الجمهورية الجزائرية الديمقراطية الشعبية
Democratic And Popular Republic of Algeria
وزارة التعليم العالي و البحث العلمي
Ministry of Higher Education And Scientific Research

Ziane Achour University of Djelfa
Faculty of Exact and Computer Sciences
Departement of Mathematics and Computer Science

جامعة زيان عاشور بالجلفة
كلية العلوم الدقيقة والإعلام الآلي
قسم الرياضيات والإعلام الآلي

Doctoral thesis

Domain: Mathematics & Computer Sciences

Field: Computer Science

Option: Mathematical Optimization for Signal Processing and Communication Networks

By:

Mr. Youcef GHIBECHE

Optimal Design of Digital Filters

Defended on: **February 05th, 2019**

Jury Members:

Mr. Farid MESSELMI	Professor	Univ of Djelfa	President
Mr. Slami SAADI	MC-A	Univ of Djelfa	Supervisor
Mr. Ahmed HAFAIFA	Professor	Univ of Djelfa	Co-Supervisor
Mr. Amer AISSA	Professor	Univ of Laghouat (UATL)	Examiner
Mr. Abdellah KOUZOU	Professor	Univ of Djelfa	Examiner
Mr. Abdelhalim MAYOUF	Professor	Univ of Djelfa	Examiner

Academic Year: **2018-2019**

DEDICATION

To my mother and my father, who have always loved me unconditionally and who have taught me to work hard for the things that i aspire to achieve.

To my dear brothers.

To all members of my family.

To my friends.

And to all who have contributed one day to my education and training.

ACKNOWLEDGEMENT

First and foremost, I would like to thank Allah Almighty who gave us the strength, courage and patience to carry out this thesis work.

I would like to sincerely thank my supervisor, ***Dr. SAADI Slami****, for doing everything from the inception of the project idea to the last step in it, which enabled me to improve and get an understanding on the subject. Without his invaluable advices, this work would not have been possible.*

Also, I am heartily thankful to ***Pr. HAFAIFA Ahmed,*** *for his encouragement, supervision and support.*

I would like to thank the members of the jury who do me the honor of participating in the defense:

- *Pr. Farid MESSELMI*
- *Pr. Amer AISSA*
- *Pr. Abdelhalim MAYOUF*
- *Pr. Abdellah KOUZOU*

Finally, thanks to all those who contributed, from near or far, to the realization of this work.

TABLE OF CONTENTS

LISTE OF FIGURES

LIST OF TABLES

LISTS OF ACRONYMS AND SYMBOLS

Acronyms

AP	Awareness Probability
ASIC	Application Specific Integrated Circuits
BFGS	Broyden-Fletcher-Goldfarb-Shanno
BLT	Bilinear Transform
CSA	Crow Search Algorithm
DFP	Davidon-Fletcher-Powell
DSP	Digital Signal Processing
ECG	Electrocardiogram
FIR	Finite Impulse Response
FL	Flight Length
FPGA	Field-Programmable Gate Array
GA	Genetic Algorithm
IIR	Infinite Impulse Response
LHP	Left Half Plane
PLD	Programmable Logic Device
PSO	Particle Swarm Optimization
RF	Radio Frequency
VHDL	VHSIC Hardware Description Language
VHSIC	Very High Speed Integrated Circuit

Symbols

a_K and b_K	the filter coefficients
ϵ	ripple factor
δ_p	related to passband ripple
δ_s	related to stopband ripple
α	the selectivity factor
δ	the change in vector x
$o\left(\|\delta\|_2^2\right)$	the reminder
γ	the change in the gradient
iter	Iteration

INTRODUCTION

1. Introduction

Digital signal processing is the numerical manipulation of signals that have been already digitized, so that the information that they contain could be analyzed, filtered or converted, using computers or more specified, a digital signal processor, and then feeds them back to use in the real world. In DSP, one of the most powerful tool and essential components are the digital filters and they used in different applications such as automatic control, telecommunications, speech processing, and other areas [1].

Depending on the form of impulse response of the system, digital filters can be classified into two general categories which are infinite impulse response (IIR) filters and finite impulse response (FIR) filters. The major significant advantage of IIR is that it gives much superior performance compared with the finite impulse response (FIR) filter, and it fits the desired filter specifications with a less filter order [2].

In general, the design of IIR filter design follows two principally methods [3]: transformation technique and optimization technique. In the classical methods for IIR filter design, the lowest order and the coefficients of the filter are selected for a standard prototype low pass Butterworth, Chebyshev Type-I, Chebyshev Type-II, and Elliptic filters which are converted to digital low pass, high pass, band pass, and band stop IIR digital filter employing various transformation techniques such as bilinear transformation etc..[4].

Many optimization approaches founded depending on novel modern heuristics optimization algorithms. Numerous papers highlighted the advantages of using these modern heuristic algorithms for the designing of IIR filters. In [5] an explanation of the most significant features of using a simulated annealing algorithm in designing digital filters with a linear phase; after that, the simulated annealing algorithm was used to design a finite impulse response filter (FIR), and the result was not notable. Moreover, it requires a huge time of computations. Another global optimization method known as the adaptive simulated annealing was applied to digital IIR filter design in [6]. The genetic algorithm (GA) has gained great interest in the field of designing digital IIR filter [7-10]. Many works in the literature listed the applying of GA in digital filter design, in [11] GA used to design a 1-D IIR filter with canonical-signed-digit coefficients constrained to a low-pass filter; while others investigated the use of GAs for the design of numerous kinds of digital filters [12], it was found that GAs require a large amount of computation. More different works based on GA are reported in the literatures. These are orthogonal genetic algorithm [12], hybrid Taguchi GA [13], hybrid genetic algorithm [14], and real coded genetic algorithm [15]. Another class of optimization that has been successfully applied to the IIR filter designs called the particle swarm optimization (PSO). The application of PSO algorithm to IIR adaptive filter structures was introduced in [16]. A quantum-behaved particle swarm optimization algorithm was employed to IIR filter design [17], while [18] introduced the applying of PSO method to the designing of two-dimensional IIR filters, where [19] succeeded in using the PSO algorithm to design digital IIR filters in a realistic time domain condition where the desired filter output is corrupted by noise, while a novel hybrid particle swarm optimization and gravitational search algorithm (GSA) proposed for IIR filter design in [20]. The artificial bee colony (ABC) [21,22] was recommended for a variety of benchmark adaptive IIR representations in 2009 and 2011, respectively. The seek optimization algorithm, which imitates human search character, was suggested to resolve five IIR system identifications and was compared with GA and PSO [23]. The cat swarm optimization, opposition-based bat algorithm, and artificial immune systems were employed for IIR model identification [24-26]. The results show better identification performance of the three algorithms compared with those obtained through PSO and GA. Another swarm intelligence algorithm, differential evolution, and its alternative with wavelet mutation were used in designing both IIR and FIR digital filters [27,28]. Others proposed the GSA for linear IIR filter and nonlinear rational filter modeling [29]; a combined version of the GSA and wavelet mutation was presented in [30], and it used for designing IIR digital filter. In [31], a different algorithm called the ant colony optimization (ACO) algorithm with global optimization capability was suggested for designing digital IIR filter. Though, ACO has a tendency to local minima in complex problem, and its convergence rapidity is also delayed [32]. In [33,34], for the digital adaptive IIR filters design, a technique based on the tabu search algorithm was suggested. An additional method imitates the flash pattern and characteristics of fireflies algorithm used for adaptive IIR filter to find its optimal coefficients, it accomplished that is excellent and sufficient to handle indefinite system identification problem [35]. These swarm-based methods prove very good performance in IIR system identification and complex optimization problem.

To realize the digital filters for purposes that necessitate high throughput and constrained power dissipation, a devoted hardware is necessary. The efficiency of the hardware design is basing on the structure selected for digital filters implementation [36]. The implementation of digital IIR filters can be done through the use of programmable digital processors. Nevertheless, in realizing a high order filter several complicated computations are required which regularly influence the universal performance of the familiar digital signal processors from where the rapidity, cost suppleness, stability, etc[37]. To deal with all these difficulties, designers count more on developed digital electronics technologies that arrive with friendly software development tools. Depending on this improvement, the Field-Programmable Gate Array (FPGA) turn into an exceptionally cost-effective means of realizing computationally exhaustive digital signal processing algorithms to develop global system performance [37]. During the design and after shipping like maintenance, FPGA be able to offer a large suppleness and reliability because its re-configurable logic elements [38]. Moreover, recent FPGAs can provide high speed processing, thus, high speed IIR realization since they contain a parallel structure [39,40].

Based on its importance for achieving the best optimal filters design, the comparison needs to be made between methods. For this, in this thesis, for the purpose of looking for more efficient filters design techniques, a novel optimization algorithm called Crow Search Algorithm (CSA) in a combination with ℓp-norm is introduced for digital IIR filter design compared with the classical butter classical designs using butter synthesis method. Also, this thesis highlights the optimal implementation of high-pass and band-stop IIR filters using the Crow Search Algorithm, basing on MATLAB Simulink model and Xilinx system generator blocks, and their performance is studied for noise removal from ECG signals.

2. Thesis motivation and objective

Digital signal processing is now pervasive as it is used in everything from digital photo cameras, MP3 players, digital television, mobile phones to automobiles and advanced medical imaging equipment. In DSP, the function of the digital filter is to remove unwanted parts of the signal, such as random noise, or to extract useful parts of the signal, such as the components lying within a certain frequency rang. Due to this importance of the digital filter, it is very important to design a filter that can provide a much better performance.

In the field of signal processing there is a significant need of a special class of digital filters known as infinite impulse response or IIR filters. IIR filters are used in a great many areas of Digital Signal Processing (DSP) and telecommunication systems for different applications. These filters have various advantages over the other types of digital filters. The fundamental one is that the IIR filter can give a very good performance than the FIR filter does, and they can satisfy the given filter specifications with a much lower filter order. In order to design a filter prior knowledge of the desired response is required.

This thesis considers only the design of IIR filters, in which the design requirements on magnitude and phase responses are both considered. In general the design of digital filter design involves 1) specification of a desired frequency response, 2) actual design which includes converting the given specification into a suitable filter structure, and 3) the implementation which involves the producing the resulting design. The first and third steps are more reliant on the particular application, which is in our case as an example, a noise removal from ECG signals.

The aim of this thesis is to apply novel meta-heuristic algorithms for designing an IIR filter. A low pass and high pass digital IIR filters are considered as examples to examine the performance of the new design method, and to compare the performance of the proposed methods, a comparison was made with the Butterworth method. And for the implementation filters, the performance of the considered two types of filters has been experienced for noise elimination from ECG signal to verify the applicability of this optimal approach.

modest

3. Organization

This thesis is organized in four chapters and a general introduction and conclusion. An introduction on the research developed here is given as a general introduction. Chapter 1 presents a review of the basis of the digital filters, and introduces an overview of the FPGA and techniques used for the implementation. Chapter 2 presents a brief on the classical approach for IIR filter design. The proposed method that employ for designing the IIR digital filters are detailed in Chapter 3. In Chapter 4, presents the results of applying the proposed method in the previous chapter for a low-pass and high-pass filters, and the FPGA implementation is carried out for different filter specifications. Finally, we end this thesis by a general conclusions and suggestions for future research.

References

[1] Cousseau JE, Werner S, Donate PD. Factorized all-pass based IIR adaptive notch filters. IEEE Trans Signal Process. 2007;55(11):5225-5236.

[2] Oppenheim AV, Schafer RW, Buck JR. Discrete-time Signal Processing. Second ed. Prentice Hall; 1999.

[3] Kaur R, Patterh MS, Dhillon JS. Design of optimal L1 stable IIR digital filter using hybrid optimization algorithm. Int J Comput Appl (0975 – 8887). 2012;38(2, January).

[4] Saha SK, Kar R, Mandal D, Ghoshal SP. Digital stable IIR low pass filter optimization using PSO-CFIWA. In: 2012 1st International Conference on Recent Advances in Information Technology (RAIT); 2012:196-201.

[5] Benvenuto N, Marchesi M. Applications of simulated annealing for the design of digital filters. IEEE Trans Signal Process. 1992;40(2):323-331.

[6] Chen S, Istepanian RH, Luk BL. Digital IIR filter design using adaptive simulated annealing. Digital Signal Process. 2001;11(3):241-251.

[7] Nambiar R, Tang CKK, Mars P. Genetic and learning automata algorithms for adaptive digital filters. Proc ICASSP 1992. 1992;4:41-44.

[8] Wilson PB, Macleod MD. Low implementation cost IIR digital filter design using genetic algorithms. Workshop on Natural Algorithms in Signal Processing, Chelmsford, Essex, UK; 1993:4/1–4/8.

[9] White MS, Flockton SJ. Genetic algorithms for digital signal processing. Lect Notes Comput Sci. 1994;865:291-303.

[10] Ng SC, Leung SH, Chung CY, Luk A, Lau WH. The genetic search approach: a new learning algorithm for adaptive IIR filtering. IEEE Signal Process Mag, November Issue. 1996;13(6):38-46.

[11] Liang L, Ahmadi M, Ahmed M, Wallus K. Design of canonical signed digital filters using genetic algorithms. IEEE Trans Signal Process. 2003;3(1):2043-2047.

[12] Ahmad SU, Antoniou A. Design of digital filters using genetic algorithms. IEEE Trans Signal Process. 2006;1(1):1-9.

[13] Tang W, Shen T. Optimal design of FRM- based FIR filters by using hybrid Taguchi genetic algorithm. Int Conf Green Circuits Syst (ICGCS-2010). 2010;392-397.

[14] Ahmad SU, Antoniou A. A hybrid genetic algorithm for the design of IIR digital filters. 2007 Canadian conference on electrical and computer engineering, vol. 2012, no. August; 2007:286-292.

[15] Kaur R, Patterh MS, Dhillon JS. Digital IIR filter design using real coded genetic algorithm. Int J Inf Technol Comput Sci (IJITCS). 2013;5(7):27-35. https://doi.org/10.5815/ijitcs.2013.07.03

[16] Krusienski DJ, Jenkins WK. Particle swarm optimization for adaptive IIR filter structures. Proc Congr Evol Comput, Jun. 2004;965-970.

[17] Fang W, Sun J, Xu W-B. Design IIR digital filters using quantum-behaved particle swarm optimization. Proc. 2nd Int. Conf. Natural Computation, Xian, China, 24–28 September, Part II. 2006;637-640.

[18] Das S, Konar A. A swarm intelligence approach to the synthesis of two-dimensional IIR filters. Eng Appl Artif Intel. 2007;20(8):1086-1096.

[19] Chen S, Luk BL. Digital IIR filter design using particle swarm optimisation. Int J Model Identif Control. 2010;9(4):327-335.

[20] Jiang S, Wang Y, Ji Z. A new design method for adaptive IIR system identification using hybrid particle swarm optimization and gravitational search algorithm. Nonlinear Dyn. 2015;79(4):2553-2576.

[21] Karaboga N. A new design method based on artificial bee colony algorithm for digital IIR filters. JFrankl Inst. 2009;346(4):328-348.

[22] Karaboga N, Cetinkaya MB. A novel and efficient algorithm for adaptive filtering: artificial bee colony algorithm. Turk J Electr Eng Comput Sci. 2011;19(1):175-190.

[23] Dai C, Chen W, Zhu Y. Seeker optimization algorithm for digital IIR filter design. IEEE Trans Ind Electron. 2010;57(5):1710-1718.

[24] Panda G, Pradhan PM, Majhi B. IIR system identification using cat swarm optimization. Expert Syst Appl. 2011;38(10):12671-12683.

[25] Saha SK, Kar R, Mandal D, Ghoshal SP. A new design method using opposition-based BAT algorithm for IIR system identification problem. Int J Bio-Inspired Comput. 2013;5(2):99-132.

[26] Kalinli A, Karaboga N. Artificial immune algorithm for IIR filter design. Eng. Appl. Artif.Intell. 2005;18(8):919-929.

[27] Karaboga N. Digital IIR filter design using differential evolution algorithm. EURASIP J Appl Signal Process. 2005;8:1269-1276.

[28] Mandal S, Ghoshal SP, Kar R, Mandal D. Differential evolution with wavelet mutation in digital FIR filter design. J Optim Theory Appl. 2012;155(1):315-324.

[29] Rashedi E, Nezamabadi-Pour H, Saryazdi S. Filter modeling using gravitational search algorithm. Eng Appl ArtifIntell. 2011;24(1):117-122.

[30] Saha S, Kar R, Mandal D, Ghoshal S. Optimal IIR filter design using gravitational search algorithm with wavelet mutation. J King Saud Univ—Comput Inform Sci. 2015;27(1):25-39.

[31] Karaboga N, Kalinli A, Karaboga D. Designing IIR filters using ant colony optimization algorithm. J Eng Appl ArtifIntell. Apr. 2004;17(3):301-309.

[32] Hlaing ZCSS, Khine MA. Solving traveling salesman problem by using improved ant colony optimization algorithm. Int J Inf Educ Technol. 2011;1(5):404-409.

[33] Kalinli A, Karaboga N. A new method for adaptive IIR filter design based on Tabu search algorithm. Int J ElectronCommun(AEÜ). 2005;59(2):111-117.

[34] Kalinli A, Karaboga N. New method for adaptive IIR filter design based on tabu search algorithm," to appear in. AEU Int J Electron Commun. 2004.

[35] Upadhyay P, Kar R, Mandal D, Ghoshal SP. A new design method based on firefly algorithm for IIR system identification problem. J King Saud Univ Eng Sci. 2014. https://doi.org/10.1016/j.jksues.2014.03.0015

[36] Subhankar Bhattacharjeea , Sanjib Silb , Amlan Chakrabartic, "Evaluation of Power Efficient FIR Filter for FPGA Based DSP Applications", International Conference on Computational Intelligence: Modeling Techniques and Applications (CIMTA) 2013. Procedia Technology 10 (2013) 856 – 865.

[37] Emmanuel S. Kolawole, WarsameH.Ali, Penrose Cofie, John Fuller, C. Tolliver, Pamela Obiomon, Design and Implementation of Low Pass, High-Pass and Band-Pass Finite Impulse Response (FIR) Filters Using FPGA Circuits and Systems, 2015, 6, 30-48.

[38] H. S. O. Migdadi and R. A. Abd-alhameed, "FIR implementation on FPGA: Investigate the FIR order on SDA and PDA algorithms", Journal of Internet Technologies and Applications (ITA), 2015, pages 417-421.

[39] Mirzaei, S.; Hosangadi, A.; Kastner, R., "FPGA Implementation of High Speed FIR Filters Using Add and Shif Method,". International Conference on Computer Design. 2006. ICCD 2006, vol., no., pp.308,313,1-4 Oct. 2007.

[40] Ali M. AI-Haj , "Fast Discrete Wavelet Transformation Using FPGAs and Distributed Arithmetic", International Journal of Applied Science and Engineering, 2003.

Chapter 1

DIGITAL FILTERS BASICS

1. Introduction

In electronic systems, Filters usually are utilized to emphasize signals in specific frequency ranges and attenuate signals in other frequency ranges. Such a filter has a gain which is dependent on signal frequency. For example, consider a case where a desired signal at frequency f1 has been incorporated with an undesirable signal at f2. If the incorporated signal is passed through a circuit (Figure1.1) that has very small gain at f2 relative to f1, the undesirable signal will be rejected, while the desired signal can be preserved. Note that our object is to eliminate the undesirable signal, so, as long as the elimination of the signal at f2 is done sufficiently compared to preserving the useful signal, we are not interested with the gain of the filter at any frequency other than f1 and f2; and we can say that the performance of this filter will be acceptable.

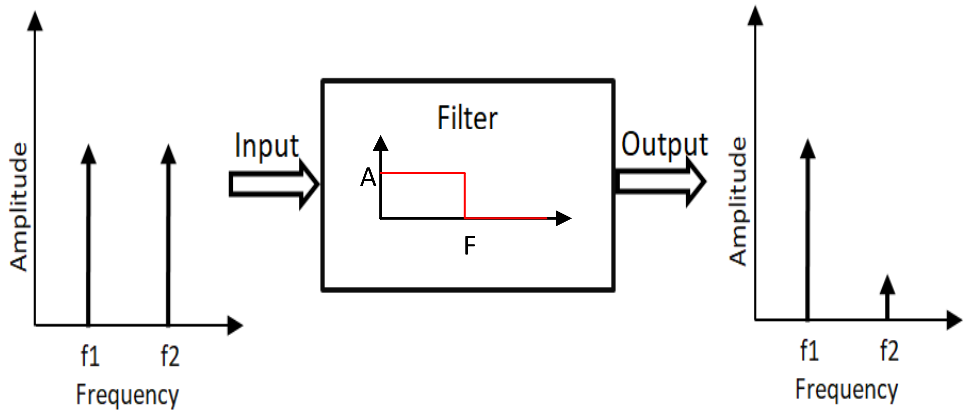


Fig. 1.1 Example of the functioning of a filter

2. Classification of filters

The elimination of undesirable parts of the signal, like a random noise, or the extraction of useful parts of the signal, like the components located in a specific frequency range, is generally the filters function, where in general, based on the physical realization, are classified into two categories, analog filters and digital filters. For the analog filter, an analog electronic circuits are used, which consisting of components such as resistors, capacitors and op-amps to produce the wanted filtering effect. For the digital filter, a digital processor is used to execute numerical calculations on sampled values of the signal. [1]

- **Advantages of using digital filter**

Some of the principal advantages of digital filters compared to analog filters are listed in the following [2]

1. A digital filter is software programmable, which makes it easy to design and test.
2. Digital filters necessitate just the arithmetic operations, which mean the implementation of a digital filter requires only adders, multipliers, delays, etc. yielding to a very simple hardware requirements compared with the equivalent analog circuit.
3. Unlike the analog filters, the digital filters are not influenced by ageing nor the temperature, and respect to time and temperature together, a digital filter is very stable.
4. Digital filters are much greater in In terms of performance compared to analog filters.
5. Speedy DSP processors can deal with complicated combinations of filters in parallel or cascade, making the multiple filtering possible.
6. Digital filters have the ability to process signals in different ways; for example, certain types of digital filter have the ability to adapt to modify the characteristics of the signal.

3. Ideal filter

An Ideal filter will not modify the component frequencies of the input signal, nor will it insert other frequencies to that signal, on the other hand, it will modify the relative amplitudes of the different frequency components and/or their phase relationships.

A filter $H(\omega)$ is ideal if

$$|\mathrm{H}(\omega)| = \begin{cases} 1, if\ \omega\, is\, in\, the\ passband \\ 0, if\ \omega\, is\, in\, the\, stopband \end{cases} \tag{1.1}$$

where $|\mathrm{H}(\omega)|$ is the amplitude response.

The four types of an ideal filter are shown in figure 1.2.

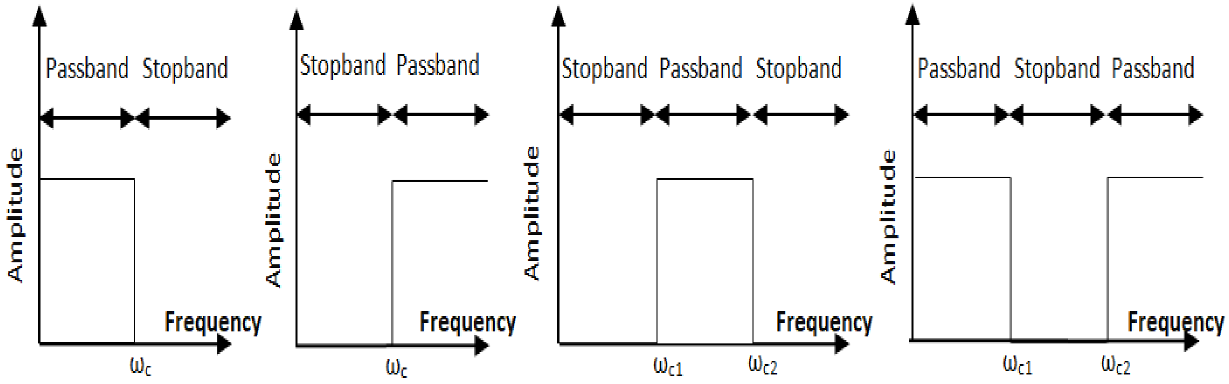


Fig. 1.2 Responses of an ideal filter

where ω_c, ω_{c1} and ω_{c2} specify the cut-off frequencies.

4. Practical filter

Real filters are different from the ideal filters, because they cannot achieve all the criteria of an ideal filter. Real filters, in practice, also have a transition band that laying between the pass band and the stop band. This transition band is the band of frequencies which necessarily has to be present because filters with sharp characteristics are unrealizable.

Figure 1.3 shows these types of filters.

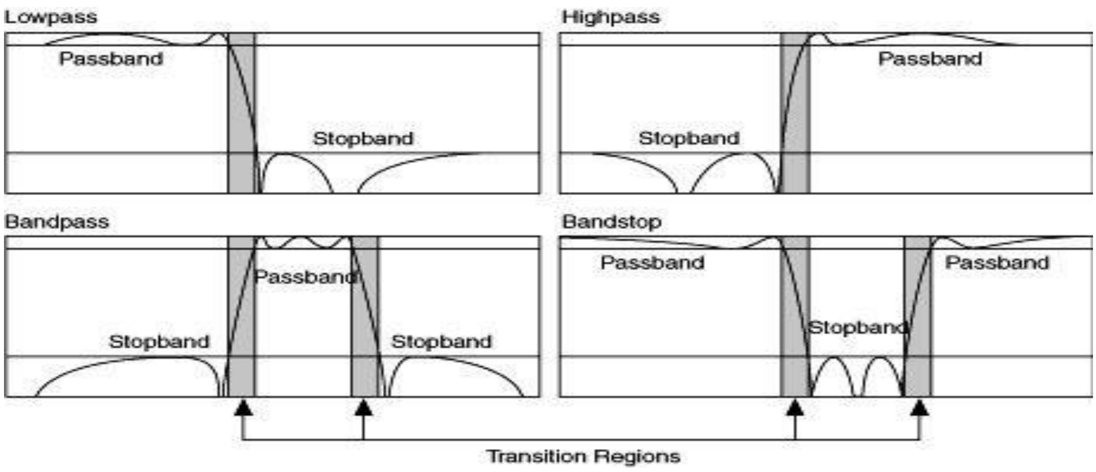


Fig. 1.3 Response of non-ideal filters [3]

5. Digital filters

In Digital Signal Processing (DSP) and telecommunication systems, the most powerful tools and fundamental part are the digital filters, which can be classified into two broad categories based on the shape of impulse response of the system, which are finite impulse response (FIR) filters and infinite impulse response (IIR) filters [4].

5.1. FIR filters

Finite impulse response (FIR) filter is a filter whose impulse response is of limited duration, because it settles to zero in finite period. The implementation of an FIR filter is generally performed using numbers of adders, multipliers and delays, to generate the filter's output. The finite impulse response filters also known as non-recursive or feed forward, which is a feed-forward filter without feedback.

The FIR filter transform function can be expressed as:

$$H(z) = \sum_{n=0}^{N} h(n) * Z^{-n} \tag{1.2}$$

where N is the order of the filter which has N+1 coefficients. $h(n)$ is the filter impulse response.

The difference equation representation is

$$y(n) = \sum_{k=0}^{N} b_k x(n-k) \tag{1.3}$$

$x(n)$ is the input signal and $y(n)$ is the output signal.

Figure 1.4 shows the basic block diagram of FIR filter of order N.

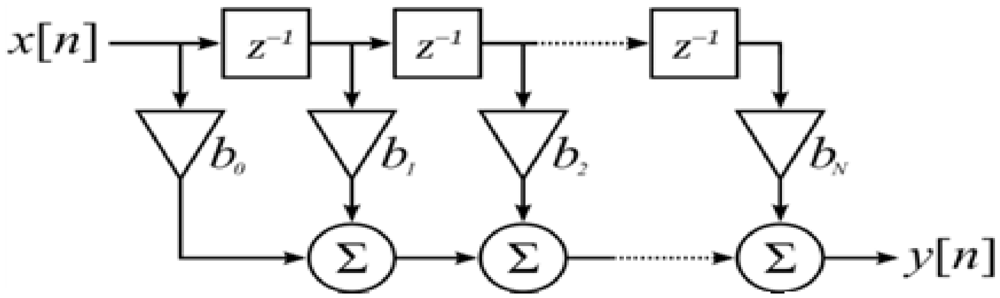


Fig. 1.4 the basic block diagram of FIR filter

Due to its finite impulse response, the FIR filter is considered as a stable filter. The stability is assured for the reason that the poles of the system are laying inside of the unit circle on the z-domain. In other words, the poles of transfer function are placed in the origin on the z-domain. [5]

❖ Properties of FIR filter

- ✓ The outputs depend only on the feedforward inputs, which mean it requires no feedback.
- ✓ Since they require no feedback, in other word, all the poles are located at the origin within the unit circle (that considered the stability condition in a Z transformed system), FIR filter are naturally stable.
- ✓ They have a linear phase; therefore, by making the coefficient sequence symmetric in the designing, easily yielding to an FIR filter with a linear phase.
- ✓ FIR filters are employed for tapping of higher order filters.
- ✓ FIR filters known as all-zero filters, due to the fact that they have only zeros.

