9-12-2019

Digital Filter Design Using Improved Teaching-Learning-Based Optimization

Miao Zhang
University of Windsor

Digital Filter Design Using Improved Teaching-Learning-Based Optimization

By

Miao Zhang

A Dissertation
Submitted to the Faculty of Graduate Studies
through the Department of Electrical and Computer Engineering
in Partial Fulfillment of the Requirements for
the Degree of Doctor of Philosophy
at the University of Windsor

Windsor, Ontario, Canada

2019

Digital filter design using improved teaching-learning-based optimization

by

Miao Zhang

APPROVED BY:

W. Zhu, External Examiner
Concordia University

Z. Kobti
School of Computer Science

C. Chen
Department of Electrical and Computer Engineering

H. Wu
Department of Electrical and Computer Engineering

H. K. Kwan, Advisor
Department of Electrical and Computer Engineering

September 12, 2019

DECLARATION OF ORIGINALITY

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

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

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

ABSTRACT

Digital filters are an important part of digital signal processing systems. Digital filters are divided into finite impulse response (FIR) digital filters and infinite impulse response (IIR) digital filters according to the length of their impulse responses. An FIR digital filter is easier to implement than an IIR digital filter because of its linear phase and stability properties. In terms of the stability of an IIR digital filter, the poles generated in the denominator are subject to stability constraints. In addition, a digital filter can be categorized as one-dimensional or multi-dimensional digital filters according to the dimensions of the signal to be processed. However, for the design of IIR digital filters, traditional design methods have the disadvantages of easy to fall into a local optimum and slow convergence.

The Teaching-Learning-Based optimization (TLBO) algorithm has been proven beneficial in a wide range of engineering applications. To this end, this dissertation focusses on using TLBO and its improved algorithms to design five types of digital filters, which include linear phase FIR digital filters, multiobjective general FIR digital filters, multiobjective IIR digital filters, two-dimensional (2-D) linear phase FIR digital filters, and 2-D nonlinear phase FIR digital filters. Among them, linear phase FIR digital filters, 2-D linear phase FIR digital filters, and 2-D nonlinear phase FIR digital filters use single-objective type of TLBO algorithms to optimize; multiobjective general FIR digital filters use multiobjective non-dominated TLBO (MOTLBO) algorithm to optimize; and multiobjective IIR digital filters use MOTLBO with Euclidean distance to optimize. The design results of the five types of filter designs are compared to those obtained by other state-of-the-art design methods. In this dissertation, two major improvements are proposed to enhance the performance of the standard TLBO algorithm. The first improvement is to apply a gradient-based learning to replace the TLBO learner phase to reduce approximation error(s) and CPU time without sacrificing design accuracy for linear phase FIR digital filter design. The second improvement is to incorporate Manhattan distance to simplify the procedure of the multiobjective non-dominated TLBO (MOTLBO) algorithm for general FIR digital filter design. The design results

obtained by the two improvements have demonstrated their efficiency and effectiveness.

ACKNOWLEDGEMENTS

I would first like to thank my supervisor, Dr. H. K. Kwan, for suggesting the research topics, the TLBO algorithm, and their design improvements, as well as for editing the write-up of my dissertation.

I would also like to acknowledge Dr. Ziad Kobti, Dr. Chunhong Chen and Dr. Huapeng Wu for their comments on my dissertation. I am grateful to Dr. Weiping Zhu, University of Concordia, for serving as my External Examiner.

I would like to thank my fellow graduate student Jiajun Liang for discussions.

Finally, I would like to thank my parents Baojian Zhang and Xiurong Liu, and my friend Heling Shao, for their support and selfless love.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS/SYMBOLS

FIR	Finite Impulse Response
IIR	Infinite Impulse Response
TLBO	Teaching-Learning-Based Optimization
MO	Multiobjective Optimization
MOTLBO	Multiobjective Teaching-Learning-Based Optimization
PSO	Particle Swarm Optimization
MOPSO	Multiobjective Particle Swarm Optimization
2-D	Two-dimensional
MM	Minimax
LS	Least Squares
WLS	Weighted Least Square
HT	Hilbert Transformer
LP	Lowpass
HP	Highpass
BP	Bandpass
BS	Bandstop
PB	Passband
SB	Stopband
MME	Maximum Magnitude Error
MSCE	Mean Squared Complex Error

GA	Genetic algorithm
DE	Differential Evolution
ABC	Artificial Bee Colony
ACO	Ant Colony Optimization
CSA	Cuckoo Search Algorithm
HS	Harmony Search
CSO	Cat Swarm Optimization
LS-MOEA	Local Search Operator Enhanced Multiobjective Evolutionary Algorithm
MOCSO-DE	Multiobjective Cat Swarm and Differential Evolution Algorithm
CLS	Constrained Least Square

CHAPTER 1
Introduction

Signal processing utilizes a variety of analog filters and digital filters. One characteristic of analog signals is time-continuity, so the independent variable of a one-dimensional analog signal is time. Through time sampling and magnitude discretization, the one-dimensional analog signal will be converted into a one-dimensional discrete digital signal. Whereas, a digital filter is a system consisting of digital multipliers, adders, and delay units. The function of a digital filter is to perform arithmetic processing on an input discrete signal to achieve the purpose of changing the signal spectrum. Therefore, designing a digital filter involves generating a transfer function that conforms a given set of specifications. In terms of hardware cost, digital filters may be more expensive than an equivalent analog filter. But from an implementation perspective, analog filters are usually built with analog components such as capacitors and inductors, where digital filters can be implemented by software or digital components. It is troublesome to replace any capacitor or inductor when the analog filter parameters are changed. Moreover, digital filters have four key advantages over analog filters [1]-[9]:

1. Digital filters are less sensitive to the external environment and have higher reliability with respect to time and temperature, unlike analog filters.

2. Digital filters enable functions such as accurate linear phase and multi-rate processing that are not possible with analog filters.

3. Digital filters can achieve arbitrary processing precision as long as the word length is increased.

4. Digital filters are more flexible and can store signals at the same time.

1.1 FIR and IIR digital filters

It is relatively easy to understand and calculate digital filters; however, to meet practical problems and challenges, numerous studies have explored their analysis [10]-[15] and realization [16]-[23]. The design of digital filters is a deceptively complex topic. In general, there are two major types of digital filters:

finite impulse response (FIR) and infinite impulse response (IIR) digital filters. Both have advantages and insufficiencies.

FIR filters have six key advantages over IIR filters [2]:

1. The transfer function of an FIR filter contains only zeros. An FIR filter with symmetric coefficients is guaranteed to provide a linear phase response, which can be critical in some applications.

2. Feedback is not necessary. This means that any rounding errors are not cumulative. This makes implementation simpler.

3. FIR filters are inherently stable since the output is a sum of a finite number of products of filter coefficients and input values.

4. FIR filters can easily be designed to be linear phase by making its filter coefficients symmetric.

5. FIR filters can implement linear-phase filtering. This means that the filters have no nonlinear phase shift across the frequency band. The lack of phase/delay distortion can be a critical advantage of FIR filters over IIR and analog filters in certain systems, such as digital data modems.
6. FIR filters can be used to correct frequency-response errors in a loudspeaker to a finer degree of precision than using IIR filters.

1.1.1 FIR Filters

FIR filter design can be categorized with two types: linear phase FIR filter design [24]-[28] and non-linear phase (general) FIR filter design [29]-[34]. According to the filter length *M* and the symmetry of impulse responses, there are four types of linear phase FIR filters [1], [5]:

1. Type I Filters

Type I FIR filters, featured by odd M value and even symmetry, are fairly universal and most versatile, but they cannot be used whenever a 90 degrees phase shift is necessary, as is the case in differentiators and Hilbert transformers.

2. Type II Filters

Type II FIR filters, featured by even M value and even symmetry and would normally not suitable for highpass or bandstop filters because their frequency response is always 0 at $\omega = \pi$. In addition, they cannot be used for applications where a 90° phase shift is necessary.

3. Type III Filters

Type III FIR filters utilize odd M value and odd symmetry and cannot be used for standard frequency filters because, in these cases, the 90° phase shift is usually undesirable. The frequency response is always 0 at $\omega = 0$ and $\omega = \pi$. They can be used for designing differentiators and Hilbert transformers.

4. Type IV Filters

Type IV FIR filters, featured by even M value and odd symmetry, cannot be used for some standard frequency-selective filters for the same reasons that type III filters cannot be used. They are well suited for differentiators and Hilbert transformers, and their magnitude approximations are usually better than type III filters because their magnitude errors are smaller. This is due to the fact that their frequency response is always 0 at $\omega = 0$.

In some applications an integer group delay is desirable. In these cases, type I or type III filters are preferred. On the other hand, for general FIR digital filters, their coefficients are not symmetrical, and their frequency responses are arbitrary. General FIR digital filters are not exactly linear phase but can be used to approximate linear phase. In this case, the impulse response will not be exactly symmetric. The minimum phase FIR filters are not appropriate for this problem as there is no phase approximation at all. FIR digital filters with nonlinear phase responses are useful in applications in which linear phase is not a requirement.

With the use of symmetric or asymmetric coefficients, FIR filters are guaranteed to be of linear phase. An FIR digital filter is always stable due to the absence of a denominator in its transfer function. The group delay of a linear phase FIR filter with M length is (M-1)/2, which is constant for all frequencies. This filter can always be designed as linear phase no matter its impulse responses are symmetrical or anti-symmetrical. Linear phase means that all frequency components of the input signal experience the same delay such that there are no phase distortion. For example, assuming an input signal is located within the passband of a lowpass FIR filter, and the corresponding output signal will be approximately equal to the input signal delayed by the group delay of the filter, which is just a shifted version of the input signal. Nevertheless, there is no linear phase response in a general FIR filter because the group delay is a function of frequency, the details of which will be clearer in the later chapters. The success of designing general FIR filters depends greatly on the method that is used to design their filter coefficients. In general, designing a general FIR filter to meet a set of given specification requires more computational time and a higher implementation cost.

1.1.2 IIR Filters

IIR filters are the most efficient type of digital filters with respect to implementing digital signal processing (DSP). IIR digital filters are more advantageous than FIR filters because of their implementation efficiency, which is necessary to meet specifications related to passband ripple and stopband attenuation. IIR digital filters can be designed [35]-[52] to meet the same specifications with a lower order than that of FIR digital filters. This is especially useful for implementation using a small signal processor, in which the number of calculations per time step is less than that of a FIR filter. This results in a savings of computational time.

IIR filters can be realized in direct-form and cascade-form [1]-[9]. A direct-form filter has a simple structure in which the filter coefficients are obtained directly from the designed coefficients of the transfer function. A cascade-form filter expresses each of its numerator and the denominator transfer functions as a product of lower-order polynomials. Thus, a cascade-form digital filter can be realized as a cascade of low-order filter sections.

The transfer function of a direct-form IIR filter can be expressed as follows:

$$H(z) = \frac{B(z)}{A(z)} = \frac{\sum_{m=0}^{M} b_m z^{-m}}{1+\sum_{n=0}^{N} a_n z^{-n}} \tag{1.1}$$

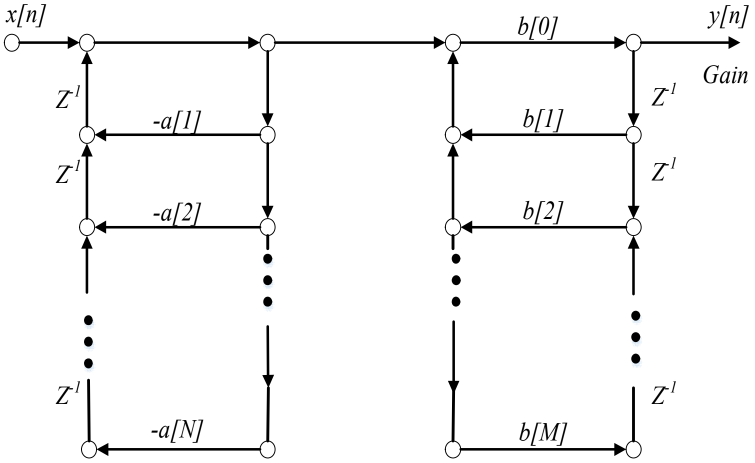


Fig.1.1 Structure of direct-form I IIR digital filter

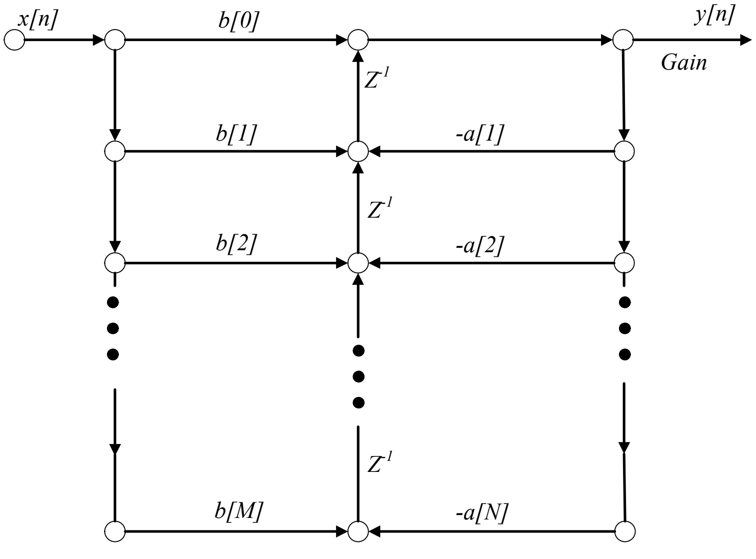


Fig. 1.2 Structure of direct-form II IIR digital filter

The transfer function of a cascade-form IIR filter can be expressed as follows:

$$H(z) = b_0 \prod_{n=1}^{\frac{N}{2}} \frac{B_n(z)}{A_n(z)} = b_0 \prod_{n=1}^{\frac{N}{2}} \frac{(1+b_{1n}z^{-1}+b_{2n}z^{-2})}{(1+a_{1n}z^{-1}+a_{2n}z^{-2})} \qquad (1.2)$$

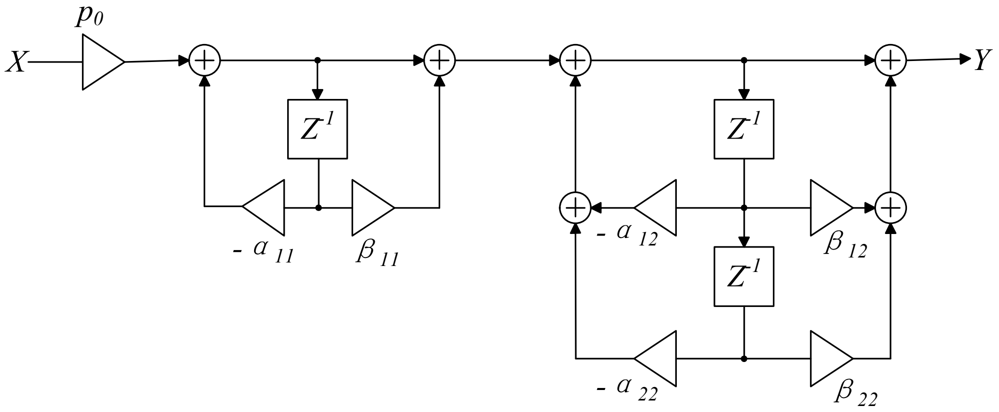


Fig. 1.3 Structure of a cascade-form IIR digital filter (3rd order)

Digital filters can be designed for different applications such as lowpass, highpass, bandpass, bandstop, delay equalizer, digital differentiator, digital integrator and digital Hilbert transformer. In the following chapters, different digital filter designs are described.

1.2 Two dimensional digital filters

Digital filter can be divided into one-dimensional (1-D), two-dimensional (2-D), or multi-dimensional (m-D) digital filters according to the dimensions of a signal to be processed. The signal processed by a one-dimensional digital filter can be a sequence of sampled values of a time function. A signal processed by a 2-D or m-D digital filter is a sequence of two or more variables. For example, a 2-D discrete image consists of sampled values on a 2-D plane. 2-D digital filters are widely used in image processing, sonar signal processing, radar signal processing, geophysical signal processing, and others. With the rapid increase in the amount of data to be processed by modern electronic devices, efficient and accurate design of various types of 2-D and multi-dimensional FIR digital filters [55]-[66] and IIR digital filters [67]-[78] is of great importance in the field of multi-dimensional digital signal processing research. Unlike a 1-D digital filter design which only involves a 1-D frequency function approximation, a 2-D digital filter design is essentially a 2-D frequency function approximation problem. The 2-D frequency function approximation theory is less developed than the 1-D frequency function approximation theory. Thus, some effective 1-D digital filter design methods cannot be extended to design a 2-D digital filter. In general, 2-D digital filter design problems are significantly more complicated than that of 1-D digital filter design problems.

Design of 2-D nonlinear-phase FIR filters [64]-[66] has attracted the attention of researchers because of their relatively lower group delay which is more desirable for some applications than the corresponding 2-D linear phase FIR filters [55]-[61]. On the other hand, the benefit of 2-D linear phase FIR filters is that they are both stable and linear phase, and this allows them to find applications in digital image processing to preserve phase linearity. In general, the transfer function of a 2-D FIR digital filter cannot be decomposed, which means that the transfer function cannot usually be expressed in identical modules suitable for design and implementation. Thus, the design problem becomes more challenging. To address this challenge, [62] describes minimax and least squared method of the separable 2-D linear phase FIR filter using an iterative alternating optimization technique. In [56], a matrix-based algorithm was used for constrained least-squares (CLS) and minimax (MM) designs of quadrantally symmetric 2-D FIR filters, both of which can be formulated as an optimization problem or converted into a sequence of subproblems with a positive-definite quadratic cost and a finite number of linear constraints expressed in terms of the filter's coefficient matrix.. In [56], design examples of 2-D linear phase FIR filters include magnitude responses of rectangular-shape, circular-shape, and diamond-shape. Because it characterized a simple problem formulation, [60] used an accelerated artificial bee colony algorithm which belongs to a class of meta-heuristic algorithms to design a 2-D linear-phase FIR filter. In [60], three shapes in 2-D linear phase FIR filters' magnitude responses: circular, diamond and elliptic are designed. Likewise, the proposed transfer function in [59] is a closed-form function of 2-D linear phase FIR lowpass filters with maximally flat passband and stopband. In this context, changing filter order or flatness degree can generate various passband shapes. Additional references selected for digital filter realization and design are given in [148]-[205].

1.3 Optimization algorithms

Traditionally, numerical optimization has been the standard optimization tool [79]. In recent years, evolutionary algorithms [80] have emerged as an increasing popular optimization tool. The development of the latter was started around 1975, in which genetic algorithm (GA) [81] was proposed. The emergence and success of genetic algorithm have greatly encouraged the enthusiasm of researchers to develop nature inspired optimization algorithms. After years of development, a large number of evolutionary algorithms [80] have been developed, including genetic algorithm, ant colony optimization (ACO) algorithm, differential evolution (DE) algorithm, and particle swarm optimization (PSO) algorithm. Thus far, evolutionary algorithms have been used in a wide range of areas including machine learning, process control, economic forecasting, and engineering forecasting. The unprecedented success of evolutionary algorithms has attracted great interest from scientists in the fields of mathematics, physics, computer science, social sciences, economics, and engineering.

Swarm intelligence algorithms [82] identify questions by learning from certain life phenomena or natural phenomena. This kind of algorithm contains the characteristics of self-organization, self-learning, and self-adaptation of natural life phenomena. In the operation process, the solution space is self-organized by the obtained calculation information. In the search process, the population evolves according to the value of the fitness function set in advance by using the survival of the fittest. Consequently, the algorithm has certain intelligence. Due to the advantages of the swarm intelligence algorithms, when applying the swarm intelligence algorithm to solve a problem, it is not necessary to describe the solution problem in advance. As a result, efficient solving is possible for some complex problems.

Swarm intelligence algorithms complete the calculation by setting the corresponding evolution mechanism for population, while the individuals in the population have a certain independence, and they may or may not exchange information, and their evolution mode depends entirely on their own situation. Therefore, for swarm intelligence algorithms, an individual is encapsulated in a complete and essential parallel mechanism. If a distributed multiprocessor is used to complete the process of a swarm intelligence algorithm, the algorithm can be set with multiple populations and each individual of the populations can be placed in different processors for evolution. Moreover, certain information exchanges can be completed during the iteration information, though the exchange is not necessary. After the iteration is completed, the survival of the fittest is performed according to the fitness value. Therefore, the implicit nature of the group intelligence algorithms can make full use of the multi-processor mechanism to achieve parallel programming and improve an algorithm's ability to solve problems. Thus, it is more suitable for the background of the rapid development of distributed computing technologies, such as cloud computing.

Swarm intelligence algorithms [83] are widely used to solve problems with high computational complexity, rather than trying all the options one by one, which will take a lot of time. They are used extensively in the field of artificial intelligence. For instance, the principles behind GA are borrowed from nature itself. They are the principles of heredity and variation. Genetics are the ability of organisms to pass on their biological and evolutionary characteristics to their offspring. Because of this ability, all creatures can leave their species characteristics to their offspring. The genetic variation of the organisms guarantees the genetic diversity of the population, and the variation is random because nature cannot know in advance which features are most suitable in the future due to variations on factors such as climate change, food increase/decrease, or the emergence of competitive species. Mutations cause new traits in the organism to survive and leave behind in new, changing habitats.

Swarm intelligence algorithms usually come from the intelligent collaborative evolution of certain life or other things in nature. The simulation uses some evolutionary mechanisms to guide the population to search for the solution space. Because this kind of algorithms lacks strict mathematical theory support, the problematic solution space uses repeated the iterative probabilistic search. As a result, swarm intelligence algorithms will have problems such as premature or low solution precision. Therefore, in many cases, the barriers to solve the problem are that swarm intelligence algorithms only obtain an approximate solution to the best solution.

Swarm intelligence falls under more recent evolutionary algorithms which include differential evolution, particle swarm optimization, ant colony optimization, artificial bee colony algorithm, teaching and learning based optimization, and others. Classic evolutionary algorithms include genetic algorithms, genetic programming, and others.

1.4 Motivations of the dissertation

This dissertation concentrates on digital filter design, including FIR digital filter and IIR digital filter designs. The optimization method uses evolutionary algorithms. After experimentations and comparisons with other heuristic algorithms such as ACO, PSO, and GA, the TLBO algorithm has shown to be superior in terms of better minimization performance and faster convergence. Designing digital filters using TLBO algorithm is a new topic. Therefore, this dissertation adopts TLBO algorithm for FIR and IIR filter design.

However, when designing more complex digital filters using the TLBO, there are limitations. Therefore, improved methods are to be developed to enhance the efficiency of the TLBO algorithm. In particular, a gradient-based learning phase is proposed to replace the original learning phase to expand the search capability for digital filter design.

1.5 Organization of the dissertation

The organization of this dissertation is as follows: In Chapter 2, the literature review of swarm intelligence algorithms is presented in Sections 2.1 and 2.2. The details of TLBO algorithm and the proposed improved TLBO algorithm are explained from Sections 2.3 to 2.4. In Chapter 3, linear phase FIR filter design using TLBO and the improved TLBO algorithm are described. In Chapter 4, the multiobjective general FIR filter design using non-dominated multiobjective TLBO with crowding distance is presented. In addition, a novel multiobjective TLBO is proposed to design general FIR filters. In Chapter 5, another multiobjective is presented to design IIR digital filter. In Chapter 6, two-dimensional linear phase FIR filter design is introduced. In Chapter 7, two-dimensional nonlinear phase FIR filter using TLBO algorithm is described. In Chapter 8, future works and conclusions are presented.

1.6 Main contributions

In this dissertation, both FIR and IIR digital filter designs under least squares (LS) and minimax (MM) criteria are designed using the standard TLBO and improved algorithms. Main contributions are summarized as follows:

1. Using the standard TLBO and improved algorithms, five types of digital filters: linear phase FIR digital filters, multiobjective general FIR digital filters, multiobjective IIR digital filters, two-dimensional linear phase FIR filters, and two-dimensional general FIR filters are designed using minimax and least squares approximations. In general, the least squares algorithm, the Parks-McClellan algorithm, the multiobjective particle swarm optimization (MOPSO) algorithm, and others are selected for comparisons. Design results indicate improved performance can be obtained.

2. A gradient-based TLBO is developed for linear phase FIR digital filter design. Design results indicate that the gradient-based TLBO can obtain similar optimal solutions but faster convergence when compared to those of the standard TLBO.

3. A non-dominated MOTLBO with crowding distance is formed by combining the standard TLBO algorithm with non-dominated set and crowding distance. Filter design results have shown that the non-dominated MOTLBO with crowding distance can achieve faster convergence than the MOPSO in multiobjective digital filter design.

4. A gradient-based MOTLBO with Manhattan distance is formed by replacing the student learning phase of the standard TLBO by a gradient learning

phase. Filter design results have shown that the gradient-based MOTLBO with Manhattan distance can achieve improvements when compared to MOPSO in least squares general FIR filter design.

CHAPTER 2

Teaching-learning-based Optimization and Improved Algorithms

2.1 Review of swarm intelligence algorithms

Inspired by social insect behaviors, researchers have developed a series of new solutions to traditional problems through the simulation of social insects. These studies are examples of swarm intelligence research. Swarm refers to a group of subjects that can communicate directly or indirectly by changing the local environment, who can collaborate to solve distribution problems. The so-called swarm intelligence refers to the characteristics of intelligent behaviors through the cooperation of non-intelligent entities. Swarm intelligence provides the foundation for finding solutions to complex distributed problems without centralized control and without providing a global model.

The optimal solution of the swarm intelligence algorithms is the process of generating new solutions by successive iterations and evolutions of corresponding rules from random initial solutions. In swarm intelligence algorithms, the set of multiple solutions is called *population*, which is denoted as *P(t)*, where *P* represents the size of the population (population size), and t represents the iteration. $c_1(t)$, $c_2(t)$, …, $c_P(t)$ represents the individual solutions in the population. A new individual in the population (offspring) is usually produced by its parents. This kind of mating combination is called the evolutionary model. The iterative process of evolutionary computation can be summarized into three basic segments: social collaboration, self-adaptation and competitive evolution. In the process of social collaboration, individuals exchange information and learn from each other. Individuals in the population continuously adjust their own state to adapt to the environment through active or passive means in the self-adaptation progress. Competing with each other means that individuals with better status in the population will have a greater chance of survival and enter the sub-population, which is the update strategy of the population [83]-[85].

In social collaboration, some individuals will be selected for information exchange and mutual learning through the corresponding selection mechanisms. The information involves four elements: (1) the method of individual selection, (2) the size of the individual, (3) the generation mechanism of the new experimental individual, and (4) the usage of the historical information of the population.

Self-adaptation mechanism refers to the individual's continuous adjustment of its status through active or passive mechanisms to adapt to the living environment in which it lives. Individuals adjust their state through global and local searches. The global search guarantees the individual's ability to explore new solutions in a wider range, which can more effectively ensure the diversity of the population and avoid premature convergence. The local search is opposite that it helps an individual to converge to the local best more easily. However, it is more effective to improve the quality of individuals by shortening the convergence of the algorithm. The self-adaptation of individuals in a population is usually used to balance the two search mechanisms. Through the above two processes, a new experimental individual can be generated.

Swarm intelligence algorithms select individuals from P parents and m temporary sub-species into the next-generation population through a competitive mechanism. In most swarm intelligence algorithms, the size of the population is generally fixed. The individual replacement strategy is divided into the whole generation replacement strategy, $r(P,\ m)$, and the partial replacement strategy, $r(P+m)$. The former refers to the P parents is completely replaced by the m child generation individuals, and the latter means that only some of the P parent individuals are replaced. In order to save elite individuals, the elite retention strategy will be selected. This means that the outstanding individuals in the parent individuals are not replaced and will enter the next generation of individuals directly.

Once the initial population within the solution space is input, the swarm intelligence algorithm framework can create an output for the best individual, represented by $c_{best}(t)$. The steps in between are described in the following, which has nine steps:

1. Initialize parameters such as population size and number of iterations.
2. Randomly initialize the population in the solution space.
3. While the termination condition is not met, do loop.
4. Calculate the fitness value of each individual of *P(t).*
5. Select some individuals for social collaborative operations.
6. Execute self-adaptation.
7. Compute competing operations to create a new generation of populations.
8. End while.
9. Output the final solution.

Through the above calculation framework, swarm intelligence algorithms apply the three operations (social collaboration, self-adaptation, and competitive evolution) to the individuals in the population. Every individual is approached the optimal solution to achieve the purpose of optimization.

PSO [86] and ACO [87] algorithms are two of the most important members of the swarm intelligence algorithm family. The basic idea is to simulate the behavior of natural biological groups to construct a stochastic optimization algorithm. The difference is that the particle swarm algorithm simulates the bird group behavior, while the ant colony algorithm simulates the ant foraging principle. These two are beneficial in this context as they are compatible with the similarities that they share, but they also complement each other due to their differences.

PSO and ACO algorithms share seven critical similarities. Firstly, both of them are within a class of uncertain algorithms. Uncertainty embodies the biological mechanisms of natural organisms and is superior to deterministic algorithms in solving certain problems. The uncertainty of the bionic optimization algorithm is accompanied by its randomness. The main steps contain random factors, so there is great uncertainty in the occurrence of events in the iterative process of the algorithm.

In addition, both of them are within a class of probabilistic global optimization algorithms. The advantage of the non-deterministic algorithms is that they have more chances to solve the global optimal solution. Likewise, neither is dependent on the strict mathematical nature of the optimization problem itself. The optimization process does not rely on the mathematical nature of the optimization problem itself, such as continuity, conductivity, and the mathematical description of the objective function and constraints. Both are bionic optimization algorithms based on multiple agents. Each agent in bionic optimization algorithms cooperates with each other to better adapt to the environment and demonstrates the ability to interact with the environment. Moreover, the two algorithms are intrinsic parallelism. The essential parallelism of the bionic optimization algorithm is manifested in two aspects: the inherent and intrinsic parallelism of the bionic optimization calculation. This enables bionic optimization algorithms to complete of their overall goal, which is highlighted in the movement of multiple agents' individual behaviors. Finally, both algorithms have self-organization and evolution. With memory function, all particles retain the knowledge of the solution. Thus, in an uncertain complex environment, bionic optimization algorithms can continuously improve the individual adaptability of the algorithm through self-learning.

However, PSO and ACO algorithms differ with respect to their mechanisms. Firstly, PSO algorithm is a heuristic algorithm with a fairly simple principle. Compared with other bionic algorithms, it requires less code and parameters. PSO algorithm shares information through the best advantages currently searched, which is largely a single item information sharing mechanism. PSO algorithm is less affected by the dimension of the problem being solved. The mathematical basis of particle swarm optimization is relatively weak, and there is still a lack of profound and universal theoretical analysis. In the study of convergence analysis, it is necessary to further transform deterministic to random. In contrast, ACO algorithm uses a positive feedback mechanism, which is one of the most significant features of other bionic algorithms. An individual in the ant colony algorithm can only perceive local information and cannot directly use global information. Basic ACO algorithm generally requires longer search times and is prone to stagnation. The convergence performance of the ant colony algorithm is sensitive to the setting of initialization parameters. In addition, ACO algorithm has a more mature convergence analysis method and can estimate the convergence speed.

As an important branch of science, swarm intelligence algorithms have been widely recognized, rapidly promoted and applied in many engineering fields, such as system control, artificial intelligence, pattern recognition, production scheduling, and computer engineering. In view of complexity, constraint, nonlinearity, multi-pole and difficult modeling of practical engineering problems, swarm intelligence algorithms are a major research target, and many researchers have sought to create an algorithm suitable for large-scale parallel and intelligent features. Since the 1980s, some novel optimization algorithms have been developed by simulating or revealing certain natural phenomena or processes. These algorithms include artificial neural networks, chaos, genetic algorithms, evolutionary programming, simulated annealing, tabu search, and hybrid optimization strategies. Collectively the novel algorithms provide new ideas and means for solving complex problems through the inclusion of mathematics, physics, biological evolution, artificial intelligence, neuroscience and statistical mechanics. For simple-function optimization problems, classical algorithms are more efficient and can obtain the exact optimal solution of the function. However, for complex-functions and combinatorial optimization problems with nonlinear and multi-extreme characteristics, classical algorithms are often unable to obtain exact optimal solutions. The unique advantages and mechanisms of these algorithms allow them to be successfully applied in many fields, thereby attracting the attention of researchers and resulting in a research boom in this field.

2.2 Literature review of TLBO algorithm

Teaching-learning-based optimization (TLBO) algorithm, which only requires common controlling parameters, population size, and a number of generations was proposed in 2011 [88]-[93]. The parameter regulations are required under different conditions, which brings a certain limitation. More specifically, artificial bee colony algorithm (ABC) [94]-[98] involves the number of employed foragers, scouters, followers, and limits. The performance of harmony search (HS) [99]-[101] is dependent on memory consideration rate, pitch adjusting rate, and bandwidth (bw). Likewise, cuckoo search algorithm (CSA) [102]-[104] requires switching probability, step size scaling factor and Levy index. In comparison, TLBO algorithm is independent of algorithm-specific control parameters. Since TLBO algorithm was proposed, TLBO algorithm has attracted attention from researchers because of its parameter-free characteristic. In [105], TLBO algorithm was applied by manufacturing industries to solve assembly sequence problems. The TLBO algorithm was also used to design screen primer problems [106]. In [107], TLBO was used to solve optimal coordination of directional over-current relays problems. Moreover, the application of TLBO for flow shop scheduling was described in [108]-[111]. In [112], TLBO was applied to power system scheduling. Moreover, binary TLBO algorithm was used to assist for designing plasmonic nanoparticles [113].

However, when compared to other population-based algorithms, TLBO algorithm is limited by its searching ability. In [114], a novel teacher learning phase was used to enhance the global search ability for synthesis of thinned concentric circular antenna arrays. In [115], one more learning step was added to TLBO algorithm for dynamic economic/environmental dispatch considering multiple plug-in electric vehicle loads. In addition, reference [116] modified the teaching factor and number of teachers to increase the exploration ability to solve an environmental problem. In [117], a modified local search was applied to parameter estimation of nonlinear chaotic system. Reference [118] modified both teaching phase and learning phase to solve a multiobjective problem. In [119], a hybrid TLBO-PSO was used to achieve a faster convergence and avoid being trapped into local optima. Besides, [120] combined PSO and TLBO algorithms to form an effective algorithm for multiobjective optimization. Similarly, in [121], modified TLBO was combined with double DE for solving an optimal reactive power dispatch problem and comparing to eight other evolutionary algorithms, and the results indicate that the resulting algorithm is more efficient to balance global search and achieve faster convergence than others. In [122], TLBO algorithm with elite strategy was applied to natural PCR-RFLP primer design for SNP genotyping and shown to have a more effective search ability as compared to PSO and GA. Similarity, [123] adopted an improved TLBO with elite strategy for parameters identification of PEM fuel cell and solar cell models. In [124], a modified TLBO algorithm with multiple teachers and adaptive teaching factor was applied to multiobjective optimization of heat exchangers, five groups results had demonstrated that the modified TLBO can reduce the cost as compared to GA. In another study, Rao et al. also proposed another modification at the same year that included changes in the learner phase In [125], a modified TLBO was used to solve unconstrained optimization problems, which outperformed ABC and two modified ABC algorithms. In [126], a novel TLBO algorithm with a modified teacher phase was applied to energy-efficient scheduling in hybrid flow shop which performed better than GA and IGSA for at least 20 test instances and consumed less computational time. In [127], an adaptive weight was used in the teacher phase for space trusses design problem and when compared to other evolutionary algorithms, both accuracy and time-cost are improved. In [128], a ring-neighborhood topology is selected to update the marks in teacher phase and a mutation step is added at the end of learner phase for ANN applications. In [129], experience information and differential mutation were incorporated into TLBO for improved global optimization. In [130], fuzzy-based TLBO was proposed to minimize the total energy losses in power system, results indicate improved performance as compared to TLBO and other optimization techniques. Besides, in [131], TLBO with a fuzzy grouping learning strategy in which fuzzy K-means is used to create K centers, each of which acts as the mean of its corresponding group to enhance the performance of TLBO algorithm. To this end, In [132], a Gaussian bare-bones TLBO (GBTLBO) algorithm, with its modified version (MGBTLBO) were proposed for the optimal reactive power dispatch (ORPD) problem with discrete and continuous control variables in the standard IEEE power systems for reduction in power transmission loss. In [133], combines TLBO was combined with DE to design least squares linear phase FIR filters.

2.3 Teaching-learning-based optimization

In TLBO algorithm, a population is composed of P learners, and different design variables (or filter coefficients) are viewed as S different subjects attached to the learners. All the scores of a learner (or a solution) $\boldsymbol{c}_p(t)$ (for $p = 1, 2, \cdots, P$) is evaluated collectively by the value of its objective (or error) function $e[\boldsymbol{c}_p(t)]$ in an optimization problem. Within a population, the best solution is assumed to be offered and maintained by the teacher. The operations of the TLBO are divided into two phases, the teacher phase and the learner phase. In TLBO, a global search is applied to explore the starting point and then the starting point is perturbed in a local search space iteratively to arrive at an optimal solution.

1. Teacher phase

This phase is a process that a teacher teaches knowledge to the learners and improves the mean results of the class. $M_s(t)$ is the mean result of the learners in a particular subject *s* in the *t*th iteration.

$$m_s(t) = \frac{1}{P}\sum_{p=1}^{P} \boldsymbol{c}_{p,s}(t) \tag{2.1}$$

As the teacher is the most knowledgeable people and has the best result over all subject, let $\boldsymbol{c}_{best}(t)$ represent the teacher. Learners get knowledge from the teacher and the quality of the teacher decides the mean result, so the difference $Diff_s(t)$ between the result of the teacher and mean result the of the learners in each subject is expressed as

$$Diff_s(t) = rand \times (c_{best,s}(t) - t_f m_s(t)) \tag{2.2}$$

where $c_{best,s}(t)$ is the value of the teacher in subject *s*. t_f is the teaching factor which decides the value of mean to be changed, and *rand* is a random number in the range [0,1]. The value of t_f can be either 1 or 2. The value of t_f is decided randomly as,

$$t_f = round[1 + rand] \tag{2.3}$$

where t_f is not a parameter of the TLBO algorithm. The value of t_f is not given as an input to the algorithm and its value is randomly decided by the algorithm using Eq. (2.3).

Then each learner has a potential updated counterpart and therefore

$$\boldsymbol{c}'_{p,s}(t) = \boldsymbol{c}_{p,s}(t) + Diff_s(t) \tag{2.4}$$

where $\boldsymbol{c}'_p(t)$ is the updated value of $\boldsymbol{c}_p(t)$ in subject *s*.

The algorithm accepts $\boldsymbol{c}'_p(t)$ only if it gives a better value of objective function otherwise keeps the previous solution such that

$$\boldsymbol{c}_p(t) = \begin{cases} \boldsymbol{c}'_p(t) & if\ e\left[\boldsymbol{c}'_p(t)\right] < e\left[\boldsymbol{c}_p(t)\right] \\ \boldsymbol{c}_p(t) & if\ e\left[\boldsymbol{c}'_p(t)\right] \geq e\left[\boldsymbol{c}_p(t)\right] \end{cases} \tag{2.5}$$

All the accepted scores at the end of the teacher phase are maintained and these values become the input to the learner phase.

2. Learner phase

Students not only can get knowledge from the teacher but also can improve their individuals through interaction among themselves. In this step two students are randomly selected namely *A* and *B* ($\boldsymbol{c}_A(t) \neq \ \boldsymbol{c}_B(t)$**)** among the entire class. After sharing or exchanging their knowledge, the new results are gained as:

$$\boldsymbol{c}'_A(t) = \ \boldsymbol{c}_A(t) + rand \times \left(\ \boldsymbol{c}_A(t) - \ \boldsymbol{c}_B(t)\right),\ if\ \ e[\boldsymbol{c}_A(t)] < \ e[\boldsymbol{c}_B(t)] \tag{2.6}$$

$$\boldsymbol{c}'_A(t) = \ \boldsymbol{c}_A(t) + rand \times \left(\ \boldsymbol{c}_{\boldsymbol{B}}(t) - \ \boldsymbol{c}_A(t)\right),\ if\ \ e[\boldsymbol{c}_A(t)] > \ e[\boldsymbol{c}_B(t)] \tag{2.7}$$

The algorithm accepts $\boldsymbol{c}'_A(t)$ (for $p = 1, 2,\ \cdots, P$) only if it gives a better value of objective function otherwise it keeps the previous solution such that

$$\boldsymbol{c}_A(t) = \begin{cases} \boldsymbol{c}'_A(t) & if\ e[\boldsymbol{c}'_A(t)] < e[\boldsymbol{c}_A(t)] \\ \boldsymbol{c}_A(t) & if\ e[\boldsymbol{c}'_A(t)] \geq e[\boldsymbol{c}_A(t)] \end{cases} \tag{2.8}$$

2.4 Gradient-based TLBO Algorithm

Based on analysis of previous TLBO design results, TLBO algorithm has a limitation on search ability. In order to enhance the search capability, a new phase, gradient-based phase, is proposed to improve the convergence speed and accuracy of the TLBO algorithm. The gradient descent method is a traditional method of optimization, and the gradient represents the sharpest changing trend of a function at a given point. In fact, the gradient descent method is used to numerically search for local minimum or maximum values. It is an efficient, high-speed and reliable method in practical applications.

In the gradient-based TLBO, a global search is applied to explore the starting point and then the starting point is perturbed in a local search space iteratively to arrive at an optimal solution. Moreover, a parameter—step size—is added into the gradient-based phase. A proper control parameter can improve robustness.

There is a flowchart Fig. 2.1 for the improved TLBO algorithm. To this end, the gradient-based TLBO algorithm is proposed in this section.

