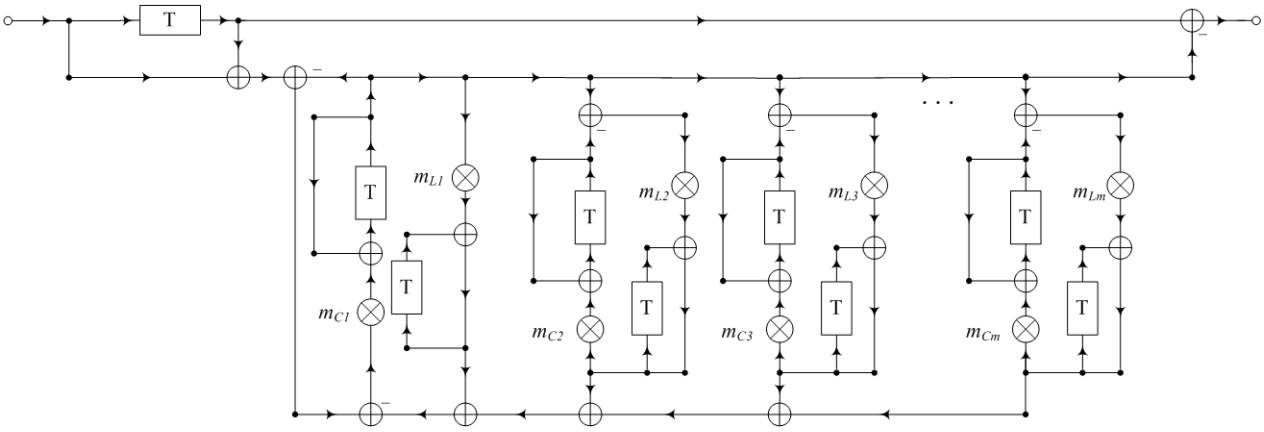

Figure 2.11: Realization of the Bilinear-LDI Digital Allpass Network $G(z)$ [34]

$$L_i^{'} = L_i \left[\frac{C_i + \frac{T^2}{4L_i}}{C_i} \right]^2 \tag{2.31}$$

$$C_i^{'} = \frac{C_i^2}{C_i + \frac{T^2}{4L_i}} \tag{2.32}$$

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

- Since the voltage/current signal-flow graph of the precompensated network [34] consists of analog integrators only and it has no analog differentiators, it can be used for bilinear-LDI realization method. Therefore, the analog integrators in the signal-flow graph of the precompensated network are replaced by LDI digital integrators, and by impedance-scaling, the resulting network is scaled by $z^{-\frac{1}{2}}$ to eliminate any half-delay elements. The resulting digital network is displayed in Fig. 2.11. The multiplier coefficients in Fig. 2.11 are as follows:

$$m_{L_i} = \frac{T}{L_i^{'}} \tag{2.33}$$

$$m_{C_i} = \frac{T}{C_i^{'}} \tag{2.34}$$

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

2.5 CSD Number System

CSD is based on the ternary number system (-1, 0, and 1). It is a unique representation of a binary number with minimum number of 1 and -1 digits. One of the main applications of CSD numbers, therefore, is in multiplication operation where it allows a minimum number of combined additions and subtractions to produce the product. It is shown that for a n-bit multiplication, the number of major operations (addition/subtraction and shift) never exceeds $\frac{n}{2}$, and on average this number is reduced to $\frac{n}{3}$, as the word size grows [99].

As mentioned before, from a hardware implementation point of view, a suitable design employs finite-wordlength multiplier coefficient values with sparse non-zero coefficients. Therefore, CSD number system is employed in this thesis. Subsequently, PSO 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 permitting the PSO algorithm to converge to a filter satisfying design specifications. Conversely, making the LUTs too many entries greatly increase the solution space, and this slows down the rate of convergence of the PSO algorithm. Let us consider a FRM digital filter consisting of CSD multiplier coefficients $\hat{m}_{FRM} \in CSD(L, l, f)$, where $CSD(L, l, f)$ represents the set of all possible CSD numbers having a wordlength of L digits and a maximum number of l non-zero digits with f digits in the fractional part. In this way, the CSD multiplier coefficients $\hat{m}_{FRM}$ can be expressed in the general form

$$\hat{m}_{FRM} = \sum_{n=1}^{L} D_n \times 2^{(R-n)} \tag{2.35}$$

and satisfying the constraints

$$D_n \in \{-1, 0, 1\} \tag{2.36}$$

$$D_n \times D_{n+1} = 0 \tag{2.37}$$

$$\sum_{n=1}^{L} |D_n| \leq l \tag{2.38}$$

with R representing a fixed value radix-point in the range $0 < R < L$. Constraint (2.37) implies

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

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 L and l is more complicated, and depend on the passband and stopband ripple specification and stopband frequency of the overall FRM digital filter $H(z)$, as well as the order and stopband frequency of the interpolation digital subfilter $H_a(z)$. 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 L or the maximum number of non-zero digits l.

It was shown that the worst case normalized CSD quantization is not very sensitive to changes in L, especially after 8 bits [100]. Therefore, it may be necessary to increase l in addition to increasing L 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 l, increasing l is much more detrimental to hardware efficiency than simply increasing L, and l 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 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 l have a non-uniform distribution, which means that the quantization error is not constant over the CSD range [100]. The worst case quantization increases as it advances from the least significant to the most significant end of the CSD number range. This behaviour remains more or less the same regardless of the chosen values of L or l. This quantization pattern plays

an important role in deciding what values of L and l while building the required LUTs. If, for instance, $H(z)$ is to have a wide band, then the corresponding FIR masking subfilters $F_0(z)$ 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 $H(z)$ therefore normally requires large values of L and/or l to reach an acceptably low ripple size as compared to a narrowband $H(z)$.

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

2.6 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 digital subfilters $H_a(z)$ and $H_b(z)$ to be BIBO stable. Likewise, in order for the interpolation digital 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 $G_1(z)$ to 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 PSO algorithm.

In the course of PSO algorithm, the infinite-precision multiplier coefficients m_{L_i} and m_{C_i} can only take quantized values $\hat{m}_{L_i}$ and $\hat{m}_{C_i}$ that belong to $CSD(L, l, f)$. In order for the bilinear-LDI digital allpass networks $G_0(z)$ and $G_1(z)$ to remain BIBO stable, it is required that the values of the corresponding quantized reactive

elements $\hat{L}_i$ and $\hat{C}_i$ remain positive [101] in the course of optimization. This is due to the properties of the bilinear frequency transformation from analog to digital domain. In order to find the conditions for BIBO stability and in accordance with Eqns. (2.33) and (2.34), one has:

$$\hat{L}_i^{'} = \frac{T}{\hat{m}_{L_i}} \tag{2.39}$$

$$\hat{C}_i^{'} = \frac{T}{\hat{m}_{C_i}} \tag{2.40}$$

Moreover, in accordance with Eqns. (2.29-2.32), one has:

$$\hat{L}_1^{'} = \hat{L}_1 \tag{2.41}$$

$$\hat{C}_1^{'} = \hat{C}_1 + \frac{T}{2} + \frac{T^2}{4\hat{L}_1} + \sum_{i=2}^{m} \frac{\hat{C}_i \frac{T^2}{4\hat{L}_i}}{\hat{C}_i + \frac{T^2}{4\hat{L}_i}} \tag{2.42}$$

$$\hat{L}_i^{'} = \hat{L}_i \left[\frac{\hat{C}_i + \frac{T^2}{4\hat{L}_i}}{\hat{C}_i} \right]^2 \tag{2.43}$$

$$\hat{C}_i^{'} = \frac{\hat{C}_i^2}{\hat{C}_i + \frac{T^2}{4\hat{L}_i}} \tag{2.44}$$

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

By substituting Eqns. (2.39) and (2.40) into Eqns. (2.41-2.44), and by solving the resulting equations for the reactive elements $\hat{L}_i$ and $\hat{C}_i$, one can obtain

$$\hat{L}_1 = \frac{T}{\hat{m}_{L_1}} \tag{2.45}$$

$$\hat{C}_1 = \frac{T \left\{ \frac{4}{\hat{m}_{C_1}} - \hat{m}_{L_1} - 4 \left(\sum_{i=2}^{m} \frac{1}{\frac{4}{\hat{m}_{L_i}} - \hat{m}_{C_i}} \right) - 2 \right\}}{4} \tag{2.46}$$

$$\hat{L}_i = \frac{T(\hat{m}_{L_i}\hat{m}_{C_i} - 4)^2}{16\hat{m}_{L_i}} \tag{2.47}$$

$$\hat{C}_i = \frac{-4T}{\hat{m}_{C_i}(\hat{m}_{L_i}\hat{m}_{C_i} - 4)} \tag{2.48}$$

From Eqns. (2.45-2.48), $\hat{L}_i > 0$ and $\hat{C}_i > 0$ provide that

$$\hat{m}_{L_1} > 0 \tag{2.49}$$

Table 2.2: Relations for Elements of Back-Transformed Reactance

Element	Equation	Inequality Constraints
$\hat{L}_1$	$\dfrac{T}{\hat{m}_{L_1}}$	$\hat{m}_{L_1} > 0$
$\hat{C}_1$	$\frac{1}{4}T\left\{\frac{4}{\hat{m}_{C_1}} - \hat{m}_{L_1} - 4\left(\sum_{i=2}^{m} \frac{1}{\frac{4}{\hat{m}_{L_i}} - \hat{m}_{C_i}}\right) - 2\right\}$	$\hat{m}_{C_1} < 4\left\{\hat{m}_{L_1} + 4\left(\sum_{i=2}^{m} \frac{1}{\frac{4}{\hat{m}_{L_i}} - \hat{m}_{C_i}}\right) + 2\right\}^{-1}$
$\hat{L}_i$	$\dfrac{T(\hat{m}_{L_i}\hat{m}_{C_i} - 4)^2}{16\hat{m}_{L_i}}$	$\hat{m}_{L_i} > 0$
$\hat{C}_i$	$\dfrac{-4T}{\hat{m}_{C_i}(\hat{m}_{L_i}\hat{m}_{C_i} - 4)}$	$\hat{m}_{C_i} < 4\left(\hat{m}_{L_i}\right)^{-1}$

$$\hat{m}_{L_i} > 0 \tag{2.50}$$

$$\hat{m}_{C_i} < \frac{4}{\hat{m}_{L_i}} \tag{2.51}$$

$$\hat{m}_{C_1} < \frac{4}{\left\{\hat{m}_{L_1} + 4\left(\sum_{i=2}^{m} \frac{1}{\frac{4}{\hat{m}_{L_i}} - \hat{m}_{C_i}}\right) + 2\right\}} \tag{2.52}$$

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 $\hat{m}_{FRM} \in CSD(L, l, f)$ such that the inequality constraints (2.49-2.52) are satisfied. The equations and corresponding condition required for BIBO stability are summarized in Table 2.2.

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 $\hat{m}_{L_i}$, $\hat{m}_{C_i} \in CSD(L, l, f)$ such that the inequality constraints of Table 2.2 are satisfied.

It should be pointed out that constraint (2.51) is most stringent when $\hat{m}_{L_i}$ is at its largest possible value. Similarly, constraint (2.52) is most stringent when $\hat{m}_{L_1}$, $\hat{m}_{L_i}$ and $\hat{m}_{C_i}$ are all at their largest possible values (while $\hat{m}_{L_i}$ and $\hat{m}_{C_i}$ still adhere to constraint $\hat{m}_{C_i} < 4\left(\hat{m}_{L_i}\right)^{-1}$).

2.7 Proposed PSO of FRM Digital Filters

The proposed particle swarm optimization of FRM FIR digital filters and BIBO stable FRM IIR digital filters is carried out over the CSD multiplier coefficient space $CSD(L_{0\,\text{or}\,1}, l_{0\,\text{or}\,1}, f_{0\,\text{or}\,1})$, where $L_{0\,\text{or}\,1}$ represents the multiplier coefficient wordlength, where $l_{0\,\text{or}\,1}$ represents the maximum number of non-zero digits, and where $f_{0\,\text{or}\,1}$ represents the number of fractional part digits (for FIR or IIR digital subfilters, respectively).

The starting point of any stochastic algorithm plays an important role in the convergence behavior of the optimization algorithm [83]. Therefore, it is important to generate the initial swarm in proper positions in the search space rather than complete random generation of the initial population. In order to achieve this, the following technique is employed:

2.7.1 Initiation of PSO

To start the PSO algorithm from a good position in the search space the infinite precision multiplier coefficient values of the seed particle are generated by using classical techniques as discussed in previous sections. These infinite precision multiplier coefficient values are turned into their finite precision counterparts by simply rounding them to their closest CSD values. This seed particle is used as the center of the swarm and a cloud of particles are generated randomly around the seed particle. It should be noted that the distance of the randomly generated particles should not be far from the seed particle. In this way, the initial swarm contains particles which have high chances of being near the optimal solution. The multiplier coefficient values of the swarm are taken from a set of CSD LUTs which are constructed as follows:

2.7.2 FRM FIR Digital Filter Template LUTs

For the case of FRM FIR digital filters, a template LUT is constructed for all multiplier coefficient values for the interpolation digital subfilter $H_a(z)$ and the masking digital subfilters $F_0(z)$ and $F_1(z)$. The elements of this LUT belong to

$CSD(L_0, l_0, f_0)$. The values of L_0, l_0 and f_0 are determined empirically based on the amplitude frequency-response of the digital subfilters $H_a(z)$, $F_0(z)$ and $F_1(z)$.

2.7.3 FRM IIR Digital Filter Template LUTs

For the case of FRM IIR digital filters, it is necessary and sufficient to choose the values of the multiplier coefficients, such that the inequality constraints (2.49-2.52) are satisfied. In order to achieve this, the LUTs are constructed as follows:

- One LUT is constructed for all multiplier coefficient values $\hat{m}_{FIR} \in CSD(L_0, l_0, f_0)$ for the masking digital subfilters $F_0(z)$ and $F_1(z)$. The values of L_0, l_0 and f_0 are determined empirically based on the amplitude frequency-response of the masking digital subfilters $F_0(z)$ and $F_1(z)$.

- A LUT is constructed for all multiplier coefficient values $\hat{m}_{IIR} \in CSD(L_1, l_1, f_1)$ for the digital allpass networks $G_0(z)$ and $G_1(z)$. Once again, the values of L_1, l_1 and f_1 are determined empirically. Also, it is expedient to assume that $\hat{m}_{IIR}$ have only positive values.

- The above CSD LUT is used to form one size-reduced LUT per the 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 $CSD(L_1, l_1, f_1)$, and from above by the corresponding value of the finite-wordlength coefficients for the seed particle. The size-reduced LUTs are augmented before PSO process commences. The purpose of this augmentation is to ensure that the exploration space include as many of those CSD multiplier coefficients $\hat{m}_{L_1}$, $\hat{m}_{C_1}$, $\hat{m}_{L_i}$ and $\hat{m}_{C_i}$ which still satisfy the BIBO stability constraints (2.49-2.52).

The above constructed LUTs are used as template LUTs. There are two problems concerning the PSO of FRM IIR digital filters over the CSD multiplier coefficient space. To overcome these problems, the template LUTs must be further processed. These two problems and the way to solve them are discussed in the following.

2.7.4 PSO indirect search method

In PSO, the required new particle position is obtained from the previous position of the particle through the addition of a random (normalized) velocity value. However, by directly applying the conventional PSO to the above optimization over the CSD multiplier coefficients, one may obtain new particle positions whose coordinate values are no longer in $CSD(L_{0\,\text{or}\,1}, l_{0\,\text{or}\,1}, f_{0\,\text{or}\,1})$. In order to overcome this problem, the optimization search is carried out indirectly via the indices to the LUT CSD values (as opposed to LUT CSD values themselves). In this way, the CSD coordinate values for each particle position are obtained by integer indices to the CSD LUTs. The key point in the indirect search rests with ensuring that the index set is closed, i.e. by ensuring that each index points to a valid CSD value in the LUT, and that the resulting particle in the course of PSO adheres to the prespecified CSD number format.

If the velocity values are replaced by their closest integer values, the update equations become modified to

$$\hat{v}_{kj}^{i} = [w\hat{v}_{kj}^{i-1} + c_1 r_1(\hat{x}_{best_{kj}}^{i-1} - \hat{x}_{kj}^{i-1}) + c_2 r_2(\hat{g}_{best_j}^{i-1} - \hat{x}_{kj}^{i-1})]^{1} \tag{2.53}$$

$$\text{if} \quad \hat{v}_{kj}^{i} < \hat{v}_{min} \quad ; \quad \hat{v}_{kj}^{i} = \hat{v}_{min}$$

$$\text{if} \quad \hat{v}_{kj}^{i} > \hat{v}_{max} \quad ; \quad \hat{v}_{kj}^{i} = \hat{v}_{max}$$

$$\hat{x}_{kj}^{i} = \hat{x}_{kj}^{i-1} + \hat{v}_{kj}^{i} \tag{2.54}$$

Here, $\hat{x}_{kj}$, $\hat{v}_{kj}$, $\hat{x}_{best_{kj}}$, $\hat{g}_{best_j}$, $\hat{v}_{min}$ and $\hat{v}_{max}$ are all integer values where $\hat{v}_{min} < 0$ and $\hat{v}_{max} > 0$. In addition, w is limited in the interval $[0, 0.5)$ (as discussed shortly).

2.7.5 Barren layers

Due to their finite length, the template LUTs inevitably lead to a bounded optimization search space. In order to ensure that the particles do not cross over to the outside of the search space in the course of PSO, the search space is constructed as a combination of two regions, namely the interior and barren layers. The barren layer

[1] $[R]$ denotes rounding R to its closest integer, where R is assumed to be a real value.

is constructed to yield relatively low fitness values, and is represented as header and footer in the template LUT. There are two problems concerning the construction of the barren layers:

barren layer entries

The first problem in the construction of barren layers concerns how to make the fitness values in the barren layer relatively low. This problem can be resolved by filling the header part by unrealistically large, and the footer part by unrealistically small CSD multiplier coefficient values.

barren layer width

The second problem, on the other hand, concerns how to determine the width of the barren layer such that the particles do not cross over to the outside of the search space even under the worst case scenario. These two problems relate to the number of entries and the CSD values of the entries in header and footer parts of the template LUTs. To overcome this problem, let us consider the j-th variable in the k-th particle is in the boundaries of one of the template LUTs in iteration $i-1$. The worst case scenario occurs when x_{kj}^{i-1} moves toward the barren layer with the peak permissible velocities (v_{max} for the header, and v_{min} for the footer. If in the i-th iteration x_{kj}^{i} is in the footer:

$$\hat{x}_{best_{kj}}^{i} > \hat{x}_{kj}^{i} \tag{2.55}$$

$$\hat{g}_{best_{j}}^{i} > \hat{x}_{kj}^{i} \tag{2.56}$$

and if it is in the header:

$$\hat{x}_{best_{kj}}^{i} < \hat{x}_{kj}^{i} \tag{2.57}$$

$$\hat{g}_{best_{j}}^{i} < \hat{x}_{kj}^{i} \tag{2.58}$$

Eqns. (2.55-2.58) show that the velocity of the particle in iteration $i+1$ tends to move the particle in a direction opposite to the direction of the barren regions. Here, the worst case happens when $r_1 = r_2 = 0$. In this way, the number of entries

L_f in the footer part, and the number of entries L_h in the header part is determined in accordance with

$$\begin{aligned} L_f &= |\hat{v}_{min}| + [w|\hat{v}_{min}|] + [w[w|\hat{v}_{min}|]] + \dots \\ &\leq |\hat{v}_{min}| + \frac{|\hat{v}_{min}|}{2} + \frac{|\hat{v}_{min}|}{4} + \dots \\ &= 2|\hat{v}_{min}| \end{aligned} \tag{2.59}$$

$$\begin{aligned} L_h &= \hat{v}_{max} + [w\hat{v}_{max}] + [w[w\hat{v}_{max}]] + \dots \\ &\leq \hat{v}_{max} + \frac{\hat{v}_{max}}{2} + \frac{\hat{v}_{max}}{4} + \dots \\ &= 2\hat{v}_{max} \end{aligned} \tag{2.60}$$

Let us recall that since $0 \leq w < 0.5$,

$$\text{if} \quad v : \text{positive integer} \quad \Rightarrow \quad [wv] \leq \frac{v}{2} \tag{2.61}$$

In addition, after some iterations $\hat{v}_{kj}^{i+1} = 0$. Otherwise, if $w \geq 0.5$, $\hat{v}_{kj}^{i+1}$ can never become zero, and the width of the barren layer will be infinity.

The augmented LUTs remains fixed in the course of PSO, restricting automatic particle movement inside the limited search space. Modifying the index values inside each particle by adding the current indices to the length of the footer barren region, L_f, PSO algorithm is ready to start the optimization of FRM digital filters.

2.8 Design Methodology

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

1. *Designing the 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 λ_M and distance $\tilde{D}_M$ for each interpolation factor M are given as:

$$\lambda_M = \max[|(\frac{\pi}{2} - \tilde{\theta})|, |(\frac{\pi}{2} - \tilde{\phi})|] \tag{2.62}$$

$$\tilde{D}_M = \frac{\pi}{M} - \frac{2\lambda}{M} \tag{2.63}$$

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 attenuation 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 implemented 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 [102]

$$D = \frac{10^{0.1R_a} - 1}{10^{0.1R_p} - 1} \tag{2.64}$$

$$q = 10^{\frac{-\log(16D)}{N_{IIR}}} \tag{2.65}$$

$$q = q_0 + 2q_0^5 + 15q_0^9 + 150q_0^{13} \tag{2.66}$$

$$k_p = \left[\frac{1 - 2q_0}{1 + 2q_0}\right]^2 \tag{2.67}$$

$$k = \sqrt{1 - k_p^2} \tag{2.68}$$

In order to satisfy the passband edge specification, the digital passband edge

$\omega_p = \tilde{\theta}$ for Case I filters. The digital stopband edge ω_a is then determined using the analog ratio k. (Here, frequency warping from digital to analog domain, and vice versa, given by Eqn. (2.16) needs to be taken into account.) Similarly, $\omega_a = \tilde{\phi}$ for Case II filters, and ω_p 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 particle*: The seed FRM digital filter particle is formed as follows:

   - A particle with B_1 coordinates is formed in which each coordinate serves as an index of the corresponding CSD LUT for each multiplier coefficient constituent in the interpolation digital subfilters.
     - In case of FRM IIR digital filters, the multiplier coefficients correspond to the bilinear-LDI allpass digital networks $G_0(z)$ and $G_1(z)$.
     - In case of FRM FIR digital filters, the multiplier coefficient correspond to the interpolation digital subfilters $H_a(z)$ and $H_b(z)$.
   - A particle with B_2 coordinates is formed in which each coordinate serves as an index of the corresponding CSD LUT for each multiplier coefficient in the FIR masking digital subfilters $F_0(z)$ and $F_1(z)$.