5.2. IIR filters

The Infinite Impulse Response (IIR) filter, also known as recursive filters which means it presents a feedback and that the output is based on the combination of feed-forward inputs and feedback outputs. It has an impulse response in which the number of non-zeros samples are infinite

In general, the infinite impulse response filter is described by the following difference equation:

$$y(n)=\sum_{i=0}^{N}a_i.x(n-i)-\sum_{i=1}^{M}b_i.y(n-i) \tag{1.4}$$

Where x and y represent the input and output signal, respectively.

Therefore, the filter transfer function can be presented in the following general form [6]:

$$H(z)=\frac{Y(z)}{X(z)}=\frac{\sum_{i=0}^{N}a_i.Z^{-i}}{1+\sum_{i=1}^{M}b_i.Z^{-i}} \tag{1.5}$$

Where a_K and b_K represent the filter coefficients, these specify the characteristics of the filter. In General, the numerator and denominator have the same order M = N.

The architecture of the IIR transfer function in the z-domain could be illustrated in the Figure 1.5 below.

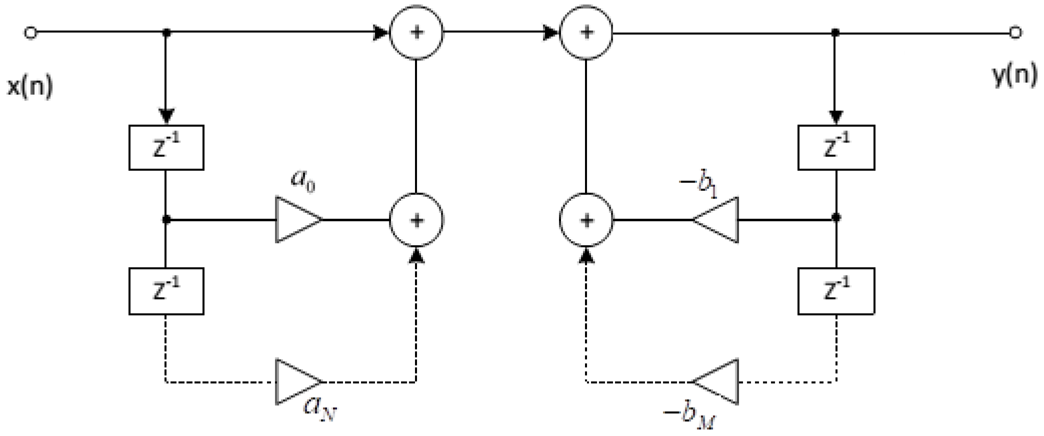


Fig. 1.5 The basic block diagram of IIR filter

- **Properties of IIR filter**

  - ✓ They have non-linear Phase.
  - ✓ Require fewer coefficients, this means lower computational cost than the FIR filters; they require fewer delay elements, adders, and multipliers.
  - ✓ IIR filters are more efficient and can provide much better performance than FIR filters because both poles and zeros are present.
  - ✓ On the other hand, feedback can cause the filter to become unstable if the poles located outside the unit circle.
  - ✓ IIR filters have the possibility of making polyphase implementation, while FIR filters can permanently be made casual.
  - ✓ IIR filter has infinite number of non-zero impulse responses and utilized for applications where linear phase characteristic is not necessary.
  - ✓ IIR filters are used lower-order tapping.
  - ✓ An IIR filter can give a sharper cut-off than an FIR filter of the same order.
  - ✓ IIR filters are recursive, which means they present a feedback; the outputs depend on a combination of feedforward inputs and feedback outputs.

6. FPGA overview

The FPGA or Field programmable Gate Array [7], is integrated circuit designed to be programmed by a designer after manufacturing, to turn into almost any type of digital circuit or system. The FPGA configuration is typically customized using a hardware description language (HDL), similar to that used for an Application Specific Integrated Circuit (ASIC), but unlike ASIC, they provide cheaper solution and faster time to market. Another advantages of the FPGA, is that they take a short time to configure and they are cheap devices; a partial reconfiguration of a portion of FPGA can be easily done; The ability to update the functionality by just downloading a new application bitstream.

As it shown in the figure 1.6, an FPGA consists of:

- ✓ Configurable logic blocks which implement the logic functions.
- ✓ Programmable routing channel, which connects the logic functions.
- ✓ I/O blocks that used to make off-chip connections through the routing channel.

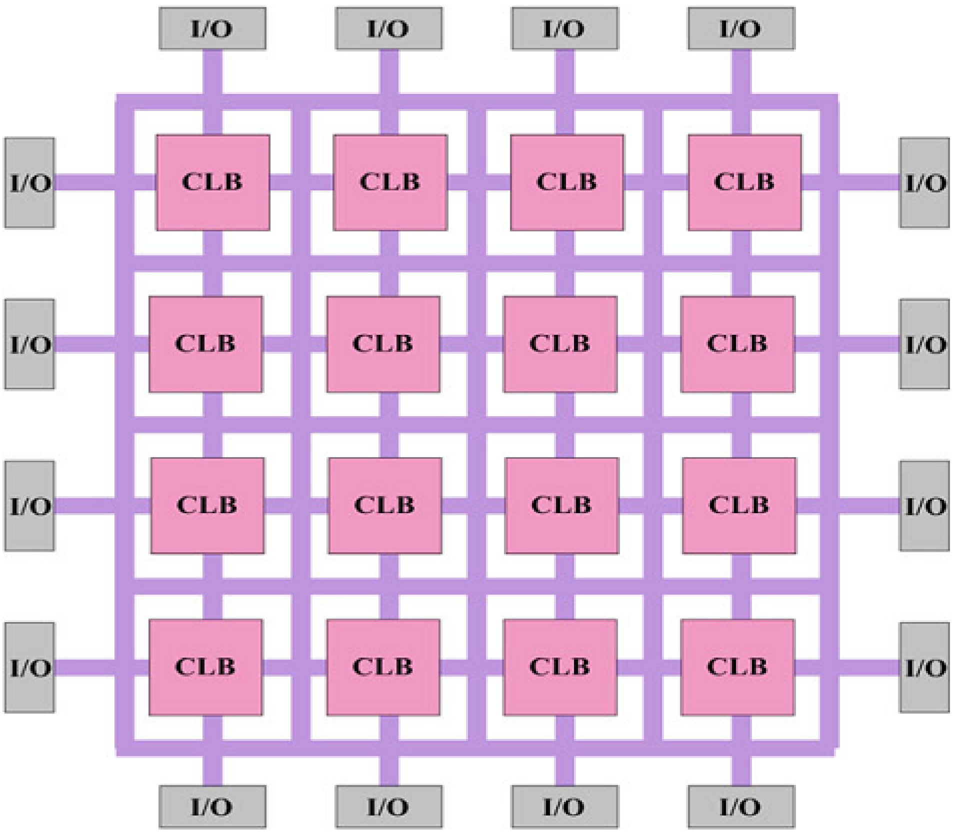


Fig. 1.6 Overview of FPGA architecture [8]

6.1. The VHDL

VHDL is for VHSIC Hardware Description Language, where VHSIC stands for Very High Speed Integrated Circuit, which is a programming language used to describes a digital systems, such as field-programmable gate arrays and integrated circuits, by function, data flow behavior, and/or structure, based on a custom logic design.

The common form of a VHDL structure is made in the concept of BLOCKS which are the essential designing components of a VHDL design. Based on these design blocks a function of digital circuit can be easily described.

There are a few VHDL development environments provided by programmable logic chip manufacturers. Here is a summary list of some of the main free environments available:

- Quartus II Web Edition by Altera.
- ISE Webpack of the company Xilinx.
- ispLever starter from Lattice Company.
- Libeiro of the company Actel.

6.2. Xilinx system generator

The 'Xilinx system generator' [8]is a high-level tool for designing high-performance DSP systems using FPGAs. The system generator tool enables us to integrate Xilinx with Simulink, it generates a .ise file which is utilized in Xilinx using the model file of Simulink.

The Xilinx block sets function within gateways only i.e. 'gateway-in' and 'gateway-out' [9] blocks which are available in Xilinx Block set library.

Any sample based input is to be passed through gateway-in block before being fed to any Xilinx block set, and then final output can be seen on 'scope' by observing the output from gateway-out.

7. Conclusion

In this chapter we present an overall of the basics of the digital filters, we introduce the fundamental concepts and terms associated with the filters; an overview of the FPGA and techniques used for the implementation are presented here. This chapter can serve as a starting point for our thesis objective.

References

[1] S. Das and A. Konar, “Design of two dimensional IIR filters with modern search heuristics: A comparative study,” Int. J. Comput. Intell. Applicat., vol. 6, no. 3, pp. 329–355, 2006.

[2] L. Tan, Digital Signal Processing: Fundamentals and Applications. Academic Press, 2007.

[3] S. K. Mitra, Digital Signal Procesing: A Computer-Based Approach, 3rd Edition, McGraw-Hill, 2008.

[4] A.V. Oppenheim, R.W. Schafer, J.R. Buck, Discrete-time Signal Processing, Second ed., Prentice Hall, 1999.

[5] Markovic Dejan and Brodersen Robert. DSP Architecture Design Essentials. USA, New York: Springer; 2012.

[6] Nambiar R, Tang CKK, Mars P. Genetic and learning automata algorithms for adaptive digital filters. Proc ICASSP 1992. 1992;4:41-44.

[7] U. Farooq, Z. Marrakchi, and H. Mehrez, “FPGA architectures: An overview,” in Tree-based Heterogeneous FPGA Architectures. New York, NY, USA: Springer, 2012, pp. 7–48.

[8] “Xilinx ISE 9.2i Software Manuals: Constraints Guide, and Development System Reference Guide”. San Jose, California, USA, pp. 1 -844

[9] Xilinx Inc., “System Generator for DSP”: User Guide, Release 10.1,Snn Jose, Clifornia, USA, PP.1 -40: www.xilinx.com, 2008.

Chapter 2

REVIEW OF TECHNIQUES FOR IIR FILTER DESIGN

1. Introduction

The filter coefficients calculation is probably the mainly essential step in digital filter design, since the coefficients play the significant part in specifying the characteristics of the filter. There are numerous methods for designing IIR filters; in general, the design of this kind of filters follows two techniques [1]: transformation technique and optimization technique.

In the transformation techniques (or the classical methods) often involve the complication of transforming an analogue filter to its digital equivalent, where the minimum order and the coefficients of the filter are chosen for a typical prototype low pass Butterworth, Chebyshev Type-I, Chebyshev Type-II or Elliptic filters then converted to digital low pass, high pass, band pass, band stop IIR digital filter through applying various transformation techniques like the bilinear transformation etc [2]. In this chapter we will present the details of these techniques.

2. Infinite Impulse Response filter

An infinite impulse response (IIR) filters are linear time invariant (LTI) systems described by the following difference equation:

$$y(n)=\sum_{k=0}^{N}a_k.x(n-k)-\sum_{k=1}^{M}b_k.y(n-k) \tag{2.1}$$

Where $y(n)$ is the output of the filter and it can be written as

$$y(n)=\sum_{k=0}^{\infty}h(k)x(n-k) \tag{2.2}$$

The filter transfer function can be expressed in term of the impulse response by the following general form

$$H(z)=\sum_{k=0}^{\infty}h(k)z^{-k} \tag{2.3}$$

The frequency response is expressed as follows:

$$H\left(e^{j\omega}\right)=\sum_{n=0}^{\infty}h(n)e^{-j\omega} \tag{2.4}$$

In the difference equation described before, we can rewrite the transfer function as:

$$H(z)=\frac{B(z)}{A(z)} \tag{2.5}$$

Where

$$B(z)=\sum_{n=0}^{N}a(n)z^{-n} \tag{2.6}$$

$$A(z)=1+\sum_{n=1}^{M}b(n)z^{-n} \tag{2.7}$$

Or

$$H(z)=\frac{a_0+a_1z^{-1}+a_2z^{-2}+\cdots+a_Nz^{-N}}{1+b_1z^{-1}+b_2z^{-2}+\cdots+b_Mz^{-M}} \tag{2.8}$$

$H(z)$ can also be written as

$$H(z)=\frac{z^{-N}}{z^{-M}}\cdot\frac{a_0z^{N}+a_1z^{N-1}+a_2z^{N-2}+\cdots+a_N}{z^{M}+b_1z^{M-1}+b_2z^{M-2}+\cdots+b_M} \tag{2.9}$$

Where the roots of the polynomial

$$a_0z^{N}+a_1z^{N-1}+a_2z^{N-2}+\cdots+a_N$$

Are the zeros of $H(z)$.

And for the polynomial

$$z^{M}+b_1z^{M-1}+b_2z^{M-2}+\cdots+b_M$$

Its roots are the poles of $H(z)$.

- **Remarks on IIR digital filters**

IIR digital filters have several properties, among them listed in the following [3]:

- They have a feedback from output to input which make them a recursive.

- They have a non linear phase.
- Both poles and zeros are present. If the poles are located outside the unit circle. The filter becomes unstable.
- The IIR filters implementation is more susceptible to finite precision effects compared with the FIR filters.
- The main advantage of the IIR filters compared to FIR filters, is that they can provide a much better performance and sharp cut-off than the FIR filters, and they can satisfy the given filter specifications with a much lower filter order.

3. The classical methods for IIR filter design

For IIR filter design, the most common method is to design an analog IIR filter and then convert it into an equivalent digital filter. This method has several properties, among them [3]:

- Not all IIR digital filters can be achieved using the transformation method, which in turn make it limited suppleness. On the other hand, the converting of an analog filter to digital filter runs Excellent for designing standard types: low-pass, high-pass, band-pass, and band-stop filters. Still, for the non-standard types, these classical techniques usually not preferred.
- They developed to achieve a good magnitude response but can produce a very nonlinear phase response, in other words, there is no control on the phase response; this considered as disadvantage of these methods.
- The resulting digital filter have same order for both the numerator and denominator of its transfer function H(z).
- These methods used only for IIR filter design, which means they can’t be used not for designing an FIR filter.

3.1. ANALOG FILTERS

The analog filter transfer function is given by the Laplace transform of its impulse response:

$$H_a(s) = \int_{-\infty}^{\infty} h(t) e^{-st} dt \tag{2.10}$$

To indicate an analog transfer function, we used the subscript a . The transfer function of a realizable analog filter can be expressed as a rational function:

$$H_a(s) = \frac{P(s)}{Q(s)} \tag{2.11}$$

where $P(s)$ and $Q(s)$ are polynomials in s. Through evaluation $H_a(s)$ on the imaginary axis, the analog filter's frequency response is obtained, and it's written as:

$$H_a(j\omega) = H_a(s)\big|_{s=j\omega} \tag{2.12}$$

All the poles of $H_a(s)$ should located in the left half plane (LHP) for a stable, causal filter.

The magnitude square response:

$$\begin{aligned} |H_a(j\omega)|^2 &= H_a(j\omega)\cdot\overline{H_a(j\omega)} \\ &= H_a(j\omega)\cdot H_a(-j\omega) \\ &= H_a(s)\cdot H_a(-s)\big|_{s=j\omega} \end{aligned}$$

Define

$$R(s) \coloneqq H_a(s)\cdot H_a(-s) \tag{2.13}$$

This means

$$R(j\omega) = |H_a(j\omega)|^2 \tag{2.14}$$

For ease of use, we set

$$\mathrm{F}(\omega) \coloneqq \mathrm{R}(\mathrm{j}\omega) \tag{2.15}$$

so that we don't have to convey j over again, then

$$\mathrm{R}(\mathrm{s}) = \mathrm{F}(\mathrm{s}/\mathrm{j}) \tag{2.16}$$

And

$$\mathrm{F}(\omega) = |\mathrm{H_a}(\mathrm{j}\omega)|^2 \tag{2.17}$$

Based on the following process, the developing of classical analog filters is basically done by the process below:

– Design the function $F(\omega)$ so that $R(s)=F(s/j)$ can be spectrally factored, then compute a spectral factor to obtain $H_a(s)$. That means, given $R(s)$, find $H_a(s)$ so that

$$R(s)=H_a(s)\cdot H_a(-s)$$

For that, it must consider the following issue:

Design a rational function $F(\omega)=|H_a(j\omega)|^2$ such that:

- $F(\omega)\geq 0$
- $F(\omega)$ is an even function of ω

The $F(\omega)$ of a classical lowpass analog filter is formulated by writing it as:

$$F(\omega)=\frac{1}{1+\epsilon^2 V(\omega)^2} \quad (2.18)$$

The three most commonly used analog filters are the Butterworth, Chebyshev (with its two types), and elliptic filters. These filters are described below, and they are basically depending on the theory in [3].

3.1.1. The Butterworth analog filter

For the Butterworth filter, which considered as the simplest case, the functional form of $V(\omega)$ is

$$V(\omega)=\omega^N$$

Giving the magnitude response of the Butterworth filter:

$$F(\omega)=\frac{1}{1+\epsilon^2\omega^{2N}} \quad (2.19)$$

As an example, for $N=3$ and $\epsilon=0.5$, the functions $V(\omega)$ and $F(\omega)$ are illustrated bellow (figure 2.1).

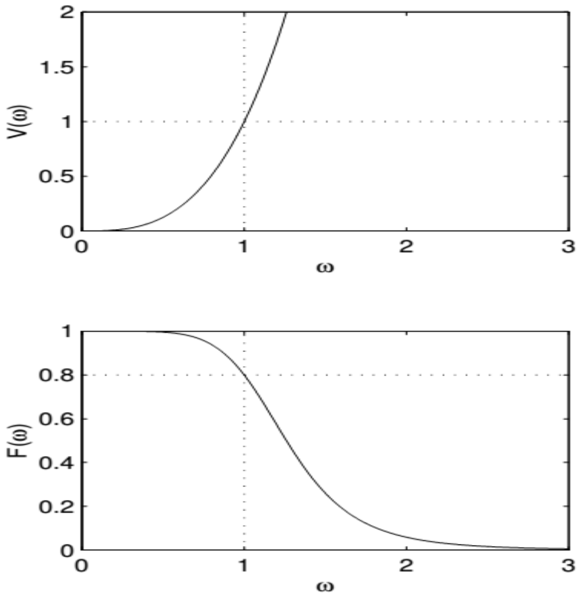


Figure 2.1 Butterworth analog filter

We start first by obtain $R(\mathrm{s})$ to get the transfer function $H_a(s)$:

$$R(\mathrm{s}) = F(\mathrm{s}/\mathrm{j}) = \frac{1}{1+\epsilon^2 (\mathrm{s}/\mathrm{j})^{2\mathrm{N}}}$$
$$= \frac{1}{1+\epsilon^2 (\mathrm{s}^2/\mathrm{j}^2)^{\mathrm{N}}}$$
$$= \frac{1}{1+\epsilon^2 (-\mathrm{s}^2)^{\mathrm{N}}}$$

As you cans see, there are no finite zeros since the numerator is simply 1, and by setting the denominator of $R(\mathrm{s})$ to zero allowed us to find the poles. This can be done as follow:

$$1+\epsilon^2 (-1)^{\mathrm{N}} \mathrm{s}^{2\mathrm{N}} = 0$$
$$\mathrm{s}^{2\mathrm{N}} = \frac{-1}{\epsilon^2 (-1)^{\mathrm{N}}}$$
$$\mathrm{s}^{2\mathrm{N}} = \frac{-1(-1)^{\mathrm{N}}}{\epsilon^2}$$

This becomes

- **When N is even**

Setting the denominator of $R(\mathrm{s})$ to zero gives

$$\mathrm{s}^{2\mathrm{N}} = \frac{-1}{\epsilon^2}$$

To get the poles expression, we can set

$$-1 = e^{j\pi}$$

Or

$$-1 = e^{j(\pi + 2\pi k)}$$

This leads to the following chain

$$\begin{aligned} s^{2N} &= \frac{-1}{\epsilon^2} \\ &= \frac{e^{j(\pi+2\pi k)}}{\epsilon^2} \\ s &= \left(\frac{e^{j(\pi+2\pi k)}}{\epsilon^2}\right)^{\frac{1}{2N}} \\ &= \frac{e^{j(\pi+2\pi k)/(2N)}}{\epsilon^{1/N}} \\ s &= \frac{e^{j(1+2k)\frac{\pi}{2N}}}{\epsilon^{1/N}} \end{aligned}$$

For $0 \le k \le 2N-1$

- **When N is odd**

$$s^{2N} = \frac{1}{\epsilon^2}$$

To get the poles expression, we can set

$$\begin{aligned} &1 = e^{j2\pi} \\ &or \\ &1 = e^{j2\pi k} \end{aligned}$$

This leads to the following chain

$$\begin{aligned} s^{2N} &= \frac{1}{\epsilon^2} \\ &= \frac{e^{j2\pi k}}{\epsilon^2} \\ s &= \left(\frac{e^{j2\pi k}}{\epsilon^2}\right)^{\frac{1}{2N}} \end{aligned}$$

$$s = \frac{e^{j\pi k/N}}{\epsilon^{1/N}}$$

For $0 \le k \le 2N-1$

After these roots are obtained, we can determine the roots of $Q(s)$ by picking those in LHP.

The roots in the LHP are presented by the expression below, useful whatever N is, whether even or odd.

$$s_k = \frac{1}{\epsilon^{1/N}} \cdot e^{j(2k+1+N)\pi/(2N)} \tag{2.20}$$

For $0 \le k \le 2N-1$

These are the poles of an Nth order Butterworth analog filter.

3.1.2. The Chebyshev-I analog filter

For the Chebyshev-I analog filter, the functional form of $V(\omega)$ is basically depends on the Chebyshev polynomials $C_N(\omega)$.

For the Chebyshev-I filter:

$$V(\omega) = C_N(\omega)$$

Then

$$F(\omega) = \frac{1}{1+\epsilon^2 C_N(\omega)^2} \tag{2.21}$$

The notable Chebyshev polynomials can be generated by the following recursive formula:

$$C_0(\omega) = 1$$
$$C_1(\omega) = \omega$$
$$C_{k+1}(\omega) = 2\omega C_k(\omega) - C_{k-1}(\omega)$$

The next few $C_k(\omega)$ are

$$C_2(\omega) = 2\omega^2 - 1$$
$$C_3(\omega) = 4\omega^3 - 3\omega$$
$$C_4(\omega) = 8\omega^4 - 8\omega^2 + 1$$
$$C_5(\omega) = 16\omega^5 - 20\omega^3 + 5\omega$$

As it shown in the following figure (fig.2.2), the Chebyshev polynomial oscillates between −1 and 1 in the interval $-1 \le \omega \le 1$. This will produce an equiripple behavior in the pass-band of the resulting analog filter.

3.1.2.1. Chebyshev polynomials

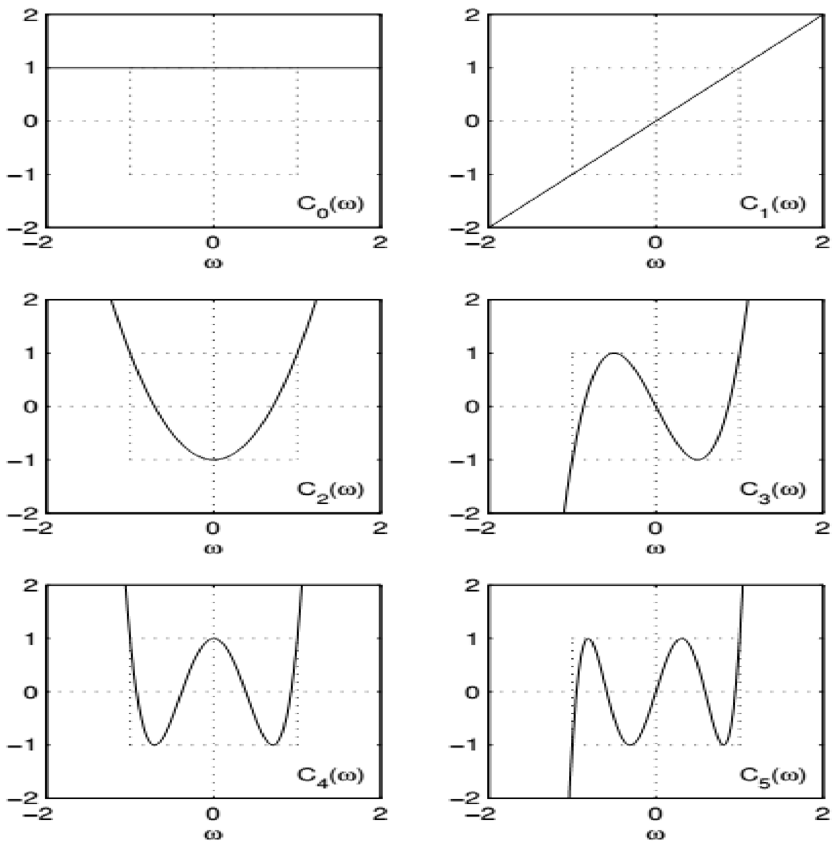


Figure 2.2 Chebyshev polynomials

Pass over the details, the poles of the Chebyshev-I filter be located at

$$s_k = -\sinh(v)\sin\left(\frac{(2k+1)\pi}{2N}\right) + j\cosh(v)\cos\left(\frac{(2k+1)\pi}{2N}\right) \tag{2.22}$$

For $0 \le k \le N-1$ where

$$v = \frac{\sinh^{-1}\left(1/\epsilon\right)}{N} \tag{2.23}$$

The Chebyshev-I analog filter frequency response is equiripple in the pass-band, and monotonic in the stop-band.

For this filter, $\left|H_a(j\omega)\right|$ settle between the bounds 1 and $1-\delta_p$ in the pass-band. The value of the pass-band ripple R_p is related to δ_p by

$$1-\delta_p = 10^{-R_p/20}$$

Or

$$R_p = -20\log_{10}\left(1-\delta_p\right)$$

3.1.3. The Chebyshev-II analog filter

The Chebyshev-II analog filter or the inverse-Chebyshev filter is designed that have a monotonic pass-band and an equiripple stop-band. The frequency response of this filter can be achieved by following these two steps:

- ✓ First, subtract the Chebyshev-I $F(\omega)$ from 1.
- ✓ Second, perform the change of variables $\omega \leftarrow 1/\omega$.

An example of a Chebyshev-II analog filter is presented in the figure below fig.2.3.

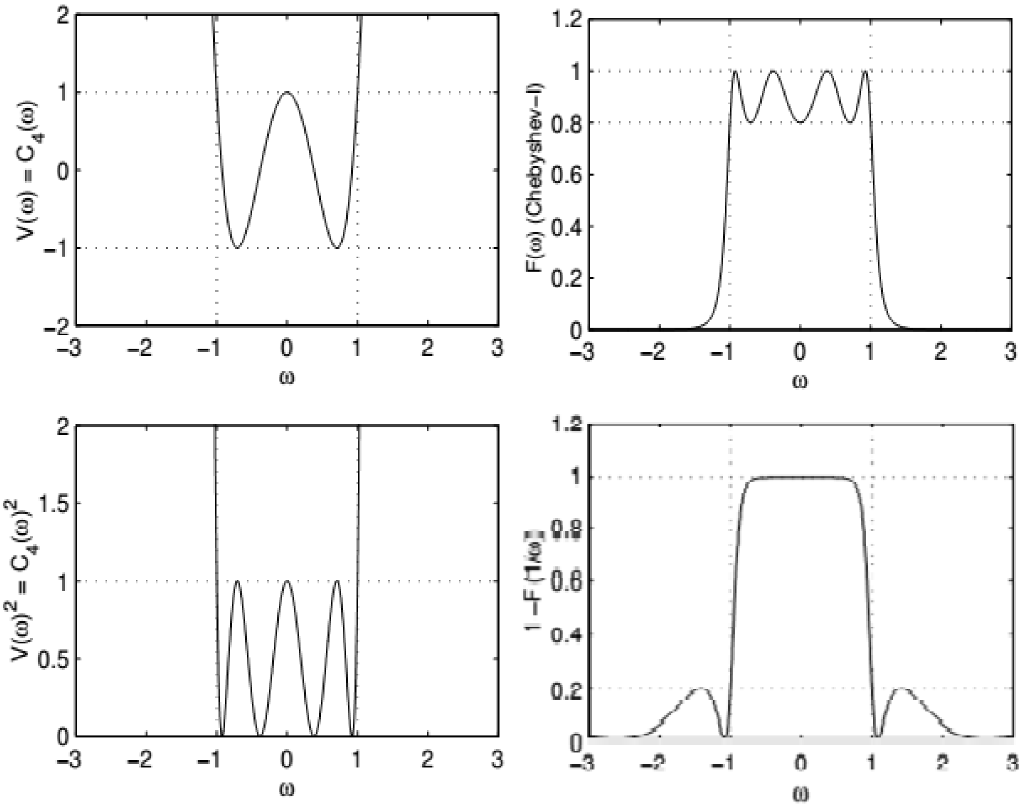


Figure 2.3 Frequency response of the Chebyshev-I,II

As the figure shows, the frequency response of the Chebyshev-II filter has a monotonic passband and an equiripple stop-band.