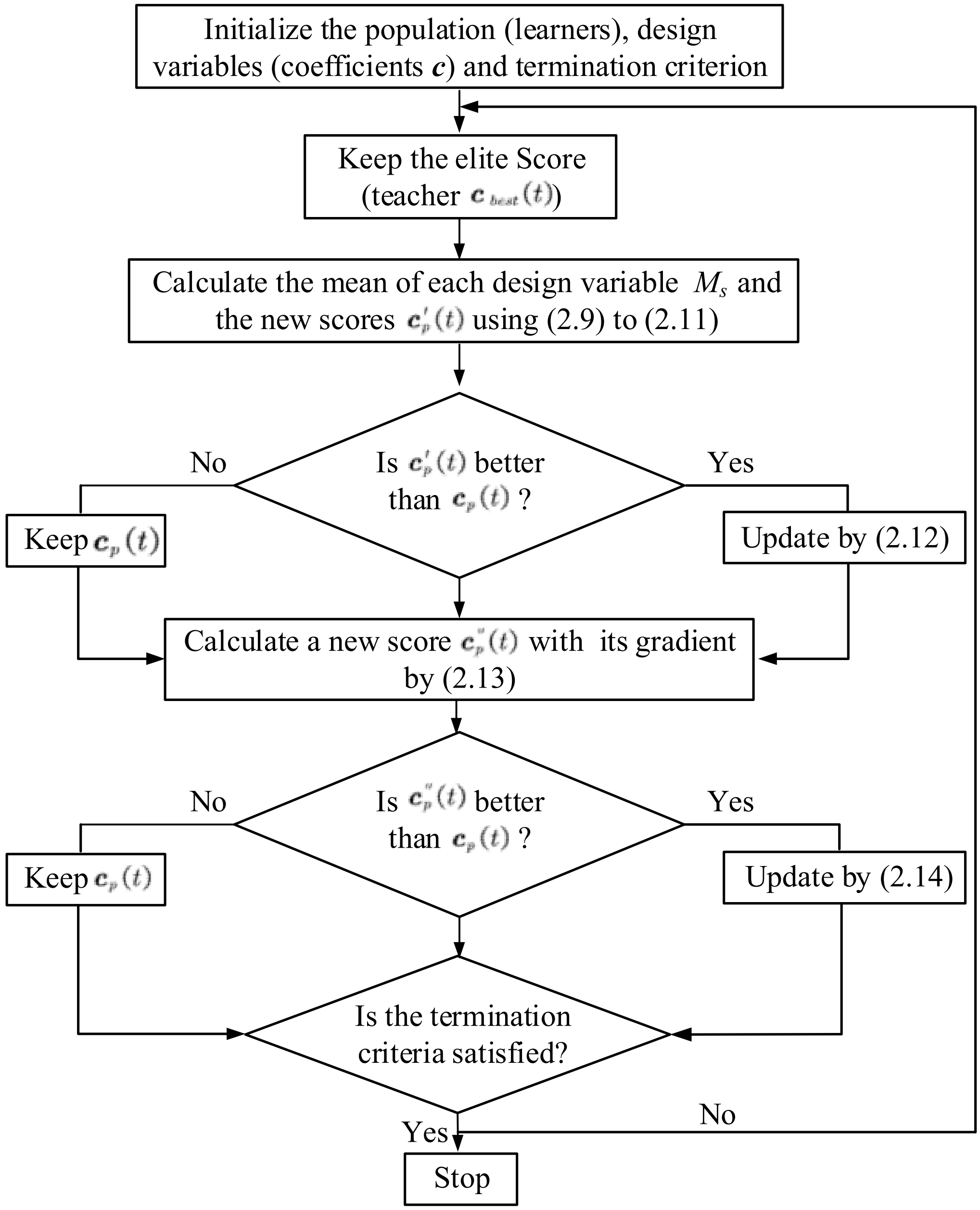

Initialize the population (learners), design variables (coefficients $\boldsymbol{c}$) and termination criterion
Keep the elite Score (teacher $\boldsymbol{c}_{best}(t)$)
Calculate the mean of each design variable M_s and the new scores $\boldsymbol{c}'_p(t)$ using (2.9) to (2.11)
Is $\boldsymbol{c}'_p(t)$ better than $\boldsymbol{c}_p(t)$?
No
Yes
Keep $\boldsymbol{c}_p(t)$
Update by (2.12)
Calculate a new score $\boldsymbol{c}''_p(t)$ with its gradient by (2.13)
Is $\boldsymbol{c}''_p(t)$ better than $\boldsymbol{c}_p(t)$?
No
Yes
Keep $\boldsymbol{c}_p(t)$
Update by (2.14)
Is the termination criteria satisfied?
Yes
No
Stop

Fig. 2.1 Flowchart of gradient-based TLBO algorithm

In the gradient-based TLBO algorithm, a population is composed of P learners, and different design variables (or filter coefficients) are viewed as S different subjects attached to the learners. All the scores of a learner (or a solution) $\boldsymbol{c}_p(t)$ (for $p = 1, 2, \cdots, P$) is evaluated collectively by the value of its objective (or error) function $e[\boldsymbol{c}_p(t)]$ in an optimization problem.

Within a population, the best solution is assumed to be offered and maintained by a teacher. The operations of the gradient-based TLBO is divided into two phases, the teacher phase and the gradient-based learning phase. In the gradient-based TLBO, a search is applied to explore the starting point and then the starting point is perturbed in a local search space iteratively to arrive at an optimal solution.

1. Teacher phase

This phase is a process that a teacher teaches knowledge to the learners and improves the mean results of the class. $m_s(t)$ (for $s = 1, 2, \cdots, S$) is the mean result of the learners in a specific subject s in the tth iteration, which is same as the teacher phase in the original TLBO algorithm. As the teacher is the most knowledge people and has the best result on that subject, let $\boldsymbol{c}_{best}(t)$ represent the teacher. Learners get knowledge from the teacher and the quality of the teacher decides the mean result, so the difference $Diff_s(t)$ between the teacher and mean result of the learners in each subject is expressed as

$$Diff_s(t) = rand * (\boldsymbol{c}_{best,s}(t) - t_f m_s(t)) \qquad (2.9)$$

where t_f is the teaching factor which decides the value of mean to be changed, and $rand$ is a random number in the range [0,1]. The value of t_f can be either 1 or 2. The value of t_f is decided randomly as,

$$t_f = round[1 + rand] \qquad (2.10)$$

T_F is not a parameter of the TLBO algorithm. The value of T_F is not given as an input to the algorithm and its value is randomly decided by the algorithm using Eq. (2.10).

$Diff_s(t)$is added to current score of learners in different subjects to generate new learners:

$$\boldsymbol{c}'_{p,s}(t) = \boldsymbol{c}_{p,s}(t) + Diff_s(t) \qquad (2.11)$$

where $\boldsymbol{c}'_p(t)$ is the updated value of $\boldsymbol{c}_p(t)$ in subject *s*.

The algorithm accepts $\boldsymbol{c}'_p(t)$ only if it gives a better value of objective function otherwise it keeps the previous solution such that

$$\boldsymbol{c}_p(t) = \begin{cases} \boldsymbol{c}'_p(t) & if\ e\left[\boldsymbol{c}'_p(t)\right] < e\left[\boldsymbol{c}_p(t)\right] \\ \boldsymbol{c}_p(t) & if\ e\left[\boldsymbol{c}'_p(t)\right] \geq e\left[\boldsymbol{c}_p(t)\right] \end{cases} \tag{2.12}$$

All the accepted scores at the end of the teacher phase are maintained and these values become the input to the learner phase.

2. Gradient-based learning phase

The learner phase is not efficient due to the random selection approach in original TLBO algorithm. For this reason, a gradient-based phase is proposed to speed up the global search, which leads to (3.17)

$$\boldsymbol{c}''_p(t) = \boldsymbol{c}_p(t) - \mu \frac{\partial e\left[\boldsymbol{c}_p(t)\right]}{\partial \boldsymbol{c}_p(t)} \tag{2.13}$$

where μ stands for the step size.

To obtain the optimal result the score vector $\boldsymbol{c}_p(t)$ is selected through (2.14)

$$\boldsymbol{c}_p(t) = \begin{cases} \boldsymbol{c}''_p(t) & if\ e\left[\boldsymbol{c}''_p(t)\right] < e\left[\boldsymbol{c}_p(t)\right] \\ \boldsymbol{c}_p(t) & if\ e\left[\boldsymbol{c}''_p(t)\right] \geq e\left[\boldsymbol{c}_p(t)\right] \end{cases} \tag{2.14}$$

After applying this new phase to the proposed algorithm, the performance could be enhanced, which is reflected in the convergence. At the end of each iteration, if the value of error function of the best learner is smaller than the one of the global best learner, the subject score vector of the teacher is delivered to the next iteration.

CHAPTER 3

Linear phase FIR filter design

Digital Hilbert transformers have many applications in the field of digital signal processing, for instance, latency analysis in neuro-physiological signals [134]; psychoacoustic design of bizarre stimuli [135]; communication problems of speech data compression [136]; multi-channel acoustic echo cancellation of regularization of convergence [137]; and auditory prostheses in signal processing [138]. Digital Hilbert transformer design can be formulated as a Type 3 and Type 4 linear phase FIR digital filter optimization problem. Both Type 3 and Type 4 linear phase FIR filters are characterized by odd symmetry. In this Chapter, a gradient-based TLBO algorithm is proposed to improve the performance of the standard TLBO algorithm. Least squares and minimax designs are used to minimize error function, several design examples of Type 3 and Type 4 Hilbert transformers using the standard TLBO algorithm and the gradient-based TLBO algorithm are presented. Design results are compared to those obtained by the least squares minimization and the Parks-McClellan algorithm [24].

In this Chapter, Section 3.1 describes the formulation of Hilbert transformer design problem and objective functions; Section 3.2 presents the design examples and results; Section 3.2.1 shows the Type 3 bandpass HT designs; Section 3.2.2 describes Type 4 highpass HT designs; Section 3.2.3 provides a comparison with least squares algorithm and minimax algorithm; and Section 3.3 gives conclusions.

3.1 Problem formulation

3.1.1 Type 3 Linear Phase FIR Digital Filters

The causal transfer function of an *M* odd and odd symmetry FIR digital filter can be expressed [1] as

$$H(z) = \left\{\sum_{n=0}^{\frac{M-3}{2}} h(n) z^{-n}\left[1 - z^{-[M-(2n+1)]}\right]\right\} + \frac{1}{2} h\left(\frac{M-1}{2}\right) z^{-\frac{M-1}{2}}\left[1 - z^{0}\right] \quad (3.1)$$

where $h(n)$ for $n = 0$ to $(M-1)/2$ represent the unique set of impulse responses.

According to the M odd and odd symmetrical features of the Type 3 linear phase Hilbert transformer, the coefficient vector **h** is given by

$$\mathbf{h} = [h(0), h(1), h(2), \cdots, h(n), \cdots, h(M-2), h(M-1)] \tag{3.2}$$

$$h(n) = -h(M-1-n) \quad \text{for } 0,1,2,3, \cdots, \left(\tfrac{M-3}{2}\right) \tag{3.3}$$

The frequency response can be evaluated by substituting $z = e^{j\omega T}$ into (3.1) as

$$H(\omega) = je^{-j\left(\frac{M-1}{2}\right)\omega T} \sum_{n=0}^{\frac{(M-3)}{2}} 2h(n) \sin\left[\left(\tfrac{M-1}{2} - n\right) \omega T\right] \tag{3.4}$$

The *N (=M-1)*th-order Type 3 linear phase digital Hilbert transformer can be represented by

$$H(\omega) = je^{-j\frac{N}{2}\omega T} A(\mathbf{c}, \omega) \tag{3.5}$$

where

$$A(\boldsymbol{c}, w) = \mathbf{c}^T \,\mathbf{sin}(\omega) \tag{3.6}$$

$$\mathbf{c} = \left[c_1, c_2, c_3, \dots, c_{\left(\frac{M-3}{2}\right)}\right]^T \tag{3.7}$$

$$= 2\left[h\left(\frac{M-1}{2} - 1\right), \cdots, h(2), h(1), h(0)\right]^T$$

$$c_0 = h\left(\tfrac{M-1}{2}\right) = 0 \tag{3.8}$$

$$\mathbf{sin}(\omega) = \left[\sin(\omega T) \quad \sin(2\omega T) \quad \cdots \quad \sin\left(\tfrac{N}{2}\omega T\right)\right]^T \tag{3.9}$$

From (3.4), the term j corresponds to $e^{j\pi/2}$, the group delay response of the even *N*th-order Type 3 linear phase FIR filter is

$$\tau(\omega) = -\frac{\partial\theta(\omega)}{\partial\omega T} = -\frac{\partial\left[-\left(\frac{N}{2}\right)\omega T+\frac{\pi}{2}\right]}{\partial\omega T} = \frac{N}{2} \tag{3.10}$$

3.1.2 Type 4 Linear Phase FIR Digital Filters

The transfer function of a causal *M* even and odd symmetry linear FIR digital filter [1] is given by

$$H(z) = \sum_{n=0}^{\frac{M-2}{2}} h(n)z^{-n}\left[1 - z^{-[M-(2n+1)]}\right] \tag{3.11}$$

The Type 4 linear phase digital Hilbert transformer is $M(= N + 1)$ even and odd symmetry such that its M impulse responses can be expressed as

$$\mathbf{h} = [h(0), h(1), h(2), \ldots, h(n), \ldots, h(M-2), h(M-1)] \tag{3.12}$$

$$h(n) = -h(M-1-n) \text{ for } n = 0,1,2,3,\ldots,\left(\frac{M}{2}-1\right) \tag{3.13}$$

The frequency response of a Type 4 linear phase digital Hilbert transformer can be evaluated by substituting $z = e^{j\omega T}$ into (3.11) as

$$H(\omega) = je^{-j\frac{M-1}{2}\omega T}\sum_{n=0}^{\frac{M-2}{2}} 2h(n)\sin\left(\frac{M-1}{2}-n\right)\omega T = je^{-j\frac{M-1}{2}\omega T}A(\mathbf{c},\omega) \tag{3.14}$$

where

$$A(\boldsymbol{c}, w) = \mathbf{c}^T\,\mathbf{sin}(\omega) \tag{3.15}$$

$$\mathbf{c} = \left[c_0, c_1, c_2, c_3, \ldots, c_{\left(\frac{M-1}{2}\right)}\right]^T = 2\left[h\left(\frac{M}{2}-1\right), \cdots, h(2), h(1), h(0)\right]^T \tag{3.16}$$

$$\mathbf{sin}(\omega) = \left[\sin(\tfrac{\omega T}{2}) \quad \sin\left(\frac{3\omega T}{2}\right) \quad \cdots \quad \sin\left(\frac{M-1}{2}\omega T\right)\right]^T \tag{3.17}$$

From (3.14), the term *j* corresponds to $e^{j\pi/2}$, the group delay response of the odd *N*th-order Type 4 linear phase digital Hilbert transform is

$$\tau(\omega) = -\frac{\partial\theta(\omega)}{\partial\omega T} = -\frac{\partial\left[-\left(\frac{N}{2}\right)\omega T+\frac{\pi}{2}\right]}{\partial\omega T} = \frac{N}{2} \tag{3.18}$$

3.1.3 Hilbert Transformer

The ideal frequency response of a digital Hilbert transformer is denoted as (3.19) in [80

$$H_d(\omega) = -je^{-j\tau\omega} \quad \text{for } 0 \le \omega \le \pi \tag{3.19}$$

where the parameter τ denotes the group delay. The frequency response $H_d(w)$ is related to the desired amplitude response $A_d(w_i)$ by

$$\begin{gathered} H_d(\omega) = je^{-j\tau\omega}A_d(\omega) = je^{-j\tau\omega}(-1) \\ \text{for } 0 \le \omega \le \pi \end{gathered} \tag{3.20}$$

According to (3.20), the ideal amplitude response of the digital Hilbert transformer is defined by

$$A_d(\omega) = -1 \quad \text{for } 0 \le \omega \le \pi \tag{3.21}$$

3.1.4 Objective Functions

3.1.4.1 Minimax design

The optimization problem for Type 3 and Type 4 Hilbert transformer is to search for an optimal coefficient vector **c** that minimizes the weighted minimax (MM) objective function $e(\mathbf{c})$ with respect to **c** as defined [1] by

$$\begin{gathered} \min_{\mathbf{c}} e(\mathbf{c}) = \min_{\mathbf{c}}[\textstyle\sum_{i=1}^{I} W(\omega_i)|H(\mathbf{c}, w_i) - H_d(\omega_i)|^p]^{1/p} \\ \text{for } \ \omega_i \in F_o \end{gathered} \tag{3.22}$$

where F_o is the optimization frequency grid; and p is even.

For simplicity, the amplitude response is considered, the weighted minimax function $e(\mathbf{c})$ is expressed by

$$\begin{gathered} \min_{\mathbf{c}} e(\mathbf{c}) = \min_{\mathbf{c}}[\textstyle\sum_{i=1}^{I} W(\omega_i)|A(\mathbf{c}, w_i) - A_d(\omega_i)|^p]^{1/p} \\ \text{for } \ \omega_i \in F_o \end{gathered} \tag{3.23}$$

3.1.4.2 Least squares design

The optimization problem both for Type 3 and Type 4 Hilbert transformer is to search for an optimal coefficient vector $\mathbf{c}$ that minimizes the weighted least square (LS) objective function $e(\mathbf{c})$ with respect to $\mathbf{c}$ as defined [1] by

$$\min_{\mathbf{c}} e(\mathbf{c}) = \min_{\mathbf{c}}[\sum_{i=1}^{I} W(\omega_i)|A(\mathbf{c}, \omega_i) - A_d(\omega_i)|^2] \\ \text{for } \omega_i \in F_o \tag{3.24}$$

3.2 Designs and Results

In this chapter, Type 3 linear phase FIR bandpass digital Hilbert transformers of orders *N*=14, 26, 38 and 50 and Type 4 linear phase FIR highpass digital Hilbert transformers [1] of orders *N*=13, 25, 37 and 49 are designed using the standard TLBO algorithm for MM design, as well as the standard TLBO algorithm and the gradient-based TLBO algorithm for LS design.

The initialization of filter coefficients is obtained by adding ±random values to the filter coefficient values obtained by the Matlab function firpm.m for MM designs and by the Matlab function firls.m for LS designs to ensure all the coefficients are within the bound limits.

3.2.1 Type 3 bandpass HT MM designs

The MM errors obtained by the Matlab function firpm.m and the standard TLBO, and maximum iteration numbers and CPU time in seconds of different Type3 linear phase Hilbert transformer designs are listed in Table 3.1. The obtained MM errors indicate that all the standard TLBO designs are better than those of the firpm.m designs. The passband frequency grid and the passband frequency points for optimization and evaluation are listed in Table 3.2. In Table 3.2, the passband transition width decreases as the filter order increases. The corresponding peak MM error, peak error frequency and the number of iterations required for convergence of each design are listed in Table 3.3 and Table 3.4. In Tables 3.3 and Table 3.4, the TLBO designs are compared to the MM designs using the Matlab functions firpm.m. The amplitude response, impulse response of Type 3 and Type 4 Hilbert transform designs are shown in Fig. 3.1 to Fig. 3.8, respectively.

Table 3.1 FIR Filter and TLBO (Gradient-based TLBO) Parameters (Key: LP-FIR 3: Type 3 FIR Hilbert transformer)

Symbol	Description	Value			
$W(\omega_i)$	Frequency weights for $0 \leq \omega_i \leq \pi$	1			
$c_k^{[U]}$	Upper bound of filter coefficients	0.94			
$c_k^{[U]}$	Lower bound of filter coefficients	-0.94			
P	Gradient-based TLBO population size	14	26	38	50
N	Order of LP-FIR3 filter	14	26	38	50
S	Number of distinct coefficients of LP-FIR3 filter	7	13	19	25
τ	Group delay of LP-FIR3 filter	7	13	19	25

Table 3.2 Frequency Grid and Desired Amplitude Response

	N	Frequency Grid	$A_d(\omega_i)$
Optimization	14	[0.10:0.005:0.90]	[21:181]
	26	[0.08:0.005:0.92]	[17:185]
	38	[0.06:0.005:0.94]	[13:189]
	50	[0.04:0.005:0.96]	[9:173]
Evaluation	14	[0.10:0.001:0.90]	[101:901]
	26	[0.08:0.001:0.92]	[81:921]
	38	[0.06:0.001:0.94]	[61:941]
	50	[0.04:0.001:0.96]	[41:961]

Table 3.3 Hilbert transformer minimax errors (Keys: T: Type; BP: Bandpass; HP: Highpass; N: Filter order; A: Algorithm; 1: firpm.m; 2: TLBO; CPU: Time in sec)

T	N	A	Minimax error	Iteration	CPU
3 BP	14	1	0.063859638428037	-	0.11
		2	0.048437168979108	2,500	1.14
	26	1	0.056149417077486	-	0.10
		2	0.014541694349349	7,500	6.86
	38	1	0.050680262166871	-	0.14
		2	0.010494629986522	7,500	9.57
	50	1	0.057868729255083	-	0.11
		2	0.017519610534098	2,000,000	3576.12

Table 3.4 Hilbert transformer minimax peak error (Keys: T: Type; BP: Bandpass; N: filter order; MM: Minimax; A: Algorithm; 1: firpm.m; 2: TLBO; L: Peak error frequency; C: Converged iteration number)

T	N	A	Peak MM error	L	C
3 BP	14	1	0.062508534703398	0.282	-
		2	0.047738450350362	0.147	226
	26	1	0.055209896344805	0.814	-
		2	0.014405922412530	0.101	1,679
	38	1	0.049964067125621	0.494	-
		2	0.010421858229430	0.893	1,668
	50	1	0.057192424608021	0.743	-
		2	0.017715320356500	0.520	1,826,795

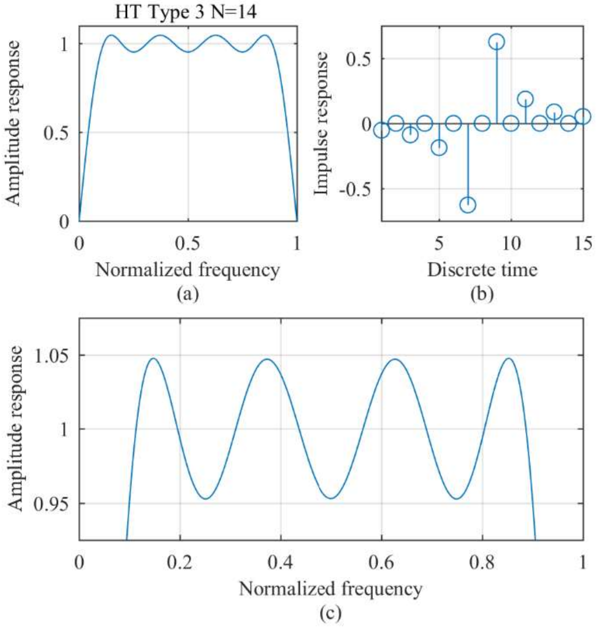


Fig. 3.1 Hilbert transformer minimax design for Type 3 N =14 (a) Amplitude response, (b) Impulse response, (c) Passband amplitude response

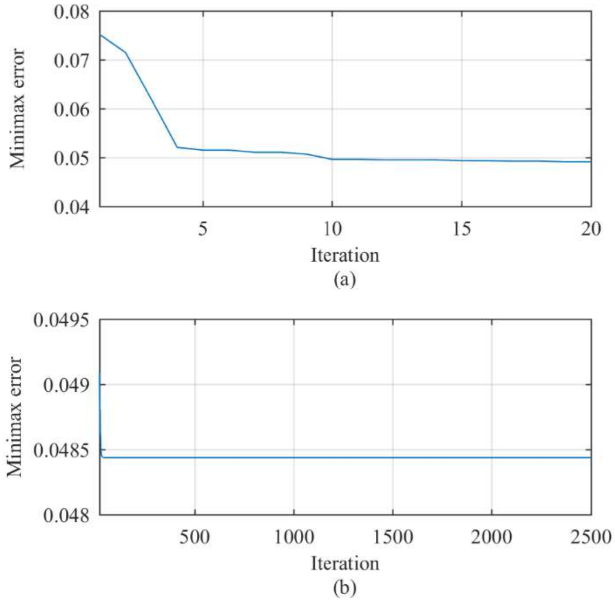


Fig. 3.2 Convergence curve of minimax error for Hilbert transformer design for Type 3 N =14 (a) Interval 1, (b) Interval 2

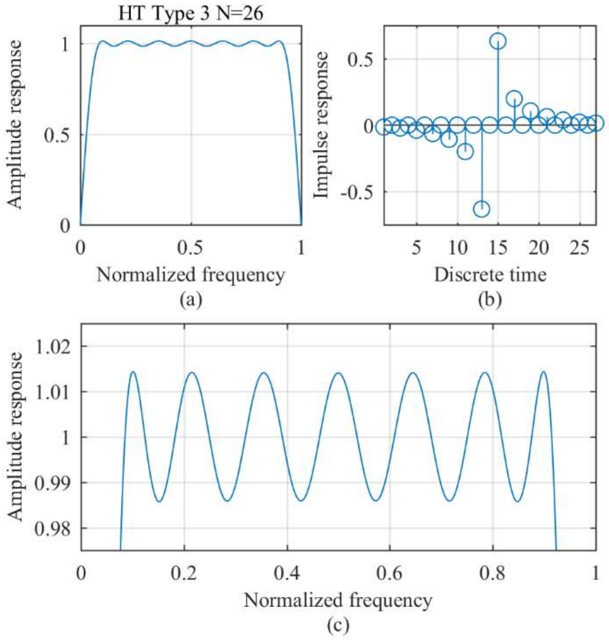


Fig. 3.3 Hilbert transformer minimax design for Type 3 N =26 (a) Amplitude response, (b) Impulse response, (c) Passband amplitude response

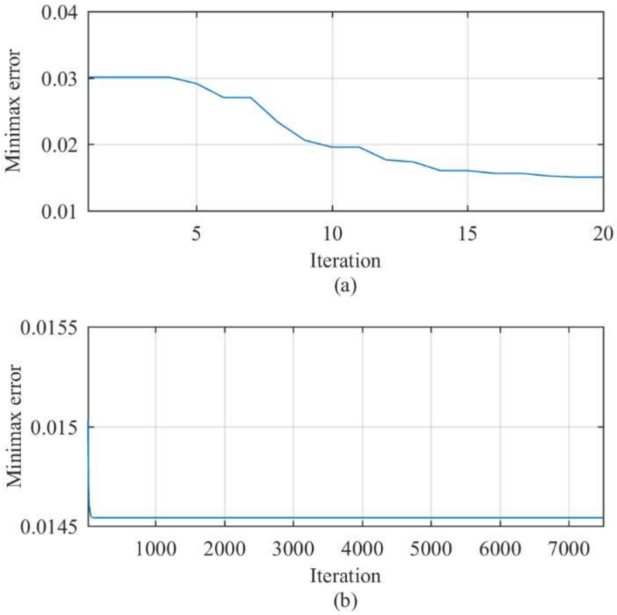


Fig. 3.4 Convergence curve of minimax error for Hilbert transformer design for Type 3 N =26 (a) Interval 1, (b) Interval 2

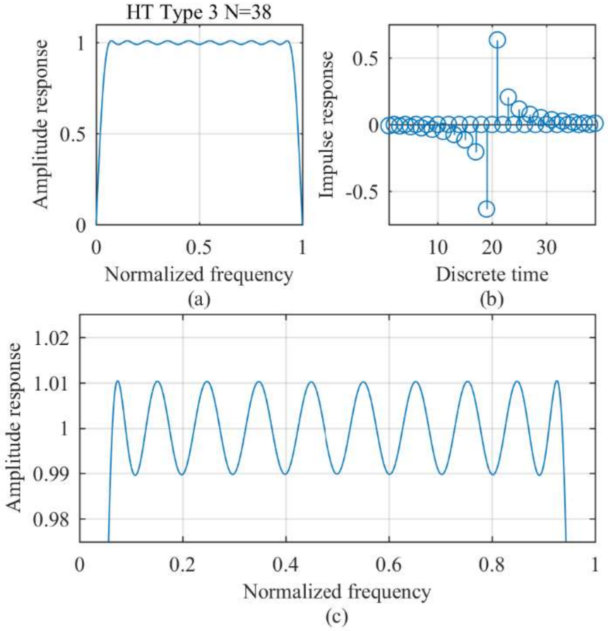
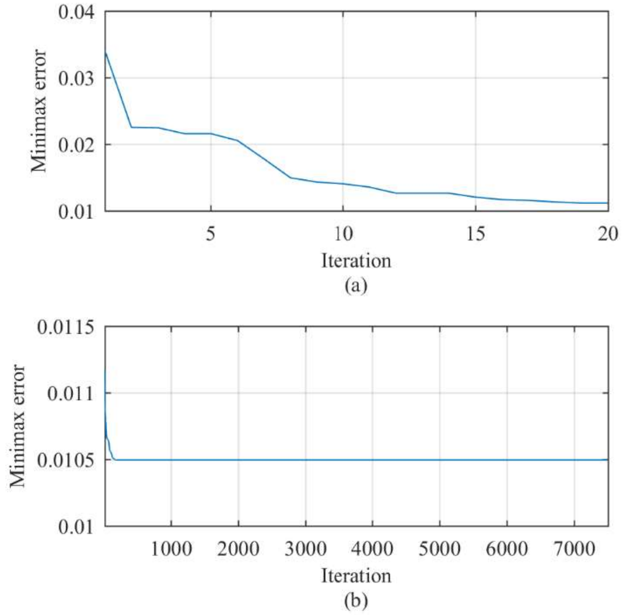


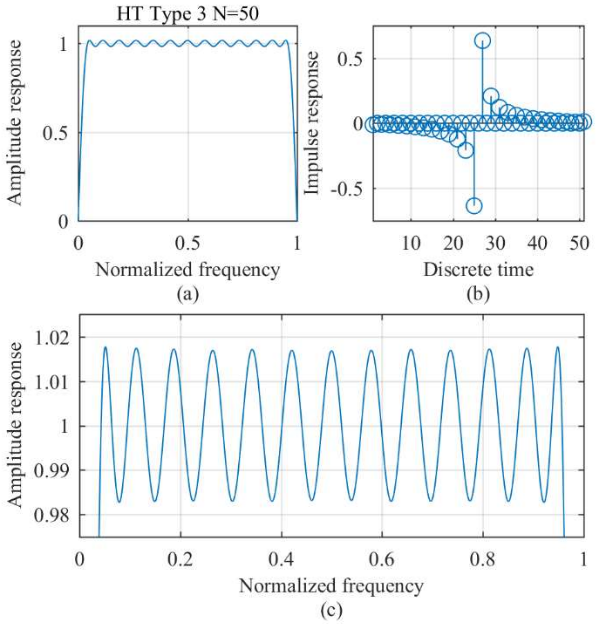
Fig. 3.5 Hilbert transformer minimax design for Type 3 N =38 (a) Amplitude response, (b) Impulse response, (c) Passband amplitude response



(a)

(b)

Fig. 3.6 Convergence curve of minimax error for Hilbert transformer design for Type 3 N =38 (a) Interval 1, (b) Interval 2



(a)

(b)

(c)

Fig. 3.7 Hilbert transformer minimax design for Type 3 N =50 (a) Amplitude response, (b) Impulse response, (c) Passband amplitude response

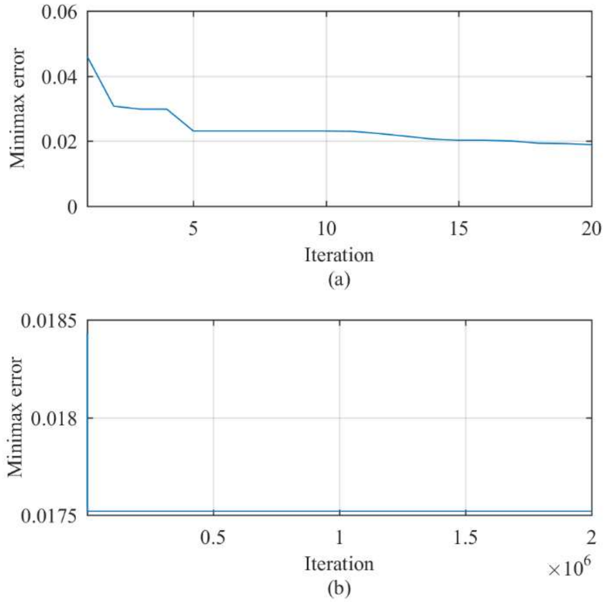


Fig. 3.8 Convergence curve of minimax error for Hilbert transformer design for Type 3 N =50 (a) Interval 1, (b) Interval 2

3.2.2 Type 4 highpass HT MM designs

The MM errors obtained by the Matlab function firpm.m and the standard TLBO, and maximum iteration numbers and CPU time in seconds of different Type 4 linear phase Hilbert transformer designs are listed in Table 3.5. The obtained MM errors indicate that all the standard TLBO designs are better than those of the firpm.m designs. The passband frequency grid and the passband frequency points for optimization and evaluation are listed in Table 3.6. In Table 3.6, the passband transition width decreases as the filter order increases. The corresponding peak MM error, peak error frequency and the number of iterations required for convergence of each design are listed in Table 3.7 and Table 3.8. In Tables 3.7 and Table 3.8, the TLBO designs are compared to the MM designs using the Matlab functions firpm.m. The amplitude response, impulse response of Type 3 and Type 4 Hilbert transform designs are shown in Fig. 3.9 to Fig. 3.16, respectively.

Table 3.5 FIR Filter and TLBO (Gradient-based TLBO) Parameters (Key: LP-FIR 4: Type 4 Hilbert transformer)

Symbol	Description	Value			
$W(w_i)$	Frequency weights for $0 \leq w_i \leq \pi$	1			
$c_k^{[U]}$	Upper bound of filter coefficients	0.95			
$c_k^{[U]}$	Lower bound of filter coefficients	-0.95			
P	Gradient-based TLBO population size	13	25	37	49
N	Order of LP-FIR4 filter	13	25	37	49
S	Number of distinct coefficients of LP-FIR4 filter	7	13	19	25
τ	Group delay of LP-FIR4 filter	6.5	12.5	18.5	24.5

Table 3.6 Frequency Grid and Desired Amplitude Response

	N	Frequency Grid	$A_d(\omega_i)$
Optimization	13	[0.10:0.005:1]	[21:201]
	25	[0.08:0.005:1]	[17:201]
	37	[0.06:0.005:1]	[13:201]
	49	[0.04:0.005:1]	[9:201]
Evaluation	13	[0.10:0.001:1]	[101:1001]
	25	[0.08:0.001:1]	[81:1001]
	37	[0.06:0.001:1]	[61:1001]
	49	[0.04:0.001:1]	[41:1001]

Table 3.7 Hilbert transformer minimax errors (Keys: T: Type; HP: Highpass; N: Filter order; A: Algorithm; 1: firpm.m; 2: TLBO; CPU: Time in sec)

T	N	A	Minimax error	Iteration	CPU
4 HP	13	1	0.063040414548992	-	0.08
		2	0.062675998771068	25,000	11.38
	25	1	0.017736735997146	-	0.08
		2	0.017593020576835	75,000	77.21
	37	1	0.012085967205373	-	-
		2	0.012039653786367	100,000	138.81
	49	1	0.019316280983026	-	0.08
		2	0.018912632578120	1,000,000	1919.82

Table 3.8 Hilbert transformer minimax peak error (Keys: T: Type; HP: Highpass; N: filter orders; MM: Minimax; A: Algorithm; 1: firpm.m; 2: TLBO; L: Peak error frequency; C: Converged iteration number)

T	N	A	Peak MM error	L	C
4 HP	13	1	0.061784018811740	0.273	-
		2	0.062108754071569	0.100	24,194
	25	1	0.017421661549390	0.608	-
		2	0.017582610593351	0.103	73,691
	37	1	0.011822832074982	0.947	-
		2	0.011928934257905	0.076	35,075
	49	1	0.018939292915125	0.392	-
		2	0.019591494065284	0.520	500,798

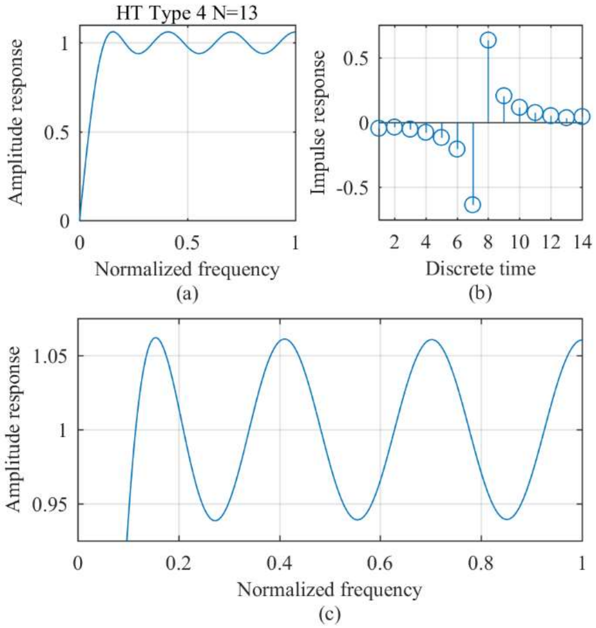


Fig. 3.9. Hilbert transformer minimax design for Type 4 N =13 (a) Amplitude response, (b) Impulse response, (c) Passband amplitude response

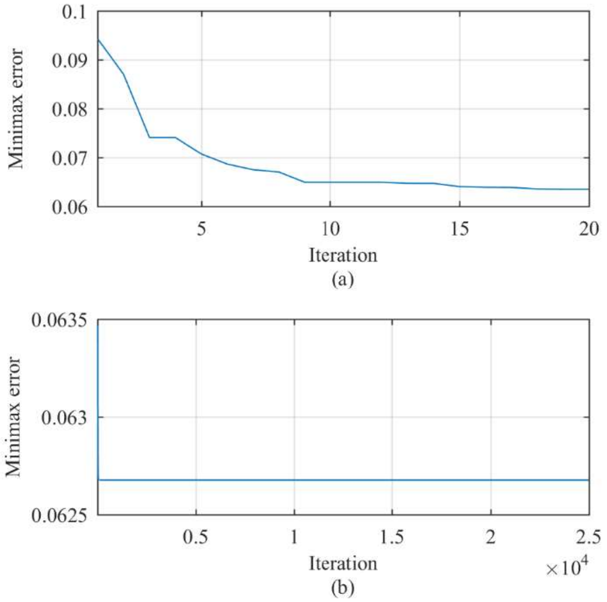


Fig. 3.10. Convergence curve of minimax error for Hilbert transformer design for Type 4 N =13 (a) Interval 1, (b) Interval 2

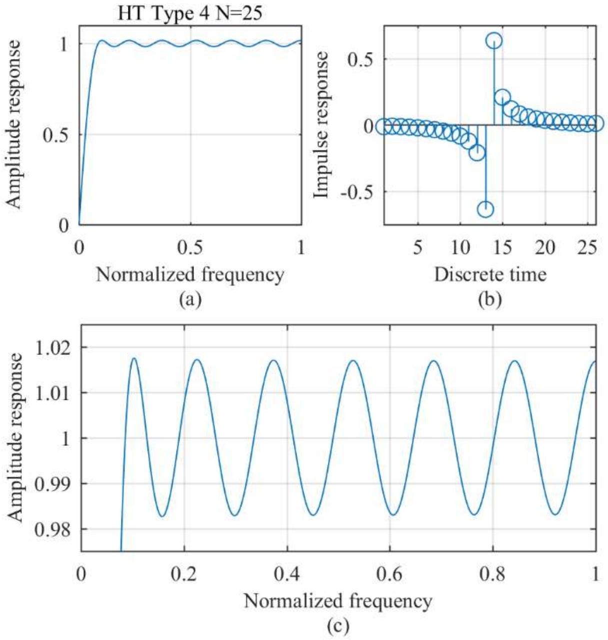


Fig. 3.11. Hilbert transformer minimax design for Type 4 N =25 (a) Amplitude response, (b) Impulse response, (c) Passband amplitude response

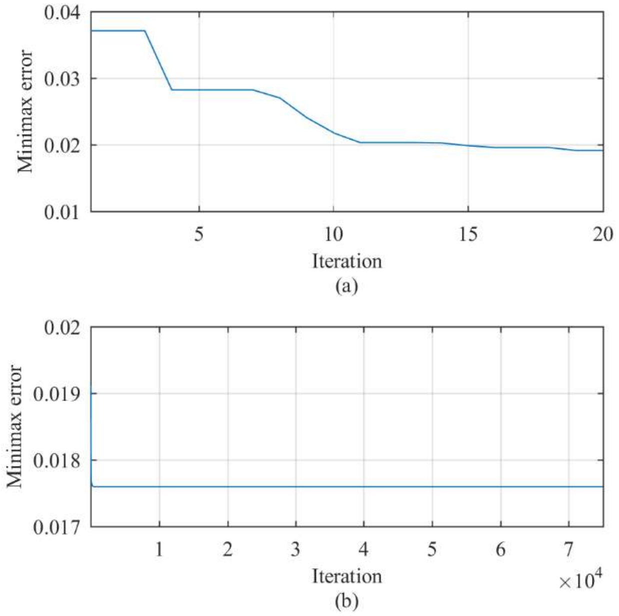


Fig. 3.12. Convergence curve of minimax error for Hilbert transformer design for Type 4 *N* =25 (a) Interval 1, (b) Interval 2

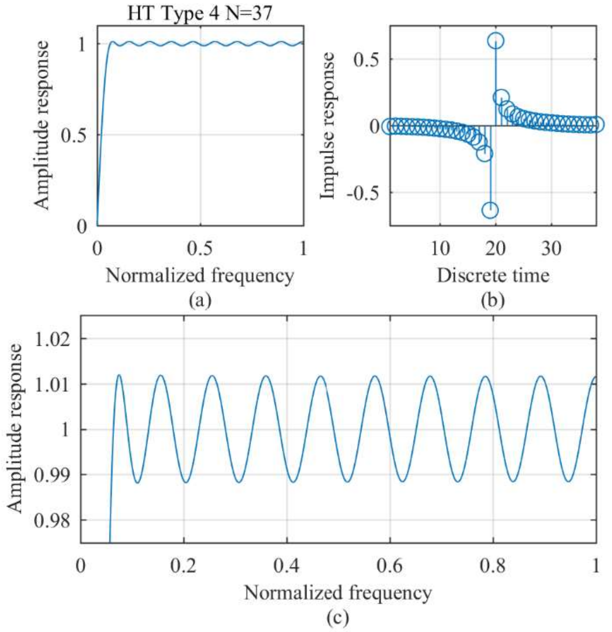


Fig. 3.13. Hilbert transformer minimax design for Type 4 *N* =37 (a) Amplitude response, (b) Impulse response, (c) Passband amplitude response

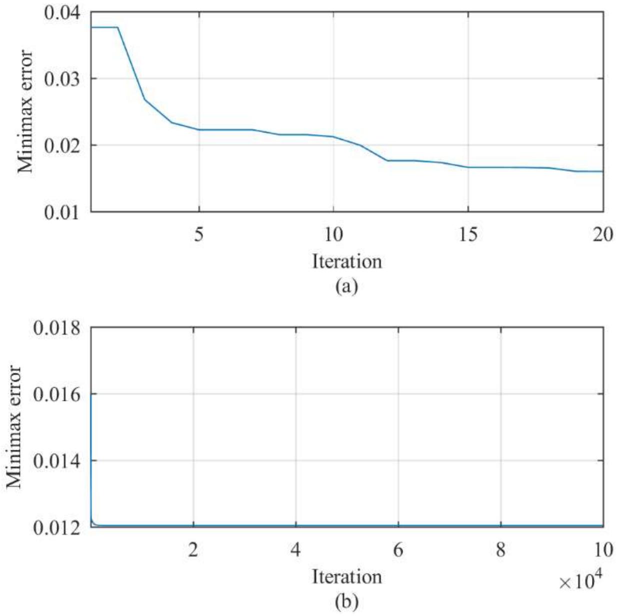


Fig. 3.14. Convergence curve of minimax error for Hilbert transformer design for Type 4 N =37 (a) Interval 1, (b) Interval 2

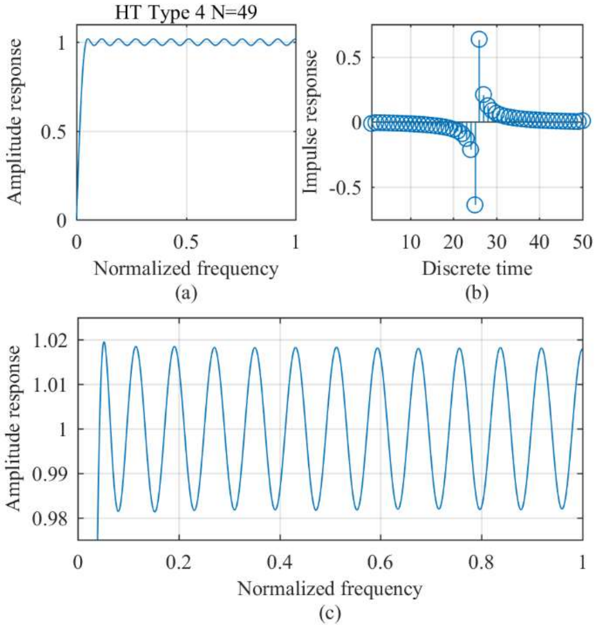


Fig. 3.15. Hilbert transformer minimax design for Type 4 N =49 (a) Amplitude response, (b) Impulse response, (c) Passband amplitude response

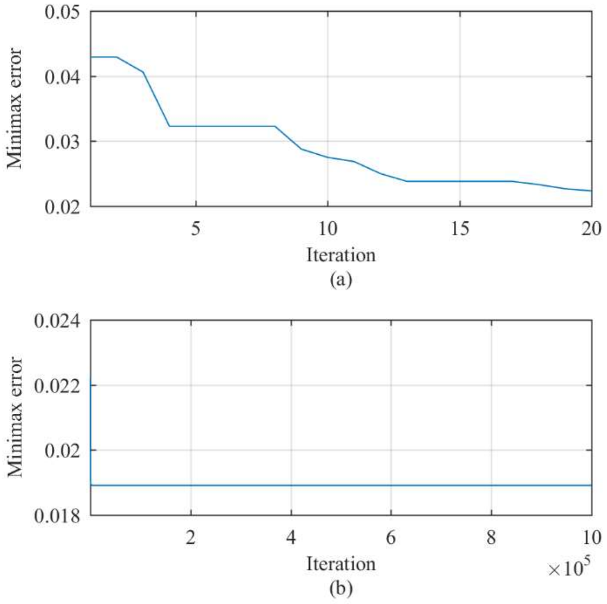


Fig. 3.16. Convergence curve of minimax error for Hilbert transformer design for Type 4 N =49 (a) Interval 1, (b) Interval 2

3.2.3 Type 3 bandpass HT LS designs

Specifications of Type 3 Hilbert transformers and standard TLBO and gradient-based TLBO algorithm parameters are listed in Table 3.1. The passband frequency grid and the passband frequency points for optimization and evaluation are listed in Table 3.2. The parameter μ in gradient-based TLBO algorithm is assumed as $\mu = 0.0001$.

