IIR FILTER DESIGN USING MULTI OBJECTIVE PARTICLE SWARM OPTIMIZATION

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF DEGREE OF

Master of Technology

in

Information Systems

By

ANKITA

(2K11/ISY/01)

Under the Guidance of

Dr. O. P. Verma

(HOD, IT Department)

DEPARTMENT OF INFORMATION TECHNOLOGY
DELHI TECHNOLOGICAL UNIVERSITY
DELHI

2011-2013

CERTIFICATE

This is to certify that the thesis entitled, “IIR FILTER DESIGN USING MULTI OBJECTIVE PARTICLE SWARM OPTIMIZATION” submitted by Ankita (2K11/ISY/01) in partial fulfillment of the requirements for the award of Master of Technology Degree in Information Systems during session 2011-2013 at Delhi Technological University is an authentic work carried out by her under my supervision and guidance.

To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other University / Institute for the award of any Degree or Diploma.

Dr. O. P. Verma
Head of Department
Department of Information Technology
Delhi Technological University
Delhi

ACKNOWLEDGEMENT

I take the opportunity to express my sincere gratitude to my project mentor Dr. O.P. Verma, Head of Department, Department of Information Technology, Delhi Technological University, Delhi for providing valuable guidance and constant encouragement throughout the project .It is my pleasure to record my sincere thanks to him for his constructive criticism and insight without which the project would not have shaped as it has.

I humbly extend my words of gratitude to other faculty members of this department for providing their valuable help and time whenever it was required.

Ankita

Roll No. 2K11/ISY/01

M.Tech (Information Systems)

E-mail: ankitawadhwa89@yahoo.in

Abstract

IIR filter design is considered very important and difficult task in digital signal processing. One of the disadvantages of IIR filters is their non linear phase characteristics. Evolutionary algorithms are introduced in recent past into IIR filter design methods. IIR filter design requires concurrent minimization of order of filter, magnitude response error and linear phase response error. The proposed method designs an IIR filter with minimum order, linear phase and minimum magnitude response error. The algorithm finds the coefficients of the transfer function of desired filter. The filter is designed using Pareto based multi-objective optimization approach minimizing three objective functions simultaneously. In this thesis Multi Objective problem is solved using Pareto based Multi Objective Particle Swarm Optimization (MOPSO). Pareto based algorithms produces a set of non-dominated solutions called Pareto optimal set in one run of algorithm. It is left to the decision maker to select one solution from Pareto optimal set based on application of the filter. In literature, only Genetic based algorithms have been used in IIR filter design. PSO results in a better convergence rate. Results of proposed approach are compared with conventional approaches. Different filter types namely Low pass, High pass, Band pass and Band stop are designed. Performance of the approach depend upon some factors like number of iterations, size of repository, population size and other parameters of swarm. Experimental results show that proposed approach result in better magnitude response and have more linear phase than other approaches.

List of figures

Fig 2.1: Block diagram of a filter 5

Fig 2.2: Block diagram of FIR and IIR filters 6

Fig 2.3: Ideal low pass filter 8

Fig 2.4: Practical low pass filter 8

Fig 2.5: Ideal high pass filter 9

Fig 2.6: Practical high pass filter 9

Fig 2.7: Ideal bandpass filter 10

Fig 2.8: Practical bandpass filter 10

Fig 2.9: Ideal bandstop filter 11

Fig 2.10: practical bandstop filter 11

Fig 2.11: A stable IIR filter 12

Fig 2.12: An unstable IIR filter 12

Fig 2.13: 7^{th} and 12^{th} order Butterworth filters 15

Fig 3.1: Pareto front of MOP with 2 objective functions: Cost and efficiency 21

Fig 3.2: mapping from decision variable space to objective function space 22

Fig 3.3: Possible cases for the archive controller 28

Fig 3.4: insertion of element in the grid when it lies within the current boundaries 29

Fig 3.5: insertion of element in the grid when it lies outside the current boundaries 30

Fig 5.1: Butterworth filters a) Low Pass b) High pass c) Band pass d) Band stop 38

Fig 5.2: Chebyshev I filters a) Low Pass b) High pass c) Band pass d) Band stop 39

Fig 5.3: Chebyshev II filters a) Low Pass b) High pass c) Band pass d) Band stop 40

Fig 5.4: Elliptical filters a) Low Pass b) High pass c) Band pass d) Band stop 41

Fig 5.5: a) Magnitude response for LP b) Phase response for LP 42

c) Magnitude response for HP d) Phase response for HP 42

Fig 5.6: a) Magnitude response for BP b) Phase response for BP 43

d) Magnitude response for BS d) Phase response for BS 43

Fig 5.6: Comparison of different methods for low pass filter design 44

Fig 5.7: Comparison of different methods for high pass filter design 45

Fig 5.8: Comparison of different methods for band pass filter design 45

Fig 5.9: Comparison of different methods for band stop filter design.. 46

List of Tables

Table 2.1: IIR vs FIR filters 17

Table 4.1: Design Criteria 34

Table 5.1: Comparison of filter orders 44

Table 5.2: Performance comparison for low pass filter 47

Table 5.3: Performance comparison for high pass filter 47

Table 5.4: Performance comparison for band pass filter 47

Table 5.5: Performance comparison for band stop filter 47

Table of Contents

CERTIFICATE ii

ACKNOWLEDGEMENT iii

Abstract iv

List of figures v

List of Tables vii

Chapter 1: INTRODUCTION 1

1.1 Background 1

1.2 Motivation 2

1.3 Present work 3

1.4 Thesis organization 3

Chapter 2: IIR FILTERS 5

2.1 Digital filters 5

2.2 IIR Filters 6

2.3 Filter specifications 7

2.3.1 Low pass filter specifications 8

2.3.2 High pass filter specifications 9

2.3.3 Band pass filter specifications 10

2.3.4 Band stop filter specifications 11

2.4 IIR filter stability 12

2.5 IIR filter design: Classical approaches 14

2.5.1 Butterworth filter 14

2.5.2 Chebyshev Type I filters 15

2.5.3 Chebyshev type II filters 16

2.5.4 Elliptical filters 16

2.6 IIR vs FIR filters 17

2.7 Summary 18

Chapter 3: MULTI-OBJECTIVE OPTIMIZATION 19

3.1 Introduction 19

3.2 Definitions 19

3.2.1 Multi-objective optimization 19

3.2.2 Pareto optimality 20

3.2.3 Pareto dominance 20
3.2.4 Pareto optimal set 20
3.2.5 Pareto Front 21
3.3 Multi objective evolutionary algorithms 22
3.4 Multi objective Particle Swarm Optimization 23
3.4.1 Single objective Particle Swarm Optimization 24
3.4.2 Multi objective Particle Swarm Optimization 25
3.5 Summary 30
Chapter 4: PROPOSED APPROACH 31
4.1 Introduction 31
4.2 Filter structure 31
4.3 Objective Functions 32
4.3.1 Magnitude response error 32
4.3.2 Linear phase response error 33
4.4 Initialization of parameters and design criteria 33
4.5 Proposed method description 34
4.6 Summary 35
Chapter 5: RESULTS AND DISCUSSIONS 36
5.1 Performance metrics 37
5.1.1 Magnitude response error 37
5.1.2 Linear phase response error 37
5.1.3 Order of the filter 37
5.2 Results of Butterworth filters 38
5.3 Results of Chebyshev I filters 39
5.4 Results of Chebyshev II filters 40
5.5 Results of Elliptical filters 41
5.5 Results of Proposed approach 42
5.6 Comparison of proposed approach with conventional approaches 43
5.7 Comparison of proposed approach with Genetic based approaches 46
Chapter 6: CONCLUSION 49
References 51

Chapter 1: INTRODUCTION

1.1 Background

In recent years, application of evolutionary algorithms to the design of digital Infinite impulse response (IIR) filters has gained much significance. IIR filter design is considered a very difficult task in digital signal processing. Classical techniques for IIR filter design are based on designing a prototype analog filter. Specifications of desired IIR filter are used to construct the analog filter prototype. Designed analog filters can be Butterworth filters, Chebyshev type I filters, Chebyshev type II filters or Elliptical filters [1, 2, 3, and 4]. These filters differ in terms of ripples present in passband or stopband. Butterworth filters are also known as 'maximally flat magnitude response' passband filters, Chebyshev type I filters have equal ripples in the passband, Chebyshev type II filters have equal ripples in the stopband and elliptical filters have equal ripples both in passband as well as stopband. Analog filter obtained from any of the above method is then converted to an equivalent digital filter. Two most commonly known methods for converting a filter from analog to digital domain are bilinear transformation and impulse invariance [5]. IIR filters designed using these methods suffer from a few disadvantages in terms of accuracy, filter order (and hence computational cost), stability and phase linearity [6]. This initiated the use of evolutionary algorithms in digital IIR filter design. There are several advantages of using evolutionary algorithms. First, inefficiency caused by transforming filter from analog to digital domain is removed. Second, multiple objectives can be dealt with simultaneously [7]. Etter et al. [8] introduced the use of Genetic algorithms (GA) for IIR filter design. Results were not very impressive due to novelty of GA at that time. Tang et al. [9]

proposed the use of Hierarchical Genetic Algorithm (HGA) to achieve multiple objectives of minimum magnitude response error and filter order. However linear phase requirement was still not met. Wang et al [10] minimized three objective functions namely magnitude response error, order of the filter and linear phase response error simultaneously to obtain an optimal IIR filter. It can be observed from the above literature review that most of the existing methods are based on genetic algorithms to achieve optimal filter design. The approach proposed in this thesis introduces multi objective particle Swarm optimization (MOPSO) in the design of digital IIR filter. PSO has an advantage of high speed of convergence and can be considered more suitable for multi-objective optimization problems [11]. In PSO, instead of using mutation and selection operators in a random way as in GA, particles are directed to move towards promising areas as they learn from their own local best positions as well as from best positions of other particles in the search space.

1.2 Motivation

IIR filter design can be viewed as a multi objective optimization problem which requires simultaneous minimization of three objective functions namely magnitude response error, order of the filter and linear phase response error. The main motivation behind using MOPSO for IIR filter design is that Pareto based Multi objective evolutionary algorithms (MOEAs) provide a set of solutions in a single run of the algorithm. This set of solution is called Pareto front and can be explored for a single solution based on application of the filter. For example, if the filter is required to process speech signals then linear phase may not be an important characteristic. In case of single objective optimization algorithms separate run is needed to find a solution for each application. Another reason for using MOPSO instead of GA is its high speed of convergence to find a set of solutions. In GA mutation and crossover operators are random in nature and thus search is not a directed one. However in PSO, each particle's position is calculated based on its own previous best position and global best position of all the particles in the space. Thus each particle is directed towards a promising region to find a solution resulting in higher rate of convergence.

1.3 Present work

This thesis proposes an algorithm for digital IIR filter design based on pareto based multi objective particle swarm optimization (MOPSO). Filter transfer function is realized in cascade form. This approach does not require analog to digital conversion. Coefficients of transfer function are obtained using MOPSO. Low pass, high pass, band pass and band stop filter types are designed. Since the minimum possible order for low pass and high pass filters is three and for band pass and band stop filters is four, the proposed approach fixes the filter order to minimum possible values to reduce the complexity. MOPSO has to deal with two objective functions namely magnitude response error and linear phase response error. Results of the proposed approach are compared with classical methods such as Butterworth filters, Chebyshev I filters, Chebyshev II filters and elliptical filters [1, 2, 3, 4].

1.4 Thesis organization

Chapter 1: Introduction

Chapter 2: **IIR Filters:** This chapter discusses the basics of digital filters, followed by IIR filters, IIR transfer function, filter specifications, IIR filter design methods and advantages of IIR filters.

Chapter 3: **Multi objective evolutionary Algorithms:** This chapter briefly describes the theory of evolutionary algorithms. We have also presented single objective and multi objective optimizations along with the terminology used for them. Single objective PSO algorithm is first given briefly to facilitate good understanding of multi objective PSO which is discussed at the end of the chapter.

Chapter 4: **Proposed Approach:** This chapter presents the algorithm proposed for IIR filter design using MOPSO. Objective functions are defined in this chapter. Filter specifications and stability conditions used are also given.

Chapter 5: **Results and Discussions:** Results of the proposed approach are discussed and compared with the existing classical methods. Comparison is done in terms of filter order, magnitude response error and linear phase response error.

Chapter 6: **Conclusion and future work**: Conclusion of the thesis is presented in this chapter. Also the scope and areas in which future investigation can be done are discussed.

2.1 Digital filters

A filter is a device that removes undesirable part of a signal such as noise or extracts significant part of a signal such as a particular range of frequencies. Filters have applications in almost all fields including signal processing, audio and video processing, medical areas, image processing and many more [12]. Basic block diagram of a filter is shown in fig 2.1.

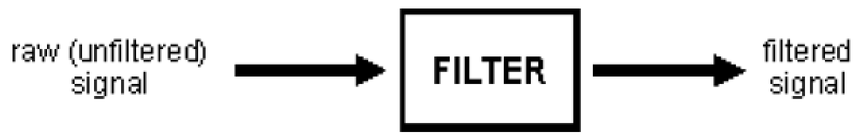

Fig 2.1: Block diagram of a filter

Filters can be classified as analog or digital filters based on the type of signals they operate upon. Digital filters offer several advantages over analog filters and hence they have nearly replaced analog filters. Some of the advantages of digital filters over analog filters are:

- Programmable: Digital filters are programmable i.e. they can be changed without changing the underlying hardware. Whereas changing an analog filter requires corresponding changes in the circuit.
- Easy to design, implement and test on a computer.
- Invariant to time, temperature and humidity whereas analog filters are unstable with respect to these factors.
- Free from aging and variations in manufacturing.

Although digital filters are expensive than analog filters, their benefits increase performance-to-cost ratio of digital filters [14]. Therefore digital filters are being widely used in digital signal processing. Digital filters can be classified as Finite impulse response (FIR) filters and Infinite impulse response (IIR) filters. IIR filters are filters with a feedback. They are recursive in nature. Hence present output is a function of present and past inputs as well as past outputs. FIR filters are non-recursive in nature. They are also known as feed forward filters. Block diagrams of FIR and IIR filters are shown in fig 2.2.

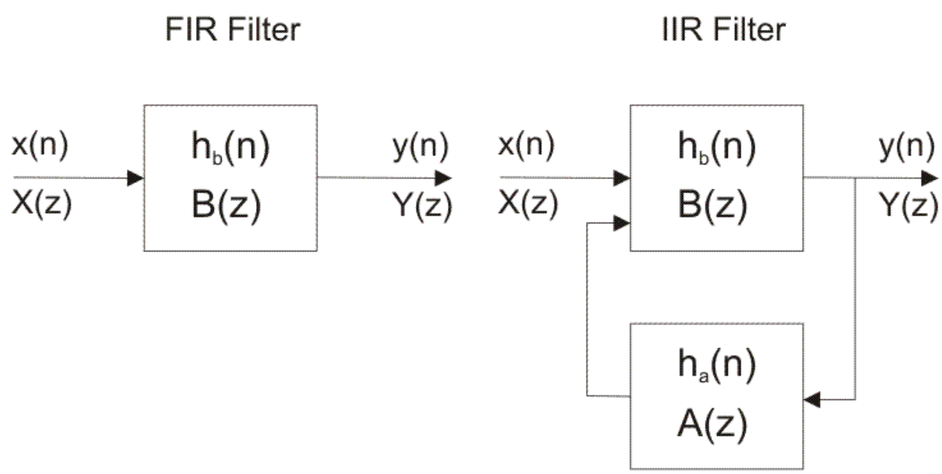

Fig 2.2: Block diagram of FIR and IIR filters

Basic properties of FIR filters are include phase linearity, stability and higher filter order. IIR filters may become unstable, have lower filter order resulting in less complexity and possess non-linear phase. This thesis focuses on IIR filters. IIR filter basics are discussed in further sections.