Based on the two previous steps, for the Chebyshev-II analog filter $F(\omega)$ is specified by

$$F(\omega) = 1 - \frac{1}{1+\epsilon^2 C_N\left(1/\omega\right)^2} \tag{2.24}$$

Or

$$F(\omega)=\frac{\epsilon^2 C_N(1/\omega)^2}{1+\epsilon^2 C_N(\omega)^2} \tag{2.25}$$

Since the numerator is not equal to 1, the Chebyshev-II filter has zeros. It clarifies that all of its zeros located on the imaginary axis. Pass over the details, the zeros are characterized by

$$z_k=\frac{1}{\cos((2k+1)\pi/(2N))} \tag{2.26}$$

For this filter, $|H_a(j\omega)|$ settle between the bounds 0 and δ_s in the stop-band. The value of the stop-band ripple R_s is related to δ_s by

$$\delta_s=10^{-R_s/20}$$

Or

$$R_s=-20\log_{10}(\delta_s)$$

For $0\le k\le N-1$.

Obviously, the poles of the Chebyshev I and II analog filters are reciprocals.

3.1.4. The Elliptic analog filter

For the elliptic analog filter, $V(\omega)$ is the Chebyshev rational function $R_N(\omega,\alpha)$ that based on a parameter α :

$$F(\omega)=\frac{1}{1+\epsilon^2 R_N(\omega,\alpha)^2} \tag{2.27}$$

The elliptic analog filter is equiripple in the pass-band and in the stop-band. For specific characterizations, the elliptic filter gives the minimal-degree filter.

The functional form for $V(\omega)$ is highly intricate as it is depend on elliptic functions.

The magnitude of the frequency response of the elliptic filter $|H_a(j\omega)|$ will lie within the bounds $1-\delta_p$ and 1 in the pass-band and it will locate within the bounds 0 and δ_s in the stop-band. The values R_p and R_s are related to the pass-band ripple by

$$\delta_p = 1 - 10^{-R_p/20}$$
$$\delta_s = 10^{-R_s/20}$$

Or

$$R_p = -20\log_{10}\left(1-\delta_p\right)$$
$$R_s = -20\log_{10}\left(\delta_s\right)$$

3.2. Converting analog filters to digital filters

3.2.1. The bilinear transformation

The most popular method for converting an analog filter to a digital one is the Bilinear transform (BLT). In method, the imaginary axis of Laplace plan is mapped to the unit circle in the z plan.

- The BLT is basically consisting on the change of variables.

$$s = \frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}} \tag{2.28}$$

- In this method, the points lied in the LH of s-plan are mapped inside the unit circle $|z| \le 1$ in the z plan.

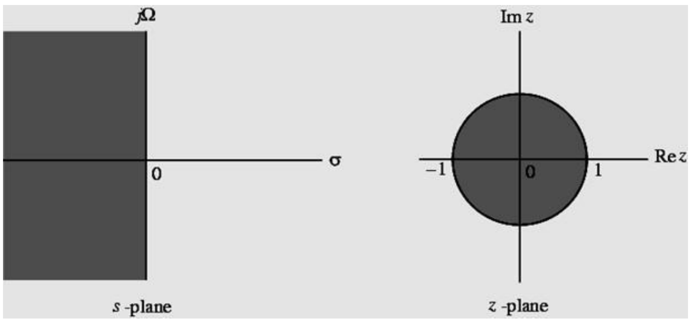


Fig. 2.1 s-plane to z -plane mapping in the BLT technique.

- This transformation technique is a non-linear one which create a phenomenon known as the warping effect.

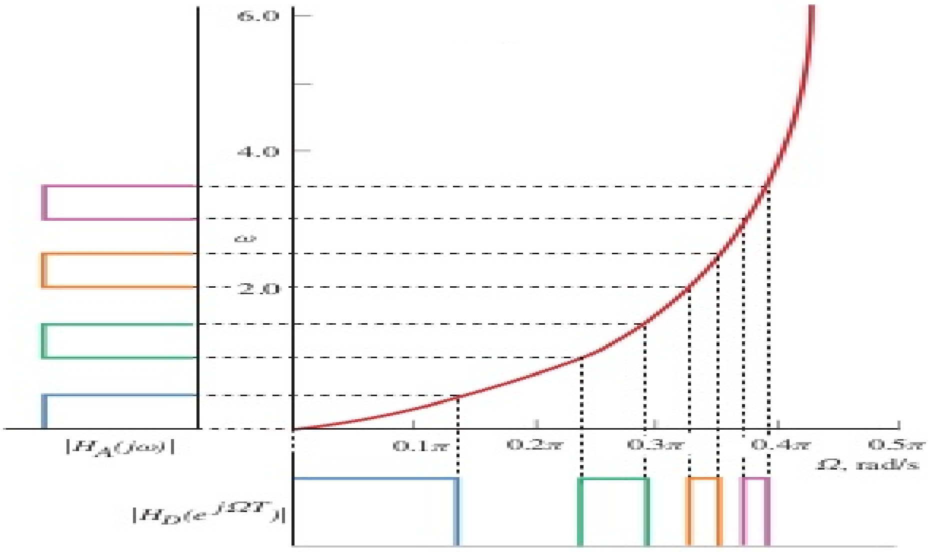


Fig. 2.2 Influence of the warping effect on the amplitude response.[4]

This warping effect can be eliminated by a prewarping technique using the following formula

$$\Omega = \frac{2}{T}\tan\left(\frac{\omega}{2}\right) \tag{2.29}$$

<u>The design steps:</u>

- ✓ For the given specifications, find prewarping analog frequencies.
- ✓ Using the obtained analog frequencies, find the transfer function $H_a(s)$ of the analog filter.
- ✓ Substitute, $s = \frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}$ into the transfer function found in step 2 to get the digital filter transfer function $H(z)$.

❖ **<u>REMARKS ON THE BLT</u>**

- ✦ $H(z)$ and $H_a(s)$ have the same order;
- ✦ The points located in the left half of Laplace plan are mapped inside the unit circle $|z| \le 1$ in the z plan;
- ✦ A causal stable analog system is transformed using the BLT into a causal stable digital system, since all poles in the LHP are mapped to the inside of the unit circle.

3.2.2. Impulse-invariance method

Another popular method for converting analog filter to digital filter is the impulse-invariance method.

- $h(n) = h_a(nT)$ for $0 \le n \le N-1$. The impulse response of the digital filter is corresponded to the samples of the analog filter impulse response.
- The shape of $H_a(j\Omega)$ does not conserve.
- Unlike the BLT does unlike the BLT technique, the Impulse-invariance method generally does not conserve the stability of causal systems.

The impulse-invariance method, unlike the bilinear Transformation, is not generally preferred for IIR digital filter design.

4. The Optimal methods for IIR filter design

On the other hand, the optimization methods can be utilized to pass over these steps, by working directly on the coefficients or the pole-zero positions, in order to obtain the coefficients which best fit the desired magnitude and/or phase responses. An objective function could be formulated which leads these methods towards these responses, so that the designer doesn't have to know anything about the existing operations involved.

The filter transfer function can be expressed as:

$$H(z) = \frac{B(z)}{A(z)} = \frac{\sum_{i=0}^{N} a_i . Z^{-i}}{1 + \sum_{i=1}^{M} b_i . Z^{-i}} \tag{2.30}$$

Where a_i and b_i represent the filter coefficients, which consequently determine the characteristics of the filter. An important task for the designer is to find values of them.

So, the design of the filter can be considered as an optimization problem of cost function $J(x)$ stated as $\min J(x)$ *where* $x = [a_0,\ a_1,\ a_2, \ldots,\ a_L,\ b_1,\ b_2,\ \ldots,\ b_M]$ is filter coefficient vector.

The cost function in the designing of IIR filter is to minimize difference between the ideal and the designed magnitude response with respect to the transfer function coefficients, the aim is to minimize $J(x)$ by adjusting x.

The Lp-norm approximation error for the magnitude response is defined as [5]:

$$J(x) = \sum_{i=0}^{K} \left\{ \left\| H_I(\omega_i) \right| - \left| H_D(\omega_i, x) \right\|^{p} \right\}^{1/p} \tag{2.31}$$

where $H_I(\omega_i)$ and $H_D(\omega_i, x)$ are the desired and actual responses of the filter, respectively.

The optimization methods, in which this can be achieved, will be covered in the next chapters.

5. Conclusion

As conclusion, this chapter gave an overview on the three classical methods that used for recursive filter design. All these design methods involving a transfer function of an analog prototype filter: Butterworth, Chebyshev or elliptic, and by employing one of the two converting techniques, the bilinear transformation or the impulse-invariance method, we obtain a digital IIR filter.

References

[1] Kaur R, Patterh MS, Dhillon JS. Design of optimal L1 stable IIR digital filter using hybrid optimization algorithm. Int J Comput Appl (0975 – 8887). 2012;38(2, January).

[2] Saha SK, Kar R, Mandal D, Ghoshal SP. Digital stable IIR low pass filter optimization using PSO-CFIWA. In: 2012 1st International Conference on Recent Advances in Information Technology (RAIT); 2012:196-201.

[3] I. Selesnick, IIR Filter Design. Lecture Notes, EL 713, NYU Polytechnic School of Engineering (2011). http://eeweb.poly.edu/iselesni/EL713/index.html

[4] A. Antoniou, Digital Signal Processing: Signals, Systems, and Filters. New York: McGraw-Hill, 2005.

[5] Jinn-Tsong Tsai and Jyh-Horng Chou, "Optimal design of digital IIR filters by using an improved immune algorithm" IEEE Transactions on Signal Processing, Vol. 54, No. 12, pp. 4582–4596, December 2006.

Chapter 3

DESIGN ALGORITHMS AND CRITERIA

1. Introduction

In previous Chapter, several methods for designing recursive filters have been presented. These methods drive to a full description of the transfer function in closed form, whether in terms of its zeros and poles or its coefficients. They are, as a result, very effective and lead to very accurate designs. Their major disadvantage is that they are useable only for the design of filters with piecewise-constant amplitude responses [1].

A different way to get the solution of the approximation problem in digital filters design is through the use of optimization techniques [2-6]. These methods can be utilized to pass over these steps, by working directly on the coefficients or the pole-zero positions, in order to obtain the coefficients which best fit the desired magnitude and/or phase responses, so that the designer doesn't have to know anything about the existing operations involved. In these methods, and for a given discrete time transfer function, an objective function could be formulated depending on some desired specification (amplitude and/or phase response). A norm of the objective function is then minimized with respect to the coefficients vector. As the value of the norm approaches to the zero, the obtained results (amplitude or phase response) approach to the desired ones.

Such a filter can be designed, in general, by following these two steps [1]:

1) An objective function is formulated, which is basing on the difference between the actual and desired amplitude response.
2) The obtained objective function is then minimized, with taking into account the transfer-function coefficients.

From Andreas Antoniou book [1], we present an overview about the lp norm method as it shown in the following.

2. ℓp-Norm Method

The distance approach refers to an approximation problem for which different norms may be chosen, such as the L_p norms.

The main goal of the design algorithm of digital IIR filter is to search for filter coefficients a_k and b_k.

An Nth-order recursive filter with N even can be represented by the transfer function [7]:

$$H(z) = H_0 \prod_{j=1}^{J} \frac{a_{0j} + a_{1j}z + z^2}{b_{0j} + b_{1j}z + z^2} \tag{3.1}$$

where a_{0j}, a_{1j} and b_{0j}, b_{1j} are real coefficients, $J = N/2$, and H_0 is a positive multiplier constant. The amplitude response of the filter can be expressed as

$$M(x,\omega) = \left| H\left(e^{j\omega T}\right) \right| \tag{3.2}$$

where

$$x = \left[a_{01} a_{11} b_{01} b_{11} \ldots b_{1J} H_0\right]^T$$

is a column vector with 4J + 1 elements and ω is the frequency.

Let $M_0(\omega)$ be the specified amplitude response. The difference between $M(x,\omega)$ and $M_0(\omega)$ is, in effect, the approximation error and can be expressed as

$$e(x,\omega) = M(x,\omega) - M_0(\omega) \tag{3.3}$$

By sampling $e(x,\omega)$ at frequencies $\omega_1, \omega_2, \ldots, \omega_K$, the column vector

$$E(x) = \left[e_1(x) e_2(x) \ldots e_K(x)\right]^T$$

can be formed where

$$e_i(x) = e(x, \omega_i) \tag{2.4}$$

for $i = 1, 2, \ldots, K$.

The approximation problem at hand can be solved by finding a point $x = \breve{x}$ such that

$$e_i(\breve{x}) \approx 0$$

for $i = 1, 2, \ldots, K$.

An objective function satisfying number of fundamental requirements can be defined as

$$\Psi(x) = \left[\sum_{i=1}^{K} \left| e_i(x) \right|^P \right]^{1/P} \tag{3.5}$$

where p is an integer.

The required design can be obtained by solving the optimization problem

$$\underset{x}{\text{minimize}} \ \Psi(x) \tag{3.6}$$

This optimization problem can be solved by using a minimax algorithm which is deployed to find the optimal solution for the objective functions.

A significant type of optimization algorithms, which have been found to be very efficient for the design of digital filters, is the type of quasi-Newton algorithms. These are basically depending on Newton's method to find the minimum in quadratic convex functions.

2.1. QUASI-NEWTON ALGORITHMS

Like the Newton algorithm, Quasi-Newton algorithms are formed for the convex quadratic problem and then, they are used for the general problem. The search direction, which considered as the main attitude in these algorithms, is based on an $n \times n$ matrix S that plays the same role as the inverse Hessian in the Newton algorithm. This matrix is developed using obtainable information and is designed to be an approximation of H^{-1}. In addition to that, with the increase in the number of iterations, S becomes gradually an increasingly accurate representation of H^{-1}. For convex quadratic objective functions, S turn out to be equal to H^{-1} in $n+1$iterations where n is the variables number [1].

2.1.1. Basic Quasi-Newton Algorithm

Consider a function $f(x)$ of n variables, where $x = [x_1 \ x_2, \cdots x_n]^T$ is a column vector, and let $\delta = [\delta_1 \ \delta_2 \cdots \delta_n]^T$ be a change in x.

At the points x_k and x_{k+1} the gradients of $f(x)$ will be g_k and g_{k+1}, respectively. If:

$$x_{k+1} = x_k + \delta_k$$

then the elements of g_{k+1} given as follow using the Taylor series

$$g_{(k+1)m} = g_{km} + \sum_{i=1}^{n} \frac{\partial g_{km}}{\partial x_{ki}} \delta_{ki} + \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\partial^2 g_{km}}{\partial x_{ki} \partial x_{kj}} \delta_{ki} \delta_{kj} + o\left(\|\delta\|_2^2 \right)$$

for $m = 1, 2, \ldots n$. Now if $f(x)$ is quadratic, the second and higher derivatives of $f(x)$ are constant and zero, respectively, and as a result the second and higher derivatives of g_{km} are zero. Thus

$$g_{(k+1)m} = g_{km} + \sum_{i=1}^{n} \frac{\partial g_{km}}{\partial x_{ki}} \delta_{ki}$$

and since

$$g_{km} = \frac{\partial f_k}{\partial x_{km}}$$

$$g_{(k+1)m} = g_{km} + \sum_{i=1}^{n} \frac{\partial^2 f_k}{\partial x_{ki} \partial x_{km}} \delta_{ki}$$

for $m = 1, 2, \ldots n$. Therefore, g_{k+1} is give by:

$$g_{k+1} = g_k + H\delta_k$$

where H is the Hessian $f(x)$. Otherwise, we can set

$$\gamma_k = H\delta_k \tag{3.7}$$

where

$$\delta_k = x_{k+1} - x_k$$

and

$$\gamma_k = g_{k+1} - g_k$$

The previous analysis demonstrate that, if the gradient of $f(x)$ is identified at two points x_k and x_{k+1}, a relation can be concluded that gives a specific amount of information about H, namely, Eq. (3.10). Since H is a real symmetric matrix with $n \times (n+1)/2$ unknowns and Eq. (3.10) provides only n equations, H cannot be determined uniquely through the use of Eq. (3.10). This problem can be defeated through the evaluating of the gradient sequentially at $n+1$ points, say at $x_0, x_1, \cdots, x_n$, such that the changes in x, to be precise

$$\delta_0 = x_1 - x_0$$
$$\delta_1 = x_2 - x_1$$
$$\vdots \qquad \vdots$$
$$\delta_{n-1} = x_n - x_{n-1}$$

from a set of linearly independent vectors. Based on these circumstances, Eq. (3.10) yields

$$[\gamma_0 \ \gamma_1 \cdots \ \gamma_{n-1}] = H[\delta_0 \ \delta_1 \cdots \delta_{n-1}]$$

Therefore, H can be uniquely determined as

$$H = [\gamma_0 \ \gamma_1 \cdots \ \gamma_{n-1}][\delta_0 \ \delta_1 \cdots \delta_{n-1}]^{-1} \tag{3.8}$$

Further progress to the development of the quasi-Newton method can be achieved by generating the matrix H^{-1} from calculated data using a set of linearly independent vectors $\delta_0 \, \delta_1 \cdots \delta_{n-1}$ that are themselves generated from available data. This objective can be accomplished by generating the vectors

$$\delta_k = -S_k g_k \tag{3.9}$$

$$x_{k+1} = x_k + \delta_k \tag{3.10}$$

and
$$\gamma_k = g_{k+1} - g_k$$

and then making an additive correction to S_k of the form

$$S_{k+1} = S_k + C_k \tag{3.11}$$

for $k = 1, 2, \ldots, n-1$. If a correction matrix C_k can be found such that the conditions

$$S_{k+1}\gamma_i = \delta_i \qquad for \quad 0 \le i \le k \tag{3.12}$$

are satisfied and the vectors $\delta_0, \delta_1, \cdots, \delta_{n-1}$ and $\gamma_0, \gamma_1, \cdots, \gamma_{n-1}$ generated by this process are linearly independent, then for the case $k = n-1$ we can write

$$S_n [\gamma_0 \ \gamma_1 \cdots \ \gamma_{n-1}] = [\delta_0 \ \delta_1 \cdots \delta_{n-1}]$$

Or

$$S_n = [\gamma_0 \ \gamma_1 \cdots \ \gamma_{n-1}][\delta_0 \ \delta_1 \cdots \delta_{n-1}]^{-1} \tag{3.13}$$

Now from Eqs. (3.11) and (3.16), we have

$$S_n = H^{-1} \tag{3.14}$$

and if $k = n$, Eqs. (3.12) and (3.17) yield the Newton direction

$$\delta_n = -H^{-1} g_n \tag{3.15}$$

2.1.2. INEXACT LINE SEARCHES

Generally speaking, in optimization algorithms, the bigger part of the computational effort is spent preforming line searches. Thus, the measure of calculation needed to fix a problem will in general relies on decisively on the effectiveness and exactness of the line search utilized. If a high-accuracy line search is compulsory in a specific algorithm, a lot of computational effort reducing the objective function can be spent by the algorithm regarding for scalar α. Therefore, low-accuracy or inexact line searches are generally chosen, as long as their utilization does not influence the convergence properties of the algorithm. Quasi-Newton algorithms have been observed to be very tolerant to line-search imprecision. Therefore, inexact line searches are quite often utilized in these algorithms.

Further insights concerning this line search can be found in [8].

2.1.3. Practical Quasi-Newton Algorithm

A practical quasi-Newton algorithm that removes the complication related to the previous Algorithms is described under. The algorithm is supple, effective, and very trustworthy, and it is suitable for designing digital filters [1].

1. (*Initialize algorithm*)

a. *Input* $\mathbf{x}_0$ *and* ε_1.

b. *Set* $k = m = 0$.

c. *Set* $\rho = 0.1,\ \sigma = 0.7,\ \tau = 0.1,\ \chi = 0.75,\ \widehat{M} = 600,\ and\ \varepsilon 2 = 10^{-10}$.

d. *Set* $\mathbf{S}_0 = \mathbf{I}_n$.

e. *Compute* f_0 *and* $\mathbf{g}_0$, *and set* $m = m + 2$. *Set* $f_{00} = f_0$ *and* $\Delta f_0 = f_0$.

2. (*Initialize line search*)

a. *Set* $\mathbf{d}_k = -\mathbf{S}_k \mathbf{g}_k$.

b. *Set* $\alpha_L = 0$ *and* $\alpha_U = 10^{99}$.

c. *Set* $f_L = f_0$ *and compute* $f_L^{'} = \mathbf{g}(\mathbf{x}_k + \alpha_L \mathbf{d}_k)^T \mathbf{d}_k$.

d. (*Estimate* α_0)

If $\left| f_L^{'} \right| > \varepsilon_2$, *then compute* $\alpha_0 = -2\Delta f_0 / f_L^{'}$; *otherwise, set* $\alpha_0 = 1$.

If $\alpha_0 \le 0$ *or* $\alpha_0 > 1$, *then set* $\alpha_0 = 1$.

3. *Set* $\delta_k = \alpha_0 \mathbf{d}_k$ *and compute* $f_0 = f(\mathbf{x}_k + \delta_k)$.

Set $m = m + 1$.

4. (*Interpolation*)

If $f_0 > f_L + \rho(\alpha_0 - \alpha_L) f_L^{'}$ *and* $\left|(f_L - f_0)\right| > \varepsilon_2$ *and* $m < \widehat{M}$, *then do*:

a. *If* $\alpha_0 < \alpha_U$, *then set* $\alpha_U = \alpha_0$.

b. *Compute* $\breve{\alpha}_0$ *using* the *Eq*: $\breve{\alpha}_0 = \alpha_L + \dfrac{(\alpha_0 - \alpha_L) 2 f_L^{'}}{2[f_L - f_0 + (\alpha_0 - \alpha_L) f_L^{'}]}$

c. *Compute* $\breve{\alpha}_{0L} = \alpha_L + \tau(\alpha_U - \alpha_L)$; *if* $\breve{\alpha}_0 < \breve{\alpha}_{0L}$, *then set* $\breve{\alpha}_0 = \breve{\alpha}_{0L}$.

d. *Compute* $\breve{\alpha}_{0U} = \alpha_U - \tau(\alpha_U - \alpha_L)$; *if* $\breve{\alpha}_0 > \breve{\alpha}_{0U}$, *then set* $\breve{\alpha}_0 = \breve{\alpha}_{0U}$.

e. *Set* $\alpha_0 = \breve{\alpha}_0$ *and go to step* 3.

5. *Compute* $f_0^{'} = \mathbf{g}(\mathbf{x}_k + \alpha_0 \mathbf{d}_k)^T \mathbf{d}_k$ *and set* $m = m + 1$.

6. (*Extrapolation*)

If $f_0^{'} < \sigma f_L^{'}$ *and* $\left|(f_L - f_0)\right| > \varepsilon_2$ *and* $m < \widehat{M}$, *then do*:

a. *Compute* $\alpha 0 = (\alpha_0 - \alpha_L) f_0^{'} / (f_L^{'} - f_0^{'})$.

b. *If* $\alpha_0 \le 0$, *then set* $\breve{\alpha}_0 = 2\alpha_0$; *otherwise, set* $\breve{\alpha}_0 = \alpha_0 + \Delta\alpha_0$.

c. *Compute* $\breve{\alpha}_{0U} = \alpha_0 + \chi(\alpha_U - \alpha_0)$; *if* $\breve{\alpha}_0 > \breve{\alpha}_{0U}$, *then set* $\breve{\alpha}_0 = \breve{\alpha}_{0U}$.

d. *Set* $\alpha_L = \alpha_0,\ \alpha_0 = \breve{\alpha}_0,\ f_L = f_0,\ f_L^{'} = f_0^{'}$ *and go to step* 3.

7. (*Check termination criteria and output results*)

a. *Set* $\mathbf{x}_{k+1} = \mathbf{x}_k + \delta_k$.

b. *Set* $\Delta f_0 = f_{00} - f_0$.

c. *If* $\left(\|\delta_k\|_2 < \varepsilon_1 \text{ and } |\Delta f 0| < \varepsilon_1 \right)$ *or* $m \ge \widehat{M}$,

then output $\breve{\mathbf{x}} = \breve{\mathbf{x}}_{k+1},\ f(\breve{\mathbf{x}}) = f_{k+1}$, *and stop.*

d. *Set* $f_{00} = f_0$.

8. (*Prepare for next iteration*)

a. *Compute* $\mathbf{g}_{k+1}$ *and set* $\gamma_k = \mathbf{g}_{k+1} - \mathbf{g}_k$.

b. *Compute* $D = \delta_k^T \gamma_k$; *if* $D \le 0$, *then set* $\mathbf{S}_{k+1} = \mathbf{I}_n$; *otherwise,*

compute $\mathbf{S}_{k+1}$ *using Eq.* (3.19) *or Eq.* (3.20).

c. *Set* $k = k + 1$ *and go to step* 2.

2.2. MINIMAX ALGORITHMS

The design of digital filters can be accomplished by minimizing one of the norms (L1, L2 and L_∞). If the L1 or L2 norm is minimized, then the sum of the magnitudes or the sum of the squares of the basic errors is minimized. The minimum error attained mostly turns out to be unequally distributed compared to the frequency and may present large peaks, which are often objectionable. If prescribed amplitude response specifications are to be met, the magnitude of the largest elemental error should be minimized and, therefore, the L_∞ norm of the error function should be used. Algorithms developed specifically for the minimization of the L_∞ norm are known as minimax algorithms and lead to designs in which the error is uniformly distributed with respect to frequency. The solutions obtained tend to be equiripple, which is, in result, the minimax solution for filters that have piecewise-constant amplitude responses.

The most principal minimax algorithm is the named least-pth algorithm, which includs minimizing an objective function of the form given in Eq. (3.5) for greater values of p, like p = 2, 4, 8, . . . , and is as follows [9].

The implementation of the minimax algorithm generally follows the procedure given by these steps:

Step 1: *Input* $\breve{x}_0$ *and*. *Set* $K = 1,\ p = 2,\ \mu = 2,\ \hat{E}_0 = 10^{99}$.
Step 2: *Initialize frequencies* $\omega_1,\ \omega_2,\ \ldots,\ \omega_K$.
Step 3: *Using* $\breve{x}_{k-1}$ as *initial point, minimize* $\Psi(x)$ *in eq.*3.5,
with respect to x, *to obtain* $\breve{x}_k$ *and set* $\hat{E}_k = \hat{E}(\breve{x}_k)$.
Step 4: *If* $\left|\hat{E}_{k-1} - \hat{E}_k\right| < \varepsilon_1$, *then output* $\breve{x}_k$ *and* $\hat{E}_k$, *and stop.*
Else, set $p = \mu p,\ k = k + 1$ *and go to Step* 3.

The fundamental principle for this algorithm is that the solving of the minimax problem come as result of solving a series of closely connected problems whereby the solution of one makes the solution of the next one more tractable. Parameter μ in step 1, which of course should be an integer, must not be too big in order to prevent the numerical ill-conditioning. A value of 2 was found to give good results.

The minimization in step 3 can be done by the algorithm described in the previous section. The gradient of $\Psi_k(x)$ is given by [9]

$$\nabla\Psi_k\left(x\right)=\left\{\sum_{i=1}^{k}\left[\frac{\left|e_i\left(x\right)\right|}{\hat{E}\left(x\right)}\right]^p\right\}^{(1/p)-1}\sum_{i=1}^{k}\left[\frac{\left|e_i\left(x\right)\right|}{\hat{E}\left(x\right)}\right]^{p-1}\nabla\left|e_i\left(x\right)\right| \tag{3.16}$$

The preceding algorithm works very well, except that it requires a considerable amount of computation.