Table 3.9 to Table 3.16 show the Hilbert transformer design results and the corresponding coefficients obtained in term of least square error using gradient-TLBO algorithm as compared to those obtained using original TLBO algorithm, using the least-squares error minimization (by Matlab function firls.m).

The amplitude response, impulse response, and passband amplitude error of Type 3 Hilbert transformer design with four different orders using gradient-based TLBO algorithm are shown in Fig. 3.17 to Fig. 3.24, respectively.

Table 3.9 Hilbert transformer LS Design Results (A: algorithm; 1: firls.m; 2: TLBO algorithm; 3: gradient-based TLBO algorithm; Iter: Iteration; LS: Least square algorithm; C: Converged iteration number; CPU: Time in sec)

A	Least squares error	LS peak error	Iter	C	CPU
1	0.093043692045098	0.102273605696324	-	-	0.06
2	0.095997154035938	0.098726468477840	2,500	468	0.49
3	0.092586981295538	0.098726467261742	2,500	390	0.43

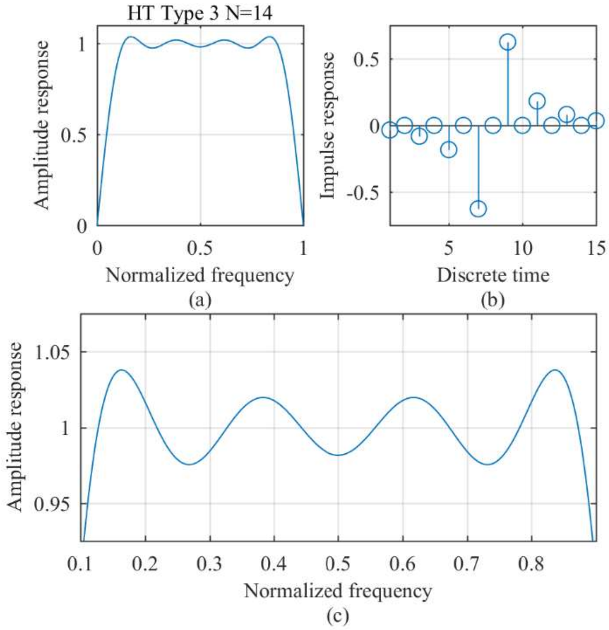


Fig. 3.17 Hilbert transformer least squares design for Type 3 N=14 (a) Amplitude response, (b) Impulse response, (c) Passband amplitude response

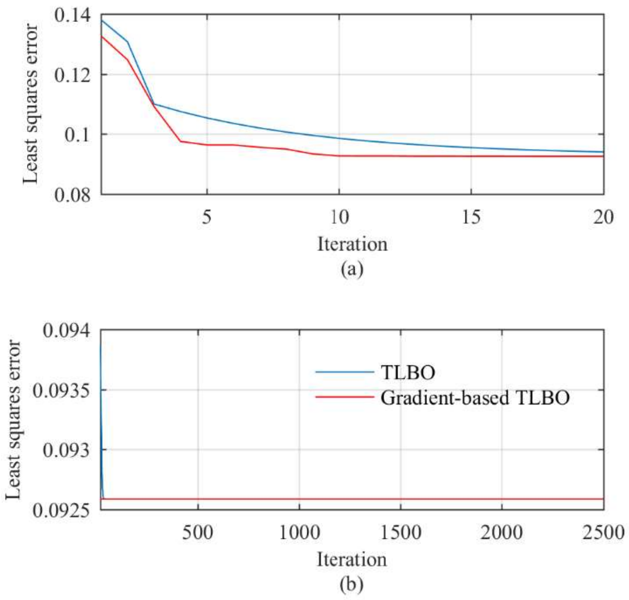


Fig. 3.18 Convergence curve of least squares error for Hilbert transformer design for Type 3 N =14 (a) Interval 1, (b) Interval 2

Table 3.10 Filter Coefficients of T3-N14

h(n)	Gradient-based TLBO	TLBO
h(0)= -h(14)	-0.0347055755819885	-0.0347055751541252
h(1)=- h(13)	-2.52487407105958e-17	0
h(2)=- h(12)	-0.0815794956241600	-0.0815794953886600
h(3)=- h(11)	-1.03809875839792e-17	0
h(4)=- h(10)	-0.182284596152073	-0.182284596151148
h(5)=- h(9)	-3.08439395381962e-17	0
h(6)=- h(8)	-0.626206770896815	-0.626206770813803
h(7)	0	0

Table 3.11 Hilbert transformer LS Design Results (A: algorithm; 1: firls.m; 2: TLBO algorithm; 3: gradient-based TLBO algorithm; Iter: Iteration; LS: Least square algorithm; C: Converged iteration number; CPU: Time in sec)

A	Least squares error	LS peak error	Iter	C	CPU
1	0.036463095254589	0.102391692050880	-	-	0.17
2	0.008196731534691	0.033488078957269	7,500	4,687	2.68
3	0.008196731534693	0.033488078483567	7,500	837	2.46

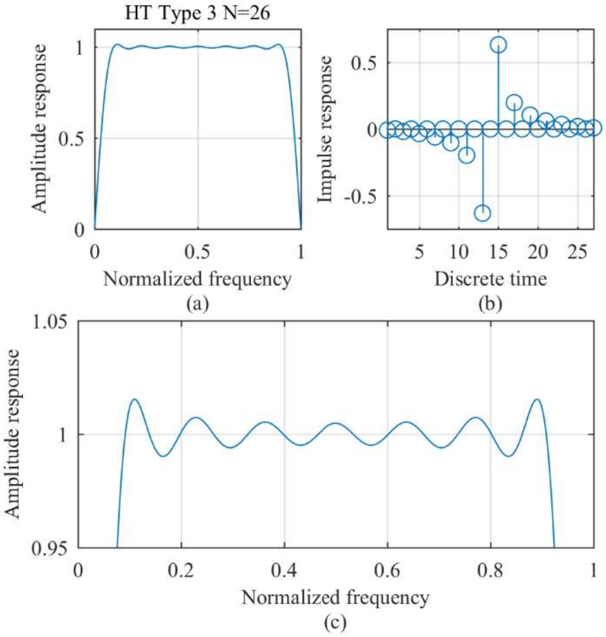


Fig. 3.19 Hilbert transformer least squares design for Type 3 N =26 (a) Amplitude response, (b) Impulse response, (c) Passband amplitude response

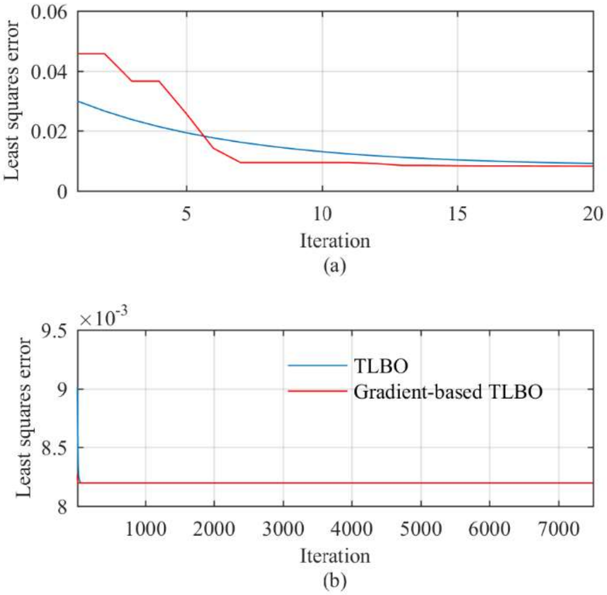


Fig. 3.20 Convergence curve of least squares error for Hilbert transformer design for Type 3 N =26 (a) Interval 1, (b) Interval 2

Table 3.12 Filter Coefficients of T3-N26

h(n)	Gradient-based TLBO	TLBO
h(0)=- h(26)	-0.00867093444496660	-0.0149075046314589
h(1)=- h(25)	-3.98393803867777e-17	0
h(2)=- h(24)	-0.0188226413444431	-0.0221003319022681
h(3)=- h(23)	4.05272966562784e-17	0
h(4)=- h(22)	-0.0349233812103061	-0.0382588890671321
h(5)=- h(21)	3.06294633179801e-17	0
h(6)=- h(20)	-0.0604148311493852	-0.0634815690527765
h(7)=- h(19)	-4.02145384689133e-17	0
h(8)=- h(18)	-0.103957352277172	-0.106428569979422
h(9)=- h(17)	3.26122188437851e-17	0
h(10)=- h(16)	-0.197525091829811	-0.199130267448950
h(11)=- h(15)	-2.67957736667951e-17	0
h(12)=- h(14)	-0.631611719387810	-0.632164551201469
h(13)	0	0

Table 3.13 Hilbert transformer LS Design Results (A: algorithm; 1: firls.m; 2: TLBO algorithm; 3: gradient-based TLBO algorithm; Iter: Iteration; LS: Least square algorithm; C: Converged iteration number; CPU: Time in sec)

A	Least squares error	LS peak error	Iter	C	CPU
1	0.027305696418579	0.097961888212676	-	-	0.12
2	0.003931412924729	0.024318361430496	7,500	5929	3.59
3	0.003931412924729	0.019158132394615	7,500	2670	3.24

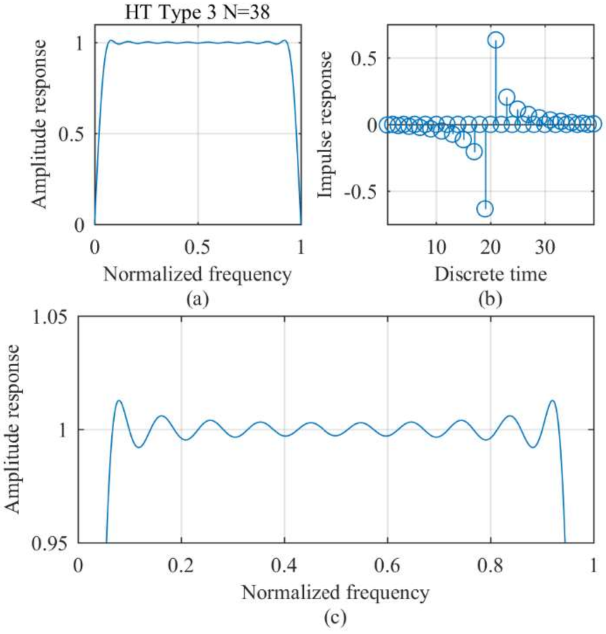


Fig. 3.21 Hilbert transformer least squares design for Type 3 N =38 (a) Amplitude response, (b) Impulse response, (c) Passband amplitude response

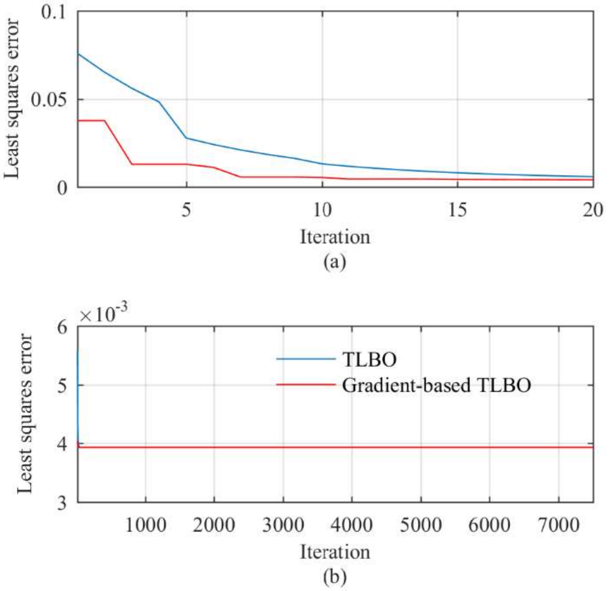


Fig. 3.22 Convergence curve of least squares error for Hilbert transformer design for Type 3 N =38 (a) Interval 1, (b) Interval 2

Table 3.14 Filter Coefficients of T3-N38

h(n)	Gradient-based TLBO	TLBO
h(0)=- h(38)	-0.00450585755598610	-0.00450585748798801
h(1)=- h(37)	4.13688861912307e-17	0
h(2)=- h(36)	-0.00856492884089291	-0.00856492831333441
h(3)=- h(35)	-5.93130524467890e-17	0
h(4)=- h(34)	-0.0143789514946784	-0.0143789512381007
h(5)=- h(33)	3.01510883009364e-17	0
h(6)=- h(32)	-0.0224934579235578	-0.0224934578425080
h(7)=- h(31)	-8.53718991076579e-17	0
h(8)=- h(30)	-0.0337684442056097	-0.0337684439185709
h(9)=- h(29)	2.69287412908123e-17	0
h(10)=- h(28)	-0.0497662949224837	-0.0497662944214584
h(11)=- h(27)	2.73830646514252e-17	0
h(12)=- h(26)	-0.0738148198933766	-0.0738148196770647
h(13)=- h(25)	2.72737448153388e-20	0
h(14)=- h(24)	-0.114627434532271	-0.114627433951268
h(15)=- h(23)	-2.79759347161554e-17	0
h(16)=- h(22)	-0.204398651095386	-0.204398650749998
h(17)=- h(21)	-2.91387265880787e-17	0
h(18)=- h(20)	-0.633984845554837	-0.633984845369711
h(19)	0	0

Table 3.15 Hilbert transformer LS Design Results (A: algorithm; 1: firls.m; 2: TLBO algorithm; 3: gradient-based TLBO algorithm; Iter: Iteration; LS: Least square algorithm; C: Converged iteration number; CPU: Time in sec)

A	Least squares error	LS peak error	Iter	C	CPU
1	0.023120387104181	0.023120387104181	-	-	0.05
2	0.009157825241546	0.039736025210419	7,500	2470	5.41
3	0.009157825241546	0.039736021619859	7,500	700	4.58

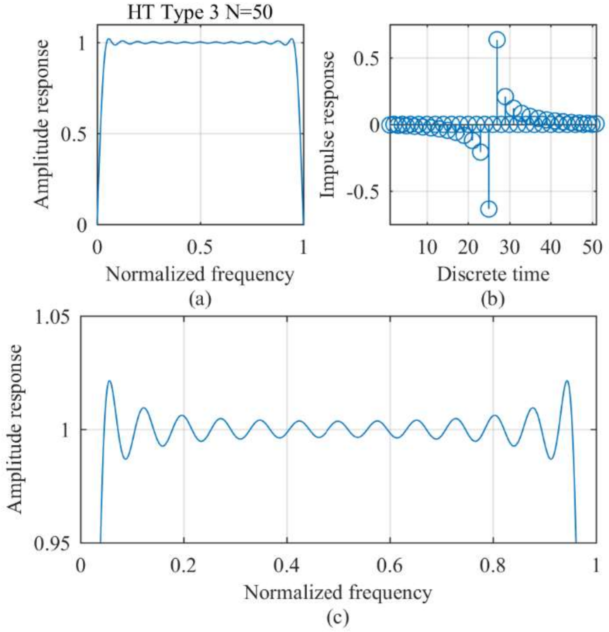


Fig. 3.23 Hilbert transformer least squares design for Type 3 *N* =50 (a) Amplitude response, (b) Impulse response, (c) Passband amplitude response

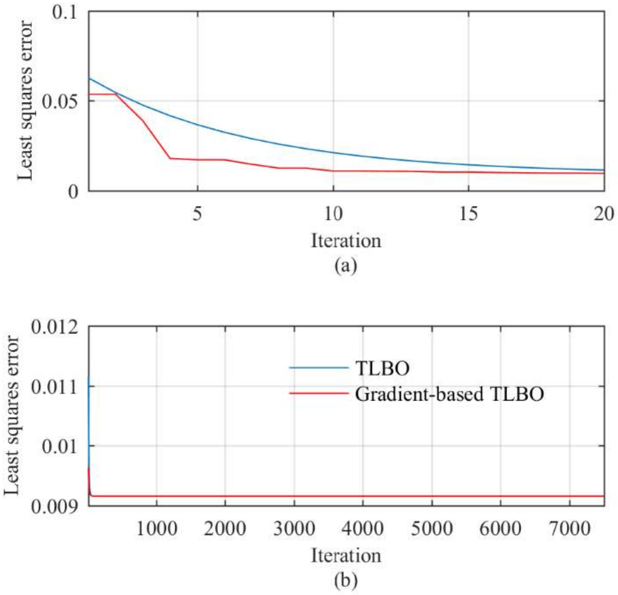


Fig. 3.24 Convergence curve of least squares error for Hilbert transformer design for Type 3 *N* =50 (a) Interval 1, (b) Interval 2

Table 3.16 Filter Coefficients of T3-N50

h(n)	Gradient-based TLBO	TLBO
h(0)=- h(50)	-0.00467224664444162	-0.0122793439331088
h(1)=- h(49)	-3.68833667708071e-17	0
h(2)=- h(48)	-0.00719048289834964	-0.00904980596848222
h(3)=- h(47)	3.45272071110195e-17	0
h(4)=- h(46)	-0.0103889936231642	-0.0162921829702269
h(5)=- h(45)	-5.83034034438175e-17	0
h(6)=- h(44)	-0.0144102265177017	-0.0212247897342324
h(7)=- h(43)	2.72281252917441e-17	0
h(8)=- h(42)	-0.0194528570298624	-0.0275282902044841
h(9)=- h(41)	-5.76370969694656e-17	0
h(10)=- h(40)	-0.0258132394574186	-0.0355726851840625
h(11)=- h(39)	2.84064784387112e-17	0
h(12)=- h(38)	-0.0339656443354446	-0.0461444174658750
h(13)=- h(37)	-8.82086725356694e-17	0
h(14)=- h(36)	-0.0447303936105520	-0.0608727869149715
h(15)=- h(35)	5.35730798834895e-17	0
h(16)=- h(34)	-0.0596662421867969	-0.0831233942211755
h(17)=- h(33)	3.11386297920183e-17	0
h(18)=- h(32)	-0.0821350755208761	-0.121666743609350
h(19) =- h(31)	-3.00830867228597e-17	0
h(20) =- h(30)	-0.120920705644299	-0.121666743609350
h(21) =- h(29)	-2.42816047387683e-17	0
h(22) =- h(28)	-0.208320020984180	-0.208793935251685
h(23) =- h(27)	-2.83929234851383e-17	0
h(24) =- h(26)	-0.635316748130881	-0.635481047083814
h(25)	0	0

3.2.4 Type 4 highpass HT LS designs

Specifications of Type 4 Hilbert transforms and TLBO and gradient-based TLBO algorithm parameters are listed in Table 3.5 and Table 3.6, respectively. The passband frequency grid and the passband frequency points for optimization and evaluation are listed in Table 3.6. The parameter μ in gradient-based TLBO algorithm is assumed as $\mu = 0.0001$.

Table 3.17 to Table 3.24 show the Hilbert transform design results and the corresponding coefficients obtained in term of least square error value using gradient-based TLBO algorithm as compared to those obtained using original TLBO algorithm, using the least-squares error minimization (by Matlab function firls.m).

The amplitude response, impulse response, and passband amplitude error of Type 4 Hilbert transform design with four different orders using gradient-based TLBO algorithm are shown in Fig. 3.25 to Fig. 3.32, respectively.

Table 3.17 Hilbert transformer LS Design Results (A: algorithm; 1: firls.m; 2: TLBO algorithm; 3: gradient-based TLBO algorithm; Iter: Iteration; LS: Least square algorithm; C: Converged iteration number; CPU: Time in sec)

A	Least squares error	LS peak error	Iter	C	CPU
1	0.108848444093498	0.146237503043823	-	-	0.05
2	0.108445670945980	0.140880599458578	2,500	2,214	0.46
3	0.108445670945980	0.140880598427550	2,500	680	0.45

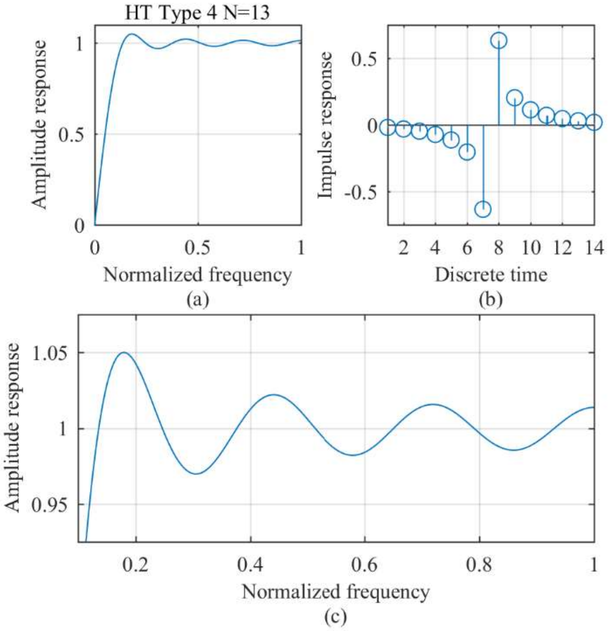


Fig. 3.25 Hilbert transformer least squares design for Type 4 N=13 (a) Amplitude response, (b) Impulse response, (c) Passband amplitude response

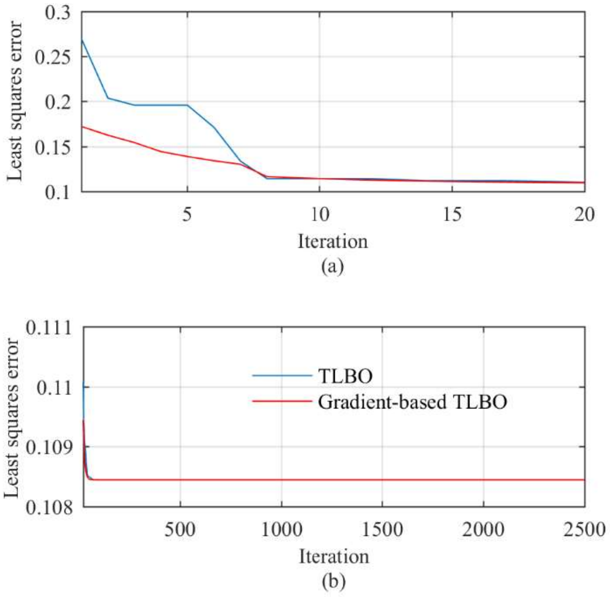


Fig. 3.26 Convergence curve of least squares error for Hilbert transformer design for Type 4 N =13 (a) Interval 1, (b) Interval 2

Table 3.18 Filter Coefficients of T4-N13

h(n)	Gradient-based TLBO	TLBO
h(0)= -h(13)	-0.0190400492050795	-0.0190400489310466
h(1)=- h(12)	-0.0307955992939464	-0.0307955991972672
h(2)=- h(11)	-0.0472028004062931	-0.0472028008777906
h(3)=- h(10)	-0.0718441850018401	-0.0718441848823209
h(4)=- h(9)	-0.113143404277388	-0.113143403689404
h(5)=- h(8)	-0.203556672258847	-0.203556671903639
h(6)=- h(7)	-0.633657560003241	-0.633657560271522

Table 3.19 Hilbert transformer LS Design Results (A: algorithm; 1: firls.m; 2: TLBO algorithm; 3: gradient-based TLBO algorithm; Iter: Iteration; LS: Least square algorithm; C: Converged iteration number; CPU: Time in sec)

A	Least squares error	LS peak error	Iter	C	CPU
1	0.007971674013970	0.051359397816245	-	-	0.04
2	0.007804483323906	0.045368423275444	7,500	5,202	2.81
3	0.007804483323906	0.045368420843728	7,500	1,574	2.81

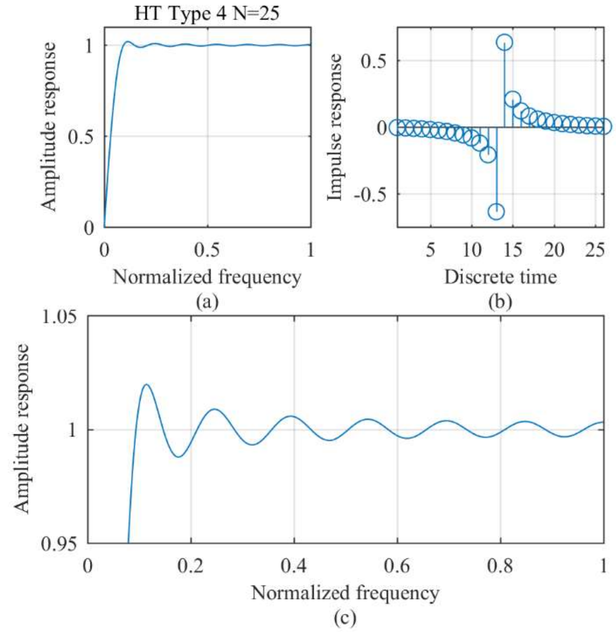


Fig. 3.27 Hilbert transformer least squares design for Type 4 *N* =25 (a) Amplitude response, (b) Impulse response, (c) Passband amplitude response

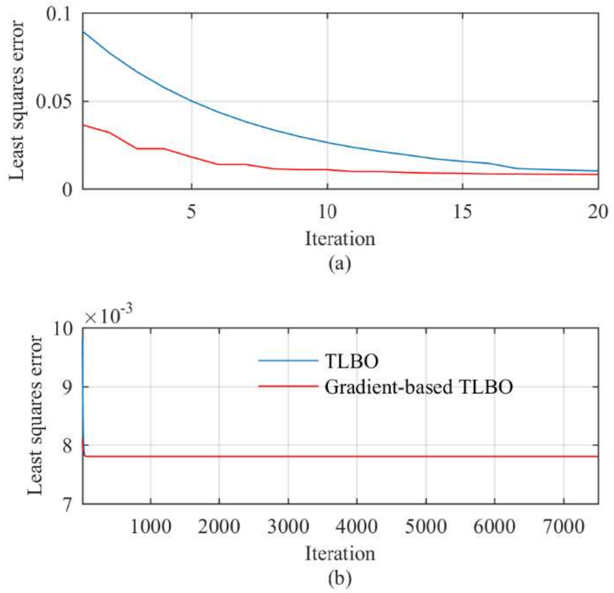


Fig. 3.28 Convergence curve of least squares error for Hilbert transformer design for Type 4 *N* =25 (a) Interval 1, (b) Interval 2

Table 3.20 Filter Coefficients of T4-N25

h(n)	Gradient-based TLBO	TLBO
h(0)=- h(25)	-0.00437952304179973	-0.0122980929236367
h(1)=- h(24)	-0.00686037729448305	-0.00904381464791903
h(2)=- h(23)	-0.00999993496025033	-0.0122426754499805
h(3)=- h(22)	-0.0140054064699285	-0.0162608223821338
h(4)=- h(21)	-0.0190153298905472	-0.0212495504218741
h(5)=- h(20)	-0.0253896604417872	-0.0275285113366878
h(6)=- h(19)	-0.0335411714171256	-0.0355431569734051
h(7)=- h(18)	-0.0443522913754090	-0.0461481794228097
h(8)=- h(17)	-0.0593187294597909	-0.0608712487872964
h(9)=- h(16)	-0.0818629560815587	-0.0831114792422225
h(10)=- h(15)	-0.120704321447705	-0.121631210481234
h(11)=- h(14)	-0.208199065312663	-0.208758262686165
h(12)=- h(13)	-0.635265462243589	-0.635459351922514

Table 3.21 Hilbert transformer LS Design Results (A: algorithm; 1: firls.m; 2: TLBO algorithm; 3: gradient-based TLBO algorithm; Iter: Iteration; LS: Least square algorithm; C: Converged iteration number; CPU: Time in sec)

Alg	Least squares error	LS peak error	Iter	C	CPU
1	0.003275047559394	0.037777852172935	-	-	0.06
2	0.003122479864504	0.030691063670993	7,500	7,109	3.75
3	0.003122479864504	0.030691059673572	7,500	920	3.45

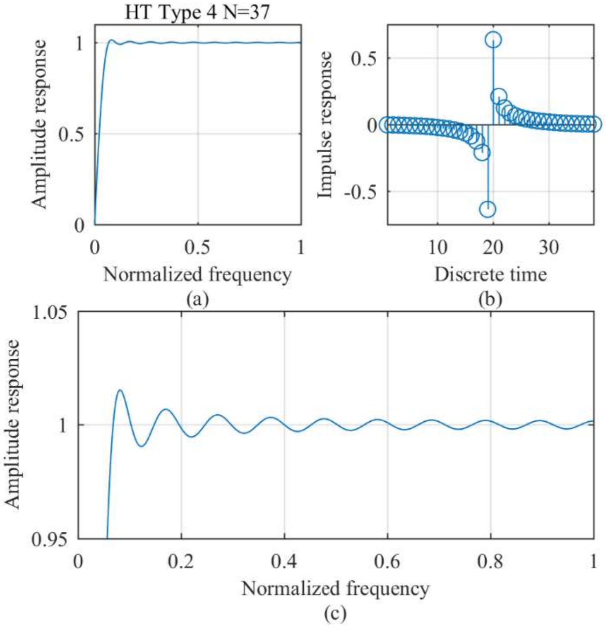


Fig. 3.29 Hilbert transformer least squares design for Type 4 N =37 (a) Amplitude response, (b) Impulse response, (c) Passband amplitude response

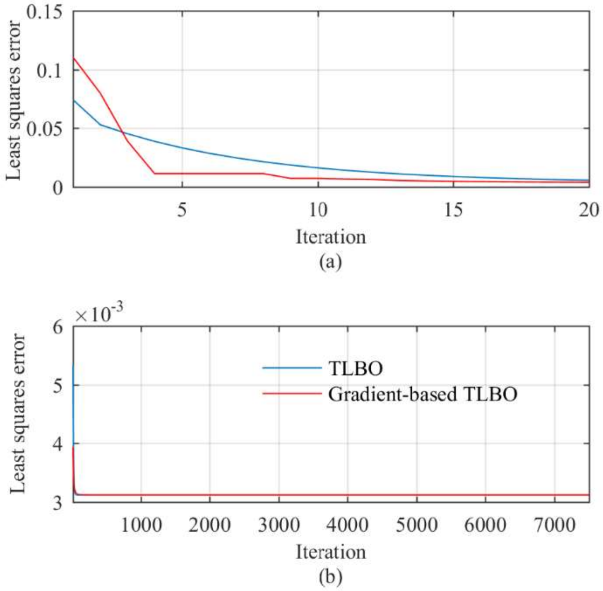


Fig. 3.30 Convergence curve of least squares error for Hilbert transformer design for Type 4 N =37 (a) Interval 1, (b) Interval 2

Table 3.22 Filter Coefficients of T4-N37

h(n)	Gradient-based TLBO	TLBO
h(0)=- h(37)	-0.00223931071519717	-0.00223931047119015
h(1)=- h(36)	-0.00323026238054813	-0.00223931047119015
h(2)=- h(35)	-0.00441445800880022	-0.00441445788697314
h(3)=- h(34)	-0.00585338843626972	-0.00585338841827510
h(4)=- h(33)	-0.00754796779298459	-0.00754796761993759
h(5)=- h(32)	-0.00957236193091621	-0.00957236155267608
h(6)=- h(31)	-0.0119457839694081	-0.0119457837828658
h(7)=- h(30)	-0.0147693206882151	-0.0147693204815458
h(8)=- h(29)	-0.0181026287868733	-0.0181026284637478
h(9)=- h(28)	-0.0221091152517117	-0.0221091148266532
h(10)=- h(27)	-0.0269475642916262	-0.0269475641247959
h(11)=- h(26)	-0.0329455524178297	-0.0329455524795749
h(12)=- h(25)	-0.0405482758784443	-0.0405482757645655
h(13)=- h(24)	-0.0506182539862791	-0.0506182536229811
h(14)=- h(23)	-0.0646959018253678	-0.0646959016339860
h(15)=- h(22)	-0.0861953934994939	-0.0861953933571656
h(16)=- h(21)	-0.123891271924348	-0.123891271910270
h(17)=- h(20)	-0.210140426969042	-0.210140426812869
h(18)=- h(19)	-0.635923368401816	-0.635923368427021

Table 3.23 Hilbert transformer LS Design Results (A: algorithm; 1: firls.m; 2: TLBO algorithm; 3: gradient-based TLBO algorithm; Iter: Iteration; LS: Least square algorithm; C: Converged iteration number; CPU: Time in sec)

A	Least squares error	LS peak error	Iter	C	CPU
1	0.006575264794247	0.058523578652831	-	-	0.04
2	0.006163099412912	0.046388369100268	7,500	1,870	5.49
3	0.006163099412912	0.046388362539238	7,500	679	5.34

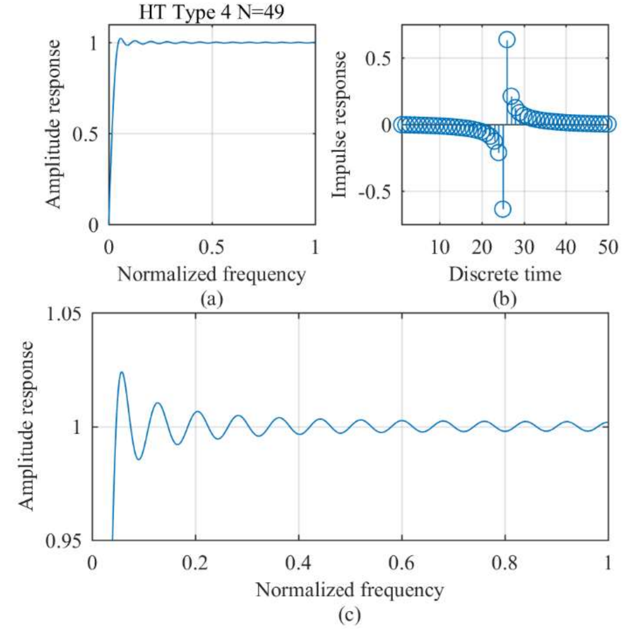


Fig. 3.31 Hilbert transformer least squares design for Type 4 N =49 (a) Amplitude response, (b) Impulse response, (c) Passband amplitude response

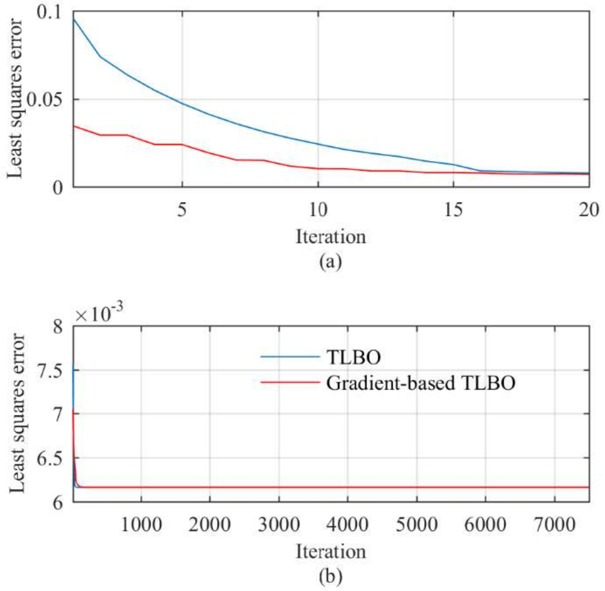


Fig. 3.32 Convergence curve of least squares error for Hilbert transformer design for Type 4 N =49 (a) Interval 1, (b) Interval 2

Table 3.24 Filter Coefficients of T4-N49

h(n)	Gradient-based TLBO	TLBO
h(0)=- h(49)	-0.00235759102200656	-0.0109691636335392
h(1)=- h(48)	-0.00298856980510919	-0.00409689702658207
h(2)=- h(47)	-0.00368365844424425	-0.00476575923951288
h(3)=- h(46)	-0.00448738217280165	-0.00556873883748923
h(4)=- h(45)	-0.00537176740365309	-0.00647457302683714
h(5)=- h(44)	-0.00638394717357022	-0.00748127226699699
h(6)=- h(43)	-0.00749933316901716	-0.00854060640351370
h(7)=- h(42)	-0.00876949717307407	-0.00983665569997877
h(8)=- h(41)	-0.0101757221761658	-0.0112212247847178
h(9)=- h(40)	-0.0117774282414077	-0.0127999429185273
h(10)=- h(39)	-0.0135665325670962	-0.0145663079755267
h(11)=- h(38)	-0.0156170782320765	-0.0165809679086858
h(12)=- h(37)	-0.0179414551933409	-0.0188811149051089
h(13)=- h(36)	-0.0206429844276568	-0.0215227556049619
h(14)=- h(35)	-0.0237769952924307	-0.0246115552392823
h(15)=- h(34)	-0.0275117144630903	-0.0282784808051461
h(16)=- h(33)	-0.0320042190918667	-0.0327095164958862
h(17)=- h(32)	-0.0375895084596627	-0.0382158408582262
h(18)=- h(31)	-0.0447136035802478	-0.0452653174399299
h(19) =- h(30)	-0.0542488552092505	-0.0547169434296792
h(20) =- h(29)	-0.0677387367629846	-0.0681275086221576
h(21) =- h(28)	-0.0886092649221661	-0.0889084862920638
h(22) =- h(27)	-0.125639786925093	-0.125858282370807
h(23) =- h(26)	-0.211200479707128	-0.211327437275774
h(24) =- h(25)	-0.636277768819045	-0.636324442471481

3.2.5 Comparison with references [143]

In [143], the author provides an order-30 FIR bandpass Hilbert transformer design. In this part, the same parameters are selected as [143] shown in Table 3.25 and Table 3.26. Table 3.27 describes the comparisons of least square error with TLBO and gradient-based TLBO algorithm. The amplitude response, impulse response, and error convergence are shown in Fig. 3.33 to Fig. 3.35.