3. *Generation of Initial Swarm*: An initial swarm of K particles is formed by generating a random cloud around the seed particle as discussed in section 2.7.1.

4. *Fitness Evaluation*: The fitness function for CSD FRM IIR digital filters is defined in accordance with

$$fitness_{magnitude} = -20log[max(\varepsilon_p, \varepsilon_a)] \tag{2.69}$$

$$fitness_{group-delay} = \varsigma_p \tag{2.70}$$

$$fitness = fintess_{magnitude} - fitness_{group-delay} \tag{2.71}$$

where

$$\varepsilon_p = \underbrace{max}_{\omega \in \Delta\omega_p} [W_p |H(e^{j\omega}) - 1|] \tag{2.72}$$

$$\varepsilon_a = \underbrace{max}_{\omega \in \Delta\omega_a} [W_a |H(e^{j\omega})|] \tag{2.73}$$

$$\varsigma_p = \underbrace{max}_{\omega \in \Delta\omega_p} [W_{gd} |\tau(\omega) - \mu_\tau|] \tag{2.74}$$

with $\Delta\omega_p$ representing the passband frequency region(s), with $\Delta\omega_a$ representing the stopband frequency region(s), and with $\tau(\omega)$ representing the group-delay frequency response of the FRM IIR digital filter [2]. Here, W_p, W_a, and W_{gd} represent weighting factors for the passband and stopband magnitude responses, and for the group-delay response, respectively. Moreover, μ_τ represents the average group-delay over the passband region. In [103], a convenient way to represent digital networks in terms of matrix representation is presented. This technique can be used to find the magnitude and group delay frequency response of the digital network in Fig. 2.11. Let us consider the input to the digital network in Fig. 2.11 to be x_D and the output of it to be y_D. In addition, let the output of the i-th time delay in Fig. 2.11 to be x_i and the input to the i-th time delay to be y_i. The transfer function matrix of the network, $\mathbf{T}$, can be found as

$$\mathbf{y} = \mathbf{T}\mathbf{x} \tag{2.75}$$

where $\mathbf{y} = [y_D, y_1, y_2, \ldots, y_{2m+1}]^t$ [3]and $\mathbf{x} = [x_D, x_1, x_2, \ldots, x_{2m+1}]^t$, and $\mathbf{T}$ is a $(2m+2) \times (2m+2)$ matrix with the entries obtained as Eqn. (2.76).

$$\mathbf{T} = \begin{bmatrix} 0 & 1 & -1 & 0 & 0 & 0 & \ldots & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 \\ m_{C_1} & m_{C_1} & 1-m_{C_1}(1+\sum_{i=1}^{m} m_{L_i}) & -m_{C_1} & m_{C_1}m_{L_2} & -m_{C_1} & \ldots & m_{C_1}m_{L_m} & -m_{C_1} \\ 0 & 0 & m_{L_1} & 1 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & m_{C_2}m_{L_2} & 0 & 1-m_{C_2}m_{L_2} & m_{C_2} & \ldots & 0 & 0 \\ 0 & 0 & m_{L_2} & 0 & -m_{L_2} & 1 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & & \\ 0 & 0 & m_{C_m}m_{L_m} & 0 & 0 & 0 & & 1-m_{C_m}m_{L_m} & m_{C_m} \\ 0 & 0 & m_{L_m} & 0 & 0 & 0 & & -m_{L_m} & 1 \end{bmatrix} \tag{2.76}$$

[2] In case of FRM FIR digital filters, ς_p is a constant and has no effect in the optimization process. In this thesis $\varsigma_p = 0$ for FRM FIR digital filters.

[3] $\mathbf{X}^t$ denotes the transpose of the matrix $\mathbf{X}$.

Since $x_i = z^{-1}y_i$, the transfer function $G(z) = \frac{y_D}{x_D}$ can be found as

$$G(z) = z^{-1}\mathbf{e}[\mathbf{I} - z^{-1}\mathbf{D}]^{-1}\mathbf{c} \tag{2.77}$$

where $\mathbf{e}$ is a row vector and $\mathbf{c}$ is a column vector of length $2m+1$, and where $\mathbf{I}$ is the identity matrix and $\mathbf{D}$ is a $(2m+1) \times (2m+1)$ matrix in accordance with

$$\mathbf{T} = \begin{bmatrix} 0 & \mathbf{e} \\ \\ \mathbf{c} & \mathbf{D} \end{bmatrix} \tag{2.78}$$

The matrix $\mathbf{T}$ is also useful in finding the group delay of $H(z)$. The group-delay of $H(e^{j\omega})$ is given by

$$\tau(\omega) = -\text{Im}\left\{\frac{1}{H(e^{j\omega})}\frac{dH(e^{j\omega})}{d\omega}\right\} \tag{2.79}$$

With the help of Eqn. (2.10), the expression $\frac{dH(e^{j\omega})}{d\omega}$ can be written as

$$\begin{aligned}\frac{dH(e^{j\omega})}{d\omega} = \frac{1}{2}\Big[&\frac{dG_0(e^{j\omega})}{d\omega}(F_0(e^{j\omega}) + F_1(e^{j\omega}))+ \\ &\frac{d(F_0(e^{j\omega}) + F_1(e^{j\omega}))}{d\omega}G_0(e^{j\omega})+ \\ &\frac{dG_1(e^{j\omega})}{d\omega}(F_0(e^{j\omega}) - F_1(e^{j\omega}))+ \\ &\frac{d(F_0(e^{j\omega}) - F_1(e^{j\omega}))}{d\omega}G_1(e^{j\omega})\Big]\end{aligned} \tag{2.80}$$

The derivative of FIR filters can be easily found from their transfer function. In order to find the derivative of the digital allpass networks $G_0(z)$ and $G_1(z)$, the following expression can be used

$$\frac{dG(e^{j\omega})}{d\omega} = -je^{-j\omega}\sum_{i=1}^{2m+1} G_{xi}(e^{j\omega})G_{iy}(e^{j\omega}) \tag{2.81}$$

where $G_{xi}(z)$ is the transfer function between x_D and y_i, and where $G_{iy}(z)$ is the transfer function between x_i and y_D. The transfer functions $G_{xi}(z)$ and $G_{iy}(z)$ can be found from the transfer function matrix $\mathbf{T}$ as follows

$$G_{xi}(z) = a_{xi} + z^{-1}\mathbf{e}_{xi}[\mathbf{I} - z^{-1}\mathbf{D}]^{-1}\mathbf{c} \tag{2.82}$$

$$G_{iy}(z) = a_{iy} + z^{-1}\mathbf{e}[\mathbf{I} - z^{-1}\mathbf{D}]^{-1}\mathbf{c}_{iy} \tag{2.83}$$

where a_{xi} and a_{iy} are scalars, $\mathbf{e}_{xi}$ is a row vector and $\mathbf{c}_{iy}$ is a column vector of length $2m+1$, in accordance with $[a_{xi} \quad \mathbf{e}_{xi}]$ is the i-th row of the matrix $\mathbf{T}$, and $[a_{iy} \quad \mathbf{c}_{iy}^t]^t$ is the i-th column of the matrix $\mathbf{T}$. Having the expressions for $H(e^{j\omega})$ and $\frac{dH(e^{j\omega})}{d\omega}$, the group delay can be obtained in accordance with Eqn. (2.79).

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

$$W_{gd} = \frac{\zeta \times fitness_{magnitude}}{fitness_{group-delay}} \tag{2.84}$$

where ζ is a fixed constant such that $0 < \zeta < 1$, and where $fitness_{magnitude}$ and $fitness_{group-delay}$ are obtained by examining the seed FRM digital filter particle. The weighting factor for the group-delay increases as $\zeta \to 1$.

2.9 Summary

This chapter has presented a novel technique for PSO of FRM digital filters incorporating FIR and IIR digital interpolation subfilters. In case of FRM IIR digital filters, the bilinear-LDI approach is employed to reduce the number of multiplier coefficients in the IIR digital subfilter. In order to map an analog prototype filter to the digital domain using bilinear-LDI technique, elliptic filters with minimum Q-factor is used. In this way, the IIR interpolation subfilters constituent in the FRM IIR digital filter are guaranteed to be power complementary.

The design methodology for design and optimization of FRM digital filters using PSO technique has been presented. There are three main problems in the PSO of FRM digital filters. Due to the randomness of the operations of addition and subtraction in the underlying PSO, one may obtain a FRM digital filter particle whose multiplier coefficients do not conform to CSD number format. This problem has been resolved by constructing a set of LUTs containing permissible CSD numbers, and by using the indices of the LUTs as optimization variables in PSO. Therefore, an integer-based PSO is developed to search over the indices of the LUTs.

In case of FRM IIR digital filters, PSO may produce particles which are not BIBO stable. This problem is not present for FRM FIR digital filters due to the

inherent stability feature of FIR filters. To overcome this problem, a set of BIBO stability constraints has been developed and the LUTs have been successively augmented until the BIBO stability constraints remain satisfied. In this way, the FRM IIR digital filter particles generated in the course of PSO are guaranteed to be BIBO stable.

On the other hand, the particles may go over the boundaries of LUTs in the course of optimization. This problem has been resolved by introducing barren layers. Barren layers are added to the header and the footer of the LUTs and they are characterized by low fitness values compared to the main entries of the LUTs, i.e. a particle with a coordinate in a barren layer has a lower fitness value than a particle without a coordinate in the barren layers.

The optimization of FRM digital filters using PSO algorithm concerns both the magnitude and the group-delay frequency-responses of the FRM digital filter. Matrix equations have been developed to determine the magnitude and the group-delay frequency-responses of FRM digital filters. This improves the evaluation of the fitness value of each particle and simplifies the objective function used in the course of PSO.

Chapter 3

Novel PSO for High-Level Synthesis of Digital Filters

In this chapter, PSO algorithm is exploited and applied to the development of a new optimization technique for the high-level synthesis of digital filter data-paths. In this algorithm, a powerful encoding scheme is introduced that encodes the information present in the digital filter DFG into successive swarm of particles. These particles carry information about the main steps that are required for the high-level synthesis of digital filter data-path, i.e. scheduling, allocation, and binding. This information can be used to calculate the cost function associated with the time and area requirements in the corresponding hardware realization of the digital filter data-path. The cost associated with the final digital filter data-path is minimized for obtaining global area-optimal, time-optimal, or combined area-cum-time-optimal data-paths subject to user-specified constraints on the number of physical arithmetic functional units employed. The final point in the optimization is the identification of the data-path that optimizes the area and time in the corresponding data-path encoded swarm.

The proposed PSO algorithm guarantees that the data-dependency relationships in the digital filter DFG remain satisfied under the operations of addition and subtraction in the underlying PSO algorithm. In addition, a technique is developed to avoid any functional unit violation (that may occur because of the random nature of the operations of addition and subtraction in the underlying PSO algorithm) through the course of high-level synthesis of digital filters.

This chapter proceeds as follows. Section 3.1 presents an introduction to the high-level synthesis of digital systems and introduces the steps that are required for the high-level synthesis of digital filters. Section 3.2 presents a powerful encoding scheme to encode the digital filter data-paths into particles that can be used in the course of PSO algorithm. This scheme is capable of satisfying the data-dependency relationships in the digital filter DFG. Section 3.3 explains how the initial swarm of particles is generated in order to increase the speed of convergence of PSO. Section 3.4 describes the constraints that are present in the course of high-level synthesis of digital filters using PSO and solutions to these constraints are provided in this section. Section 3.5 is concerned with the evaluation of the cost function that is being used in the course of high-level synthesis of digital filters using PSO. Finally, Section 3.6 provides a summary of this chapter.

3.1 High-Level Synthesis of Digital Filters

High-level synthesis of digital filters is the act of mapping a behavioral description of the digital filter to the RTL model in order to execute the variable assignments in the digital filter data-path. Since the amount of computation in each state is determined in the RTL model, one must first define the number and type of resources (arithmetic functional units, multiplexors, registers, etc.) to be used in the data-path. Allocation is the task of defining necessary resources for a given design specification associated with a digital filter data-path. The next task in mapping a behavioral description into an RTL model is partitioning the behavioral description into time-steps. In this way, the allocated resources can be used to compute all the variable assignments present in each time-step. This partitioning of behavioral description into time intervals is called scheduling. Although scheduling assigns each operation to a particular time-step, it does not assign it to a particular operator. To obtain the proper implementation, one can assign each variable to a storage unit (e.g. a register), each operation to a functional unit (e.g. an adder or a multiplier), and each transfer from input or output to units and among units to an interconnection unit (e.g. a multiplexor). This task is called binding (or resource sharing). Binding

determines the structure of the digital filter data-path but it is unable to define the structure of the control unit. In the following subsections, the main tasks that are required for the high-level synthesis of digital filters are explained [104].

3.1.1 Allocation

Allocation is the task of determining the type and quantity of resources used in a given digital filter DFG. Other tasks, e.g. clocking scheme, memory hierarchy, and pipelining style, can also be determined by allocation. The goal of allocation is to make the design having a good performance, while keeping the cost of it below a reasonable limit. If the original behavioral description of the digital filter data-path contains inherent parallelism, allocating more hardware resources increases area and cost, but it also creates more opportunities for parallel operations or storage accesses, resulting in a better performance. On the other hand, allocating fewer resources decreases area and cost, but it also forces operations to execute sequentially, resulting in a poorer performance. To perform the required trade-offs, allocation must determine the exact area and performance values. A simple approximation of cost and performance consists of the number of functional units and support cells, and the number of time-steps, respectively. This approximation can be used to come up with an optimal allocation scheme that decreases the cost, while keeping the performance in a good level.

3.1.2 Scheduling

Scheduling is the act of assigning operations and memory accesses in a digital filter data-path, into clock cycles or time-steps. There are two types of scheduling algorithms based on the optimization goal and the user-specified constraints.

- *Resource-constrained scheduling* tries to maximize usage of the allocated resources. This scheduling algorithm occurs if all the available resources and the maximum number of time-steps during allocation has been specified by the user. The goal of resource-constrained scheduling algorithm is to generate a design with the best possible performance, or the fewest number of time-steps. Resource-constrained scheduling usually produces a design that

has one time-step at a time and then it schedules operations so as not to exceed resource constraints or violate data-dependencies. Resource-constrained scheduling guarantees that at the time-step for which it schedules an operation, an operator which can execute that operation is available and all the predecessors of the corresponding operation have been scheduled.

- *Time-constrained scheduling* happens if a list of resources is not available prior to scheduling, but a desired overall performance is specified by the user. The goal of this scheduling algorithm is to produce a design with the lowest possible cost, or the fewest number of functional units. In time-constrained scheduling, the maximum number of time-steps available for operations is fixed by the user. Based on this performance constraint and the data dependency constraints, the earliest time-step and the latest time-step, into which an operation can be scheduled are computed. Using the earliest and the latest time-steps, bounds for all operations, one can estimate the maximum number of functional units or the cost of the design. Time-constrained scheduling algorithms select an operation, evaluate the cost of scheduling it in each time-step between the earliest and the latest time-steps, and select the state that results in the least cost. The important goal is to minimize the number of functional units in any time-step.

When the critical path is defined in a digital filter data-path, scheduling must ensure that the design uses faster functional units for operations on the critical path and slower units for operations outside the critical path. In this way, an optimal design of the digital filter data-path may be achieved.

3.1.3 Binding

The binding task assigns the operations and memory accesses within each clock cycle to available hardware units. A resource such as a functional, storage, or interconnection unit can be shared by different operations, data accesses, or data transfers if they are mutually exclusive. For example, two operations assigned to two different time-steps are mutually exclusive since they will never execute simul-

taneously. Therefore, they can be executed with a single hardware unit. Binding consists of three subtasks based on the unit type:

- *Storage binding* assigns variables to storage units. Storage units can be of many types, including registers, register files, and memory units. Two variables that are not alive simultaneously in a given state can be assigned to the same register. Two variables that are not accessed simultaneously in a given state can be assigned to the same port of a register file or memory.

- *Functional unit binding* assigns each operation in a time-step to a functional unit. A functional unit or a pipeline stage can execute only one operation per clock cycle.

- *Interconnection binding* assigns an interconnection unit such as a multiplexor or bus for each data transfer among ports, functional units, and storage units.

Although listed separately, the three subtasks are intertwined and must be carried out concurrently for optimal results.

In the next section, the above concepts are used to encode the digital filter data-path into particles in such a way that the resulting particles can be utilized in the course of PSO for high-level synthesis of digital filter data-path.

3.2 Digital Filter DFG Encoding Scheme

This section presents a powerful DFG encoding scheme for the scheduling, allocation, and binding of digital filter data-paths in the underlying PSO algorithm. In the proposed encoding scheme, the digital filter DFG is encoded into a particle containing two partitions P^1 and P^2 [58], where the partition P^1 contains the information regarding scheduling of the digital filter data-path, and the partition P^2 embodies the information regarding the corresponding DFG allocation and binding. These partitions are formed, manipulated, and maintained in such a way that they preserve the data-dependency relationships in the original digital filter signal flow-graph under the operations of addition and subtraction by the underlying PSO algorithm.

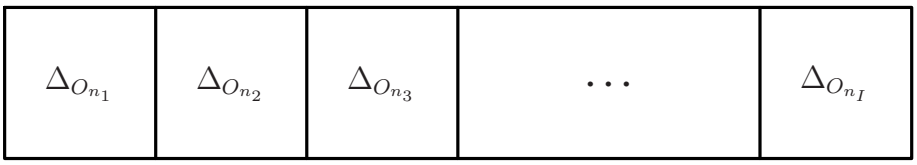

Figure 3.1: Particle Structure for Partition P^1

3.2.1 DFG Encoding for Scheduling

As discussed before, the partition P^1 contains the information regarding the scheduling of the digital filter data-path. To achieve this goal, partition P^1 in the DFG particle is generated to incorporate a set of I coordinates as shown in Fig. 3.1, where the j-th coordinate represents the delay $\Delta_{O_{n_j}}$ from the first possible time-step where an operation O_{n_j} may be scheduled in the high-level synthesis of digital filter data-path, where $1 \leq n_j \leq I$ and $1 \leq j \leq I$ and where I denotes the total number of operations present in the DFG.

In this way, the problem of scheduling the digital filter data-path reduces to that of determining the operation identifiers n_1,n_2,. . .,n_I such that the data-dependency relationships in the digital filter DFG remain satisfied under the operations of addition and subtraction by the PSO algorithm. One straightforward approach to determining these identifiers is to perform an ASAP scheduling and order the operations O_{n_j} in ascending order of their ASAP schedule time-step. However, the problem of such an ordering is that this process does not lead to a unique ordering process. In order to resolve this problem, one must take into consideration the critical path information in the operation ordering process. This critical path information can be used to order the operations lying on a longer critical path before those lying on a shorter critical path because the latter operations have smaller degrees of freedom.

The critical paths in the digital filter DFG can be determined by performing the following tasks:

- ASAP scheduling is performed on the digital filter DFG and the resulting total number of time-steps is recorded as T_{ASAP}.
- ALAP scheduling is performed on the digital filter DFG with the maximum number of time-steps fixed at T_{ASAP}.

- The difference between the ALAP schedule time-step $t_{\text{ALAP}_{n_j}}$ and the ASAP schedule time-step $t_{\text{ASAP}_{n_j}}$ for each operation O_{n_j} in the digital filter DFG is computed and stored as $t_{\text{diff}_{n_j}}$.

In this way, the operations with lower values of $t_{\text{diff}_{n_j}}$ reside on longer critical paths and the operations with higher values of $t_{\text{diff}_{n_j}}$ reside on shorter critical paths. Consequently, the operations are ordered in the ascending order of their $t_{\text{diff}_{n_j}} + t_{\text{ASAP}_{n_j}}$ values in order to take into account both the critical path information and the data-dependency relationships in the digital filter DFG. But by definition:

$$t_{\text{diff}_{n_j}} + t_{\text{ASAP}_{n_j}} = t_{\text{ALAP}_{n_j}} \tag{3.1}$$

Therefore, the operations happen to be ordered in ascending order of their ALAP schedule time-steps.

3.2.2 DFG Encoding for Allocation and Binding

In the high-level synthesis of digital filter data-paths, it is frequently required to optimize not only the cost associated with the physical arithmetic functional units employed, but also that associated with the required support cells (multiplexors and registers). The latter cost is influenced by two important factors, namely the allocation of operators to various operations and the ordering of signals for symmetrical operations (i.e. digital additions).

The partition P^2 in the DFG particle is formed to incorporate a set of I coordinates as shown in Fig. 3.2, where $\Theta_{O_{n_{j-I}}}$ in the j-th coordinate represent the operator number and where b_{j-I} represents the order of the signals within the $\Theta_{O_{n_{j-I}}}$-th operator executing operation $O_{n_{j-I}}$, where $(I+1) \leq j \leq 2I$ and $b_{j-I} \in \{-1, 1\}$ indicating whether or not the two input signals associated with the operator are swapped for symmetrical operations. Here, $1 \leq \Theta_{O_{n_{j-I}}} \leq \theta_n$ with θ_n being a user-specified number and $1 \leq n \leq N_M$ where N_M is the number of different functional units available in the DFG.

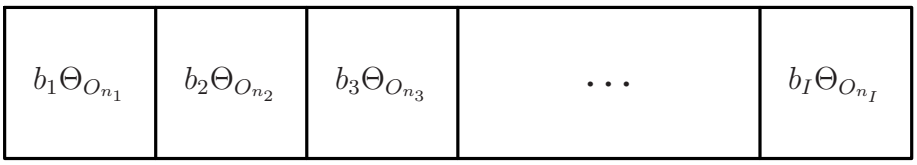

Figure 3.2: Particle Structure for Partition P^2

3.3 Formation of the Initial Swarm

To start the PSO algorithm from a good position in the search space one can perform the ASAP scheduling or the ALAP scheduling of the digital filter data-path and use the corresponding order of operations to form P^1 of the seed particle. To form P^2, one can use the operator numbers for the corresponding operations in the ASAP or ALAP scheduling of the digital filter data-path. A random choice of b_j is preferable for the formation of the seed particle. This seed particle is used as the center of the swarm and a cloud of particles is generated randomly around the seed particle. It should be noted that the distance of the randomly generated particles should not be far from the seed particle. In this way, the initial swarm contains particles which have high chances of being near the optimal solution.

3.4 Constrained PSO for Digital Filter Data-Paths

This section is concerned with the constraints associated with PSO of digital filter data-paths. These constraints can be classified into two categories. One constraint is related to the range of values that a coordinate in the position of a particle can be changed. The other constraint is associated with the functional unit violation that may occur in the course of PSO. These two constraints are discussed in the following two subsections.

3.4.1 Coordinates Limit

Due to the random nature of operations in the PSO algorithm, the position value of a particle can be any infinite-precision number. In the optimization of digital filter data-paths, the resulting particle position cannot take any value and it should be constrained to a specific number system.