The minimax algorithms considered will generate filters which possibly will be stable because the obtained transfer function probably has poles outside the unit circle of the z plane. However, this obstacle can be easily overcome by substituting the offending poles by their reciprocals and concurrently adjusting the multiplier constant H_0 so as to compensate for the variation in gain. More details on this stabilization technique can be found in [1].

On the other hand, researchers have developed many design methods based on modern heuristics optimization algorithms (check the introduction), in the next section we will discuss one of those that it related to the proposals of this thesis which is CSA.

3. Crow search algorithm CSA

Crows (crow family or corvids) are considered the smartest birds. They have the biggest brain compared to their body size, which they contain brain a little less than a human brain, based on a brain-to-body ratio. Proofs for the intelligence of crows are widely available. They have proven self-awareness in mirror tests and the ability of having tool-making. Crows can memorize faces and alert each other the moment a stranger one comes near. Furthermore, they can use tools, communicate in sophisticated ways and remember where they hiding their foods up to many months later [10–12].

Crows were known to observe other birds, to know where they hide their food, and steal it the moment the bird leaves. If a crow commits a pilfering, it will take extra precautions like changing its hiding location to prevent being a victim in the future. Moreover, crows employ their own experience of being thieves to predict the behavior of a thief, and they can find the securest way to keep their caches from being stolen [13].

In this section, depending on the previous-mentioned intelligent behaviors, a population-based metaheuristic algorithm, CSA, is set. The main attitudes of CSA are scheduled as:

- Crows exist in the nature of group.
- Crows remember the location of their hiding sites.

- Crows pursue each other to make stealing.
- Crows keep their reserves from being stolen by a probability.

It is implicit that present is a d-dimensional situation counting a quantity of crows. The number of crows (flock size) is N, and the position of crow i at time (iteration) $iter$ in the search space is specified by a vector:

$$x^{i,iter}\left(i=1,2,\ldots,N;iter=1,2,\ldots,iter_{max}\right)\ \ where\ x^{i,iter}=\left[x_1^{i,iter},x_2^{i,iter},\ldots,x_d^{i,iter}\right]$$

Where $iter_{max}$ denoted the maximum number of iterations. Every crow characterized by a memory wherein the hiding place position of the crow is saved. At iteration $iter$ n, the position of hiding place of crow i is shown by $m^{i,iter}$. This is the best position that crow i have achieved so far. actually, in remembrance of every crow, the location of its finest skill has been stored. Crows travel in the surroundings and look for improved provisions supply (hiding sites). Suppose that at iteration $iter$, crow j decided visiting its hiding place, $m^{j,iter}$. On this iteration, crow i make a decision to pursue crow j to come near the hiding position of crow j. In this situation, two possibilities can occur:

Situation 1: Crow j does not recognize that crow i is following it. As a consequence, crow i will move toward to the hiding position of crow j. In this case, the latest location of crow i is achieved by:

$$x^{i,iter+1}=x^{i,iter}+r_i\,fl^{i,iter}\left(m^{j,iter}-x^{i,iter}\right) \qquad (3.17)$$

where r_i is a random number with unvarying distribution between 0 and 1 and $fl^{i,iter}$ designate the travel span of crow i at iteration $iter$. Figure 3.1 illustrates the diagram of this state and the effect of $fl^{i,iter}$ on the search capability. Little values of $fl^{i,iter}$ guide to neighboring seek (at the surrounding area of $\mathrm{x}^{\mathrm{i,iter}}$), and big value outcomes in universal seek (distant from $\mathrm{x}^{\mathrm{i,iter}}$). As Figure 3.1A shows, if the value of $fl^{i,iter}$ is selected less than 1, the next position of crow i is on the dash line between $\mathrm{x}^{\mathrm{i,iter}}$ and $\mathrm{m}^{\mathrm{i,iter}}$. As Figure3.1B indicates, if the value of $fl^{i,iter}$ is selected more than 1, the next position of crow i is on the dash line which may exceed $\mathrm{m}^{\mathrm{i,iter}}$.

Situation 2: Crow j recognizes that crow i is following it. As a consequence, in order to keep its store from being stolen, crow *j* will trick crow i by going away to a different location of the search space. Totally, states 1 and 2 can be expressed as follows:

$$\begin{cases} x^{i,iter+1} = x^{i,iter} + r_i\, fl^{i,iter}\left(m^{j,iter} - x^{i,iter}\right) & \text{for} \quad r_j \geq AP^{j,iter} \\ \quad random\ position & o.w \end{cases} \tag{3.18}$$

where r_j is a random number with uniform distribution between 0 and 1 and $AP^{j,iter}$ denotes the knowledge probability of crow j at iteration $iter$.

Metaheuristic algorithms must offer a suitable equilibrium between diversification and intensification [14]. In CSA, intensification and diversification are principally controlled by the awareness probability (*AP*) parameter. Reducing in the value of the awareness probability, CSA inclines to conduct the search on a close area where an existing best solution can be achieved in this area. Consequently, using small values of *AP* make the intensification increase. The opposite of that, increasing the value of the awareness probability, the probability of searching the vicinity of existing best solution decreases and CSA inclines to discover the search space on a global scale (randomization). Consequently, using large values of *AP* make the diversification increase.

The main reasons of using CSA are it’s easy to implement, few parameters to adjust, fast convergence speed, and high efficiency, which in turn makes it very attractive for applications in different engineering areas.[15]

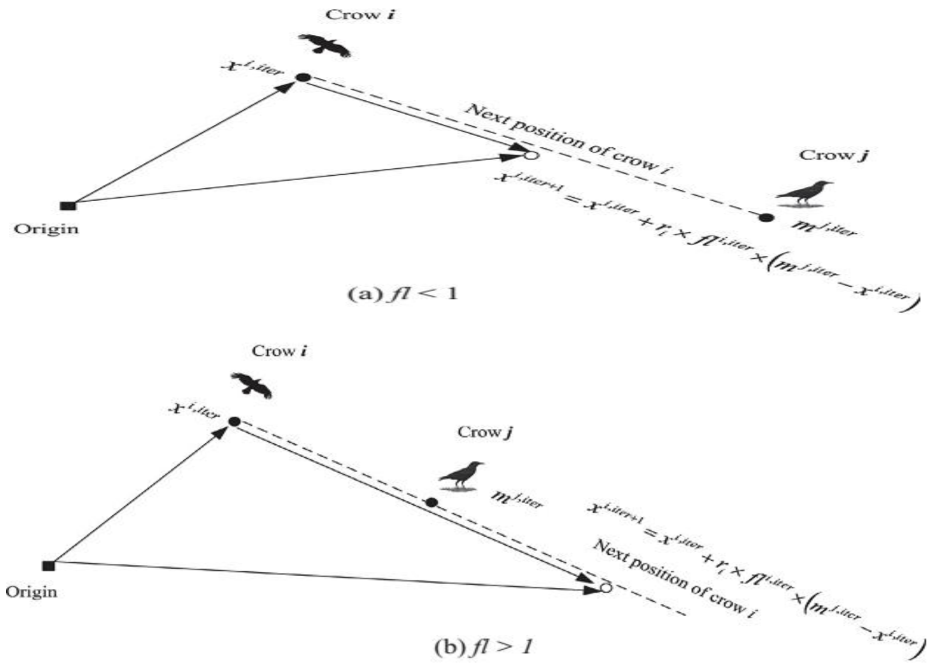


Fig. 3.1 Diagram for the two situations of the CSA (a) fl < 1 and (b) fl > 1. [15]

3.1.1 CSA implementation for optimization

The step-wise procedure for the implementation of CSA is given in this section.

1. **Initialize problem and adaptable factors:**

   The optimization problem, assessment variables, and constraints are identified. After that, the adaptable decisions of CSA (flock size N), maximum number of iterations $(iter_{max})$, flight length (fl), and awareness probability (AP) are valued.

2. **Initialize location and memory of crows:**

   N crows are randomly located in a d-dimensional seek room as the elements of the group. Every crow designates a practical solution of the problem, and d is the number of assessment variables.

$$Crows = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_d^1 \\ x_1^2 & x_2^2 & \dots & x_d^2 \\ \vdots & \vdots & & \vdots \\ x_1^N & x_2^N & \dots & x_d^N \end{bmatrix}$$

The memory of every crow is initialized. Given that at the beginning iteration, the crows have no knowledge, it is assumed that they have buried their provisions at their original locations.

$$Memory = \begin{bmatrix} m_1^1 & m_2^1 & \dots & m_d^1 \\ m_1^2 & \mathrm{m}_2^2 & \dots & m_d^2 \\ \vdots & \vdots & & \vdots \\ m_1^N & m_2^N & \dots & m_d^N \end{bmatrix}$$

3. **Calculate fitness (objective) function:**

   For every crow, the superiority of its location is calculated by introducing the assessment variable values into the objective function.

4. **Create new location:**

   Crows create new location in the seek room: assume crow i wishes to create a new location. For this aim, this crow randomly selects one of the flock crows and follows it to discover the position of the foods hidden by this crow m^j. The new location of crow i is obtained by the preceding position equation eq(3.24). This procedure is replicated for the entire crows.

5. **Verify the viability of new locations:**

   The viability of the new location of every crow is verified. If the new location of a crow is possible, the crow revises its location. If not, the crow continues in the present location and does not travel to the produced new location.

6. **Calculate fitness function of new locations:**

   The fitness function cost for the new location of every crow is calculated.

7. **Update memory:**

   The crows bring up to date their memory as:

$$m^{i,iter+1} = \begin{cases} x^{i,iter+1} & f\left(x^{i,iter+1}\right) \text{ is better than } f\left(m^{i,iter}\right) \\ m^{i,iter} & o.w \end{cases} \tag{3.19}$$

where $f(\cdot)$ denotes the objective function value.

It is observed that if the fitness function cost of the new location of a crow is superior than the fitness function value of the memorized location, the crow revises its memory by the new position.

8. **Check termination criterion:**

   Steps 4 to 5 are repeated until $iter_{max}$ is reached. When the termination condition is reached, the most excellent location of the memory in terms of the objective function cost is considered as the solution of the optimization problem.

Pseudo code of CSA is shown below:

$Randomly\ initialize\ the\ position\ of\ a\ flock\ of\ N\ crows\ in\ the\ search\ space$
$Evaluate\ the\ position\ of\ the\ crows$
$Initialize\ the\ memory\ of\ each\ crow$
$while\ iter < iter_{max}$
$\quad for\ i = 1 : N(all\ N\ crows\ of\ the\ flock)$
$\quad\quad Randomly\ choose\ one\ of\ the\ crows\ to\ follow(for\ example\ j)$
$\quad\quad Define\ an\ awareness\ probability$
$\quad\quad\quad if\ \mathrm{r}_j \geq AP^{j,iter}$
$\quad\quad\quad\quad x^{i,iter+1} = x^{i,iter} + r_i \times fl^{i,iter} \times \left(m^{j,iter} - x^{i,iter}\right)$
$\quad\quad\quad else$
$\quad\quad\quad\quad x^{i,iter+1} = a\ random\ position\ of\ search\ space$
$\quad\quad\quad end\ if$
$\quad end\ for$
$Check\ the\ feasibility\ of\ new\ positions$
$Evaluate\ the\ new\ position\ of\ the\ crows$
$Update\ the\ memory\ of\ crows$
$end\ while$

3.1. Comparison of CSA with GA, PSO and HS

The CSA, compared to the other famous algorithms such as GA, PSO and HS, uses a population of searchers to discover the search space. The probability of obtaining a best solution and get away from local optima increase through the use of a population. Besides to the flock size and maximum number of iterations, other parameters must to adjusted are present in optimization algorithms. The time-consuming work or the parameter setting is one of the most problems in optimization algorithms. As a result, as long as the number of these parameters that need to adjust is small, the algorithms are easy to implement. In CSA, just a two (2) parameters need to be adjusted, which are the flight length and awareness probability. While, In PSO algorithm four (4) parameters need to adjust: the inertia weight, maximum value of velocity, individual learning factor and social learning factor; HS necessitates three (3) parameters: the value of harmony memory considering rate, pitch adjusting rate and bandwidth of generation; In GA six (6) parameters must be determined: selection method, crossover method, crossover probability, mutation method, mutation probability and replacement method.

Moreover, CSA, like GA and PSO, is not a greedy algorithm because if a crow finds a new position which is not good than its current position, it will change its position to the new one. While the diversity of generated solutions can be increased in the non-greedy algorithms. In HS, a novel solution can't be refused if its fitness value is greater than the fitness of the bad harmony of memory.

In addition, Like HS and PSO, CSA contains memory in which best solutions are saved. In PSO, every particle is attracted to the best position ever generated by itself and the best position ever generated by the ensemble. Consequently, at every iteration, the best solutions obtained so far are directly used. While in CSA, at every iteration, every crow chooses randomly one of the flock crows (it maybe itself) and travels to its hiding place (the best solution generated by that crow). This means that, in CSA, at every iteration, the best positions obtained so far are directly used to find better positions.

4. Conclusion

This chapter divided into two principal sections. In the first section, we present the formulas of the gradient vector and the Hessian matrix, these formulas are then used in the Newton algorithm, in order to compute the gradient vector, only even power p is considered. Basics of the quasi-Newton and minimax algorithms were presented.

In the second section, a new metaheuristic algorithm, named CSA, is presented. CSA is population-based optimization algorithm which is quite simple with only two parameters to adjust, which in turn makes it very attractive for applications in different engineering fields. Compared with GA, PSO and HS, CSA has smaller number of adjustable parameters and for this, it is very asy to implement.

References

[1] A. Antoniou, Digital Signal Processing: Signals, Systems, and Filters. New York: McGraw-Hill, 2005.

[2] K. Steiglitz, “Computer-aided design of recursive digital filters,” IEEE Trans. Audio Electroacoust., vol. 18, pp. 123–129, June 1970.

[3] A. G. Deczky, “Synthesis of recursive digital filters using the minimum p-error criterion,” IEEE Trans. Audio Electroacoust., vol. 20, pp. 257–263, Oct. 1972.

[4] J. W. Bandler and B. L. Bardakjian, “Least pth optimization of recursive digital filters,” IEEE Trans. Audio Electroacoust., vol. 21, pp. 460–470, Oct. 1973.

[5] C. Charalambous, “Minimax design of recursive digital filters,” Computer Aided Design, vol. 6, pp. 73–81, Apr. 1974.

[6] C. Charalambous, “Minimax optimization of recursive digital filters using recent minimax results,” IEEE Trans. Acoust., Speech, Signal Process., vol. 23, pp. 333–345, Aug. 1975.

[7] Ghibeche Y, Saadi S, Hafaifa A. Optimal design of IIR filters based on least ℓp-norm using a novel meta-heuristic algorithm. Int J Numer Model. 2018;e2480. https://doi.org/10.1002/jnm.2480.

[8] R. Fletcher, Practical Methods of Optimization, Unconstrained Optimization, vol. 1, New York: Wiley, 1980. (See also R. Fletcher, Practical Methods of Optimization, 2nd ed., New York: Wiley, 1990.)

[9] C. Charalambous, “A unified review of optimization,” IEEE Trans. Microwave Theory and Techniques, vol. MTT-22, pp. 289–300, Mar. 1974.

[10] Rincon, Paul, Science/nature|crows and jays top bird IQ scale, BBC News.

[11] Prior H, Schwarz A, Güntürkün O. Mirror-induced behavior in the magpie (pica pica): evidence of self-recognition. PLoS Biol 2008;6(8):e202.

[12] <https://en.wikipedia.org/wiki/Hooded_crow>.

[13] Clayton N, Emery N. Corvide cognition. Curr Biol 2005;15:R80–1.

[14] Yang XS. Metaheuristic optimization. Scholarpedia 2011;6. 11472.

[15] Askarzadeh A. A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm. Comput Struct. 2016;169:1-12. https://doi.org/10.1016/j.compstruc.2016.03.001

Chapter 4

RESULTS AND DISCUSSIONS

1. Introduction

The aim of this chapter is the applying of novel meta-heuristic algorithms, which described in previous chapter, for designing an IIR filter. This chapter has two main parts. In the first part, a low pass and high pass IIR digital filters design examples are take into account to explore the performance of filter designed with the CSA method, and to study the performance of the proposed method, a comparison was made with the Butterworth method. The other section highlights the optimal implementation of high-pass and band-stop IIR filters on the FPGA, the performance of the considered two types of filters has been experienced for noise elimination from ECG signal to verify the applicability of this optimal approach.

Consider the IIR filter with the input-output relationship governed by:

$$y(k)=\sum_{i=1}^{N}a_i.x(k-i)-\sum_{i=1}^{M}b_i.y(k-i) \tag{4.1}$$

where x and y represent the input and the output signal, respectively.

The transfer function of the filter can be expressed as::

$$H(z)=\frac{B(z)}{A(z)}=\frac{\sum_{i=0}^{L}a_i.Z^{-i}}{1+\sum_{i=1}^{M}b_i.Z^{-i}} \tag{4.2}$$

Where a_i and b_i denote the coefficients of the filter, a significant task in the filter designing is to find their values. So, this filter design can be considered as an optimization problem of cost function $J(\omega)$ stated as $\min J(\omega)$ where $\omega=[a_0,\ a_1,\ a_2,\ldots,\ a_L,\ b_1,\ b_2,\ \ldots,\ b_M]$ is vector of filter coefficient. The aim is to minimize the objective function $J(\omega)$ with respect to the vector coefficients ω.

Usually, the cost function can be expressed by

$$J(\omega)=\frac{1}{N}\sum_{k=1}^{N}\left(d(k)-y(k)\right)^2 \tag{4.3}$$

where $d(k)$ and $y(k)$ are the desired and actual responses of the filter, respectively, and N is the number of samples exploited for the computation of cost function.

2. Using CSA and ℓ p- norm to solve the optimization problem

The distance approach refers to an estimate problem for which different norms could be selected, such as the ℓp-norms with p = 2. Solutions are reached using CSA instead of standard Newton's method. Assume that the amplitude response of a recursive filter is required to approach some specified amplitude response as closely as possible.

Such a filter can be designed, in general, by following these two steps:

- An objective function is formulated, which is basing on the difference between the actual and desired amplitude response.
- The obtained objective function is then minimized, with taking into account the transfer-function coefficients.

An objective function fulfilling different essential requirements can be expressed in the following:

$$\Psi(x) = \left[\sum_{i=1}^{K} \left| e_i(x) \right|^P \right]^{1/P} \tag{4.4}$$

where p is an integer.

The required design can be obtained by solving the optimization problem

$$\underset{x}{\text{minimize}} \ \Psi(x)$$

This optimization problem can be minimized using CSA in which an optimal solution for the objective functions can be achieved. The implementation of the crow search algorithm follows the next procedure steps:

$\textbf{Step 1}$: *Input* $\breve{x}_0$ *and. Set* $K = 1,\ p = 2,\ \mu = 2,\ \hat{E}_0 = 10^{99}$.
$\textbf{Step 2}$: *Initialize frequencies* $\omega_1, \omega_2, \ldots, \omega_K$. *Which means,*
initialize the real coded particles (x) *of the np population;*
each consists of an equal number of numerator and
denominator filter coefficients bk and ak, respectively;
the total coefficients $D = (ord + 1)\backslash 2$ *for designed filter with order ord;*
$\textbf{Step 3}$: *Using* $\breve{x}_{k-1}$ as *initial point, minimize* $\Psi(x)$ *in eq.4.4,*
with respect to x, *to obtain* $\breve{x}_k$ *and set* $\hat{E}_k = \hat{E}(\breve{x}_k)$.
$\textbf{Step 4}$: *If* $\left|\hat{E}_{k-1} - \hat{E}_k\right| < \varepsilon_1$, *then output* $\breve{x}_k$ *and* $\hat{E}_k$, *and stop.*
Else, set $p = \mu p,\ k = k + 1$ *and go to Step* 3.

The minimization can be done using the CSA in Step 3 based on the implemented pseudo code illustrated in previous chapter.

The parameters for the CSA code utilized are listed in the following:

- Number of runs: $iter_{max}$ = 500;
- Problem dimension: Pd = 1;
- Flight length: fl = 50;
- Awareness probability: AP = 0.1.

3. Simulation results

In this part, two types of the IIR digital filter which are low pass and high pass filters are considered as a design examples to examine the functioning of filters designed with the proposed method. To compare the performance of this method, a classical Butterworth method is used, and the results are obtained through simulations. This work is carried out on an Intel Core I7, 2.40 GHz CPU with RAM equal to 6 GB. The simulations are done using MATLAB programming language to demonstrate the potentiality of proposed method for the design of IIR digital filters. Computation complexity and run time depend on the filter order are differ from some minutes to tens of minutes. To defend the properties of this method, four (04) examples are made. The design parameters for simulation utilized for all examples are listed in the following:

- ✓ Minimum and maximum values of the coefficients are: − 2 and +2, respectively;
- ✓ Number of samples: 200;
- ✓ The sampling frequency is taken to be fs = 1 Hz;
- ✓ Pass band ripple: δp = 0.01;

- ✓ Stop band ripple: δs = 0.0 01;

Example 1: (High pass filter) in this example, the design of 20th-order high pass digital filter is considered with the following IIR design specification: pass band border frequency ωp = 0.4 π, stop band border frequency ωp = 0.3 π.

A comparison between CSA optimal design and butter synthesis approach in magnitude and phase responses of the high-pass filter with order ord = 20 is illustrated below in Figure 4.1.

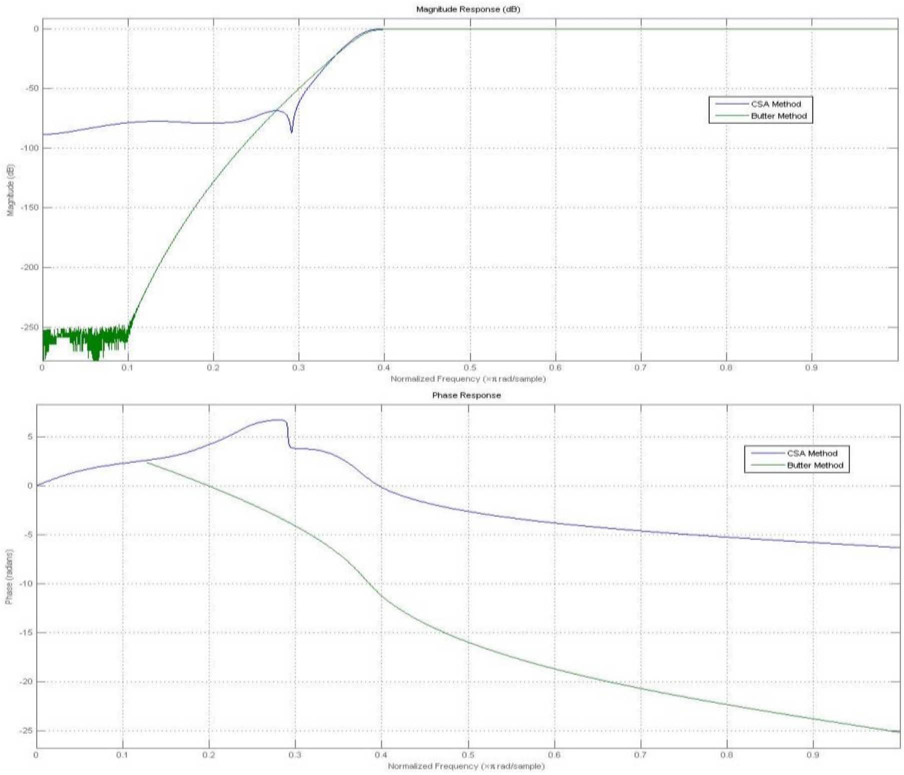


Fig. 4.1 Comparison between the proposed method and butter approach in magnitude and phase responses of the high-pass filter with ord = 20

Example 2: (High pass filter) in this example, the design of 8th-order high pass digital filter is considered with the same specification as specified above. A comparison in the magnitude and phase responses of this high-pass filter with order ord = 8 is presented in the next figure (Figure 4.2), between the CSA optimal design and butter synthesis.

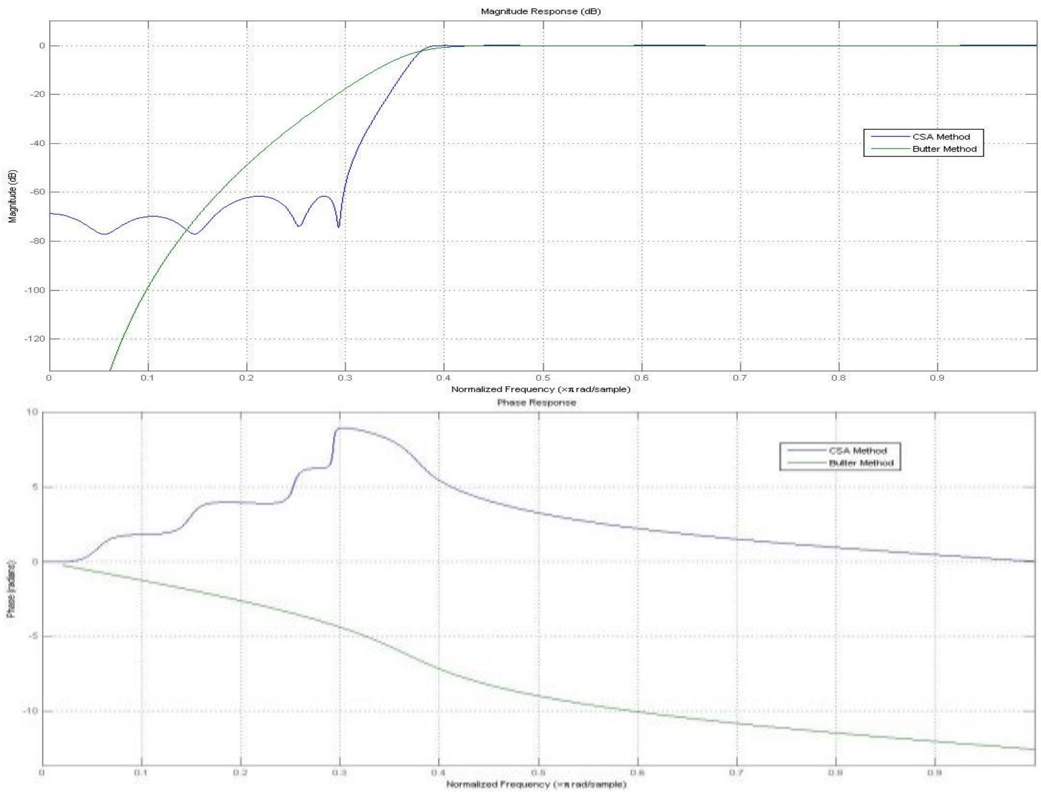


Fig. 4.2 Comparison between the proposed method and butter approach in magnitude and phase responses of the high-pass filter with ord = 8

The results show clearly that the magnitude responses of the high pass filter, for the two orders, obtained from the optimal method have a minimum transition band compared to those using the butter method.

The pole-zero behavior of the high-pass filter has been summarized, it can be seen that the pole-zeros location of the designed filter falls within the unit circle. This demonstrates that the designed filter is stable employing the proposed design method. Figure 4.3 presents a comparison between CSA optimal design and butter synthesis approach in pole-zero positions of the high-pass filters.