2.2 IIR Filters

IIR filters are filters with a feedback and hence provide better frequency response than FIR filter of equal order. IIR filters provide excellent solutions in applications where phase linearity is not of much importance. The effect of feedback in the filter is that when an impulse signal is applied to the filter, it results in a response that never decays to zero, hence it is named Infinite Impulse response filter. As discussed in section 2.1, output of the filter depends on previous and present input as well as past output. Transfer function of the filter in Z-domain can be expressed as [13]:

$$H(z) = \frac{b_0 + b_1 z^{-1} + ... + b_N z^{-N}}{1 + a_1 z^{-1} + ... a_M z^{-M}}$$

$$= \frac{\sum_{k=0}^{N} b_k z^{-k}}{1 + \sum_{k=1}^{M} a_k z^{-k}} \tag{2.1}$$

where a_k and b_k are filter coefficients.

IIR filter can also be expressed in cascade form as:

$$H(z) = K \prod_{i=1}^{n} \frac{1+b_i z^{-1}}{1+a_i z^{-1}} \prod_{j=1}^{m} \frac{1+d_{j1} z^{-1} + d_{j2} z^{-2}}{1+c_{j1} z^{-1} + c_{j2} z^{-2}} \tag{2.2}$$

Where K is the filter gain, a_i and b_i for i=1, 2… n are first order coefficients, c_{j1}, c_{j2}, d_{j1}, and d_{j2} for j=1, 2… m are the second order coefficients. Once all the filter coefficients are calculated, filter gain K is calculated so that magnitude response is normalized in the range [0, 1].

2.3 Filter specifications

While designing any filter, the first step is to determine the filter specifications. Filter specification includes the passband and stopband frequencies as well as attenuation permitted in passband and stopband. The range of frequencies in which signal is allowed to pass is called passband. On the other hand, the range of frequencies in which signal is not allowed to pass or is rejected is called stopband. Practically signals get distorted in both passband and stopband [19]. There is a limit on the amount of attenuation allowed in both the bands and is known as passband attenuation and stopband attenuation respectively. Filter specifications [16] for low pass, high pass, band pass and band stop filters are discussed in this section.

2.3.1 Low pass filter specifications

Low pass filters allow low frequency components of the signal to pass through them and reject the higher frequency components. Ideal low pass filter [18] is shown in fig 2.3.

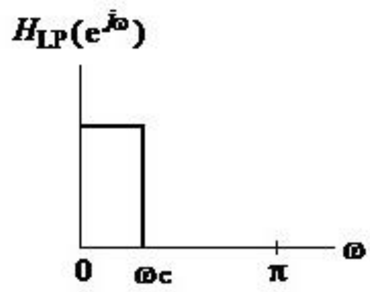

Fig 2.3: Ideal low pass filter

Where w_c is the cut off frequency. Ideal low pass filters cannot be realized in practicality as the transition from passband to stopband is discontinuous. Thus realizable low pass filter contains a transition band as well. This is shown in fig 2.4.

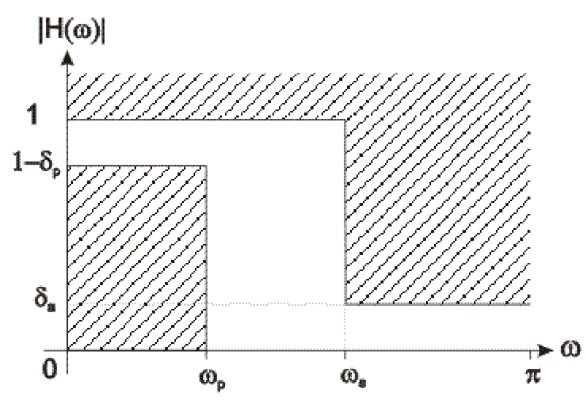

Fig 2.4: Practical low pass filter

Where $[0, w_p]$ is the passband, $[w_p, w_s]$ is the transition band, $[w_s, \infty]$ is the stopband, δp is maximum passband attenuation and δ_s is minimum stopband attenuation.

2.3.2 High pass filter specifications

High pass filters allow higher frequency components of a signal to pass through them and reject lower frequency components. Ideal high pass filter [18] is shown in fig 2.5

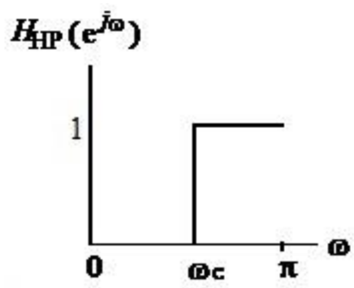

Fig 2.5: Ideal high pass filter

Where w_c is the cut off frequency. Ideal high pass filters cannot be realized in practicality as the transition from stopband to passband is discontinuous. Thus realizable high pass filter contains a transition band as well. This is shown in fig 2.6.

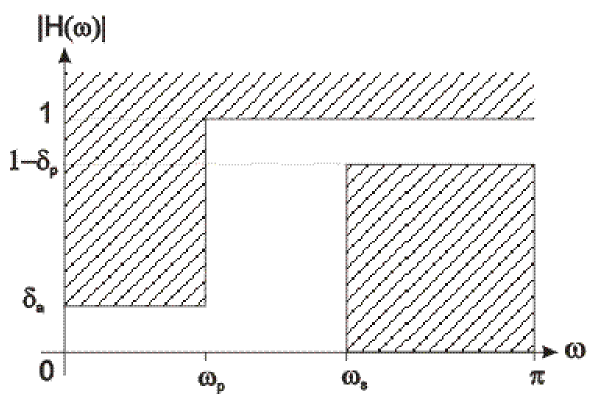

Fig 2.6: Practical high pass filter

Where $[0, w_s]$ is the passband, $[w_s, w_p]$ is the transition band, $[w_p, \infty]$ is the stopband, δp is maximum passband attenuation and δ_s is minimum stopband attenuation.

2.3.3 Band pass filter specifications

Band pass filters allow only those frequency components of a signal to pass through them that lie within a particular range. Components lying outside that range are rejected. Ideal band pass filter [18] is shown in fig 2.7.

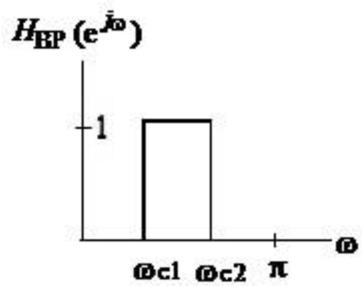

Fig 2.7: Ideal bandpass filter

Where w_{c1} and w_{c2} are the cut off frequencies. Like low pass and high pass filters, ideal band pass filters also cannot be realized practically [19]. Thus realizable band pass filter contains two transition bands as shown in fig 2.8.

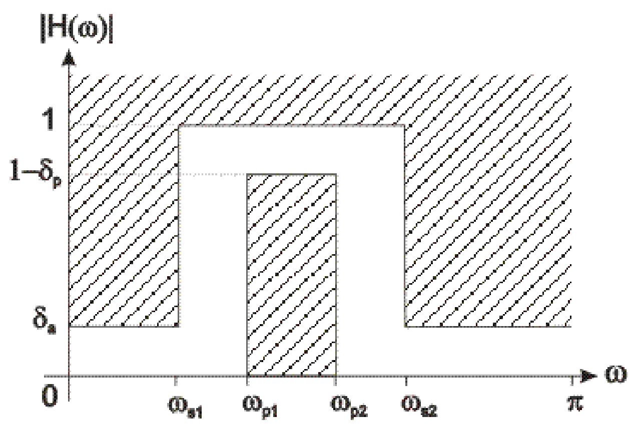

Fig 2.8: Practical bandpass filter

Where [0, w_{s1}] and [w_{s2}, ∞] are the stop bands, [w_{p1}, w_{p2}] is the passband, [w_{s1}, w_{p1}] and [w_{p2},w_{s2}] are transition bands, δ_p is maximum passband attenuation and δ_s is minimum stopband attenuation.

2.3.4 Band stop filter specifications

Band pass filters reject frequency components of a signal that lie within a particular range. Components lying outside that range are accepted. Ideal band stop filter [18] is shown in fig 2.9.

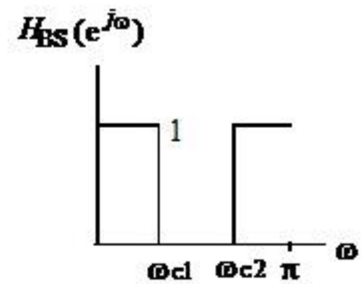

Fig 2.9: Ideal bandstop filter

Where w_{c1} and w_{c2} are the cut off frequencies. Like other filters discussed in previous sections, ideal band stop filters are also practically not possible to realize. Thus realizable band stop filter contains two transition bands as shown in fig 2.10.

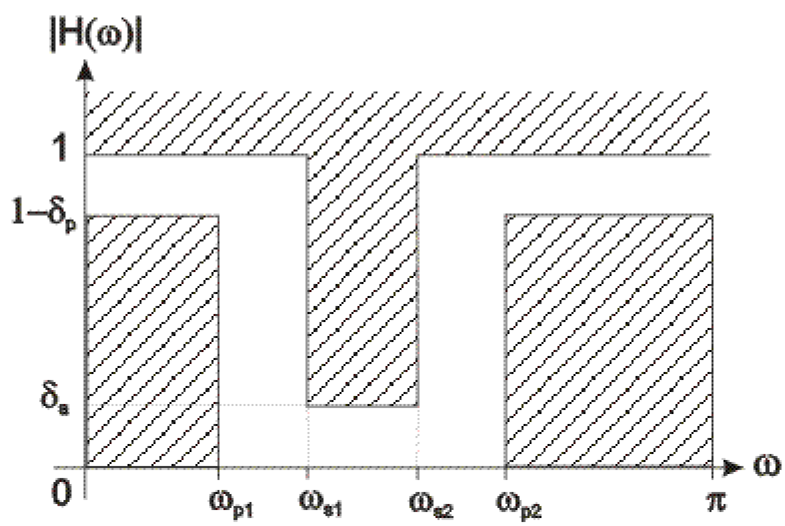

Fig 2.10: practical bandstop filter

Where [0, w_{p1}] and [w_{p2}, ∞] are the pass bands, [w_{s1}, w_{s2}] is the stopband, [w_{p1}, w_{s1}] and [w_{s2}, w_{p2}] are the transition bands, δ_p is maximum passband attenuation and δ_s is minimum stopband attenuation.

2.4 IIR filter stability

In IIR filters, stability is determined by using poles of z-transform of the transfer function. Poles can be calculated by determining roots of the denominator of transfer function. If all the poles of the filter lie within the unit circle in z-domain, then the filter is a stable one [17]. Poles lying outside the unit circle leads to instability in IIR filters. An example of stable IIR filter is shown in fig 2.11.

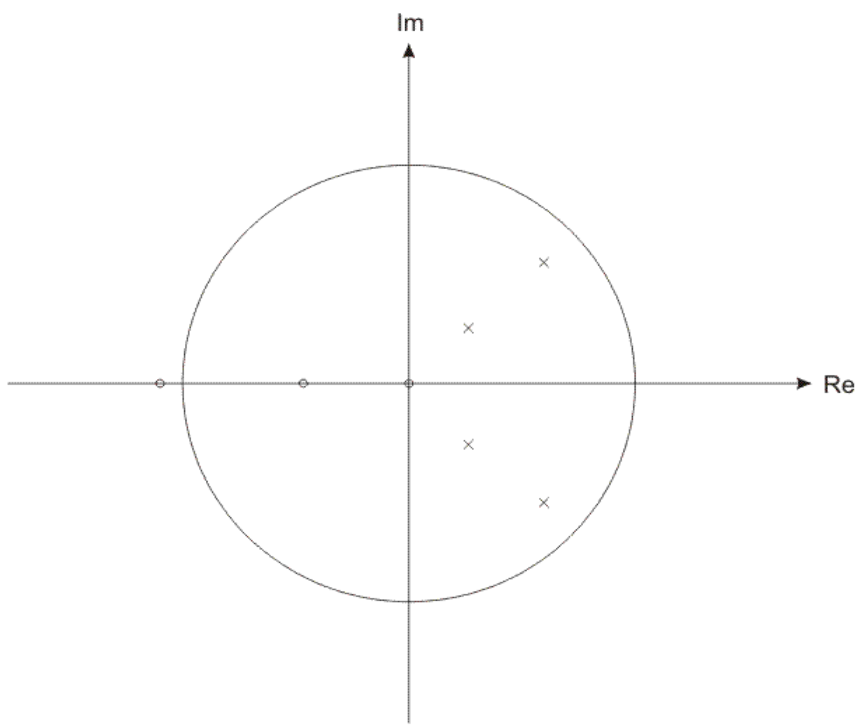

Fig 2.11: A stable IIR filter

In the above figure small crosses denote the poles and small circles denote the zeros of the transfer function. Re and Im denote real and imaginary axis respectively. It can be seen from the figure that all the poles of the filter lie within the unit circle providing it stability. It can be seen that one zero of the filter is lying outside the circle but zeros don't affect the stability of the filter. An unstable IIR filter is shown in fig 2.12.

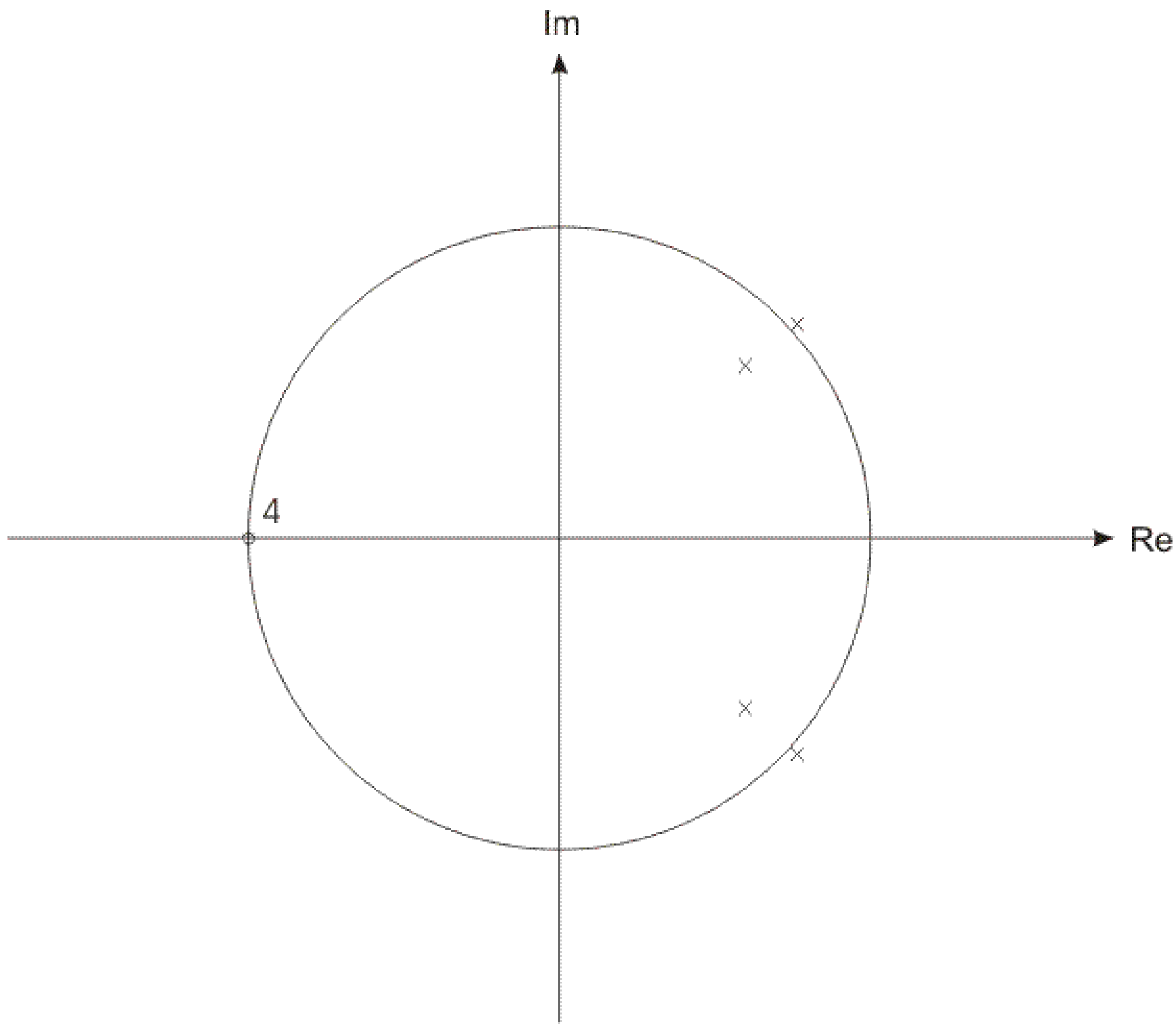

Fig 2.12: An unstable IIR filter

Two poles lying outside the unit circle in above figure depict the instability of the filter. Since FIR filters are non-recursive in nature all the poles lie on the origin resulting in stability of the filter. Thus poles are used to test the stability of IIR filters only. FIR filters are always stable.