Partition P^1 in the particle is constructed with coordinates containing the delay from the first possible time-step where an operation may be scheduled. This delay is an integer number and cannot be less than zero. Therefore, the first constraint can be applied by limiting the coordinate values in partition P^1 to be nonnegative integers.

Partition P^2 in the particle, on the other hand, is constructed with coordinates containing the operator number and a sign value that can take the values of $+1$ or -1. Let us consider N_{add} represent the number of physical adder cells available for the implementation of the digital filter data-path and N_{mult} represent the total number of physical digital multipliers present in the implementation of the digital filter data-path. The numbers N_{add} and N_{mult} are fixed by the designer as a part of the input specification. Then, the coordinate containing an adder as an operator can be varied in the intervals $[-N_{\text{add}}, -1]$ and $[1, N_{\text{add}}]$, and the coordinate containing a multiplier as an operator can be varied in the intervals $[-N_{\text{mult}}, -1]$ and $[1, N_{\text{mult}}]$. Taking into consideration these constraints, the new PSO algorithm will be:

$$\hat{v}_{kj}^{i} = [w\hat{v}_{kj}^{i-1} + c_1 r_1(\hat{x}_{best_{kj}}^{i-1} - \hat{x}_{kj}^{i-1}) + c_2 r_2(\hat{g}_{best_j}^{i-1} - \hat{x}_{kj}^{i-1})] \tag{3.2}$$

$$\text{if} \quad \hat{v}_{kj}^{i} < \hat{v}_{min} \quad ; \quad \hat{v}_{kj}^{i} = \hat{v}_{min}$$

$$\text{if} \quad \hat{v}_{kj}^{i} > \hat{v}_{max} \quad ; \quad \hat{v}_{kj}^{i} = \hat{v}_{max}$$

$$\hat{x}_{kj}^{i} = \hat{x}_{kj}^{i-1} + \hat{v}_{kj}^{i} \tag{3.3}$$

$$\text{if} \quad \hat{x}_{kj}^{i} < \hat{x}_{j_{min}} \quad ; \quad \hat{x}_{kj}^{i} = \hat{x}_{j_{min}}$$

$$\text{if} \quad \hat{x}_{kj}^{i} > \hat{x}_{j_{max}} \quad ; \quad \hat{x}_{kj}^{i} = \hat{x}_{j_{max}}$$

Here, $\hat{x}_{kj}$, $\hat{v}_{kj}$, $\hat{x}_{best_{kj}}$, $\hat{g}_{best_j}$, $\hat{v}_{min}$, $\hat{v}_{max}$, $\hat{x}_{j_{min}}$ and $\hat{x}_{j_{max}}$ are all integer values where $\hat{v}_{min} < 0$ and $\hat{v}_{max} > 0$. In the above equations, $\hat{x}_{j_{min}}$ and $\hat{x}_{j_{max}}$ can be found by using the partition information, i.e. if the coordinate is in partition P^1, then $\hat{x}_{j_{min}} = 0$ and there is no limit for $\hat{x}_{j_{max}}$. On the other hand, if the coordinate is in partition P^2 and the coordinate represents an adder, then $\hat{x}_{j_{min}} = -N_{\text{add}}$ and $\hat{x}_{j_{max}} = N_{\text{add}}$, while if the coordinate is in partition P^2 and the coordinate represents a multiplier, then $\hat{x}_{j_{min}} = -N_{\text{mult}}$ and $\hat{x}_{j_{max}} = N_{\text{mult}}$. One may notice that there is a possibility that the value of a coordinate become zero in partition P^2 which violates the conditions discussed before. Random selection of a value for the corresponding

coordinate between the values $+1$ and -1 can solve this problem.

3.4.2 Functional Unit Satisfaction

In the course of optimization of digital filter data-paths, one can easily notice that sometimes the optimization results in particles suffering from concurrent assignment of the same functional unit to two or more operations in the same time-step. One way to resolve this situation is to include a violation term in the cost function of the digital filter particle. The other way is to reassign the functional units for violating operations only [105]. Additionally, one can keep the operator number unchanged and increment the time-step where the operation may be scheduled until it finds a free slot. In this thesis, the third scheme is used which looks practical in the implementation of PSO for high-level synthesis of digital filters.

3.5 Formulation of the Cost Function for Digital Filter Data-Paths

This section is concerned with the calculation of the cost function associated with the above data-path encoded swarm. The desired cost function can be expressed as a linear combination of the cost C_1 associated with the hardware requirement, and the cost C_2 associated with the time requirement of the digital filter data-path in accordance with:

$$C = \omega_1 C_1 + \omega_2 C_2 \tag{3.4}$$

where ω_1 and ω_2 are user-specified weighting factors.

3.5.1 Evaluation of the Cost Associated with the Hardware Requirements of the Digital Filter Data-Path

The digital filter data-path consists of digital adders and multipliers as the constituent arithmetic functional units. In order to facilitate data-transfer, additional hardware resources such as registers and multiplexors are also required. In this way, the cost associated with the hardware requirement of the data-path can be

computed in accordance with:

$$C_1 = A_{\text{add}} + A_{\text{mult}} + A_{\text{reg}} + A_{\text{mux}} \tag{3.5}$$

where A_{add}, A_{mult}, A_{reg} and A_{mux} represent the cost associated with the constituent digital adders, multipliers, registers, and multiplexors, respectively.

Digital Adder and Multiplier Costs

Let G_{add} represent the number of gate-equivalents associated with the implementation of a basic full adder. Then, the total cost associated with the digital adder present in the data-path is computed in accordance with:

$$A_{\text{add}} = W N_{\text{add}} G_{\text{add}} \tag{3.6}$$

where W is the signal wordlength. The total cost associated with the digital multipliers may be obtained in accordance with:

$$A_{\text{mult}} = N_{\text{mult}} G_{\text{mult}} \tag{3.7}$$

where G_{mult} represents the total number of gate-equivalents required in the implementation of each of these multipliers.

Register Costs

The computation of the costs associated with the registers and multiplexors requires the knowledge of the life-times of the various variables in the DFG. Let the time-step where a signal is first generated by an operation in the DFG be represented by l_{begin}, and let the time-step when the same signal is last consumed by an operation in the DFG be represented by l_{end}. In a bit-parallel implementation, a signal is active for one time-step from the time-step when it is last used in the DFG. In this way, the life-time of the signal in the scheduled DFG spans over the time-interval $[l_{\text{begin}}, l_{\text{end}}]$. By using this life-time information, the REAL algorithm [53] can be used to determine the number N_{reg} of registers required. REAL implements an algorithm for register allocation based on track assignment in routing. The algorithm is referred to as the left edge algorithm and has been proven optimal [106]. The track problem assignment is solved as follows [53]:

- Sort the wire segments in increasing order of their left edges.
- Assign the first segment (the leftmost edge) to the first track.
- Find the first wire whose left edge is to the right of the last selected wire and assign it to the current track.
- If no more wires can be assigned to the current track, start a new track and begin again from the second step. Repeat until all wires are assigned to tracks.

Although the left edge algorithm is based on a greedy search, it gives optimal results and the goal of it is to allocate the wire segments to tracks so as to minimize the total number of needed tracks.

For DFGs with no delay-free loops or conditional branches, the register allocation problem is the same as the track assignment as described above. In this thesis, the registers are modeled as tracks, and l_{begin} and l_{end} are modeled as the left and right edges of wires, respectively. The set of variables and their life-times can be used to build a life-time table. Given the life-time table for a DFG the goal is to assign variables to registers so as to minimize the total number of registers needed to store the variables. Two variables cannot share a register if they overlap in time.

Let the number of gate-equivalents required for the implementation of a unit register (a D-flip-flop) be represented as G_{reg}. Then, the cost associated with the registers is computed in accordance with:

$$A_{\text{reg}} = W N_{\text{reg}} G_{\text{reg}} \tag{3.8}$$

Multiplexor Costs

The various operations in the DFG can either be performed by different functional units, or the same functional unit can be used to perform different operations. In the latter case, the functional unit has to be multiplexed among different operations. The final digital filter data-path will usually consist of a few classes of multi-input multiplexors. The number of required multiplexors in each class is determined from the knowledge of the time-steps in which the various operations are scheduled in the DFG. Mainly, there are three types of multiplexors in the DFG.

- *Multiplexors connected to the inputs of the adders.* In order to calculate the type of multiplexors that are connected to the inputs of an adder, the variables that are connected to the first input of the adder are listed. The next step is to find the registers in which the corresponding variables are stored. The number of different registers that has been found shows the number of inputs to the multiplexor. This process is repeated for the second input of the adder. Therefore, a maximum of two multiplexors is needed for an adder and the number of inputs to the multiplexors can be calculated as described above.

- *Multiplexors connected to the inputs of the multipliers.* Let us consider the multiplier coefficients are stored in an external memory. This external memory is connected to one of the inputs of the multiplier. Therefore, a maximum of one multiplexor is associated with the inputs of a multiplier. To find the number of inputs for this multiplexor, one can perform a same procedure as discussed in the previous item, i.e. the variables connected to the input of the multiplier are listed and their corresponding registers are found. The number of different registers equals the number of inputs to the multiplexor.

- *Multiplexors connected to the inputs of the registers.* Each register needs at most one multiplexor. The number of inputs to each multiplexor can be found by using the information in the lifetime table. Each register stores a number of variables. Each variable correspond is either an output of an adder or a multiplier, or it is the input of the system. In either case, the number of different adders, multipliers, and inputs of the system, corresponding to the variables associated with each register is the number of inputs of those multiplexors that have to be allocated at the input of the registers.

Let N_{mux_p} represent the number of p-input multiplexors in the resulting digital filter data-path, and let G_{mux_p} represent the number of gate-equivalents required in the implementation of a p-input multiplexor. The cost associated with the multiplexors can be calculated in accordance with:

$$A_{\text{mux}} = W \sum_{p} N_{\text{mux}_p} G_{\text{mux}_p} \tag{3.9}$$

3.5.2 Evaluation of the Cost Associated with the Time Requirements of the Digital Filter Data-Path

The cost C_2 is evaluated in terms of the total number of time-steps required to implement the digital filter data-path, determined by the assignment of the operations constituting the DFG to various time-steps.

It should be noted that two distinct operations in the DFG cannot be bound to the same functional unit unless these operations are scheduled at least one time-step apart. In addition, the computational delay associated with the functional unit has to be taken into consideration. This computational delay was calculated in [107]. The duration T_{clk} of each time-step associated with the above digital filter data-path can be calculated as [108]

$$T_{\text{clk}} = t_{\text{DFF}} + 3t_{\text{G}} + 2t_{\text{mux}} + Wt_{\text{FA}} \tag{3.10}$$

where t_{DFF}, t_{G}, t_{mux} and t_{FA} represent the propagation delays through a D-flip-flop, a typical gate, a two-input multiplexor, and a full adder, respectively. Then C_2 may be obtained as:

$$C_2 = t_{\text{total}} T_{\text{clk}} \tag{3.11}$$

where t_{total} represents the total number of time-steps required to implement the digital data-path. Having determined the cost functions C_1 and C_2 in Eqns. (3.5) and (3.11), one can determine the cost function associated with the digital filter data-path by using Eqn. (3.4). The result is then used by the PSO algorithm to optimize the data-path.

3.6 Summary

This chapter has presented a novel technique for high-level synthesis of digital filters using particle swarm optimization technique. In this technique, a powerful encoding scheme is presented that is capable of turning the information in a digital filter data-paths into particles which can be used by PSO algorithm to optimize the digital filter data-paths. The encoding scheme is in such a way that preserves the data-dependency relationships present in the DFG of the digital filter. This is

achieved by sorting the operations in the DFG by their ALAP scheduling time-steps which takes into account the critical path information of the operations in the digital filter data-path. The particle is divided into two parts where the first part contains the information associated with the scheduling of different operations in the digital filter data-path, and the second part is carrying the information regarding the allocation of different operators to operations and the ordering of signals for symmetrical operations in the digital filter DFG.

There are two problems concerning the PSO for high-level synthesis of digital filter data-paths. On one hand, the coordinates of each particle is bound to certain integer numbers which depend on the user-specified constraints. On the other hand, there is a possibility that in the course of PSO, two or more operators are allocated for a single operation in a time-step. This is due to the random nature of operations of addition and subtraction in the underlying PSO. To avoid this, the time-step of the violating operation is increased until it reaches an empty spot.

The cost associated with the PSO for high-level synthesis of digital filters is formed to obtain an area-optimal, time-optimal, or combined area-cum-time-optimal solution. In order to achieve this, the number of support cells (registers, and multiplexors) which are required for the high-level synthesis of the digital filter data-path is calculated and the cost associated with the hardware implementation of the arithmetic functional units together with the support cells is evaluated. In addition, the time requirements of the digital filter data-path is considered. This will provide PSO with an objective function to obtain a combined area-cum-time-optimal digital filter data-path.

Chapter 4

Application Examples

In chapters 2 and 3, the design and optimization of FRM digital filters, and the high-level synthesis of digital filters using PSO were discussed, respectively. This chapter is concerned with the application of PSO to the design, high-level synthesis, and optimization of digital filters by a number of practical examples.

Two sets of lowpass FRM digital filters are used, with the first one having an FIR interpolation subfilter and the second one having an IIR interpolation subfilter. Moreover, two sets of bandpass FRM digital filters are utilized, where again, one having FIR digital filters as interpolation subfilters and the other having IIR digital filters as interpolation subfilters. In case of IIR digital subfilters, the bilinear-LDI technique is used to realize the FRM IIR digital filter. In addition, an example is made to illustrate the application of PSO to the high-level synthesis of a benchmark digital filter.

This chapter proceeds as follows. In section 4.1, an example is presented to show the application of PSO to the design and optimization of a lowpass FRM FIR digital filter. Section 4.2 presents an example illustrating the design and optimization of a bandpass FRM FIR digital filter using PSO. In section 4.3, the application of PSO to the design and optimization of a lowpass FRM IIR digital filter is presented. In this section, the design parameters are the same as the design parameters of the example in section 4.1 and comparisons are made to show the improvement in PSO of FRM IIR digital filter. In addition, a comparison is made between PSO and GA in this application. In section 4.4, PSO is applied to a bandpass FRM IIR digital filter, while the design parameters are the same as the example in section 4.2. Com-

parisons are made to illustrate the validity of the proposed technique. Section 4.5 presents an example to show the application of PSO to the high-level synthesis of a benchmark elliptic wave digital (WD) filter. Finally, section 4.6 presents a brief summary of this chapter.

4.1 Lowpass FRM FIR Digital Filter Design Example

This section is concerned with the design and optimization of a lowpass FRM FIR digital filter satisfying the magnitude response design specifications given in Table 4.1 over the CSD multiplier coefficient space.

The parameters for the PSO of lowpass FRM FIR digital filter is shown in Table 4.2 and the CSD parameters are presented in Table 4.3.

The first step to design the FRM FIR digital filter is to find the length of the digital subfilters $H_a(z)$, $F_0(z)$ and $F_1(z)$. Given the design specification in Table 4.1, The lengths of the digital subfilters $H_a(z)$, $F_0(z)$ and $F_1(z)$ are found to be 79, 24, and 42, respectively (based on Parks-McClellan approach), resulting in $N = 145$. The passband and stopband edge frequencies of the digital subfilters $H_a(z)$, $F_0(z)$ and $F_1(z)$ are determined by using the design equations given in [3]. Moreover, the passband ripple and stopband loss of these subfilters are set at 85% of the corresponding values given in the design specifications in Table 4.1 (in order to account for any second-order effects when using the design equations in [3]). In this way,

Table 4.1: Design Specifications for Lowpass FRM FIR Digital Filter

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	0.61π[Rad]
Normalized Sampling Period T	1[s]
Interpolation Factor M	6

Table 4.2: PSO Design Parameters for Lowpass FRM FIR Digital Filter

K	w	c_1	c_2	$\hat{v}_{min}$	$\hat{v}_{max}$	L_f	L_h
700	0.4	2	2	-5	5	10	10

Table 4.3: CSD Parameters for Lowpass FRM FIR Digital Filter

L_0	l_0	f_0
11	3	10

the derived design specifications for the digital subfilters $H_a(z)$, $H_b(z)$, $F_0(z)$ and $F_1(z)$ are obtained as shown in Table 4.4.

Finally, by using Parks McClellan approach, the subfilters $H_a(z)$, $F_0(z)$ and $F_1(z)$ can be designed. Consequently, the magnitude frequency response of the overall infinite-precision lowpass FRM FIR digital filter $H(z)$ is obtained as shown in Fig. 4.1. Based on the infinite-precision lowpass FRM FIR digital filter, the corresponding CSD FRM FIR initial digital filter is obtained through rounding the infinite-precision multiplier coefficient values to their closest CSD values. The resulting CSD FRM FIR digital filter has a magnitude frequency response as shown in Fig. 4.2

By applying the proposed PSO to the above CSD FRM FIR digital filter and after about 100 iterations, the discrete PSO converges to the optimal lowpass FRM FIR digital filter having a magnitude frequency response as shown in Fig. 4.3. In addition, Fig. 4.4 gives us a closer look at the magnitude frequency response in the passband region of the lowpass FRM FIR digital filter.

Table 4.5 represents the comparison of the CSD lowpass FRM FIR digital filters before and after PSO.

Table 4.4: Band-Edge Frequencies, Passband Ripples and Stopband Losses for Digital Subfilters $H_a(z)$, $H_b(z)$, $F_0(z)$ and $F_1(z)$ for Lowpass FRM FIR digital filter

Subfilter	Passband Edge Frequency	Stopband Edge Frequency	Passband Ripple	Stopband Loss
$H_a(z)$	0.34π	0.4π	0.085 dB	46 dB
$H_b(z)$	0.4π	0.34π	0.085 dB	46 dB
$F_0(z)$	0.4π	0.6π	0.085 dB	46 dB
$F_1(z)$	0.61π	0.723π	0.085 dB	46 dB

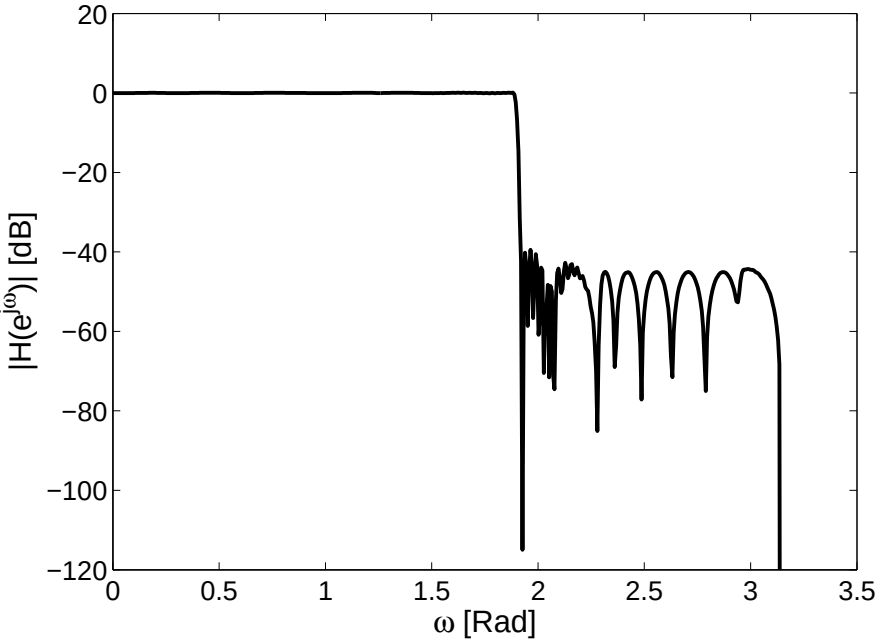

Figure 4.1: Magnitude Frequency-Response of the Overall Infinite-Precision Lowpass FRM FIR Digital Filter $H(e^{j\omega})$

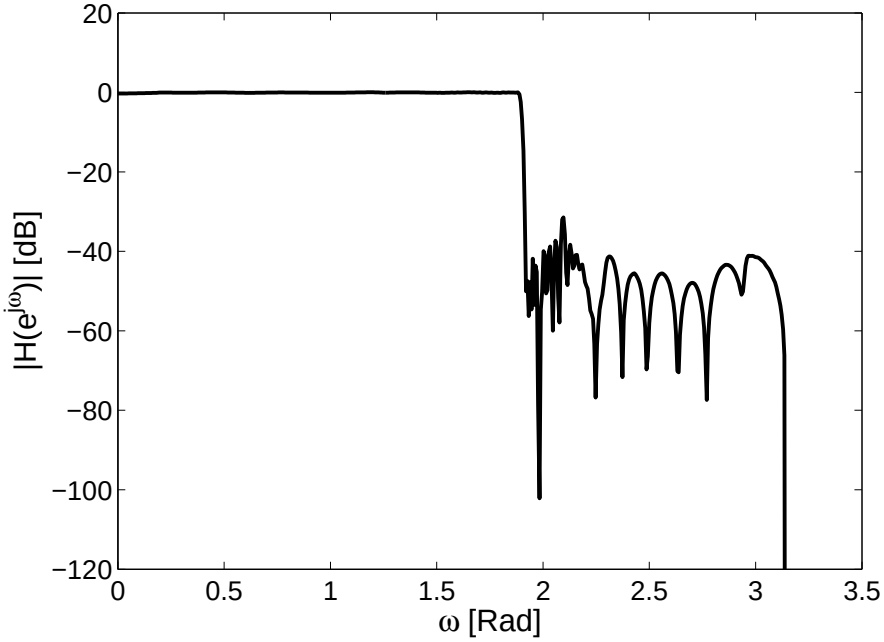

Figure 4.2: Magnitude Frequency-Response of the Overall Lowpass CSD FRM FIR Digital Filter $H(e^{j\omega})$ Before PSO

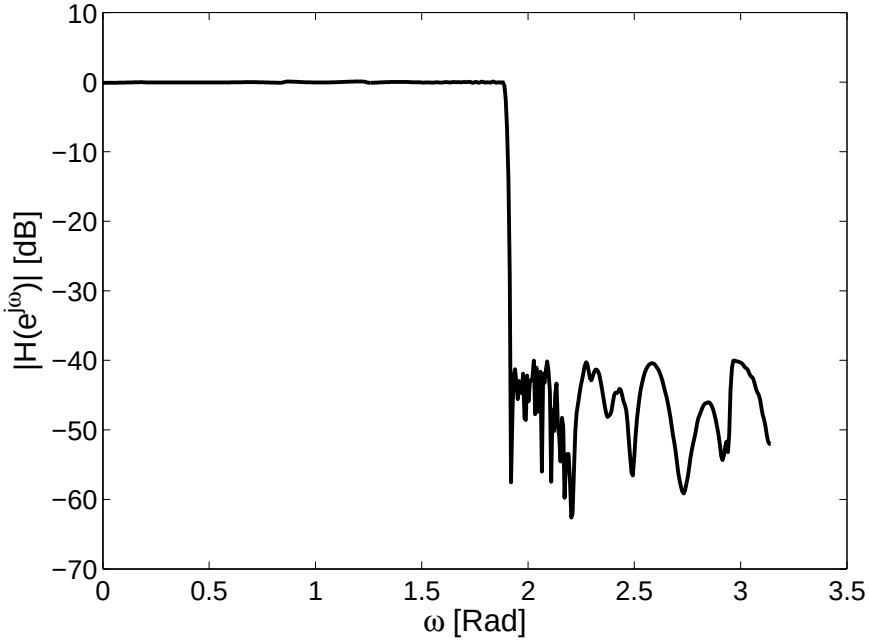

Figure 4.3: Magnitude Frequency-Response of the Overall Lowpass CSD FRM FIR Digital Filter $H(e^{j\omega})$ After PSO

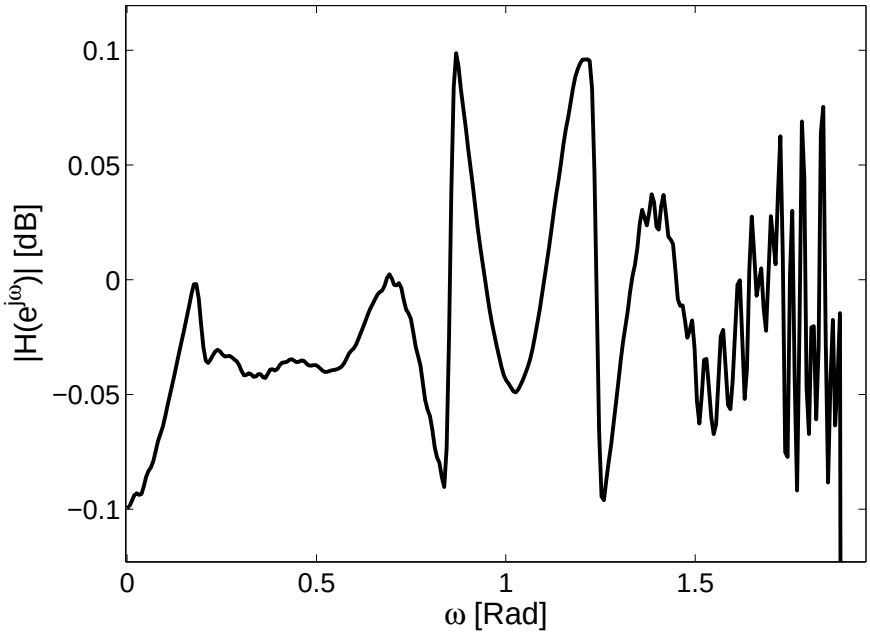

Figure 4.4: Magnitude Frequency-Response in the Passband Region of the Overall Lowpass CSD FRM FIR Digital Filter $H(e^{j\omega})$ After PSO

Table 4.5: Frequency-Response Analysis of the Lowpass CSD FRM FIR Digital Filter Before and After PSO

Frequency-Response Characteristic	Before PSO	After PSO
Maximum Passband Ripple A_p	0.2788[dB]	0.0996[dB]
Minimum Stopband Loss A_a	31.4681[dB]	40.0269[dB]

4.2 Bandpass FRM FIR Digital Filter Design Example

In this section, the design of a bandpass FRM FIR digital filter over the CSD multiplier coefficient space is considered. The given magnitude response design specifications are as given in Table 4.6.