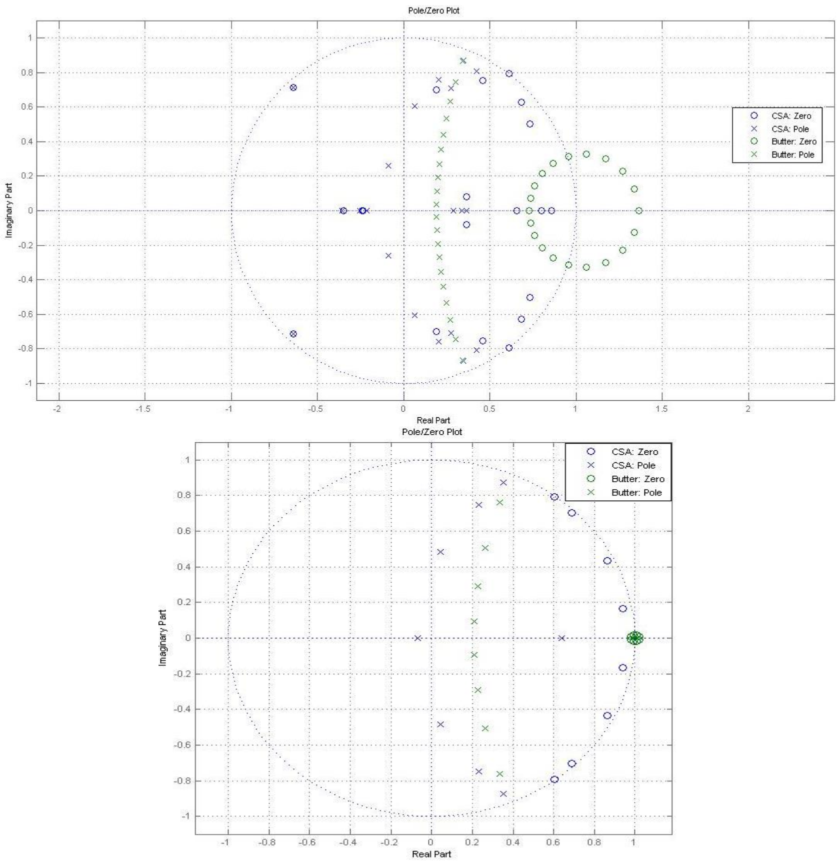


Fig. 4.3 Comparison between the proposed method and butter approach in pole-zero positions of the high-pass filters: A, ord = 20; B, ord = 8

Example 3: (Low pass filter) in this example, the design of 20^{th}-order low pass digital filter is considered with the following IIR design specification: passband edge frequency ωp = 0.5 π, stopband edge frequency ωp = 0.6 π. The obtained magnitude and phase responses of this example using CSA optimal design compared with those obtained using the butter synthesis approach are presented in Figure 4.4.

Example 4: (Low pass filter) in this example, the design of 8^{th}-order low pass digital filter is considered with the same specification as specified in example 3. The obtained magnitude and phase responses of this 8th-order low pass filter using the CSA optimal design and butter synthesis approach are illustrated in Figure 4.5.

You can see that the obtained magnitude responses, for both types of filters, using the CSA optimal method have a minimum transition band compared to those obtained using the butter approach.

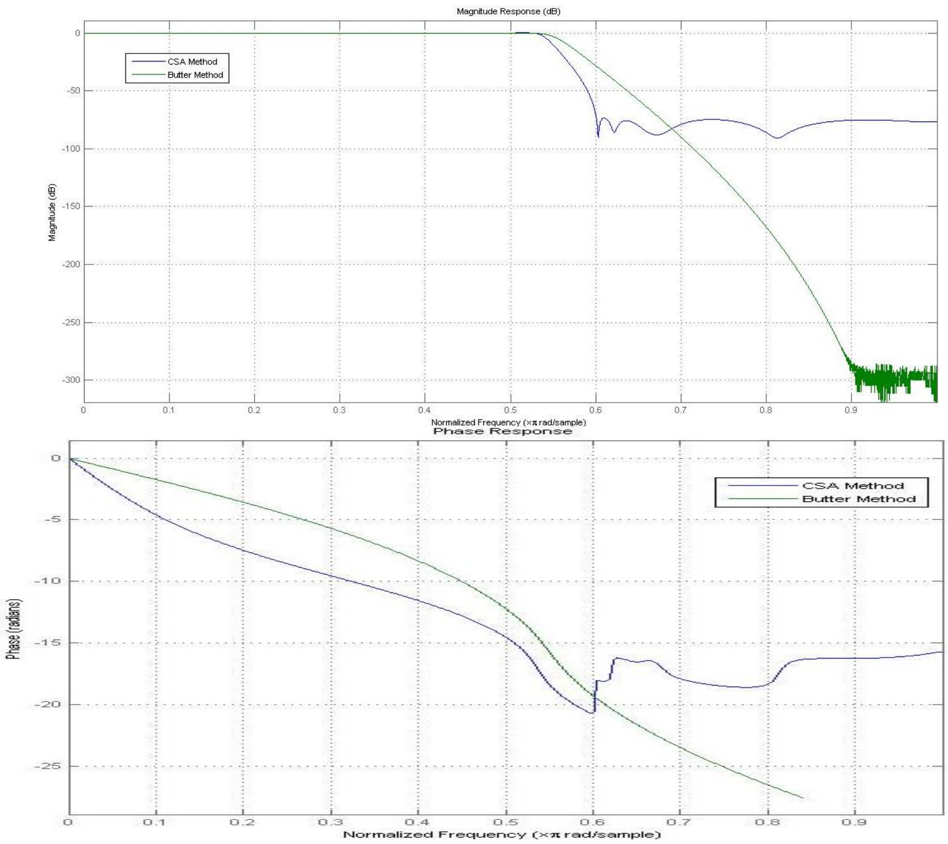


Fig. 4.4 Comparison between the proposed method and butter approach in magnitude and phase responses of the low-pass filter with ord = 20

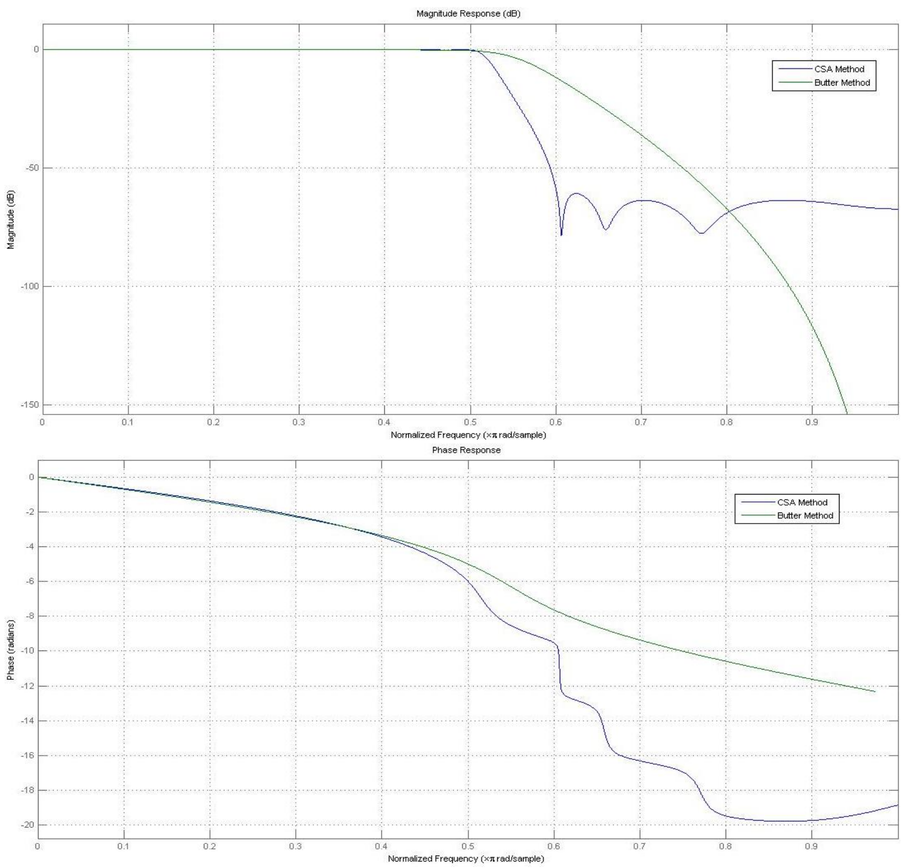


Fig. 4.5 Comparison between the proposed method and butter approach in magnitude and phase responses of the low-pass filter with ord = 8

The pole-zero behavior of the low-pass filter has been summarized. It can be seen that for the 8th order, the pole-zero location of the designed filter falls within the unit circle; this shows that the designed filter is stable. For the 20th, we can say the same just for few zeros that located outside the unit circle, this can't effect on the stability of the filter; it is also stable using the proposed method. Figure 4.6 presents a comparison between CSA optimal design and butter synthesis approach in pole-zero positions of the low-pass filters. Table 1 reviews the numerical values of both high pass and low pass filters coefficients obtained with optimal design for order 8 and order 20.

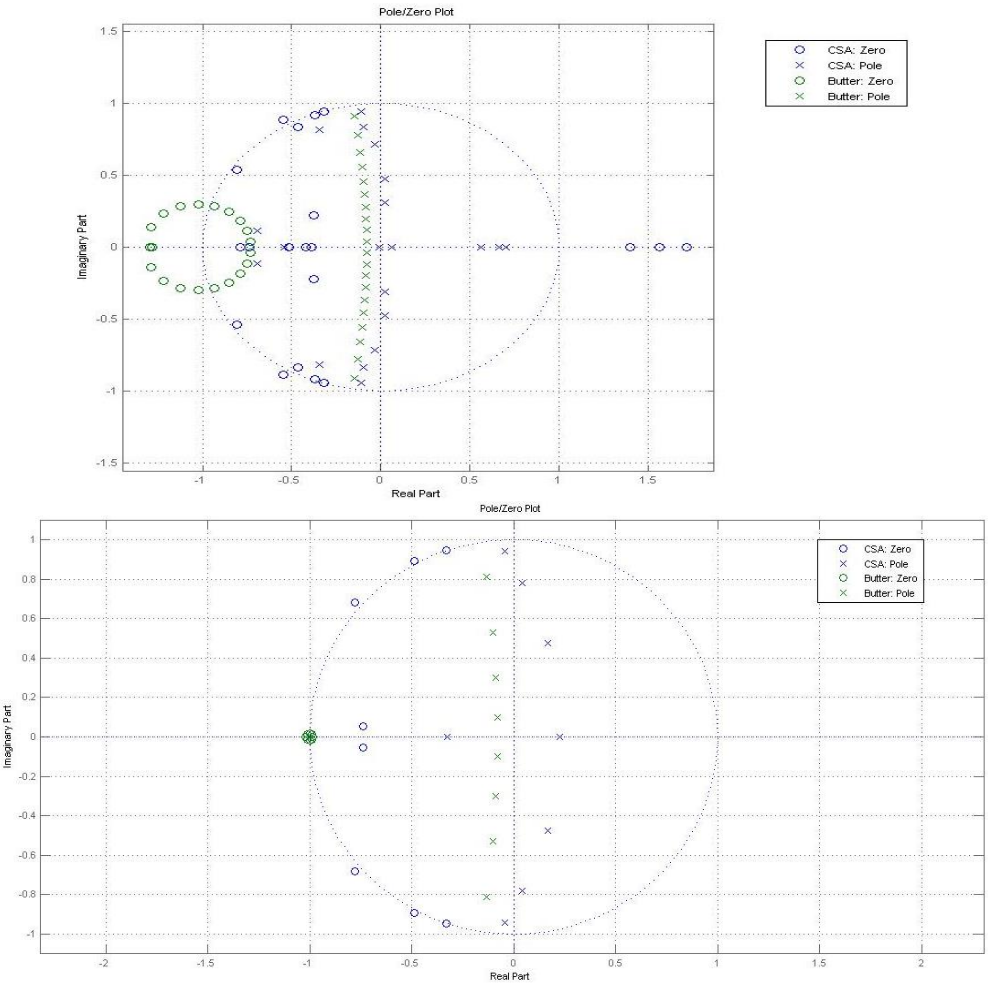


Fig. 4.6 Comparison between the proposed method and butter approach in pole-zero positions of the low-pass filters: A, ord = 20; B, ord = 8

From the above results, we can say that using the proposed method allows to achieve a minimum transition band compared with those obtained using the butter design approach. Another thing tack into account, which is the stability of the filter, for all the 4th filters, they are stable since all the poles are located inside the unit circle.

Figure 4.7 shows the evolution of the objective function with iterations, and we see its rapid convergence.

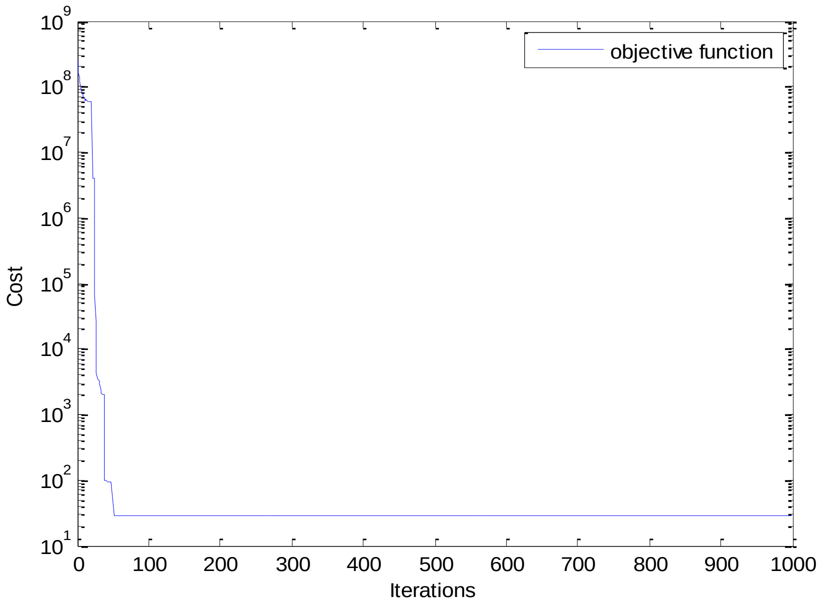


Fig. 4.7 The objective function behavior with CSA iteration

Tab. 4.1 The obtained coefficients for both high and low pass filters using the proposed method

HIGH-PASS				LOW-PASS			
Order 20		Order 8		Order 20		Order 8	
Numerator	Denominator	Numerator	Denominator	Numerator	Denominator	Numerator	Denominator
0.0513185840	1	0.0732737641	1	-0.0052432750	1	0.0324720954	1
-0.3215860998	-1.3003821702	-0.4538806651	-1.8313497037	-0.0206145480	1.0055928369	0.1510802856	-0.2367936014
0.9901772956	3.1440709324	1.3228629106	2.8445676377	-0.0272409115	2.2286310051	0.3734199892	1.6383031079
-1.9579885479	-2.3301871321	-2.3548888776	-2.4653661378	0.0310335580	1.0955670218	0.6012956736	-0.3265282912
2.7874341181	3.6936406523	2.7905824825	1.6862730561	0.1808034982	0.5911175896	0.6819043233	0.7495552896
-3.1163313435	-2.3321894871	-2.2507344319	-0.7879915194	0.3048339229	-0.5411982328	0.5524849613	-0.0579551612
3.0326093213	3.2827730021	1.2069152785	0.2224536900	0.1603450845	-1.1404961434	0.3119978734	0.0514390714
-2.8491699287	-2.0056576373	-0.3941362009	-0.0618363198	-0.3637984534	-0.6679572307	0.1128018241	0.0271885405
2.6252357984	1.8480111579	0.0602264887	-0.0055582463	-0.9875451096	-0.4648262852	0.0196140888	-0.0101546706
-2.1609047388	-0.7253139196			-1.1639771058	0.0842418699		
1.4046056668	0.3067894368			-0.5803585436	0.1852963373		
-0.6186059172	0.0097759461			0.4847112114	0.1108733827		
0.1052070159	-0.0562453117			1.3779023268	0.0886182072		
0.0734743095	0.0232755434			1.6364492831	-0.0067660205		
-0.0630654080	-0.0097223700			1.3063224576	-0.0046555197		
0.0185610966	-0.0015237594			0.7615944073	-0.0053818797		
0.0008764245	0.0005720475			0.3281170278	-0.0048975287		
-0.0020867167	0.0001448390			0.1021768945	-0.0000242212		
0.0003796674	0.0000752190			0.0217768679	-0.0003926854		
0.0000490827	-0.0000096471			0.0028412781	0.0000228679		
-0.0000157742	-0.0000044339			0.0001709539	0.0000002371		

Based on the previous obtained results, we can conclude several advantages of using the CSA for the IIR filter design, which are listed in the following:

- CSA is quite simple that has just two parameters (flight length and AP) to adjust, which in turn makes it very attractive for different application that need both flexibility and precise results.
- Easy to implement compared to the other meta-heuristic optimization algorithms mentioned in literature, since it requires small number of adjustable parameters.
- Depending on the comparison that made above, CSA reaches the best stop band attenuation and minimum transition band compared with those obtained using the butter design approach. In addition, it provides the best magnitude response compared with butter design method.
- The identical length filter is efficiently designed with 85% and 80% less time than classical approach, and the resultant filters can be implemented in hardware for different applications with a significant amount of accuracy.
- It keeps computational time in addition to cost, and thus it is a further efficient and elegant approach of optimization for the design of IIR filters. Finds the solution of the investigated problems in around few seconds (less execution time).
- It has a fast convergence speed.

The next section presents other results, this time it is for the optimal implementation of high-pass and band-stop IIR filters on the FPGA, with the help of the xilinx system generator and the xilinx ISE (Integrated Synthesis Environment) based on the below mentioned steps; the performance of the obtained results, for both high-pass and band-stop filters, are then studied for noise elimination from ECG signals.

3.1. The FPGA implementation

3.1.1. Filters general implementation steps

Step 1: Design the filter using FDA Tool;

Step 2: Identify the Filters Optimal Coefficients using CSA;

Step 3: Create Simulink Model;

Step 4: Complete the simulation model using Xilinx basic elements (Xilinx system generator block is compulsory);

Step 4: Code generation using SysGen;

Step 5: Simulate and debug the logic program with the help of Xilinx ISE;

Step 7: Get Detail summary report of the device utilization, Time and power analysis and generate RTL schematic of the Designed filter.

3.1.2. Filter Design Data

The digital filter information is provided under in tabular form, the table 4.2 explains the detail information of filters used for design, and while table 4.3 illustrates real implementation cost in terms of number of components such as multipliers and adders used, table 4.4 shows filters specifications used during implementation of the filter where: (Units: [Hz] for Fs, Fstop and Fpass / [dB] for Apass and Astop):

- Fs is the sampling frequency;
- F-stop is the stop-band frequency;
- F-pass is the pass-band frequency;
- A-pass is the pass-band attenuation;
- A-stop is the stop-band attenuation.

Tab. 4.2 Filter Design Data

Filter Structure	Direct-Form II
Number of Sections	2
Filter Stability	Yes
Linear Phase	No
Design algorithm	Least Pth-norm

Tab. 4.3 Filter Implementation Cost

Number of Multipliers	9
Number of Adders	8
Number of States	4
Multiplications per Input Sample	9
Additions per Input Sample	8

Tab. 4.4 IIR Filter Specifications

Filter type	High pass	Bandpass
Filter order	4	
Frequency Specification	Fs: 360 Fpass: 1.5 Fstop: 1.4	Fs: 360 Fstop1: 49 Fstop2 : 121 Fpass1: 50 Fpass2: 120
Magnitude Specification	Astop: 60	Astop1 : 60 Astop2 : 60 Apass: 1

3.1.3. Filters Hardware Realization

The IIR filter can be implemented using the Direct Form II structures. The System is modeled in the Simulink, the figure 4.8 shows design of the high pass filter using FDA Tool whereas figure 4.9 shows realization model of the filter.

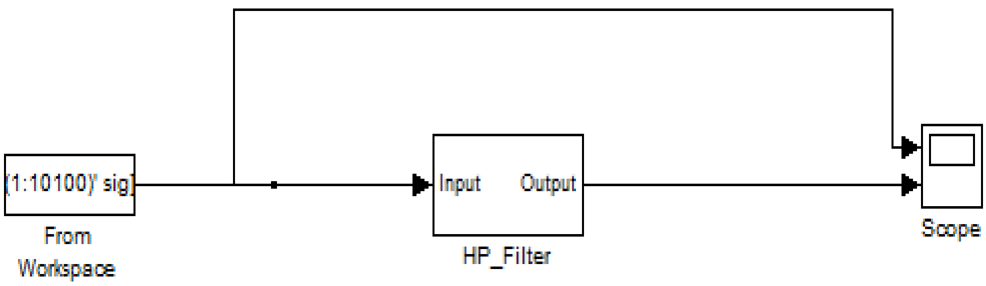


Fig. 4.8 Design of IIR High pass filter using FDAtool

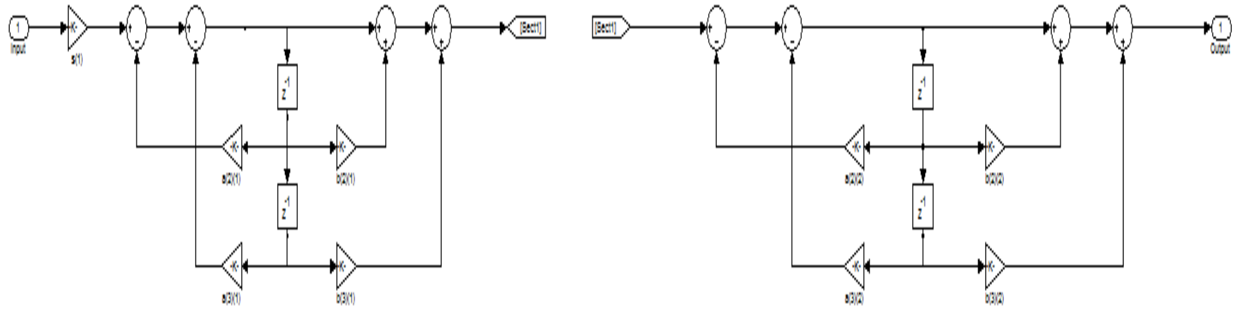


Fig. 4.9 Realization model of IIR High pass filter using

Following the same procedure, band-stop filter is equally generated based on its specifications.

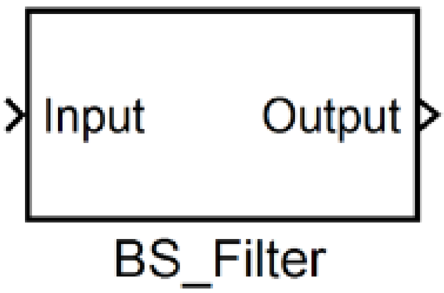


Fig. 4.10 Realization model of IIR Band-stop filter

3.1.4. FPGA implementation using Xilinx System generator (XST)

After designing the filters based on their specifications from Matlab, System Generator is then used for the appropriate IIR FPGA filter implementation for high-pass and band-stop filter as shown in Figures 4.11,12.

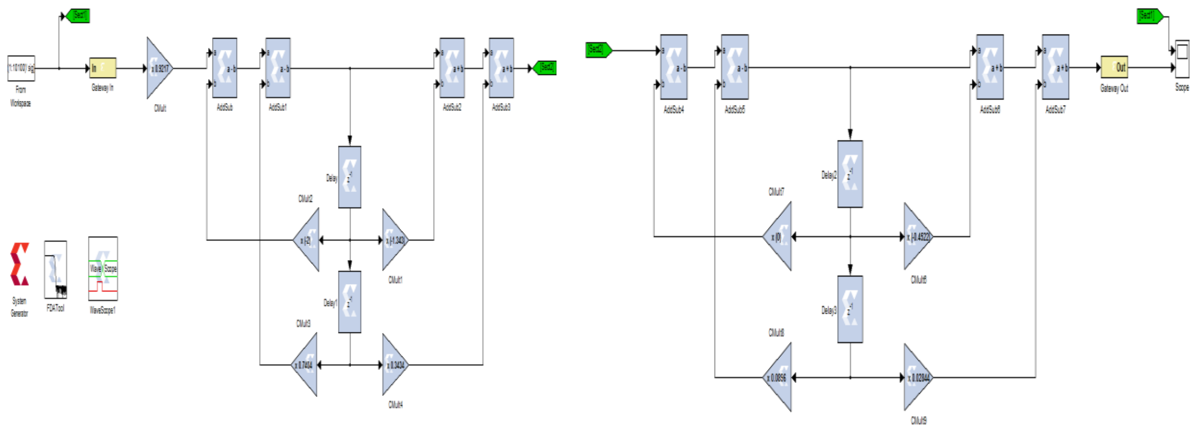


Fig. 4.11 High-pass IIR filter

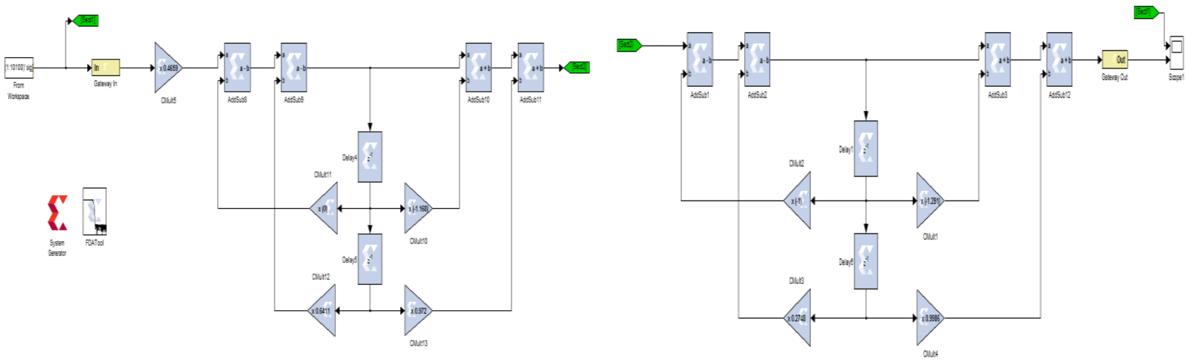


Fig. 4.12 Band-stop IIR filter

In the process of implementing the IIR filters on FPGA, the different architectures described in the previous section which were designed in Simulink, are here implemented on FPGA circuit. Subsequently, a project file was imported from Simulink which was made using system generator block. This file opens as a project file in Xilinx ISE 14.7 software containing VHDL code for various blocks which are required to provide the necessary functionality.

- **RTL Schematic**

The actual high level RTL schematic and low level RTL schematic for the High pass and band stop filters are shown in the figures 4.13,14 below:

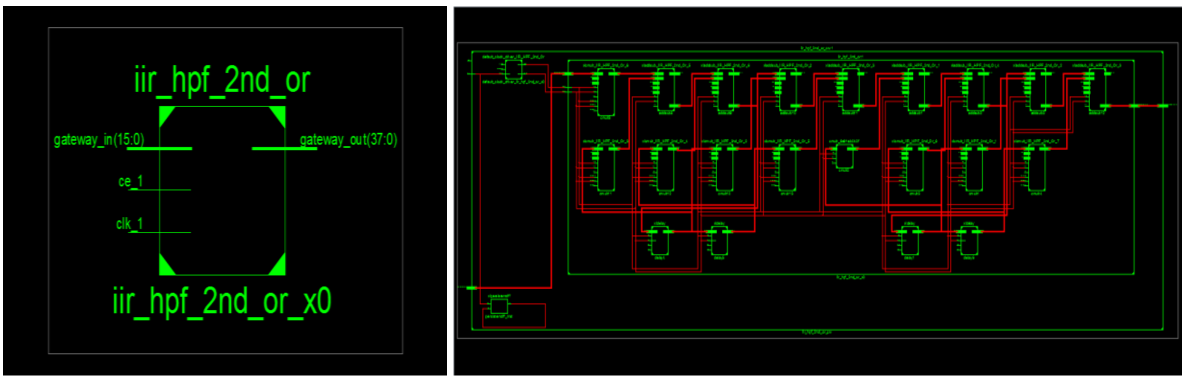


Fig. 4.13 Low and High level RTL schematic of the HPF model.

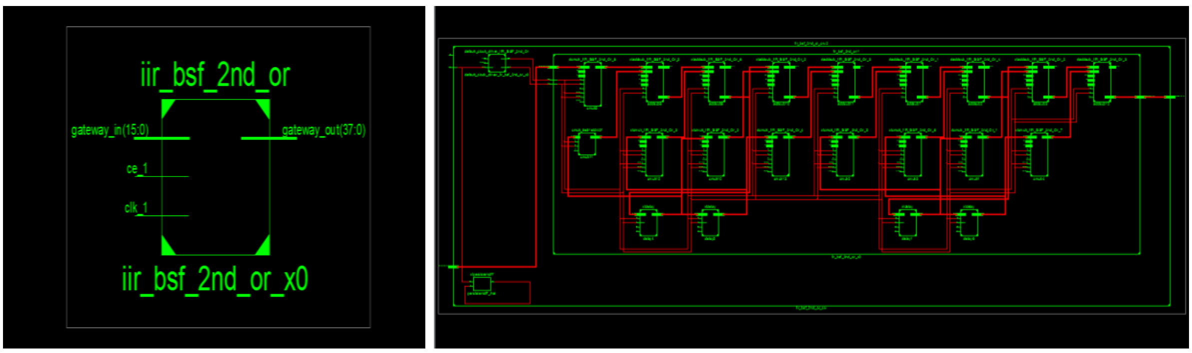


Fig. 4.14 Low and High level RTL schematic of the BSF model.