Table 3.25 FIR Filter Parameters (Key: LP-FIR 3: type 3 Hilbert transformer)

Symbol	Description	Type 3
$W(\omega_i)$	Frequency weights for $0 \leq w_i \leq \pi$	1
$c_k^{[U]}$	Upper bound of filter coefficients	0.94
$c_k^{[U]}$	Lower bound of filter coefficients	-0.94
P	Population size	30
N	Order of filter	30
S	Number of distinct coefficients	15
τ	Group delay of LP-FIR3 filter	15

Table 3.26 Passband frequency grid and points for optimization and evaluation

	N	Frequency Grid	$A_d(\omega_i)$
Optimization	30	[0.08:0.005:0.92]	[17, 185]
Peak error evaluation	30	[0.08:0.001:0.92]	[81, 921]

Table 3.27 Least squares errors of 30th-order linear phase FIR Hilbert transformer design (Keys: T: Type; BP: Bandpass; A: Algorithm; 1: firls.m; 2: TLBO; 3: Gradient-based TLBO; CPU: Time in sec; Iter: Iteration; LS: Least squares algorithm; C: Converged iteration number; CPU: Time in sec)

A	Least squares error	LS peak error	Iter	C	CPU
Ref. [143]	0.083875775860366	0.995453860868507	-	-	-
1	0.032201865888951	0.091401172598384	-	-	0.06
2	0.002684004032368	0.019158136245702	7,500	5,290	3.22
3	0.002684004032368	0.019158132394615	7,500	1,953	2.64

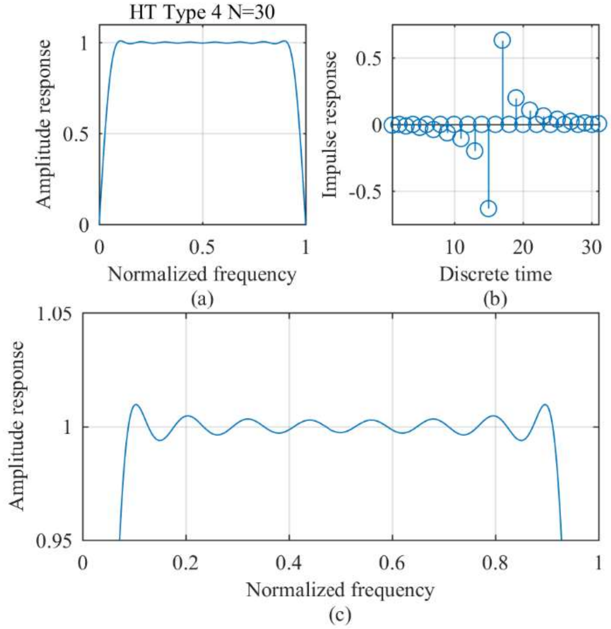


Fig. 3.33 Hilbert transformer least squares design for Type 3 N =30 (a) Amplitude response, (b) Impulse response, (c) Passband amplitude response

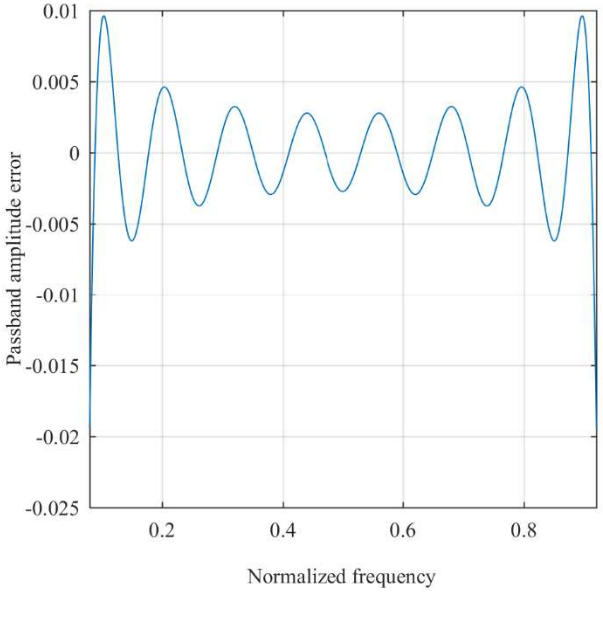


Fig. 3.34 Passband amplitude error of Hilbert transformer design using gradient-based TLBO for Type 3 N =30

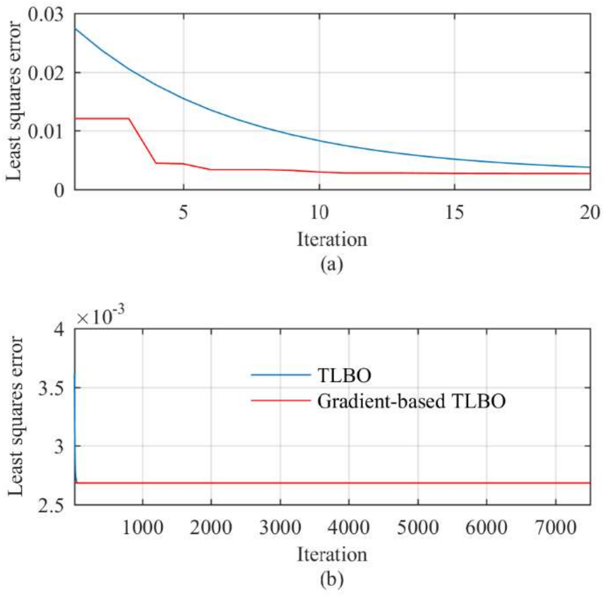


Fig. 3.35 Convergence curve of least squares error for Hilbert transformer design for Type 3 N =30 (a) Interval 1, (b) Interval 2

Table 3.28 Filter Coefficients of T3-N30

h(n)	Gradient-based TLBO	TLBO
h(0)=- h(30)	-0.00520062366889396	-0.00520062315968292
h(1)=- h(29)	3.12232286574276e-17	0
h(2)=- h(28)	-0.0117620171476612	-0.0117620163537757
h(3)=- h(27)	-5.63450496057750e-17	0
h(4)=- h(26)	-0.0222304423011025	-0.0222304414053637
h(5)=- h(25)	5.64178206761915e-17	0
h(6)=- h(24)	-0.0383550775954851	-0.0383550771086034
h(7)=- h(23)	3.14087089604126e-17	0
h(8)=- h(22)	-0.0635439096171635	-0.0635439090967275
h(9)=- h(21)	-2.69749304463702e-17	0
h(10)=- h(20)	-0.106466209215744	-0.106466208379068
h(11)=- h(19)	-1.09007965007902e-17	0
h(12)=- h(18)	-0.199148337246562	-0.199148336838218
h(13)=- h(17)	-2.88206099566837e-17	0
h(14)=- h(16)	-0.632173338685553	-0.632173338552925
h(15)	0	0

3.3 Analysis

The MM design results in Sections 3.2.1 and 3.2.2 reveals that the standard TLBO achieves smaller MM errors and MM peak errors than the Matlab functions firpm.m. Although the TLBO algorithm requires longer time, the Hilbert transformer designed by TLBO can reach lower errors. The number of convergence iterations is usually less than the maximum number of iterations.

Moreover, the LS design results listed in Section 3.2.3 and 3.2.4 indicate that both standard TLBO algorithm and gradient-based TLBO algorithm can obtain smaller LS errors than those of the LS algorithm. The LS errors of gradient-based TLBO algorithm are the same as those of TLBO algorithm. However, their CPU time is longer than that of the LS algorithm. In general, gradient-based TLBO algorithm requires less CPU time than that of the standard TLBO algorithm. From Fig. 3.18, 3.20, 3.22, 3.24, 3.26, 3.28, 3.30 and 3.32, in the beginning 20 iterations, the LS errors of the gradient-based TLBO algorithm decreases faster than that of the standard TLBO, and converges faster than that of the standard TLBO, especially for high orders. All the peak LS errors using gradient-based TLBO are slightly smaller than those that of the standard TLBO. Moreover, the converged iteration numbers of both algorithms are within the maximum iteration numbers.

In addition, the LS design is compared to that of [143]. The LS errors obtained by the gradient-based TLBO and the standard TLBO algorithm are approximately 1/40 of [143], and the peak LS error using the gradient-based TLBO and the standard TLBO is 1/10 smaller than that of [143]. The LS peak error with the gradient-based TLBO is slightly smaller than that with the standard TLBO. The maximum LS error is around 0.013 as shown in Fig. 2 of [143], while that of the gradient-based TLBO is about 0.01 as shown in Fig. 3.34. However, the gradient-based TLBO only requires 1/3 of the converged iteration number of the standard TLBO algorithm, and the CPU time consumed is 20% less than that of TLBO.

3.4 Conclusions

In terms of minimax design of type 3 and type 4 FIR Hilbert transformers, the minimax errors and the peak minimax errors obtained by the standard TLBO algorithm are much better than the results obtained by the Matlab function firpm.m. In terms of least squares design of type 3 and type 4 FIR Hilbert transformers, both the gradient-based TLBO and the standard TLBO algorithms are capable of achieving the smaller least square errors than the least squares algorithm using the Matlab function firls.m. The gradient-based TLBO can result in lower LS peak errors, reduce CPU time and faster convergence in LS designs. In Section 3.2.5, when compared to the type 3 Hilbert transformer designs in [143] and the standard TLBO, the gradient-based TLBO can obtain better and slightly better performances respectively.

In fact, the gradient-based TLBO is an efficient method as compared to the standard TLBO algorithm for designing linear phase FIR digital filters. Moreover, replacing the standard learning phase with the gradient descent optimization can speed up the search ability for faster convergence. However, a tuning parameter is required in the gradient-based learning phase. All the designs were carried out on an Intel Core i7-4810MQ, 2.8 GHz (3.8 G Hz turbo) with 16GB RAM laptop computer.

CHAPTER 4

General FIR Digital Filter Design using Multiobjective TLBO algorithm

Multiobjective optimization (also known as multi-objective programming, vector optimization, multi-criteria optimization, multi-objective optimization or Pareto optimization) is an area of multiobjective decision making, which involves the optimization of multi-objective functions and multi-objective optimization problems. Multi-objective optimization has been applied to many areas including engineering, science, economics, and logistics. When there are trade-offs between two or more conflicting goals, optimal decisions need to be taken. Multiobjective particle swarm optimization (MOPSO) [139]-[142] is a population-based algorithm that uses non-dominated sets to preserve solutions that satisfy requirements. Compared to other swarm intelligence algorithms, Particle Swarm Optimization (PSO) replaces a single search space with global optimal particles for optimization; it uses particle velocity and one-way mutation operations, unlike other swarm intelligence algorithms that use omnidirectional variation.

In this chapter, the multiobjective approach is adopted and combined with TLBO to compose MOTLBO for least-*p*th design of general FIR lowpass, highpass and bandpass filters. Two objective functions, the least-*p*th minimax magnitude error and the least-*p*th group delay error, are considered and design results are given.

The organization of this chapter: Section 4.1 lists the equations of general FIR digital filters and how they can be used to formulate least squares and minimax objective functions. The details of the MOPSO algorithm are listed in Section 4.2, which is to be used as a comparison algorithm. Section 4.3 describes the non-dominated MOTLBO with crowding distance. The proposed gradient-based MOTLBO is described in Section 4.4. The minimax design examples using MOPSO and MOTLBO are given in Section 4.5. In addition, Section 4.6 presents the gradient-based MOTLBO with Manhattan distance and its design results and comparison with the non-dominated MOTLBO.

4.1 General FIR digital filter design

4.1.1 General FIR digital filters

The transfer function of an *N*th-order general FIR filter [80] consists of $(N+1)$ impulse responses which can be expressed as

$$H(z) = \sum_{n=0}^{N} c(n) z^{-n} = \boldsymbol{c}^T \boldsymbol{z}(z) \tag{4.1}$$

The distinct coefficient vector $\mathbf{c}$ is described as

$$\mathbf{c} = [c_0, c_1, c_2, c_3, \dots, c_N]^T \tag{4.2}$$

The frequency response $H(w)$ of a general FIR filter can be evaluated by substituting $z = e^{j\omega t}$ into (4.1)

$$H(\omega) = \sum_{n=0}^{N} c_n z^{-n} |_{z=e^{j\omega T}} = \sum_{n=0}^{N} c_n e^{-j\omega nT}$$

$$= \sum_{n=0}^{N} c_n \cos n\omega T - j \sum_{n=0}^{N} c_n \sin n\omega T = A(\omega) e^{j\theta(\omega)} \tag{4.3}$$

where the magnitude response $A(\omega)$ is equal to

$$A(\omega) = \{[\sum_{n=0}^{N} c_n \cos n\omega T]^2 + [\sum_{n=0}^{N} c_n \sin n\omega T]^2\}^{1/2} \tag{4.4}$$

The phase response $\theta(\omega)$ is equal to

$$\theta(\omega) = -\tan^{-1}\left[\frac{\sum_{n=0}^{N} c_n \sin n\omega T}{\sum_{n=0}^{N} c_n \cos n\omega T}\right] \tag{4.5}$$

The group delay $\tau(\omega)$ of a general FIR filter is given by

$$\tau(\omega) = -\frac{\partial\theta(\omega)}{\partial\omega T} = \frac{1}{1+c^2}\frac{\partial c}{\partial\omega T} \tag{4.6}$$

where

$$\mathrm{c} = \frac{\sum_{n=0}^{N} c_n \sin n\omega T}{\sum_{n=0}^{N} c_n \cos n\omega T} \tag{4.7}$$

$$\frac{\partial c}{\partial \omega T} = \frac{[\sum_{n=0}^{N} c_n \cos n\omega T][\sum_{n=0}^{N} nc_n \cos n\omega T]}{[\sum_{n=0}^{N} c_n \cos n\omega T]^2} + \frac{[\sum_{n=0}^{N} c_n \sin n\omega T][\sum_{n=0}^{N} nc_n \sin n\omega T]}{[\sum_{n=0}^{N} c_n \cos \boldsymbol{n\omega T}]^2} \tag{4.8}$$

4.1.2 Objective Functions

4.1.2.1 LS error

The optimization problem is to search for an optimal coefficient vector $\mathbf{c}$ that minimizes the least squares objective function $e(\mathbf{c})$ with respect to $\mathbf{c}$ as defined [80] by

$$\begin{gathered}\min_{\mathbf{c}} e(\mathbf{c}) = \min_{\mathbf{c}} [\textstyle\sum_{i=1}^{I} W(\omega_i)|H(\mathbf{c}, \omega_i) - H_d(\omega_i)|^2] \\ \text{for } \omega_i \in F_o\end{gathered} \tag{4.9}$$

where F_o is the optimization frequency grid.

For simplicity, the amplitude response is to be optimized in the least squares objective function $e_{amp}(\mathbf{c})$ which can be expressed by

$$\begin{gathered}\min_{\mathbf{c}} e_{amp}(\mathbf{c}) = \min_{\mathbf{c}} [\textstyle\sum_{i=1}^{I} W(\omega_i)|A(\mathbf{c}, \omega_i) - A_d(\omega_i)|^2] \\ \text{for } \omega_i \in F_o\end{gathered} \tag{4.10}$$

where $A_d(\omega_i) = 1$ in passband and $A_d(\omega_i) = 0$ in stopband. F_o denotes the frequency grid for optimization.

Similarly, the objective function of the least squares group delay error $e_{gd}(\mathbf{c})$ in the passband can be expressed by

$$\begin{gathered}\min_{\mathbf{c}} e_{gd}(\mathbf{c}) = \min_{\mathbf{c}} [\textstyle\sum_{i=1}^{I} W(\omega_i)|\tau(\mathbf{c}, \omega_i) - \tau_d(\omega_i)|^2] \\ \text{for } \omega_i \in F_{op}\end{gathered} \tag{4.11}$$

where $\tau_d(\omega)$ is the ideal group delay. F_{op} denotes the frequency grid in passband for optimization.

The goal of the optimization problems for designing general FIR digital lowpass, highpass, bandpass and bandstop filters is to find an optimal coefficient vector **c** which minimizes the objective functions $e_{mag}(\mathbf{c})$ and $e_{gd}(\mathbf{c})$ simultaneously.

4.1.2.2 Minimax error

The optimization problem is to search for an optimal coefficient vector **c** that minimizes the minimax objective function $e(\mathbf{c})$ with respect to **c** as defined in [80] by

$$\min_{\mathbf{c}} e(\mathbf{c}) = \min_{\mathbf{c}}[\textstyle\sum_{i=1}^{I} W(\omega_i)|H(\mathbf{c},\omega_i) - H_d(\omega_i)|^{pp}]^{1/pp} \quad \text{for } \omega_i \in F_o \tag{4.9}$$

where F_o is the frequency grid for optimization.

For simplicity, the amplitude response is to be optimized in the minimax objective function $e_{amp}(\mathbf{c})$ which is expressed by

$$\min_{\mathbf{c}} e_{amp}(\mathbf{c}) = \min_{\mathbf{c}}[\textstyle\sum_{i=1}^{I} W(\omega_i)|A(\mathbf{c},\omega_i) - A_d(\omega_i)|^{pp}]^{1/pp} \quad \text{for } \omega_i \in F_o \tag{4.10}$$

where $A_d(\omega_i) = 1$ in passband and $A_d(\omega_i) = 0$ in stopband.

Similarly, the objective function of the minimax group delay error $e_{gd}(\mathbf{c})$ among the passband can be calculated by

$$\min_{\mathbf{c}} e_{gd}(\mathbf{c}) = \min_{\mathbf{c}}[\textstyle\sum_{i=1}^{I} W(\omega_i)|\tau(\mathbf{c},\omega_i) - \tau_d(\omega_i)|^{pp}]^{1/pp} \quad \text{for } \omega_i \in F_{op} \tag{4.11}$$

where $\tau_d(\omega)$ is the ideal group delay. F_{op} denotes the frequency grid in passband for optimization.

The goal of the optimization problems for designing general FIR digital lowpass, highpass, bandpass and bandstop filters is to find an optimal coefficient vector **c** which minimizes the objective functions $e_{mag}(\mathbf{c})$ and $e_{gd}(\mathbf{c})$ simultaneously.

4.2 MOPSO Algorithm

4.2.1 Basic Concepts

Definition 1 (Pareto Dominance): A vector $\vec{u} = (u_1, u_2, \dots, u_k)$ is said to dominate $\vec{v} = (v_1, v_2, \dots, v_k)$ (denoted by $\vec{u} \preccurlyeq \vec{v}$) if and only if u is partially less than v, i.e., $\forall_i \in \{1, \dots, k\}, u_i \leq v_i \wedge \exists_i \in \{1, \dots, k\}: u_i \leq v_i$.

Definition 2 (Pareto Optimal Set): For a given MOP $\vec{f}(x)$, the Pareto optimal set $(\mathcal{P}^*)$is defined as

$$\mathcal{P}^* := \left\{x \in \Omega \middle| \neg \exists x' \in \Omega \vec{f}(x') \preccurlyeq \vec{f}(x)\right\} \tag{4.18}$$

Definition 3 (Pareto Front): For a given MOP $\vec{f}(x)$ and Pareto optimal set $(\mathcal{P}^*)$, the Pareto front $(\mathcal{PF}^*)$is defined as

$$\mathcal{PF}^* := \left\{\vec{u} = \vec{f} = (f_1(x), \dots, f_k(x)) \middle| x \in \mathcal{P}^*\right\} \tag{4.19}$$

In the general case, it is impossible to find an analytical expression of the line or surface that contains these points. The normal procedure to generate the Pareto front is to compute the feasible points Ω and their corresponding $f(\Omega)$. When there are a sufficient number of these, it is then possible to determine the non-dominated points and to produce the Pareto front.

4.2.2 MOPSO

In order to apply the PSO [142] strategy for solving multi-objective optimization problems, it is obvious that the original scheme has to be modified. Instead, in multi-objective optimization, we aim to find a set of different solutions (the so-called Pareto optimal set). In general, when solving a multi-objective problem, the steps of MOPSO [139] are shown as follows:

Step 1. Initialize the population *POP*, the speed of each particle velocity and the repository *REP*;

Step 2. Evaluate each of the particles in *POP*;

Step 3. Store the positions of the particles that represent non-dominated vectors in *REP*;

Step 4. Generate hypercubes of the search space explored so far, and locate the particles using these hypercubes as a coordinate system where each particle's coordinates are defined according to the values of its objective functions.

Step 5. Initialize the memory of each particle (this memory serves as a guide to travel through the search space. This memory is also stored in the repository): *PBESTS=POP*

Step 6. Do iterations:

Compute the speed of each particle:

$$VEL[i+1] = W \times VEL[i] + C_1 \times (PBESTS[i] - \ POP[i])$$

$$+C_2 \times (REP[h] - POP[i] \tag{4.20}$$

where the inertia weight W takes a value of 0.4; and C_1 and C_2 are random numbers in the range [0 1]; $PBESTS$ is the best position that the particle has had; $REP[h]$ is a value that is taken from the repository; the index h is selected in the following way: those hypercubes containing more than one particle are assigned a fitness equal to the result of dividing any number more than 1 (we used the same number of the size of population in our experiments) by the number of particles that they contain. This aims to decrease the fitness of those hypercubes that contain more particles and it can be seen as a form of fitness sharing. Then, we apply roulette-wheel selection using these fitness values to select the hypercube from which we will take the corresponding particle. Once the hypercube has been selected, we select randomly a particle within such hypercube. $POP[i]$ is the current value of the particle i.

Compute the new positions of the particles by adding the speed produced from the previous step

$$POP[i] = POP[i] + VEL[i] \tag{4.21}$$

1. Maintain the particles within the search space in case they go beyond their boundaries (avoid generating solutions that do not lie on valid search space). When a decision variable exceeds its boundaries, then do two things: 1) the decision variable takes the value of its corresponding boundary (either the lower or the upper boundary) and 2) its velocity is multiplied by (-1) so that it searches in the opposite direction.

2. Evaluate each of the particles in POP.

3. Update the contents of REP together with the geographical representation of the particles within the hypercubes. This update consists of inserting all the currently non-dominated locations into the repository. Any dominated locations from the repository are eliminated in the process. Since the size of the repository is limited, whenever it gets full, we apply a secondary criterion for retention: those particles located in less populated areas of objective space are given priority over those lying in highly populated regions.

4. When the current position of the particle is better than the position contained in its memory, the particle's position is updated using

$$PBESTS[i] = POP[i] \tag{4.22}$$

The criterion to decide what position from memory should be retained is simply to apply Pareto dominance (i.e., if the current position is dominated by the position in memory, then the position in memory is kept; otherwise, the current position replaces the one in memory; if neither of them is dominated by the other, then we select one of them randomly).

5. If not end of iteration, go to step 6, if finish, end iteration.

Pseudocode of MOPSO

```
Begin
Initialize swarm
Initialize leaders in an external archive
Quality (leaders)
G=0
While g<gmax
    For each particle
        Select position
        Mutation
        Evaluation
            Update pbest
     Endfor
     Update leaders in the external archive
     Quality (leaders)
     g++
  Endwhile
Report results in the external archive
End
```

4.3 MOTLBO algorithm based on non-dominated solutions

A description of TLBO can be found in Chapter 2. In MOTLBO, each learner has m objectives to be minimized and the computational steps are listed below:

Step 1: Initialization of all the parameters. In MOTLBO algorithm, a population is composed of P learners, and different design variables (or filter coefficients) are viewed as S different subjects attached to the learners. All the scores of a learner $\boldsymbol{c}_p(t)$ (for $p = 1, 2, \cdots, P$) are evaluated collectively by its objective function values $e_1[\boldsymbol{c}_p(t)]$, $e_2[\boldsymbol{c}_p(t)]$, …, $e_m[\boldsymbol{c}_p(t)]$ (m is the number of objective functions) in an optimization problem. Among the entire class, the best learner is viewed as the teacher $\boldsymbol{c}_{best}(t)$ who teaches knowledge to the learners and improves the mean results of the class.

Step 2: The MO optimization algorithm searches for the best non-dominated solution. Instead, multi-objective optimization usually constructs a Pareto optimal set (also called non-dominated solution set). The non-dominated solution set of a problem with multiple objectives does not consist of a single solution. Such set of teachers is usually stored in a repository (*REP*), which contains the non-dominated solutions found so far. In a non-dominated set, each non-dominated solution corresponds to the output of m objective values. For example, if m=2, $e_{amp}[\boldsymbol{c}_p(t)]$ and $e_{gd}[\boldsymbol{c}_p(t)]$. The crowding distance [144] is selected to be the method to choose top solutions for a non-dominated set. Extreme solutions are those with the largest values of objective function while others are denoted as intermediate solutions. The crowding distance d of each extreme solution is assigned with an infinite value. On the other hand, d for each intermediate solution is determined by the absolute distance of its two neighboring solutions. The solution with the minimum crowding distance is considered as the best solution $\boldsymbol{c}_{best}(t)$.

$$\min_{\boldsymbol{c}} d = \left|e_{mag}(\mathbf{c}_{rep+1}) - e_{mag}(\mathbf{c}_{rep-1})\right| + \left|e_{gd}(\mathbf{c}_{rep+1}) - e_{gd}(\mathbf{c}_{rep-1})\right| \tag{4.23}$$

where *rep* stands for the index of a solution in the repository from 1 to *REP*. And when $rep = 1$ and $rep = REP$, the crowding distance d is infinite.

Moreover, the initialization of population is expressed as

$$\boldsymbol{c}_{p,s}(t) = rand \times (U_s - L_s) \quad \text{(for } s = 1, 2, \cdots, S\text{)} \tag{4.24}$$

where U_s is the upper bound of the filter coefficient for subject s and L_s is the lower bound.

Step 3: Teacher phase:

In non-dominated MOTLBO, each non-dominated solution corresponds to a pair of output values from two objective values. The one with the smallest crowding distance d is selected as the teacher in MOTLBO. This phase is a process that a teacher teaches a class of learners to improve their mean score $m_s(t)$ in each subject s during iteration t.

$$m_s(t) = \frac{1}{P}\sum_{p=1}^{P} \boldsymbol{c}_p(t) \tag{4.25}$$

$m_s(t)$ is the mean result of the learners in a specific subject s in the tth iteration. As the teacher is the most knowledge people and has the best result on that subject. Learners get knowledge from the teacher and the quality of the teacher decides the mean results, so the difference $Diff_s(t)$ between the result of the teacher and mean result of the learners in each subject is expressed as

$$Diff_s(t) = rand * (\boldsymbol{c}_{best,s}(t) - t_f m_s(t)\ ) \tag{4.26}$$

where t_f is the teaching factor which decides the value of mean to be changed, and $rand$ is the random number in the range [0,1]. The value of t_f can be either 1 or 2. The value of t_f is decided randomly as

$$t_f = round[1 + rand] \tag{4.27}$$

$Diff_s(t)$ is added to current score of learners in different subjects to generate new learners:

$$\boldsymbol{c}'_{p,s}(t) = \boldsymbol{c}_{p,s}(t) + Diff_s(t) \tag{4.28}$$

where $\boldsymbol{c}'_p(t)$ is an updated learner of $\boldsymbol{c}_p(t)$.

A selection operator decides the acceptance of an updated mark $\boldsymbol{c}'_{p,s}(t)$ or an existing mark $\boldsymbol{c}_{p,s}(t)$: If $\boldsymbol{c}'_{p,s}(t)$ dominates $\boldsymbol{c}_{p,s}(t)$, $\boldsymbol{c}'_{p,s}(t)$ is added into the population and $\boldsymbol{c}_{p,s}(t)$ is abandoned. If $\boldsymbol{c}'_{p,s}(t)$ is Pareto equal to $\boldsymbol{c}_{p,s}(t)$, then one of them would be chosen randomly. Otherwise, no update is applied to $\boldsymbol{c}_{p,s}(t)$.

Step 4: Learner phase:

Two random learners namely A and B $\boldsymbol{c}_A(t) \neq \boldsymbol{c}_B(t)$ are selected from the entire class, then a judgement is needed: If $\boldsymbol{c}_A(t)$ dominates $\boldsymbol{c}_B(t)$, $\boldsymbol{c}_A(t)$ is updated using (4.29); if $\boldsymbol{c}_B(t)$ dominates $\boldsymbol{c}_A(t)$, $\boldsymbol{c}_A(t)$ is updated using (4.30). If $\boldsymbol{c}_A(t)$ is Pareto-equal to $\boldsymbol{c}_B(t)$, (4.29) and (4.30) have the same probability to be selected to be adopted for update.

$$\boldsymbol{c}''_A(t) = \boldsymbol{c}_A(t) + rand \times \left(\boldsymbol{c}_A(t) - \boldsymbol{c}_B(t)\right) \tag{4.29}$$

$$\boldsymbol{c}''_A(t) = \boldsymbol{c}_A(t) + rand \times \left(\boldsymbol{c}_{\boldsymbol{B}}(t) - \boldsymbol{c}_A(t)\right) \tag{4.30}$$

The algorithm accepts $\boldsymbol{c}''_p(t)$ (*p=1,…,P*) only if it gives a better value of objective function otherwise it keeps the previous solution such that

$$\boldsymbol{c}_p(t) = \begin{cases} \boldsymbol{c}''_p(t) & if\ e_{amp}\left[\boldsymbol{c}''_p(t)\right] < e_{amp}\left[\boldsymbol{c}_p(t)\right] \\ \boldsymbol{c}_p(t) & if\ e_{amp}\left[\boldsymbol{c}''_p(t)\right] \geq e_{amp}\left[\boldsymbol{c}_p(t)\right] \end{cases} \tag{4.31}$$

Step 5：Continue to Step 2 until the maximum number of iterations is reached. If all iterations completed, stop.

4.4 Gradient-based MOTLBO Algorithm

In the gradient-based MOTLBO algorithm, assume there are two objectives, $e_{amp}\left[\boldsymbol{c}_p(t)\right]$ and $e_{gd}\left[\boldsymbol{c}_p(t)\right]$, which are required to be optimized simultaneously. The Manhattan distance incorporates gradient-based MOTLBO algorithm to approach the optimal equilibration and select a unique optimal solution.

The current ideal points $\boldsymbol{ideal}(t)$ of the two objective functions can be expressed by

$$\boldsymbol{ideal}(t) = \left[ideal_{amp}^{min}(t), ideal_{gd}^{min}(t)\right] \tag{4.32}$$

where

$$\begin{cases} ideal_{amp}^{min}(t) = \min\left\{ \underset{p=1\text{to}P}{min} e_{amp}\left[\boldsymbol{c}_p(t)\right], ideal_{amp}^{min}(t-1) \right\} \\ ideal_{gd}^{min}(t) = \min\left\{ \min_{p=1\text{to}P} e_{gd}\left[\boldsymbol{c}_p(t)\right], ideal_{gd}^{min}(t-1) \right\} \end{cases} \tag{4.33}$$

During the current iteration, $\boldsymbol{ideal}(t)$ is a pair of numeric values updated iteratively by any new-found minimum values of $e_{amp}[\boldsymbol{c}(t)]$ and $e_{gd}[\boldsymbol{c}(t)]$. However, $\boldsymbol{ideal}(0)$ should be initialized by a pair of positive infinite values.

The Manhattan distance between the two objective functions and their ideal points is defined by

$$d(t) = \left| e_{amp}[\boldsymbol{c}(t)] - ideal_{amp}^{min}(t) \right| + \left| e_{gd}[\boldsymbol{c}(t)] - ideal_{gd}^{min}(t) \right| \tag{4.34}$$

Therefore, the Manhattan distance can be used to formulate a joint objective function as

$$e[\boldsymbol{c}(t)] = d(t) \tag{4.35}$$

The operations of the gradient-based MOTLBO is divided into two phases, the teacher phase and the gradient-based phase. The detailed steps are described in the following:

1. Teacher phase (same as Section 2.4 from Eqs. (2.9) to (2.12))

2. Gradient-based learning phase

The learner phase is not efficient enough due to random selection approach in the original TLBO algorithm. For this reason, a gradient-based phase to speed up the search is formulated as

$$\boldsymbol{c}_p''(t) = \boldsymbol{c}_p(t) - \mu \times \frac{\partial e\left[\boldsymbol{c}_p(t)\right]}{\partial \boldsymbol{c}_p(t)} \tag{4.36}$$

According to the Limit Theory [206], the partial derivation of an objective function against a solution can be approximated by

$$\frac{\partial e\left[\boldsymbol{c}_p(t)\right]}{\partial c_{p,s}(t)} = \lim_{\Delta c \to 0} \frac{e\left[\boldsymbol{c}_p(t) + \Delta \boldsymbol{c}\right] - e\left[\boldsymbol{c}_p(t)\right]}{\delta} \tag{4.37}$$

$$\begin{cases} \Delta c_j = \delta & j = s \\ \Delta c_j = 0 & j \neq s \end{cases} \tag{4.38}$$

where μ stands for the step size. $c_{p,s}(t)$ (for $s = 1, 2, \cdots, S$) is the score of p learner in a specific subject s. δ is a very small increment.

To obtain the optimal result, the learner $\boldsymbol{c}_p(t)$ is selected by

$$\boldsymbol{c}_p(t) = \begin{cases} \boldsymbol{c}''_p(t) & if\ e\left[\boldsymbol{c}''_p(t)\right] < e\left[\boldsymbol{c}_p(t)\right] \\ \boldsymbol{c}_p(t) & if\ e\left[\boldsymbol{c}''_p(t)\right] \geq e\left[\boldsymbol{c}_p(t)\right] \end{cases} \tag{4.39}$$

At the end of each iteration, if the joint error function value of the best learner is smaller than that of the teacher, the teacher is replaced by the best learner.

3. Elitist replacement

To avoid duplicate solutions, any duplicate solution within a population is replaced by

$$\boldsymbol{c}_p(t) = \boldsymbol{c}_{best}(t) + rand(U_s - L_s) \tag{4.40}$$

For a faster convergence, an elite mechanism is included in the gradient-based MOTLBO algorithm. At the end of each iteration, the worst learner is replaced by the $\boldsymbol{c}_{best}(t)$.

4.5 General FIR digital filter design examples

In this section, the population size of all designs is 24 the same as the filter order. During the optimization process, 201 and 1001 frequency points are used for calculating peak magnitude and group delay errors of general FIR lowpass (LP), highpass (HP), bandpass (BP) and bandstop (BS) filters. The corresponding parameters are listed in Tables 4.1 to 4.3 for two examples in Section 4.5.1 and 4.5.2. Table 4.1 shows the parameters of general FIR filter design and the MO algorithms. Table 4.2 lists the lower and upper bounds of filter coefficients. All the initial filter coefficients are randomly generated between the lower and upper bounds and a maximum of 5,000,000 iterations is used for each least squares optimization design. The cutoff frequencies of different filters are listed in Table 4.3.

Table 4.1 General FIR filter and MO parameters

Symbol	Description	Value
$W(\omega_i)$	Frequency weights for $0 \leq \omega_i \leq \pi$	1
pp	Least pth-order	128
P	Population size	24
REP	Number of repository	25

Table 4.2 General FIR filter coefficient lower and upper limits

Symbol	Lower limit	Upper limit
LP	-0.095492965855137	0.45
HP	-0.471586444919843	0.675
BP	-0.333461973118477	0.375
BS	-0.183685705580247	1.275

Table 4.3 Cutoff frequencies (LP: Lowpass; HP: Highpass; BP: Bandpass; BS: Bandstop; PB: Passband; SB: Stopband; Gd: Group delay)

	PB_1	SB_1	PB_2	SB_2	Gd
LP	0.3 π	0.4 π	-	-	10
HP	0.55 π	0.45 π	-	-	10
BP	0.35 π	0.25 π	0.6 π	0.7 π	10
BS	0.3 π	0.4 π	0.65 π	0.55 π	10

4.5.1 Minimax design using non-dominated MOTLBO

In this section, minimax LP, HP, BP and BS filters of order *N*=24 and group delay=10 are designed using non-dominated MOTLBO with crowding distance. MOPSO with non-dominated selection and crowding distance is adopted for comparison. Magnitude errors and group delay errors form the two objectives. A maximum of 500,000 iterations is used for each of the minimax filter design problems.

Tables 4.4 summarizes the minimax magnitude errors and group delay errors of the LP, HP, BP and BS general FIR filters obtained by the non-dominated

MOTLBO and the MOPSO. In addition, the peak passband and stopband magnitude errors and peak group delay errors are listed in Tables 4.5 and 4.6. The magnitude response, group delay, impulse response, and passband and stopband errors of LP, HP, BP and BS general FIR filter design results are shown in Figs. 1 to 14, respectively.

Table 4.4 General FIR filter minimax errors (Alg: 1. MOTLBO; 2: MOPSO; MM_mag: Minimax magnitude error; MM_gd: Minimax group delay error; CPU: Time in sec)

	Alg	MM_mag	MM_gd	CPU
LP	1	0.051595993269366	0.022106423454150	3,472
	2	0.286879169754300	0.127985819390463	7,507
HP	1	0.059132967245666	0.033338187094149	3,562
	2	0.546975870769137	1.02964026741799	5,349
BP	1	0.062515715877254	0.025908371119439	7,805
	2	0.475582031018334	0.630372035638652	5,379
BS	1	0.086730527282005	0.097147985900193	47,034
	2	0.293244261014239	1.774471424988321	7,425

Table 4.5 General FIR filter minimax peak errors (Alg: 1. MOTLBO; 2: MOPSO; PB: passband; Gd: Group delay)

	Alg	Peak PB 1 error	Peak Gd error
LP	1	0.050626348035382	0.021715013424096
	2	0.277311764528364	0.126951332248135
HP	1	0.057369111479355	0.033191488714319
	2	0.530321346844630	1.006905686764069
BP	1	0.061190723685223	0.025589300457492
	2	0.463250139769333	0.619926487815061
BS	1	0.084044222113074	0.095921756102353
	2	0.285545315559481	1.730652966475295

Table 4.6 General FIR filter minimax peak errors (Alg: 1. MOTLBO; 2: MOPSO; PB: passband; SB: stopband)

	Alg	Peak SB 1 error	Peak SB 2 (for BP)/ PB 2 (for BS) error
LP	1	0.050266946168661	-
	2	0.279423454354376	-
HP	1	0.058224796835638	-
	2	0.531280509994558	-
BP	1	0.059902840101656	0.060400018311995
	2	0.459582445772840	0.463037787995851
BS	1	0.085021564089306	0.085095000530041
	2	0.286864126013092	0.272406694529927

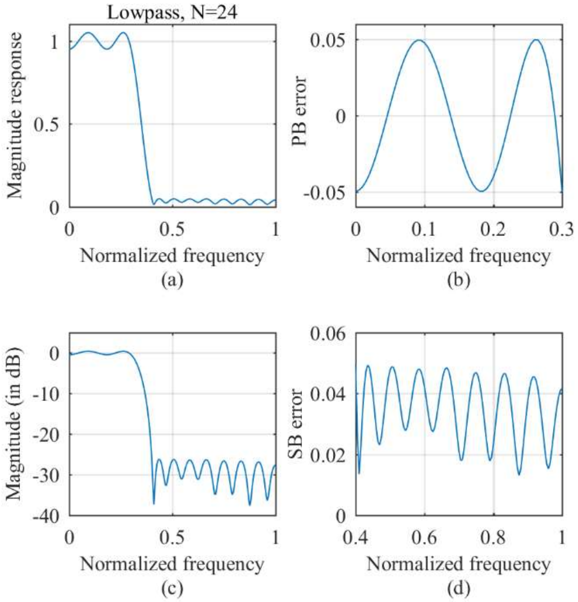


Fig. 4.1 G-FIR LP filter MM design for $N = 24$ and group delay=10 (a) Magnitude response, (b) Passband magnitude error, (c) Magnitude response in dB, (d) Stopband magnitude error

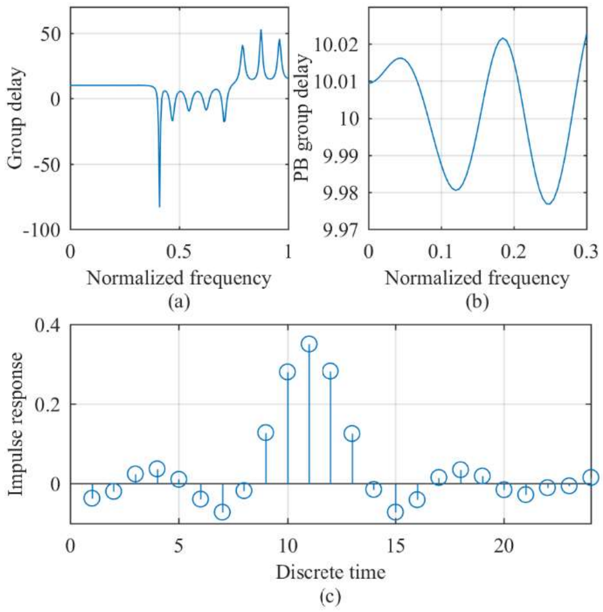


Fig. 4.2 G-FIR LP filter MM design for $N = 24$ and group delay=10 (a) Group delay, (b) Passband group delay, (c) Impulse response

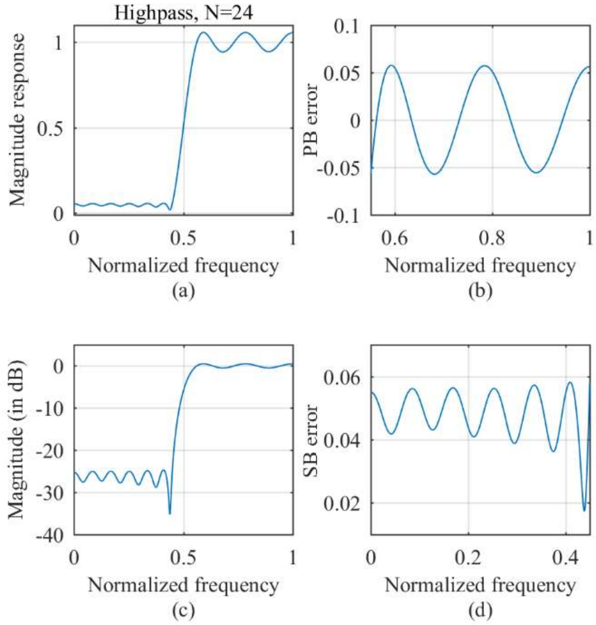


Fig. 4.3 G-FIR HP filter MM design for $N = 24$ and group delay=10 (a) Magnitude response, (b) Passband magnitude error, (c) Magnitude response in dB, (d) Stopband magnitude error

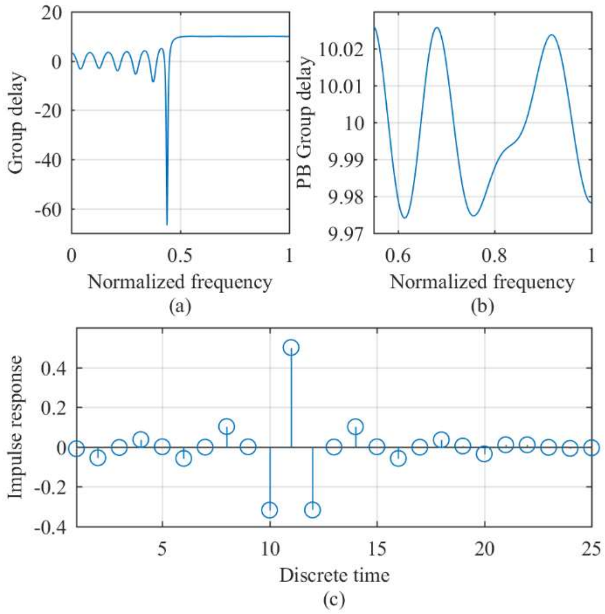


Fig. 4.4 G-FIR HP filter MM design for $N = 24$ and group delay=10 (a) Group delay, (b) Passband group delay, (c) Impulse response

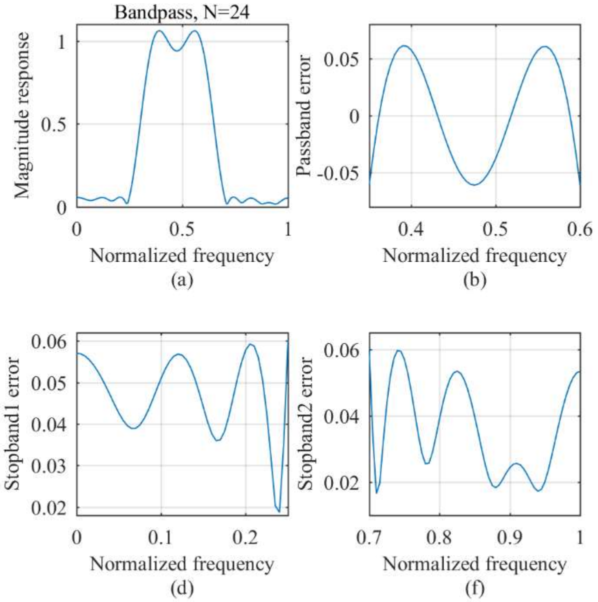


Fig. 4.5 G-FIR BP filter MM design for $N = 24$ and group delay=10 (a) Magnitude response, (b) Passband magnitude error, (c) Stopband_1 magnitude error, (d) Stopband_2 magnitude error

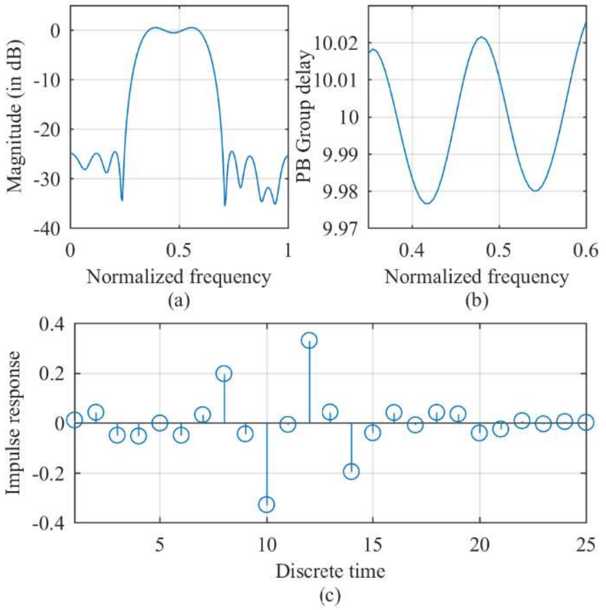


Fig. 4.6 G-FIR BP filter MM design for $N = 24$ and group delay=10 (a) Magnitude response in dB, (b) Group delay, (c) Impulse response

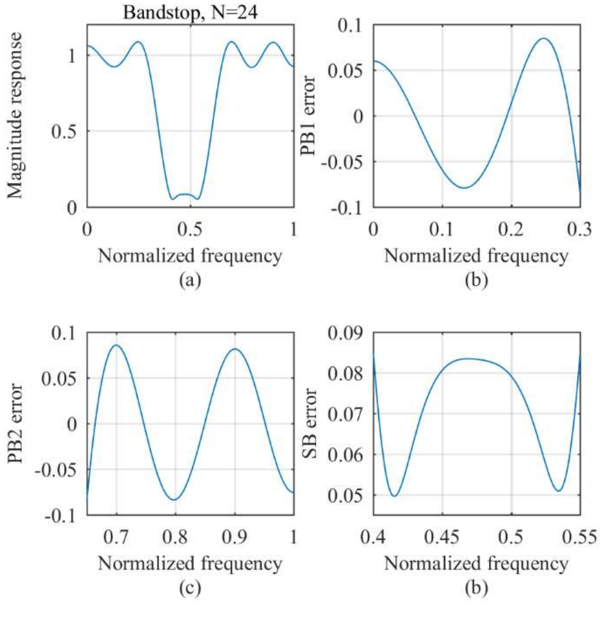


Fig. 4.7 G-FIR BS filter MM design for $N = 24$ and group delay=10 (a) Magnitude response, (b) Passband_1 magnitude error, (c) Passband_2 magnitude error, (d) Stopband magnitude error

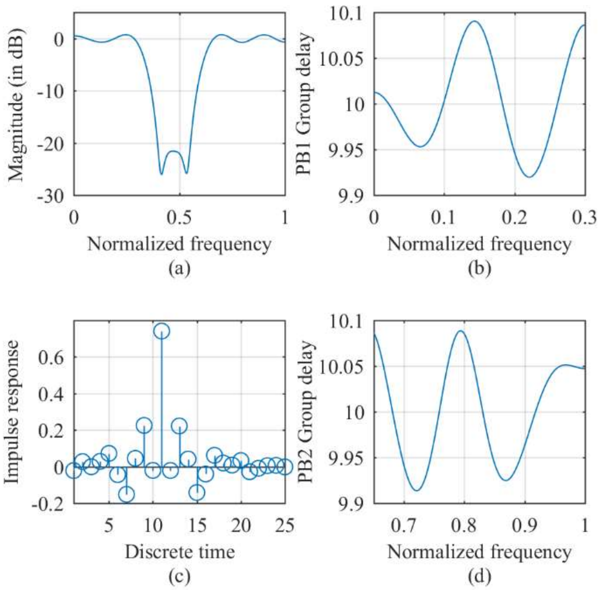


Fig. 4.8 G-FIR BS filter MM design for $N = 24$ and group delay=10 (a) Magnitude response in dB, (b) Passband_1 group delay, (c) Impulse response, (d) Passband_2 group delay

In this section, least-*p*th minimax design of general FIR filters using the non-dominated MOTLBO with crowding distance has been presented. As compared to the non-dominated MOPSO with crowding distance, the results in Tables 4.4 to 4.6 indicate that the MOTLBO can obtain smaller magnitude and group delay minimax errors as well as passband and stopband peak errors. Moreover, except BS filter design, the MOTLBO requires less CPU time than the MOPSO.