2.5 IIR filter design: Classical approaches

Classical techniques for IIR filter design are based on designing a prototype analog filter. Specifications of desired IIR filter are used to construct the analog filter prototype. Designed analog filters can be Butterworth filters, Chebyshev type I filters, Chebyshev type II filters or Elliptical filters. These filters are then converted back to digital filters using of the following techniques.

- Bilinear transformation method
- Impulse invariance method

The evolutionary algorithm based techniques for IIR filter design do not depend on analog prototype for the filter design. Digital filter is designed directly by defining a transfer function and optimizing its coefficients to meet the filter specifications. Since the results of Butterworth, Chebyshev I, Chebyshev II and elliptical filters are compared with the proposed approach, these filters are briefly discussed in this section.

2.5.1 Butterworth filter

Butterworth filters possess flat magnitude response in the passband [1]. There are no ripples present in passband or stopband. This results in a very high transition width between passband and stopband. Design specifications of Butterworth filter includes filter order and cut-off frequency [19]. If transition width is reduced it may result in increase of filter order. Fig 2.13 compares 7th order and 12th order Butterworth filters.

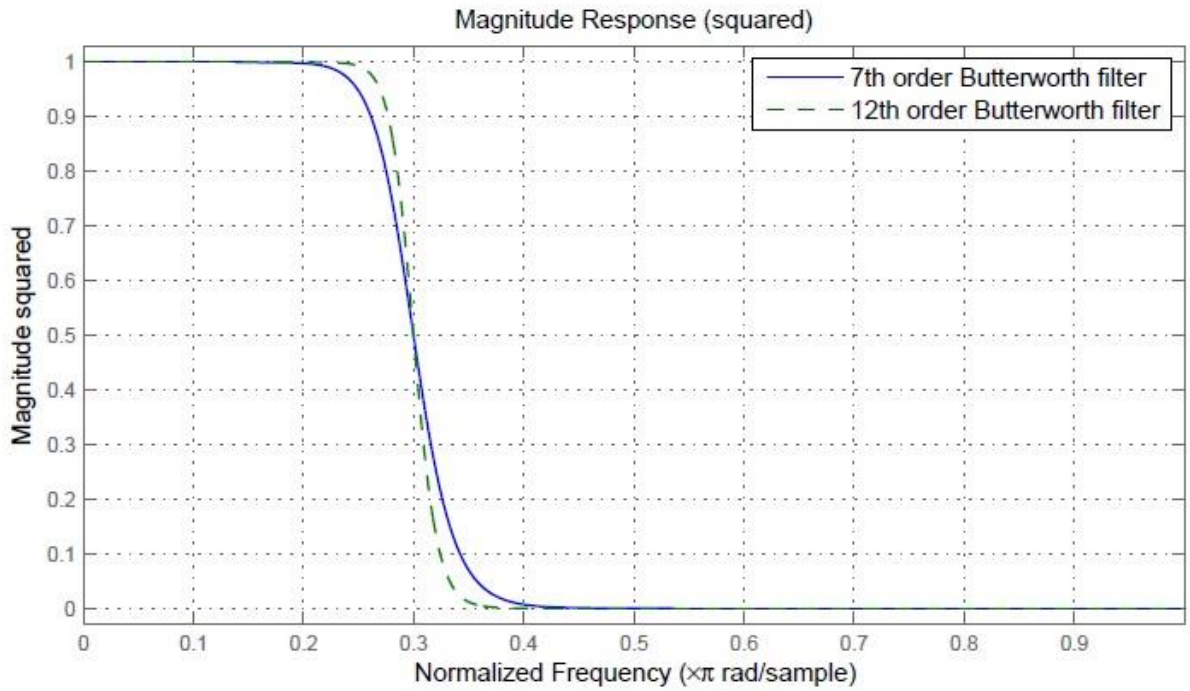

Fig 2.13: 7^{th} and 12^{th} order Butterworth filters

2.5.2 Chebyshev Type I filters

Chebyshev type I filters allow presence of ripples in the passband whereas the stopband has flat magnitude response. Transition width can be reduced as compared to Butterworth filter but with the introduction of ripples in the passband. Fig 2.14 compares Chebyshev I filter with the Butterworth filter.

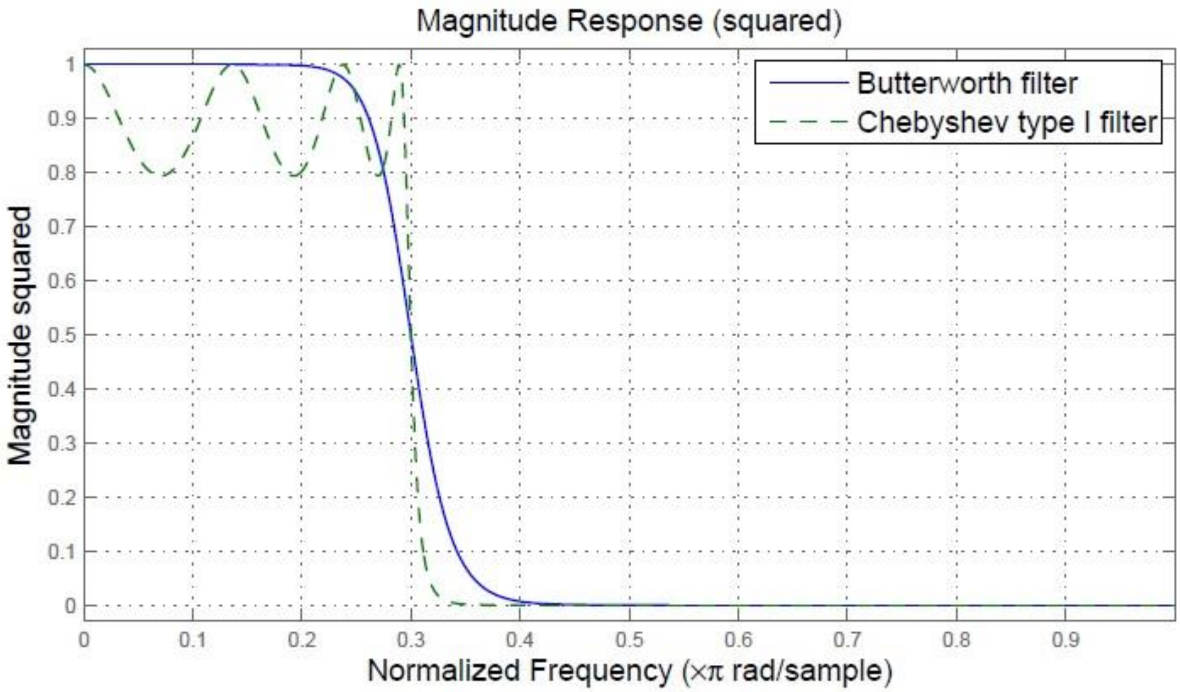

Fig 2.14: Chebyshev and Butterworth filter of same order

It can be observed from the above figure that Chebyshev type I filter has smaller transition width as compared to Butterworth filter of same order [15]. This is achieved by allowing ripples in the passband.

2.5.3 Chebyshev type II filters

Chebyshev type II filters have flat magnitude response [15] in the passband and equiripple in the stopband. These filters are preferred over both the filters discussed in previous sections. Chebyshev II filters are also called inverse Chebyshev filters and allow for a lower transition width.

2.5.4 Elliptical filters

Elliptical filters allow ripples in passband as well as stopband [20]. They can be viewed as a generalization of Chebyshev and Butterworth filters. Fig 2.15 shows elliptic and Chebyshev II filters.

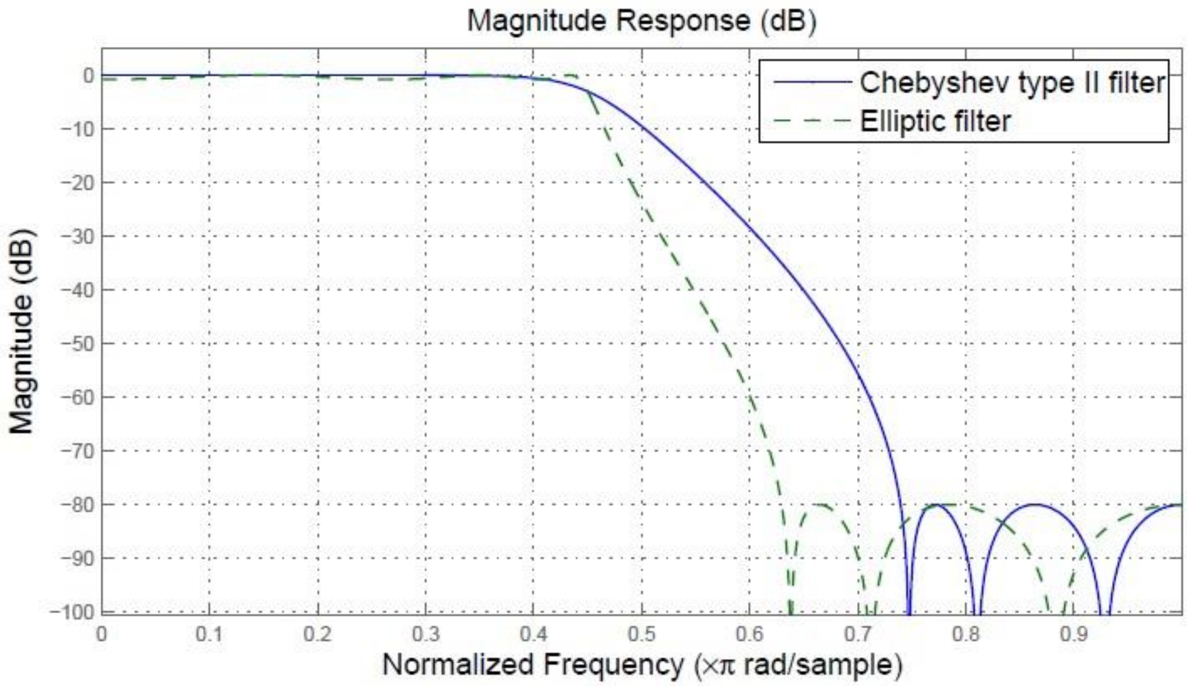

Fig 2.15: Chebyshev type II and Elliptic filters

2.6 IIR vs FIR filters

While designing a filter it is difficult to decide which type of filter i.e. IIR or FIR should be designed [19]. Depending upon the requirements of the application, a desired filter type is chosen. The factors that affect this decision are given in table 2.1.

Table 2.1: IIR vs FIR filters

Property	FIR	IIR
Phase linearity	Linear in phase	Non linear phase
Order	High filter order	Low filter order
Stability	Always stable	May become unstable
Derivation	No analog prototype required	Analog prototype designed first
Computational resources	More number required	Less number required

The use of evolutionary algorithms can provide phase linearity and stability to IIR filters without the need of designing an analog prototype. Thus such IIR filters can be used for almost any application and provide more performance to cost ratio.

2.7 Summary

This chapter contains the basic definitions and concepts related to IIR filters. Filter specifications for Low pass, high pass, band pass and band stop types are discussed. Some classical approaches to IIR filter design are also explained briefly. Chapter is concluded with a comparison of FIR and IIR filter properties.

Chapter 3: MULTI-OBJECTIVE OPTIMIZATION

3.1 Introduction

Most of the real world problems deal with simultaneous objective functions. Single objective optimization differs from multi-objective optimization as concept of ‘optimum’ has to be re-defined. Multi objective optimization algorithms generate more than one trade-off solutions in single run of the algorithm [22]. Decision maker selects one of these solutions depending upon the application. Classical approaches to achieve multiple objectives simultaneously involve conversion of multi-objective problem into a single objective problem [21]. The drawback of this approach is that algorithm has to be run many times to generate pareto-optimal set of solutions. Solutions which are not worse than other solutions in terms of all objective functions are said to be acceptable optimal trade-off solutions. These solutions form a set called pareto-optimal solutions [23]. Section 3.2 discusses some of the definitions related to the multi objective optimization.

3.2 Definitions

3.2.1 Multi-objective optimization

Multi-objective optimization problems deal with k objective functions simultaneously. These problems deal with minimization of all functions, maximization of all functions or a combination of minimization and maximization of these functions [24]. Let P € R_n be an n-dimensional search space, and f_i (x), i=1, 2… k be k objective functions. Then mathematically F(x) can be given as:

$$F(x) = Minimize \begin{bmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_k(x) \end{bmatrix} \tag{3.1}$$

Subject to some constraints that depend upon the application. Multi objective algorithms generate multiple solutions with the use of Pareto optimal theory [23].

3.2.2 Pareto optimality

A solution x ∈ Ω is said to be Pareto optimal wrt universe Ω iff there exists no x′ ∈ Ω for which s_1 = F (x′) = (f_1 (x′) . . . f_k (x′)) dominates s_2 = F(x) = (f_1(x) . . . f_k(x)) [25]. In other words, x′ is said to be Pareto optimal if there is no vector x which would decrease some objective function value without causing a simultaneous increase in at least one objective function value(assuming all objectives need to be minimized).

3.2.3 Pareto dominance

A solution $s_1 = (s_{11}, s_{12} ... s_{1k})$ is said to dominate another solution $s_2 = (s_{21}, s_{22} ... s_{2k})$ (denoted by $s_1 \preceq s_2$) iff s_1 is partially less than s_2, i.e. [26]

$$\begin{aligned} &\forall i \in \{1, 2...k\}, s_1 \preceq s_2 \\ &\wedge \exists i \in \{1, 2...k\}, s_1 \prec s_2 \end{aligned} \tag{3.2}$$

3.2.4 Pareto optimal set

For a given Multi objective problem F(x), the Pareto Optimal Set, P*, is defined as:

$$P^* = \{x \in \Omega \mid \neg \exists x' \in \Omega, F(x') \le F(x)\} \tag{3.3}$$

Pareto optimal solutions are those solutions in decision space, whose objective function values cannot be further minimized simultaneously (assuming minimization). This entire set of solutions is represented by P^* and are also known as admissible, efficient, non-inferior or nondominated solutions [23].