- **Devices utilization summery**

Following tables give the utilization summary of the designed systems.

Tab. 4.5 Device Utilization Summary for HPF

.Device Utilization Summary			
Slice Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	88	393,600	1%
Number of Slice LUTs	908	196,800	1%
Number used as logic	896	196,800	1%
Number of occupied Slices	261	49,200	1%
Number with an unused Flip Flop	837	925	90%
Number with an unused LUT	17	925	1%
Number of fully used LUT-FF pairs	71	925	7%
Number of bonded IOBs	55	600	9%
Number of BUFG/BUFGCTRLs	1	32	3%
Number of STARTUPs	1	1	100%
Average Fanout of Non-Clock Nets	3.27		

Tab. 4.6 Device Utilization Summary for BSF

Device Utilization Summary			
Slice Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	88	407,600	1%
Number used as Flip Flops	80		
Number of Slice LUTs	968	203,800	1%
Number used as logic	958	203,800	1%
Number of occupied Slices	290	50,950	1%
Number with an unused Flip Flop	881	969	90%
Number with an unused LUT	1	969	1%
Number of fully used LUT-FF pairs	87	969	8%
Number of bonded IOBs	55	400	13%
Number of BUFG/BUFGCTRLs	1	32	3%
Average Fanout of Non-Clock Nets	3.30		

Note that the number of LUTs (Look up Table)used is almost negligible compared to the existing number of LUTs, we can say in this case that consumption is very low, this allows us integrating other computing stages in parallel, which proved the parallelism nature of FPGA. The performance of the designed filters is studied for noise elimination from ECG signals.

As important information in the ECG signal lies in the frequency range of 2 Hz to 100Hz, it is decided to design a high pass filter of cutoff frequency 1.5Hzto remove the baseline noise. Sampling frequency used in the design of filter is 360Hz.The signals used are taken from the universal database MIT/BIH. The following figure (figure4.15) shows the input output signals before and after filtration where the baseline noise is filtered.

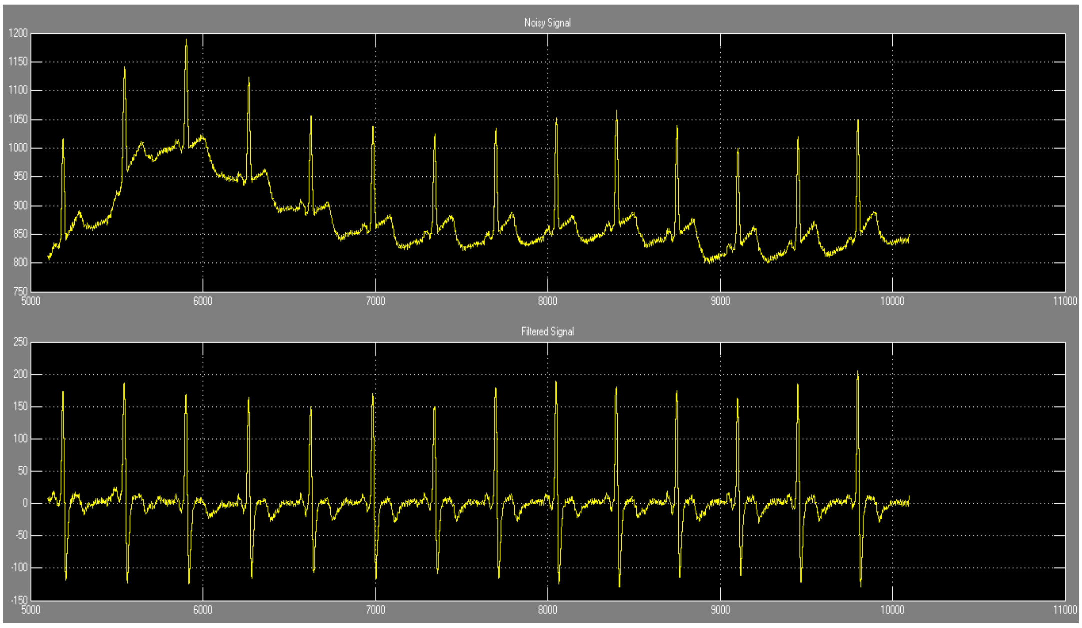


Fig. 4.15 The noisy base line ECG signal (top) and the filtered signal (bottom)

According to obtained results, we can say that the high pass filter filters successfully the noisy ECG signals by the baseline drift. The ECG signal is not distorted by the filtering. Another result proves the performance of the designed filters, this time we use the band-stop filter for removal of the 50 Hz noise. Our filter can remove noise while preserving 50 Hertz frequency content of the ECG measurement signals. This is a filter that attenuates signals in a very narrow frequency band. The used signal in the following is a signal which has both the noise due to variation of the base line and the noise caused by the power supply. Therefore, before applying the band-stop filter at frequency 50Hz, we apply the high pass filter described above to eliminate the drift in the baseline. After removing noise from the baseline, we applied the band-stop filter to the output signal of the high pass filter.

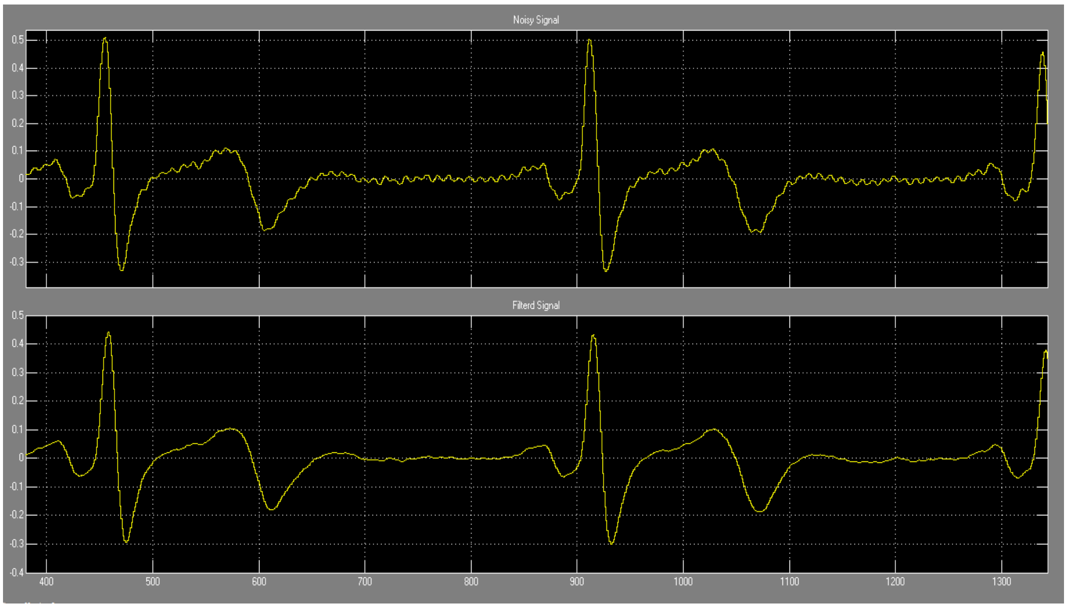


Fig. 4.16 The 50 Hz Noisy ECG signal (top) and the filtered signal (bottom).

Figure 4.16 shows the time domain representation of the ECG noisy signal by the power supply frequency 50Hz and that of the resulting filtered ECG signal. Based on the obtained results, we can say that the band stop filter filters successfully the 50 Hz noise from the ECG signals. One more time, the ECG signal is not distorted by the filtering.

4. Conclusion

In this chapter, we suggested a novel optimal approach for designing infinite impulse response (IIR) digital filters. A low pass and high pass IIR digital filters design examples are considered to investigate the performance of filter designed with the new method, and to compare the performance of the proposed methods, a comparison was made with the Butterworth method.

A two optimized IIR filters types (high pass and band stop), have been after that implemented on a Xilinx reconfigurable field-programmable gate array (FPGA) using hardware description language: VHDL. The performance of the considered two types of filters has been experienced for noise elimination from ECG signal to verify the applicability of this optimal approach.

Conclusion

A fundamental aspect of signal processing is filtering. With the aid of computer programs performing filter design algorithms, designing and optimizing filters can be done relatively quickly. This modest research was based on the design and implementation of IIR digital filters.

In this work, we try to design an optimal IIR filter. The optimality of the design is based on minimizing the error in the sense of weighted least ℓp-norm, for this, one of the main goals of this work is to illustrate the versatility and relevance of ℓp-norm in the design of IIR digital filters.

A novel meta-heuristics Algorithm called CSA is introduced. By decrease of the AP value, CSA tends to perform the search on a limited region wherever a current good result is reached in this region.

On the other hand, by rising of the AP value, the probability of seeking the neighborhood of present high-quality results diminishes and CSA tends to investigate the search room on a total range (randomization). Consequently, employing big values of AP raises diversification. The results obtained show the effectiveness of this design approach and thus approaches the best previous designs in comparison with a conventional IIR designed using butter synthesis approach.

To assure better stability and causality, any poles or zeros that lie outside of the unit circle are reflected back inside. Thus, it is proved that CSA is superior to some other reported meta-heuristic techniques for optimal IIR filter design.

Two specific cases have been considered here: a low pass and high pass IIR digital filters to investigate the performance of filter designed with the CSA method. A comparison was made with the Butterworth method to compare its performance. The reason for choosing the types of the filters is that these cases are particularly useful for applications in digital communications.

A low pass and high pass IIR digital filters design examples are considered to investigate the performance of filter designed with the new method, and to compare the performance of the proposed methods, a comparison was made with the Butterworth method. And for the implementation filters, the performance of the considered types of filters has been experienced for noise elimination from ECG signal to verify the applicability of this optimal approach.

Though lot of work has been done in the field of digital filter design but then also room is still left for doing further exploration with the evolutionary optimization methods and using them for the design of high performance digital FIR &IIR filters.

List of Scientific Productions:

- **International Publication**

1. Ghibeche Y, Saadi S, Hafaifa A. Optimal design of IIR filters based on least ℓp-norm using a novel meta-heuristic algorithm. Int. J. Numer Model. 2018; e2480. https://doi.org/10.1002/jnm.2480

Journal Edited By: Eric Michielssen. **Impact factor: 0.816.** ISI Journal Citation Reports @ Ranking: 2017:215/260 (Engineering, Electrical & Electronic). ISI Journal Citation Reports @ Ranking: 2017:79/103 (Mathematics, Interdisciplinary Applications). Online ISSN:1099-1204. © John Wiley & Sons, Ltd.

- **International Communications**

1. Youcef GHIBECHE, Slami SAADI, Ahmed HAFAIFA, Optimal Design of IIR filters using Least ℓp-norm: Application to ECG Signal Filtering, The 2nd International Conference on Applied Automation and Industrial Diagnostics (ICAAID 2017). Djelfa on 16-17 September 2017, Algeria.
2. Youcef GHIBECHE, Slami SAADI, Atef BENHAOUAS, Design and FPGA Implementation of a Viterbi Decoder for an OFDM Transmission system, The 2nd International Conference on Applied Automation and Industrial Diagnostics (ICAAID 2017). Djelfa on 16-17 September 2017, Algeria.

ملـخـص:

في هذا العمل، طريقة جديدة تم عرضها من اجل تصميم المرشحات الرقمية ذات الاستجابة الغير محدودة؛ هذه الطريقة تعتمد على تقنية المعيار ذو الاس p مع خوارزمية تحسين الأدلة العليا المسماة خوارزمية بحث الغراب. عدة أمثلة لتصميم المرشحات الرقمية ذات الاستجابة الغير محدودة وفق الطريقة الجديدة المقترحة قورنت بنتائج متحصل عليها بطريقة كلاسيكية. النتائج المتحصل عليها بينت ان المرشحات المصممة وفق الطريقة الجديدة احسن من تلك وفق الطريقة الكلاسيكية، من حيث الدقة والمتانة.

المرشحات الرقمية ذات الاستجابة الغير محدودة المحسنة المتحصل عليها، تم تنفيذها على بطاقة FPGA من نوع xc6vsx315t-Virtex 6 3ff1-156 باستعمال لغة توصيف العتاد: VHDL. أداء هاته المرشحات الرقمية تم دراستها في حالة التخلص من الضجيج لإشارة تخطيط كهربائية القلب.

كلمـات مفتاحيـــة: خوارزمية بحث الغراب، تصميم المرشحات الرقمية ذات الاستجابة الغير محدودة، المعيار ذو الاس p، الأدلة العليا، VHDL، FPGA، تخطيط كهربائية القلب.

Abstract:

In this work, a new optimal infinite impulse response (IIR) filter design method is presented based on the ℓp-norm method (combined with) the meta-heuristic optimization algorithm named crows search algorithm (CSA). Several examples of RII filter design using the proposed approach are presented, compared with those obtained by the classic butter method. The results obtained show that the filters resulting from the new approach are better in terms of accuracy and robustness. An optimized RII filters are then implemented on a Xilinx Virtex6 xc6vsx315t-3ff1-156 FPGA card using a hardware description language: VHDL. The performance of the designed filters is then studied in the case of noise elimination from an ECG signal

Key Words : Crow Search Algorithm, IIR design, Lp-norm, meta-heuristic, VHDL; FPGA; ECG.

Résumé:

Dans ce travail, une nouvelle méthode de conception de filtre de réponse impulsionnelle infinie (RII) optimal est présenté basée sur la méthode du ℓp-norm combiné avec l’algorithme d'optimisation méta-heuristique nommé l’algorithme de recherche corbeaux (ARC). Plusière exemples de conception de filtres RII en utilisant l’approche proposée sont présentés, comparées avec ceux obtenus par la méthode classique de butterworth. Les résultats obtenus montrent que les filtres résultants de la nouvelle approche sont meilleurs en en termes de précision et de robustesse. Des filtres RII optimisés obtenu sont ensuite implémentés sur une carte FPGA de type Virtex6 xc6vsx315t-3ff1-156 de Xilinx en utilisant un langage de description matérielle: VHDL. Les performances des filtres conçus sont ensuite étudiées dans le cas d’élimination des bruits d'un signal ECG.

Mots Clés: Algorithme de recherche corbeaux, Conception des filtres RII, Lp-norm, méta-heuristique, VHDL; FPGA; ECG.

Received: 28 March 2018 | Revised: 26 May 2018 | Accepted: 7 July 2018
DOI: 10.1002/jnm.2480

Optimal design of IIR filters based on least ℓ_p-norm using a novel meta-heuristic algorithm

Youcef Ghibeche | Slami Saadi | Ahmed Hafaifa

LAADI Laboratory, Faculty of Exact Sciences and Informatics, Ziane Achour University of Djelfa (UZAD), BP3117 Djelfa, Algeria

Correspondence
Slami Saadi, LAADI Laboratory, Faculty of Exact Sciences and Informatics, Ziane Achour University of Djelfa (UZAD), BP3117 Djelfa, Algeria.
Email: saadisdz@gmail.com

Abstract
In this work, an optimal infinite impulse response (IIR) filter is designed using least P_{th} norm and new meta-heuristic optimization algorithm called Crow Search Algorithm (CSA). The two approaches try to get a resolution to the specified problem but use unlike approaches to perform so. Furthermore, the ℓp-norm is used to design optimum IIR filters. The main motivation of using CSA is its user-friendly optimization tool for both beginner and professional users. This Algorithm provides a good stability between variety and strength which are mainly controlled by the parameter of awareness probability. Some examples of IIR filter design cases are treated, and a comparison is done. It is revealed that the design objective is successfully attained, and the resultant filter parameters are better using the ℓp-norm combined with the proposed CSA meta-heuristic optimization algorithms compared with IIR filter designed using butter synthesis approach in terms of accuracy and robustness. Classical approaches are largely outperformed by CSA for the same filter order.

KEYWORDS
Crow Search Algorithm, IIR design, Lp-norm, meta-heuristic

1 | INTRODUCTION

In recent years, design of infinite impulse response (IIR) digital filters has appeared as an essential research domain in the field of signal processing (digital signal processing). The main significant benefit of IIR is that it provides significantly superior performance compared with the finite impulse response (FIR) filter, and it suits the requested filter specifications with a less filter order.[1] Digital IIR filters have been widely used in a variety of fields such as automatic control, telecommunications, speech processing, and other areas.[2] The IIR filter design mostly exploits two methods[3]: transformation technique and optimization technique. In the classical approach for IIR filter design, the minimum order and the filter coefficients are selected for a standard prototype low pass Butterworth, Chebyshev Type-I, Chebyshev Type-II, and Elliptic filters which are transformed to digital low pass, high pass, band pass, and band stop IIR digital filter using different transformation techniques such as bilinear transformation etc..[4] Numerous optimization methods have been developed based on modern heuristics optimization algorithms. Several papers address the advantages of these modern heuristic tools for the designing of IIR filters. Benvenuto and Marchesi[5] explained the most important features of employing a simulated annealing algorithm in designing digital filters with a linear phase; the algorithm was after that applied to the design of the FIR filter, and the result was not notable. Furthermore, it demands a large amount of computations. Chen et al[6] applied another universal optimization technique called the adaptive simulated annealing to digital IIR filter design. The genetic algorithm (GA) has obtained large interest in application to digital IIR filter design.[7-10] In Liang et al,[11] they used GA to design a 1-D IIR filter with canonical-signed-digit coefficients constrained to a low-pass filter; Ahmad and Antoniou[12] investigated the use of GAs for the design of numerous kinds of digital filters. It was found that GAs require a large amount of computation. Another different works based on GA are reported in the literatures. These are orthogonal genetic algorithm,[12] hybrid Taguchi GA,[13] hybrid genetic algorithm,[14] and real coded genetic algorithm.[15] The particle swarm optimization (PSO) has been applied successfully to the IIR filter designs. Krusienski and Jenkins[16] introduced the application of PSO technique to IIR adaptive filter structures. A quantum-behaved particle swarm optimization algorithm was employed to design IIR filter,[17] while Das and Konar[18] applied the PSO algorithm to design two-dimensional IIR filters, Chen and Luk[19] made use of PSO algorithm for designing digital IIR filters in a realistic time domain situation where the wanted filter output is corrupted by noise, and Jiang et al[20] proposed a novel hybrid particle swarm optimization and gravitational search algorithm (GSA) for IIR filter design. Karaboga[21] and Karaboga and Cetinkaya[22] recommended artificial bee colony for a variety of benchmark adaptive IIR representations in 2009 and 2011, respectively. The seek optimization algorithm, which imitates human search character, was suggested by Dai et al[23] to resolve five IIR system identifications and was compared with GA and PSO. In Panda et al, Saha et al, and Kalinli and Karaboga[24-26] cat swarm optimization, opposition-based bat algorithm, and artificial immune systems were employed for IIR model identification. The results reveal greater identification performance of the three algorithms compared with that attained through GA and PSO. An additional swarm intelligence algorithm, differential evolution, and its alternative with wavelet mutation were used in Karaboga and Mandal et al[27,28] for digital IIR and FIR filter design. In Rashedi et al,[29] they proposed the GSA for linear IIR filter and nonlinear rational filter modeling; Saha et al[30] presented a combined version of the GSA and wavelet mutation and take advantage of it for the design of an IIR filter. In Karaboga et al,[31] ant colony optimization algorithm with global optimization capability is recommended for digital IIR filter design. Though, ACO has a tendency to local minima in complex problem, and its convergence rapidity is also delayed.[32] Kalinli and Karaboga[33] suggested a scheme for the design of digital adaptive IIR filters based on tabu search algorithm. A different method simulated the flash pattern and characteristics of fireflies algorithm applied to get optimal sets of adaptive IIR filter coefficients; Upadhyay et al[34] accomplished that is excellent and sufficient to handle indefinite system identification problem. These swarm-based methods demonstrate high-quality performance in IIR system identification and complex optimization problem. Kalinli and Karaboga[35] suggested a method for the design of digital adaptive IIR filters based on tabu search algorithm.

The comparison between methods is important for achieving the best optimal filters design. In this work, we try to improve prior classical designs using butter synthesis approach by introducing a novel optimization algorithm called Crow Search Algorithm (CSA) in a combination with ℓp-norm. An assessment is made in the purpose of looking for more efficient filters design approaches.

2 | BASICS OF IIR FILTER DESIGN

Digital IIR filters are time invariant linear systems. In general, the IIR filter is described by the following difference equation:

$$y(n) = \sum_{K=0}^{N} a_K . x(n-K) - \sum_{K=0}^{M} b_K . y(n-K) \tag{1}$$

where x represents the input signal and y the output signal. Thus, the filter transfer function can be expressed in the following general form[7]:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{K}^{N} a_K . Z^{-1}}{\sum_{K}^{M} b_K . Z^{-1}} \tag{2}$$

where a_K and b_K represent the filter coefficients; these determine the characteristics of the filter. Generally, the numerator and denominator have the same degree M = N. The IIR filter frequency response is expressed as follows:

$$H(j\omega) = |H(j\omega)| e^{-j\varphi(j\omega)}. \tag{3}$$

Expressions of the amplitude and the phase shift may be determined from expression (2). They have the following shapes, respectively[8]:

$$|H(j\omega)|^2 = \left[H(z)H\left(z^{-1}\right)\right]_{z=e^{j\omega}} \tag{4}$$

$$\phi(j\omega) = -\frac{1}{2j}\ln f\left[\frac{H(z)}{H(z^{-1})}\right]_{z=e^{j\omega}}. \tag{5}$$

Assume that the amplitude response of a recursive filter is required to approach some specified amplitude response as closely as possible. Such a filter can be designed in two general steps, as follows[36]:

1. An objective function which is dependent on the difference between the actual and specified amplitude response is formulated.
2. The objective function reached is minimized among the transfer-function coefficients.

Thus, the design of this filter is regarded like an optimization problem of cost function $J(\omega)$ declared: min $J(\omega)$ where $\omega = [a_0, a_1, a_2, \cdots a_N, b_1, b_2, \cdots, b_M]$ is the filter coefficients vector. The purpose is to minimize the cost function $J(\omega)$ by regulating ω. The cost function is generally known as

$$J(\omega) = \frac{1}{N}\sum_{k=1}^{N}(d(k) - y(k))^2 \tag{6}$$

where $d(k)$ and $y(k)$ are the wanted and real responses of the filter, in that order, and N is the number of samples used for the computation of cost function.

3 | ℓP-NORM METHOD

The distance approach is an approximation problem for which various norms could be selected, like the ℓp-norms. In order to compute the gradient vector, only even power p is considered. The most important purpose of the design algorithm of digital IIR filter is to look for filter coefficients a_k and b_k.

A recursive filter with an even order *Ord* can be represented by the transfer function:

$$H(z) = H_0\prod_{j=1}^{J}\frac{a_{0j} + a_{1j}z + z^2}{b_{0j} + b_{1j}z + z^2}$$

where a_{ij} and b_{ij} are real coefficients, $J = Ord/2$, and H_0 is a positive multiplier constant. The amplitude response of the filter can be expressed as

$$M(x, \omega) = \left|H\left(e^{j\omega T}\right)\right|$$

where: $x = [a_{01}a_{11}b_{01}b_{11}...b_{1J}H_0]^T$ is a column vector with 4 J + 1 elements and ω is the frequency.

Let $M_0(\omega)$ be the specified amplitude response. The difference between $M(x, \omega)$ and $M_0(\omega)$ is, in effect, the approximation error and can be expressed as

$$e(x, \omega) = M(x, \omega) - M_0(\omega).$$

By sampling $e(x, \omega)$ at frequencies ω_1, ω_2, ..., ω_K, the column vector:

$$E(x) = [e_1(x)e_2(x)...e_K(x)]^T$$

can be formed where: $e_i(x) = e(x, \omega_i)$ For $i = 1, 2...K$.

The approximation problem at hand can be solved by finding a point $x = \breve{x}$ such that: $e_i(\breve{x}) \approx 0, \ \text{for } i = 1, 2....K$.

4 | CROW SEARCH ALGORITHM (CSA)

Crow Search Algorithm (CSA) is a novel population-based meta-heuristic technique. CSA is based on the smart manners of crows. CSA runs through the idea that crows accumulate their extra foodstuff in secret sites and recover it while it is required. The main reasons of using CSA are its easy implementation, few parameters to adjust, fast

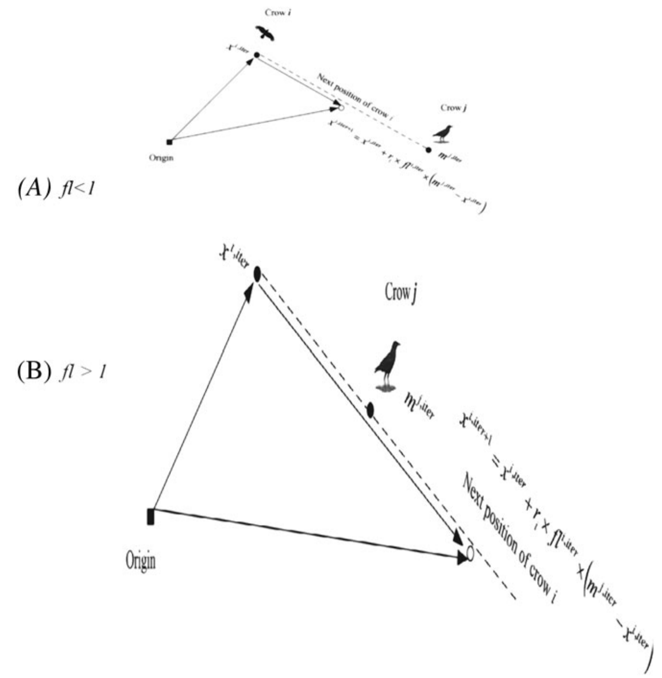


FIGURE 1 Flowchart of state 1 in CSA (A) fl < 1 and (B) fl > 1. Crow i can go to every position on the dash line

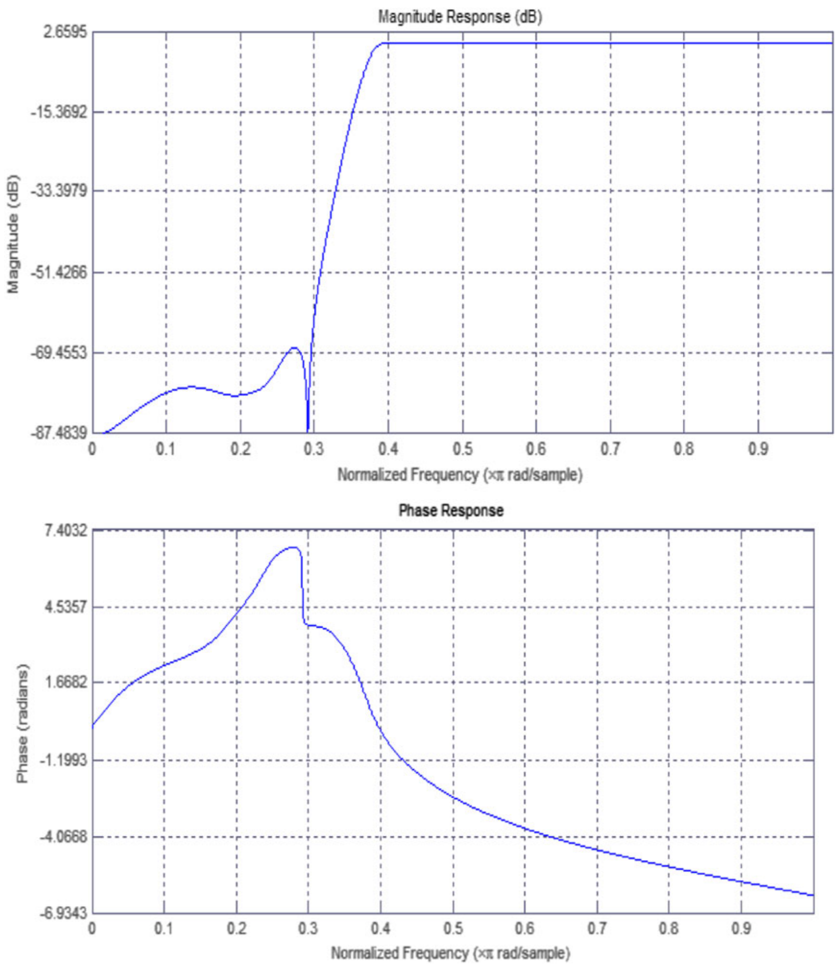


FIGURE 2 Magnitude and phase responses of the high-pass filter with ord = 20

convergence speed, and high efficiency, which in turn makes it very attractive for applications in different engineering areas.[37]

Based on the above-mentioned intelligent behaviors, a population-based meta-heuristic algorithm, CSA, is set. The main attitudes of CSA are scheduled as

- Crows exist in the nature of group.
- Crows remember the location of their hiding sites.
- Crows pursue each other to make stealing.
- Crows keep their reserves from being stolen by a probability.