The parameters for the PSO of bandpass FRM FIR digital filter is shown in Table 4.7 and the CSD parameters are presented in Table 4.8.

As before, the first step to the design of the bandpass FRM FIR digital filter is to find the lengths of the digital subfilters $H_{a_{lp}}(z)$, $F_{0_{lp}}(z)$, $F_{1_{lp}}(z)$, $H_{a_{hp}}(z)$, $F_{0_{hp}}(z)$ and $F_{1_{hp}}(z)$. Using the design specifications given in Table 4.6, The lengths of the digital subfilters $H_{a_{lp}}(z)$, $F_{0_{lp}}(z)$ and $F_{1_{lp}}(z)$ are found to be 79, 24, and 42, respectively. Also, the lengths of the digital subfilters $H_{a_{hp}}(z)$, $F_{0_{hp}}(z)$ and $F_{1_{hp}}(z)$ are found to be 49, 23, and 35, respectively, resulting in $N = 252$. The passband ripple and stopband loss of these subfilters are set at 85% of the corresponding values given in Table 4.6. In this way, the derived design specifications for the digital subfilters $H_{a_{lp}}(z)$, $F_{0_{lp}}(z)$, $F_{1_{lp}}(z)$, $H_{a_{hp}}(z)$, $F_{0_{hp}}(z)$ and $F_{1_{hp}}(z)$ are obtained as shown in Table 4.9.

Finally, by using Parks McClellan approach, the subfilters $H_{a_{lp}}(z)$, $F_{0_{lp}}(z)$, $F_{1_{lp}}(z)$, $H_{a_{hp}}(z)$, $F_{0_{hp}}(z)$ and $F_{1_{hp}}(z)$ can be designed. The magnitude frequency response of the overall infinite-precision bandpass FRM FIR digital filter $H(z)$ is as shown in Fig. 4.5.

Based on the infinite-precision bandpass FRM FIR digital filter, the corresponding CSD FRM FIR digital filter is obtained to have a magnitude frequency response as shown in Fig. 4.6

By applying the proposed discrete PSO to the above CSD FRM FIR digital filter and after about 200 iterations, discrete PSO converges to the optimal bandpass FRM FIR digital filter having a magnitude frequency response as shown in Fig. 4.7. Fig. 4.8 gives us a closer look at the magnitude frequency response in the passband region of the bandpass FRM FIR digital filter.

Table 4.10 compares the CSD bandpass FRM FIR digital filters before and after

Table 4.6: Design Specifications for Bandpass FRM FIR Digital Filter

Maximum Passband Ripple A_p	0.1[dB]
Minimum Stopband Loss A_a	40[dB]
Lower Stopband-Edge Normalized Frequency ω_{a_1}	0.31π[Rad]
Lower Passband-Edge Normalized Frequency ω_{p_1}	0.33π[Rad]
Upper Passband-Edge Normalized Frequency ω_{p_2}	0.60π[Rad]
Upper Stopband-Edge Normalized Frequency ω_{a_2}	0.61π[Rad]
Normalized Sampling Period T	1[s]
Lowpass Filter Interpolation Factor M_{lp}	6
Highpass Filter Interpolation Factor M_{hp}	5

Table 4.7: PSO Design Parameters for Bandpass FRM FIR Digital Filter

K	w	c_1	c_2	$\hat{v}_{min}$	$\hat{v}_{max}$	L_f	L_h
700	0.4	2	2	-5	5	10	10

Table 4.8: CSD Parameters for Bandpass FRM FIR Digital Filter

L_0	l_0	f_0
11	3	10

Table 4.9: Band-Edge Frequencies, Passband Ripples and Stopband Losses for Digital Subfilters $H_{a_{lp}}(z)$, $H_{b_{lp}}(z)$, $F_{0_{lp}}(z)$, $F_{1_{lp}}(z)$, $H_{a_{hp}}(z)$, $H_{b_{hp}}(z)$, $F_{0_{hp}}(z)$ and $F_{1_{hp}}(z)$ for Bandpass FRM FIR Digital Filter

Subfilter	Passband Edge Frequency	Stopband Edge Frequency	Passband Ripple	Stopband Loss
$H_{a_{lp}}(z)$	0.34π	0.4π	0.085 dB	46 dB
$H_{b_{lp}}(z)$	0.4π	0.34π	0.085 dB	46 dB
$F_{0_{lp}}(z)$	0.4π	0.6π	0.085 dB	46 dB
$F_{1_{lp}}(z)$	0.61π	0.723π	0.085 dB	46 dB
$H_{a_{hp}}(z)$	0.35π	0.45π	0.085 dB	46 dB
$H_{b_{hp}}(z)$	0.45π	0.35π	0.085 dB	46 dB
$F_{0_{hp}}(z)$	0.31π	0.09π	0.085 dB	46 dB
$F_{1_{hp}}(z)$	0.47π	0.33π	0.085 dB	46 dB

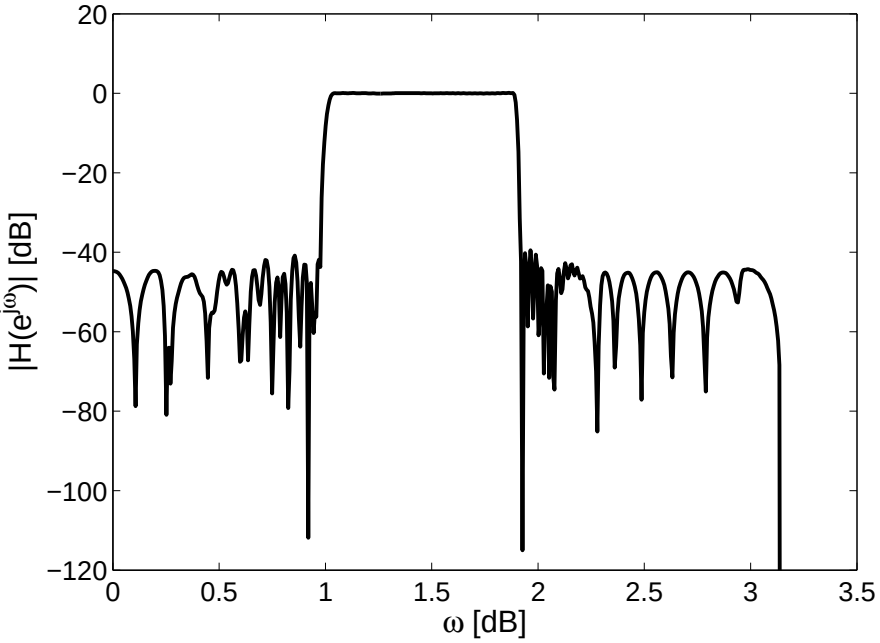

Figure 4.5: Magnitude Frequency-Response of the Overall Infinite-Precision Bandpass FRM FIR Digital Filter $H(e^{j\omega})$

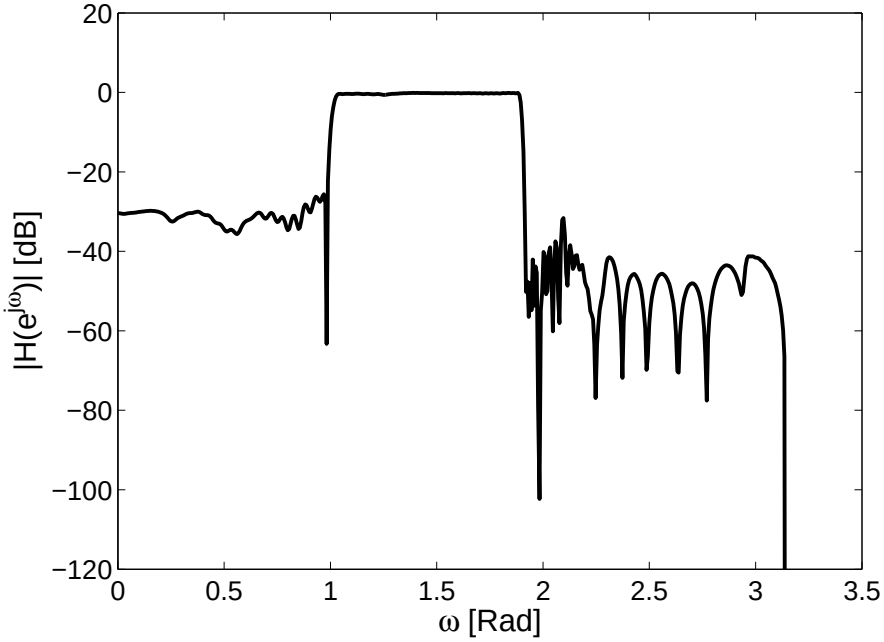

Figure 4.6: Magnitude Frequency-Response of the Overall Bandpass CSD FRM FIR Digital Filter $H(e^{j\omega})$ Before PSO

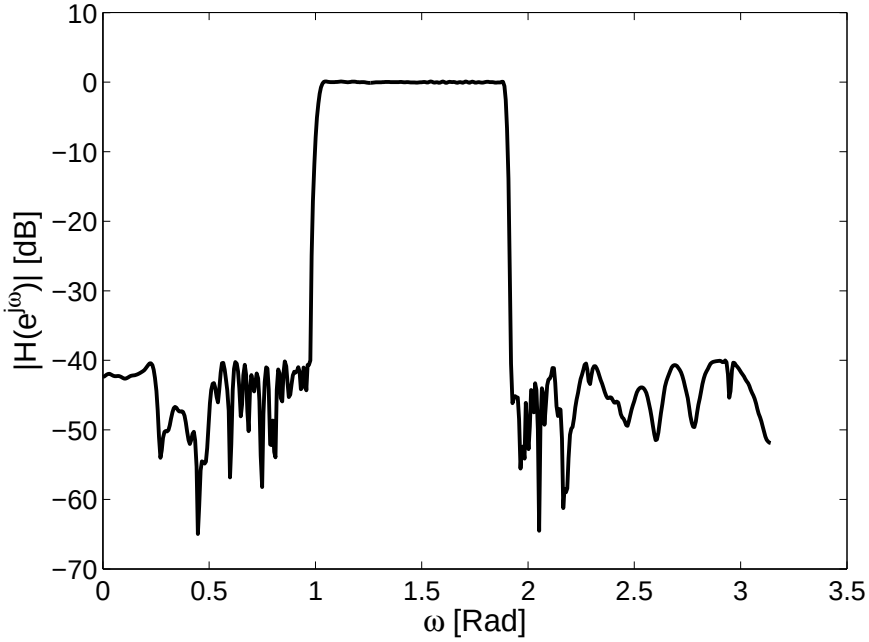

Figure 4.7: Magnitude Frequency-Response of the Overall Bandpass CSD FRM FIR Digital Filter $H(e^{j\omega})$ After PSO

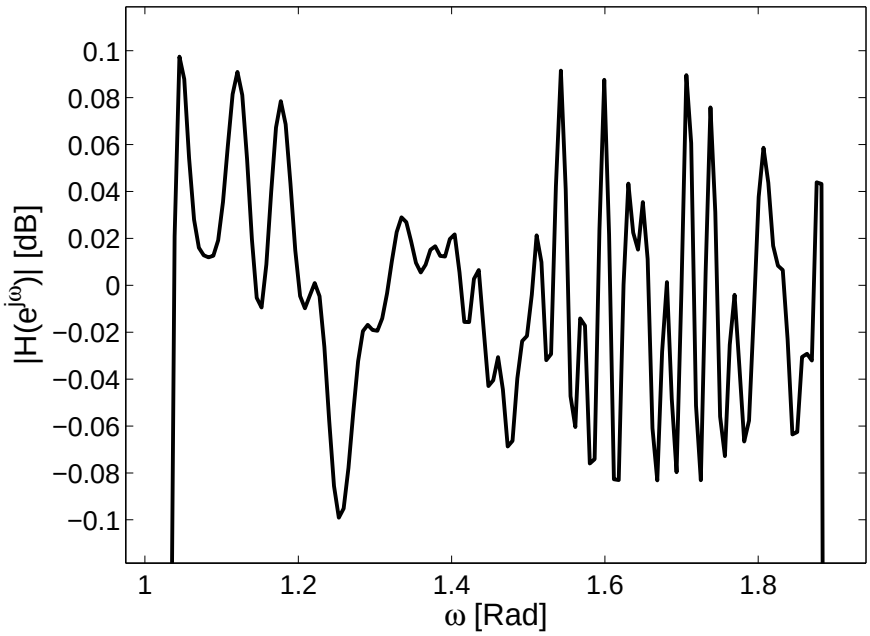

Figure 4.8: Magnitude Frequency-Response in the Passband Region of the Overall Bandpass CSD FRM FIR Digital Filter $H(e^{j\omega})$ After PSO

Table 4.10: Frequency-Response Analysis of the Bandpass CSD FRM FIR Digital Filter Before and After PSO

Frequency-Response Characteristic	Before PSO	After PSO
Maximum Passband Ripple A_p	0.6277[dB]	0.0991[dB]
Minimum Stopband Loss A_a	25.6378[dB]	40.0125[dB]

PSO.

4.3 Lowpass FRM IIR Digital Filter Design Example

Consider the design of a lowpass FRM IIR digital filter satisfying the magnitude response design specifications given in Table 4.11 over the CSD multiplier coefficient space.

The parameters for the PSO of lowpass FRM IIR digital filter is shown in Table 4.12 and the CSD parameters are presented in Table 4.13.

Given the design specification in Table 4.11, The order of the digital allpass

Table 4.11: Design Specifications for Lowpass FRM IIR Digital Filter

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	0.61π[Rad]
Normalized Sampling Period T	1[s]
Interpolation Factor M	6

Table 4.12: PSO Design Parameters for Lowpass FRM IIR Digital Filter

K	w	c_1	c_2	$\hat{v}_{min}$	$\hat{v}_{max}$	L_f	L_h
700	0.4	2	2	-5	5	10	10

Table 4.13: CSD Parameters for Lowpass FRM IIR Digital Filter

L_0	l_0	f_0	L_1	l_1	f_1
11	3	10	12	3	7

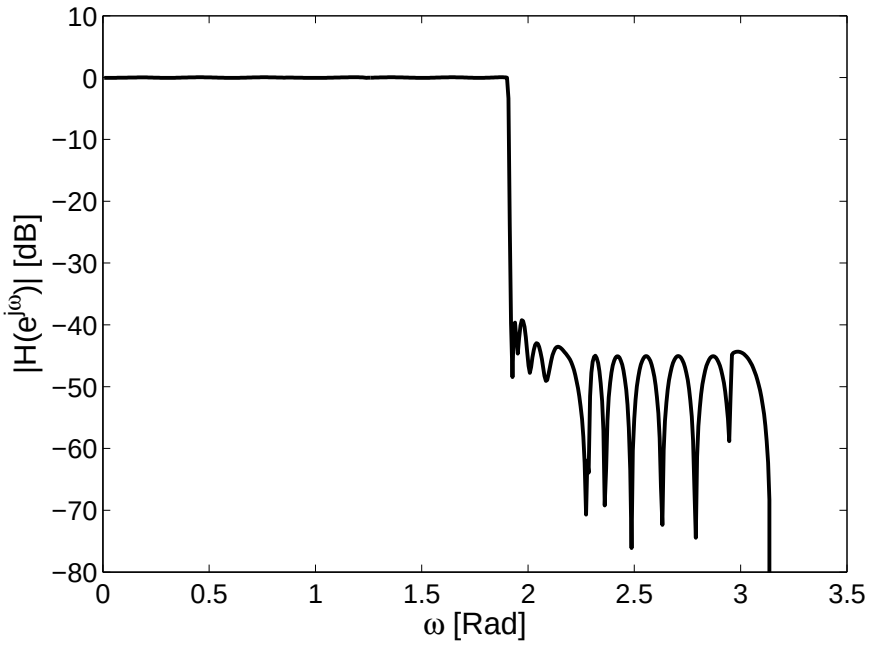

Figure 4.9: Magnitude Frequency-Response of the Overall Infinite-Precision Lowpass FRM IIR Digital Filter $H(e^{j\omega})$

networks $G_0(z)$ and $G_1(z)$ are found to be 5 and 4, respectively. In addition, the digital masking subfilters $F_0(z)$ and $F_1(z)$ have the same length as the previous example, i.e. 24 and 42, respectively, resulting in $N = 75$. In this example a set of ten CSD LUTs are required, nine LUTs for the multiplier coefficients $m_{C_{0,1}}$, $m_{C_{0,2}}$, $m_{C_{0,3}}$, $m_{L_{0,2}}$, $m_{L_{0,3}}$, $m_{C_{1,1}}$, $m_{L_{1,1}}$, $m_{C_{1,2}}$ and $m_{L_{1,2}}$ constituent in the digital allpass networks $G_0(z)$ and $G_1(z)$, and one template LUT for all the multiplier coefficients constituent in the masking digital subfilters $F_0(z)$ and $F_1(z)$.

Finally, by using Parks McClellan approach, the subfilters $F_0(z)$ and $F_1(z)$ can be designed. Also, by using the EMQF technique, the digital allpass networks $G_0(z)$ and $G_1(z)$ can be designed. Consequently, the magnitude and group delay frequency responses of the overall infinite-precision lowpass FRM IIR digital filter $H(z)$ is obtained as shown in Figs. 4.9 and 4.10.

Based on the infinite-precision lowpass FRM IIR digital filter, the corresponding CSD FRM IIR initial digital filter is obtained to have a magnitude and group delay frequency responses as shown in Figs. 4.11 and 4.12.