3.2.5 Pareto Front

For a given Multi objective problem, F(x), and Pareto Optimal set, P*, the Pareto Front PF* is defined as:

$$PF^* = \{u = F(x) \mid x \in P^*\} \tag{3.4}$$

All the solutions in Pareto optimal set when plotted in objective function space constitute Pareto front. In fig 3.1 one example of Pareto front is shown. Single objective optimization problems have a single optimal solution, whereas multi objective problems have uncountable set of solutions on a Pareto front [26].

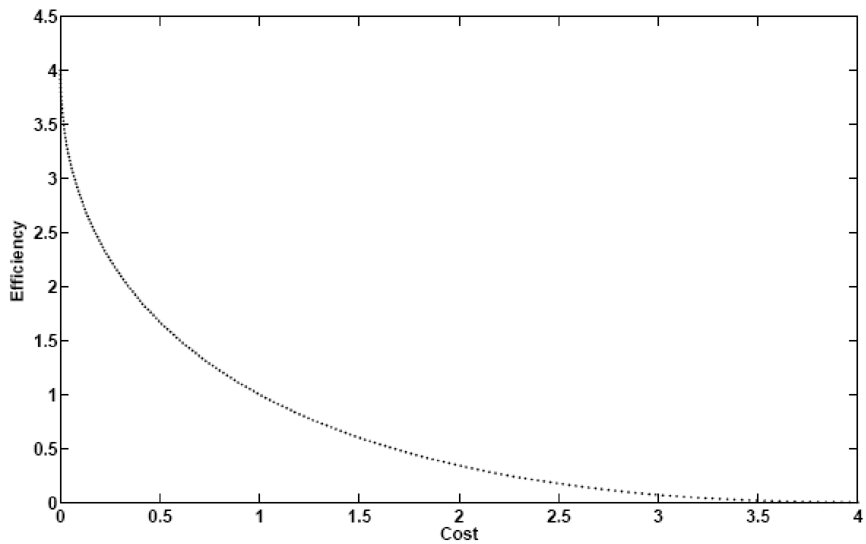

Fig 3.1: Pareto front of MOP with 2 objective functions: Cost and efficiency

Each point in the Pareto front corresponds to a solution in the decision space. This solution in the decision space represents tradeoffs in the decision space. The MOPs evaluation function,

$f:\Omega \rightarrow \Lambda$ maps variables in decision space $x=(x_1, x_2...x_n)$ to the vectors $y=(a_1, a_2...a_k)$ in solution space.

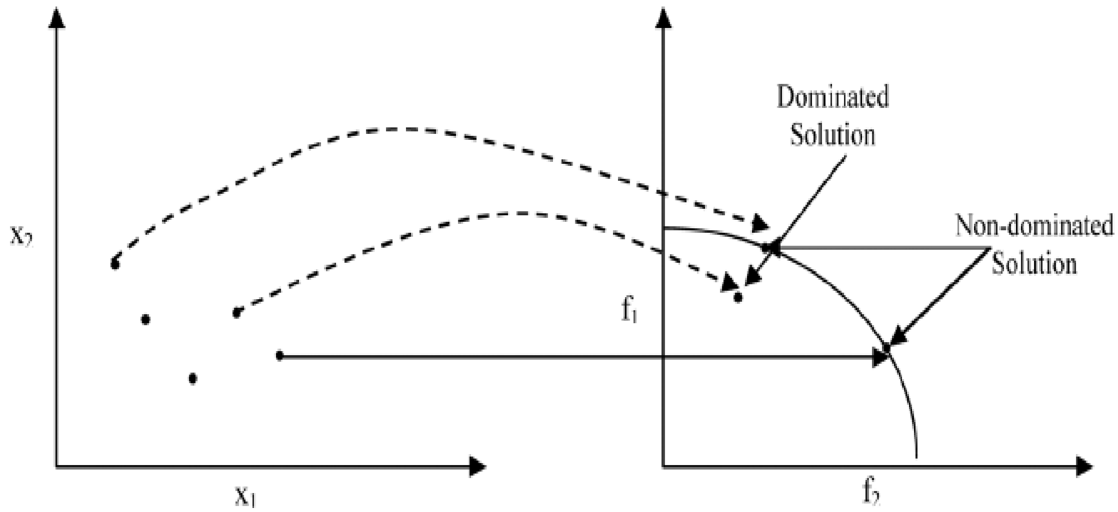

Fig 3.2: mapping from decision variable space to objective function space

3.3 Multi objective evolutionary algorithms

Evolutionary algorithms belong to a class of algorithms that are inspired by natural evolution of particles. In recent years, these algorithms have been applied to various optimization problems. Real world problems are often subject to constraints and may have many conflicting objectives. Many multi objective evolutionary algorithms (MOEAs) have been proposed in past for such applications. These algorithms generate a set of solutions rather than a single solution as in single objective optimization problems. There are various ways in which multiple objectives can be achieved based on which MOEAs can be classified into 3 broad categories.

- Weighted sum approach: This approach converts multiple objectives into a single objective function by taking weighted sum of these objective functions. Sum of weights is equal to one. Problem is then solved similar to single objective optimization problem [27].

- Lexographical approach: in this approach each objective function is assigned a priority. Most important objective functions are assigned higher priorities than other ones. Objective functions are thus optimized according to the priorities assigned.
- Pareto based approach: This approach aims at finding multiple nondominated solutions in a single run of the algorithm. Selecting one solution from the Pareto front requires some previous problem knowledge. Some of these approaches include Multi-objective (MOGA) proposed by Fonseca and Fleming in 1993 [28], Non-dominated Sorting GA (NSGA) proposed by Srinivas and Deb in 1994[29, 30].

MOEA used in this thesis belongs to third category of algorithms. Pareto Based multi objective particle swarm optimization (MOPSO) has been used to achieve multiple objectives in the filter design process. Section 3.4 first briefly describes single objective PSO followed by MOPSO algorithm.

3.4 Multi objective Particle Swarm Optimization

Kennedy and Eberhart proposed Particle Swarm Optimization (PSO) algorithm in 1995 as an extension of animal's cognitive and social behavior system [21]. PSO is inspired by behavior of group of particles like flocking of birds and schooling of fishes. As other evolutionary algorithms, PSO also deals with a population (called swa*rm*) of possible solutions (called *particles*) which are updated at each iteration of the algorithm. However, the updation of swarm is different in case of PSO. PSO follows a cooperative rather than a competitive approach. Instead of using random crossover, selection and mutation operators to update the population and favor those individuals that are performing best, PSO updates velocity of particles at each iteration adaptively. The particles move towards promising areas in the search space by learning exploiting from their own experience, as well as the experience of other particles. In MOPSO a separate memory called external repository is used to store each particle's best position it has ever visited in the search space. Section 3.4.1 discusses single objective PSO briefly.

3.4.1 Single objective Particle Swarm Optimization

Algorithm consists of a number of particles moving in search space with the aim to reach a global minimum value of fitness function. Each particle in the search space represents a candidate solution and has some velocity according to which it moves in the search space. It also possesses memory to keep information about its previously visited space. Hence movement of each particle is influenced by two factors: local best solution due to itself and global best solution due to all the particles in the solution space. During an iteration of the algorithm, local and global best positions of the particle are updated if better solution is found. The process is repeated until the desired result is achieved or specified number of iterations have completed. In N-dimensional space, position of i^{th} particle is denoted as $S_i = (s_{i1}, s_{i2}... s_{iN})$; velocity is denoted as $V_i= (v_{i1}, v_{i2}... v_{iN})$; particle's local and global best positions are denoted as $Pbest_i = (p_{i1}, p_{i2}... p_{iN})$ and $gbest = (p_{g1}, p_{g2}... p_{gN})$ respectively. Velocity of particle [15] at i^{th} iteration is given by eqn 3.5.

$$V_i^{k+1} = \omega V_i^k + \alpha * rand_1 * (pbest_i - s_i^k) + \beta * rand_2 * (gbest - s_i^k) \tag{3.5}$$

Position of particle [15] at i^{th} iteration is given by eqn 3.6.

$$s_i^{k+1} = s_i^k + V_i^{k+1} \tag{3.6}$$

Where i=1, 2 ...M; M is the number of particles; d=1, 2...N; N is the number of dimensions; α is cognitive parameter; β is social parameter; $rand_1$ and $rand_2$ are random numbers between 0 and 1; ω is inertia weight; k = 1, 2, 3... ; are iteration steps.

3.4.2 Multi objective Particle Swarm Optimization

Coello et al [31] proposed the first Pareto based Multi-objective PSO (MOPSO). In MOPSO, the nondominated solutions called leaders are stored in an external repository. The search space is divided into Hypercubes. Each hypercube has a fitness that is assigned with a value inversely proportional to the number of particles in the hypercube. One leader is selected from a hypercube with best fitness value. There are different approaches to select a leader. Thus, the velocity update for the *i*-th particle [32] becomes

$$vel[i+1] = \omega * vel[i] + r_1 * (pbest[i] - pos[i]) + r_2 * (rep[h] - pos[i]) \tag{3.7}$$

Where ω is the inertia weight, r_1 and r_2 are random numbers between [0, 1], $pbest[i]$ is the local best position of particle i, $rep[h]$ is the global best position in the external repository. After calculating velocity of the particle [32] using equation 3.8, its position can be updated as:

$$pos[i+1] = pos[i] + vel[i+1] \tag{3.8}$$

Where $pos[i+1]$ and $pos[i]$ are the positions of particle at $(i+1)^{st}$ and i^{th} iteration respectively. $vel[i+1]$ is the velocity of particle at $(i+1)^{st}$ iteration calculated using equation 3.7. The external repository has limited size. When it becomes full better solutions are inserted into it based on some criterion discussed in next section.

3.4.2.1 Main algorithm

The algorithm of MOPSO is given as follows:

Step 1) Initialize the position of each particle in the population *pop*.

Step 2) Initialize the velocity of each particle *vel*.

Step 3) Evaluate objective function values for each of the particles in *pop*.

Step 4) Store the position values of nondominated particles in the external repository *rep*.

Step 5) Create hyper cubes of the search space exploited till now, and assuming this hypercube system to be a coordinate system, locate each particles position wrt its objective function values.

Step 6) Update each particle's memory values. This memory can be used to guide the particles to fly towards promising regions. These values are also stored in external repository.

Step 7) WHILE maximum number of iterations

DO

a) Compute the speed of each particle using equation 3.7. Rep[h] represents global best position that is taken from the external repository. Each hypercube is assigned a fitness value inversely proportional to the number of particles it contains. Thus hyper cubes with more number of particles have greater fitness as compared to hyper cubes with less number of particles. One hypercube is then selected using roulette-wheel selection [31]. After selecting the hypercube, a particle is selected randomly from that hypercube.

b) New positions of the particles are updated using equation 3.8. This equation uses velocity of the particle obtained from the previous step.

c) Particles are maintained within the search space. They are not allowed to go beyond the boundaries. Thus solutions that lie beyond valid search space are discarded. If a variable crosses its boundary two steps are taken.

1) The decision variable is assigned a value equal to its corresponding lower or upper boundary

2) The velocity of corresponding particle is reversed by multiplying it by (-1) to direct the search in opposite direction.

d) Evaluate objective function values for each of the particles in *pop*.

e) Update the contents of *rep* with the nondominated solutions. Dominated solutions in the repository are eliminated due to size constraint of the repository. A secondary criterion for retention of particles in repository is based on number of particles in the objective solution space.

f) If a particle with a position better than the position contained in its memory is found, the position of the particle is updated with the new better value.

$$pbest[i] = pop[i] \tag{3.9}$$

Concept of Pareto dominance is used to decide the position in memory to be retained. If the current position dominates the position in memory, it replaces that position in memory. Otherwise none of them dominates the other; any one position is selected randomly.

g) Increment the loop counter

8) END WHILE

3.4.2.2 External repository

External repository is used in Pareto based MOPSO to store the non dominated solutions. The non dominated or efficient solutions found during the search process are stored in the repository. Repository is of fixed size. To manage the repository it has two main parts: archive controller and the adaptive grid.

Archive controller:

The archive controller is used as a decision maker, to take a decision whether to accept a solution in the archive. An archive is a memory space that contains non dominated solutions. The decision process can be given as follows: Initially when the external repository archive is empty, then the non-dominated solutions found are simply added to the archive [31]. If archive already contains some solutions then the current nondominated solutions found at each iteration are compared with the archive contents. Four cases may arise during the comparison:

Case1: If the new solution is dominated by the solutions in the archive; discard the new solution. Archive remains unaltered.

Case2: If the new solution is not dominated by any solution in the archive and there is space in the archive; new solution is added to the archive.

Case3: If the new solution dominates some of the solution in the archive; replace them with the new solution.

Case 4: If the external archive becomes full i.e. it reaches its maximum limit, then adaptive grid procedure is adopted to update the archive contents.

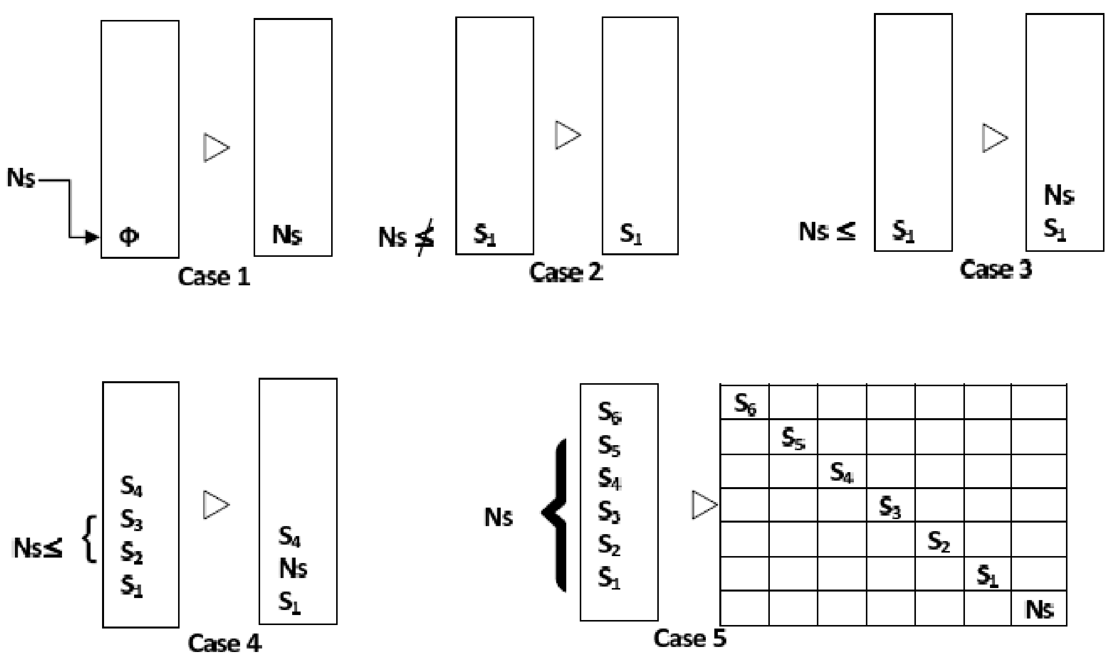

Fig 3.3: Possible cases for the archive controller

Adaptive Grid:

In Pareto based approaches the basic idea is to use an external archive to store efficient solutions in the archive. Objective function space of solutions in the archive is divided into regions as shown in Fig. 3.4. If the solution inserted in the archive goes outside the current boundaries of the grid, then the grid has to be recalculated and each solution within it has to be relocated. This is the reason it is called adaptive grid. Adaptive grid method is preferred over other methods like nitching because of its lower computational cost [31, 32].