It is implicit that present is a d-dimensional situation counting a quantity of crows. The number of crows (flock size) is *Nc*, and the position of crow *i* at time (iteration) *iter* in the search space is specified by a vector:

$$x^{i,iter}(i = 1, 2, ...Nc; iter = 1, 2, ..., iter_{max}) \text{ where } x^{i,iter} = \left[x_1^{i,iter}, x_2^{i,iter}, ...x_d^{i,iter}\right]$$

And itermax is the maximum number of iterations. Each crow has a memory in which the position of its hiding place is memorized. At iteration *iter*, the position of hiding place of crow *i* is shown by *mi,iter*. This is the best position that crow *i* have obtained so far. In fact, in remembrance of every crow, the location of its finest skill has been stored. Crows travel

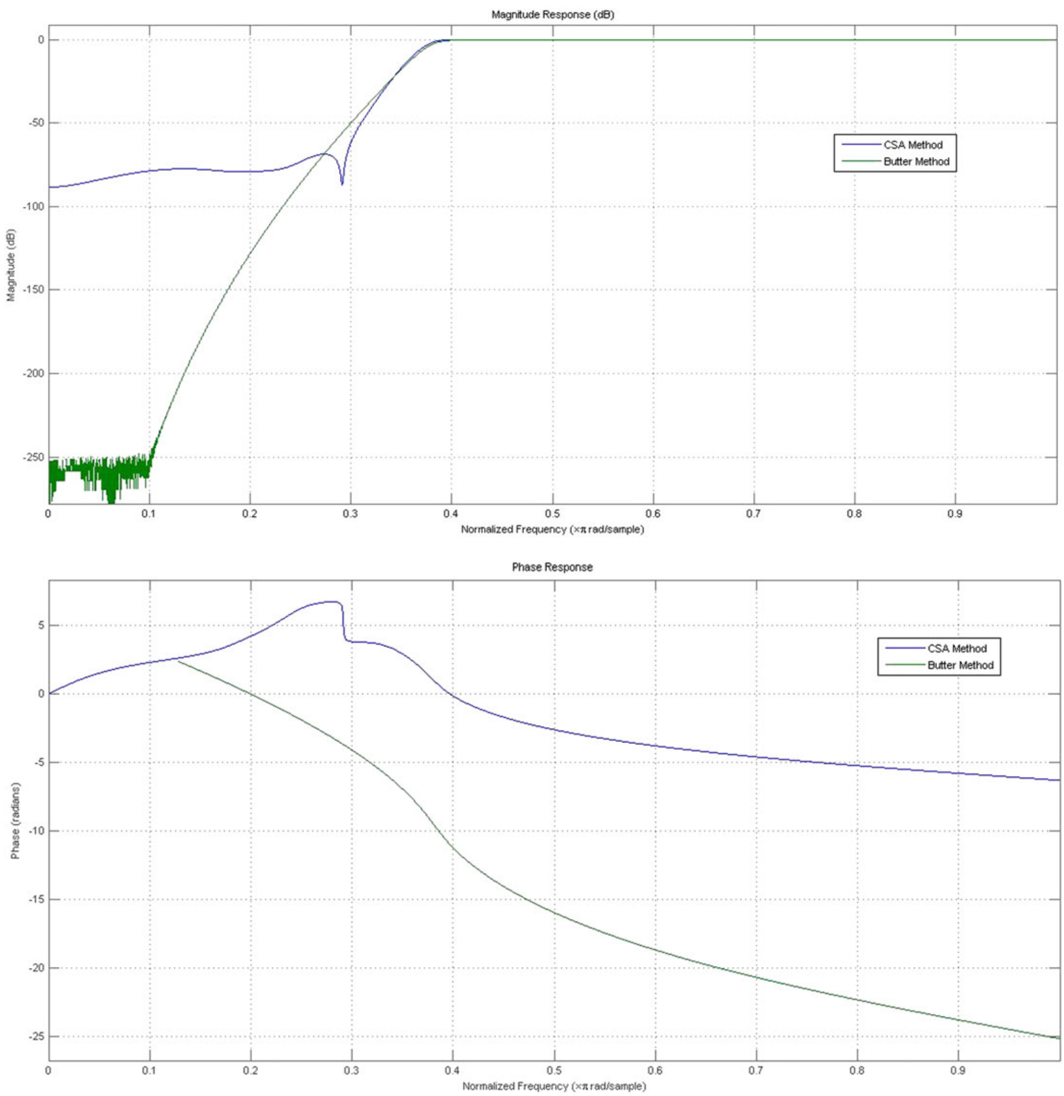


FIGURE 3 Comparison between CSA optimal design and butter synthesis approach in magnitude and phase responses of the high-pass filter with ord = 20

in the surroundings and look for improved provisions supply (hiding sites). Assume that at iteration iter, crow *j* wants to visit its hiding place, *mi,iter*. On this iteration, crow *i* make a decision to pursue crow *j* to come near the hiding position of crow *j*. In this case, two situations may occur:

Situation 1: Situation 1:Crow *j* does not recognize that crow i is next it. As a consequence, crow *i* will move toward to the hiding position of crow *j*. In this case, the latest location of crow *i* is achieved by

$$x^{i,iter+1} = x^{i,iter+1} + r_i fl^{i,iter}\left(m^{i,iter} - x^{i,iter}\right)$$

where r_i is a random number with unvarying distribution between 0 and 1 and $fl^{i,\ iter}$designate the travel span of crow *i* at iteration iter. Figure 1 illustrates the diagram of this state and the effect of $fl^{i,\ iter}$ on the search capability. Little values of $fl^{i,\ iter}$ guide to neighboring seek (at the surrounding area of $x^{i,\ iter}$), and big value outcomes in universal seek (distant from $x^{i,\ iter}$). As Figure 1A shows, if the value of $fl^{i,\ iter}$ is selected less than 1, the next position of crow *i* is on the dash line between $x^{i,\ iter}$ and *mi,iter*. As Figure 1B indicates, if the value of $fl^{i,\ iter}$ is selected more than 1, the next position of crow *i* is on the dash line which may exceed *mi,iter*.

Situation 2: Situation 2:Crow j recognizes that crow i is next it. As a consequence, in order to keep its store from being stolen, crow j will trick crow i by going away to a different location of the seek room. Totally, states 1 and 2 can be expressed as follows:

$$x^{i,iter+1} = x^{i,iter} + r_i X fl^{i,iter} X\left(m^{i,iter} - x^{i,iter}\right) \ \ r^j \geq = AP^{j,iter}$$

where *rj* is a random number with uniform distribution between 0 and 1 and *APj,iter* denotes the knowledge probability of crow *j* at iteration *iter*.

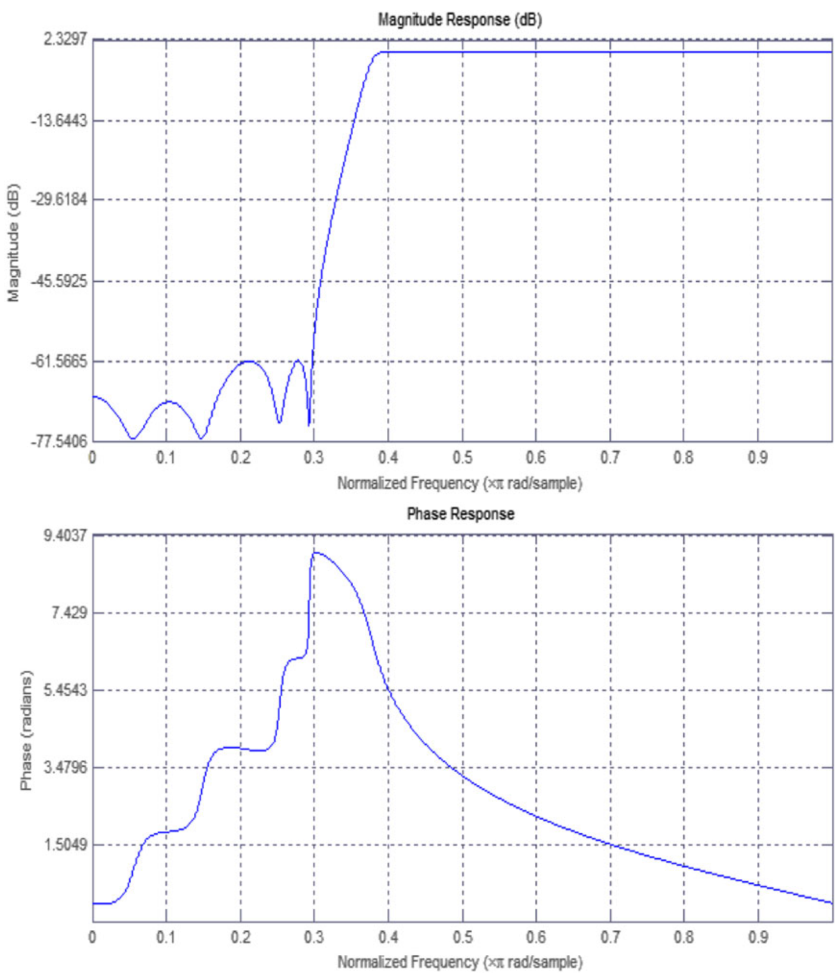


FIGURE 4 Magnitude and phase responses of the high-pass filter with ord = 8

Pseudo code of CSA is shown below:

1. Initialize problem and adaptable factors: The optimization problem, assessment variables, and constraints are identified. After that, the adaptable factors of CSA (group size Pd), maximum number of iterations (itermax), flight length (fl), and awareness probability (AP) are esteemed.
2. Initialize location and memory of crows: N crows are arbitrarily located in a d-dimensional seek room as the elements of the group. Every crow designates a practical solution of the problem, and d is the number of assessment variables.

$$\text{Crows} = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_d^1 \\ x_1^2 & x_2^2 & \cdots & x_d^2 \\ \vdots & \vdots & \vdots & \vdots \\ x_1^N & x_2^N & \cdots & x_d^N \end{bmatrix}$$

The memory of every crow is initialized. Given that at the beginning iteration, the crows have no knowledge, it is assumed that they have buried their provisions at their original locations.

3. Calculate fitness (objective) function:

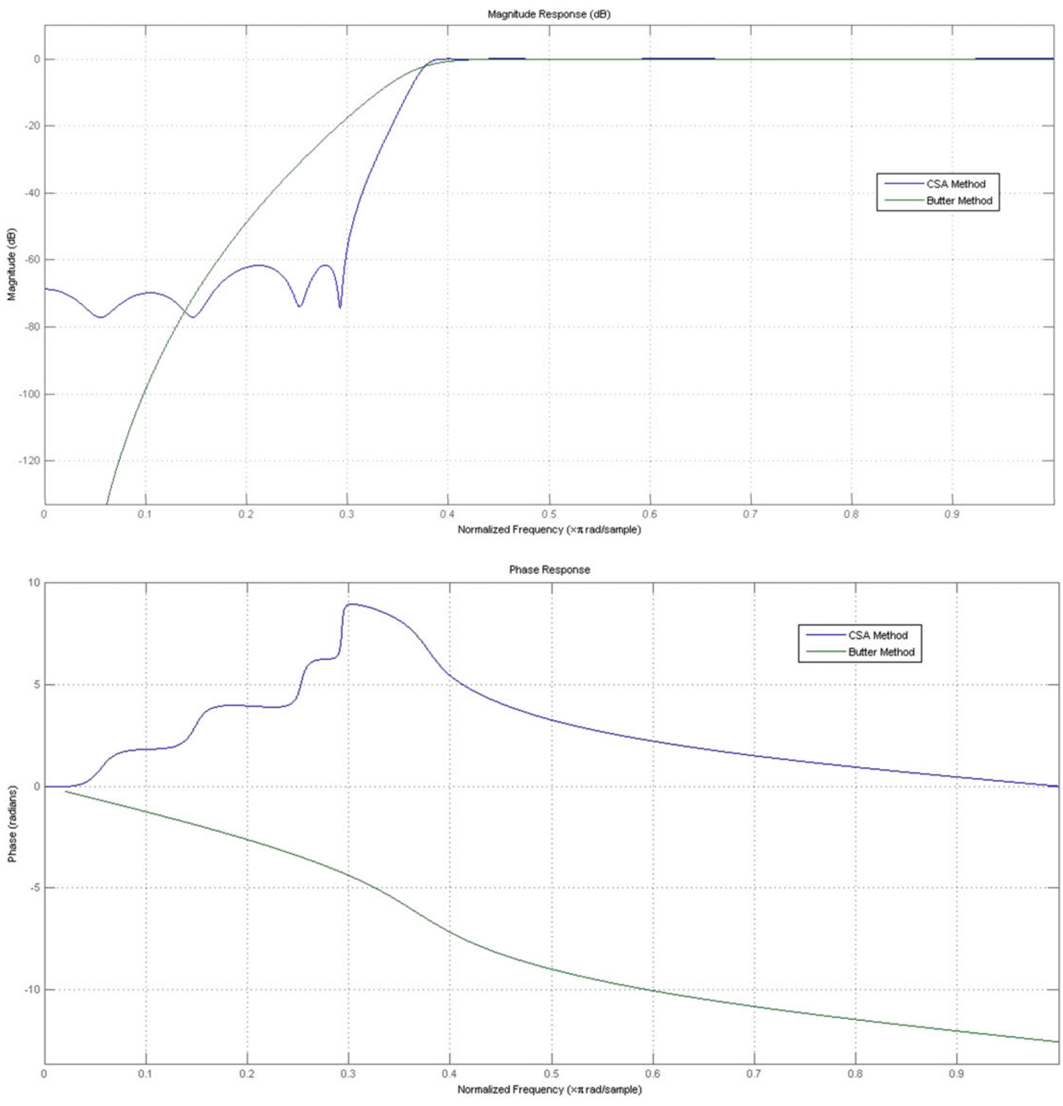


FIGURE 5 Comparison between CSA optimal design and butter synthesis approach in magnitude and phase responses of the high-pass filter with ord = 8

For every crow, the superiority of its location is calculated by introducing the assessment variable values into the objective function.

4. Create new location:

Crows create new location in the seek room: assume crow *i* wishes to create a new location. For this aim, this crow randomly selects one of the flock crows (for example crow *j*) and follows it to discover the position of the foods hidden by this crow *mj,iter*. The new location of crow *i* is obtained by the preceding position equation. This procedure is replicated for the entire crows.

5. Verify the viability of new locations:

The viability of the new location of every crow is verified. If the new location of a crow is possible, the crow revises its location. If not, the crow continues in the present location and does not travel to the produced new location.

6. Calculate fitness function of new locations:

The fitness function cost for the new location of every crow is calculated.

7. Revise memory:
8. The crows bring up to date their memory as:

$$m^{i,iter+1} = \begin{cases} x^{i,iter+1} f\left(x^{i,iter+1}\right) \textit{ is better } \text{than } f\left(m^{i,iter}\right) \\ m^{i,iter} o.w \end{cases}$$

where *f(.)* denotes the objective function value.

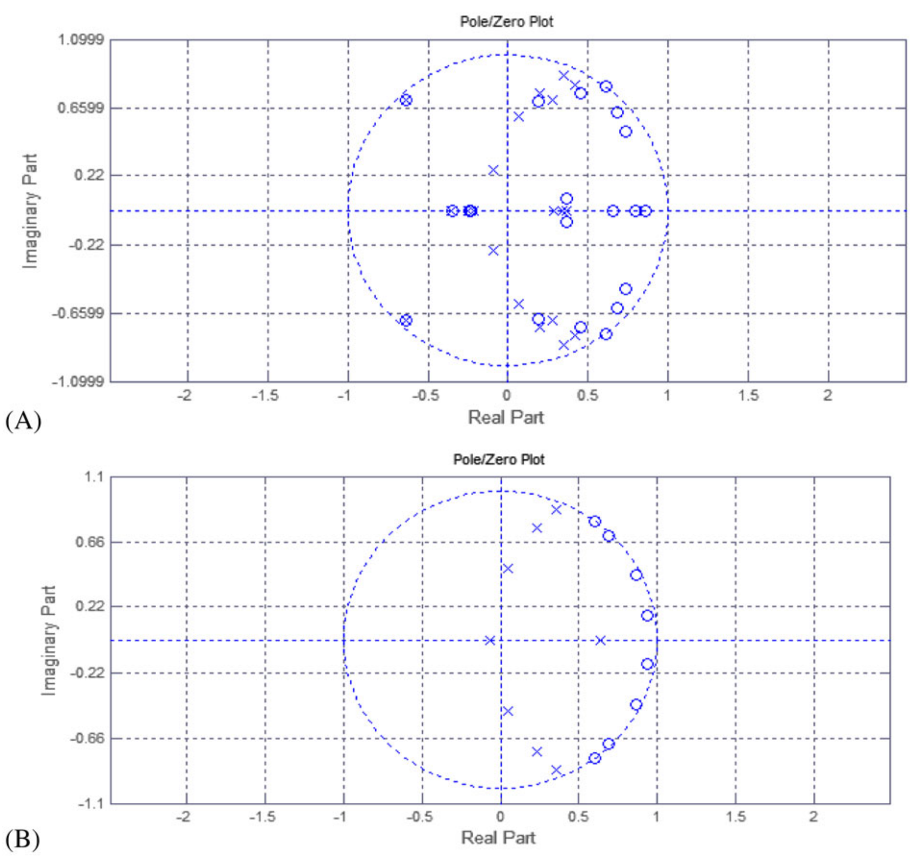


FIGURE 6 Pole-zero position of the high-pass filters: A, ord = 20; B, ord =8

It is observed that if the fitness function cost of the new location of a crow is superior than the fitness function value of the memorized location, the crow revises its memory by the new position.

$$\text{Memory} = \begin{bmatrix} m_1^1 & m_2^1 & \cdots & m_d^1 \\ m_1^2 & m_2^2 & \cdots & m_d^2 \\ \vdots & \vdots & \vdots & \vdots \\ m_1^N & m_2^N & \cdots & m_d^N \end{bmatrix}$$

9. Check termination criterion:

Steps 4 to 5 are repeated until *itermax* is reached. When the termination condition is reached, the most excellent location of the memory in terms of the objective function cost is considered as the solution of the optimization problem.[37]

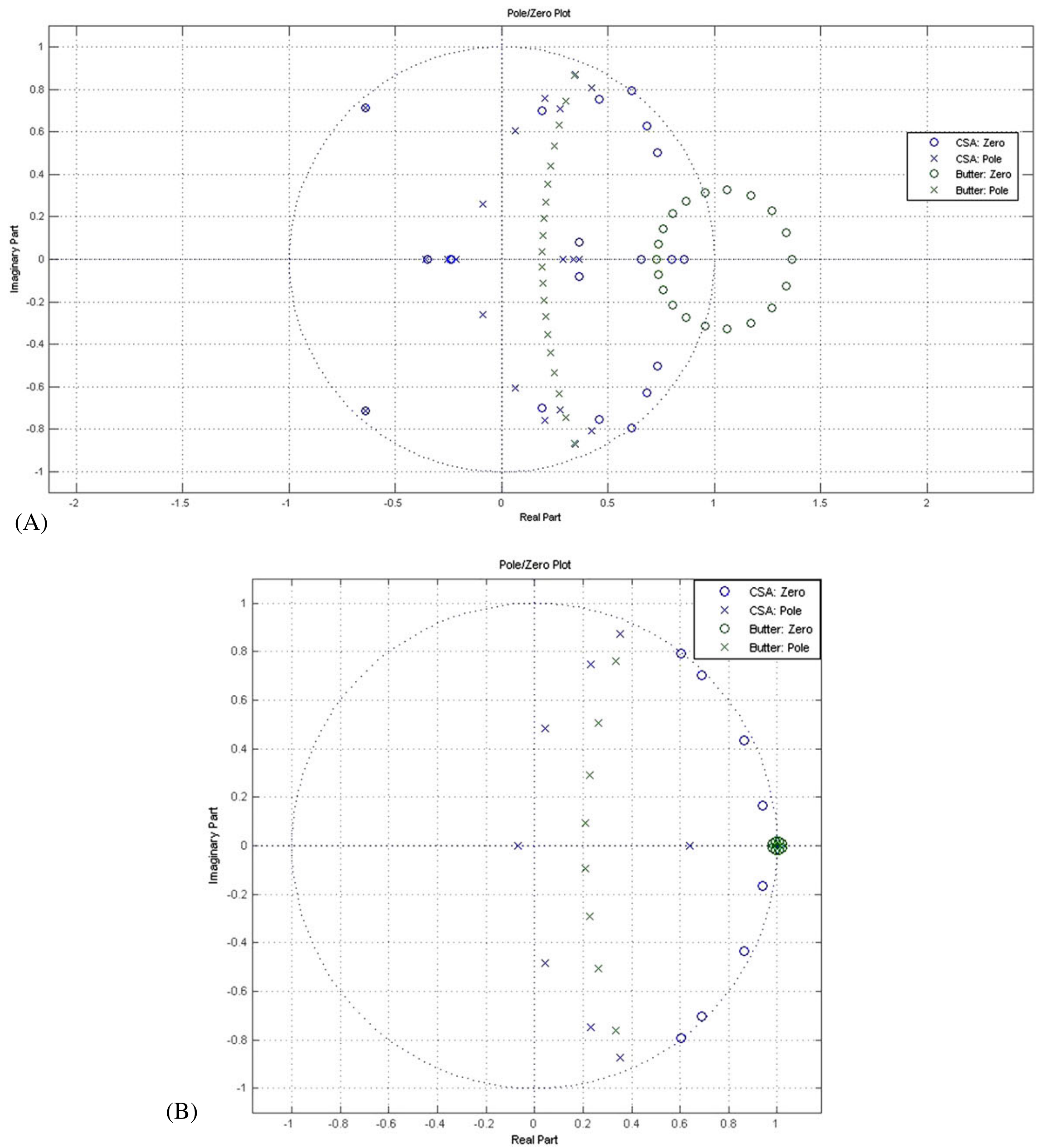


FIGURE 7 Comparison between CSA optimal design and butter synthesis approach in pole-zero positions of the high-pass filters: A, ord = 20; B, ord = 8

5 | OPTIMAL FILTER DESIGN

Consider the IIR filter with the input-output relationship governed by

$$y(k) = \sum_{i=1}^{L} a_i.x(k-i) - \sum_{i=0}^{M} b_i.y(k-i)$$

where x represents the input signal and y the output signal.

The filter transfer function is stated:

$$H(z) = \frac{A(z)}{B(z)} = \frac{\sum_{i=0}^{L} a_i.Z^{-i}}{1 + \sum_{i=1}^{M} b_i.Z^{-i}}$$

where a_i and b_i represent the filter coefficients. An important task for the designer is to find values of them. So, the design of this filter can be considered as an optimization problem of cost function $J(\omega)$ stated as *min* $J(\omega)$ where $\omega = [a_0, \; a_1, \; a_2, ..., a_L, b_1, \; b_2, ..., b_M]$ is filter coefficient vector. The purpose is to minimize the cost function $J(\omega)$ through regulating ω. The cost function is generally expressed by

$$J(\omega) = \frac{1}{N} \sum_{k=1}^{N} (d(k) - y(k))^2$$

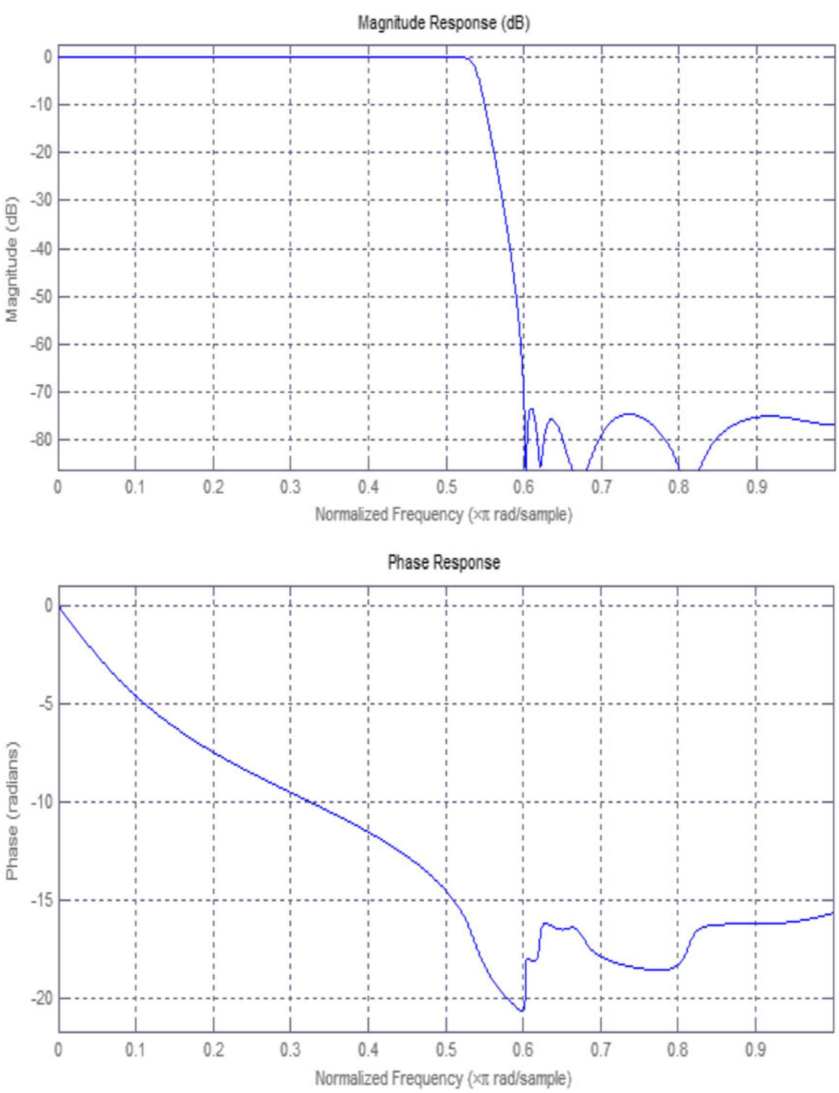


FIGURE 8 Magnitude and phase responses of the low-pass filter with ord = 20

where $d(k)$ and $y(k)$ are the wanted and real responses of the filter, respectively, and N is the number of samples exploited for the computation of cost function.

The commonly used approach for IIR filter design is to represent the problem as an optimization problem with the mean square error as the error fitness function.