By applying the proposed PSO to the initial FRM IIR digital filter and after

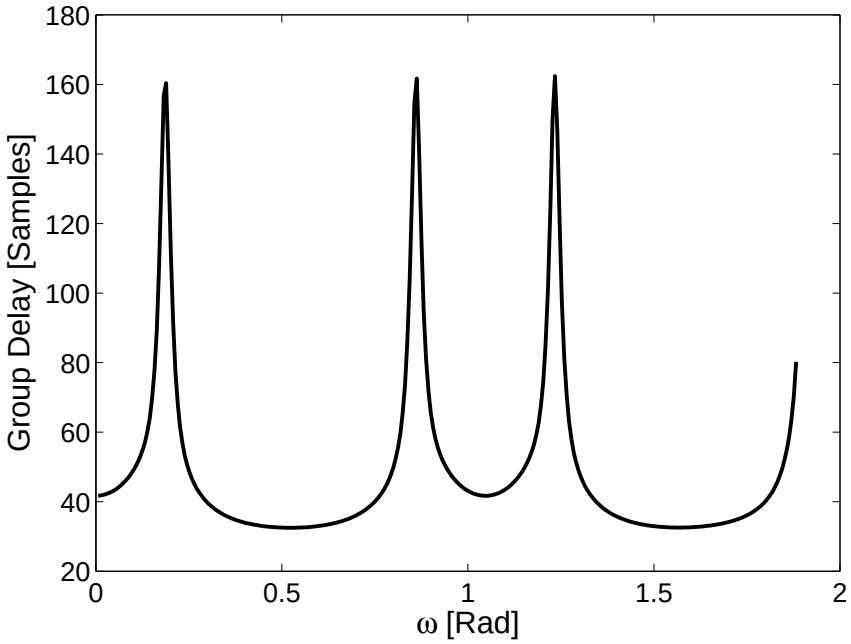

Figure 4.10: Group Delay Frequency-Response of the Overall Infinite-Precision Lowpass FRM IIR Digital Filter $H(e^{j\omega})$

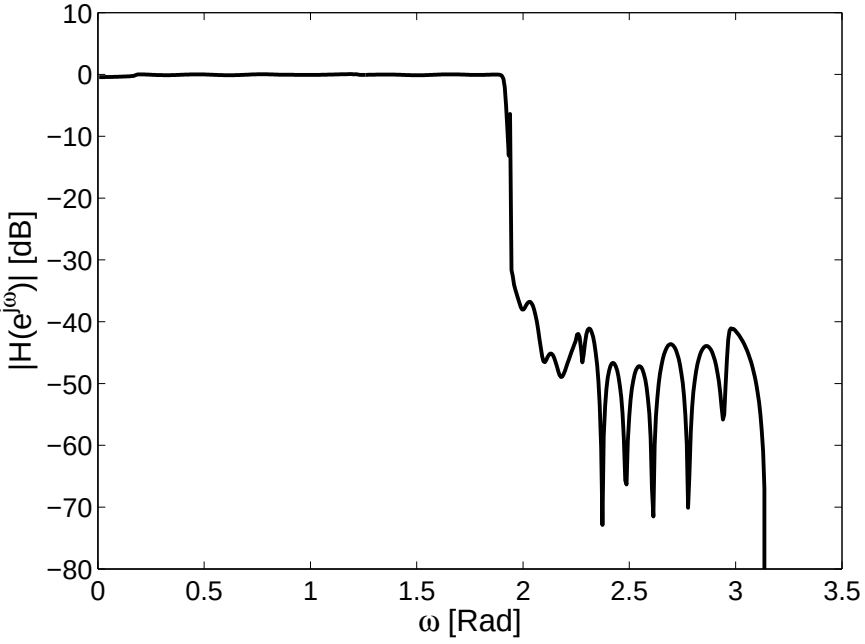

Figure 4.11: Magnitude Frequency-Response of the Overall CSD Lowpass FRM IIR Digital Filter $H(e^{j\omega})$ Before PSO

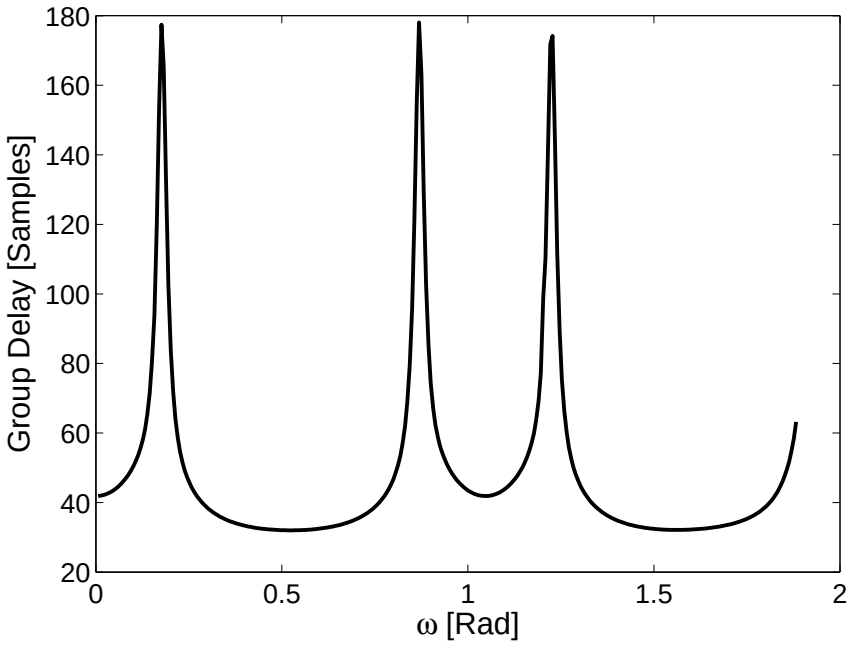

Figure 4.12: Group Delay Frequency-Response of the Overall CSD Lowpass FRM IIR Digital Filter $H(e^{j\omega})$ Before PSO

about 70 iterations, the PSO converges to the optimal lowpass FRM IIR digital filter having a magnitude frequency response as shown in Fig. 4.13. In addition, Fig. 4.14 gives us a closer look to the magnitude frequency response of the passband region of the lowpass FRM IIR digital filter. Fig. 4.15 illustrates the group delay frequency response of the optimized lowpass FRM IIR digital filter. The values of the multiplier coefficients $m_{L_{0\text{ or }1},p}$ and $m_{C_{0\text{ or }1},p}$ are obtained as summarized in Table 4.14.

Table 4.15 represents the comparison of the CSD lowpass FRM IIR digital filters before and after PSO.

Since the design specifications are the same for this example and lowpass FRM FIR digital filter example in section 4.1, comparisons can be made between these two filters in terms of the number of optimization variables and the speed of convergence as summarized in Table 4.16.

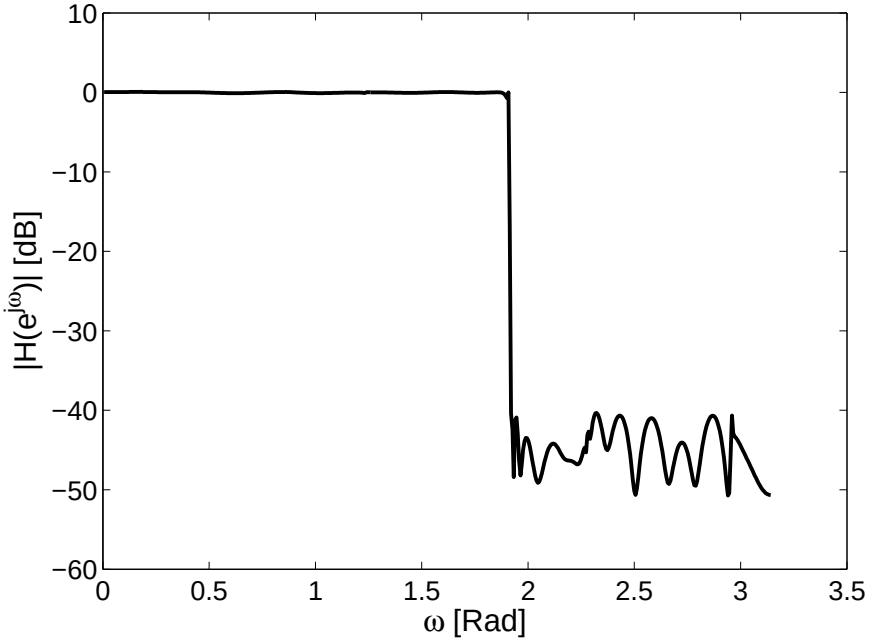

Figure 4.13: Magnitude Frequency-Response of the Overall CSD Lowpass FRM IIR Digital Filter $H(e^{j\omega})$ After PSO

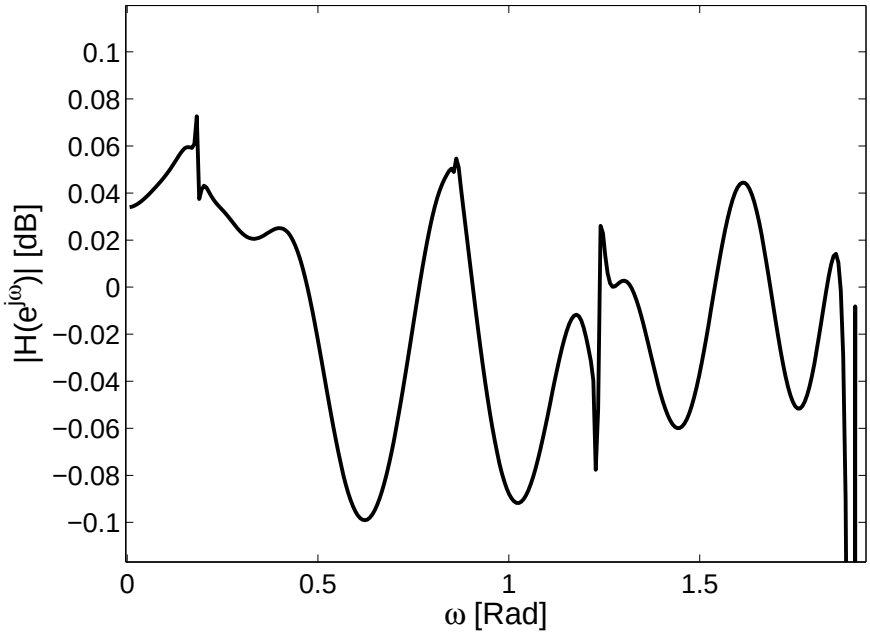

Figure 4.14: Magnitude Frequency-Response of the Passband Region of the Overall CSD Lowpass FRM IIR Digital Filter $H(e^{j\omega})$ After PSO

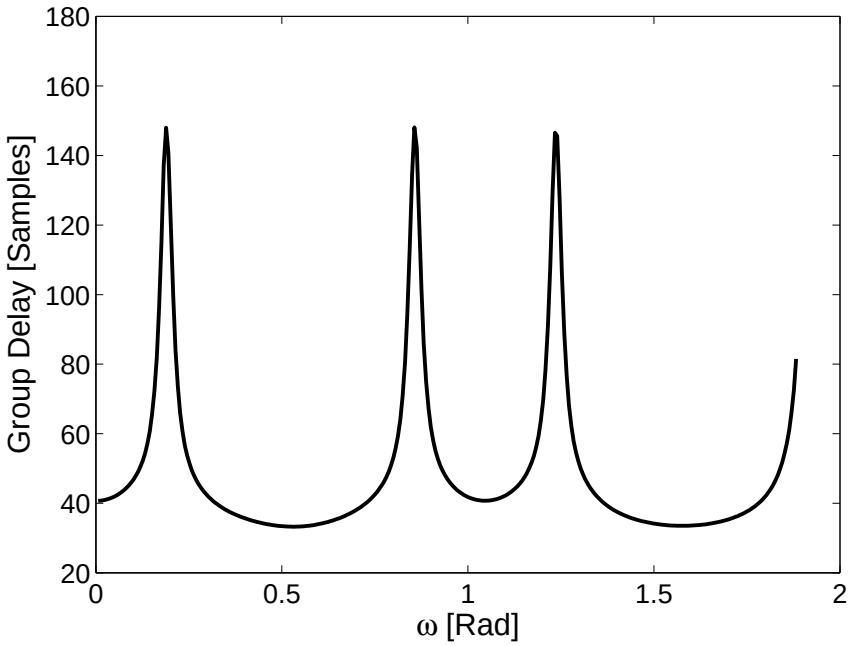

Figure 4.15: Group Delay Frequency-Response of the Overall CSD Lowpass FRM IIR Digital Filter $H(e^{j\omega})$ After PSO

Table 4.14: Digital Multiplier Values for Lowpass FRM IIR Digital Filter

Multiplier	CSD Representation	Decimal Value
$m_{C_{0,1}}$	$00001.000\bar{1}0\bar{1}0$	0.9219
$m_{C_{0,2}}$	10101.0000000	21
$m_{C_{0,3}}$	$00010.0\bar{1}000\bar{1}0$	1.7344
$m_{L_{0,2}}$	$00000.000100\bar{1}$	0.0547
$m_{L_{0,3}}$	$00000.1000\bar{1}0\bar{1}$	0.4609
$m_{C_{1,1}}$	$00001.00\bar{1}0\bar{1}00$	0.8438
$m_{C_{1,2}}$	01000.1000000	8.5
$m_{L_{1,1}}$	$00001.00\bar{1}0\bar{1}00$	0.8438
$m_{L_{1,2}}$	00000.0010010	0.1406

Table 4.15: Frequency-Response Analysis of the CSD Lowpass FRM IIR Digital Filter Before and After PSO

Frequency-Response Characteristic	Before PSO	After PSO
Maximum Passband Ripple A_p	0.4345[dB]	0.0991[dB]
Minimum Stopband Loss A_a	4.9451[dB]	40.336[dB]
Maximum Group Delay	178[Samples]	148[Samples]

Table 4.16: Comparison between PSO of Lowpass FRM FIR Digital Filter and PSO of Lowpass FRM IIR Digital Filter

Characteristic	Lowpass FRM FIR Digital Filter	Lowpass FRM IIR Digital Filter
Number of Optimization Variables N	145	75
Average Number of Iterations	100	70

4.3.1 Comparison with DCGA

In this section a comparison has been made between the proposed algorithm and the Diversity Controlled (DC) GA [23] for the optimization of the lowpass FRM IIR digital filter satisfying the design specifications in Table 4.11. This comparison was made because these two techniques are both LUT-based and the initialization of the algorithms are the same. The parameters for DCGA optimization of the lowpass FRM IIR digital filter are $c = 0.8$, $\alpha = 0.3$ and $\zeta = 0.4$ [109] and the number of chromosomes in the population pool is 1000.

In this comparison, 20 different trials are performed for the optimization of a lowpass FRM IIR digital filter while each trial contains 200 iterations. The average fitness value of the best particle in the swarm (best chromosome in the population pool) in these 20 trials is calculated in each iteration. The result is shown in Fig. 4.16.

Two observations can be obtained from this figure:

- The initialization of the two techniques is tried to be the same, i.e. the seed

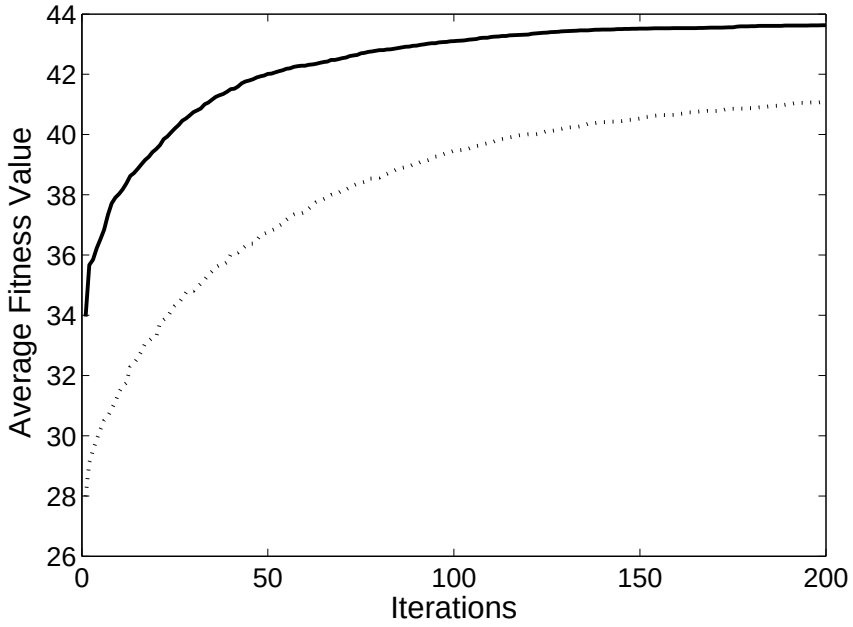

Figure 4.16: Comparison Between the Proposed PSO (Solid Line) and DCGA (Dotted Line) for the Optimization of a Lowpass FRM IIR Digital Filter

chromosome in DCGA optimization of the lowpass FRM IIR digital filter is obtained exactly as the seed particle is obtained in the proposed PSO. The population pool, then, is generated through random mutation of the seed chromosome with small mutation probability. As it can be seen in Fig. 4.16, the initialization of the proposed PSO is much better than that of the DCGA. This is an advantage for the proposed PSO in terms of finding the optimal solution while the population is still diverse.

- The speed of convergence for the proposed PSO is higher than the DCGA technique. This is obvious in Fig. 4.16 since the slope of the graph for the proposed PSO is greater than the slope of the graph for DCGA.

4.4 Bandpass FRM IIR Digital Filter Design Example

Consider the design of a bandpass FRM IIR digital filter satisfying the magnitude response design specifications given in Table 4.17 over the CSD multiplier coeffi-

Table 4.17: Design Specifications for Bandpass FRM IIR Digital Filter

Maximum Passband Ripple A_p	0.1[dB]
Minimum Stopband Loss A_a	40[dB]
Lower Stopband-Edge Normalized Frequency ω_{a_1}	0.31π[Rad]
Lower Passband-Edge Normalized Frequency ω_{p_1}	0.33π[Rad]
Upper Passband-Edge Normalized Frequency ω_{p_2}	0.60π[Rad]
Upper Stopband-Edge Normalized Frequency ω_{a_2}	0.61π[Rad]
Normalized Sampling Period T	1[s]
Lowpass Filter Interpolation Factor M_{lp}	6
Highpass Filter Interpolation Factor M_{hp}	5

Table 4.18: PSO Design Parameters for Bandpass FRM IIR Digital Filter

K	w	c_1	c_2	$\hat{v}_{min}$	$\hat{v}_{max}$	L_f	L_h
700	0.4	2	2	-5	5	10	10

cient space.

The parameters for the PSO of bandpass FRM IIR digital filter is shown in Table 4.18 and the CSD parameters are presented in Table 4.19.

Given the design specification in Table 4.17, The order of the digital allpass networks $G_{0_{lp}}(z)$, $G_{1_{lp}}(z)$, $G_{0_{hp}}(z)$ and $G_{1_{hp}}(z)$ are found to be 3, 4, 3 and 4, respectively. In addition, the digital masking subfilters $F_{0_{lp}}(z)$, $F_{1_{lp}}(z)$, $F_{0_{hp}}(z)$ and $F_{1_{hp}}(z)$ have the same length as the previous example, i.e. 24, 42, 25 and 35 respectively, resulting in $N = 140$. In this example a set of fifteen CSD LUTs are

Table 4.19: CSD Parameters for Bandpass FRM IIR Digital Filter

L_0	l_0	f_0	L_1	l_1	f_1
11	3	10	12	3	7

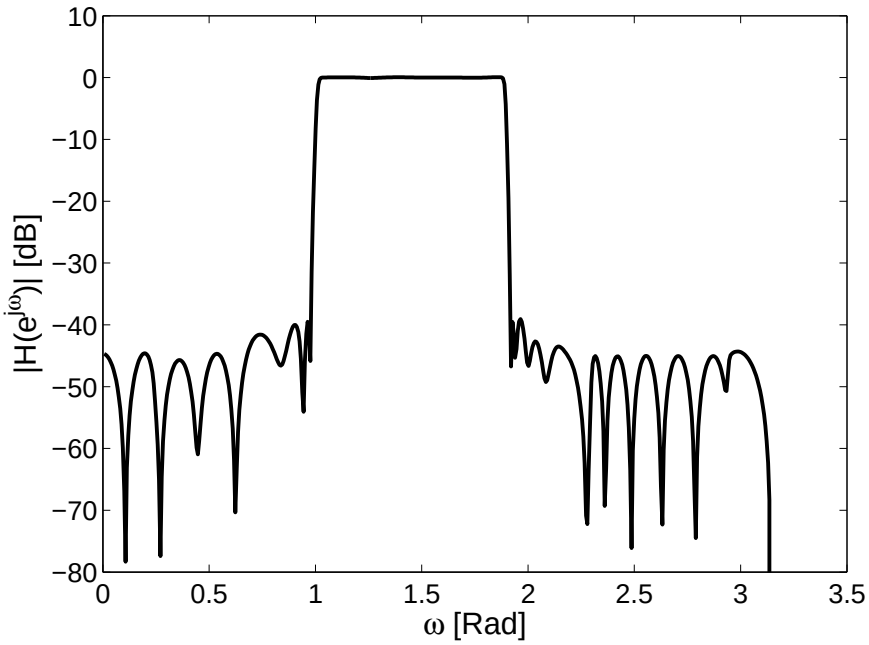

Figure 4.17: Magnitude Frequency-Response of the Overall Infinite-Precision Bandpass FRM IIR Digital Filter $H(e^{j\omega})$

required, fourteen LUTs for the multiplier coefficients $m_{C_{0,1}}$, $m_{C_{0,2}}$, $m_{C_{0,3}}$, $m_{L_{0,2}}$, $m_{L_{0,3}}$, $m_{C_{1,1}}$, $m_{L_{1,1}}$, $m_{C_{1,2}}$ and $m_{L_{1,2}}$ constituent in the digital allpass networks $G_{0_{lp}}(z)$, $G_{1_{lp}}(z)$, $G_{0_{hp}}(z)$ and $G_{1_{hp}}(z)$, and one template LUT for all the multiplier coefficients constituent in the masking digital subfilters $F_{0_{lp}}(z)$, $F_{1_{lp}}(z)$, $F_{0_{hp}}(z)$ and $F_{1_{hp}}(z)$.

Finally, by using Parks McClellan approach, the subfilters $F_{0_{lp}}(z)$, $F_{1_{lp}}(z)$, $F_{0_{hp}}(z)$ and $F_{1_{hp}}(z)$ can be designed. Also, by using the EMQF technique, the digital allpass networks $G_{0_{lp}}(z)$, $G_{1_{lp}}(z)$, $G_{0_{hp}}(z)$ and $G_{1_{hp}}(z)$ can be designed. Consequently, the magnitude and group delay frequency responses of the overall infinite-precision bandpass FRM IIR digital filter $H(z)$ is obtained as shown in Figs. 4.17 and 4.18.

Based on the infinite-precision bandpass FRM IIR digital filter, the corresponding CSD FRM IIR initial digital filter is obtained to have a magnitude and group delay frequency responses as shown in Figs. 4.19 and 4.20.