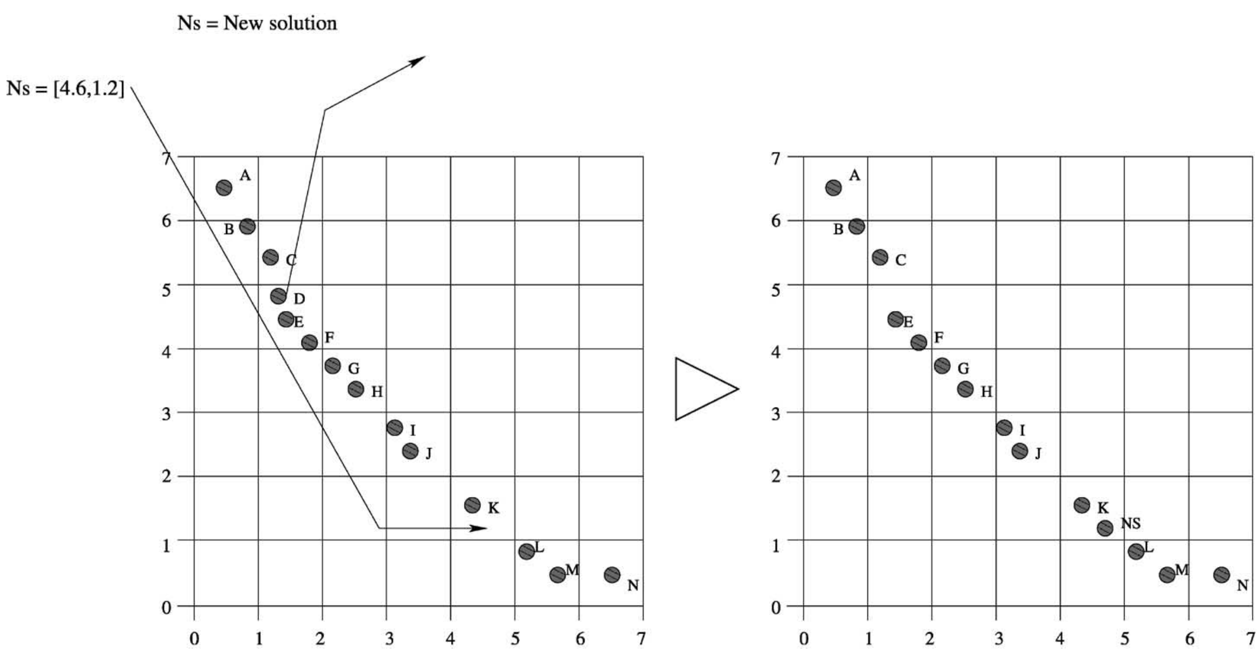

Fig 3.4: insertion of element in the grid when it lies within the current boundaries

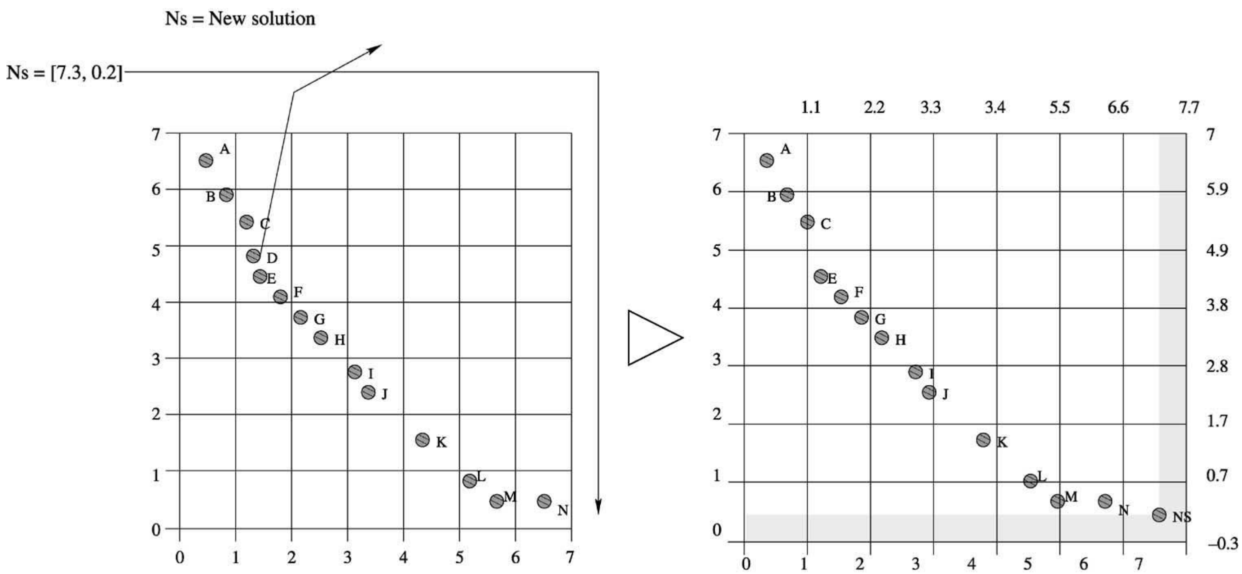

Fig 3.5: insertion of element in the grid when it lies outside the current boundaries

3.5 Summary

The essence of multi objective evolutionary algorithms lies in their capability to find a number of nondominated or efficient solutions in single run of the algorithm. This advantage has increased the incorporation of MOEAs into various applications. In this chapter, basic terminology related to multi objective evolutionary algorithms are discussed. Also multi objective PSO algorithm is presented in detail.

4.1 Introduction

IIR filter design is considered one of the difficult tasks in digital signal processing. IIR filters use less computational resources and hence are used in applications where resources are at a premium. IIR filters designed using conventional approaches cannot achieve stability and phase linearity simultaneously. With the introduction of MOEAs in IIR filter design it is possible to optimize multiple objectives concurrently. MOEAs consider all objectives equally and provide a set of solutions to the decision maker. This set of solutions is called Pareto optimal solutions. In this thesis Pareto based Multi objective Particle swarm optimization is used to design IIR filter. Algorithm aims at finding a set of coefficients of the filter transfer function which meet the stated requirements. IIR filter design requires concurrent minimization of three objective functions namely 1) magnitude response error 2) Linear phase response error and 3) order of the filter. Filter in designed in cascade form. Basic filter structure is discussed in the next section.

4.2 Filter structure

IIR filter design in the proposed work starts with a pre assumed filter structure. In cascade form filter structure can be represented as

$$H(z) = K \prod_{i=1}^{n} \frac{1+b_i z^{-1}}{1+a_i z^{-1}} \prod_{j=1}^{m} \frac{1+d_{j1} z^{-1}+d_{j2} z^{-2}}{1+c_{j1} z^{-1}+c_{j2} z^{-2}} \tag{4.1}$$

Where K is the filter gain, a_i and b_i for i=1, 2… n are first order coefficients, c_j1, c_j2, d_j1, and d_j2 for j=1, 2… m are the second order coefficients. Once all the filter coefficients are calculated, filter gain K is calculated so that magnitude response is normalized in the range [0, 1]. LP, HP, BP and BS filter types are designed with minimum possible order of 3,3,4,4 respectively. Once the filter structure is decided, MOPSO is used to obtain filter coefficients that satisfy other two objective functions.

4.3 Objective Functions

IIR filter design methods in the literature deal with single objective function. Single objective methods generally tend to ignore secondary and tertiary objectives necessary for the optimal filter design. Proposed approach deals with 2 objective functions 1) magnitude response error and 2) linear phase response error. Order of the filter is kept o minimum possible for the desired filter type. Definitions of these fitness functions are discussed in this section.

4.3.1 Magnitude response error

IIR filter design should satisfy some magnitude response conditions

a) The passband attenuation should be less than δ_1.

b) The stopband attenuation should not be less than $1 - \delta_2$.

The passband and stopband cutoff frequencies are represented by ω_p and ω_s, respectively. Then magnitude response error can be defined as [9, 10]:

$$H_p(\omega) = \begin{cases} 1 - \delta_1 - \left|H\left(e^{j\omega}\right)\right|, & \left|H\left(e^{j\omega}\right)\right| < 1 - \delta_1 \\ 0, & \left|H\left(e^{j\omega}\right)\right| \geq 1 - \delta_1 \end{cases} \tag{4.2}$$

Where ω lies in passband and δ_1 is the attenuation in passband.

$$H_s(\omega) = \begin{cases} \left|H\left(e^{j\omega}\right)\right| - \delta_2, & \left|H\left(e^{j\omega}\right)\right| \geq \delta_2 \\ 0, & \left|H\left(e^{j\omega}\right)\right| \leq \delta_2 \end{cases} \tag{4.3}$$

Where ω lies in the stopband range and δ_2 is the stopband attenuation. $H_p(\omega)$ and $H_s(\omega)$ are the passband and stopband magnitude response errors respectively. Thus first objective function is given as:

$$f1 = \sum_{\omega_p} Hp\left(\omega_p\right) + \sum_{\omega_s} Hs\left(\omega_s\right) \tag{4.4}$$

Where ω_p and ω_s are the sampling frequencies in passband and stopband respectively.

4.3.2 Linear phase response error

The linear phase response is considered in the passband as well as transition band. The non linearity in phase response can lead to a large amount of distortion in the response of the filter. Phase response of the filter is sampled at equal intervals of frequency. The phase sequence is obtained as [9, 10]:

$$Ph = \{\theta_1, \theta_2 \ldots \theta_n\} \tag{4.5}$$

After obtaining the phase sequence, the phase difference between the consecutive values in phase sequence can be calculated as:

$$\Delta Ph = \{\Delta\theta_1, \Delta\theta_2 \ldots \Delta\theta_{n-1}\} \tag{4.6}$$

$\Delta\theta_{j+1} = \Delta\theta_{j+1} - \Delta\theta_j$. If the phase is linear then the values in ΔPh are same. This is because the linearity in phase results in equal difference values in the phase sequence. Phase response error can be calculated by finding the variance of ΔPh sequence. Thus, the second objective function is given as

$$f2 = \text{variance}\ (\Delta Ph) \tag{4.7}$$

4.4 Initialization of parameters and design criteria

In proposed method, IIR filter is designed in cascade form. A filter structure is assumed based on the desired filter type. Number of coefficients can be determined using the filter structure. These coefficients are represented by a particle's position. Each particle's initial position and velocity are chosen randomly. Swarm size of 50 is taken. Maximum number of iterations is fixed to 70. Inertia weight, social parameter and cognitive parameter are initialized to 1, 2, and 2 respectively. Design criteria for the different filter types [10] is summarized in table 4.1

Table 4.1: Design Criteria

Filter type	ω_p	ω_s	δ_1	δ_2
LP	[0, 0.2Π]	[0.3Π, Π]	0.108	0.1778
HP	[0.8Π,Π]	[0,0.7Π]	0.108	0.1778
BP	[0.4Π, 0.6 Π]	[0,0.25Π] U [0.75Π,Π]	0.108	0.1778
BS	[0 ,0.25Π] U [0.75Π, Π]	[0.4Π,0.6 Π]	0.108	0.1778

4.5 Proposed method description

First the position and velocity of particles are initialized randomly. Particles position represents coefficient values and dimension of search space is equal to the number of coefficients. Objective functions are then evaluated for each particle. These are discussed in previous section. External repository is initialized with nondominated particles. Hypercubes are generated and particles are located within it according to their objective function values. A leader is selected from hyper cubes and position and velocity of particles are updated using update equations discussed in chapter 3. Updation process guides the particles towards promising areas. This is because search is guided by best values obtained so far by the particle as well as its neighbors. At the end of fixed number of iterations, a set of solutions called Pareto optimal set is obtained.

4.6 Summary

The algorithm proposed for the design of IIR filter is presented in this chapter. To obtain an optimal design, it is treated as a multi objective optimization problem. Objective functions namely magnitude response error and linear phase response error are also defined. While designing any filter type, foremost requirement is the design criteria. It specifies cutoff frequencies, desired attenuation values and other filter requirement. Thus design criteria and initial parameter values are also discussed.

Chapter 5: RESULTS AND DISCUSSIONS

This chapter shows the results of filter design using proposed approach. Four conventional approaches namely Butterworth, Chebyshev I, Chebyshev II and elliptical filter design methods are compared with the proposed approach. Results of two evolutionary based algorithm namely hierarchical genetic algorithms (HGA) for IIR filter design [9] and Cooperative coevolutionary GA (CCGA) [10] based IIR filter design are also shown and compared with the approach proposed in this thesis. Metrics used for comparison are magnitude response error, order of the filter and linear phase response error. All the codes are implemented in MATLAB7.0. Performance of IIR filters depends on various factors. These factors together known as design criteria need to be kept same for comparing different approaches. Design criteria specification includes

a) Filter type i.e. low pass, high pass, band pass or band stop

b) Cutoff frequencies: it is a single value for low pass and high pass filters and two values for band pass and band stop filter types

c) Maximum passband attenuation

d) Minimum stopband attenuation

The design criteria used in this thesis in given in table 4.1 and is used for all the approaches to facilitate fair comparison among the techniques.

5.1 Performance metrics

Three performance metrics namely magnitude response error, linear phase response error and order of the filter obtained are used for comparing the results of proposed approach with other approaches in literature.

5.1.1 Magnitude response error

Magnitude response error of a filter is the amount of deviation from ideal magnitude response. Ideally magnitude response error should be zero. Hence MOPSO approach minimizes Magnitude response error and it can be used as a metric to compare the results of IIR filter design. Mathematically it has been defined in equation (4.4) of chapter 4.

5.1.2 Linear phase response error

Linear phase response error is the measure of non linearity in phase response of the filter. One of the major drawbacks of conventional IIR filters is their non linear phase. MOEA based approaches minimize linear phase response error to ensure linear phase in IIR filter. Mathematical definition is discussed in equations (4.5) to (4.7) .

5.1.3 Order of the filter

If i are the number of first order blocks and j are the number of second order blocks in the transfer function, then order of the filter is given by

$$\text{Order} = i+2j \tag{5.1}$$

An ideal filter design requires minimum filter order. Hence the approach which minimizes the filter order satisfying other design requirements is considered a better method.

5.2 Results of Butterworth filters

This section discusses the results of Butterworth design method for IIR filters. Butterworth filters do not have ripples in the passband and stopband. Magnitude response for all four filter types i.e. low pass, high pass, band pass and band stop are shown in Fig 5.1. Design criteria used is same as that for the proposed approach.