4.5.2 Least squares design with gradient-based MOTLBO

In this section, LP, HP, BP and BS filters of order *N*=24 and group delay=10 are designed using the gradient-based MOTLBO with Manhattan distance, the non-dominated MOTLBO with crowding distance, and the non-dominated MOPSO with crowding distance. Magnitude error and group delay error are selected to be the two objectives. The parameters of FIR filters and the parameters for this MO optimization problem are listed in Tables 4.1 to 4.3.

The population size of all designs is set to be 24 which is the same as the order. Optimization frequency points are 201 and 1001 frequency points are used for calculating peak magnitude and group delay errors of LP, HP and BP filters. The initialization is random selected between the two boundaries as listed in Table 4.2, respectively. A maximum of 5,000,000 iterations is used for each design. The parameter μ for the gradient-based MOTLBO algorithm is taken as $\mu = 0.0001$.

Tables 4.7 to 4.16 summarize the LP, HP and BP general FIR filter design results and the filter coefficients obtained by the three MO algorithms. The magnitude response, group delay, impulse response, and passband and stopband errors of LP, HP and BP general FIR filters results are shown in Figs. 4.9 to 4.11, respectively.

Table 4.7 Least squares errors of general lowpass FIR filter (Alg: 1. Gradient-based TLBO; 2: MOTLBO; 3: MOPSO; LS_mag: Least squares magnitude error; LS_gd: Least squares group delay error; CPU: Time in sec)

Alg	LS_mag	LS_gd	CPU
1	0.339986772891189	0.003339806954644	37,496
2	0.341240087075854	0.065774675197247	17,597
3	0.355283043410460	0.046855035402898	87,115

Table 4.8 General lowpass FIR filter least squares peak errors (Alg: 1: gradient-based MOTLBO; 2: MOTLBO; 3: MOPSO; PB: passband; Gd: Group delay)

Alg	Peak PB error	Peak SB error	Peak Gd error
1	0.098923060961945	0.105709664581663	0.008468220489736
2	0.096561668409053	0.109941543296126	0.055469679919591
3	0.083446268186972	0.118340306220507	0.049339320031688

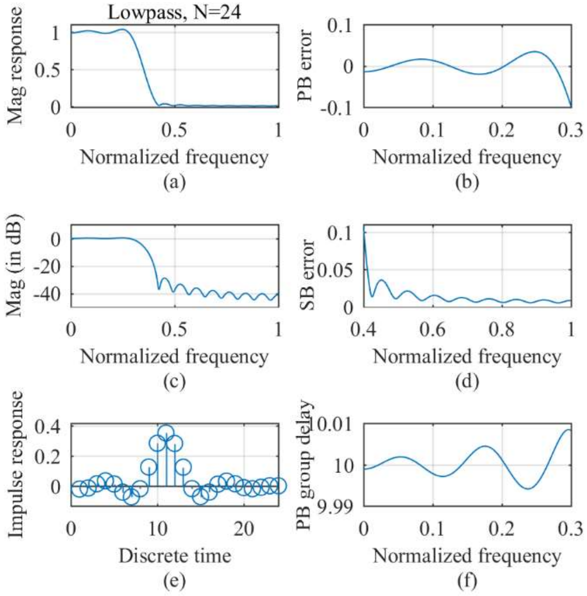


Fig. 4.9 G-FIR LP filter LS design for $N = 24$ and group delay=10 (a) Magnitude response, (b) Passband magnitude error, (c) Magnitude response in dB, (d) Stopband magnitude error, (e) Impulse response, (f) Passband group delay

Table 4.9 Gradient-based MOTLBO LS coefficients of LP

h(n)	Gradient-based MOTLBO
h(0)	-0.0188481016121670
h(1)	-0.0118487854225372
h(2)	0.0158723221659257
h(3)	0.0352676896020524
h(4)	0.0147029130415978
h(5)	-0.0393185504337924
h(6)	-0.0705052053953384
h(7)	-0.0167344424963207
h(8)	0.126306294307527
h(9)	0.283095036294402
h(10)	0.350678013750484
h(11)	0.281812506353807
h(12)	0.125337968066907
h(13)	-0.0161373613406683
h(14)	-0.0690048560923351
h(15)	-0.0387810623326952
h(16)	0.0134547627378284
h(17)	0.0336351340055623

h(18)	0.0160923704146386
h(19)	-0.00936139156938863
h(20)	-0.0166430128929620
h(21)	-0.00692678468109317
h(22)	0.00284759132966295
h(23)	0.00260800086687530
h(24)	-0.00196545274506152

Table 4.10 General FIR highpass filter least squares errors (Alg: 1. Gradient-based TLBO; 2: MOTLBO;3: MOPSO; LS_mag: Least squares magnitude error; LS_gd: Least squares group delay error; CPU: Time in sec)

Alg	LS_mag	LS_gd	CPU
1	0.506943634650731	0.009410451539491	40,240
2	0.527710094389086	0.255725430656471	30,197
3	0.984043985459745	0.559220788314725	41,096

Table 4.11 General highpass FIR filter least sqaures peak errors (Alg: 1: gradient-based MOTLBO; 2: MOTLBO; 3: MOPSO; PB: passband; Gd: Group delay)

Alg	Peak PB 1 error	Peak SB 1 error	Peak Gd error
1	0.116101931533324	0.124568608616491	0.012601347881148
2	0.118433961627229	0.110767537720580	0.074760162080670
3	0.187417661664103	0.055000986489871	0.105304936636333

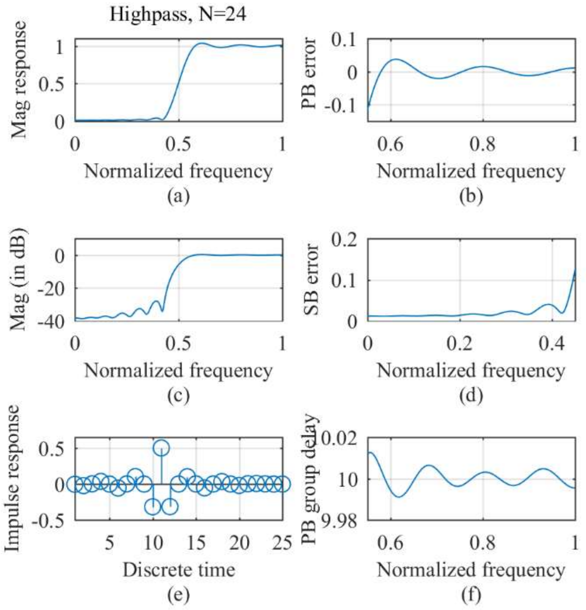


Fig. 4.10 G-FIR HP filter LS design for $N = 24$ and group delay=10 (a) Magnitude response, (b) Passband magnitude error, (c) Magnitude response in dB, (d) Stopband magnitude error, (e) Impulse response, (f) Passband group delay

Table 4.12 Gradient-based MOTLBO LS coefficients of HP

h(n)	Gradient-based MOTLBO
h(0)	-0.00438110784052637
h(1)	-0.0228802627004948
h(2)	0.00188262754246727
h(3)	0.0353357739742804
h(4)	-0.00147844609826433
h(5)	-0.0563715910294968
h(6)	0.00118192482713300
h(7)	0.102090050567776
h(8)	-0.00104422073547493
h(9)	-0.317538718160422
h(10)	0.500816620502163
h(11)	-0.315517067858937
h(12)	-0.000560381060992884
h(13)	0.0999374101747395
h(14)	0.000327323766010231
h(15)	-0.0541738214748310
h(16)	-3.88694886111156e-05

h(17)	0.0330149902970949
h(18)	-0.000699483434070693
h(19)	-0.0202715491837200
h(20)	0.00297047095131086
h(21)	0.00642388117827076
h(22)	0.00144005141526301
h(23)	-0.00171242512335909
h(24)	-0.00134063685217593

Table 4.13 General bandpass FIR filter least squares errors (Alg: 1. Gradient-based TLBO; 2: MOTLBO;3: MOPSO; LS_mag: Least squares magnitude error; LS_gd: Least squares group delay error; CPU: Time in sec)

Alg	LS_mag	LS_gd	CPU
1	0.568618874845898	0.006468893497583	35,085
2	0.611472805674160	0.262731869483094	32,888
3	0.637558405690025	0.011288715005017	43,047

Table 4.14 General bandpass FIR filter least squares peak errors (Alg: 1: gradient-based MOTLBO; 2: MOTLBO; 3: MOPSO; PB: passband; Gd: Group delay)

Alg	Peak PB 1 error	Peak Gd error
1	0.128964004999802	0.008167557928356
2	0.113264581394086	0.069866156083149
3	0.130050320858247	0.010250159973689

Table 4.15 General bandpass FIR filter least squares peak errors (Alg: 1: gradient-based MOTLBO; 2: MOTLBO; 3: MOPSO; SB 1: stopband_1; SB 2: stopband_2)

Alg	Peak SB 1 error	Peak SB 2
1	0.083322943973746	0.089636249424859
2	0.094886110144657	0.084331875384215
3	0.062102596182494	0.048499256055548

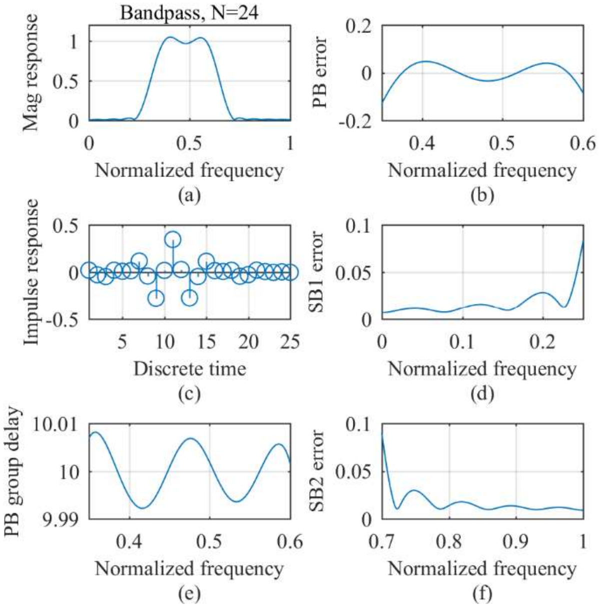


Fig. 4.11 G-FIR BP filter LS design for $N = 24$ and group delay=10 (a) Magnitude response, (b) Passband magnitude error, (c) Impulse response, (d) Stopband_1 magnitude error, (e) Passband group delay, (f) Stopband_2 magnitude error

Table 4.16 Gradient-based MOTLBO LS coefficients of BP

h(n)	Gradient-based MOTLBO
h(0)	0.0210235413525887
h(1)	-0.0271409833248832
h(2)	-0.0458457394805605
h(3)	0.0210197889992720
h(4)	0.00963309559035452
h(5)	0.0154934169535717
h(6)	0.117556803076730
h(7)	-0.0406706135119709
h(8)	-0.275039366719336
h(9)	0.0188703749159843
h(10)	0.345518029699649
h(11)	0.0282575824729041
h(12)	-0.271182646205469
h(13)	-0.0456211529293038
h(14)	0.114933829923214
h(15)	0.0165236492458097

h(16)	0.00778011174843973
h(17)	0.0203001430554704
h(18)	-0.0425075483492738
h(19)	-0.0257264511402188
h(20)	0.0219455058532533
h(21)	0.00694683966166235
h(22)	-0.00159483088879241
h(23)	0.00385599418795169
h(24)	-0.00109377549934151

4.6 Analysis

Section 4.5.1 describes the minimax design using the non-dominated MOTLBO (with crowding distance), and the results in Table 4.4 reveal that non-dominated MOTLBO algorithm (with crowding distance) can obtain much smaller minimax magnitude errors and minimax group delay errors. In addition, the required CPU time is approximately 1/2 of that of (non-dominated) MOPSO (with crowding distance) for lowpass and highpass designs. However, the non-dominated MOTLBO takes more time than the MOPSO for the designs of bandpass and bandstop filters. The peak passband and stopband magnitude errors, and the peak group delay errors are listed in Tables 4.5 and 4.6. All the peak errors of the non-dominated MOTLBO are less than those of the MOPSO.

Section 4.5.2 describes the least squares design using the gradient-based MOTLBO (with Manhattan distance) whose effectiveness is demonstrated in comparison to other methods. From Tables 4.7, 4.10 and 4.13, the non-dominated MOTLBO (with crowding distance) can achieve smaller LS magnitude error compared to the (non-dominated) MOPSO (with crowding distance), while the obtained LS group delay errors are usually larger than those of the MOPSO. In order to solve this problem, the gradient-based MOTLBO is adopted to design these filter examples. Based on the same parameter settings, the least squares magnitude errors of the LP, HP and BP filter designs obtained by the gradient-based MOTLBO are mostly slightly smaller than those obtained by the non-dominated MOTLBO and the MOPSO. On the contrast, the passband LS group delay errors obtained by the gradient-based MOTLBO are much less than those of the other two designs. From

Tables 4.8, 4.11 and 4.14, the least squares peak passband and stopband errors are similar for the gradient-based MOTLBO and the non-dominated MOTLBO, and the least squares peak group delay errors of the gradient-based MOTLBO are around 1/7 of the non-dominated MOTLBO. It has been shown that by replacing the original learning phase with the gradient descent optimization can speed up the search ability to reach smaller errors. However, the CPU time of the gradient-based MOTLBO takes longer time to optimize, which is between the non-dominated MOTLBO and the MOPSO. In conclusion, the error results of the gradient-based MOTLBO has verified that the gradient-based learning phase is successful. However, a tuning of the algorithm-specific parameter is required by the gradient-based learning phase and experiments reveal that proper control parameter settings can make the gradient-based MOTLBO more efficient.

4.7 Conclusions

In terms of general FIR filter MM designs, the non-dominated MOTLBO (with crowding distance) algorithm can achieve much smaller MM magnitude errors and MM group delay errors than the (non-dominated) MOPSO (with crowding distance). In terms of general FIR filter least squares designs, the gradient-based MOTLBO algorithm (with Manhattan distance) and the non-dominated MOTLBO (with crowding distance) can obtain similar LS peak magnitude errors and LS peak group delay errors. The non-dominated MOTLBO cannot reach the respective optimal levels of magnitude and group delay errors simultaneously, although its CPU time cost is the smallest among the three algorithms. The gradient-based MOTLBO algorithm can achieve the smallest LS magnitude error and LS group delay error and requires less CPU time than the MOPSO. All the designs are carried out on an Intel Core i7-4810MQ, 2.8 GHz (3.8 G Hz turbo) with 16GB RAM laptop computer.

CHAPTER 5

IIR Digital Filter Design Using Multiobjective TLBO Algorithm

A multiobjective optimization algorithm can be used to design IIR digital filters involving two or more objective functions. In [51]-[52], a non-dominated LS-MOEA was applied to IIR filter design, in which the magnitude error, phase error, and filter order formed three objectives. The results obtained were compared to NSGA-II. In [53], a MOCSO-DE combining cat swarm optimization (CSO) algorithm and differential evolution (DE) algorithm was used to design IIR filters.

There are many other design applications of multiobjective optimization algorithm. In [145], a MOTLBO algorithm was applied to solve a carbon footprint environmental problem by modifying the teacher phase of the TLBO to avoid being trapped into a local optimization. This algorithm was based on a non-dominated set and applied crowding distance to select the best solution from a non-dominated set. In [124], a MOTLBO algorithm was used to solve two objective functions consisting of the maximization of heat exchanger efficiency and the minimization of total cost of the exchanger. The designing results were compared to the results obtained using GA to show its efficiency. In [146], a non-dominated MOTLBO was applied to locate automatic voltage regulators in distribution systems. The results showed that the accuracy obtained was better than those using GA and PSO. In [147], a discrete MOTLBO with decomposition was proposed to solve the problem of community detection of complex networks.

Many multiobjective algorithms use non-dominated methods, which requires more computational time and takes up more computer memories. In this chapter, a Euclidean-distance-based approach developed in [29], [46] is combined with MOTLBO to form a Euclidean-distance-based MOTLBO [47] to design cascade-form IIR filters. Section 5.1 introduces the IIR filter design problem; Section 5.2 explains the Euclidean-distance-based MOTLBO; Section 5.3 shows some filter examples and results; and Section 5.4 gives conclusions.

5.1 IIR Filter Design Problem

5.1.1 IIR digital filter

In general, the cascade-form transfer function of an even *N*-order cascade IIR digital filter [80] can be expressed as

$$H(z) = b_0 \prod_{n=1}^{N/2} \frac{B_n(z)}{A_n(z)} = b_0 \prod_{n=1}^{N/2} \frac{(1+b_{1n}z^{-1}+b_{2n}z^{-2})}{(1+a_{1n}z^{-1}+a_{2n}z^{-2})} \tag{5.1}$$

In (5.1), b_{in} and a_{in} for $i = 1,2$ and $n = 1$ to $N/2$ are real-valued coefficients, and b_0 is a scaling constant. The corresponding coefficient vector **c** consists of *(2N+1)* distinct coefficients to be optimized as

$$\mathbf{c} = \left[b_{11}\, b_{21}\, a_{11}\, a_{21} \cdots b_{1,N/2}\, b_{2,N/2}\, a_{1,N/2}\, a_{2,N/2}\, b_0\right]^T \tag{5.2}$$

Substituting $z = e^{j\omega}$ into (5.1), the frequency response of the *N*th-order cascade IIR digital filter can be expressed as

$$H(w) = H(z)|_{z=e^{j\omega T}} = |H(w)|e^{j\theta(w)} \tag{5.3}$$

5.1.2 Stability constraints

For a second-order IIR filter, the stability triangle offers a set of necessary and sufficient conditions to ensure stability [80] given by

$$-2 < a_{1n} < 2 \tag{5.4}$$

$$-1 < a_{2n} < 1 \tag{5.5}$$

$$a_{2n} > |a_{1n}| - 1 \tag{5.6}$$

Stability constraints are to be applied for coefficient update during optimization.

5.1.3 Objective function

The relationship between the maximum passband ripple A_p (in dB) and its corresponding peak error of magnitude response in passband δ_p is

$$\delta_p = \frac{10^{Ap/20}-1}{10^{Ap/20}+1} \tag{5.7}$$

Similarly, the relationship between the minimum stopband attenuation A_s (in dB) and its corresponding peak error of magnitude response in stopband δ_s is

$$\delta_s = 10^{-A_s/20} \tag{5.8}$$

The peak-error function of magnitude response in passband $e_{mag,p}(\mathbf{c})$ can be expressed by

$$\begin{gathered} e_{mag,p}(\mathbf{c}) = \max_{\mathbf{c}} \big| |H(\mathbf{c},\omega_i)| - H_d(\omega_i) \big| \\ \text{for } \omega_i \in \Omega_P \end{gathered} \tag{5.9}$$

where $H_d(\omega_i) = 1$ in passband; and Ω_P denotes a union of frequencies of interest in passband.

The objective function of magnitude response in passband subject to a given constraint can be expressed as

$$f_1(\boldsymbol{c}) = \varphi\big(e_{mag,p}(\mathbf{c}) - \delta_{p0}\big) \tag{5.10}$$

where δ_{p0} is the specified peak error constraint of magnitude response in passband; and $\varphi(x)$ is a sectional-continuous function that

$$\varphi(x) = \begin{cases} 0 & x \le 0 \\ x & x > 0 \end{cases} \tag{5.11}$$

Similarly, the peak-error function of magnitude response in stopband $e_{mag,s}(\mathbf{c})$ is described by

$$\begin{gathered} e_{mag,s}(\mathbf{c}) = \max_{\mathbf{c}} \big| |H(\mathbf{c},\omega_i)| - H_d(\omega_i) \big| \\ \text{for } \omega_i \in \Omega_S \end{gathered} \tag{5.12}$$

where $H_d(\omega_i) = 0$ in stopband; and Ω_S denotes a union of frequencies of interest in stopband.

The objective function of magnitude response in stopband subject to a given constraint is derived as

$$f_2(\boldsymbol{c}) = \varphi\left(e_{mag,s}(\mathbf{c}) - \delta_{s0}\right) \tag{5.13}$$

where δ_{s0} is the specified peak error constraint of magnitude response in stopband.

To prevent the magnitude response in the transition band from exceeding its required maximum value, a magnitude limitation for transition band can be applied such that

$$H(\mathbf{c}, \omega_i) \leq 1 \qquad \text{for } \omega_i \in \Omega_T \tag{5.14}$$

where Ω_T is the union of frequencies in transition band.

In other words, (5.14) can be transformed into an inequality constraint that

$$g(\boldsymbol{c}) = \max_{c}(H(\mathbf{c}, \omega_i) - 1) \leq 0 \qquad \text{for } \omega_i \in \Omega_T \tag{5.15}$$

The objective function of magnitude response in transition band subject to a given constraint can be expressed as

$$f_3(\boldsymbol{c}) = \varphi\left(g(\boldsymbol{c})\right) = \varphi\left(\max_{c}(H(\mathbf{c}, \omega_i) - 1)\right) \tag{5.16}$$

The peak-error function of the group delay response in passband $e_{gd}(\mathbf{c})$ can be expressed by

$$\begin{gathered} e_{gd}(\mathbf{c}) = \max_{\mathbf{c}} |\tau(\mathbf{c}, \omega_i) - \tau_d(\omega_i)| \\ \text{for } \omega_i \in \Omega_P \end{gathered} \tag{5.17}$$

where $\tau_d(\omega)$ is the ideal group delay.

The objective function of group delay response in passband subject to a given constraint is described as

$$f_4(\boldsymbol{c}) = \varphi\left(e_{gd}(\mathbf{c}) - \delta_{gd}\right) \tag{5.18}$$

where δ_{gd} is the required peak error of group delay in passband.

The maximum group delay deviation Q is used for evaluating the obtained group delay response in passband as

$$Q = \frac{100(\tau_{max} - \tau_{min})}{(\tau_{max} + \tau_{min})} \tag{5.19}$$

where τ_{max} and τ_{min} are respectively the maximum and minimum values of group delay in passband.

The design goal is to minimize $f_1(\boldsymbol{c})$, $f_2(\boldsymbol{c})$, $f_3(\boldsymbol{c})$and $f_4(\boldsymbol{c})$ simultaneously of an IIR digital filter. The coefficient vector **c** contains all the filter coefficients that need to be optimized.

5.2 Euclidean-Distance-Based MOTLBO

Details of the original TLBO algorithm can be found in Section 2.3. The Euclidean-Distance-Based MOTLBO follows all the steps in the original algorithm.

Besides, for this four-objective minimization problem, all the objective functions can be represented as an Euclidean spatial point corresponding to a solution **c** in an Euclidean Space $\boldsymbol{S}^4$ as

$$\mathbf{f}(\mathbf{c}) = [f_1(\mathbf{c}), f_2(\mathbf{c}), f_3(\mathbf{c}), f_4(\mathbf{c})\ ] \tag{5.20}$$

The current ideal spatial point $\boldsymbol{z}(t)$ obtained at the current iteration *t* can be represented by

$$\mathbf{z}(\mathrm{t}) = \left[z_1^{min}(\mathrm{t}), z_2^{min}(\mathrm{t}), z_3^{min}(\mathrm{t}), z_4^{min}(\mathrm{t})\right] \tag{5.21}$$

where

$$z_m^{min}(\mathrm{t}) = \min\left[\min f_m(\boldsymbol{c}), z_m^{min}(t-1)\right] \tag{5.22}$$

where $z_m^{min}(t)$ (for $m = 1$ to 4) represents the minimum value of $f_m(\mathbf{c})$ that has ever reached from the first iteration to the current iteration.

The Euclidean distance between the overall ideal point $\mathbf{z}$ and the Euclidean spatial point $\mathbf{f}(\mathbf{c})$ in the Euclidean Space $\boldsymbol{S}^4$ defined in (5.20) is used to compare the qualities of obtained solutions as

$$d\big(\boldsymbol{z}, \boldsymbol{f}(\mathbf{c})\big) = \sqrt{\textstyle\sum_{m=1}^{4}[f_m(\boldsymbol{c}) - z_m]^2} \tag{5.23}$$

Within a population, the one with the smallest Euclidean distance is selected as the teacher. The optimization procedure is terminated when the value of Euclidean distance of the teacher reaches zero.

5.3 Filter Examples and Results

In this section, a cascade-form IIR lowpass filter of order *N*=10 and a highpass filter of order *N*=14 are designed using the Euclidean-distance-based MOTLBO algorithm. Filter coefficients are initialized randomly subjected to the stability constraints. Maximum passband and stopband magnitude errors and maximum passband group delay error are simultaneously minimized. The lowpass and highpass filter specifications are listed in Table 5.1. 201 frequency points are used for optimization and 1001 frequency points are used for calculating peak magnitude errors and peak group delay error for each of the lowpass and highpass filters. The power p is chosen as 128 in the least-p objective functions. A magnitude constraint of 1 is applied to the transition-band for both filter designs. The design results of the lowpass and highpass filters are respectively compared to the 6A-2 and 7A-2 designs in [54]. Tables 5.2 and 5.3 summarize the lowpass and highpass filter design results taken from the best solution obtained by the Euclidean-distance-based MOTLBO and those of the design methods in [54]. The peak magnitude and group delay responses and ripples/attenuations (in dB) of the designed lowpass and highpass filters are shown in Fig. 5.1 to Fig. 5.3 respectively. Fig. 5.2 and Fig. 5.4 show the convergence curves. The time taken are 3695 and 5628 CPU seconds for designing respectively lowpass (order 10) and highpass (order 14) IIR filters on an Intel Core i7-4810MQ, 2.8 GHz (3.8 GHz turbo) with 16GB RAM laptop computer.

Table 5.1 IIR Digital Filter Specifications

Parameters	Lowpass	Highpass
Order	10	14
Passband edge, rad/s	0.4π	0.525π
Stopband edge, rad/s	$0.56\ \pi$	$0.475\ \pi$
Iterations	1,000,000	1,000,000

Table 5.2 Lowpass Filter Design Results (PB: passband; SB: stopband; Gd: group delay)

	MOTLBO	Design 6A-2 [54]
Peak PB error	0.011867126987405	0.011870695865662
Peak SB error	0.003162049628274	0.003162390333019
Max PB ripple, dB	0.206162788909873	0.206224795410127
Min SB attenuation, dB	50.00062636222914	49.99969052480010
Peak Gd error	0.015566495490468	0.015596828111102
Q	0.157877569359126	0.159006931670001

Table 5.3 Highpass Filter Design Results (PB: passband; SB: stopband; Gd: group delay)

	MOTLBO	Design 7A-2 [54]
Peak PB error	0.028309884202968	0.028632207868323
Peak SB error	0.028183675882967	0.028212572955863
Max PB ripple, dB	0.491924505361867	0.497528383486763
Min SB attenuation, dB	31.00004728515969	30.99114609711360
Peak Gd error	0.217511682074813	0.217684985592550
Q	1.708661979958310	1.713816165208324

Table 5.4 Bandpass Filter Design Results (PB: passband; SB: stopband; Gd: group delay)

	MOTLBO	Design 3A-2 [54]
Peak PB error	0.058434315423512	0.059398878295131
Peak SB error	0.008543769439622	0.008544696148768
Max PB ripple, dB	1.016265788108064	1.033080328785178
Min SB attenuation, dB	41.36700959875172	41.36606752547557
Peak Gd error	0.000322820870707	0.000322978540844
Q	0.001260688379388	0.001264277871753

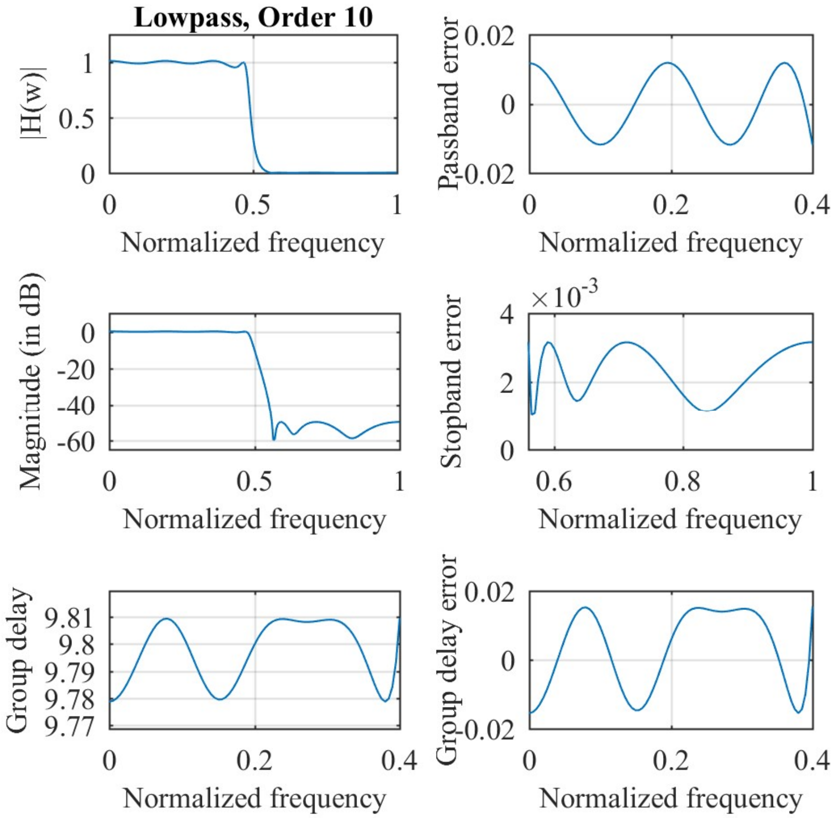


Fig. 5.1 Magnitude and group delay responses of designed lowpass filter

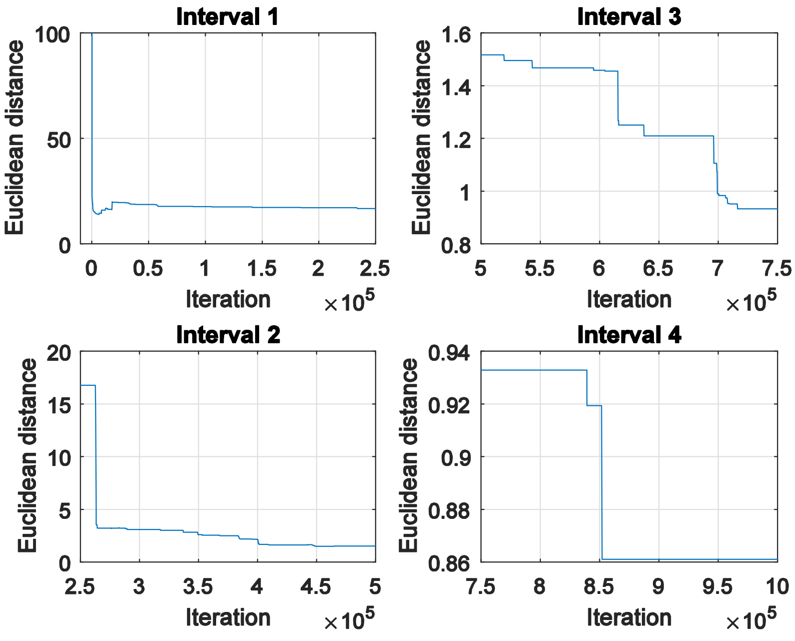


Fig. 5.2 Convergence curve of designed lowpass filter

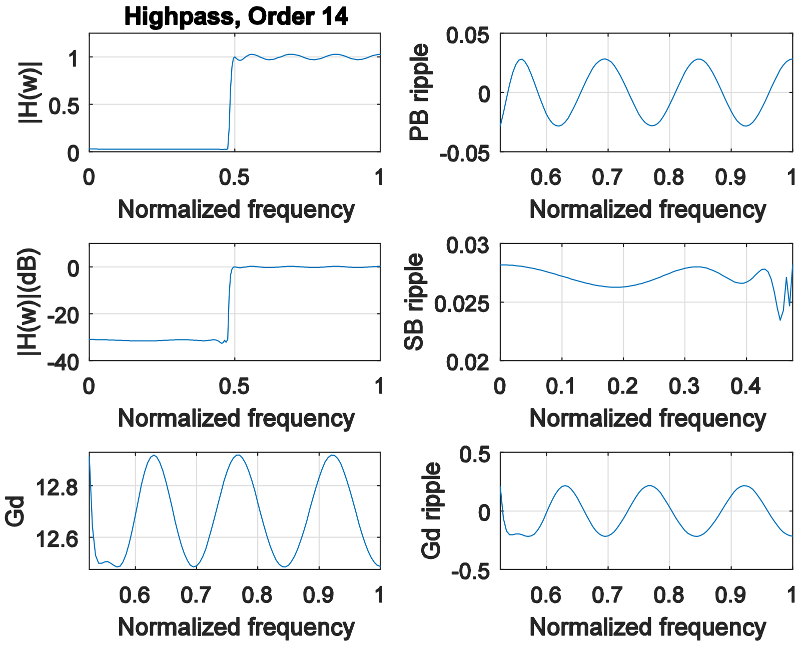


Fig. 5.3 Magnitude and group delay responses of designed highpass filter

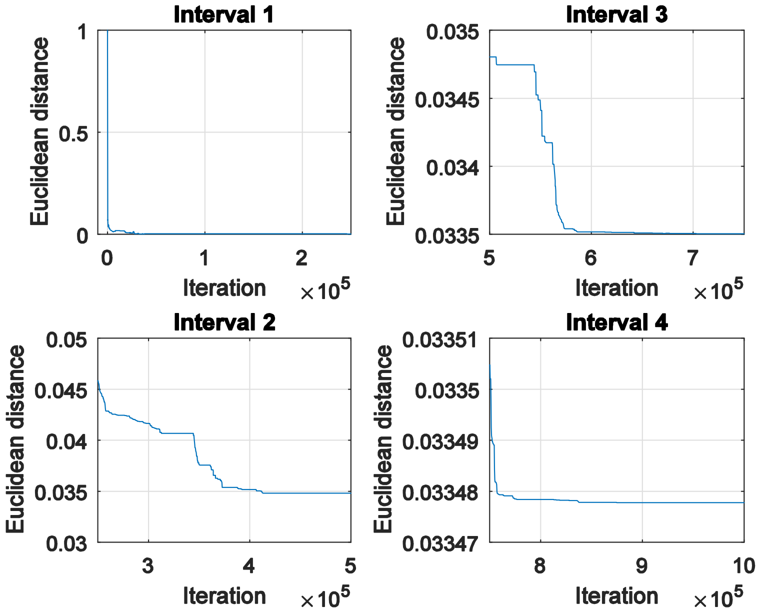


Fig. 5.4 Convergence curve of designed highpass filter

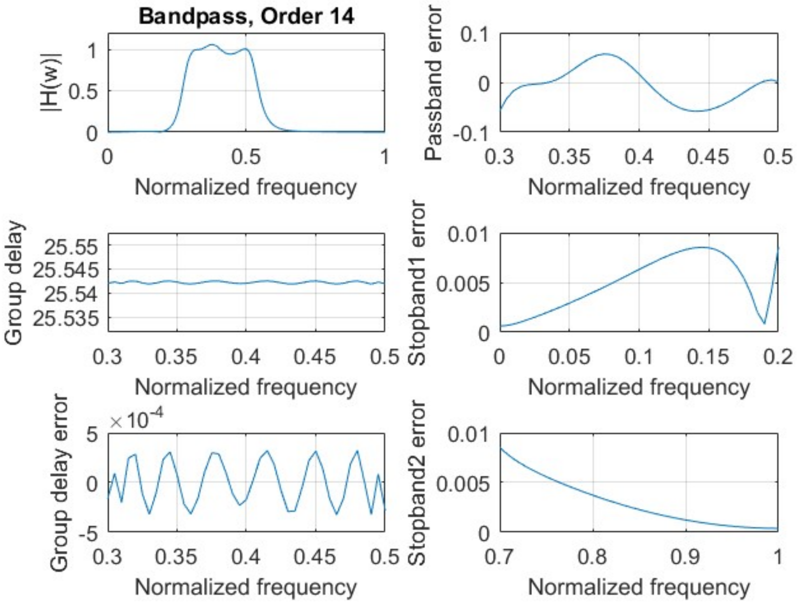


Fig. 5.5 Magnitude and group delay responses of designed bandpass filter

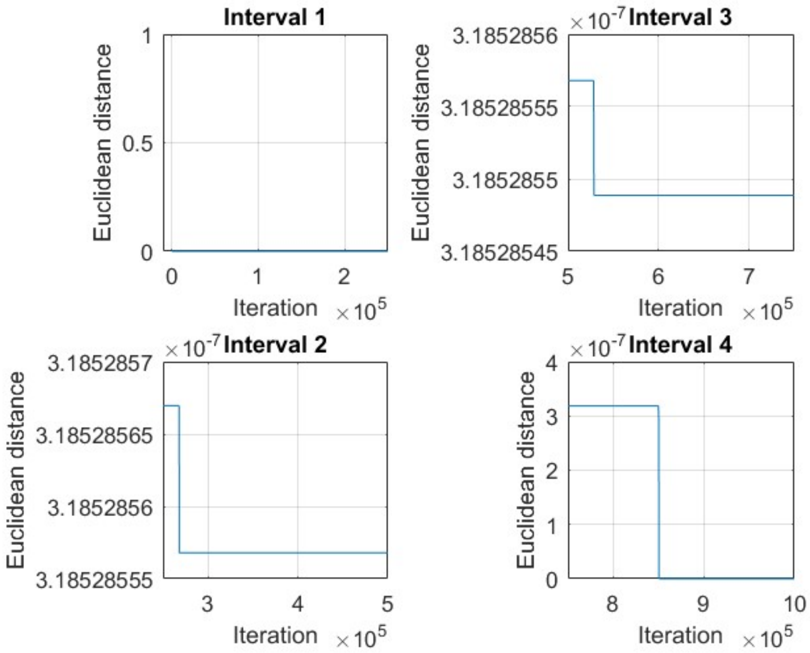


Fig. 5.6 Convergence curve of designed bandpass filter

5.4 Analysis

As compared to the-state-of-the-art design method [54], the results in Tables 5.2 to 5.4 indicate that the Euclidean-distance-based MOTLBO can obtain slightly smaller passband and stopband peak magnitude errors and passband peak group delay error in the lowpass, highpass and bandpass filter examples. As IIR filters require stability control, the design optimization takes longer time than FIR filters and the results are also difficult to obtain. Moreover, the transition band requires a constraint to ensure the quality of passband and stopband magnitude response errors. Three examples among the LP, HP and BP filter designs in [54] are selected for comparisons, the results reveal that the Euclidean-distance-based MOTLBO can optimize IIR digital filters to achieve slightly smaller magnitude and group delay errors. Figs. 5.2, 5.4 and 5.6 show that all the three filter designs can converge within the maximum iteration.