By applying the proposed PSO to the initial FRM IIR digital filter and after about 160 iterations, the PSO converges to the optimal bandpass FRM IIR digital

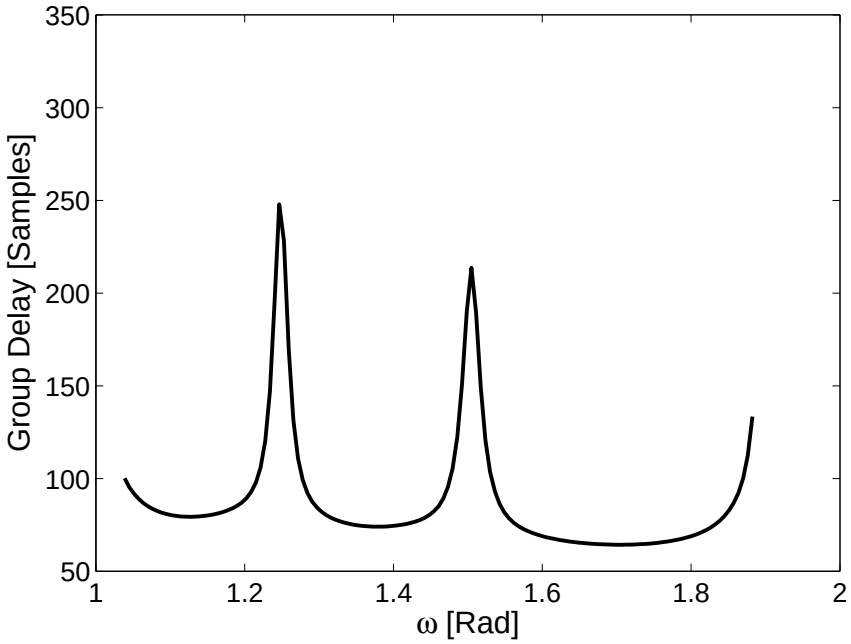

Figure 4.18: Group Delay Frequency-Response of the Overall Infinite-Precision Bandpass FRM IIR Digital Filter $H(e^{j\omega})$

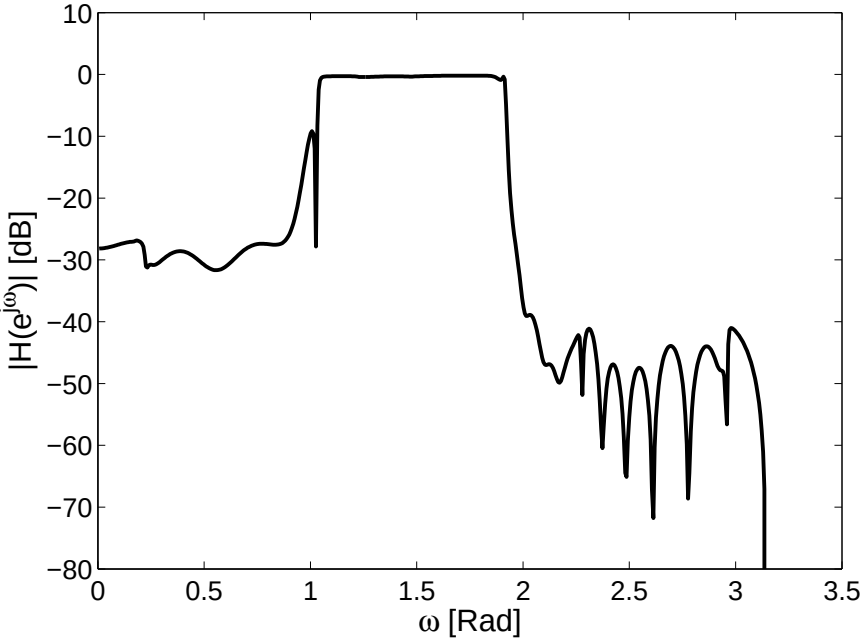

Figure 4.19: Magnitude Frequency-Response of the Overall CSD Bandpass FRM IIR Digital Filter $H(e^{j\omega})$ Before PSO

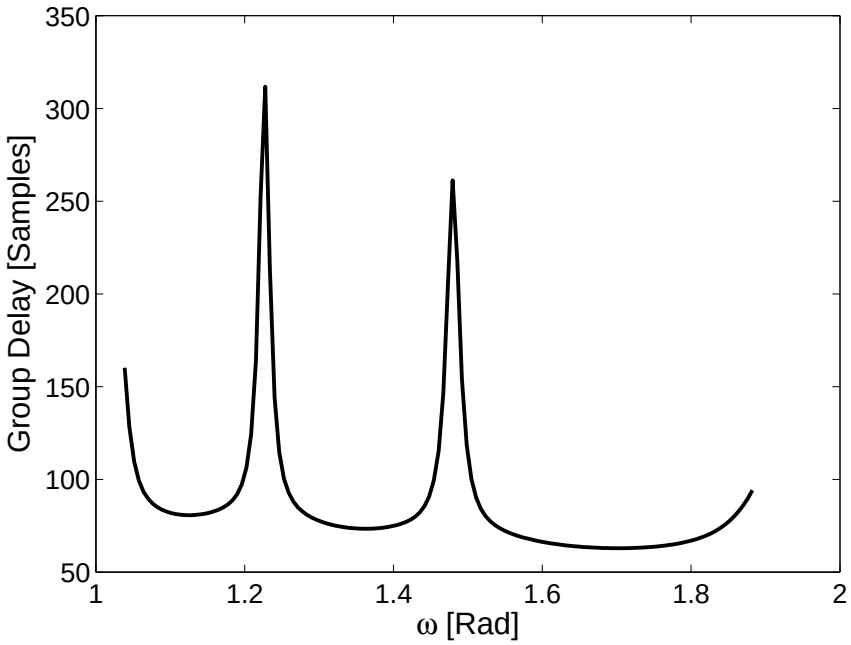

Figure 4.20: Group Delay Frequency-Response of the Overall CSD Bandpass FRM IIR Digital Filter $H(e^{j\omega})$ Before PSO

filter having a magnitude frequency response as shown in Fig. 4.21. In addition, Fig. 4.22 gives us a closer look to the magnitude frequency response of the passband region of the bandpass FRM IIR digital filter. Fig. 4.23 illustrates the group delay frequency response of the optimized bandpass FRM IIR digital filter. The values of the multiplier coefficients for the lowpass and highpass sections of the bandpass FRM IIR digital filter are obtained as summarized in Tables 4.20 and 4.21.

Table 4.22 represents the comparison of the CSD bandpass FRM IIR digital filters before and after PSO.

Since the design specifications are the same for this example and bandpass FRM FIR digital filter example in section 4.2, comparisons can be made between these two filters in terms of the number of optimization variables and the speed of convergence as summarized in Table 4.23.

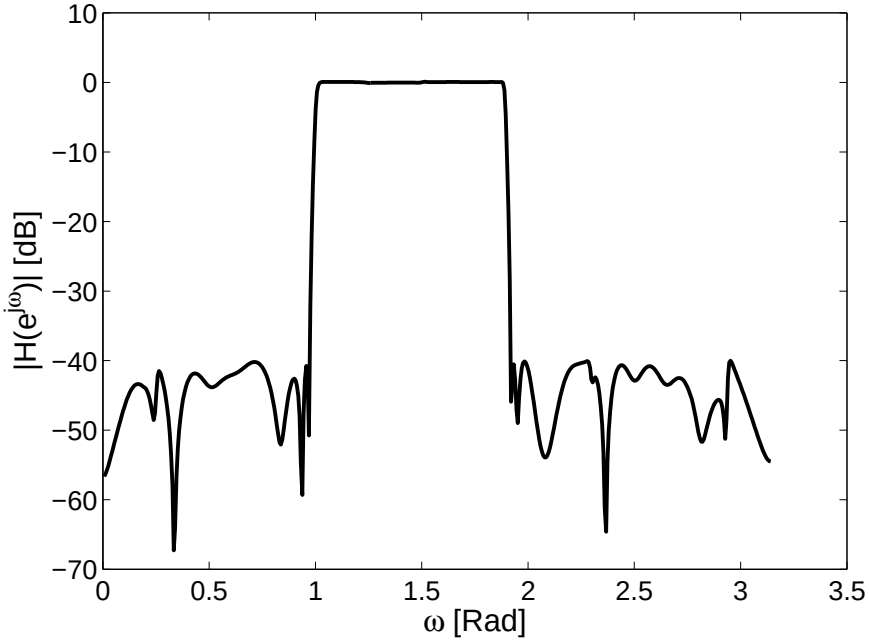

Figure 4.21: Magnitude Frequency-Response of the Overall CSD Bandpass FRM IIR Digital Filter $H(e^{j\omega})$ After PSO

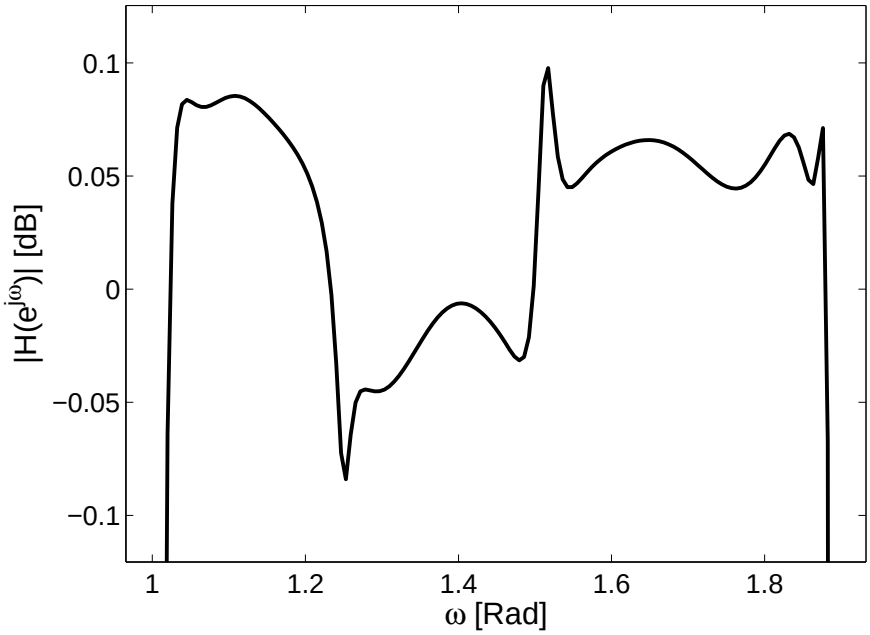

Figure 4.22: Magnitude Frequency-Response of the Passband Region of the Overall CSD Bandpass FRM IIR Digital Filter $H(e^{j\omega})$ After PSO

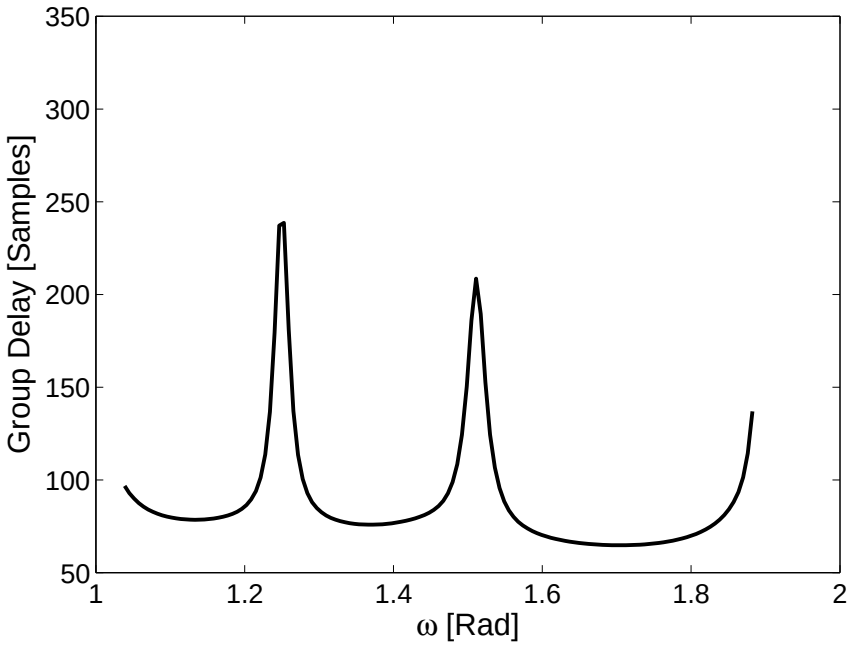

Figure 4.23: Group Delay Frequency-Response of the Overall CSD Bandpass FRM IIR Digital Filter $H(e^{j\omega})$ After PSO

Table 4.20: Digital Multiplier Values for the Lowpass Section of the Bandpass FRM IIR Digital Filter

Multiplier	CSD Representation	Decimal Value
$m_{C_{0,1}}$	$00001.000\bar{1}00\bar{1}$	0.9297
$m_{C_{0,2}}$	$00010.000\bar{1}0\bar{1}0$	1.9219
$m_{L_{0,2}}$	00000.1000010	0.5156
$m_{C_{1,1}}$	$00001.00\bar{1}00\bar{1}0$	0.8594
$m_{C_{1,2}}$	$10000.\bar{1}0\bar{1}0000$	15.375
$m_{L_{1,1}}$	$00001.000\bar{1}0\bar{1}0$	0.9219
$m_{L_{1,2}}$	$00000.0010\bar{1}0\bar{1}$	0.0859

Table 4.21: Digital Multiplier Values for the Highpass Section of the Bandpass FRM IIR Digital Filter

Multiplier	CSD Representation	Decimal Value
$m_{C_{0,1}}$	$00001.00\bar{1}0\bar{1}00$	0.8438
$m_{C_{0,2}}$	$00010.000100\bar{1}$	2.0547
$m_{L_{0,2}}$	$00000.100000\bar{1}$	0.4922
$m_{C_{1,1}}$	$00001.0\bar{1}00010$	0.7656
$m_{C_{1,2}}$	10000.0100001	16.2578
$m_{L_{1,1}}$	$00001.0000\bar{1}01$	0.9766
$m_{L_{1,2}}$	$00000.0010\bar{1}0\bar{1}$	0.0859

Table 4.22: Frequency-Response Analysis of the CSD Bandpass FRM IIR Digital Filter Before and After PSO

Frequency-Response Characteristic	Before PSO	After PSO
Maximum Passband Ripple A_p	0.8982[dB]	0.0978[dB]
Minimum Stopband Loss A_a	9.1715[dB]	40.0172[dB]
Maximum Group Delay	312[Samples]	239[Samples]

Table 4.23: Comparison between PSO of Bandpass FRM FIR Digital Filter and PSO of Bandpass FRM IIR Digital Filter

Characteristic	Bandpass FRM FIR Digital Filter	Bandpass FRM IIR Digital Filter
Number of Optimization Variables N	252	140
Average Number of Iterations	200	160

4.5 High-Level Synthesis of Digital Filters Example

In this section the proposed PSO algorithm is applied to the digital data-path synthesis of the benchmark elliptic WD filter shown in Fig. 4.24 [110]. In this figure, the input node is labeled as 1 and the output node is labeled as 42. All the outputs of the time delays in Fig. 4.24 act as an input to the system and all the inputs to the time delays act as the output of the system. Therefore, nodes 1, 8, 9, 13, 24, 31, 36, and 37 are generated in the first time-step in the DFG and nodes 7, 10, 12, 23, 35, 38, 41, and 42 are last consumed in the last time-step of the DFG.

The above high-level synthesis is performed in terms of two types of arithmetic functional units, namely two-input digital adders and digital modified booth multipliers. Moreover, the signal wordlength W is fixed at 22 bits and the coefficient wordlength is fixed at 15 bits throughout the synthesis. The maximum number of generations in the constituent PSO algorithm is fixed at 100.

The design parameters for the PSO algorithm used in the high-level synthesis of digital filters is as shown in Table 4.24

High-level synthesis by PSO algorithm leads to optimal data-paths characterized by the entries in Table 4.25.

The scheduled DFGs associated with the user-specified constraints in column 1 of Table 4.25 are as shown in Figs. 4.25, 4.27, and 4.29. In addition, The life-time table corresponding to the user-specified constraints in Table 4.25 are obtained as in

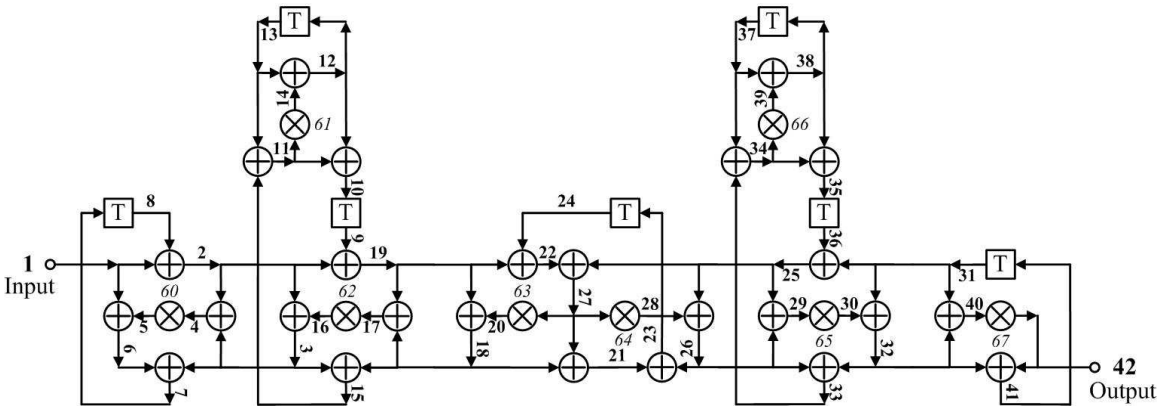

Figure 4.24: A benchmark fifth-order elliptic WD filter

Table 4.24: PSO Design Parameters for High-Level Synthesis of a Benchmark Digital Filter

K	w	c_1	c_2	$\hat{v}_{min}$	$\hat{v}_{max}$
100	0.25	2	2	-5	5

Table 4.25: Results of High-Level Synthesis with PSO

Functional Units	N_{add}	3	3	2
	N_{mult}	2	1	1
Time Steps	t_{total}	17	18	19
Multiplexors	N_{mux_2}	6	5	5
	N_{mux_3}	4	2	1
	N_{mux_4}	4	6	4
	N_{mux_5}	1	0	1
	N_{mux_6}	1	0	1
	N_{mux_7}	1	1	0
	N_{mux_8}	0	2	2
Registers	N_{reg}	13	12	12

Table 4.26: Register Allocation for Optimized Data-Path Synthesis of Benchmark WD Filter for $N_{add} = 3$ and $N_{mult} = 2$

Register #1	1	6	7			
Register #2	8	2	4	5	42	
Register #3	9	19	17	16	3	12
Register #4	13	38				
Register #5	24	22	27	21	23	
Register #6	31	14	10			
Register #7	36	25	29	30	32	35
Register #8	37					
Register #9	20	18	15	11		
Register #10	28	26	33	34		
Register #11	39					
Register #12	40					
Register #13	41					

Figs. 4.26, 4.28, and 4.30, and the register allocation for each of the user-specified constraints are obtained as shown in Tables 4.26, 4.27, and 4.28. Moreover, the average values of the cost function after 20 runs as obtained during the course of optimization for these user-specified constraints are as shown in Fig. 4.31.

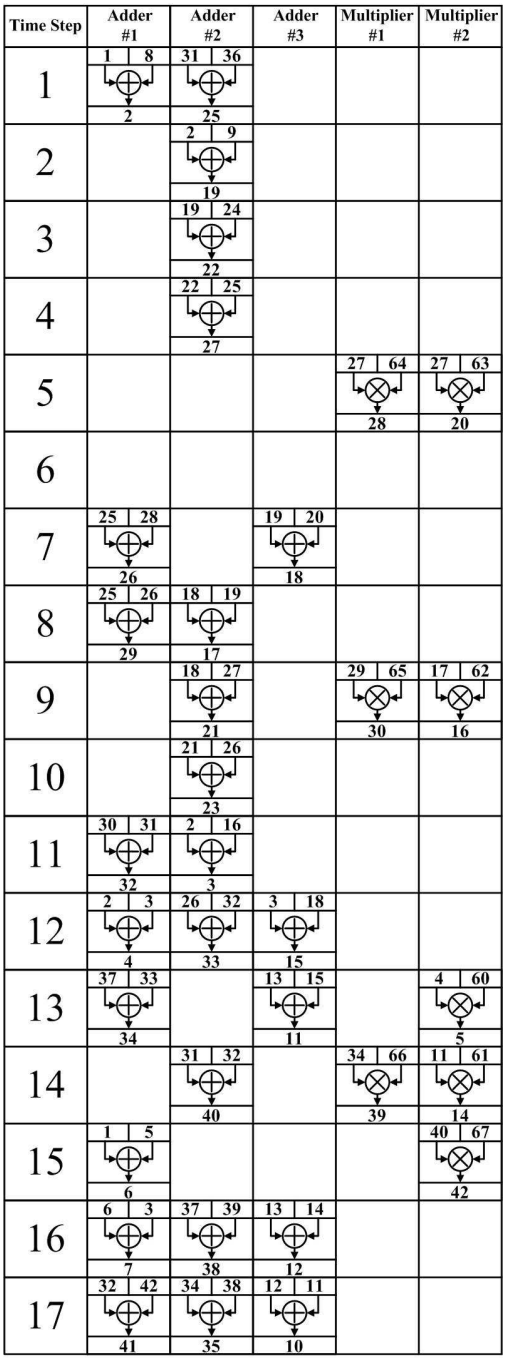

Figure 4.25: Scheduled DFG for Data-Path Synthesis of Benchmark WD Filter for $N_{add} = 3$ and $N_{mult} = 2$

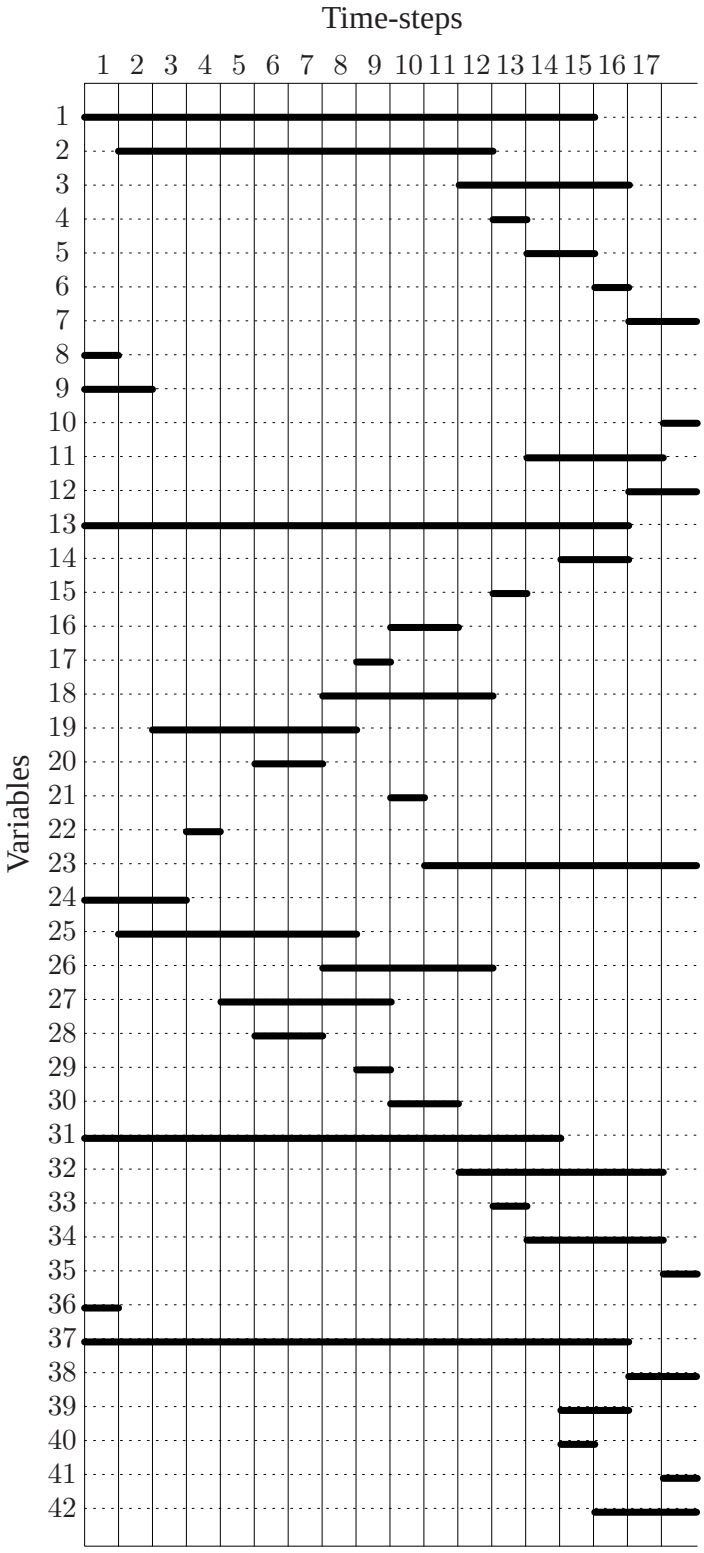

Figure 4.26: Life-Time Table for Optimized Data-Path Synthesis of Benchmark WD Filter for $N_{add} = 3$ and $N_{mult} = 2$