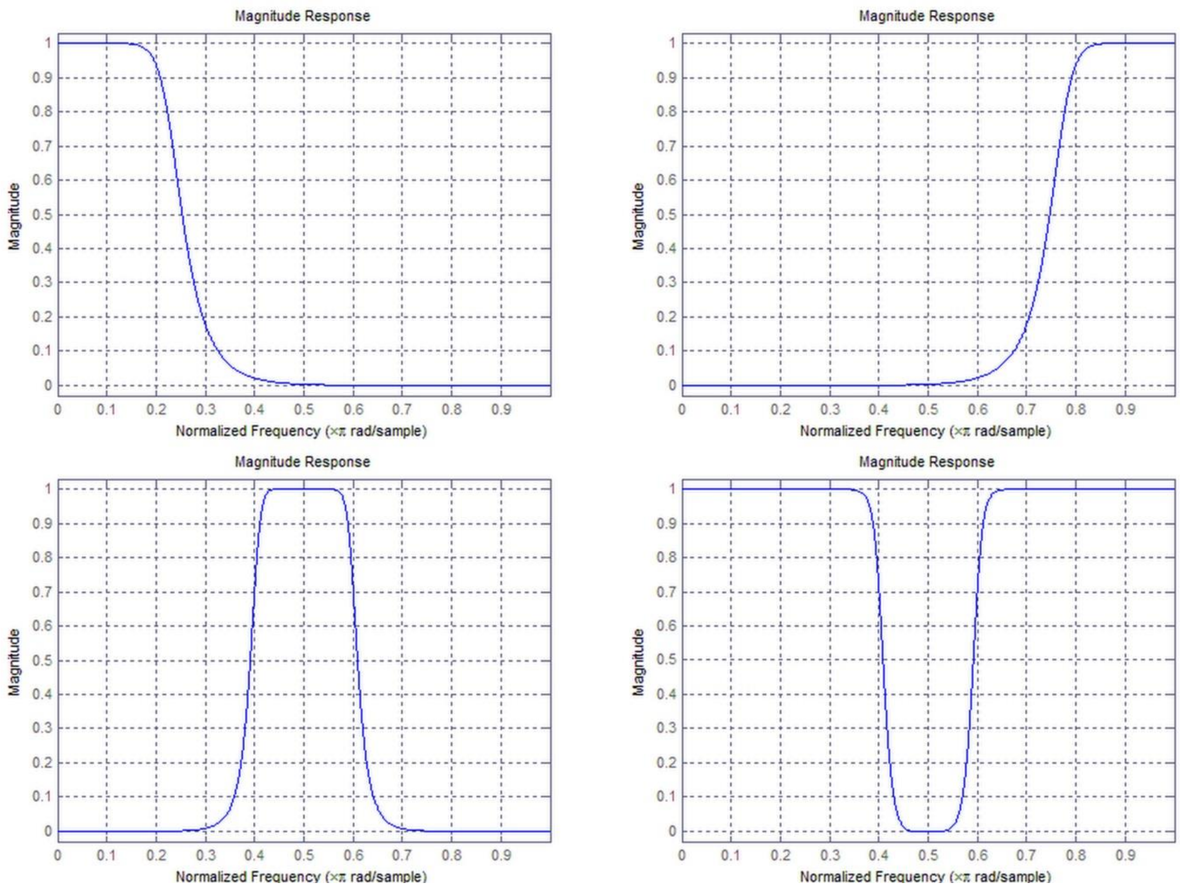

Fig 5.1: Butterworth filters a) Low Pass b) High pass c) Band pass d) Band stop

It can be observed from fig 5.1 that Butterworth filters do not have ripples in passband as well as stopband. Magnitude response is flat but with a tradeoff in transition width.

5.3 Results of Chebyshev I filters

This section discusses the results of Chebyshev I design method for IIR filters. Chebyshev I filters have the property of presence of ripples in passband only. They do not contain ripples in stopband. Magnitude response for all four filter types i.e. low pass, high pass, band pass and band stop are shown in Fig 5.2. Design criteria used is same as that for the proposed approach.

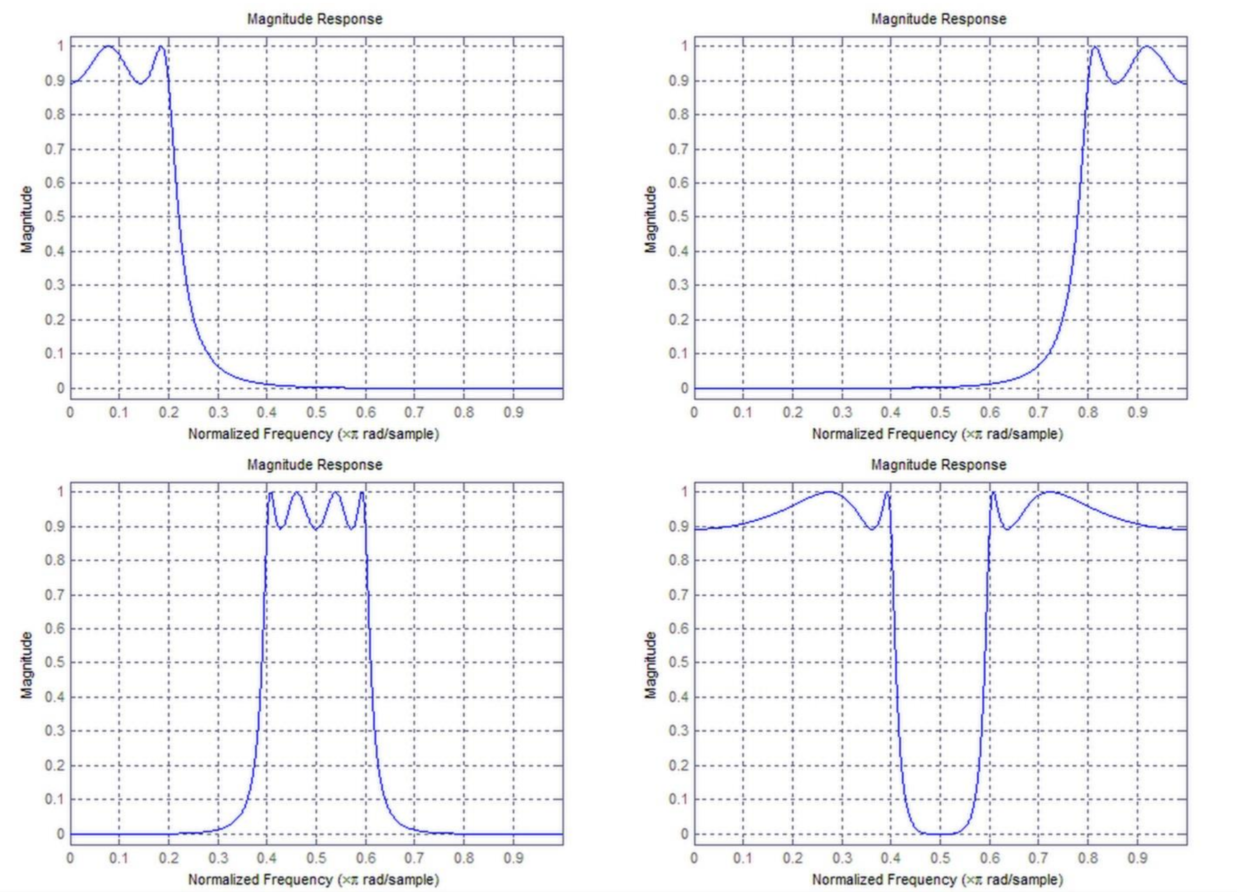

Fig 5.2: Chebyshev I filters a) Low Pass b) High pass c) Band pass d) Band stop

It can be observed from fig 5.2 that Chebyshev I filters have ripples in passband and possess flat magnitude in the stopband. There is a tradeoff between transition width and the passband ripples.

5.4 Results of Chebyshev II filters

This section discusses the results of Chebyshev II design method for IIR filters. Chebyshev II filters have the property of presence of ripples in stopband only. They do not contain ripples in passband. Magnitude response for all four filter types i.e. low pass, high pass, band pass and band stop are shown in Fig 5.3. Design criteria used is same as that for the proposed approach.

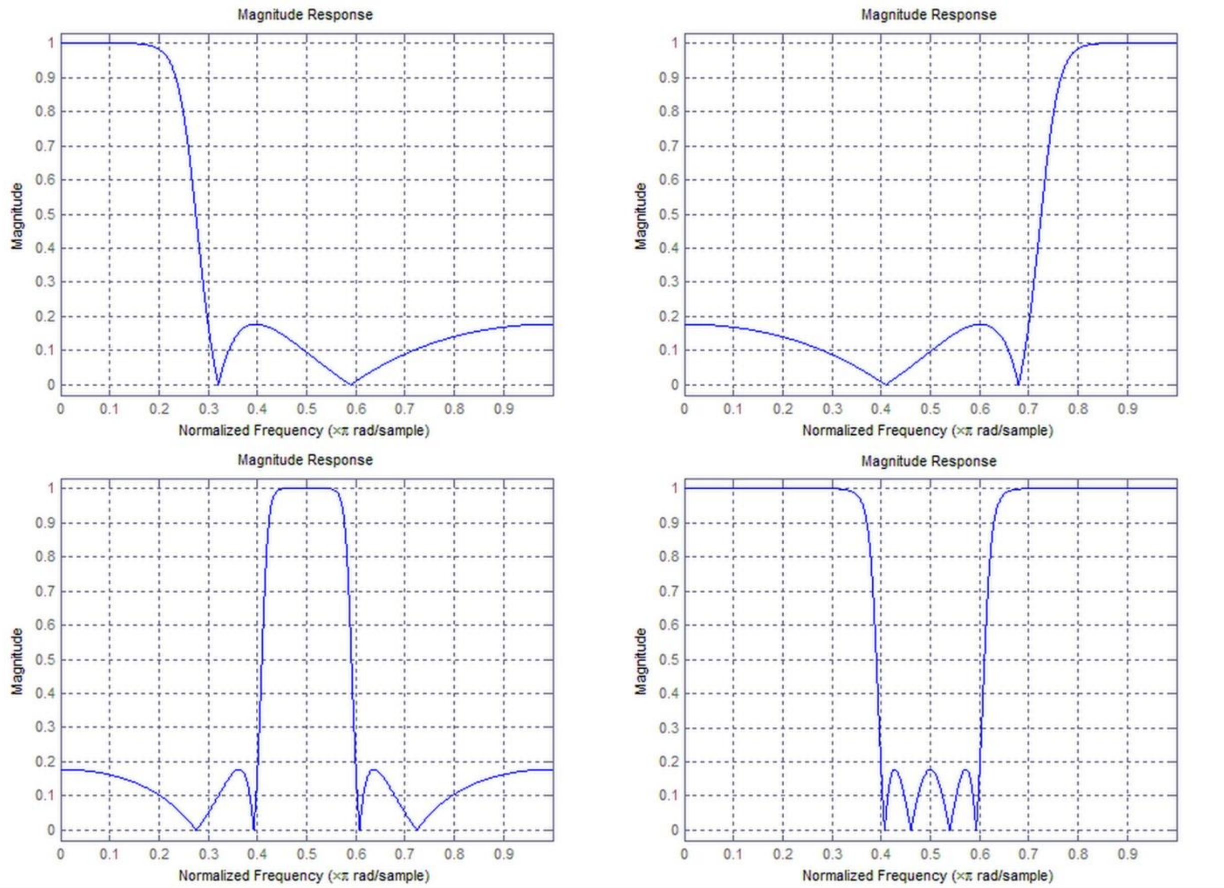

Fig 5.3: Chebyshev II filters a) Low Pass b) High pass c) Band pass d) Band stop

It can be observed from fig 5.3 that Chebyshev II filters have ripples in stopband and possess flat magnitude in the passband. There is a tradeoff between transition width and the stopband ripples.

5.5 Results of Elliptical filters

This section discusses the results of elliptical design method for IIR filters. Elliptical filters have ripples in both passband and stopband. Magnitude response for all four filter types i.e. low pass, high pass, band pass and band stop are shown in Fig 5.4. Design criteria used is same as that for the proposed approach.

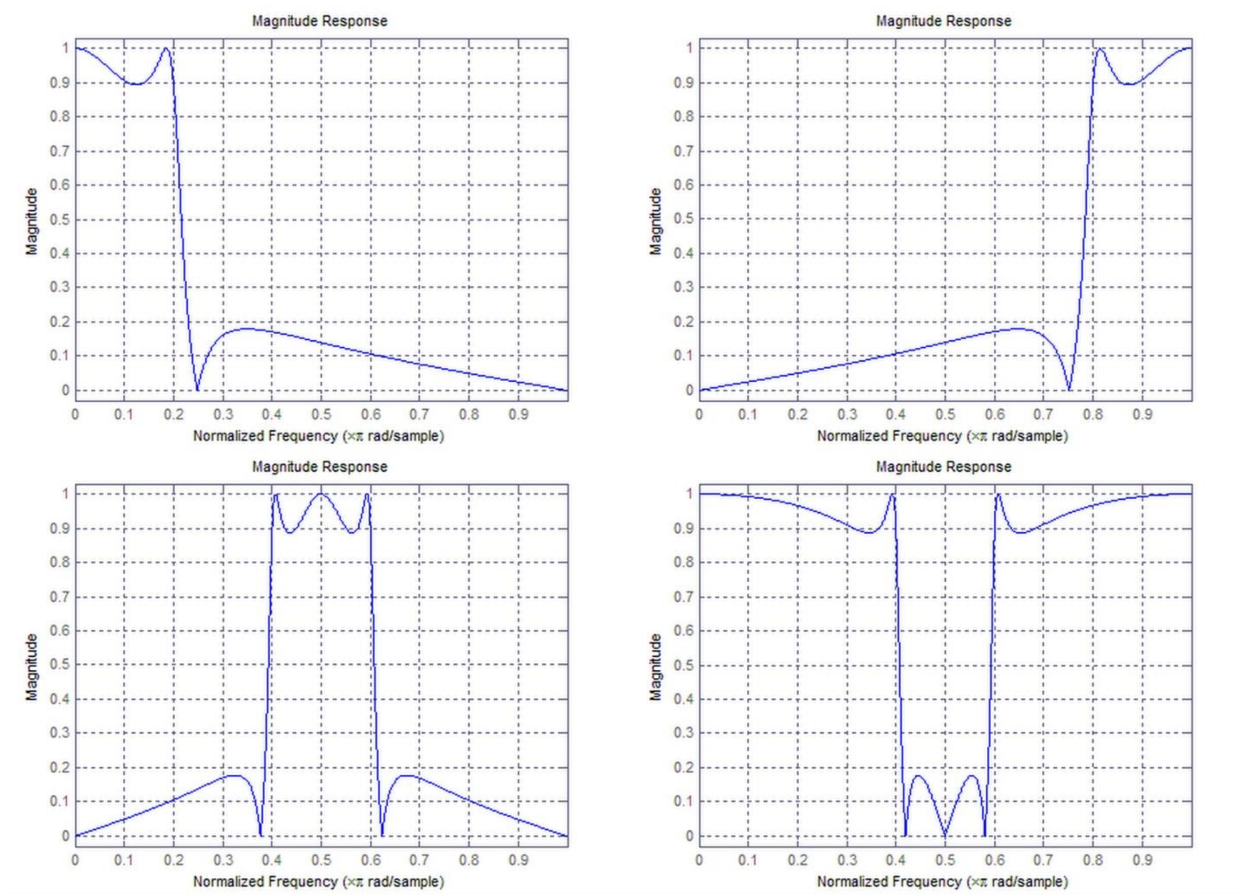

Fig 5.4: Elliptical filters a) Low Pass b) High pass c) Band pass d) Band stop

It can be observed from fig 5.4 that Elliptical filters have ripples in stopband and passband both.

All the conventional IIR filter design approaches have non linear phase characteristics. This drawback is overcome by evolutionary algorithm based approaches.