5.5 Conclusions

In this chapter, peak-error design of cascade IIR filters using Euclidean-distance-based MOTLBO has been presented. The design results reveal that the

Euclidean-distance-based MOTLBO is suitable for designing IIR filters. The state-of-the-art method [54] has designed ten IIR filters, and this chapter picks up three different types of IIR filters for comparison. The three comparisons demonstrate TLBO can achieve slightly smaller errors. All the selected LP, HP and BP IIR filters have been optimized by Euclidean-distance-based MOTLBO with improved results. All the designs were carried out on an Intel Core i7-4810MQ, 2.8 GHz (3.8 G Hz turbo) with 16GB RAM laptop computer.

CHAPTER 6

2-Dimensional Linear Phase FIR Digital Filter Design using TLBO Algorithm

With the advancement of technologies, the applications of two-dimensional (2-D) digital filters are expanding, and the design of two-dimensional digital filters remains an important research problem. Although the design problem of two-dimensional FIR filters has made progress in the past few decades, there are problems remain to be solved. Under the precondition that the maximum approximation errors in the passband and the stopband cannot exceed a given value, minimizing the filter size has been a challenging task in 2-D digital filter design. Two-dimensional linear phase finite impulse response (2-D FIR) filters [80], [55]-[61] are stable and have applications in digital image processing. The magnitude responses of 2-D linear phase FIR filters can be classified as rectangular, circular and diamond-shape. In [56], the coefficient matrix of the filter is utilized for constrained least-squares and minimax designs of quadrantally symmetric 2-D linear-phase FIR filters. In [60], an accelerated artificial bee colony algorithm which belongs to a class of metaheuristic algorithms is applied to 2-D linear-phase FIR filter design. Its filter examples include 2-D linear phase FIR filters of circular, diamond, and elliptic shapes in magnitude responses. In [59], a 2-D linear phase FIR lowpass filter with maximum flat passband and stopband is given. The proposed transfer function is expressed in closed form and various passband shapes can be created by changing the filter order or flatness degree. Two common methods for digital filter design formulation are weighted-least-squares (WLS) method [64] and minimax (MM) method [65].

In this chapter, CLS method, WLS method and MM method are applied to design 2-D linear phase FIR lowpass filters using the elitist TLBO algorithm. Section 6.1 introduces the basics and mathematic equations for 2-D linear phase FIR filter design. The optimization frequency points and two design examples are listed and described in Section 6.2. The result analysis is explained in Section 6.3.

6.1 2-D Linear Phase FIR Filter Design Problem

6.1.1 Digital filter transfer function

The causal transfer function of the $(N_1 - 1, N_2 - 1)$th-order 2-D FIR digital filter is given by

$$H(z_1, z_2) = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} c(\, n_1, n_2) z_1^{-n_1} z_2^{-n_2} \tag{6.1}$$

From (6.1), the zero-phase transfer function of a $(N_1 - 1, N_2 - 1)th$-order 2-D FIR digital filter can be expressed as

$$H(z_1, z_2) = \sum_{n_1=-N_{c1}}^{N_{c1}} \sum_{n_2=-N_{c2}}^{N_{c2}} c(\, n_1, n_2) z_1^{-n_1} z_2^{-n_2} \tag{6.2}$$

$$N_{c1} = (N_1 - 1)/2\,;\; N_{c2} = (N_2 - 1)/2 \tag{6.3}$$

The zero-phase transfer function of a 2-D linear phase octagonal symmetric FIR digital filter [80] is given by

$$H(\omega_1, \omega_2) = e^{-jN_c w_1} e^{-jN_c w_2} A(\omega_1, \omega_2) \tag{6.4}$$

where

$$A(\omega_1, \omega_2) = \sum_{k_2=0}^{M_c-1} \sum_{k_1=k_2+1}^{M_c} c(k_1, k_2) \{\cos(k_1\omega_1)\cos(k_2\omega_2) + \cos(k_2\omega_1)\cos(k_1\omega_2)\} + \sum_{k=0}^{M_c} c(k, k)\cos(k\omega_1)\cos(k\omega_2) \tag{6.5}$$

For octagonal symmetry, the coefficients of the zero-phase and the causal transfer functions [80] are related by

$$c(0,0) = c(N_{c1}, N_{c2}) \tag{6.6}$$

$$c(0, k_2) = 2c(N_{c1}, N_{c2} - k_2) \quad \text{for } 1 \le k_2 \le N_{c2} \tag{6.7}$$

$$c(k_1, 0) = 2c(N_{c1} - k_1, N_{c2}) \quad \text{for } 1 \le k_1 \le N_{c1} \tag{6.8}$$

$$c(k_1, k_2) = 4c(N_{c1} - k_1, N_{c2} - k_2) \quad \text{for } 1 \le k_1 \le N_{c1}, 1 \le k_2 \le N_{c2} \tag{6.9}$$

$$c(k_1, k_2) = c(k_2, k_1) \tag{6.10}$$

$$N_{c1} = N_{c2} \tag{6.11}$$

6.1.2 Objective function

For circularly symmetric 2-D filter design, all the discrete frequency points of interest on the (ω_1, ω_2)-plane can be expressed in terms of its radius $d = \sqrt{(\omega_1^2 + \omega_2^2)}$ and angle θ as

$$p(\omega_1, \omega_2) = \sqrt{(\omega_1^2 + \omega_2^2)}\angle\theta \qquad \text{for } 0 \leq \theta \leq 45° \tag{6.12}$$

For the case of a 2-dimensional linear phase circularly symmetric lowpass FIR digital filter, the desired magnitude response is given by

$$|A_d(\omega_1, \omega_2)| = \begin{cases} 1 & \text{for } 0 \leq \sqrt{\omega_1^2 + \omega_2^2} \leq \omega_p \\ 0 & \text{for } \left\{\omega_s \leq \sqrt{\omega_1^2 + \omega_2^2} \leq \sqrt{2}\pi\right\} \cap \{|\omega_1| \leq \pi\} \cap \{|\omega_2| \leq \pi\} \end{cases} \tag{6.13}$$

The weighted least-squares objective function of magnitude response error is defined as

$$e(\boldsymbol{c}) = \sum_{(\omega_1,\omega_2)\in\Omega_I} W_{12}\big||A(\boldsymbol{c}, \omega_1, \omega_2)| - |A_d(\omega_1, \omega_2)|\big|^2 \tag{6.14}$$

where Ω_I denotes a union of frequency points of interest in passband and stopband; and W_{12} denotes the frequency weight at (ω_1, ω_2).

The minimax objective function of magnitude response error is defined by

$$e(\boldsymbol{c}) = \max_{(\omega_1,\omega_2)\in\Omega_I}\big||A(\boldsymbol{c}, \omega_{1i}, \omega_{2i})| - |A_d(\omega_{1i}, \omega_{2i})|\big| \tag{6.15}$$

6.2 Design examples

In this section, two 2-D octagonal symmetric FIR lowpass filters are designed using the TLBO algorithm with an elitist replacement [207]. The elitist replacement is applied by replacing the worst learner in the population by the teacher solution and with added mutations on one or more randomly selected dimensions for each duplicated solution before proceeding to the next generation. The passband and stopband cutoff frequencies of each of the 2-D FIR filters are listed in Table 6.1 and Table 6.4 respectively. The 2-D FIR filter examples in size of 27x27, 35x35, 39x39 and 43x43 are designed using the minimax formulation. The initial filter coefficients of each of the filters can be obtained by Parks-McClellan algorithm (firpm.m in Matlab) after 2-D transformation [57] (ftrans2.m in Matlab). The discrete frequency points of interest on the (ω_1, ω_2)-plane are all located within the 0 to 45-degree sector divided into 30 equal sub-sectors, each angle of which is 1.5 degree. Along the full length of each of these 31 dividing radius from 0 to $\sqrt{2}\pi$, there are 41 uniformly spaced frequency points in the range of $[0, \pi]$ (or 0<= d <= π) and 17 uniformly spaced frequency points in the range of $(\pi, \sqrt{2}\pi)$ (or π < d <= $\sqrt{2}\pi$) for all the filter designs. For lowpass filter optimization, one additional passband arc-grid at $d = \sqrt{\omega_1^2 + \omega_2^2} = \omega_p - \frac{\pi}{2(41-1)}$ and one additional stopband arc-grid at $d = \sqrt{\omega_1^2 + \omega_2^2} = \omega_s + \frac{\pi}{2(41-1)}$ are inserted. In so doing, there are a total of 41+2 frequency points on each dividing radius in the range of $[0, \pi]$ (or 0<= d <= π). Evaluation is performed within the 0 to 45-degree sector divided into 60 equal sub-sectors, each angle of which is 0.75 degree. Along the full length of each of these 31 dividing radius from 0 to $\sqrt{2}\pi$, there are 81(=41+40) uniformly spaced frequency points in the range of $[0, \pi]$ (or 0<= ω <= π) and 34 (=17*2) uniformly spaced frequency points in the range of $(\pi, \sqrt{2}\pi)$ (or π < ω <= $\sqrt{2}\pi$) for all the filter designs. However, those frequency points outside the entire frequency band $0 \leq w_1, w_2 \leq \pi$ and inside the transition band are to be excluded. Meanwhile, the frequency grid of optimization and evaluation in [80] is a rectangular grid, which has exactly $5N$×$5N$ frequency points in first quadrant.

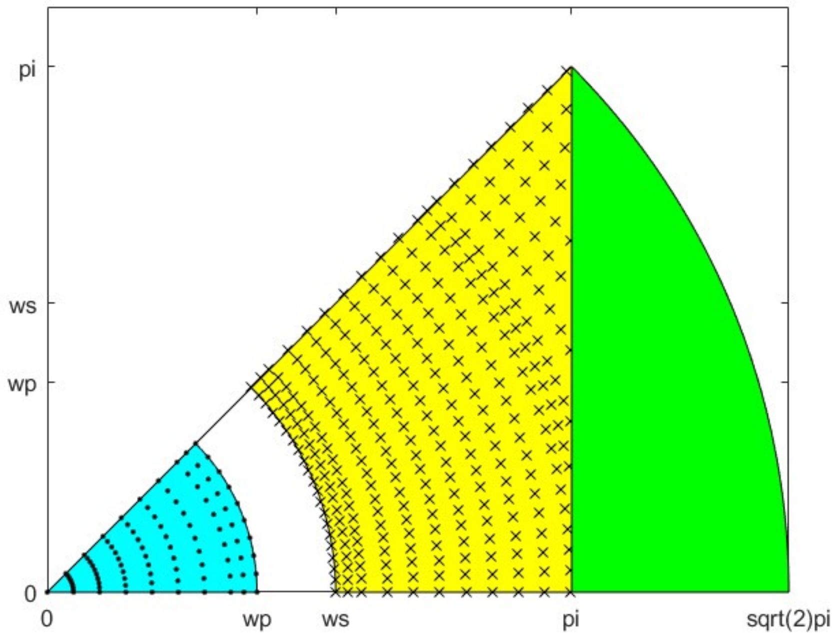


Fig. 6.1 Discrete frequency grid points for designing 2-D circularly symmetric digital filters (Blue stands for passband; Yellow stands for stopband)

Two filter examples and their maximum magnitude errors (MME) obtained by the elitist TLBO are listed in Table 6.2 and Table 6.5 respectively. The maximum magnitude errors in passband and stopband of the filter examples are listed in Table 6.3 for example 1 and Table 6.6 for example 2. Tables 6.4 and 6.7 list the numbers of iterations and CPU time (in seconds) of the two designs using an Intel Core i7-4810MQ, 2.8 GHz (3.8 GHz turbo) with 16GB RAM laptop computer. The magnitude responses in dB and the error convergence curves of the two designed filters are plotted in Fig. 6.2 to 6.8 and Fig. 6.9 to 6.13 respectively.

$$MME = \max[e(\boldsymbol{c})] \tag{6.16}$$

6.2.1 Example 1

In this section, 2-D linear phase FIR lowpass filters are designed using the elitist TLBO algorithm. The cutoff frequency specifications of the 2-D linear phase FIR filters are listed in Table 6.1. The 27x27th-order FIR filter is designed using the

WLS problem formulation and the 35x35th-order FIR filter is designed using the minimax formulation. All the WLS frequency weights are set to 1.

Table 6.1 Cutoff Frequencies of 2-D FIR Lowpass Filters

Filter order	Passband	Stopband
27×27	0.40π	0.55π
35×35	0.40π	0.55π

Table 6.2 2-D FIR Lowpass Filter Design Results

Specification	Algorithm	MME
N_1=27, N_2=27 $\tau_1 = \tau_2 = 13$	TLBO	0.011742 (WLS)
	Example 2 [61]	0.013784 (WLS)
N_1=N_2=35 $\tau_1 = \tau_2 = 17$	TLBO	2.62e-06 (CLS)
	Example 3 [56]	3.91e-04 (CLS)

Table 6.3 Computational Requirements

Filter order	Iteration	CPU time (sec)
N_1=N_2=27	10,000	8.81169e+02
N_1=N_2=35	400,000	3.8279e+04

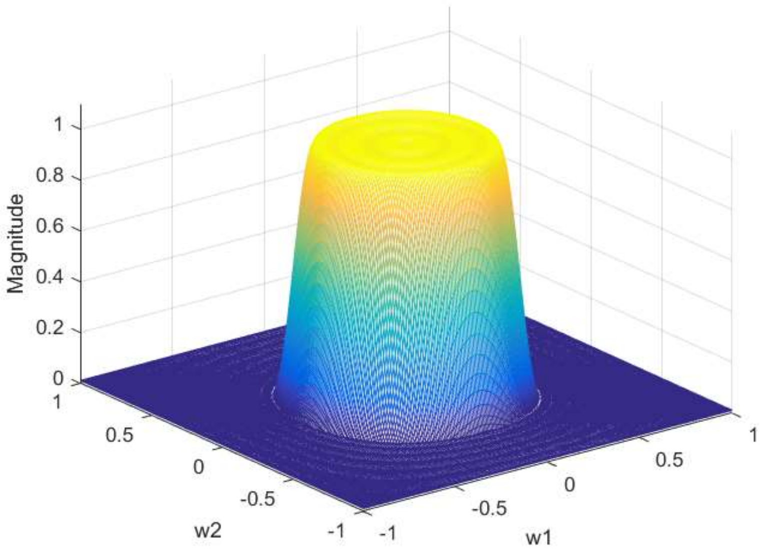


Fig. 6.2 Magnitude response of 27×27 2-D linear phase circularly symmetric FIR lowpass filter designed by TLBO

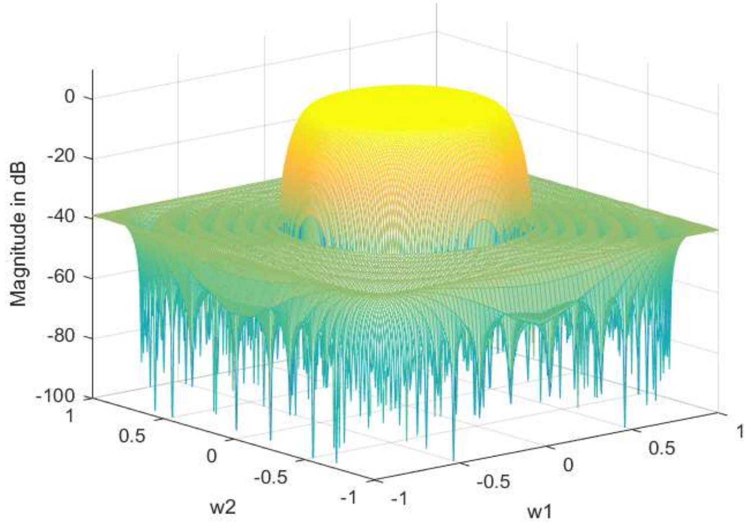


Fig. 6.3 Magnitude response in dB of 27×27 2-D linear phase circularly symmetric FIR lowpass filter designed by TLBO

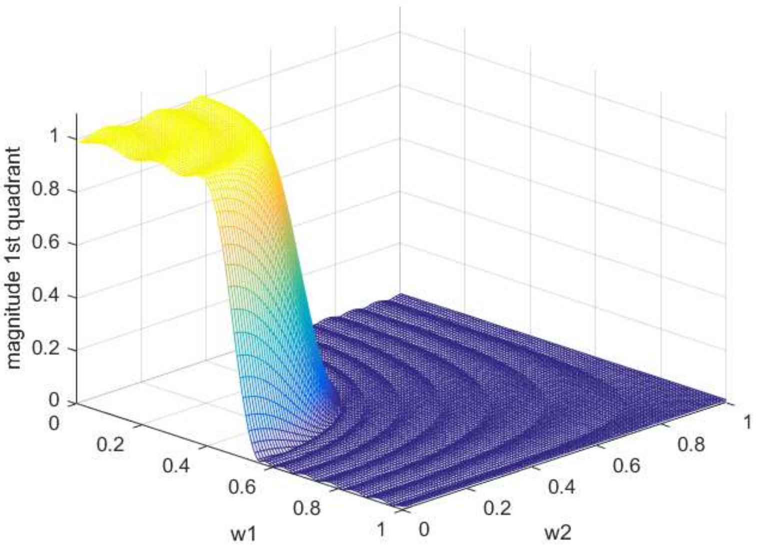


Fig. 6.4 The 1st quadrant magnitude response in dB of 27×27 2-D linear phase circularly symmetric FIR lowpass filter designed by TLBO

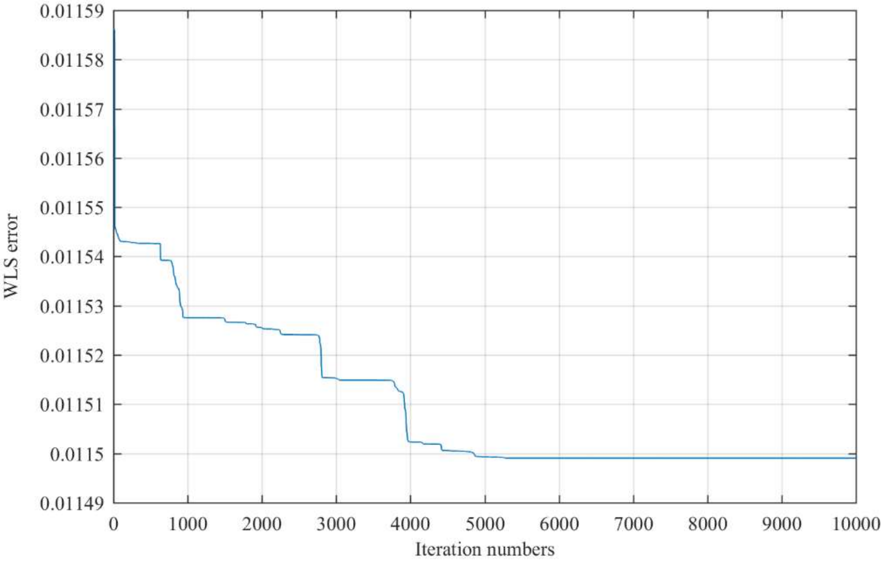


Fig.6.5 WLS error convergence curve of 27×27 2-D linear phase circularly symmetric FIR lowpass filter designed by TLBO

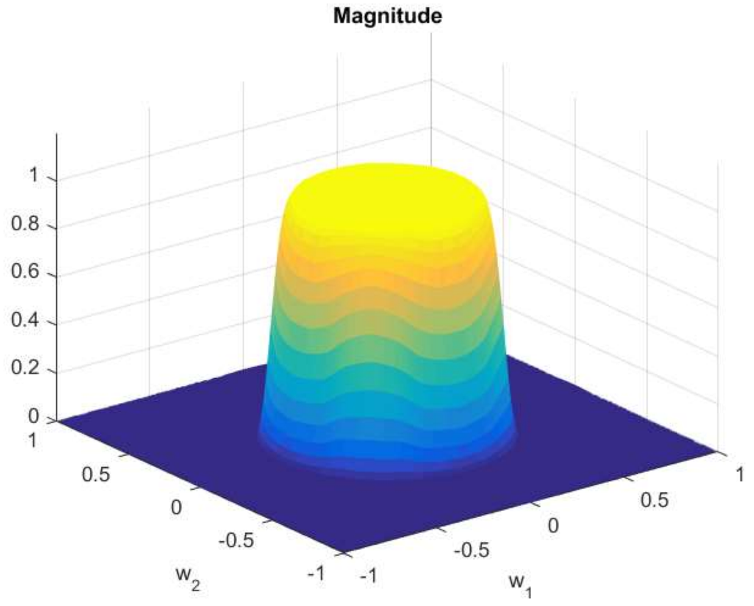


Fig. 6.6 Magnitude response of 35×35 2-D circularly symmetric FIR lowpass filter designed by TLBO

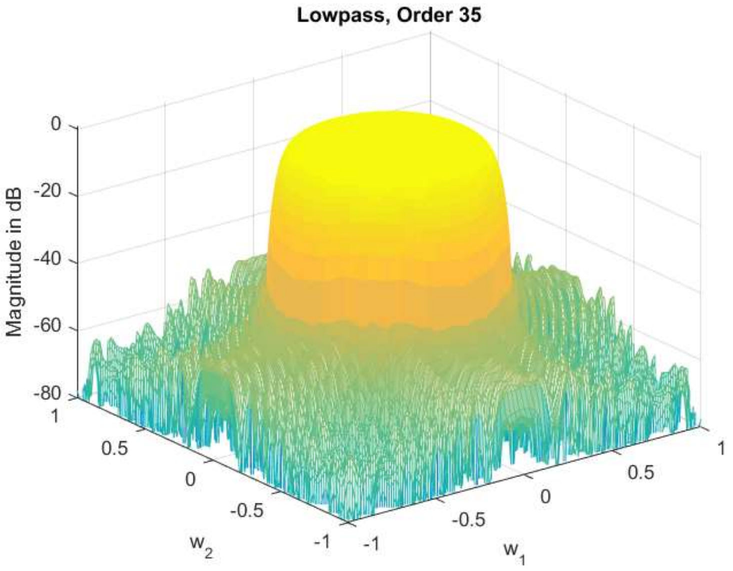


Fig. 6.7 Magnitude response in dB of 35×35 2-D circularly symmetric FIR lowpass filter designed by TLBO

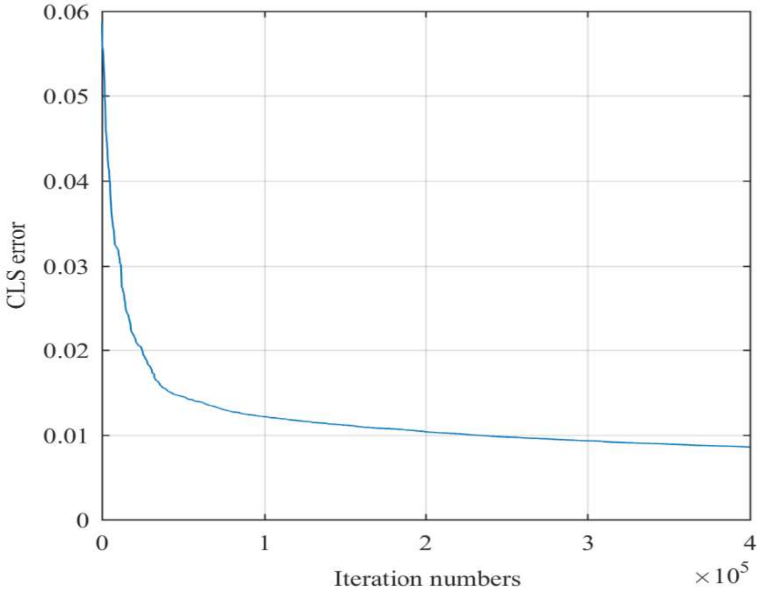


Fig. 6.8 WLS error convergence of 35×35 2-D circularly symmetric FIR lowpass filter designed by TLBO

6.2.2 Example 2

In this section, 2-D linear phase FIR lowpass filters are designed using the elitist TLBO algorithm. The cutoff frequency specifications of the 2-D linear phase FIR filters are listed in Table 6.4. The 39x39th-order and 43x43th-order FIR filter are designed using the MM problem formulation.

Table 6.4 Cutoff Frequencies of 2-D FIR Lowpass Filters

Example	Specification	Passband	Stopband
Ex. 3 [61]	N_1=N_2=39 $\tau_1 = \tau_2 = 19$	0.40π	0.50π
Ex. 3 [61]	N_1=N_2=43 $\tau_1 = \tau_2 = 21$	0.40π	0.50π

Table 6.5 2-D FIR Lowpass Filter Design Results

Filter size	Algorithm	Design	MME
39×39	TLBO	MM	0.006290
	Example 3 [61]	MM	0.010284
43×43	TLBO	MM	0.004571
	Example 3 [61]	MM	0.007418

Table 6.6 2-D FIR Lowpass Filter Design Results Using TLBO

Filter size	Design	MME (Passband)	MME (Stopband)
39×39	MM	0.006232	0.006290
43×43	MM	0.004554	0.004571

Table 6.7 TLBO Computational Requirements

Filter size	Iteration	CPU time (sec)
39×39	66,988	3.9856e+03
43×43	61,068	8.9636e+03

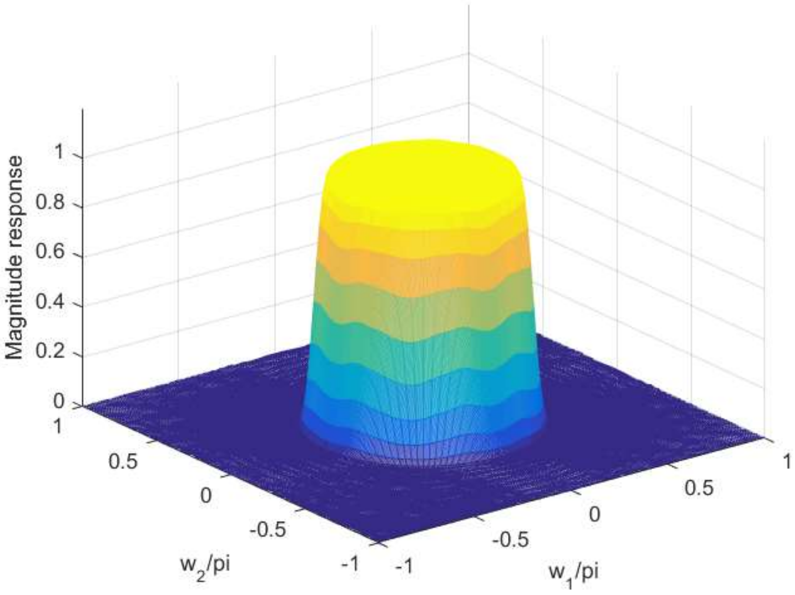


Fig. 6.9 Magnitude response of 39×39 2-D circularly symmetric FIR lowpass filter designed by TLBO

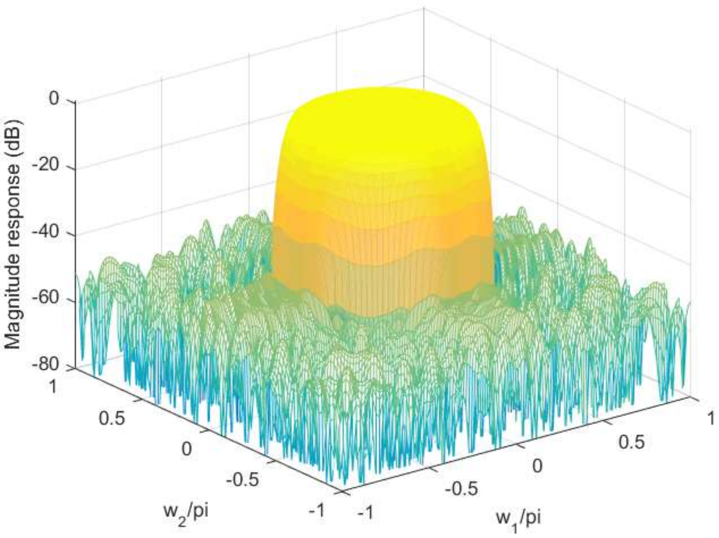


Fig. 6.10 Magnitude response in dB of 39×39 2-D circularly symmetric FIR lowpass filter designed by TLBO

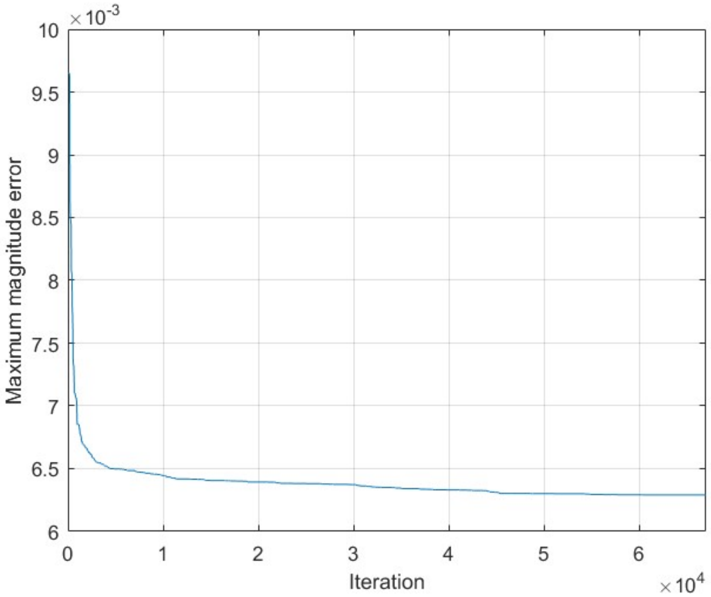


Fig. 6.11 Error convergence curve of 39×39 2-D circularly symmetric FIR lowpass filter designed by TLBO

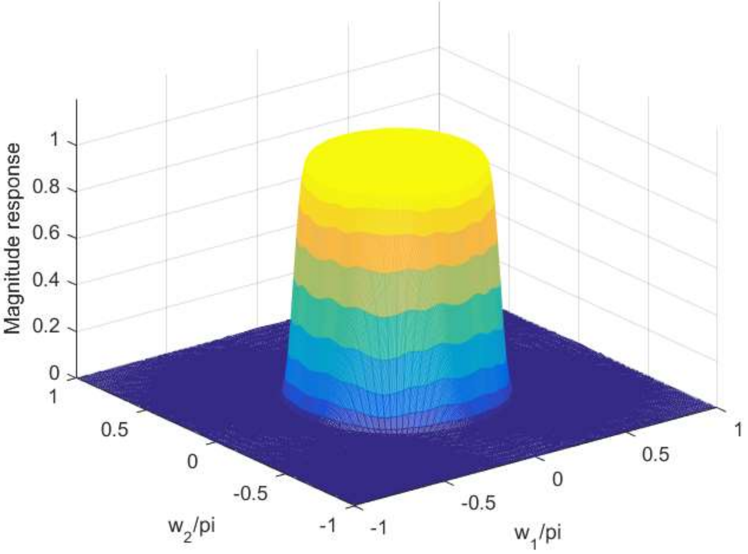


Fig. 6.12 Magnitude response of 43×43 2-D circularly symmetric FIR lowpass filter designed by TLBO

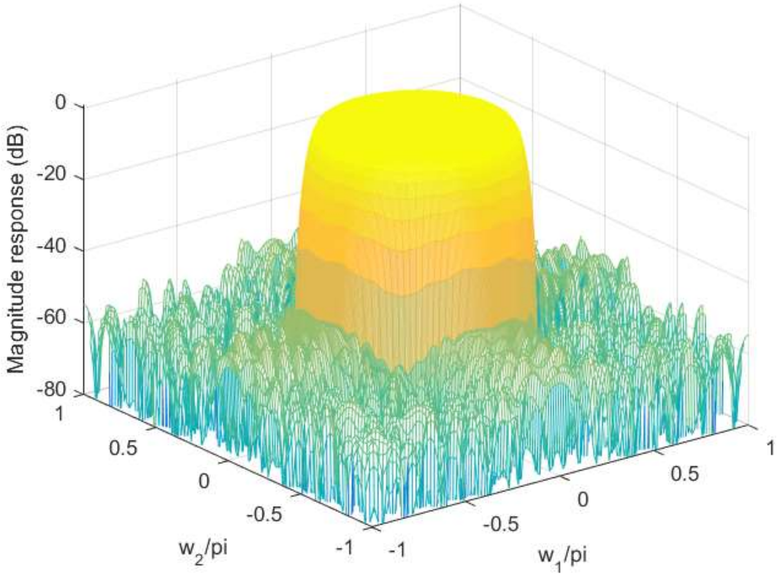


Fig. 6.13 Magnitude response in dB of 43×43 2-D circularly symmetric FIR lowpass filter designed by TLBO

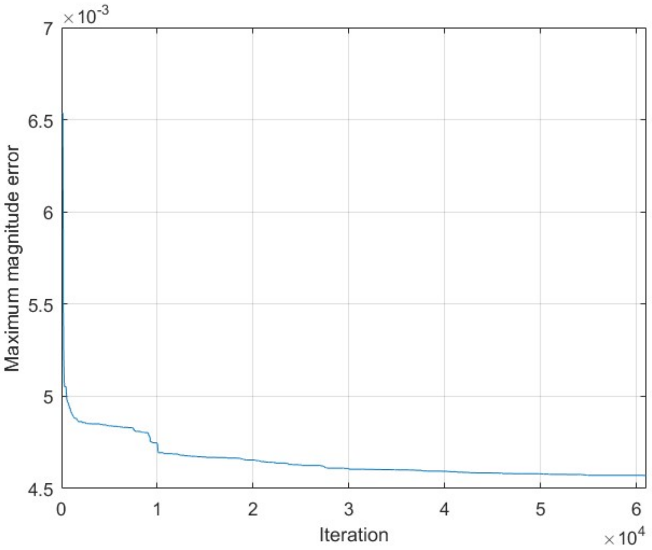


Fig. 6.14 Error convergence of 43×43 2-D circularly symmetric FIR lowpass filter designed by TLBO

6.3 Analysis

The two filter examples in this chapter use the elitist TLBO algorithm to design 2-D circularly symmetric FIR lowpass digital filters. The first example uses the WLS and CLS approaches, respectively, and are compared to two state-of-the-art methods. The MME results in Table 6.2 reveal that the elitist TLBO algorithm can achieve smaller maximum magnitude errors (MMEs) than those obtained in [56] and [61]. For filter order 27×27, the MME is close to the result in [61]. However, for filter order 35×35, the MME is around 1/70 of that obtained in [56]. Our CPU time is listed in Table 6.3. Fig. 6.5 plots the convergence curve of the 27x27 filter, which shows that the optimization converged around 5,500 iterations but within the specified maximum number of iterations. On the other hand, the convergence curve of the 35x35 filter still slowly decreases after the maximum number of iterations is reached, but the MME value is already less than the criterion, as shown in Fig. 6.8. The two different filter orders demonstrate the elitist TLBO algorithm can achieve smaller MME with CLS and WLS approach.

In Example 2, a minimax design of two higher-order 2-D circularly symmetric FIR lowpass digital filters obtained by the elitist TLBO algorithm is presented. The MME values obtained are approximately 3/5 of those obtained in [61] as shown in Table 6.5. Table 6.6 shows the passband and stopband MME values of the elitist TLBO. By increasing the filter size from 39x39 to 43x43, the obtained MME values in passband and stopband are reduced. Table 6.7 lists the CPU time for the elitist TLBO. Figs. 6.11 and 6.14 give the error convergence plots of the 39x39 and 43x43 filters. The two convergence curves are still slowly declining even their MME values obtained are already smaller than those obtained in [61].

Therefore, those four designs indicate that the elitist TLBO algorithm is effective and can achieve smaller magnitude errors comparable to those obtained by the design methods [56] and [61].

6.4 Conclusions

In this chapter, CLS, WLS, and minimax designs of 2-D circularly symmetric FIR lowpass digital filters using the elitist TLBO algorithm have been presented. The CLS, WLS, and MM design results show that the elitist TLBO algorithm exhibits good performance in 2-D linear phase FIR filter designs. From the two examples, the optimization using the elitist TLBO can obtain much smaller

MME values compared to the two state-of-the-art methods in [56] and [61]. Design results obtained by the elitist TLBO show that the error convergence curves of the two filters each with 2 different filter sizes drop rapidly at the beginning and then decline slowly and competitive MME values can be obtained within the maximum number of iterations. All the filter designs were carried out on an Intel Core i7-4810MQ, 2.8 GHz (3.8 G Hz turbo) with 16GB RAM laptop computer.

CHAPTER 7

Two-Dimensional Nonlinear Phase FIR filters Design

2-D (two-dimensional) nonlinear phase FIR filters [62]-[66] have attracted attention because their lower group delay is more desirable in some applications as compared to 2-D linear phase FIR filters [55]-[61]. Under normal conditions, the transfer function of a 2-D FIR digital filter cannot be decomposed. In [62], an iterative alternating optimization technique for optimally designing the separable 2D FIR filter in the mini-max sense and the least-square sense was proposed. In [63], a fast matrix-based algorithm was developed for the elliptic-error and phase-error constrained least-squares designs of 2-D nonlinear phase FIR filters with arbitrarily specified frequency responses. Due to a matrix-based algorithm, the filter's coefficient matrix can be solved directly and faster than that of a vector-based algorithm. Two filter examples including a 2-D circularly symmetric FIR lowpass filter and a 2-D FIR fan filter were used to demonstrate the design method. The minimax criterion minimizes the maximum approximation error for a given size, equivalently achieving the goal of minimizing the size when given a maximum approximation error. In the two-dimensional case, as the alternating theorem has not been established and the optimal solution of the minimax approximation is not unique, the algorithm has met some convergence problems. On the other hand, the weighted least squares criterion is another approach for 2-D FIR filter design. The least squares solution can be improved with a proper weight function. However, this least squares algorithm also has some convergence problems, especially in 2-D nonlinear phase FIR filter designs.

In this chapter, the WLS method is applied to design 2-D nonlinear phase FIR digital filters using TLBO algorithm. Section 7.1 describes the equations of 2-D nonlinear phase FIR digital filters. In Section 7.2, four examples are designed using TLBO and compared to a state-of-the-art method. Section 7.3 provides the experiment analysis and conclusions.

7.1 2-D Nonlinear Phase FIR Filter Design Problem

7.1.1 2-D FIR digital filter

The transfer function of a 2-dimensional nonlinear phase FIR digital filter [80] can be expressed as

$$H(z_1, z_2) = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} c(n_1T_1, n_2T_2) {z_1}^{-n_1} {z_2}^{-n_2} \tag{7.1}$$

The coefficients $h(n_1, n_2)$ for $n_1 \geq 0$ and $n_2 \geq 0$ represent the impulse response values of the 2-dimensional FIR digital filter.

The frequency response of the 2-dimensional FIR digital filter can be evaluated by substituting $z_i = e^{j\omega_i T_i}$ for $i = 1, 2$ into its digital transfer function in (7.1) as

$$\begin{aligned} H(\omega_1, \omega_2) &= \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} c(n_1, n_2) e^{-j(n_1\omega_1T_1 + n_2\omega_2T_2)} \\ &= |H(\omega_1, \omega_2)| e^{j\theta(\omega_1, \omega_2)} \end{aligned} \tag{7.2}$$

In (7.2), the magnitude response $H(\omega_1, \omega_2)$ is equal to

$$H(\omega_1, \omega_2) = \left\{ \begin{array}{l} \left[\sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} c(n_1, n_2)\, cos(n_1\omega_1T_1 + n_2\omega_2T_2)\right]^2 \\ + \left[\sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} c(n_1, n_2) sin(n_1\omega_1T_1 + n_2\omega_2T_2)\right]^2 \end{array} \right\}^{1/2} \tag{7.3}$$

7.1.2 Objective function

The desired frequency response of a 2-dimensional nonlinear phase circularly symmetric lowpass FIR digital filter is defined by

$$H_d(\omega_1, \omega_2) = \begin{cases} e^{-j(\tau_1\omega_1 + \tau_2\omega_2)} & \text{for } 0 \leq \sqrt{\omega_1^2 + \omega_2^2} \leq \omega_p \\ 0 & \text{for } \omega_s \leq \sqrt{\omega_1^2 + \omega_2^2} \leq \pi \end{cases} \tag{7.4}$$

where τ_1 and τ_2 are the desired passband group delays along the ω_1 and ω_2 directions.

The WLS objective function of frequency response error is defined as

$$e(\boldsymbol{c}) = \sum_{(\omega_1,\omega_2)\epsilon\Omega_I} W_{12}|H(\boldsymbol{c}, \omega_{1i}, \omega_{2i}) - H_d(\omega_{1i}, \omega_{2i})|^2 \qquad (7.5)$$

where Ω_I denotes a union of frequencies points of interest in passband and stopband; W_{12} denotes the frequency weight at a frequency point (ω_1, ω_2).