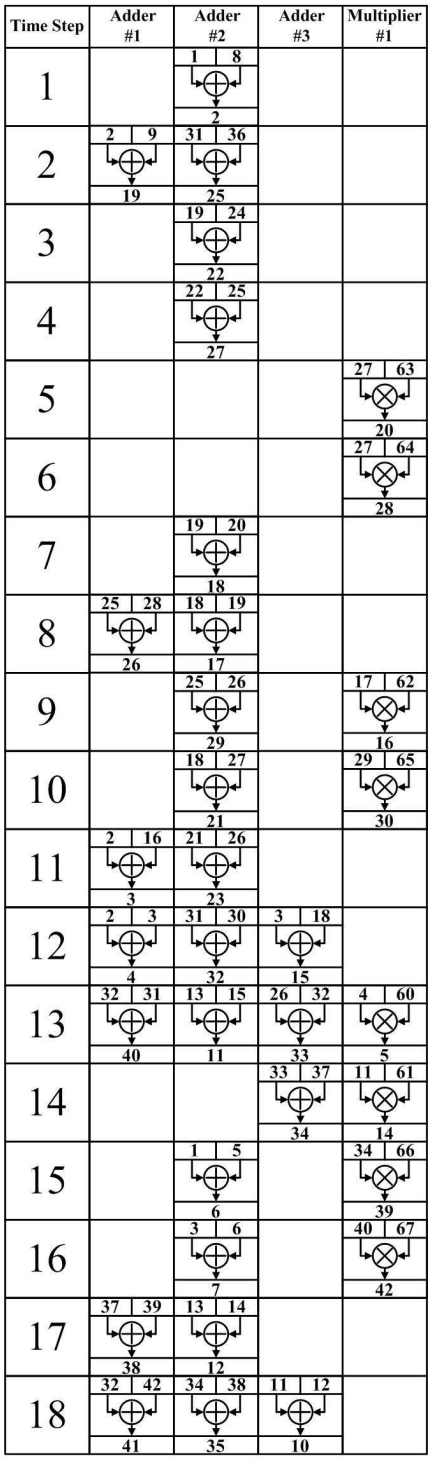

Figure 4.27: Scheduled DFG for Data-Path Synthesis of Benchmark WD Filter for $N_{add} = 3$ and $N_{mult} = 1$

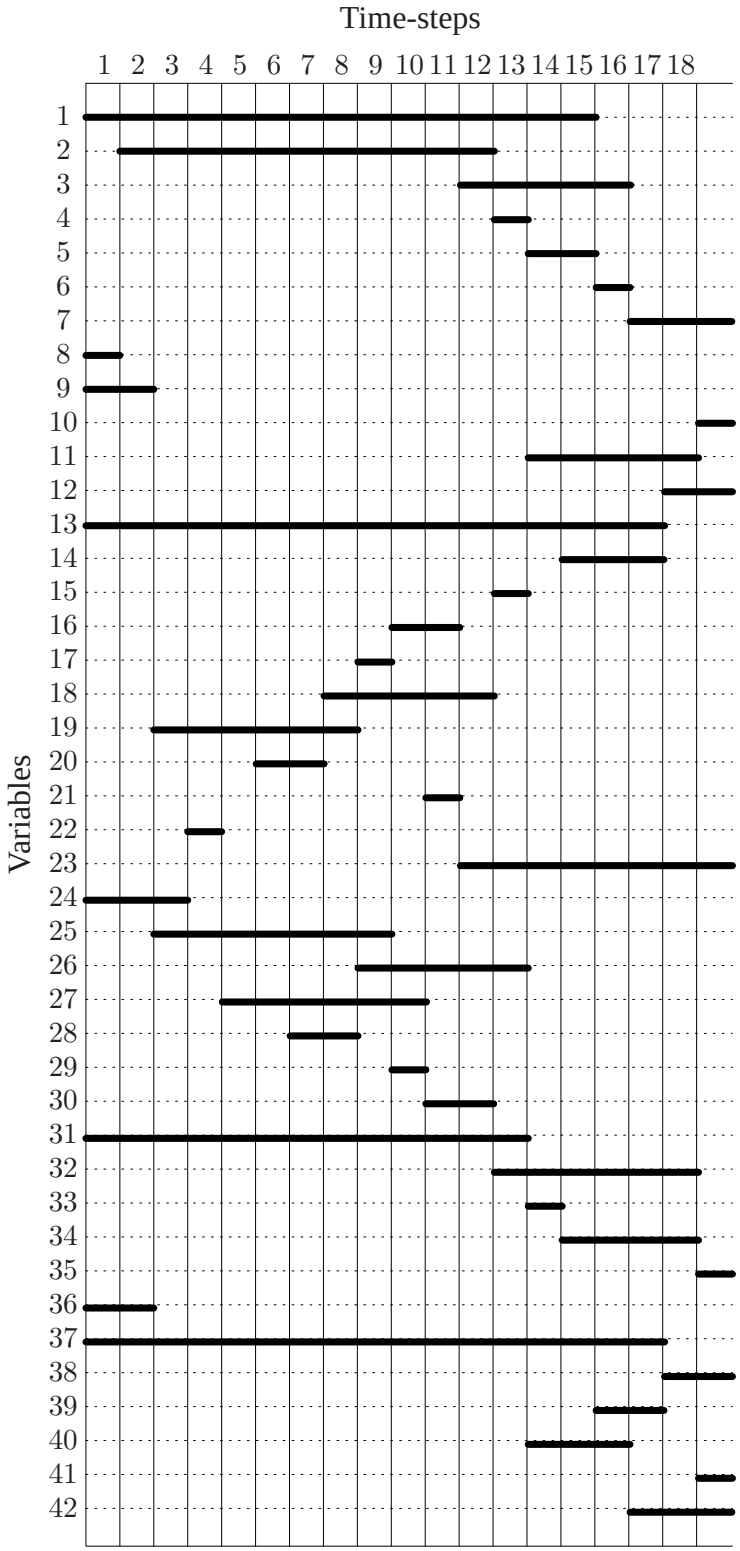

Figure 4.28: Life-Time Table for Optimized Data-Path Synthesis of Benchmark WD Filter for $N_{add} = 3$ and $N_{mult} = 1$

Table 4.27: Register Allocation for Optimized Data-Path Synthesis of Benchmark WD Filter for $N_{add} = 3$ and $N_{mult} = 1$

Register #1	1	6	7					
Register #2	8	2	4	5	39	12		
Register #3	9	19	17	16	3	42		
Register #4	13	38						
Register #5	24	22	27	21	23			
Register #6	31	11	10					
Register #7	36	25	29	30	15	33	14	35
Register #8	37							
Register #9	20	18	32					
Register #10	28	26	40					
Register #11	34							
Register #12	41							

Table 4.28: Register Allocation for Optimized Data-Path Synthesis of Benchmark WD Filter for $N_{add} = 2$ and $N_{mult} = 1$

Register #1	1	6	7			
Register #2	8	2	4	5	34	
Register #3	9	19	17	16	3	38
Register #4	13	12				
Register #5	24	22	27	21	23	
Register #6	31	14	39			
Register #7	36	25	29	30	11	10
Register #8	37					
Register #9	20	18	15	32	41	
Register #10	28	26	33	42		
Register #11	40					
Register #12	35					

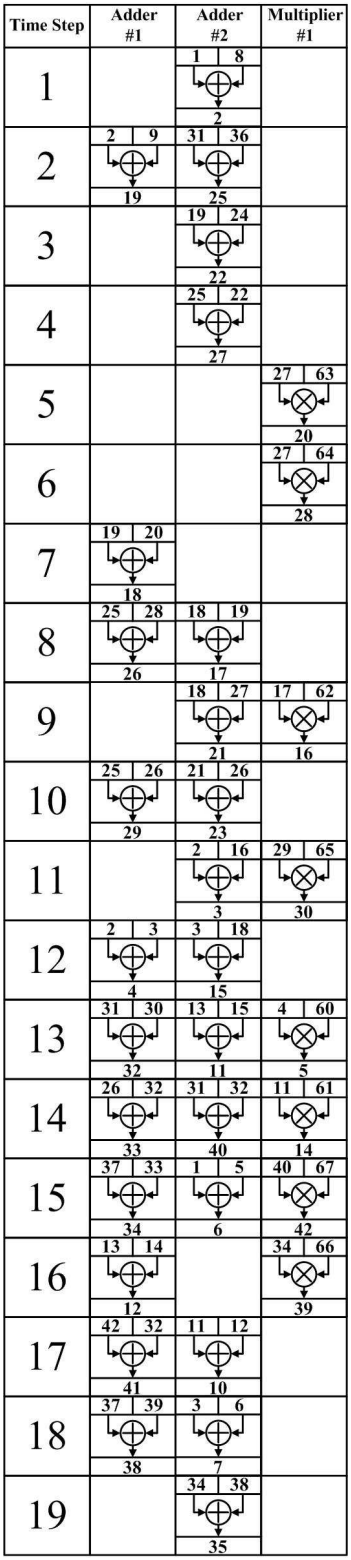

Figure 4.29: Scheduled DFG for Data-Path Synthesis of Benchmark WD Filter for $N_{add} = 2$ and $N_{mult} = 1$

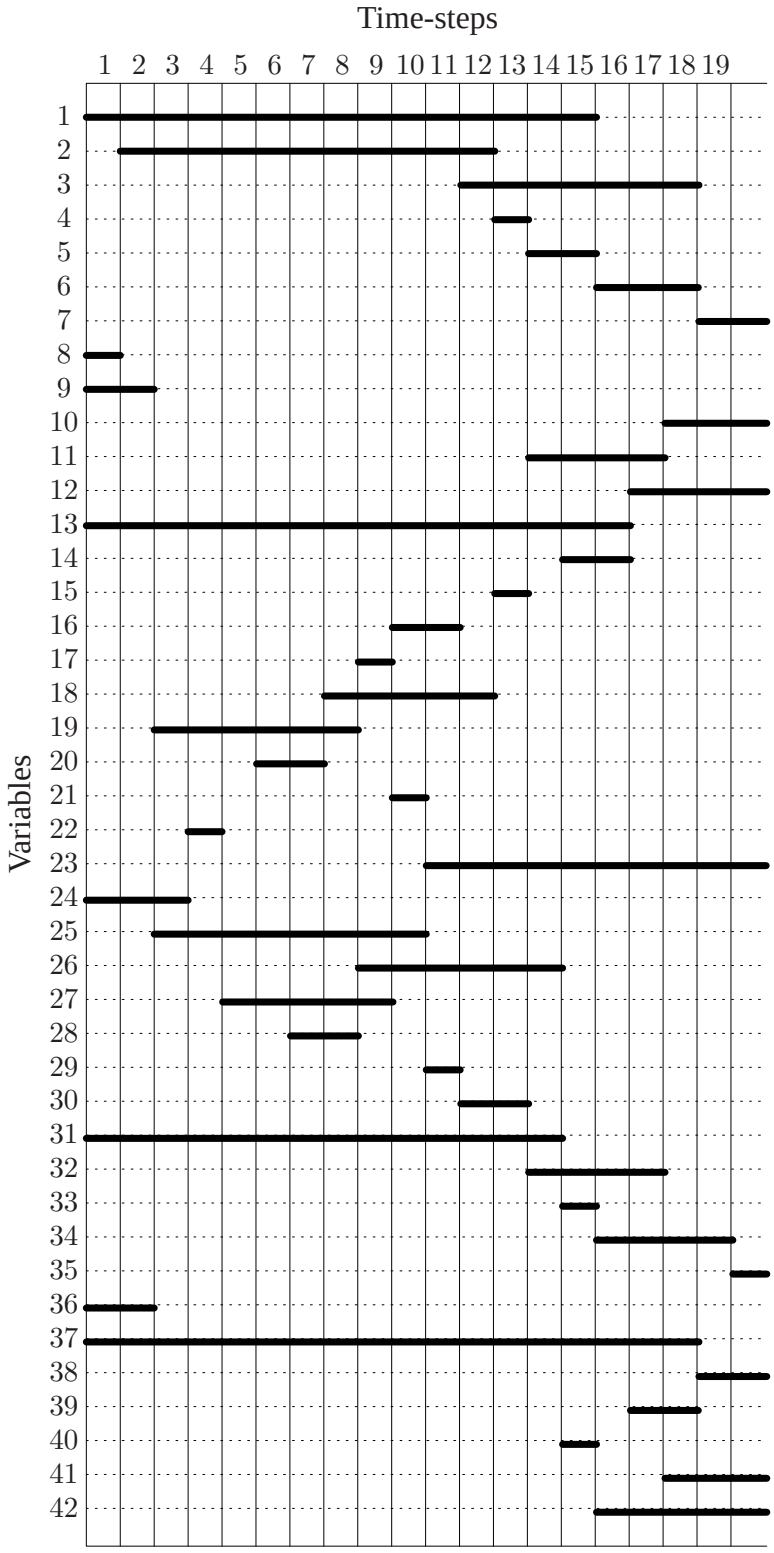

Figure 4.30: Life-Time Table for Optimized Data-Path Synthesis of Benchmark WD Filter for $N_{add} = 2$ and $N_{mult} = 1$

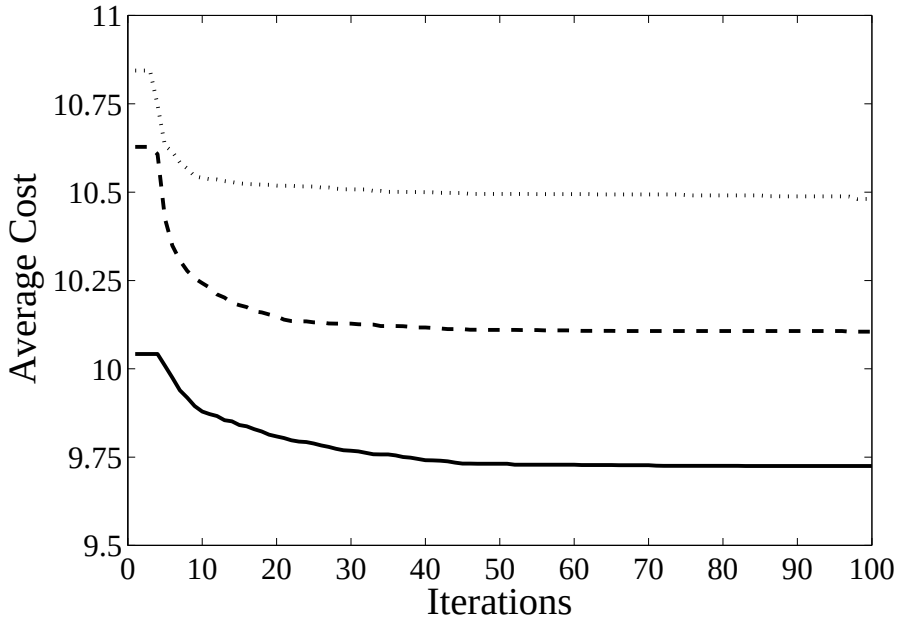

Figure 4.31: Average Value of the Cost Function Versus Number of Iterations for Different User-Specified Constraints ($N_{add} = 3$ and $N_{mult} = 2$ (solid), $N_{add} = 3$ and $N_{mult} = 1$ (dashed), $N_{add} = 2$ and $N_{mult} = 1$ (dotted))

4.6 Summary

This chapter has presented several examples to illustrate the application of the proposed PSO to the design, high-level synthesis, and optimization of digital filters. The first two examples have been concerned with the design and optimization of FRM FIR digital filters, one having a lowpass and the other having a bandpass frequency-response characteristic. The optimization has been performed over the CSD multiplier coefficient space and the results have shown that PSO can be successfully applied to the design and optimization of FRM FIR digital filters.

The second two examples have explained the application of PSO to the design and optimization of a set of two FRM IIR digital filters, one having a lowpass and the other having a bandpass frequency-response characteristic. The optimization considered both the magnitude and the group-delay frequency-responses. Similar to the first two examples, these optimizations have been performed over CSD multiplier coefficient space and the design specifications have been kept unchanged with respect to the first two examples. Therefore, comparisons could be made between

the two FRM techniques. It has shown that in case of FRM IIR digital filters, not only the number of optimization variables are reduced in comparison with their FIR counterparts, but also the speed of convergence is increased, so that one can obtain a stable FRM digital filter fast and reliably. In addition, the results for PSO of lowpass FRM IIR digital filter have been compared with the results obtained previously using DCGA, and it has been shown that PSO gives faster and more reliable solutions than DCGA.

Finally, the last example has demonstrated high-level synthesis of digital filters through the application of PSO to the high-level synthesis of a benchmark elliptic WD filter. This example concerned with the optimization of the data-path associated with the digital filter and the result has been shown to be area-cum-time-optimal. This has been achieved by optimizing the number of required support cells together with the user-specified number of arithmetic functional units, and by taking into account the cost for the hardware implementation of each of the above items. The number of time-steps in the DFG has been also minimized to obtain an optimal digital filter data-path.

Chapter 5

Conclusions

5.1 Conclusions

This thesis has been concerned with the design and optimization of a class of digital filters suitable for direct hardware realization. The digital filter should have been capable of having sharp transition bandwidth, while maintaining low complexity in term of hardware implementation. Therefore, FRM technique has been employed to achieve the previous goals. In order to further reduce the hardware complexity for the implementation of FRM digital filter, the CSD number system has been exploited for the advantage of having fewer number of nonzero bits in the representation of multiplier coefficient values constituent in the FRM digital filter.

Two approaches have been studied; FRM digital filters incorporating FIR digital filters to represent both masking digital subfilters and interpolation subfilters, and FRM digital filters incorporating FIR digital filters to represent masking digital subfilters and IIR digital filters to represent interpolation subfilters. In the latter case the hardware realization complexity is reduced due to the fewer number of coefficients present in the implementation of an IIR digital filter. To further reduce the number of coefficients, the bilinear-LDI technique has been exploited to represent digital interpolation subfilters constituent in the FRM digital filters.

A novel particle swarm optimization has been proposed for the optimization of FRM digital filters. This technique is capable of tackling three separate problems that may occur in the process of optimization of FRM digital filters. A set of LUTs is constructed and the indices of the LUTs are used to define the search space for

the PSO. Therefore, an integer-based PSO is proposed to search over the indices of the LUTs and to ensure that the resulting multiplier coefficient values constituent in the FRM digital filters conform to the CSD number system format. By successive augmentation of the LUTs with regard to the set of stability constraints, the search space is limited to FRM digital filters that are guaranteed to be BIBO stable in the course of optimization. Finally, barren layers have been introduced to guarantee that in the course of PSO, the particles remain inside the boundaries of LUTs. The cost function for PSO of FRM digital filters has been calculated to optimize both the magnitude frequency response as well as the group delay associated with the FRM digital filter.

In addition, a novel PSO has been developed for high-level synthesis of digital filters. An encoding scheme has been introduced to guarantee that the data-dependency relationships in the digital filter DFG remain satisfied under the operations of addition and subtraction in the PSO algorithm. In addition, a technique is developed to avoid any functional unit violation (that may occur because of the random nature of the operations of addition and subtraction in the underlying PSO algorithm) through the course of high-level synthesis of digital filters. The PSO is capable of optimizing the area and time constraints associated with the high-level synthesis of digital filters, by taking into account the user-specified constraints in the number of functional units available for implementation of the DFG. In addition, the number of support cells, such as registers and multiplexors, has been optimized in the course of PSO for the high-level synthesis of digital filter data-paths.

Several examples have been presented to illustrate the usefulness of the proposed techniques. PSO was applied to a set of FRM FIR digital filters, one showing a lowpass frequency response characteristic and the other having a bandpass frequency response characteristic. In addition, to show the usefulness of the PSO to the design and optimization of FRM IIR digital filters incorporating bilinear-LDI digital subfilters, a lowpass and a bandpass FRM IIR digital filter with stringent design specifications were used as examples. Moreover, PSO has been applied to the high-level synthesis of a benchmark elliptic wave digital filter and the results obtained were illustrated in an example.

5.2 Summary of Contributions

- This thesis has presented FRM FIR digital filter technique to reduce the complexity of designing a sharp transition band digital filter.
- In order to further reduce the hardware realization complexity, an IIR-based FRM digital filter incorporating interpolation digital subfilters realized using bilinear-LDI design technique has been presented.
- A step-by-step procedure for the design of FRM digital filters incorporating FIR or IIR interpolation subfilters is presented.
- A novel integer-based PSO is presented for the optimization of FRM digital filters.
- A set of CSD LUTs is constructed and modified to guarantee the BIBO stability of the resulting digital filters that may be generated throughout the PSO process.
- An indirect search method is introduced that makes the PSO to search over the indices of LUTs. In this way, the multiplier coefficient values constituent in the digital filter particle are guaranteed to conform to CSD number format.
- A novel modification to the LUTs is presented to ensure that the particles remain inside the LUTs in the course of PSO. In this modification, the LUTs are augmented in two directions with two layers that are characterized with low fitness values. These layers are called barren layers. The conditions present in the equation of PSO makes it impossible for the particles to go over the boundaries of the augmented LUTs.
- A new 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.
- A novel PSO is developed for high-level synthesis of digital filter data-paths.

- An encoding scheme is presented to encode the scheduling and allocation of the operations present in the digital filter data-path into particles, which preserves the data-dependency relationships in the digital filter data-paths.
- A new technique is developed to avoid functional unit violation that may occur during the course of PSO for high-level synthesis of digital filter data-paths.
- The cost function utilized for the high-level synthesis of digital filter data-paths allow time-optimal, area-optimal, or time-cum-area-optimal synthesis of digital filter data-paths.
- The usefulness of PSO of FRM FIR digital filters has been demonstrated.
- The usefulness of PSO of FRM IIR digital filters incorporating bilinear-LDI interpolation subfilters has been demonstrated.
- The usefulness of PSO for high-level synthesis of digital filter data-paths has been demonstrated.

5.3 Suggestions for Future Work

Future work involves the improvement of the proposed integer-based PSO algorithm. It may be reasonable to find a way for removing the rounding operations present in the proposed PSO. The LUT-based technique and the barren layers introduced in this thesis can be further processed and can be used for different applications, not only in the field of digital filter design, but also in other fields of research.