5.5 Results of Proposed approach

The proposed method aims at designing a minimum order IIR filter with linear phase characteristics and minimum magnitude response error. Algorithm's performance depends upon certain parameters like population size, number of iterations, size of repository, social and cognitive parameters etc. The approach produces better results when the population size is large and the number of iteration is more but with a corresponding increase in computation time. The outputs shown are magnitude response and phase response. These outputs are shown in figures 5.5 for low pass, high pass, band pass and band stop filter types.

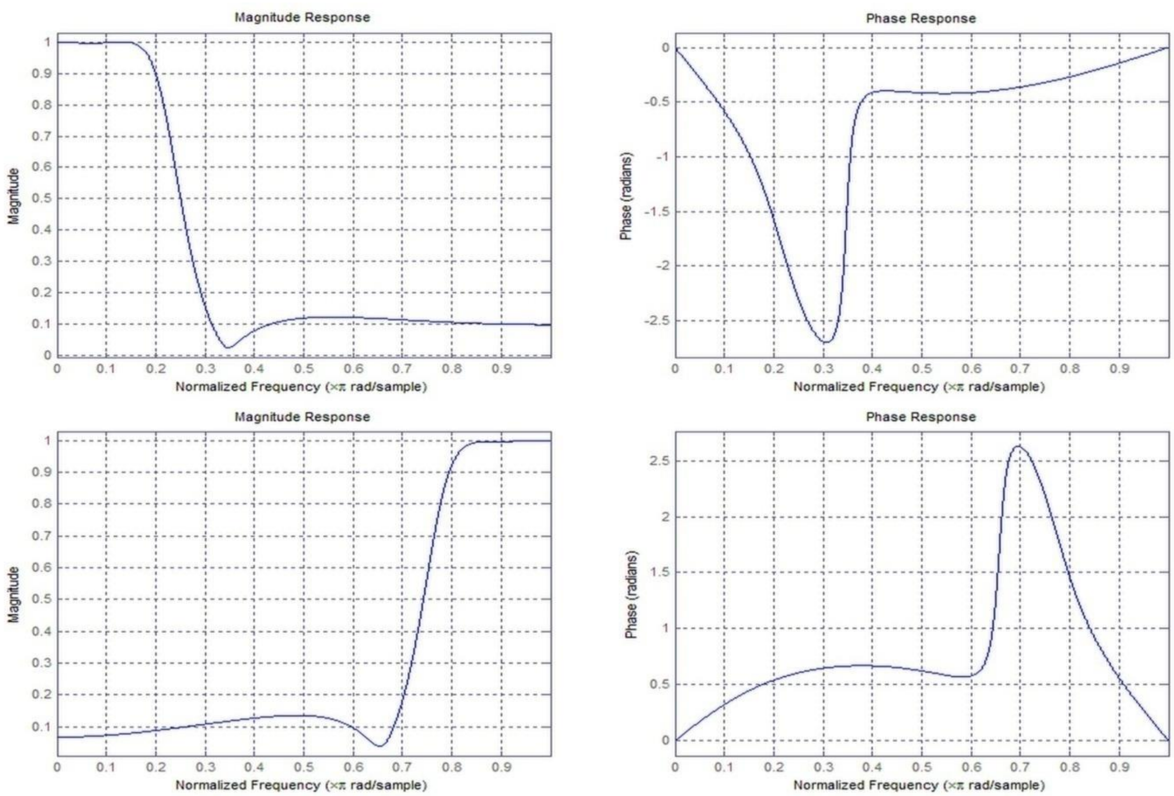

Fig 5.5: a) Magnitude response for LP b) Phase response for LP

c) Magnitude response for HP d) Phase response for HP

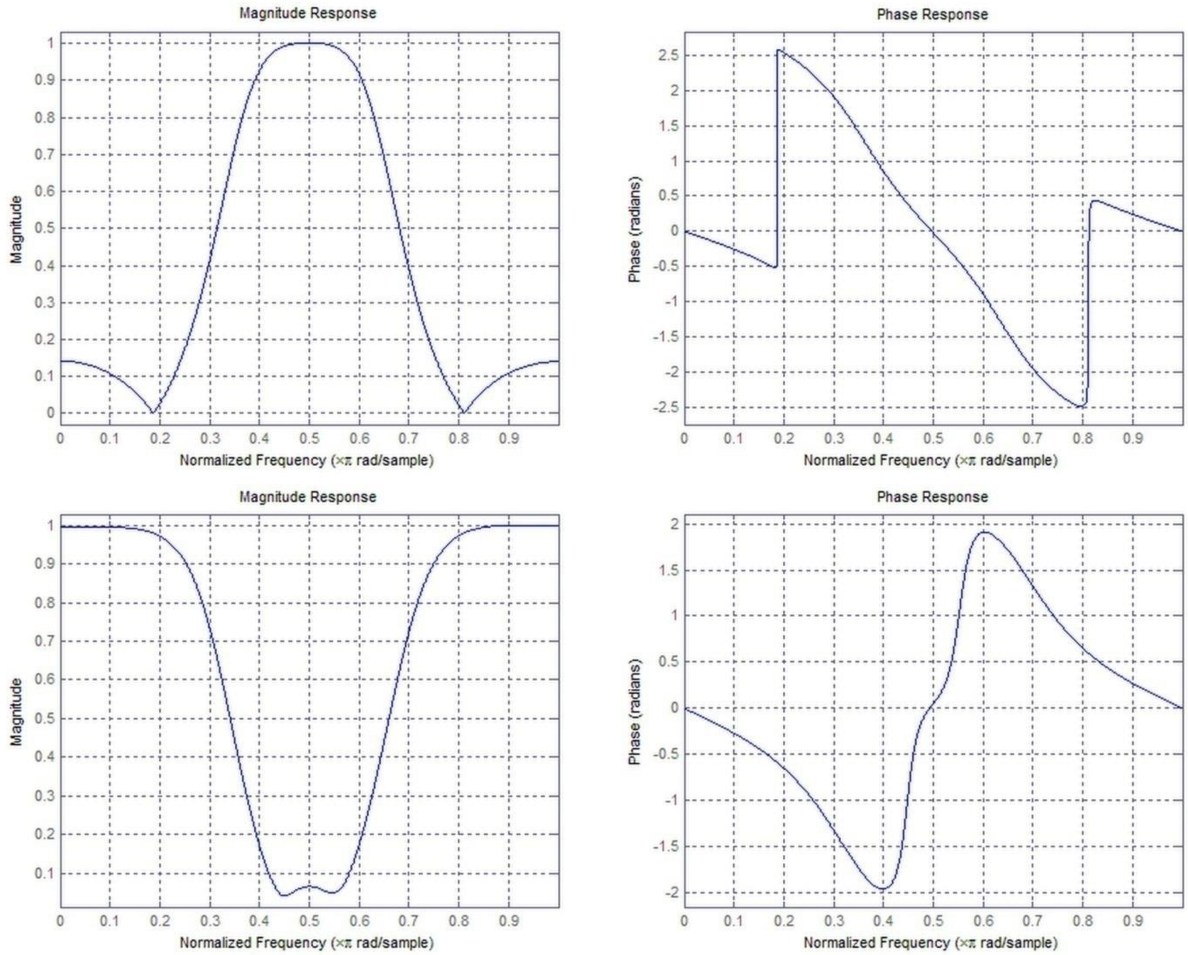

Fig 5.6: a) Magnitude response for BP b) Phase response for BP

d) Magnitude response for BS d) Phase response for BS

5.6 Comparison of proposed approach with conventional approaches

This section compares the results of conventional IIR filter design methods with the proposed approach. Comparison is made in terms of filter order and magnitude response of the filter. Since conventional approaches provide non linear phase characteristics, their phase characteristics are not shown. Table 5.1 summarizes the filter order obtained using different methods.

Table 5.1: Comparison of filter orders

Filter Type	Butterworth	Chebyshev I	Chebyshev II	Elliptical	Proposed approach
Low pass	6	4	4	3	3
High pass	6	4	4	3	3
Band pass	12	8	8	6	4
Band stop	12	8	8	6	4

Figure 5.6-5.9 compare the magnitude response obtained using different approaches. Separate comparisons are shown for LP, HP, BP and BS types.