7.2 Filter Design Examples

In this section, 2-D nonlinear phase FIR lowpass filters are designed using an elitist TLBO algorithm [207]. The elitist replacement is applied by replacing the worst learner in the population by the teacher solution and with added mutations on one or more randomly selected dimensions for each duplicated solution before proceeding to the next generation. The cutoff frequency specifications of the 2-dimensional nonlinear phase FIR filters are listed in Table 7.1 and plotted in Fig. 7.1, where the green areas denote the area of frequency points of interest for optimization. For optimization, the size of the frequency grid is taken as $I \times J$ with $I = J = 8N$ for $0 \leq \omega_{1i}, \omega_{2j} \leq \pi$. The initial coefficients of each filter design are set randomly.

The filter examples and the mean squared complex error (MSCE) results obtained using TLBO and by the Example 1 of [63] are listed in Table 7.2. Table 7.3 lists the numbers of iterations and CPU time (in seconds) of the four filter designs using an Intel Core i7-4810MQ, 2.8 GHz (3.8 GHz turbo) with 16GB RAM laptop computer. The magnitude responses in dB and the error convergence curves of the four designed filters are shown in Fig. 7.2 to Fig. 7.13 respectively.

$$MSCE = \left[\sum_{\Omega_P} e_p(\boldsymbol{c})^2 + \sum_{\Omega_S} e_s(\boldsymbol{c})^2\right] / N_{count} \qquad (7.6)$$

where Ω_P denotes a union of frequencies points of interest in passband; and Ω_S denotes a union of frequencies points of interest in stopband. $e_p(\boldsymbol{c})$ and $e_s(\boldsymbol{c})$ are the WLS objective function of frequency response error in passband and stopband, respectively. N_{count} is the number of points for evaluation.

Table 7.1 Cutoff Frequencies of 2-D Nonlinear Phase FIR Lowpass Filters

Cutoff frequencies	
Passband	$0.40\boldsymbol{\pi}$
Stopband	$0.60\boldsymbol{\pi}$

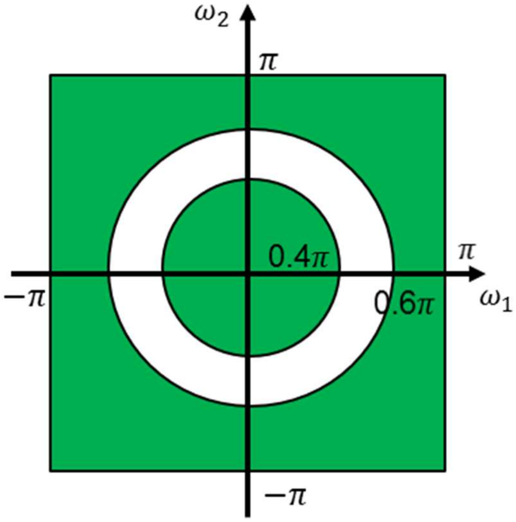


Fig. 7.1 Cutoff frequencies of 2-D nonlinear phase FIR lowpass filters

Table 7.2 2-D Nonlinear Phase FIR Lowpass Filter Design Results

Specifications	Algorithm	MSCE
N_1=13, N_2=13 $\tau_1 = \tau_2 = 5$	TLBO	3.6123e-04
	Design Example 1 [63]	4.7362e-04
N_1=15, N_2=15 $\tau_1 = \tau_2 = 6$	TLBO	2.5682e-04
	Design Example 1 [63]	3.9803e-04
N_1=21, N_2=21 $\tau_1 = \tau_2 = 9$	TLBO	4.5820e-05
	Design Example 1 [63]	3.0317e-04
N_1=31, N_2=31 $\tau_1 = \tau_2 = 13.5$	TLBO	3.4669e-06
	Design Example 1 [63]	1.5069e-04

Table 7.3 Number of Iterations and CPU Time of TLBO

Filter orders	Iterations	CPU Time (s)
N_1=N_2=13	500,000	3.8956e+03
N_1=N_2=15	1,000,000	1.3544e+04
N_1=N_2=21	500,000	6.3681e+03
N_1=N_2=31	200,000	1.7904e+05

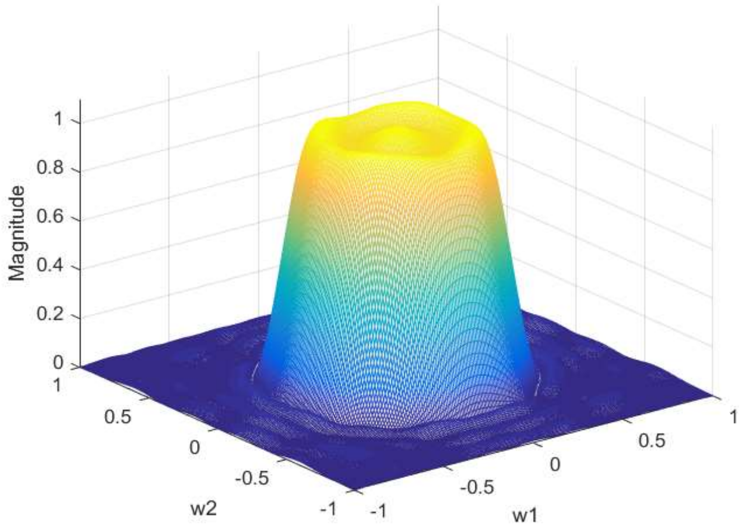


Fig. 7.2 Magnitude response of 13×13 2-D nonlinear phase FIR lowpass filter design by TLBO

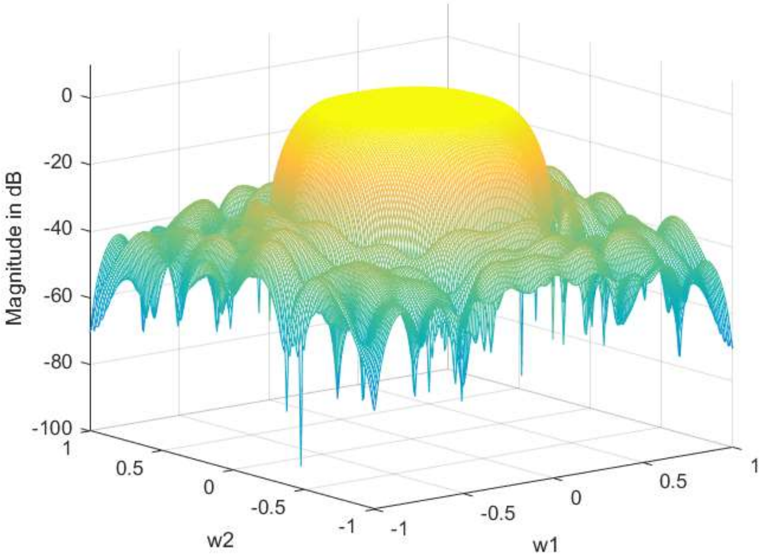


Fig. 7.3 Magnitude response in dB of 13×13 2-D nonlinear phase FIR lowpass filter design by TLBO

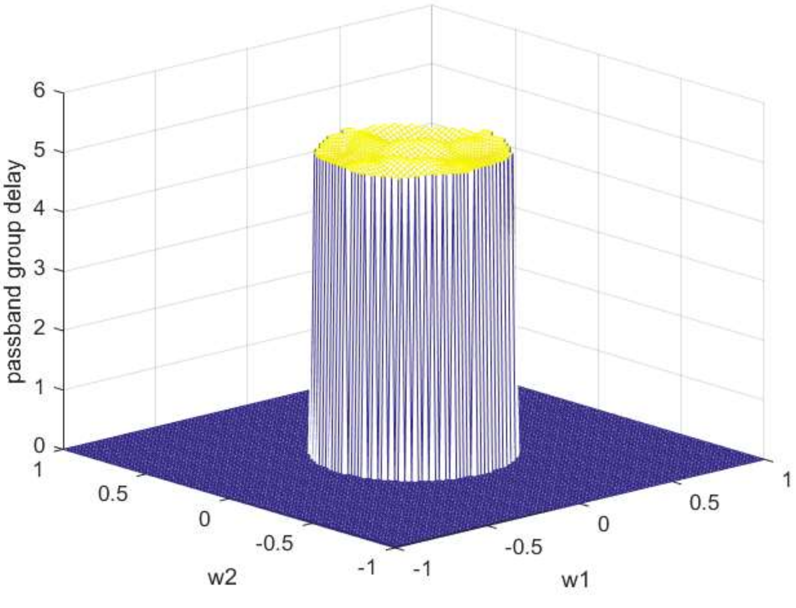


Fig. 7.4 Passband group delay of 13×13 2-D nonlinear phase FIR lowpass filter design by TLBO

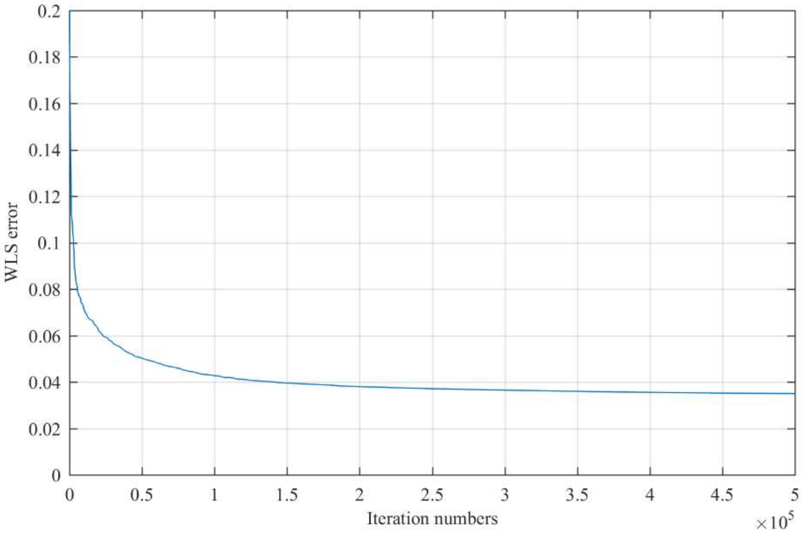


Fig. 7.5 WLS error convergence curve of 13×13 2-D nonlinear phase FIR lowpass filter design by TLBO

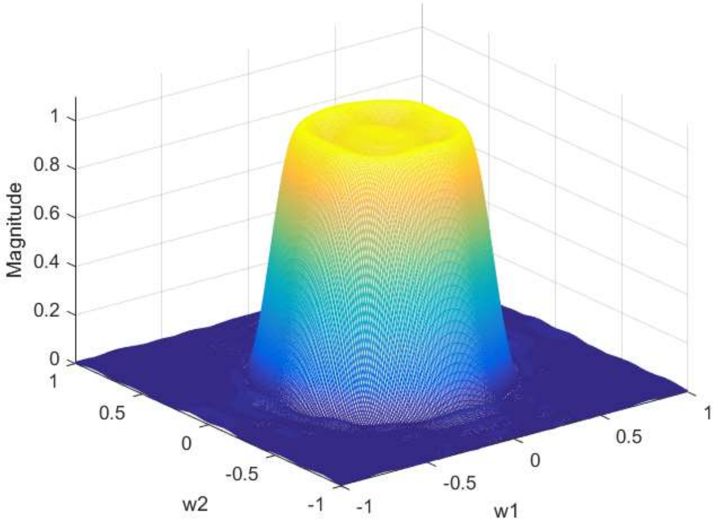


Fig. 7.6 Magnitude response of 15×15 2-D nonlinear phase FIR lowpass filter design by TLBO

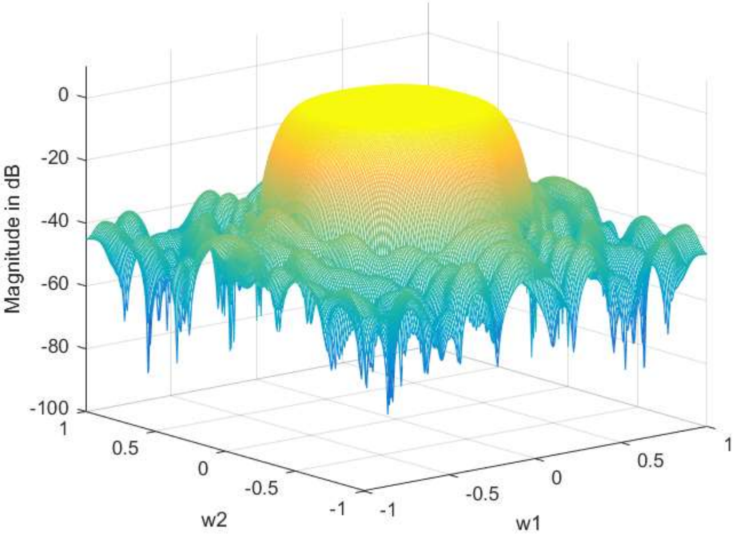


Fig. 7.7 Magnitude response in dB of 15×15 2-D nonlinear phase FIR lowpass filter design by MOTLBO

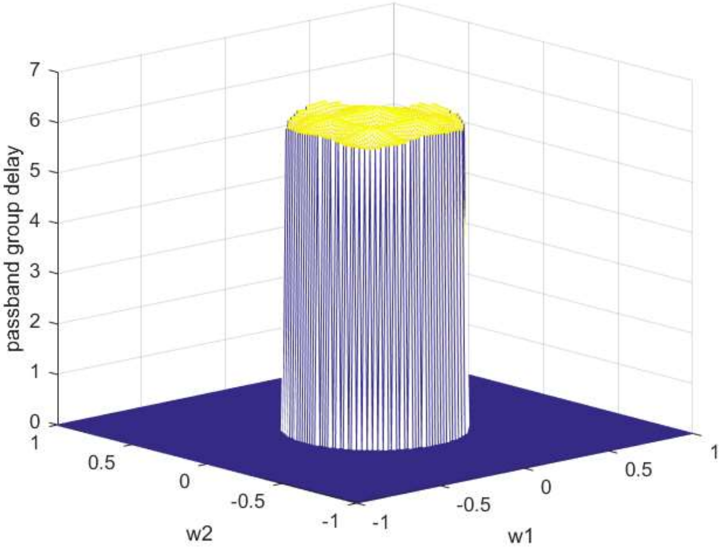


Fig. 7.8 Passband group delay of 15×15 2-D nonlinear phase FIR lowpass filter design by TLBO

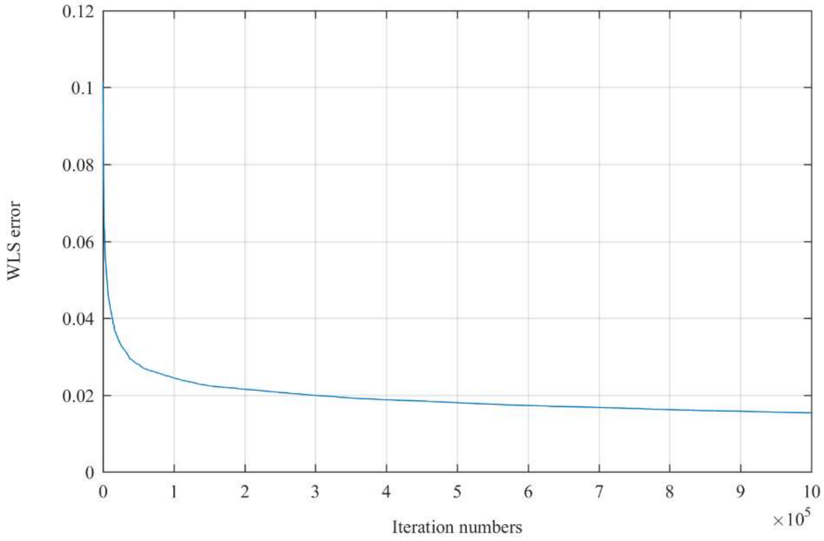


Fig. 7.9 WLS error convergence of 15×15 2-D nonlinear phase FIR lowpass filter design by TLBO

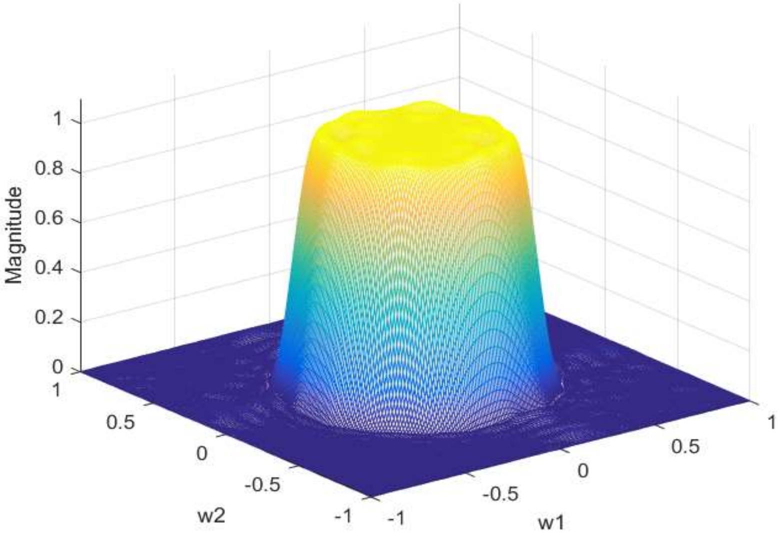


Fig. 7.10 Magnitude response of 21×21 2-D nonlinear phase FIR lowpass filter design by TLBO

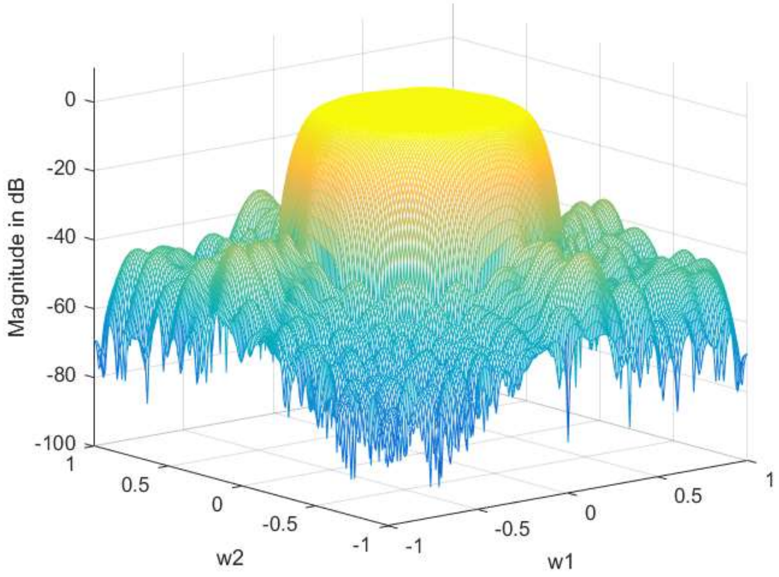


Fig. 7.11 Magnitude response in dB of 21×21 2-D nonlinear phase FIR lowpass filter design by TLBO

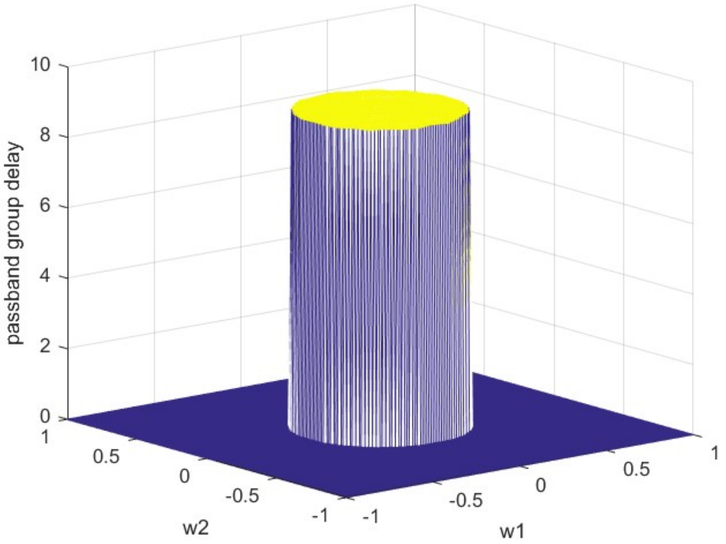


Fig. 7.12 Passband group delay of 21×21 2-D nonlinear phase FIR lowpass filter design by TLBO

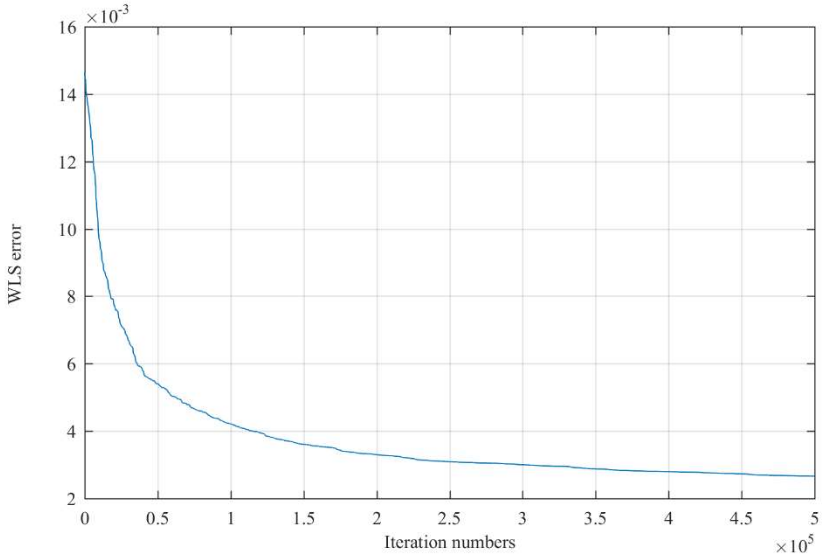


Fig. 7.13 WLS error convergence of 21×21 2-D nonlinear phase FIR lowpass filter design by TLBO

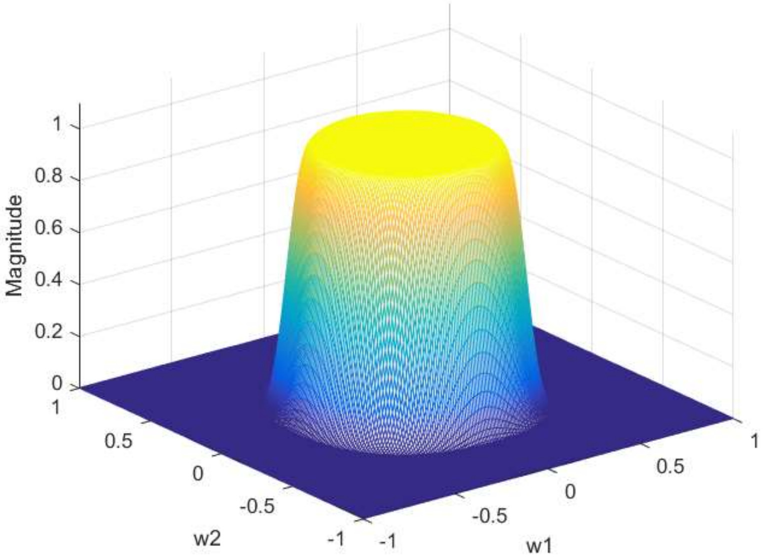


Fig. 7.14 Magnitude response of 31×31 2-D nonlinear phase FIR lowpass filter design by TLBO

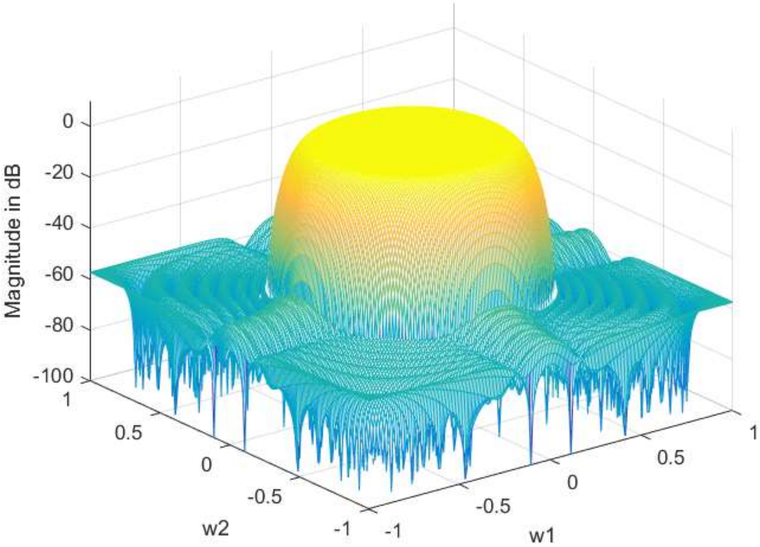


Fig. 7.15 Magnitude response in dB of 31×31 2-D nonlinear phase FIR lowpass filter design by MOTLBO

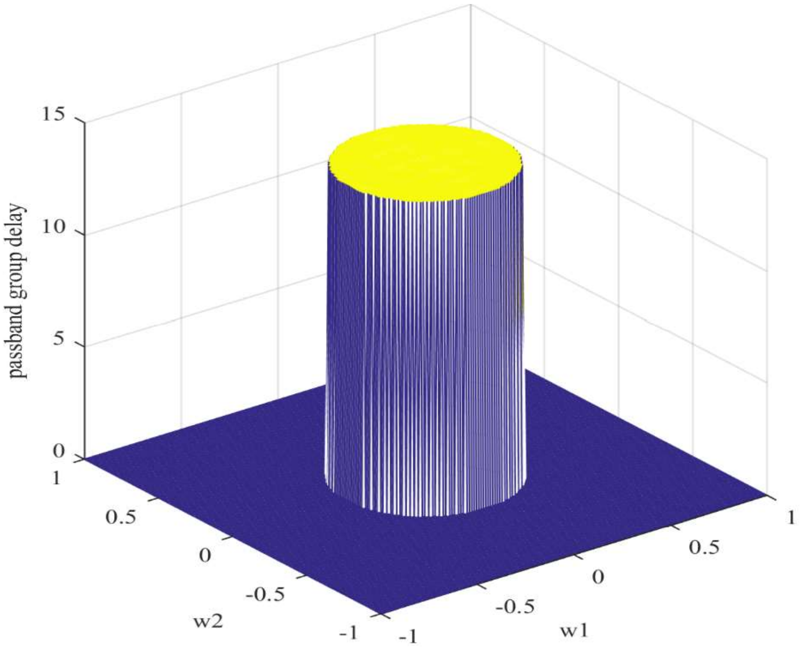


Fig. 7.16 Passband group delay of 31×31 2-D nonlinear phase FIR lowpass filter design by TLBO

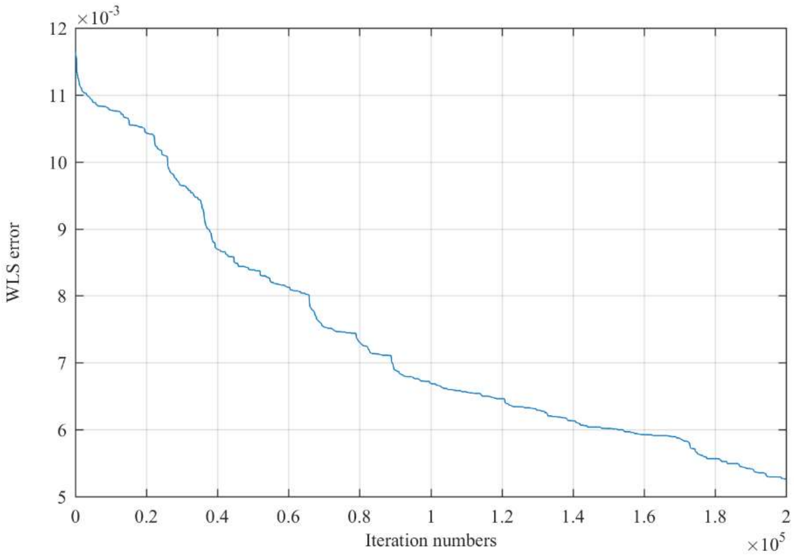


Fig. 7.17 WLS error convergence of 31×31 2-D nonlinear phase FIR lowpass filter design by TLBO

7.3 Analysis

In this chapter, four designs of the WLS design of 2-D nonlinear phase FIR digital filters using the TLBO algorithm are presented. Mean squared complex errors (MSCEs) are selected as the evaluation criteria. The design examples consisting of four different filter orders of 2-D nonlinear phase lowpass FIR filter and their design results are compared to those obtained by the-state-of-the-art method in [63] as shown in Table 7.2. The design results obtained by TLBO in terms of MSCEs are about 7/10 and 8/10 of those in [63] for filter orders of 13×13 and 15×15 respectively; and about 1/7 and 1/43 of those in [63] for filter orders of 21x21 and 31x31 respectively7.4.

The maximum number of iterations for filter orders of 13×13 and 21×21 are 500,000 iterations, filter order of 15×15 is 1,000,000 iterations, while for filter order of 31×31 is 200,000 iterations. Figs. 7.4, 7.7, 7.10 and 7.13 show the error convergence of the four filter designs, each of them decreases fast at the beginning and then declines slowly. Although the error convergence curves of the four filter designs are still slowly declining after the maximum number of iterations, all the obtained MSCE values have already met their design criteria and are smaller than those of [63].

7.4 Conclusions

In this chapter, the WLS design of 2-D nonlinear phase FIR digital filters using the elitist TLBO algorithm has been presented. The 2-D nonlinear phase FIR filters are more complex than 2-D linear phase FIR filters because more coefficients are needed to be optimized. However, the elitist TLBO can obtain smaller errors than those obtained by the state-of-the-art method in [63], especially for higher-order filter designs. All four filter designs are capable of reaching the desired errors within the maximum number of iterations. All the filter designs were carried out on an Intel Core i7-4810MQ, 2.8 GHz (3.8 G Hz turbo) with 16GB RAM laptop computer.

CHAPTER 8

Conclusions and Future Study

8.1 Conclusions

In this dissertation, TLBO and improved algorithms have been presented and used to design five different types of digital filters.

Chapter 3 has presented least squares and minimax designs of linear phase FIR Hilbert transformer using the standard TLBO algorithm and the gradient-based TLBO algorithm. The results indicated that the gradient-based TLBO is a more efficient method as compared to the standard TLBO algorithm. Though both algorithms can obtain very close LS and MM errors, the gradient-based TLBO algorithm can achieve slightly lower peak LS and MM errors and requires less CPU time for optimization. Similar results are obtained when compared to a state-of-the-art method in [143]. A tuning of the learning rate parameter is required for the gradient-based learning phase. Experiments reveal that an appropriate choice of the learning rate parameter could make the gradient-based TLBO algorithm more efficient.

In the first part of Chapter 4, minimax design of general FIR filters using the non-dominated MOTLBO (with crowding distance) has been presented. As compared to the (non-dominated) MOPSO (with crowding distance), the design results indicate that the non-dominated MOTLBO can obtain smaller magnitude and group delay minimax errors as well as smaller passband and stopband peak errors. Except for BS filter design, the non-dominated MOTLBO usually requires less CPU time than the MOPSO. The second part of Chapter 4 has presented the use of the gradient-based MOTLBO (with Manhattan distance) for least squares design of general FIR filters. The design results are compared to those obtained by the non-dominated MOTLBO (with crowding distance) and the (non-dominated) MOPSO (with crowding distance). The comparison has shown that the gradient-based MOTLBO with Manhattan distance exhibits the best performance in nearly all least squares errors and peak errors. For the non-dominated MOTLBO and the MOPSO, the comparison has shown that the non-dominated MOTLBO performs better than the MOPSO in all least squares magnitude errors, and in least squares group delay error and peak group delay errors for highpass filter design; and there is a mixed performance in least squares peak magnitude errors.

Chapter 5 has presented the Euclidean-distance-based MOTLBO to design cascade IIR filters by minimizing passband and stopband peak magnitude errors, and passband peak group delay error, subject to transition band magnitude constraint. A comparison to the design results of the lowpass, highpass, and bandpass filter examples of the state-of-the-art design method in [54] indicated that the Euclidean-distance-based MOTLBO can obtain slightly smaller passband and stopband peak errors and the passband group delay peak error.

Chapter 6 has presented the constraint least squares (CLS), weighted least squares (WLS), and minimax (MM) designs of 2-D circularly symmetric FIR lowpass digital filters using the elitist TLBO algorithm. A 45° discrete frequency grid is used for optimization. The obtained WLS, and MM designs are compared to those obtained by [63] and the obtained CLS design is compared to that of [56] (both of [63] and [56] are the state-of-the-art design methods) to demonstrate that the elitist TLBO algorithm can obtain lower maximum magnitude errors (MME) in 2-D linear phase FIR filter design.

Chapter 7 has presented the use of the elitist TLBO algorithm to design 2-D nonlinear phase FIR digital filters with WLS approximation. A 2-D nonlinear phase lowpass FIR filter example of four different filter orders are designed and compared to those obtained by a state-of-the-art design method [63] with results indicating that lower mean squared complex errors (MSCEs) can be obtained by the elitist TLBO algorithm.

8.2 Future works

A few topics worthy for future research are briefly described as follows:

Study on faster convergence speed on multiobjective design problems. This dissertation has presented TLBO and its improved algorithms for single and multiobjective digital filter design problems. In general, it is desirable to further reduce the required CPU time especially when applied to complex multiobjective design problems.

Study on improved performance on complex design problems. The proposed gradient-based algorithm has achieved good results in optimizing linear phase FIR filters and nonlinear phase FIR filters. A future study could be on its performance improvements when applied to more complex design problems.

This dissertation has mainly described the application of TLBO and its improved algorithms for digital filter design. There are other evolutionary algorithms, their effectiveness for digital filter design is a topic worthy of further research.

REFERENCES

[1] H. K. Kwan, *Multimedia Signal Processing and Applications*. dfisp.org, edition 1.3, May 19, 2018 (717 pages). ISBN-13: 978-1-988-30703-9

[2] S. K. Mitra, *Digital Signal Processing*. New York: McGraw-Hill, 4th edition, Sep. 10, 2010 (992 pages). ISBN-13: 978-0-073-38049-0

[3] P. S. R. Diniz, E. A. B. da Silva, and S. L. Netto, *Digital Signal Processing: System Analysis and Design*. London: Cambridge University Press, 2nd edition, Aug. 2010 (889 pages). ISBN-13: 978-0-521-88775-5

[4] Y. C. Lim, H. K. Kwan, and W.-C. Siu, Eds. *Trends in Digital Signal Processing - A Festschrift in Honour of A. G. Constantinides* (Pan Stanford Series on Digital Signal Processing - Volume 1). Singapore: Pan Stanford, Jul. 31, 2015 (Hardback, 577 pages). Print ISBN: 9789814669504, eBook ISBN: 9789814669511, DOI: 10.4032/9789814669511

[5] V. Cappellini, A. G. Constantinides, and P. Emiliani, *Digital Filters and Their Applications*. London, New York: Academic Press, 1978 (393 pages). ISBN 0121592502

[6] A. Antoniou, *Digital Filters: Analysis, Design, and Signal Processsing Applications*. Toronto, New York, London: McGraw-Hill, 2018 (952 pages). ISBN 978 0-07-184603-5

[7] W.-S. Lu and A. Antoniou, *Two-Dimensional Digital Filters*. New York, Hong Kong: Marcel Dekker, 1992 (398 pages). ISBN 0-8247-8434-0

[8] B. A. Shenoi, *Magnitude and Delay Approximation of 1-D and 2-D Digital Filters*. Berlin, Heidelberg, New York: Springer-Verlag, 1999 (250 pages). ISBN 3-540-64161-0

[9] R. King, M. Ahmadi, R. Gorgui-Naguib, A. Kwabwe, and M. Azimi-Sadjadi, *Digital Filtering in One and Two Dimensions*. New York, London: Plenum Press, 1989 (466 pages). ISBN 0-306-42976-4

[10] A. G. Constantinides and H. K. Kwan, “Analysis of 1-D digital filter structures,” in *Proc. of IEEE International Symp. on Circuits and Systems (ISCAS 1982)*, Rome, Italy, May 10-12, 1982, vol. 3, pp. 1233-1236.

[11] H. K. Kwan and A. G. Constantinides, “Analysis of 2-D digital filter structures,” in *Proc. of IEEE International Conf. on Computer, System and Signal Processing (ICCSSP 1984)*, Bangalore, India, Dec. 10-12, 1984, pp. 1636-1639.

[12] H. K. Kwan and A. G. Constantinides, “Analysis of 3-D digital filter structures,” in *Proc. of IEEE International Symp. on Circuits and Systems (ISCAS 1985)*, Kyoto, Japan, Jun. 5-7, 1985, vol. 3, pp. 1567-1570.

[13] H. K. Kwan, “Poles and zeros determination of arbitrary linear digital networks,” *IEEE Trans. on Circuits and Systems*, vol. CAS-30, no. 12, pp. 919-921, Dec. 1983.

[14] H. K. Kwan, “Amplitude scaling in arbitrary linear digital networks,” *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. ASSP-32, no. 6, pp. 1240-1242, Dec. 1984.

[15] H. K. Kwan and A. G. Constantinides, “Roundoff noise in arbitrary linear digital networks,” *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. ASSP-33, no. 3, pp. 734-737, Jun. 1985.

[16] H. K. Kwan, “A multi-output second-order digital filter structure for VLSI implementation,” *IEEE Trans. on Circuits and Systems*, vol. CAS-32, no. 1, pp. 108-109, January 1985.

[17] H. K. Kwan, “A multi-output first-order digital filter structure for VLSI implementation,” *IEEE Trans. on Circuits and Systems*, vol. CAS-32, no. 9, pp. 973-974, Sep. 1985.

[18] H. K. Kwan, “A multi-output second-order digital filter without limit cycle oscillations,” *IEEE Trans. on Circuits and Systems*, vol. CAS-32, no. 9, pp. 974-975, Sep. 1985.

[19] H. K. Kwan, "A multi-output wave digital biquad using magnitude truncation instead of controlled rounding," *IEEE Trans. on Circuits and Systems*, vol. CAS-32, no. 11, pp. 1185-1187, Nov. 1985.

[20] H. K. Kwan, "New realisations of first-order two-dimensional all-pass and all-pole digital filters," *Electronics Letters*, vol. 24, no. 4, pp. 224-226, Feb. 18, 1988.

[21] H. K. Kwan and Y. C. Lui, "Normalized ARMA Schur algorithm," in *Proc. of 32nd IEEE International Midwest Symp. on Circuits and Systems (MWSCAS 1989)*, Urbana-Champaign, Illinois, U.S.A., Aug. 14-16, 1989, vol. 2, pp. 1103-1106.

[22] H. K. Kwan and Y. C. Lui, "Cordic implementation of normalized ARMA Schur algorithm," in *Proc. of 32nd IEEE International Midwest Symp. on Circuits and Systems (MWSCAS 1989)*, Urbana-Champaign, Illinois, U.S.A., Aug. 14-16, 1989, vol. 1, pp. 369-372.

[23] H. K. Kwan and Y. C. Lui, "Minimal normalized lattice structure for ARMA digital filter realization," in *Proc. of IEEE International Conf. on Acoustics, Speech and Signal Processing (ICASSP 1991)*, Toronto, Ontario, Canada, May 14-17, 1991, vol. 3, pp. 1629-1632.

[24] J. H. McClellan, T. W. Parks, and L. R. Rabiner, "A computer program for designing optimum FIR linear phase digital filters," *IEEE Trans. on Audio and Electroacoustics*, vol. AU-21, no. 6, pp. 506-526, 1973.

[25] H. K. Kwan and J. Liang, "Design of linear phase FIR filters using cuckoo search algorithm," in *Proc. of 8th International Conf. on Wireless Communications and Signal Processing (WCSP 2016)*, Yangzhou, Jiangsu, China, Oct. 13-15, 2016, 4 pages.

[26] H. K. Kwan and R. Raju, "Minimax design of linear phase FIR differentiators using artificial bee colony algorithm," in *Proc. of 8th International Conf. on Wireless Communications and Signal Processing (WCSP 2016)*, Yangzhou, Jiangsu, China, Oct. 13-15, 2016, 4 pages.

[27] H. K. Kwan and M. Zhang, “Minimax design of linear phase FIR Hilbert transformer using teaching-learning-based optimization,” in *Proc. of 8th International Conf. on Wireless Communications and Signal Processing (WCSP 2016)*, Yangzhou, Jiangsu, China, Oct. 13-15, 2016, 4 pages.

[28] R. Raju and H. K. Kwan, “FIR filter design using multiobjective artificial bee colony algorithm,” in *Proc. of 2017 IEEE 30th Canadian Conf. on Electrical and Computer Engineering (CCECE 2017)*, Windsor, Ontario, Canada, Apr. 30-May 3, 2017, 4 pages.

[29] J. Liang and H. K. Kwan, “FIR filter design using multiobjective cuckoo search algorithm,” in *Proc. of 2017 IEEE 30th Canadian Conf. on Electrical and Computer Engineering (CCECE 2017)*, Windsor, Ontario, Canada, Apr. 30-May 3, 2017, 4 pages.

[30] M. Zhang and H. K. Kwan, “FIR filter design using multiobjective teaching-learning-based optimization,” in *Proc. of 2017 IEEE 30th Canadian Conf. on Electrical and Computer Engineering (CCECE 2017)*, Windsor, Ontario, Canada, Apr. 30-May 3, 2017, 4 pages.

[31] H. K. Kwan, “Asymmetric FIR filter design using evolutionary optimization,” in *Proc. of 2017 IEEE 30th Canadian Conf. on Electrical and Computer Engineering (CCECE 2017)*, Windsor, Ontario, Canada, Apr. 30-May 3, 2017, 4 pages.

[32] H. K. Kwan, “Unconstrained asymmetric FIR filter design using harmony search algorithm,” in *Proc. of 2018 IEEE 31th Canadian Conf. on Electrical and Computer Engineering (CCECE 2018)*, Quebec City, Quebec, Ontario, Canada, May 13-16, 2018, 4 pages.

[33] H. K. Kwan, “On the problem of designing IIR digital filters with short coefficient word lengths,” *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. ASSP-27, no. 6, Part 1, pp. 620-624, Dec. 1979.

[34] H. K. Kwan, “Design of passive second-order digital filters,” in *Proc. of IEEE International Conf. on Acoustics, Speech, and Signal Processing (ICASSP 1982)*, Paris, France, May 3-5, 1982, vol. 1, pp. 286-289.

[35] H. K. Kwan, “One-dimensional passive state-space digital filter design,” in *Proc. of IEEE International Symp. on Circuits and Systems (ISCAS 1984)*, Montreal, Quebec, Canada, May 7-10, 1984, vol. 1, pp. 250-253.

[36] H. K. Kwan, “Importance of feedback gain in adaptive IIR digital filtering,” *Electronics Letters*, vol. 29, no. 12, pp. 1142-1144, Jun. 10, 1993.