This thesis has presented high-level synthesis of digital filters using PSO. Future work in this area involves the automatic avoidance of functional unit violation in course of PSO. This can be achieved either by including an extra term in the objective function of PSO, or by adding an optimization variable to the underlying PSO. In the latter case, the speed of convergence for PSO may be reduced, while

in the first case, the speed of convergence is changed negligibly. Other encoding schemes can be employed to reduce the number of coordinates in a particle.

References

[1] D. R. Wilson, D. R. Corrall, and R. F. Mathias, "The Design and Application of Digital Filters," *IEEE Transactions on Industrial Electronics and Control Instrumentation*, vol. IECI-20, pp. 68–74, 1973.

[2] P. P. Vaidyanathan, "Multirate digital filters, filter banks, polyphase networks, and applications: a tutorial," *Proceedings of the IEEE*, vol. 78, pp. 56–93, 1990.

[3] Y. C. 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.

[4] ——, "A Digital Filter Bank for Digital Audio Systems," *IEEE Transactions on Circuits and Systems*, vol. 33-8, p. 848 849, Aug. 1986.

[5] Y. C. Lim, S. R. Parker, and A. G. Constantinides, "Finite Word Length FIR Filter Design Using Integer Programming over a Discrete Coefficient Space," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 30, pp. 661–664, 1982.

[6] T. Saramaki and Y. C. Lim, "Use of the Remez Algorithm for Designing FIR Filters Utilizing the Frequency-Response Masking Approach," in *1999 IEEE International Symposium on Circuits and Systems. ISCAS '99*, vol. 3, 1999, pp. 449–455.

[7] Y. J. Yu and Y. C. Lim, "FRM Based FIR Filter Design - the WLS Approach," in *2002 IEEE International Symposium on Circuits and Systems. ISCAS 2002*, vol. 3, 2002, pp. III–221 – III–224.

[8] W.-S. Lu and T. Hinamoto, "Optimal Design of Frequency-Response-Masking Filters Using Semidefinite Programming," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 50, pp. 557–568, 2003.

[9] ——, "Optimal Design of FIR Frequency-Response-Masking Filters Using Second-Order Cone Programming," in *Proceedings of 2003 IEEE International Symposium on Circuits and Systems. ISCAS '03*, vol. 3, 2003, pp. III–878 – III–881.

[10] 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.

[11] S. Chen, R. H. Istepanian, and B. L. Luk, "Digital IIR Filter Design Using Adaptive Simulated Annealing," *Digital Signal Processing*, vol. 11, no. 3, pp. 241–251, July 2001.

[12] K. sang Tang, K. fung Man, S. Kwong, and Z. feng Liu, "Design and Optimization of IIR Filter Structure Using Hierarchical Genetic Algorithms," *IEEE Transactions on Industrial Electronics*, vol. 45, no. 3, pp. 481–487, 1998.

[13] C. Dai, W. Chen, and Y. Zhu, "Seeker Optimization Algorithm for Digital IIR Filter Design," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 5, pp. 1710–1718, May 2010.

[14] M. Najjarzadeh and A. Ayatollahi, "FIR Digital Filters Design: Particle Swarm Optimization Utilizing LMS and Minimax Strategies," in *IEEE International Symposium on Signal Processing and Information Technology, 2008. ISSPIT 2008.*, 2008, pp. 129–132.

[15] C. Dai, W. Chen, Y. Zhu, and X. Zhang, "Seeker Optimization Algorithm for Optimal Reactive Power Dispatch," *IEEE Transactions on Power Systems*, vol. 24, no. 3, pp. 1218–1231, August 2009.

[16] A. Kalinli and N. Karaboga, "A New Method for Adaptive IIR Filter Design Based on Tabu Search Algorithm," *AEU - International Journal of Electronics and Communications*, vol. 59, no. 3, pp. 111–117, May 2005.

[17] N. Karaboga, A. Kalinli, and D. Karaboga, "Designing Digital IIR Filters Using Ant Colony Optimisation Algorithm," *Engineering Applications of Artificial Intelligence*, vol. 17, no. 3, pp. 301–309, April 2004.

[18] A. Kalinli and N. Karaboga, "Artificial Immune Algorithm for IIR Filter Design," *Engineering Applications of Artificial Intelligence*, vol. 18, no. 5, pp. 919–929, Dec 2005.

[19] R. Storn, "Designing Nonstandard Filters with Differential Evolution," *IEEE Signal Processing Magazine*, vol. 22, no. 1, pp. 103–106, Jan 2005.

[20] N. Karaboga, "Digital IIR Filter Design Using Differential Evolution Algorithm," *EURASIP Journal on Applied Signal Processing*, vol. 2005, no. 8, pp. 1269–1276, Jan 2005.

[21] P. Mercier, S. M. Kilambi, and B. Nowrouzian, "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.

[22] S. Bokhari and B. Nowrouzian, "DCGA Optimization of Lowpass FRM IIR Digital Filters Over CSD Multiplier Coefficient Space," in *52nd IEEE International Midwest Symposium on Circuits and Systems*, August 2009, pp. 573–576.

[23] S. Bokhari, B. Nowrouzian, and S. A. Hashemi, "A novel technique for DCGA optimization of guaranteed BIBO stable IIR-based FRM digital filters over the CSD multiplier coefficient space," in *proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2010, pp. 2710–2713.

[24] S. A. Hashemi and B. Nowrouzian, "Particle swarm optimization of FRM FIR digital filters over the CSD multiplier coefficient space," in *proceedings of 53rd IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)*, 2010, pp. 1246–1249.

[25] ——, “A novel discrete particle swarm optimization for frm fir digital filters,” *Journal of Computers, ”Submitted”*.

[26] ——, “Discrete particle swarm optimization of magnitude response of iir-based frm digital filters,” in *proceedings of 17th IEEE International Conference on Electronics, Circuits, and Systems, 2010. ICECS 2010.*, December 2010.

[27] ——, “A novel finite-wordlength particle swarm optimization technique for frm iir digital filters,” in *proceedings of 2011 IEEE International Symposium on Circuits and Systems (ISCAS), ”To appear”*, May 2011.

[28] J. Kennedy and R. Eberhart, “Particle Swarm Optimization,” in *Proceedings of IEEE International Conference on Neural Networks*, vol. 4, 1995, pp. 1942–1948.

[29] M. D. Lutovac and L. D. Milić, “IIR Filters Based on Frequency-Response Masking Approach,” in *Telecommunications in Modern Satellite, Cable and Broadcasting Service, TELSIKS 2001*, Sept. 2001, pp. 163–170.

[30] 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.

[31] J. Sun, W. Fang, and W. Xu, “A Quantum-Behaved Particle Swarm Optimization With Diversity-Guided Mutation for the Design of Two-Dimensional IIR Digital Filters,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 57, no. 2, pp. 141–145, 2010.

[32] A. Slowik and M. Bialko, “Design and Optimization of IIR Digital Filters with Non-Standard Characteristics Using Particle Swarm Optimization Algorithm,” in *14th IEEE International Conference on Electronics, Circuits and Systems, ICECS 2007*, 2007, pp. 162–165.

[33] B. Luitel and G. K. Venayagamoorthy, “Particle Swarm Optimization with Quantum Infusion for the design of digital filters,” in *IEEE Swarm Intelligence Symposium, SIS 2008*, 2008, pp. 1–8.

[34] B. Nowrouzian and L. S. Lee, “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.

[35] T. Parks and J. McClellan, “Chebyshev Approximation for Nonrecursive Digital Filters with Linear Phase,” *IEEE Transactions on Circuit Theory*, vol. CT-19, pp. 189–194, 1972.

[36] Y. C. Lim, R. Yang, D. Li, and J. Song, “Signed Power-of-Two Term Allocation Scheme for the Design of Digital Filters,” *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 46, pp. 577–584, 1999.

[37] R. I. Hartley, “Subexpression Sharing in Filters Using Canonic Signed Digit Multipliers,” *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 43, pp. 677–688, 1996.

[38] A. T. G. Fuller, B. Nowrouzian, and F. Ashrafzadeh, "Optimization of FIR Digital Filters over the Canonical Signed-Digit Coefficient Space Using Genetic Algorithms," in *1998 Midwest Symposium on Circuits and Systems*, 1998, pp. 456–459.

[39] M. C. McFarland, A. C. Parker, and R. Camposano, "Tutorial on high-level synthesis," in *Proceedings of the 25th ACM/IEEE Design Automation Conference*, ser. DAC '88, 1988, pp. 330–336.

[40] N. Park and A. Parker, "Sehwa: a software package for synthesis of pipelines from behavioral specifications," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 7, no. 3, pp. 356 –370, March 1988.

[41] J.-H. Lee, Y.-C. Hsu, and Y.-L. Lin, "A new integer linear programming formulation for the scheduling problem in data path synthesis," in *Proceedings of 1989 IEEE International Conference on Computer-Aided Design, 1989. ICCAD-89. Digest of Technical Papers.*, Nov 1989, pp. 20 –23.

[42] S. Devadas and A. Newton, "Algorithms for hardware allocation in data path synthesis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 8, no. 7, pp. 768 –781, July 1989.

[43] A. C. Parker, J. T. Pizarro, and M. Mlinar, "Maha: a program for datapath synthesis," in *Proceedings of the 23rd ACM/IEEE Design Automation Conference*, ser. DAC '86, 1986, pp. 461–466.

[44] H. Trickey, "Flamel: A high-level hardware compiler," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 6, no. 2, pp. 259 – 269, March 1987.

[45] P. Marwedel, "A new synthesis algorithm for the mimola software system," in *Proceedings of the 23rd Conference on Design Automation, 1986.*, June 1986, pp. 271 – 277.

[46] P. G. Paulin and J. P. Knight, "Force-directed scheduling for the behavioral synthesis of asics," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 8, no. 6, pp. 661 –679, June 1989.

[47] ——, "Force-directed scheduling in automatic data path synthesis," in *Proceedings of the 24th ACM/IEEE Design Automation Conference*, ser. DAC '87, 1987, pp. 195–202.

[48] R. Potasman, J. Lis, A. Nicolau, and D. Gajski, "Percolation based synthesis," in *Proceedings of the 27th ACM/IEEE Design Automation Conference, 1990.*, June 1990, pp. 444 –449.

[49] I.-C. Park and C.-M. Kyung, "Famos: an efficient scheduling algorithm for high-level synthesis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 12, no. 10, pp. 1437 –1448, Oct. 1993.

[50] K. Ito, T. Iwata, and H. Kunieda, "An optimal scheduling method for parallel processing system of array architecture," in *Proceedings of the ASP-DAC '97. Asia and South Pacific Design Automation Conference 1997.*, Jan. 1997, pp. 447 –454.

[51] P. Arató, Z. Á. Mann, and A. Orbán, "Time-constrained scheduling of large pipelined datapaths," *Journal of Systems Architecture*, vol. 51, no. 12, pp. 665 – 687, 2005.

[52] S. O. Memik, R. Kastner, E. Bozorgzadeh, and M. Sarrafzadeh, "A scheduling algorithm for optimization and early planning in high-level synthesis," *ACM Transactions on Design Automation of Electronic Systems*, vol. 10, pp. 33–57, Jan. 2005.

[53] F. Kurdahi and A. Parker, "Real: A program for register allocation," in *Proceedings of the 24th Conference on Design Automation, 1987.*, June 1987, pp. 210 – 215.

[54] B. Haroun and M. Elmasry, "Architectural synthesis for dsp silicon compilers," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 8, no. 4, pp. 431 –447, Apr. 1989.

[55] S. J. Beaty, "Genetic algorithms and instruction scheduling," in *Proceedings of the 24th annual international symposium on Microarchitecture*, ser. MICRO 24, 1991, pp. 206–211.

[56] E. Bonsma and S. H. Gerez, "A genetic approach to the overlapped scheduling of iterative data-flow graphs for target architectures with communication delays," in *Proceedings of the ProRISC Workshop on Circuits, Systems and Signal Processing*, 1997.

[57] M. Heijligers and J. Jess, "High-level synthesis scheduling and allocation using genetic algorithms based on constructive topological scheduling techniques," in *Proceedings of the IEEE International Conference on Evolutionary Computation, 1995.*, vol. 1, Dec. 1995, p. 56.

[58] J. H. Satyanarayana and B. Nowrouzian, "A new technique for the high-level synthesis of digit-serial digital filters based on genetic algorithms," *Journal of Circuits, Systems, and Computers (JCSC)*, vol. 7, no. 6, pp. 517 –535, 1997.

[59] R. F. Abdel-Kader, "Particle swarm optimization for constrained instruction scheduling," *VLSI Des.*, vol. 2008, pp. 7:1–7:7, February 2008.

[60] S.-C. Chu, Y.-T. Chen, and J.-H. Ho, "Timetable scheduling using particle swarm optimization," in *Proceedings of First International Conference on Innovative Computing, Information and Control, 2006. ICICIC '06.*, vol. 3, Sept. 2006, pp. 324 –327.

[61] X. Kong, J. Sun, and W. Xu, "Particle swarm algorithm for tasks scheduling in distributed heterogeneous system," in *Proceedings of Sixth International Conference on Intelligent Systems Design and Applications, 2006. ISDA '06.*, vol. 2, Oct. 2006, pp. 690 –695.

[62] M. Clerc and J. Kennedy, "The particle swarm - explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58 –73, Feb 2002.

[63] I. C. Trelea, "The particle swarm optimization algorithm: convergence analysis and parameter selection," *Inf. Process. Lett.*, vol. 85, pp. 317–325, March 2003.

[64] V. Kadirkamanathan, K. Selvarajah, and P. Fleming, "Stability analysis of the particle dynamics in particle swarm optimizer," *IEEE Transactions on Evolutionary Computation*, 2006.

[65] F. van den Bergh and A. P. Engelbrecht, "A study of particle swarm optimization particle trajectories," *Inf. Sci.*, vol. 176, pp. 937–971, April 2006.

[66] R. C. Eberhart and Y. Shi, "Particle swarm optimization: developments, applications and resources," in *Proceedings of the 2001 Congress on Evolutionary Computation, 2001.*, 2001.

[67] X. Li and A. P. Engelbrecht, "Particle swarm optimization: an introduction and its recent developments," in *Proceedings of the 2007 GECCO conference companion on Genetic and evolutionary computation*, ser. GECCO '07, 2007, pp. 3391–3414.

[68] Y. Shi and R. C. Eberhart, "A modified particle swarm optimizer," in *Proceedings of the 1998 IEEE International Conference on Evolutionary Computation, 1998. IEEE World Congress on Computational Intelligence.*, May 1998, pp. 69 –73.

[69] ——, "Fuzzy adaptive particle swarm optimization," in *Proceedings of the 2001 Congress on Evolutionary Computation, 2001.*, 2001.

[70] R. C. Eberhart and Y. Shi, "Tracking and optimizing dynamic systems with particle swarms," in *Proceedings of the 2001 Congress on Evolutionary Computation, 2001.*, 2001.

[71] M. Clerc, "The swarm and the queen: towards a deterministic and adaptive particle swarm optimization," in *Proceedings of the 1999 Congress on Evolutionary Computation, 1999. CEC 99.*, 1999.

[72] R. C. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization," in *Proceedings of the 2000 Congress on Evolutionary Computation, 2000.*, 2000.

[73] J. Kennedy, "The particle swarm: social adaptation of knowledge," in *Proceedings of IEEE International Conference on Evolutionary Computation, 1997.*, Apr. 1997, pp. 303 –308.

[74] P. Suganthan, "Particle swarm optimiser with neighbourhood operator," in *Proceedings of the 1999 Congress on Evolutionary Computation, 1999. CEC 99.*, 1999.

[75] A. Ratnaweera, S. Halgamuge, and H. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 240 – 255, June 2004.

[76] P. Angeline, "Using selection to improve particle swarm optimization," in *Proceedings of the 1998 IEEE International Conference on Evolutionary Computation, 1998. IEEE World Congress on Computational Intelligence.*, May 1998, pp. 84 –89.

[77] C.-F. Juang, "A hybrid of genetic algorithm and particle swarm optimization for recurrent network design," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics,*, vol. 34, no. 2, pp. 997 – 1006, April 2004.

[78] Y.-P. Chen, W.-C. Peng, and M.-C. Jian, "Particle swarm optimization with recombination and dynamic linkage discovery," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics,*, vol. 37, no. 6, pp. 1460 –1470, Dec. 2007.

[79] P. Andrews, "An investigation into mutation operators for particle swarm optimization," in *Proceedings of IEEE Congress on Evolutionary Computation, 2006. CEC 2006.*, 2006.

[80] M. Løvbjerg, T. K. Rasmussen, and T. Krink, "Hybrid particle swarm optimiser with breeding and subpopulations," 2001.

[81] J. Liang and P. Suganthan, "Dynamic multi-swarm particle swarm optimizer with local search," in *Proceedings of the 2005 IEEE Congress on Evolutionary Computation, 2005.*, vol. 1, Sept. 2005, pp. 522 –528 Vol.1.

[82] W.-J. Zhang and X.-F. Xie, "Depso: hybrid particle swarm with differential evolution operator," in *Proceedings of IEEE International Conference on Systems, Man and Cybernetics, 2003.*, vol. 4, Oct. 2003, pp. 3816 – 3821 vol.4.

[83] F. van den Bergh and A. P. Engelbrecht, "A Cooperative Approach to Particle Swarm Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 225–239, 2004.

[84] K. Parsopoulos and M. Vrahatis, "On the computation of all global minimizers through particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 211 – 224, June 2004.

[85] R. Brits, A. P. Engelbrecht, and F. V. D. Bergh, "A niching particle swarm optimizer," in *Proceedings of the Conference on Simulated Evolution And Learning*, 2002, pp. 692–696.

[86] D. Parrott and X. Li, "Locating and tracking multiple dynamic optima by a particle swarm model using speciation," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 4, pp. 440 –458, Aug. 2006.

[87] J. Kennedy and R. Mendes, "Population structure and particle swarm performance," in *Proceedings of the 2002 Congress on Evolutionary Computation, 2002. CEC '02.*, 2002.

[88] ——, "Neighborhood topologies in fully-informed and best-of-neighborhood particle swarms," in *Proceedings of the 2003 IEEE International Workshop on Soft Computing in Industrial Applications, 2003. SMCia/03.*, June 2003, pp. 45 – 50.

[89] X. Hu and R. Eberhart, "Multiobjective optimization using dynamic neighborhood particle swarm optimization," in *Proceedings of the 2002 Congress on Evolutionary Computation, 2002. CEC '02.*, 2002.

[90] J. Liang and P. Suganthan, "Dynamic multi-swarm particle swarm optimizer," in *Proceedings of 2005 IEEE Swarm Intelligence Symposium, 2005. SIS 2005.*, June 2005, pp. 124 – 129.

[91] R. Mendes, J. Kennedy, and J. Neves, "The fully informed particle swarm: simpler, maybe better," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 204 – 210, June 2004.

[92] J. Liang, A. Qin, P. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 281 – 295, June 2006.

[93] R. Yang, Y. C. Lim, and S. R. Parker, "Design of sharp linear-phase FIR bandstop filters using the frequency-response-masking technique," *Circuits, Systems, and Signal Processing*, vol. 17, no. 1, pp. 1–27, Jan. 1998.

[94] 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.

[95] 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.

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

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

[98] 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.

[99] R. Hashemian, "A new method for conversion of a 2's complement to canonic signed digit number system and its representation," in *proceedings of 1996 Conference Record of the Thirtieth Asilomar Conference on Signals, Systems and Computers.*, November 1996, pp. 904 –907 vol.2.

[100] S. B. Bokhari, "Design and Discrete Optimization of BIBO Stable FRM Digital Filters Incorporating IIR Digital Interpolation Subfilters," Master's thesis, University of Alberta, Dep. of Elec. and Comp. Eng., University of Alberta, Edmonton, AB, Canada, 2010.

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

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

[103] R. E. Crochiere, "Digital Network Theory and its Application to the Analysis and Design of Digital Filters," Ph.D. dissertation, M.I.T, Dep. of Elec. Eng., M.I.T, Cambridge, MA, May 1974.

[104] D. D. Gajski and L. Ramachandran, "Introduction to high-level synthesis," *IEEE Design Test of Computers*, vol. 11, no. 4, pp. 44 –54, 1994.

[105] A. Shahid, M. S. T. Benten, and S. M. Sait, "Gsa: scheduling and allocation using genetic algorithm," in *Proceedings of the conference on European design automation*, ser. EURO-DAC '94, 1994, pp. 84–89.

[106] A. Hashimoto and J. Stevens, "Wire routing by optimizing channel assignment within large apertures," in *Proceedings of the 8th Design Automation Workshop*, ser. DAC '71, 1971, pp. 155–169.

[107] J. H. Satyanarayana and B. Nowrouzian, "Design and fpga implementation of digit-serial modified booth multipliers," *Journal of Circuits, Systems, and Computers (JCSC)*, vol. 6, no. 5, pp. 485 –501, 1996.

[108] ——, "A comprehensive approach to the design of digit-serial modified booth multipliers," in *Proceedings of the 26th Southeastern Symposium on System Theory, 1994.*, March 1994, pp. 229 –233.

[109] F. Marquez-Stricker, Y. Wu, and B. Nowrouzian, "A Novel Technique for the Design and DCGA Optimization of Guaranteed BIBO Stable Jaumann Digital IF Filters over the CSD Multiplier Coefficient Space," in *proceedings of 2009 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2009, pp. 473–476.

[110] S. Y. Kung, T. Kailath, and H. J. Whitehouse, *VLSI and Modern Signal Processing*. Prentice Hall Professional Technical Reference, 1984.

Appendix A

Author's Contributions

Refereed Journal Publications

- **Seyyed Ali Hashemi**, and Behrouz Nowrouzian, "A novel discrete particle swarm optimization for FRM FIR digital filters", Submitted to *Journal of Computers*.

Refereed Conference Publications

- **Seyyed Ali Hashemi**, and Behrouz Nowrouzian, "A novel finite-wordlength particle swarm optimization technique for FRM IIR digital filters", To appear in *proceedings of 2011 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 15th - 18th, 2011, Rio de Janeiro, Brazil.
- **Seyyed Ali Hashemi**, and Behrouz Nowrouzian, "Discrete particle swarm optimization of magnitude response of IIR-based FRM digital filters", In *proceedings of 17th IEEE International Conference on Electronics, Circuits, and Systems (ICECS) 2010*, December 12th - 15th, 2010, Athens, Greece.
- **Seyyed Ali Hashemi**, and Behrouz Nowrouzian, "Particle swarm optimization of FRM FIR digital filters over the CSD multiplier coefficient space", In *2010 53rd IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)*, August 1st - 4th, 2010, Seattle, Washington, US.
- Syed Bokhari, Behrouz Nowrouzian, and **Seyyed Ali Hashemi**, "A novel technique for DCGA optimization of guaranteed BIBO stable IIR-based FRM

digital filters over the CSD multiplier coefficient space", In *proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 30th - June 2nd, 2010, Paris, France.