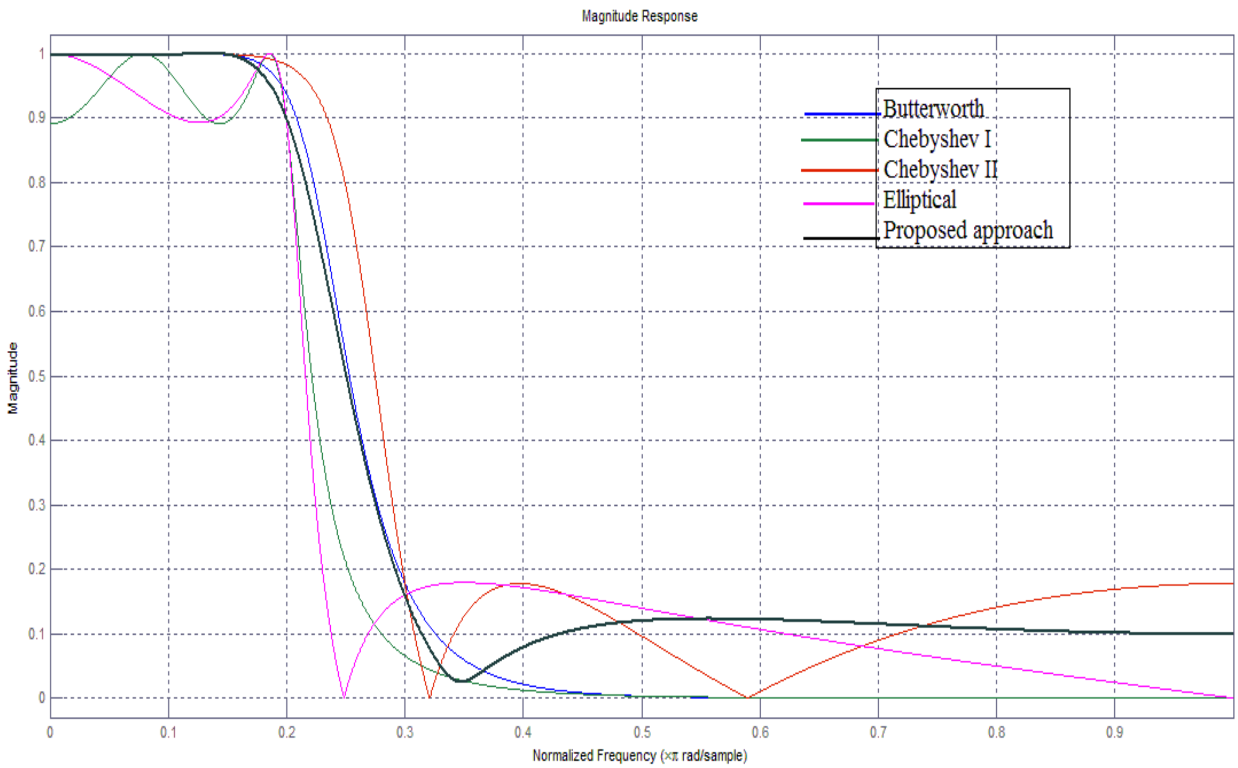

Fig 5.6: Comparison of different methods for low pass filter design

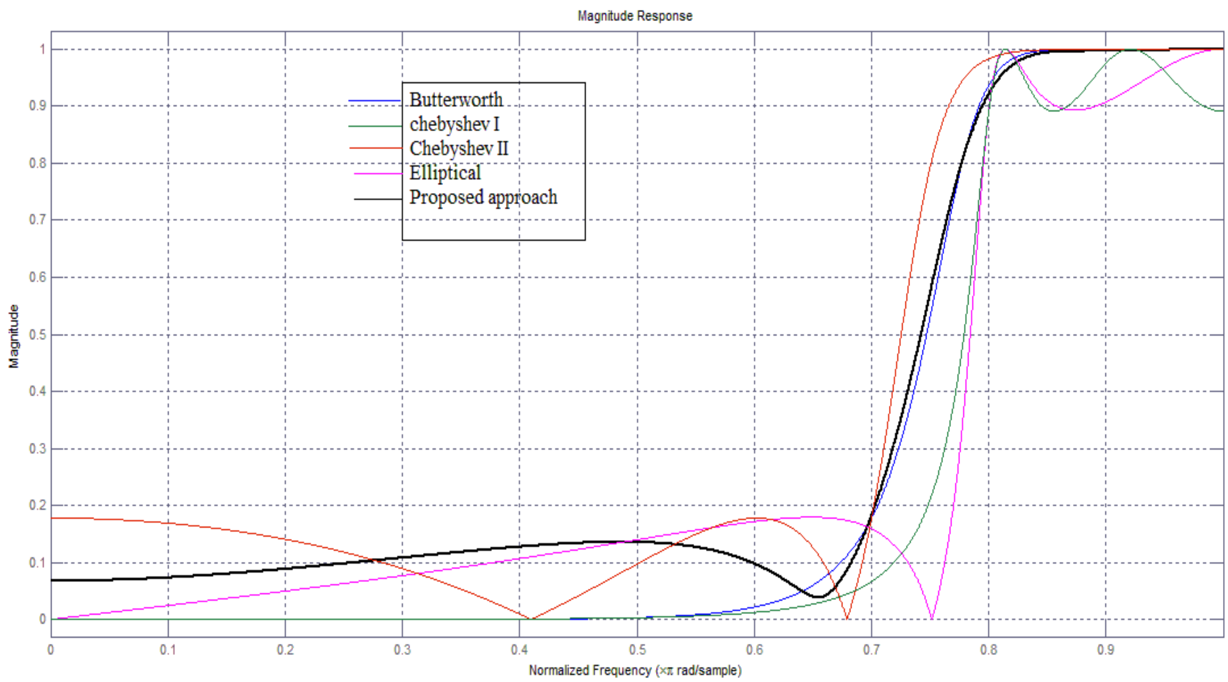

Fig 5.7: Comparison of different methods for high pass filter design

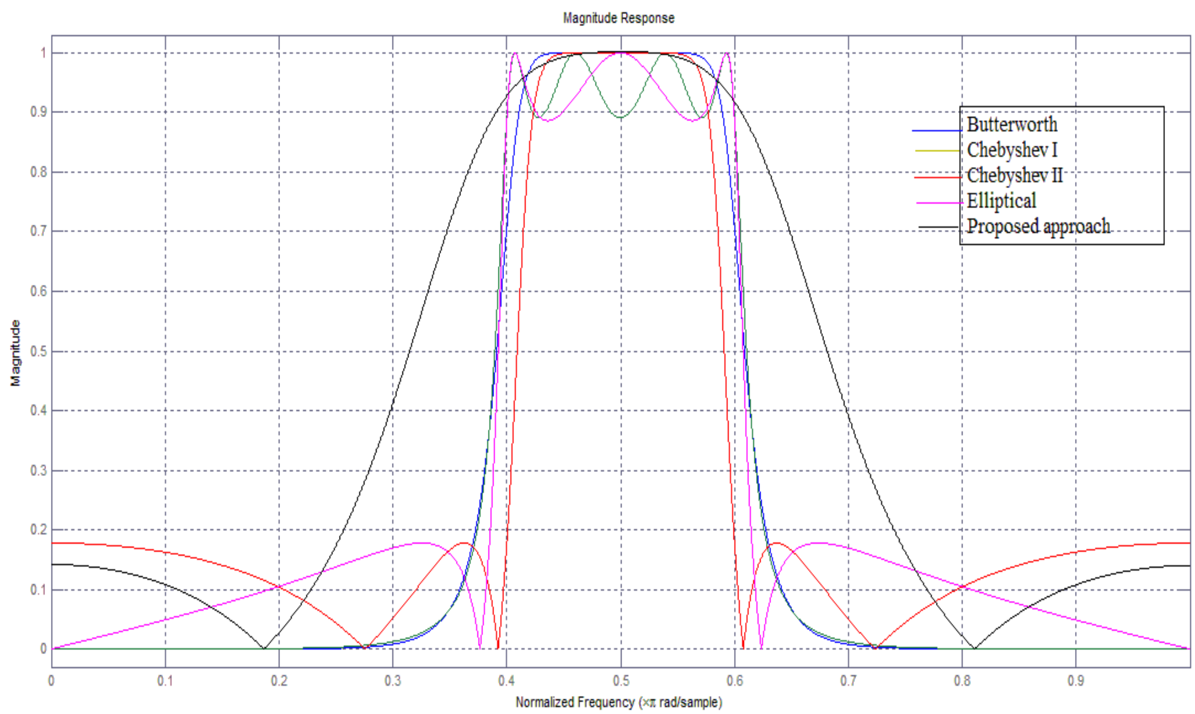

Fig 5.8: Comparison of different methods for band pass filter design

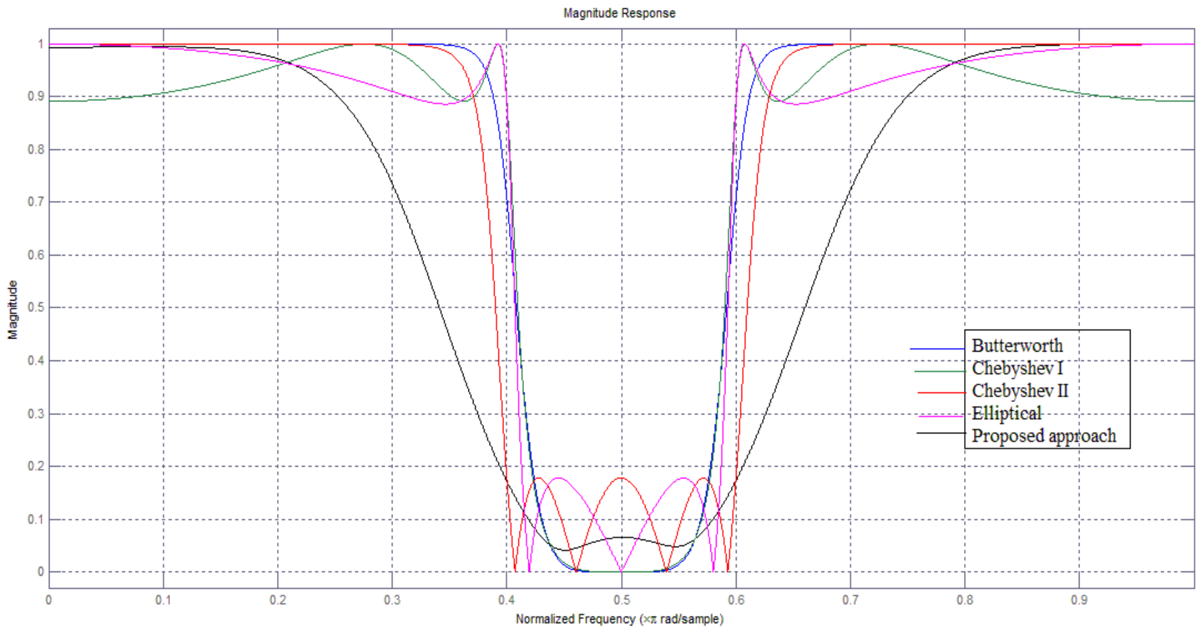

Fig 5.9: Comparison of different methods for band stop filter design

From figure 5.6-5.9 it can be summarized that in terms of magnitude response proposed approach performs best. This is because other filter types have ripples either in passband or in stopband or in both. Reducing the transition width may result in increased order of the filter. Thus keeping the order minimum, proposed approach shows minimum ripples.

5.7 Comparison of proposed approach with Genetic based approaches

This section compares the results of Genetic algorithms based approaches for IIR filter design with the proposed approach. Comparison is made in terms of filter order, magnitude response of the filter and linear phase response error of the filter. Results of proposed approach are compared with Hierarchical genetic algorithm (HGA) based IIR filter design [9] and Cooperative coevolutionary GA based IIR filter design [10] methods. Table 5.2-5.5 summarizes the performance measures for different filter types.

Table 5.2: Performance comparison for low pass filter

Approach	Lowest filter order	Passband magnitude response performance	Stopband magnitude response performance	Phase response error
HGA	3	$0.8862 \leq \lvert H(e^{j\omega}) \rvert \leq 1$	$\lvert H(e^{j\omega}) \rvert \leq 0.1800$	1.685E-04
CCGA	3	$0.9034 \leq \lvert H(e^{j\omega}) \rvert \leq 1$	$\lvert H(e^{j\omega}) \rvert \leq 0.1669$	1.474E-04
MOPSO	3	$0.9081 \leq \lvert H(e^{j\omega}) \rvert \leq 1$	$\lvert H(e^{j\omega}) \rvert \leq 0.1584$	1.0963E-04

Table 5.3: Performance comparison for high pass filter

Approach	Lowest filter order	Passband magnitude response performance	Stopband magnitude response performance	Phase response error
HGA	3	$0.9221 \leq \lvert H(e^{j\omega}) \rvert \leq 1$	$\lvert H(e^{j\omega}) \rvert \leq 0.1819$	1.1212E-04
CCGA	3	$0.9044 \leq \lvert H(e^{j\omega}) \rvert \leq 1$	$\lvert H(e^{j\omega}) \rvert \leq 0.1749$	9.7746E-05
MOPSO	3	$0.9001 \leq \lvert H(e^{j\omega}) \rvert \leq 1$	$\lvert H(e^{j\omega}) \rvert \leq 0.1744$	9.6166E-05

Table 5.4: Performance comparison for band pass filter

Approach	Lowest filter order	Passband magnitude response performance	Stopband magnitude response performance	Phase response error
HGA	6	$0.8956 \leq \lvert H(e^{j\omega}) \rvert \leq 1$	$\lvert H(e^{j\omega}) \rvert \leq 0.1772$	1.1222E-04
CCGA	4	$0.8920 \leq \lvert H(e^{j\omega}) \rvert \leq 1$	$\lvert H(e^{j\omega}) \rvert \leq 0.1654$	8.1751E-05
MOPSO	4	$0.9291 \leq \lvert H(e^{j\omega}) \rvert \leq 1$	$\lvert H(e^{j\omega}) \rvert \leq 0.1739$	6.1850E-05

Table 5.5: Performance comparison for band stop filter

Approach	Lowest filter order	Passband magnitude response performance	Stopband magnitude response performance	Phase response error
HGA	4	$0.8920 \leq \lvert H(e^{j\omega}) \rvert \leq 1$	$\lvert H(e^{j\omega}) \rvert \leq 0.1726$	2.7074E-04
CCGA	4	$0.8966 \leq \lvert H(e^{j\omega}) \rvert \leq 1$	$\lvert H(e^{j\omega}) \rvert \leq 0.1733$	1.6198E-04
MOPSO	4	$0.8968 \leq \lvert H(e^{j\omega}) \rvert \leq 1$	$\lvert H(e^{j\omega}) \rvert \leq 0.1728$	1.6034E-04

It can be seen from the results that the proposed design method can fully satisfy the magnitude response requirement, minimize the phase response error, and find the lowest order. In case of bandpass filters proposed approach provides lesser order than HGA based design. Also it can be concluded from phase response error value that proposed approach provides more linear phase than HGA or CCGA based approaches for all the filter types. Hence better filter designs are obtained using proposed approach for low pass, high pass, band pass and band stop filter types.

Chapter 6: CONCLUSION

In this thesis, a novel approach for IIR filter design using MOPSO has been introduced. In contrast to other existing MOEAs for IIR filter design PSO focuses on a directed search in promising areas of the search space. Genetic Algorithms have been used in past for IIR filter design. Main advantage of using PSO instead of genetic algorithms is its fast convergence rate. This increase in rate is due to directed nature of PSO. GAs is based on operators like mutation and selection which are random in nature. IIR filters require minimum magnitude response error, linear phase and minimum order. Filters are designed for all four types namely Low pass (LP), High pass (HP), Band pass (BP) and band stop (BS). MOPSO used in this thesis is based on concept of Pareto dominance. Pareto dominance based approaches result in a set of solutions in single simulation run of the algorithm. This is in contrast to single objective methods which give one solution per run of algorithm. Pareto optimal set constitutes the set of solutions obtained using MOEA. One of these solutions is then selected based on the application of the filter. Results obtained using proposed approach are compared with conventional approaches. Conventional approaches include Butterworth, Chebyshev I, Chebyshev II, Elliptical filters. MATLAB filter design toolbox is used to implement these conventional approaches with minimum filter order. Results are compared with the proposed approach. It can be concluded from the comparison that MOPSO achieves minimum filter order, magnitude response error and more phase linearity. The performance of the algorithm depends on certain parameters like swarm size, number of iterations, social and cognitive parameters. These are obtained experimentally and can be changed for certain applications.

The future scope includes exploring other evolutionary algorithms with multiple objective solving capabilities for the filter design. Other MOEAs may include Multi objective Bacterial foraging algorithm .Also filter structure can be made dynamic to ensure adaptability in the filter design process.

References

[1] R. W. Daniels, “*Approximation Methods for Electronic Filter Design”*. New York: McGraw-Hill, 1974.

[2] E. A. Guillemin, “*Synthesis of Passive Networks”*. NewYork: Wiley, 1957.

[3] H. Y.-F. Lam, “*Analog and Digital Filters: Design and Realization”.* Englewood Cliffs, NJ: Prentice-Hall, 1979.

[4] L. Weinberg, “*Network Analysis and Synthesis”*, Huntington, NY: Kreiger, 1975.

[5] M. D. Lutovac, D. V. Tosic, and B. L. Evans, “*Filter Design for Signal Processing Using Matlab and Mathematica”*. Upper Saddle River, NJ: Prentice-Hall, 2001.

[6] Z. M. Hussain, A. Z. Sadik, P. O’Shea, “*Digital Signal Processing- An Introduction with MATLAB Applications”*, Springer Verlang, 2011.

[7] C. A. Coello Coello, D. A. Van Veldhuizen, and G. B. Lamont, “ *Evolutionary Algorithms for Solving Multi-Objective Problems”*. Norwell, MA: Kluwer, 2002, ISBN 0-3064-6762-3.

[8] D. M. Etter, M. J. Hicks, and K. H. Cho, “*Recursive adaptive filter design using an adaptive genetic algorithm,*” in *Proc. IEEE Int. Conf. ASSP*, 1982, pp. 635–638.

[9] K.S. Tang, K.F. Man, S. Kwong, Z.F. Liu,*”Design and optimization of IIR filter structure using hierarchical genetic algorithms,”* IEEE Trans. Ind. Electron. 45 (3) (1998) 481–487, June.

[10] Yang Yu, Yu Xinjie, *“Cooperative coevolutionary genetic algorithm for digital IIR filter design*”, IEEE Trans. Ind. Electron. 54 (3) (2007), 1311–1318, June.

[11] Eberhart, R. C. & Shi, Y. (1998),”*Comparison between genetic algorithms and particle swarm optimization”.* In V. W. Porto et al. (Eds.), *Evolutionary programming: Vol. VII* (pp. 611-616),Springer.

[12] Lecture *,”DT Filter Design: IIR Filters”,* OpenCourseWare, 2006, Discrete-Time Signal Processing Massachusetts Institute of Technology Department of Electrical Engineering and Computer Science .

[13] J.J. Shynk, *“Adaptive IIR filtering”,* IEEE ASSP Mag. 6 (2) (1989) 4–21, April.

[14] V. K. Ingle and J. G. Proakis, *“Digital Signal Processing Using MATLAB,”* Thomson Books, New Delhi, 2004.

[15] A.F. Vazquez and G.J. Dolecek *"A New Method for the Design of IIR Filters With Flat Magnitude Response",* IEEE Transactions on circuits and systems, regular papers, vol. 53, no. 8, august 2006

[16] R. A. Losada and V. Pellissier, "*Designing IIR filters with a given 3-dB point,"* IEEE Signal Proc. Magazine; DSP Tips & Tricks column, vol. 22, pp. 95–98, July 2005.

[17] Samarjeet Singh, Uma Sharma *" MATLAB Based Digital IIR Filter Design ",*IJECE, ISSN 2277 - 1956, 2012/01/PP - 74 - 83.

[18] E.C. Ifeachor, B.W. Jervis*, "Digital Signal Processing: A Practical Approach,"* Pearson Education Ltd. 2002.

[19] J. G. Proakis and D. G. Manolakis, "*Digital Signal Proc-essing: Principles, Algorithms, and Applications,"* 4th Edition, Pearson Education, Inc., New Delhi, 2007.

[20] E. B. Hogenauer, "*An economical class of digital filters for decimation and interpolation,"* IEEE Trans. on Acoust. Speech and Signal Proc., vol. ASSP-29, pp. 155–162, April 1981.

[21] J. Kennedy, R. Eberhart, "*Particle Swarm Optimization,"* in Proc. IEEE int. Conf. On Neural Network, 1995.

[22] Hu, X. & Eberhart, R. (2002), *"Multi-objective optimization using dynamic neighborhood particle swarm optimization",* Proceedings of the 2002 IEEE Congress Evolutionary Compututation (pp 1677-1681). IEEE.

[23] J. Knowles and D. Corne, *"The Pareto archived evolution strategy: A new baseline algorithm for multiobjective optimization",* Proceedings of the 1999 Congress on Evolutionary Computation. Piscataway, NJ: IEEE Press, 1999, pp. 98-105.

[24] Carlos A. Coello Coello, Gary B. Lamont, "*Applications of multi-objective evolutionary algorithms",* Advances in natural computation, vol-1, Dec 2004.

[25] Deb. K, *"Multi-objective Optimization using Evolutionary Algorithms",* Chichester, U.K. Wiley, 2001

[26] Carlos A. Coello Coello, Gary B. Lamont, David A. Van Veldhuizen, *" Evolutionary algo-rithms for solving multi-objective problems"*, 2nd ed., 2007.

[27] Jin, Y., Olhofer, M., & Sendhoff, B. (2001). *"Evolutionary dynamic weighted aggregation for multi-objective optimization: Why does it work and how? "*, Proceedings of the GECCO 2001 Conference (pp. 1042-1049), San Francisco, CA

[28] Fonseca, C. M. & Fleming, P. J. (1993), "*Genetic algorithms for multi-objective optimization: Formulation, discussion and generalization*", Proceedings of the 5th International Conference on Genetic Algorithms (pp. 416-423).

[29] Srinivas, N. & Deb, K. (1994*), "Multi-objective optimization using nondominated sorting in genetic algorithms",* Evolutionary Computation, 2(3), 221-248.

[30] Srinivasan, D. & Seow, T. H. (2003),*"Particle swarm inspired evolutionary algorithm (PSEA) for multi-objective optimization problem",* Proceedings of the IEEE 2003 Congress on Evolutionary Computation (pp. 2292-2297). IEEE Press.

[31] Carlos A. Coello, Gregorio Toscano Pulido, and Maximino Salazar Lechuga, *"Handling multiple objectives with particle swarm optimization"* ,Evol. Comput., vol. 8, pp. 256-279, NO. 3, June 2004

[32] J. D. Knowles and D.W. Corne, *"Approximating the nondominated front using the Pareto archived evolution strategy",* Evol. Comput., vol. 8, pp. 149-172, 2000.