[37] H. K. Kwan, “Tunable and variable passive digital filters for multimedia signal processing,” in *Proc. of IEEE International Symp. on Intelligent Multimedia, Video & Speech Processing (ISIMP 2001)*, Hong Kong, China, May 2-4, 2001, pp. 229-232.

[38] H. K. Kwan, “High-order tunable passive digital filters,” in *Proc. of IEEE International Symp. on Circuits and Systems (ISCAS 2002)*, Scottsdale, Arizona, U.S.A., vol. II, May 26-29, 2002, pp. 700-703.

[39] H. K. Kwan, “Adaptive IIR digital filters with saturation outputs for noise and echo cancellation,” *Electronics Letters*, vol. 38, no. 13, pp. 661-663, Jun. 20, 2002.

[40] H. K. Kwan and A. Jiang, “Recent advances in FIR approximation by IIR digital filters,” in *Proc. of International Conf. on Communications, Circuits and Systems (ICCCAS 2006)*, Guilin, China, June 25-28, 2006, pp. 185-190.

[41] A. Jiang and H. K. Kwan, “IIR digital filter design with new stability constraint based on argument principle,” *IEEE Trans. on Circuits and Systems I: Regular Papers*, vol. 56, no. 3, pp. 583-593, Mar. 2009.

[42] A. Jiang and H. K. Kwan, “Minimax design of IIR digital filters using SDP relaxation technique,” *IEEE Trans. on Circuits and Systems I: Regular Papers*, vol. 57, no. 2, pp. 378-390, Feb. 2010.

[43] A. Jiang and H. K. Kwan, “Minimax design of IIR digital filters using iterative SOCP,” *IEEE Trans. on Circuits and Systems I: Regular Papers*, vol. 57, no. 6, pp. 1326-1337, Jun. 2010.

[44] Xiaoping Lai, Zhiping Lin, and H. K. Kwan, “A sequential minimization procedure for minimax design of IIR filters based on second-order factor updates,” *IEEE Trans. on Circuits and Systems II: Express Briefs*, vol. 58, no. 1, pp. 51-55, Jan. 2011.

[45] A. Jiang, H. K. Kwan, Y. Zhu, N. Xu, and X. Liu, “Efficient WLS design of IIR digital filters using partial second-order factorization,” *IEEE Trans. on Circuits and Systems II: Express Briefs*, vol. 63, no. 7, pp. 703-707, 15 Feb. 2016.

[46] J. Liang and H. K. Kwan, “IIR filter design using constrained multiobjective cuckoo search algorithm,” in *Proc. of 2018 IEEE 31th Canadian Conf. on Electrical and Computer Engineering (CCECE 2018)*, Quebec City, Quebec, Ontario, Canada, 13-16 May 2018, pp. 1-4.

[47] H. K. Kwan and M. Zhang, “IIR filter design using multiobjective teaching learning-based optimization,” in Proc. of 2018 IEEE 31th Canadian Conf. on Electrical and Computer Engineering (CCECE 2018), Quebec City, Quebec, Ontario, Canada, 13-16 May 2018, pp. 1-4.

[48] R. Raju and H. K. Kwan, “IIR filter design using multiobjective artificial bee colony algorithm,” in *Proc. of 2018 IEEE 31th Canadian Conf. on Electrical and Computer Engineering (CCECE 2018)*, Quebec City, Quebec, Ontario, Canada, 13-16 May 2018, pp. 1-4.

[49] H. K. Kwan, “Allpass group delay equalizer design using harmony search algorithm,” in *Proc. of 2018 IEEE 23rd International Conf. on Digital Signal Processing (DSP 2018)*, Shanghai, China, Nov. 2018, pp. 1-4.

[50] H. K. Kwan, “Group delay equalizer design using harmony search algorithm,” in *Proc. of IEEE 61st International Midwest Symp. on Circuits and Systems (MWSCAS 2018)*, Windsor, Ontario, Canada, Aug. 2018, pp. 688-691.

[51] Y. Wang, B. Li, and Y. Chen, “Digital IIR filter design using multi-objective optimization evolutionary algorithm,” *Applied Soft Computing*, vol. 11, no. 2, pp. 1851-1857, 2011.

[52] Y. Wang, B. Li, and Z. Li, "Fixed-point digital IIR filter design using multi-objective optimization evolutionary algorithm," in *Proc. IEEE Youth Conference on Information, Computing and Telecommunications (YC-ICT 2010)*, Beijing, China, Nov. 28-30, 2010, pp. 174-177.

[53] K. K. Dhaliwal and J. S. Dhillon, "Integrated cat swarm optimization and differential evolution algorithm for optimal IIR filter design in multi-objective framework," *Circuits Systems and Signal Processing*, vol. 36, no. 1, pp. 270-296, 2017.

[54] R. C. Nongpiur, D. J. Shpak, and A. Antoniou, "Improved design method for nearly linear-phase IIR filters using constrained optimization," *IEEE Trans. Signal Processing*, vol. 61, no. 4, pp. 895-906, Feb. 2013.

[55] H. K. Kwan and C. L. Chan, "Circularly symmetric two-dimensional multiplierless FIR digital filter design using an enhanced McClellan transformation," *IEE Proc. Part G on Circuits, Devices and Systems*, vol. 136, no. 3, pp. 129-134, Jun. 1989.

[56] X. Hong, X. Lai, and R. Zhao, "Matrix-based algorithms for constrained least-squares and minimax designs of 2-D linear-phase FIR filters," *IEEE Trans. Signal Processing*, vol. 61, no. 14, pp. 3620-3631, 2013.

[57] H. K. Kwan and M. Zhang, "2-D linear phase FIR digital filter design using teaching-learning-based optimization," in *Proc. of 2018 IEEE 23rd International Conf. on Digital Signal Processing (DSP 2018)*, Shanghai, China, Nov. 2018, pp. 1-4.

[58] F. Wysocka-Schillak, "Design of equiripple 2-D linear-phase FIR digital filters using genetic algorithm," in *Proc. Int. Conf. on Numerical Methods and Applications (NMA 2006)*, 2006, pp. 263-270.

[59] T. Shinohara, Y. Nishida, and N. Aikawa, "A design method for 2-D low-pass maximally flat FIR digital filter to realize various passband shapes," in *Proc. of 42nd Annual Conf. of the IEEE Industrial Electronics Society (IECON2016)*, 2016, pp. 821-825.

[60] S. Dhabal and P. Venkateswaran, “A novel accelerated artificial bee colony algorithm for optimal design of two dimensional FIR filter,” *Multidimensional Systems and Signal Processing*, vol. 28, no. 2, pp. 471-493, 2017.

[61] R. Zhao, X. Lai, X. Hong, and Z. Lin, “A matrix-based IRLS algorithm for the least l_p-norm design of 2-D FIR filters,” *Multidimensional Systems and Signal Processing*, vol. 30, no. 1, pp. 1-15, 2019.

[62] H. Wang, R. Zeng, X. Cheng, and Z. Jian, “An iterative technique for optimally designing a separable 2D FIR filter,” *IEEJ Trans. on Electrical and Electronic Engineering*, vol. 13, no. 4, pp. 622–626, 2018.

[63] X. Hong, X. Lai, and R. Zhao, “A fast design algorithm for elliptic-error and phase-error constrained LS 2-D FIR filters,” Multidimensional Systems and Signal Processing, vol. 27, no. 2, pp. 477-491, 2016.

[64] C.-H. Hsieh, C.-M. Kuo, Y.-D. Jou, and Y.-L. Han, “Design of two-dimensional FIR digital filters by a two-dimensional WLS technique,” *IEEE Trans. Circuits Syst. II: Analog Digit. Signal Process.*, vol. 44, no. 5, pp. 348–357, 1997.

[65] X. Lai and Y. Cheng, “A sequential constrained least-square approach to minimax design of 2-D FIR filters,” *IEEE Trans. Circuits Syst. II: Express Briefs*, vol. 54, no. 11, pp. 994-998, 2007.

[66] H. K. Kwan and M. Zhang, “2-D FIR digital filter design using teaching-learning-based optimization,” in *Proc. of IEEE 61st International Midwest Symp. on Circuits and Systems (MWSCAS 2018)*, Windsor, Ontario, Canada, Aug. 2018, pp. 684-687.

[67] H. K. Kwan and C. L. Chan, “Multidimensional spherically symmetric recursive digital filter design satisfying prescribed magnitude and constant group delay responses,” *IEE Proc. Part G on Electronic Circuits and Systems*, vol. 134, no.4, pp. 187-193, Aug. 1987.

[68] H. K. Kwan and C. L. Chan, “Design of two-dimensional circularly symmetric IIR digital low-pass filters by spectral transformation,” in *Proc. of IEEE Asian Electronics Conf.*, Hong Kong, Sep. 1-6, 1987, pp. 29-33.

[69] C. L. Chan and H. K. Kwan, “Simple and computationally efficient design of two-dimensional circularly symmetric IIR digital filters,” *International Journal of Electronics*, vol. 64, no. 2, pp. 229-237, Feb. 1988.

[70] H. K. Kwan and C. L. Chan, “Design of linear phase circularly symmetric two-dimensional recursive digital filters,” *IEEE Trans. on Circuits and Systems*, vol. CAS-36, no. 7, pp. 1023-1029, Jul. 1989.

[71] C. L. Chan and H. K. Kwan, “Simple design of 2-dimensional circularly symmetric recursive digital filters using 1-dimensional analogue filters,” *International Journal of Electronics*, vol. 67, no. 4, pp. 585-590, Oct. 1989.

[72] C. L. Chan and H. K. Kwan, “Multidimensional ellipsoidally and spherically symmetric recursive digital filter design with reduced number of optimised coefficients,” *IEE Proc. Part G on Circuits, Devices and Systems*, vol. 136, no. 6, pp. 301-305, Dec. 1989.

[73] C. L. Chan and H. K. Kwan, “Narrow transition width two-dimensional digital fan filters with sum of powers-of-two coefficients,” *International Journal of Electronics*, vol. 68, no. 3, pp. 391-398, Mar. 1990.

[74] H. K. Kwan and C. L. Chan, “Design of multidimensional spherically symmetric and constant group delay recursive digital filters with sum of powers-of-two coefficients,” *IEEE Trans. on Circuits and Systems*, vol. CAS-37, no. 8, pp. 1027-1035, Aug. 1990.

[75] H. K. Kwan, “Image data compression using 2-D lattice predictor,” *Electronics Letters*, vol. 20, no. 24, pp. 994-995, Nov. 22, 1984.

[76] H. K. Kwan and Y. C. Lui, “Image data compression using 2-D lattice modelling method,” *IEE Proc. Part F on Communications, Radar and Signal Processing*, vol. 134, no. 4, pp. 401-404, Jul. 1987.

[77] H. K. Kwan and Y. C. Lui, "Lattice implementation of two-dimensional recursive digital filters," *IEEE Trans. on Circuits and Systems*, vol. CAS-36, n3, pp. 383-386, Mar. 1989.

[78] H. K. Kwan and Y. C. Lui, "Lattice predictive modelling of 3-D random fields with application to interface predictive coding of picture sequences," *International Journal of Electronics*, vol. 66, no. 4, pp. 489-505, Apr. 1989.

[79] A. Antoniou and W.-S. Lu, *Practical Optimization Algorithms and Engineering Application*s. New York: Springer, 2007 (669 pages). ISBN-13: 978-0-387-71106-5

[80] H. K. Kwan, *Global Optimization Algorithms and Design Applications.* dfisp.org, edition 1.2, Mar. 13, 2018 (403 pages). ISBN-13: 978-1-988-30704-6

[81] J. H. Holland, "Genetic Algorithms," *Scientific American*, vol. 267, no. 1, pp. 66-73, Jul. 1992.

[82] J. Kennedy, "Swarm intelligence," in *Handbook of nature-inspired and innovative computing*, Bureau of Labor Statistics, USA: Springer, 2006, ch.6, pp. 187–219.

[83] C. Blum and D. Merkle, "Swarm intelligence," in *Swarm Intelligence Optimization Blum C Merkle Eds*, Barcelona, Spain: Springer, 2008, ch. 1, pp. 43–85.

[84] R. C. Eberhart, Y. Shi, and J. Kennedy, "The particle swarm," in *Swarm intelligence*, San Francisco, USA: Elsevier, 2001, ch. 7, pp. 287-326.

[85] E. Bonabeau, D. de R. D. F. Marco, M. Dorigo, and G. Theraulaz, "Introduction", in *Swarm intelligence: from natural to artificial systems*, New York, New York, USA: Oxford Univ. Press, 1999, ch. 1, pp. 1-23.

[86] Y. Shi, "Particle swarm optimization: developments, applications and resources," in *Proc. of 2001 congress on evolutionary computation (CEC 2001) (IEEE Cat. No. 01TH8546)*, Seoul, South Korea, vol. 1, 2001, pp. 81–86.

[87] D. Martens, M. De Backer, R. Haesen, J. Vanthienen, M. Snoeck, and B. Baesens, "Classification with ant colony optimization," *IEEE Trans. on Evolutionary Computation*, vol. 11, no. 5, pp. 651–665, Oct. 2007.

[88] R. V. Rao, "Teaching-learning-based optimization algorithm," in *Teaching learning based optimization algorithm*, Suart, India: Springer, 2015, ch. 2, pp. 9–39.

[89] R. V. Rao, V. J. Savsani, and J. Balic, "Teaching–learning-based optimization algorithm for unconstrained and constrained real-parameter optimization problems," *Engineering Optimization*, vol. 44, no. 12, pp. 1447–1462, Mar. 2012.

[90] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems," *Computer-Aided Design*, vol. 43, no. 3, pp. 303–315, Mar. 2011.

[91] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching–learning-based optimization: an optimization method for continuous non-linear large scale problems," *Information Sciences*, vol. 183, no. 1, pp. 1–15, Jan. 2012.

[92] R. Rao and V. Patel, "An elitist teaching-learning-based optimization algorithm for solving complex constrained optimization problems," *International Journal Industrial Engineering Computations*, vol. 3, no. 4, pp. 535–560, Aug. 2012.

[93] R. V. Rao and V. Patel, "An improved teaching-learning-based optimization algorithm for solving unconstrained optimization problems," *Scientia Iranica*, vol. 20, no. 3, pp. 710–720, Jun. 2013.

[94] D. Karaboga, B. Basturk, "On the performance of artificial bee colony (ABC) algorithm," *Applied Soft Computing*, vol. 6, no. 1, pp. 687- 697, Jan. 2008.

[95] D. Karaboga, B. Akay, "A comparative study of artificial bee colony algorithm," *Applied Mathematics and Computation*, vol. 214, no. 1, pp. 108-132, Aug. 2009.

[96] B. Akay, D. Karaboga, "A modified artificial bee colony algorithm for real parameter optimization," *Information Sciences*, vol. 192, pp. 120-142, Jun. 2010,

[97] D. Karaboga, B. Akay, "A modified artificial bee colony (ABC) algorithm for constrained optimization problems," *Applied Soft Computing*, vol. 11, no. 3, pp. 3021-3031, Apr. 2011.

[98] D. Karaboga, B. Gorkemli, C.Ozturk, N. Karaboga, "A comprehensive survey: Artificial bee colony (ABC) algorithm and applications," *Artificial Intelligence Review*, vol. 42, no. 1, pp. 21-57, Jun. 2014

[99] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: Harmony search," *Simulation*, vol. 76, no. 2, pp. 60-68, Feb. 2001.

[100] Z. W. Geem, *Music-Inspired Harmony Search Algorithm: Theory and Applications*, Rockville, Maryland, USA: Springer, 2010.

[101] Z. W. Geem, *Recent Advances in Harmony Search Algorithm*, Rockville, Maryland, USA: Springer, 2010.

[102] X. S. Yang and S. Deb, "Cuckoo search via Levy flights", in *Proc. 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC 2009),* Coimbatore, India, 2009, pp. 210-214.

[103] X. S. Yang, *Nature-Inspired Metaheuristic Algorithms*, Cambridge, UK, Luniver Press, 2nd edition, 2014.

[104] I. F. Jr., X.-S. Yang, D. Fister, and I. Fister, "Cuckoo search: A brief literature review," in *Cuckoo Search and Firefly Algorithm*, vol. 516, Switzerland: Springer, 2014, pp. 49-62.

[105] A. B. Gunji, B. Deepak, C. R. Bahubalendruni, and D. B. B. Biswal, "An optimal robotic assembly sequence planning by assembly subsets detection method using teaching learning-based optimization algorithm," *IEEE Trans. on Automation Science and Engineering*, vol. 15, no. 3, pp. 1369–1385, Jul. 2018.

[106] Y.-H. Cheng, "Estimation of teaching-learning-based optimization primer design using regression analysis for different melting temperature calculations," *IEEE Trans. on Nanobioscience*, vol. 14, no. 1, pp. 3–12, Jan. 2015.

[107] M. Singh, B. K. Panigrahi, and A. R. Abhyankar, “Optimal coordination of directional over-current relays using teaching learning-based optimization (TLBO) algorithm,” *International Journal of Electrical Power & Energy Systems*, vol. 50, pp. 33–41, Sep. 2013.

[108] D. Lei, L. Gao, and Y. Zheng, “A novel teaching-learning-based optimization algorithm for energy-efficient scheduling in hybrid flow shop,” *IEEE Trans. on Engineering Management,* vol. 65, no. 2, pp. 330–340, May. 2018.

[109] W. Shao, D. Pi, and Z. Shao, “A hybrid discrete optimization algorithm based on teaching–probabilistic learning mechanism for no-wait flow shop scheduling,” *Knowledge-Based Systems*, vol. 107, pp. 219–234, Sep. 2016.

[110] J. Li, Q. Pan, and K. Mao, “A discrete teaching-learning-based optimisation algorithm for realistic flowshop rescheduling problems,” *Engineering Applications of Artificial Intelligence*, vol. 37, pp. 279–292, Jan. 2015.

[111] T. Dede, “Application of teaching-learning-based-optimization algorithm for the discrete optimization of truss structures,” *KSCE Journal Civil Engineering*, vol. 18, no. 6, pp. 1759–1767, Sep. 2014.

[112] P. Khazaei, M. Dabbaghjamanesh, A. Kalantarzadeh, and H. Mousavi, “Applying the modified TLBO algorithm to solve the unit commitment problem,” in *Proc. 2016 World Automation Congress (WAC 2016)*, Rio Grande, Puetro Rico, 2016, pp. 1–6.

[113] M. Akhlaghi, F. Emami, and N. Nozhat, “Binary TLBO algorithm assisted for designing plasmonic nano bi-pyramids-based absorption coefficient,” *Journal of Modern Optics*, vol. 61, no. 13, pp. 1092–1096, May. 2014.

[114] Z. Luo, X. He, X. Chen, X. Luo, and X. Li, “Synthesis of thinned concentric circular antenna arrays using modified TLBO algorithm,” *International Journal of Antennas and Propagation*, vol. 2015, Jul. 2015.

[115] Y. Zhile, L. I. Kang, N. I. U. Qun, X. U. E. Yusheng, and A. Foley, “A self-learning TLBO based dynamic economic/environmental dispatch considering multiple plug-

in electric vehicle loads," *Journal of Modern Power Systems and Clean Energy*, vol. 2, no. 4, pp. 298–307, Dec. 2014.

[116] P. R. Krishna and S. Sao, "An improved TLBO algorithm to solve profit based unit commitment problem under deregulated environment," *Procedia Technology*, vol. 25, pp. 652–659, Dec. 2016.

[117] H. Zhang, B. Li, J. Zhang, Y. Qin, X. Feng, and B. Liu, "Parameter estimation of nonlinear chaotic system by improved TLBO strategy," *Soft Computing,* vol. 20, no. 12, pp. 4965–4980, Dec. 2016.

[118] H. Hosseinpour, T. Niknam, and S. I. Taheri, "A modified TLBO algorithm for placement of AVRs considering DGs," in *Proc. 26th International Power System Conference (PSC 26),* Tehran, Iran, 2011, pp. 1–8.

[119] H. Wang and Y. Li, "Hybrid teaching–learning-based PSO for trajectory optimisation," *Electronics Letters*, vol. 53, no. 12, pp. 777–779, Jun. 2017.

[120] T. Cheng, M. Chen, P. J. Fleming, Z. Yang, and S. Gan, "An effective PSO-TLBO algorithm for multi-objective optimization," in *Proc. 2016 IEEE Congress on Evolutionary Computation (CEC 2016)*, Vancouver, BC, CA, 2016, pp. 3977–3982.

[121] M. Ghasemi, M. M. Ghanbarian, S. Ghavidel, S. Rahmani, and E. M. Moghaddam, "Modified teaching learning algorithm and double differential evolution algorithm for optimal reactive power dispatch problem: a comparative study," *Information Sciences*, vol. 278, pp. 231–249, Sep. 2014.

[122] Y.-H. Cheng, C.-N. Kuo, and C.-M. Lai, "Effective natural PCR-RFLP primer design for SNP genotyping using teaching-learning-based optimization with elite strategy," *IEEE Trans. on Nanobioscience*, vol. 15, no. 7, pp. 657–665, Aug. 2016.

[123] Q. Niu, H. Zhang, and K. Li, "An improved TLBO with elite strategy for parameters identification of PEM fuel cell and solar cell models," *International Journal of Hydrogen Energy*, vol. 39, no. 8, pp. 3837–3854, Mar. 2014.

[124] R. V. Rao and V. Patel, “Multi-objective optimization of heat exchangers using a modified teaching-learning-based optimization algorithm,” *Applied Mathematical Modelling*, vol. 37, no. 3, pp. 1147–1162, Feb. 2013.

[125] R. V. Rao and V. Patel, “An improved teaching-learning-based optimization algorithm for solving unconstrained optimization problems,” *Scientia Iranica*, vol. 20, no. 3, pp. 710–720, Jun. 2013.

[126] D. Lei, L. Gao, and Y. Zheng, “A novel teaching-learning-based optimization algorithm for energy-efficient scheduling in hybrid flow shop,” *IEEE Transactions on Engineering Management,* vol. 65, no. 2, pp. 330–340, May. 2018.

[127] C. V. Camp and M. Farshchin, “Design of space trusses using modified teaching–learning based optimization,” *Engineering Structures*, vol. 62, pp. 87–97, Mar. 2014.

[128] L. Wang, F. Zou, X. Hei, D. Yang, D. Chen, and Q. Jiang, “An improved teaching–learning-based optimization with neighborhood search for applications of ANN,” *Neurocomputing*, vol. 143, pp. 231–247, Nov. 2014.

[129] Z. Wang, R. Lu, D. Chen, and F. Zou, “An experience information teaching-learning-based optimization for global optimization,” *IEEE Trans. on Systems, Man, Cybernetics: Systems*, vol. 46, no. 9, pp. 1202–1214, Jan. 2016.

[130] A. Moghadam and A. R. Seifi, “Fuzzy-TLBO optimal reactive power control variables planning for energy loss minimization,” *Energy Conversion and Management*, vol. 77, pp. 208–215, Jan. 2014.

[131] Z. Zhai, S. Li, Y. Liu, and Z. Li, “Teaching-learning-based optimization with a fuzzy grouping learning strategy for global numerical optimization,” *Journal of Intelligent & Fuzzy Systems*, vol. 29, no. 6, pp. 2345–2356, Nov. 2015.

[132] M. Ghasemi, M. Taghizadeh, S. Ghavidel, J. Aghaei, and A. Abbasian, “Solving optimal reactive power dispatch problem using a novel teaching–learning-based optimization algorithm,” *Engineering Applications of Artificial Intelligence*, vol. 39, pp. 100–108, Mar. 2015.

[133] S. Sharma, S. Katiyal, and L. D. Arya, "An improved Teaching-Learning-Based Optimization Algorithm applied to the design of linear phase digital FIR filter," in *Proc.* of *2015 2nd International Conference: Computing for Sustainable Global Development (INDIACom 2015)*, New Delhi, India, 2015, pp. 241–246.

[134] A. Recio-Spinoso, Y.-H. Fan, and M. A. Ruggero, "Basilar-membrane responses to broadband noise modeled using linear filters with rational transfer functions," *IEEE Trans. on Biomedical Engineering*, vol. 58, no. 5, pp. 1456–1465, May. 2011.

[135] Z. M. Smith, B. Delgutte, and A. J. Oxenham, "Chimaeric sounds reveal dichotomies in auditory perception," *Nature*, vol. 416, no. 6876, pp. 87, Mar. 2002.

[136] A. Potamianos and P. Maragos, "A comparison of the energy operator and the Hilbert transform approach to signal and speech demodulation," *Signal Processing*, vol. 37, no. 1, pp. 95–120, May. 1994.

[137] Y.-W. Liu and J. O. Smith III, "Perceptually similar orthogonal sounds and applications to multichannel acoustic echo canceling," in *Proc. 22nd International Conference on Virtual, Synthetic, and Entertainment Audio (AES22)*, Espoo, Finland, 2002, pp. 1-6.

[138] K. Nie, A. Barco, and F.-G. Zeng, "Spectral and temporal cues in cochlear implant speech perception," *Ear and Hearing*, vol. 27, no. 2, pp. 208–217, Apr. 2006.

[139] C. C. Coello and M. S. Lechuga, "MOPSO: A proposal for multiple objective particle swarm optimization," in *Proc. of the 2002 Congress on Evolutionary Computation (CEC 2002) (Cat. No. 02TH8600)*, Honolulu, HI, USA, 2002, vol. 2, pp. 1051–1056.

[140] D. D. Patil and B. D. Dangewar, "Multi-objective particle swarm optimization (MOPSO) based on Pareto dominance approach," *International Journal Computer Applications*, vol. 107, no. 4, Dec. 2014.

[141] C. E. Robles-Rodriguez *et al.*, "Multi-objective particle swarm optimization (MOPSO) of lipid accumulation in Fed-batch cultures," in *Proc. 2016 24th*

Mediterranean Conference on Control and Automation (MED 2016), Athens, Greece, 2016, pp. 979–984.

[142] C. A. C. Coello, G. T. Pulido, and M. S. Lechuga, “Handling multiple objectives with particle swarm optimization,” *IEEE Trans. on Evolutionary Computation*, vol. 8, no. 3, pp. 256–279, Jun. 2004.

[143] I. Kollar, R. Pintelon, and J. Schoukens, “Optimal FIR and IIR Hilbert transformer design via LS and minimax fitting,” IEEE Trans. on Instrumentation and Measurement, vol. 39, no. 6, pp. 847–852, Dec. 1990.

[144] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE Trans. on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[145] W. Lin *et al.*, “A multi-objective teaching−learning-based optimization algorithm to scheduling in turning processes for minimizing makespan and carbon footprint,” *Journal of Cleaner Production*, vol. 101, pp. 337–347, Aug. 2015.

[146] T. Niknam, R. Azizipanah-Abarghooee, and M. R. Narimani, “A new multiobjective optimization approach based on TLBO for location of automatic voltage regulators in distribution systems,” *Engineering Applications of Artificial Intelligence,* vol. 25, no. 8, pp. 1577–1588, Dec. 2012.

[147] D. Chen, F. Zou, R. Lu, L. Yu, Z. Li, and J. Wang, “Multi-objective optimization of community detection using discrete teaching–learning-based optimization with decomposition,” *Information Sciences*, vol. 369, pp. 402–418, Nov. 2016.

[148] H. K. Kwan and Y. C. Lui, “Normalized ARMA Levinson algorithm,” in *Proc. of IEEE International Symp. on Circuits and Systems (ISCAS 1989)*, Portland, Oregon, U.S.A., May 9-11, 1989, vol. 1, pp. 234-237.

[149] H. K. Kwan, “ARMA lattice digital filter design,” in *Proc. of 45th IEEE International Midwest Symp. on Circuits and Systems (MWSCAS2002)*, Tulsa, Oklahoma, U.S.A., vol. III, Aug. 4-7, 2002, pp. III-489 to III-492.

[150] H. K. Kwan, "Minimal ARMA lattice digital filter realization," in *Proc. of IEEE International Symp. on Circuits and Systems (ISCAS 2003)*, Bangkok, Thailand, May 25-28, 2003, pp. IV-305 to IV-308.

[151] H. K. Kwan and Y. C. Lui, "Normalized 2-D ARMA lattice parameter modeling algorithm," in *Proc. of IEEE Pacific Rim Conf. on Communications, Computers and Signal Processing (PACRIM 1989)*, Victoria, B.C., Canada, Jun. 1-2, 1989, pp. 372-375.

[152] H. K. Kwan, "Tunable bandpass passive digital filter design," in *Proc. of 45th IEEE International Midwest Symp. on Circuits and Systems (MWSCAS 2002)*, Tulsa, Oklahoma, U.S.A., vol. III, Aug. 4-7, 2002, pp. III-481 to III-484.

[153] H. K. Kwan, "Multi-output passive digital filters," in *Proc. of IEEE International Symp. on Circuits and Systems (ISCAS 2005)*, Kobe, Japan, May 23-26, 2005, vol. 3, pp. 2595-2598.

[154] H. K. Kwan, "Variable multi-output passive digital filters," in *Proc. of IEEE International Symp. on Circuits and Systems (ISCAS 2005)*, Kobe, Japan, May 23-26, 2005, pp. 3725-3728.

[155] H. K. Kwan and M. T. Tsim, "High speed 1-D FIR digital filtering architectures using polynomial convolution," in *Proc. of IEEE International Conf. on Acoustics, Speech, and Signal Processing (ICASSP 1987)*, Dallas, Texa, U.S.A., 6-9 Apr. 1987, pp. 1863-1866.

[156] H. K. Kwan, "High speed delayed 2-path 1-D linear phase FIR digital filtering architectures," in *Proc. of International Symp. on Electronic Devices, Circuits and Systems (ISEDCS 1987)*, Kharagpur, India, Dec. 16-18, 1987, pp. 876-878.

[157] H. K. Kwan, "Systolic realization of delayed 2-path 1-D linear phase FIR digital filters," in *Proc. of International Conf. on Circuits and Systems (ICCAS 1989)*, Nanjing, China, Jul. 6-8, 1989.

[158] H. K. Kwan, "New systolic realisation of recursive digital filters for high speed applications," *Electronics Letters*, vol. 28, no. 12, pp. 1087-1089, Jun. 4, 1992.

[159] H. K. Kwan, "Novel systolic array realisation of allpass digital filters for high speed applications," *Electronics Letters*, vol. 28, no. 15, pp. 1408-1409, Jul. 16, 1992.

[160] H. K. Kwan and K. Hirano, "Delayed N-path structure for high speed adaptive FIR digital filtering," *Electronics Letters*, vol. 28, no. 20, pp. 1880-1882, Sep. 24, 1992.

[161] H. K. Kwan, "Improved systolic allpass digital filters for very high speed applications," *Electronics Letters*, vol. 28, no. 22, pp. 2061-2062, Oct. 22, 1992.

[162] H. K. Kwan and Q. P. Li, "High-speed realisation of adaptive linear phase FIR digital filters," *IEE Proc. Part F on Radar and Signal Processing*, vol. 140, no. 1, pp. 48-54, Feb. 1993.

[163] H. K. Kwan, "Systolic realisation of delayed two-path linear phase FIR digital filters," *IEE Proc. Part G on Circuits, Devices and Systems*, vol. 140, no. 1, pp. 75-80, Feb. 1993.

[164] H. K. Kwan, "New form of delayed N-path recursive digital filters," *Electronics Letters*, vol. 28, no. 9, pp. 736-738, Apr. 29, 1993.

[165] H. K. Kwan and K. Hirano, "High speed 2-D FIR digital filtering using similarity transformation," in *Proc. of IEEE International Symp. on Circuits and Systems (ISCAS 1986)*, San Jose, California, U.S.A., May 5-7, 1986, vol. 2, pp. 523-526.

[166] H. K. Kwan and K. Hirano, "High speed delayed multipath two-dimensional digital filtering architecture," *IEEE Trans. on Circuits and Systems*, vol. CAS-34, no. 2, pp. 190-196, Feb. 1987.

[167] H. K. Kwan and K. Y. Wong, "Implementation of fast 2-D digital filters with multiple processors," in *Proc. of IEEE Pacific Rim Conference on*

Communications, Computers and Signal Processing (PACRIM 1987), Victoria, B.C., Canada, Jun. 4-5, 1987, pp. 447-450.

[168] H. K. Kwan, "High speed delayed 2-path realization of 2-D linear phase FIR digital filters," in *Proc. of International Conference on Circuits and Systems (ICCAS 1989)*, Nanjing, China, Jul. 6-8, 1989, pp. 1-4.

[169] H. K. Kwan and M. T. Tsim, "Efficient systolic high speed architectures for delayed multipath two-dimensional FIR and IIR digital filtering*," IEE Proc. Part G on Circuits, Devices and Systems*, vol. 137, no. 6, pp. 413-423, Dec. 1990.

[170] H. K. Kwan, "New systolic array for realizing second-order recursive digital filters," *Electronics Letters*, vol. 23, no. 9, pp. 442-443, Apr. 23, 1987.

[171] H. K. Kwan, "Systolic array implementation of linear phase FIR digital filters," in *Proc. of IEEE Asian Electronics Conference (AEC 1987)*, Hong Kong, Sep. 1-6, 1987, pp. 111-115.

[172] H. K. Kwan, "Systolic realization of linear phase FIR digital filters," *IEEE Trans. on Circuits and Systems*, vol. CAS-34, no. 12, pp. 1604-1605, Dec. 1987.

[173] H. K. Kwan and T. S. Okullo-Oballa, "Systolic array implementation of a decimator and an interpolator," *IEE Proc. Part E on Computers and Digital Techniques*, vol. 135, no. 1, pp. 70-72, Jan. 1988.

[174] H. K. Kwan, "Systolic array implementation of H(z^-1)=(a0+a1z^-N+a2z^-2N)/(1+b1z^-N+b2z^-2N) for N>=2," *IEEE Trans. on Circuits and Systems*, vol. CAS-35, no. 1, pp. 134, Jan. 1988.

[175] H. K. Kwan and T. S. Okullo-Oballa, "Systolic array implementation of IIR decimators and interpolators," *Hong Kong Engineer*, vol. 18, no. 12, pp. 27-29, Dec. 1990.

[176] H. K. Kwan, "Realisation of high-speed systolic IIR decimators and interpolators," *Electronics Letters*, vol. 29, no. 20, pp. 1748-1749, Sep. 30, 1993.

[177] H. K. Kwan, “Systolic array implementation of 2-D linear phase FIR digital filters,” in *Proc. of IEEE International Symp. on Circuits and Systems (ISCAS 1988)*, Espoo, Finland, Jun. 6-9, 1988, vol. 3, pp. 2885-2888.

[178] H. K. Kwan and T. S. Okullo-Oballa, “Systolic realization of 2-D linear phase FIR decimators and interpolators,” in *Proc. of IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM 1989)*, Victoria, B.C., Canada, Jun. 1-2, 1989, pp. 376-379.

[179] H. K. Kwan and T. S. Okullo-Oballa, “2-D systolic arrays for realization of 2-D convolution,” *IEEE Trans. on Circuits and Systems*, vol. CAS-37, no. 2, pp. 267-273, Feb. 1990.

[180] H. K. Kwan, “Systolic and parallel realization of 2-D IIR digital filters,” in *Proc. of IEEE International Symp. on Circuits and Systems (ISCAS 1990)*, New Orleans, Louisiana, U.S.A., May 1-3, 1990, vol. 3, pp. 2345-2348.

[181] T. Baran, D. Wei, and A.V. Oppenheim, “Linear programming algorithms for sparse filter design,” *IEEE Trans. on Signal Processing*, vol. 58, no. 3, pp. 1605-1617, Mar. 2010.

[182] A. Jiang, H. K. Kwan, and Y. Zhu, “Peak-error-constrained sparse FIR filter design using iterative SOCP,” *IEEE Trans. on Signal Processing*, vol. 60, no. 8, pp. 4035-4044, Aug. 2012.

[183] A. Jiang and H. K. Kwan, “WLS design of sparse FIR digital filters,” *IEEE Trans. on Circuits and Systems I: Regular Papers*, vol. 60, no. 1, pp. 125-135, Jan. 2013.

[184] D. Wei, C.K. Sestok, and A.V. Oppenheim, “Sparse filter design under a quadratic constraint: Low-complexity algorithms,” *IEEE Trans. on Signal Processing*, vol. 61, no. 4, pp. 857-870, Feb. 2013.

[185] D. Wei and A.V. Oppenheim, “A branch-and-bound algorithm for quadratically-constrained sparse filter design,” *IEEE Trans. on Signal Processing*, vol. 61, no. 4, pp. 1006-1018, Feb. 2013.

[186] A. Jiang, H. K. Kwan, Y. Zhu, X. Liu, N. Xu, and Y. Tang, “Design of sparse FIR filters with joint optimization of sparsity and filter order,” *IEEE Trans. on Circuits and Systems I: Regular Papers*, vol. 62, no. 1, pp. 195-204, Jan. 2015.

[187] A. Jiang and H. K. Kwan, “Recent Advances in Sparse FIR Filter Design using l0 and l1 Optimization Techniques,” in *Trends in Digital Signal Processing - A Festschrift in Honour of A. G. Constantinides* (Pan Stanford Series on Digital Signal Processing - Vol. 1), Y. C. Lim, H. K. Kwan, W.-C. Siu, Eds. Singapore: Pan Stanford, Jul. 2015, ch. 5, pp. 145-176.

[188] W. Ye and Y. J. Yu, “Greedy algorithm for the design of linear-phase FIR filters with sparse coefficients,” *Circuits, Systems, and Signal Processing*, vol. 35, issue 4, pp. 1427–1436, Apr. 2016.

[189] H. K. Kwan, J. Liang, and A. Jiang, “Sparse FIR filter design using iterative MOCSA,” in *Proc. of IEEE 61st International Midwest Symp. on Circuits and Systems (MWSCAS 2018)*, Windsor, Ontario, Canada, Aug. 2018, pp. 952-955

[190] R. Raju, H. K. Kwan, and A. Jiang, “Sparse FIR filter design using artificial bee colony algorithm,” in *Proc. of IEEE 61st International Midwest Symp. on Circuits and Systems (MWSCAS 2018)*, Windsor, Ontario, Canada, Aug. 2018, pp. 956-959.

[191] Z.G. Feng, K.F.C. Yiu, and S.Y. Wu, “Design of sparse filters by a discrete filled function technique,” *Circuits, Systems, and Signal Processing*, vol. 37, issue 10, pp. 4279–4294, Oct. 2018.

[192] J. Liang, H. K. Kwan, and A. Jiang, “Sparse linear phase FIR filter design using iterative CSA,” in *Proc. of 2018 IEEE 23rd International Conf. on Digital Signal Processing (DSP 2018)*, Shanghai, China, Nov. 2018, pp. 1-4.

[193] W. Chen, M. Huang, and X. Lou, “Design of sparse FIR filters with reduced effective length,” *IEEE Trans. on Circuits and Systems I: Regular Papers*, vol. 66, no. 4, pp. 1496-1506, Apr. 2019.

[194] A. Jiang, H. K. Kwan, Y. Tang, Y. Zhu, “Sparse FIR filter design via partial 1-norm optimization,” *IEEE Trans. on Circuits and Systems II: Express Briefs*, vol. 66, 2019.

[195] W.-S. Lu and T. Hinamoto, “Two-dimensional digital filters with sparse coefficients,” *Multidimensional Systems and Signal Processing*, vol. 22, no. 1-3, pp 173-189, Mar. 2011.

[196] C. Rusu and B. Dumitrescu, “Iterative reweighted l1 design of sparse FIR filters,” *Signal Processing*, vol. 92, issue 4, pp. 905–911, Apr. 2012.

[197] H. Wang, “A new separable two-dimensional finite impulse response filter design with sparse coefficients,” *IEEE Trans. on Circuits and Systems I: Regular Papers*, vol. 62, no. 12, pp. 2864-2873, Dec. 2015.

[198] J. Liang and H. K. Kwan, “Sparse 2-D FIR filter design using iterative CSA with step-descendant coefficient thresholding,” in *Proc. of 2018 IEEE 23rd International Conf. on Digital Signal Processing (DSP 2018)*, Shanghai, China, Nov. 2018, pp. 1-4.

[199] H. K. Kwan, “Linear phase equiripple IIR digital filter design,” in *Proc. of IEEE International Conf. on Acoustics, Speech and Signal Processing (ICASSP 2003)*, Hong Kong, China, vol. VI, Apr. 6-10, 2003, pp. VI-17 to VI-20.

[200] H. Zhao, H. K. Kwan, Li Wan, and L. Nie, “Design of 1-D stable variable fractional delay IIR filters using finite impulse response fitting,” in *Proc. of International Conf. on Communications, Circuits and Systems (ICCCAS 2006)*, Guilin, China, Jun. 25-28, 2006, pp. 201-205.

[201] H. Zhao and H. K. Kwan, “Design of 1-D stable variable fractional delay IIR filters,” *IEEE Trans. on Circuits and Systems II: Express Briefs*, vol. 54, no. 1, pp. 86-90, Jan. 2007.

[202] K. M. Tsui, S. C. Chan, and H. K. Kwan, “A new method for designing causal stable IIR variable fractional delay digital filters,” *IEEE Trans. on Circuits and Systems II: Express Briefs*, vol. 54, no. 11, pp. 999-1003, Nov. 2007.

[203] H. K. Kwan and A. Jiang, “FIR, allpass, IIR variable fractional delay digital filter design,” *IEEE Trans. on Circuits and Systems I: Regular Papers*, vol. 56, no. 9, pp. 2064-2074, Sep. 2009.

[204] H. K. Kwan and A. Jiang, “Integrated design of IIR variable fractional delay digital filters with variable and fixed denominators,” in *Digital Filters*, Fausto Pedro García Márquez, Ed., London: InTech, Apr. 2011, ch. 8, pp. 179-208. ISBN-13: 978-953-307190-9

[205] S.-C. Pei, J.-J. Shyu, Y.-D. Huang, and C.-H. Chan, “Improved methods for the design of variable fractional-delay IIR digital filters,” *IEEE Trans. on Circuits and Systems I: Regular Papers*, vol. 59, no. 5, pp. 989-1000, May 2012.

[206] W. E. Boyce, R. C. DiPrima, and D. B. Meade, *Elementary Differential Equations and Boundary Value Problems, Edition 11*, John Wiley & Sons, 2017.

[207] R. V. Rao, *Teaching Learning Based Optimization and its Applications*, Springer International Publishing, Switzerland, 2016.

VITA AUCTORIS

NAME:	Miao Zhang
PLACE OF BIRTH:	Baoding, Hebei, China
YEAR OF BIRTH:	1989
EDUCATION:	Baoding No.2 High School, Baoding, Hebei, China, 2008
	Hebei University, B.Sc., Baoding, Hebei, China, 2012
	Hebei University, M.Sc., Baoding, Hebei, China, 2015