5.1 | Using CSA and ℓp-norm to solve the optimization problem

The distance approach refers to an estimate problem for which different norms could be selected, such as the ℓp-norms with $p = 2$. Solutions are reached using CSA instead of standard Newton's method. Assume that the amplitude response of a recursive filter is required to approach some specified amplitude response as closely as possible. Such a filter can be designed in two general steps, as follows[38]:

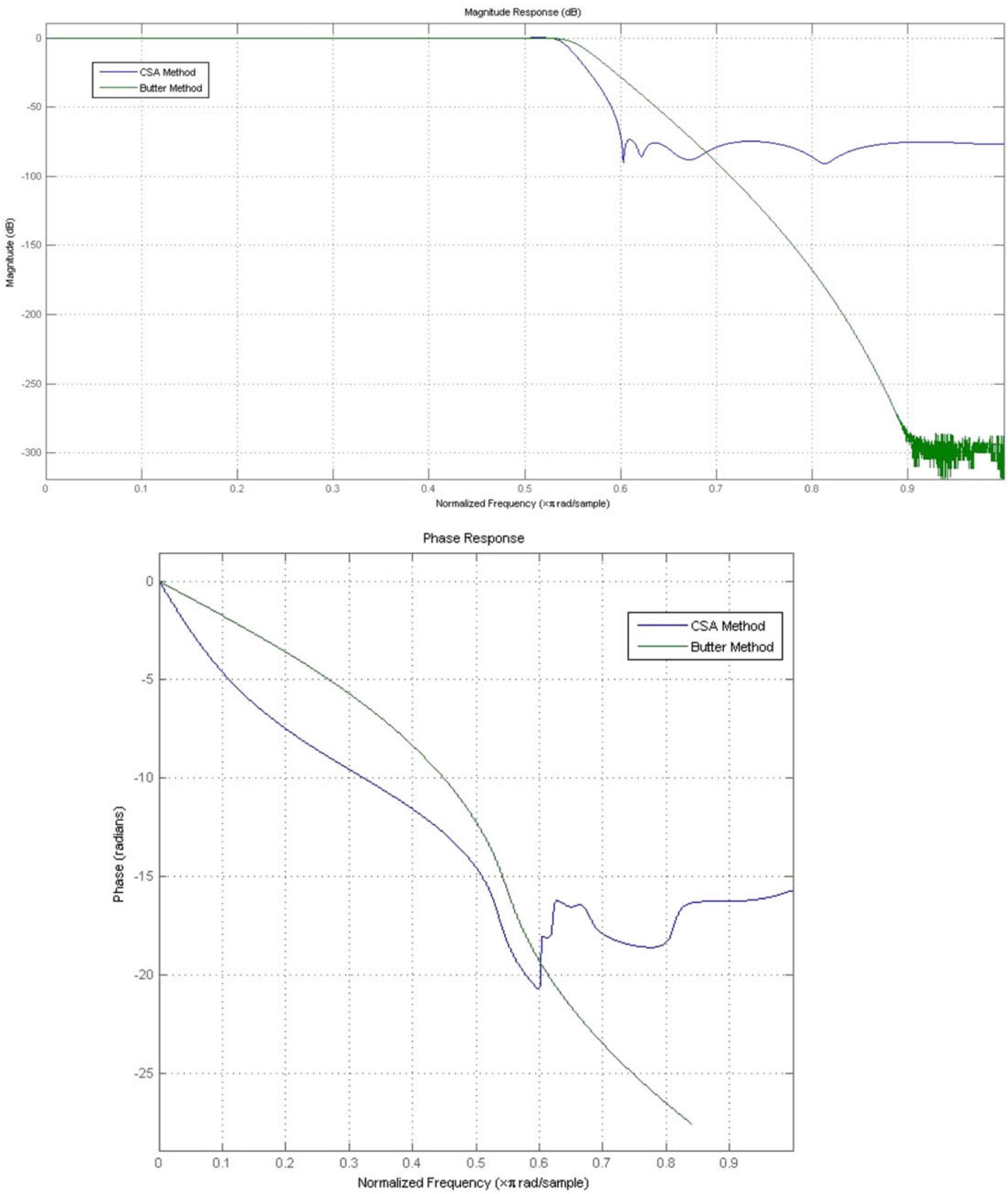


FIGURE 9 Comparison between CSA optimal design and butter synthesis approach in magnitude and phase responses of the low-pass filter with ord = 20

1 An objective function which is dependent on the difference between the actual and specified amplitude response is formulated.
2 The objective function obtained is minimized with respect to the transfer-function coefficients.

An objective function satisfying number of fundamental requirements can be defined as

$$\Psi(x) = L_P = \|E(x)\|_P = \left[\sum_{i=1}^{K} |e_i(x)|^P\right]^{1/P} \tag{7}$$

where: $\widehat{E}(x) = \max_{1 \le i \le k} |e_i(x)|$ and p is an integer.

The required design can be obtained by solving the optimization problem: $\underset{x}{\text{minimize}}\Psi(x)$. This optimization problem can be solved using CS algorithm which is deployed to get the optimal solution for the objective functions. The basic procedure for implementing the minimization CS algorithm is given by the following steps:

Step 1: Input $\check{x}_0$ and ε_1. Set: $K = 1, p = 2, \mu = 2, \widehat{E}_0 = 10^{99}$.
Step 2: Initialize frequencies $\omega_1, \omega_2, \ldots, \omega_K$. Which means, initialize the real coded particles (x) of the *np* population; each consists of an equal number of numerator and denominator filter coefficients bk and ak, respectively; the total coefficients *D = (ord + 1)\2* for designed filter with order *ord*;
Step 3: Using $\check{x}_{k-1}$ as initial point, minimize $\Psi(x)$ in Equation (7), with respect to x, to obtain $\check{x}_k$ and set: $\widehat{E}_k = \widehat{E}(\check{x}_k)$.

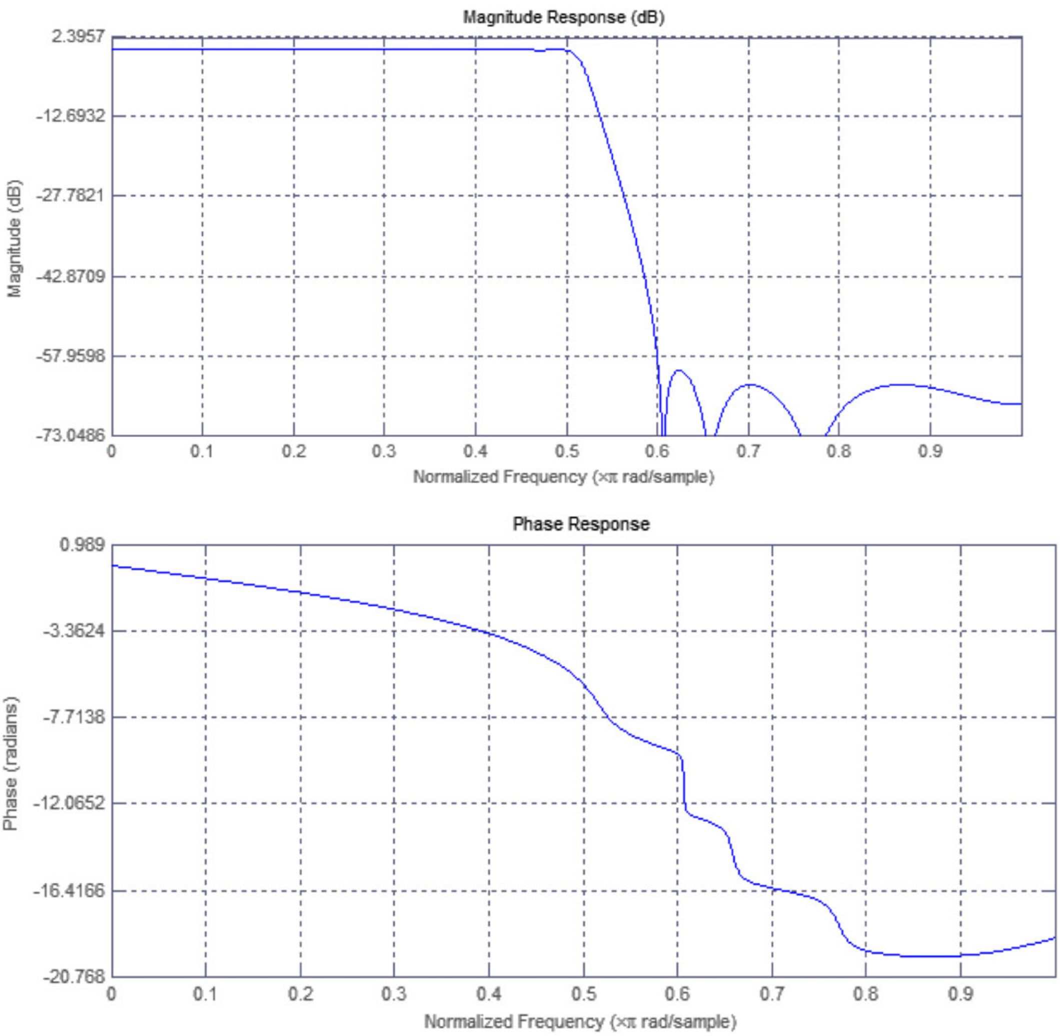


FIGURE 10 Magnitude and phase responses of the low-pass filter with ord = 8

Step 4: If $\left|\hat{E}_{k-1}-\hat{E}_k\right|<\varepsilon_1$, then output $\check{x}_k$ and $\hat{E}_k$, and stop. Else, set: $p = \mu p$, $k = k + 1$ and go to Step 3.

The minimization carried out by introducing CSA in Step 3 using the implemented pseudo code described in Section 4.
The following parameters are chosen for the CSA code:
Number of runs: itermax = 500;
Problem dimension: Pd = 1;
Flight length: fl = 50;
Awareness probability: AP = 0.1;

6 | SIMULATION RESULTS

In this section, low pas and high pass IIR digital filters design examples are considered to investigate the performance of filters designed with the combined approaches. To compare the performance of the proposed method, the results of the filter are obtained through simulations. The study implementation is carried out on an Intel Core I7, 2.40 GHz CPU with 6 GB of RAM. The simulations are performed using MATLAB programming language to demonstrate the potentiality of ℓp-norm combined with CSA algorithm for the design of IIR digital filters. Computation complexity and run

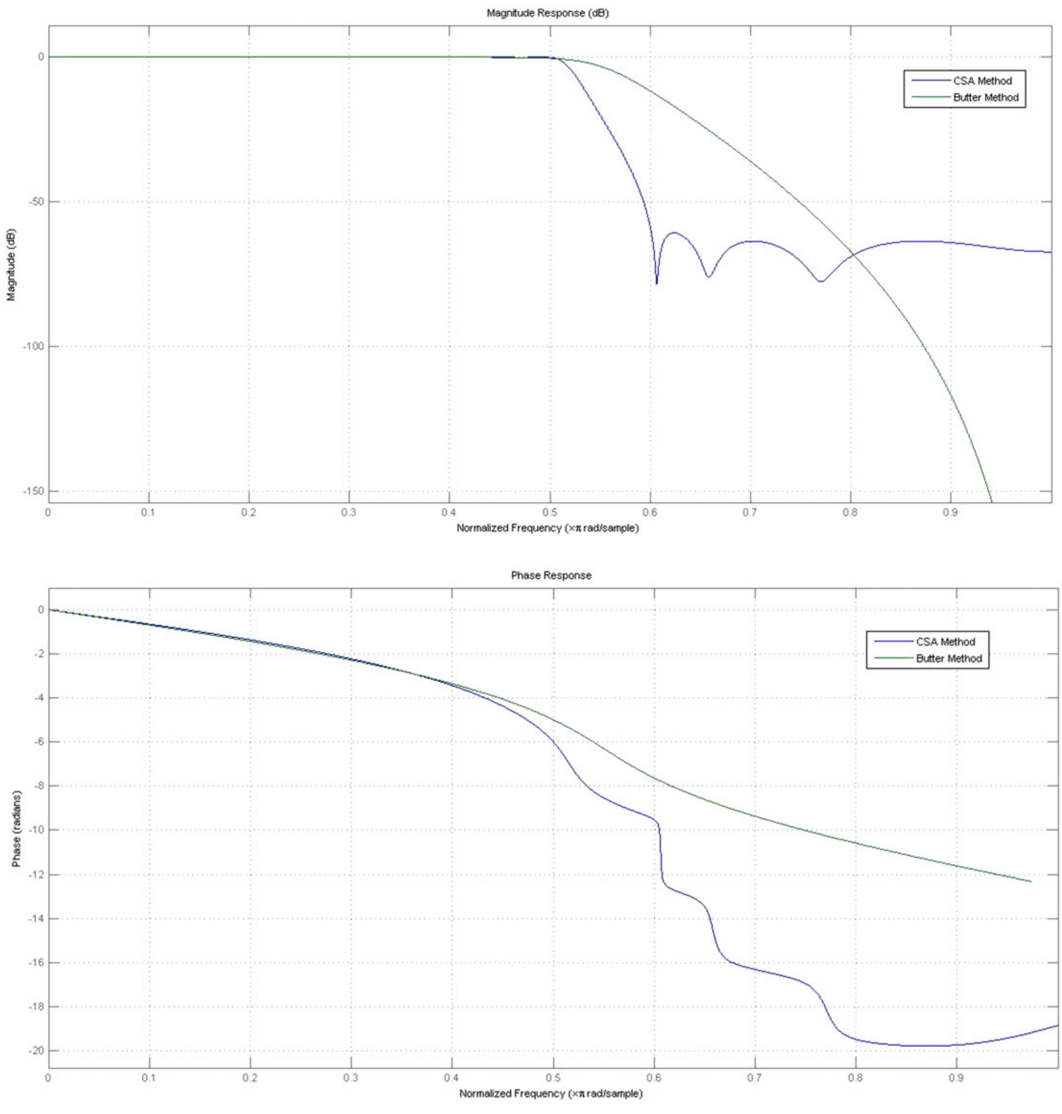


FIGURE 11 Comparison between CSA optimal design and butter synthesis approach in magnitude and phase responses of the low-pass filter with ord = 8

time depend on the filter order and vary from some minutes to tens of minutes. Four examples are presented to justify the properties of this method. Following are design simulation parameters used for all examples.

Minimum and maximum values of the coefficients are: −2 and +2, respectively;

Number of samples: 200;

The sampling frequency is taken to be fs = 1 Hz;

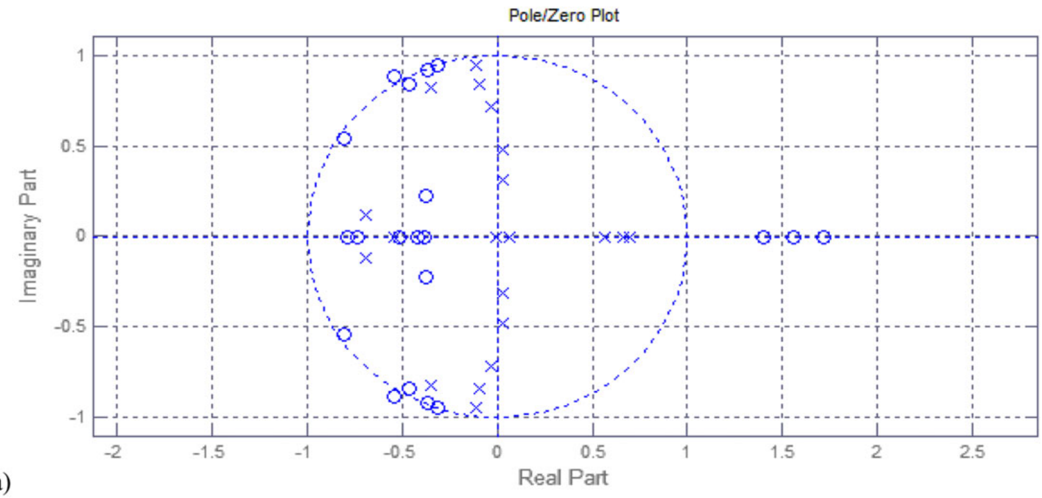
Pass band ripple: $\delta p = 0.01$;

Stop band ripple: $\delta s = 0.001$;

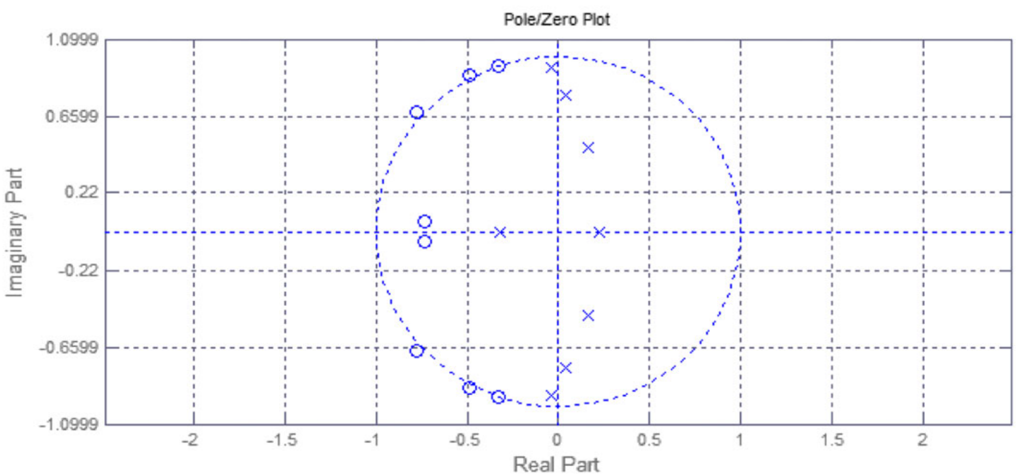
Example 1: (High pass filter) in this example, the design of 20th-order high pass digital filter is considered with the following IIR design specification: pass band border frequency $\omega_p = 0.4\pi$, stop band border frequency $\omega_p = 0.3\pi$. The magnitude and phase responses of this case are illustrated in Figure 2. A comparison between CSA optimal design and butter synthesis approach in magnitude and phase responses of the high-pass filter with order ord = 20 is presented is Figure 3.

Example 2: (High pass filter) in this example, the design of 8th-order high pass digital filter is considered with the same specification as specified above. The magnitude and phase responses of this example are shown in Figure 4. A comparison between CSA optimal design and butter synthesis approach in magnitude and phase responses of the high-pass filter with order ord = 8 is presented is Figure 5.

Figure 6 shows the pole-zero behaviour of the high-pass filter. It can be seen that the pole-zeros location of the designed filter falls within the unit circle. This demonstrates that the designed filter is stable employing the suggested approach. Figure 7 presents a comparison between CSA optimal design and butter synthesis approach in pole-zero positions of the low-pass filters.



(a)



(b)

FIGURE 12 Pole-zero position of the low-pass filters: A, ord = 20; B, ord = 8

Example 3: (Low pass filter) in this example, the design of 20th-order high pass digital filter is considered with the following IIR design specification: pass band edge frequency $\omega_p = 0.5\pi$, stopbandedge frequency$\omega_p = 0.6\pi$. The magnitude and phase responses of this example are shown in Figure 8. A comparison between CSA optimal design and butter synthesis approach in magnitude and phase responses of the high-pass filter with order ord = 20 is presented in Figure 9.

Example 4: (Low pass filter) in this example, the design of 8th-order low pass digital filter is considered with the same specification as specified in example 3. The magnitude and phase responses of this case are illustrated in Figure 10. A comparison between CSA optimal design and butter synthesis approach in magnitude and phase responses of the high-pass filter with order ord = 8 is presented in Figure 11.

In Figure 12, we have summarized the pole-zero behaviour of the low-pass filter. It can be seen that for the 8th order, the pole-zero location of the designed filter falls within the unit circle; this shows that the designed filter is stable. For the 20th, we can say the same just for few zeros; it is also stable using the proposed method. Figure 13 presents a comparison between CSA optimal design and butter synthesis approach in pole-zero positions of the low-pass filters. Table 1 reviews the numerical values of both high pass and low pass filters coefficients obtained with optimal design for order 8 and order 20. Figure 14 shows the evolution of the objective function with iterations, and we see its rapid convergence.

From the above obtained results, we can deduce the following advantages of CSA for digital IIR filter:

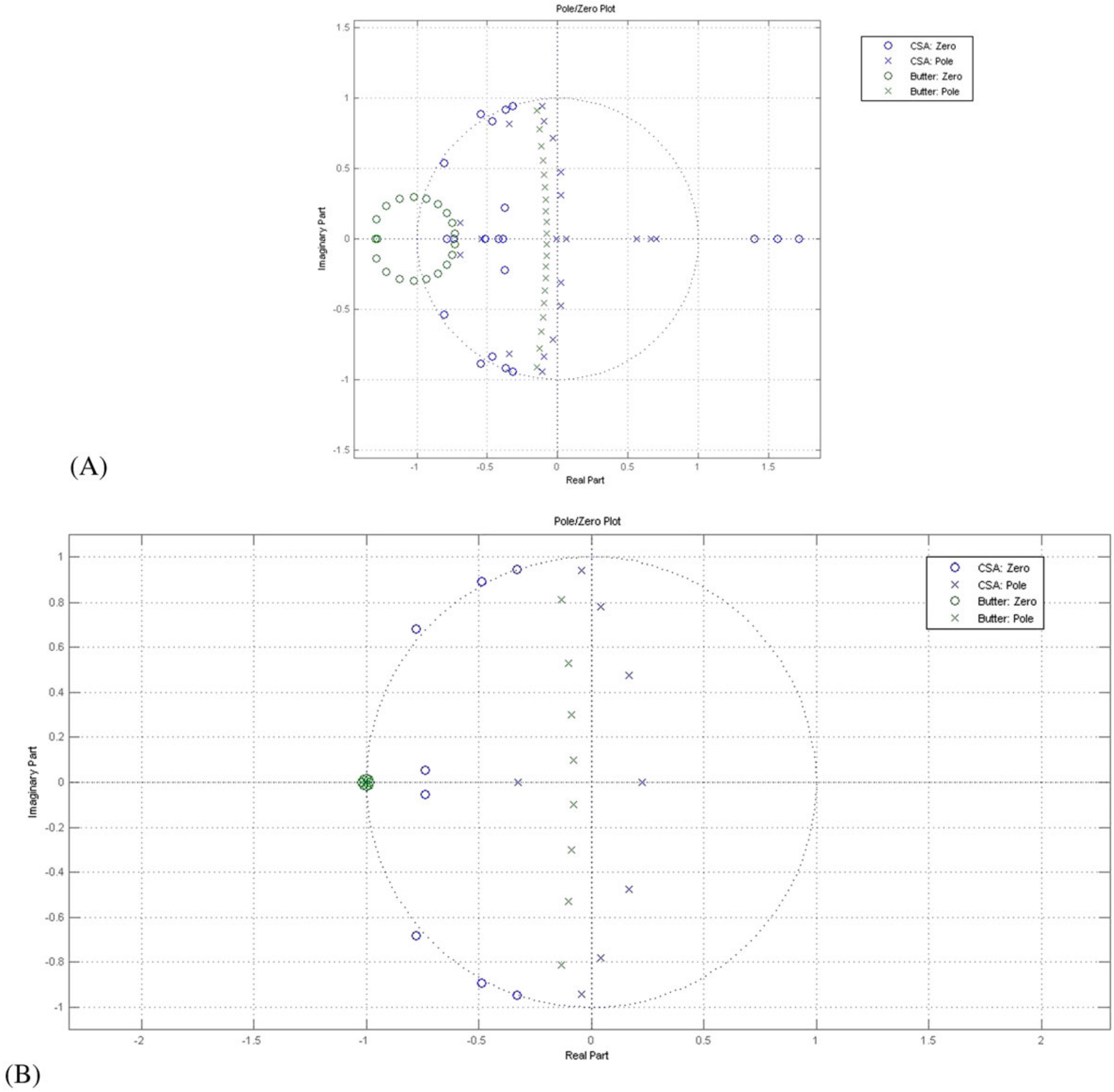


FIGURE 13 Comparison between CSA optimal design and butter synthesis approach in pole-zero positions of the low-pass filters: A, ord = 20; B, ord = 8

TABLE 1 Numerical values of both high and low pass filter coefficients obtained using CSA optimal design for ord = 8 and ord = 20

High-pass				Low-pass			
Order 20		Order 8		Order 20		Order 8	
Numerator	Denominator	Numerator	Denominator	Numerator	Denominator	Numerator	Denominator
0.0513185840	1	0.0732737641	1	−0.0052432750	1	0.0324720954	1
−0.3215860998	−1.3003821702	−0.4538806651	−1.8313497037	−0.0206145480	1.0055928369	0.1510802856	−0.2367936014
0.9901772956	3.1440709324	1.3228629106	2.8445676377	−0.0272409115	2.2286310051	0.3734199892	1.6383031079
−1.9579885479	−2.3301871321	−2.3548888776	−2.4653661378	0.0310335580	1.0955670218	0.6012956736	−0.3265282912
2.7874341181	3.6936406523	2.7905824825	1.6862730561	0.1808034982	0.5911175896	0.6819043233	0.7495552896
−3.1163313435	−2.3321894871	−2.2507344319	−0.7879915194	0.3048339229	−0.5411982328	0.5524849613	−0.0579551612
3.0326093213	3.2827730021	1.2069152785	0.2224536900	0.1603450845	−1.1404961434	0.3119978734	0.0514390714
−2.8491699287	−2.0056576373	−0.3941362009	−0.0618363198	−0.3637984534	−0.6679572307	0.1128018241	0.0271885405
2.6252357984	1.8480111579	0.0602264887	−0.0055582463	−0.9875451096	−0.4648262852	0.0196140888	−0.0101546706
−2.1609047388	−0.7253139196			−1.1639771058	0.0842418699		
1.4046056668	0.3067894368			−0.5803585436	0.1852963373		
−0.6186059172	0.0097759461			0.4847112114	0.1108733827		
0.1052070159	−0.0562453117			1.3779023268	0.0886182072		
0.0734743095	0.0232755434			1.6364492831	−0.0067660205		
−0.0630654080	−0.0097223700			1.3063224576	−0.0046555197		
0.0185610966	−0.0015237594			0.7615944073	−0.0053818797		
0.0008764245	0.0005720475			0.3281170278	−0.0048975287		
−0.0020867167	0.0001448390			0.1021768945	−0.0000242212		
0.0003796674	0.0000752190			0.0217768679	−0.0003926854		
0.0000490827	−0.0000096471			0.0028412781	0.0000228679		
−0.0000157742	−0.0000044339			0.0001709539	0.0000002371		

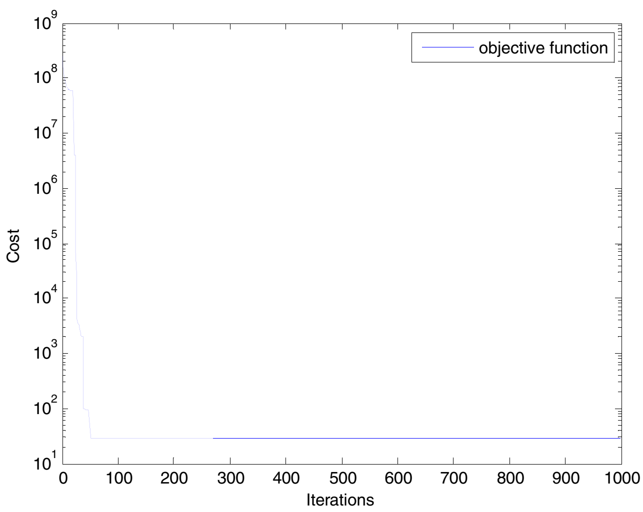


FIGURE 14 The objective function behaviour with CSA iteration

- CSA is rather simple with only two adjustable parameters (flight length and AP), which in turn makes it very attractive for such application needing both suppleness in tuning and accurate results.
- In comparison with other meta-heuristic optimization algorithms cited in literature, CSA has fewer parameters to adjust and hence is easier to implement.
- CSA achieves the best stop band attenuation and minimum transition width as compared with those of butter design approach.
- It gives the best magnitude response as compared with butter design approach.
- The identical length filter is efficiently designed with 85% and 80% less time than classical approach, and the resultant filters can be implemented in hardware for different applications with a significant amount of accuracy.
- It keeps computational time in addition to cost, and thus it is a further efficient and elegant approach of optimization for the design of IIR filters. Finds the solution of the investigated problems in around few seconds (less execution time).
- It converges to a much lower fitness value.

7 | CONCLUSION

In this work, we try to design an optimal IIR filter. The optimality of the design is based on minimizing the error in the sense of weighted least ℓp-norm. A novel meta-heuristics Algorithm called CSA is introduced. By decrease of the AP value, CSA tends to perform the search on a limited region wherever a current good result is reached in this region. On the other hand, by rising of the AP value, the probability of seeking the neighborhood of present high-quality results diminishes and CSA tends to investigate the search room on a total range (randomization). Consequently, employing big values of AP raises diversification. The results obtained show the effectiveness of this design approach and thus approaches the best previous designs in comparison with a conventional IIR designed using butter synthesis approach. To assure better stability and causality, any poles or zeros that lie outside of the unit circle are reflected back inside. Thus, it is proved that CSA is superior to some other reported meta-heuristic techniques for optimal IIR filter design.

ORCID

Slami Saadi http://orcid.org/0000-0001-8091-5232

REFERENCES

1. Oppenheim AV, Schafer RW, Buck JR. *Discrete-time Signal Processing*. Second ed. Prentice Hall; 1999.
2. Cousseau JE, Werner S, Donate PD. Factorized all-pass based IIR adaptive notch filters. *IEEE Trans Signal Process*. 2007;55(11):5225-5236.
3. Kaur R, Patterh MS, Dhillon JS. Design of optimal L1 stable IIR digital filter using hybrid optimization algorithm. *Int J Comput Appl* (0975 – 8887). 2012;38(2, January).
4. Saha SK, Kar R, Mandal D, Ghoshal SP. Digital stable IIR low pass filter optimization using PSO-CFIWA. In: *2012 1st International Conference on Recent Advances in Information Technology (RAIT)*; 2012:196-201.
5. Benvenuto N, Marchesi M. Applications of simulated annealing for the design of digital filters. *IEEE Trans Signal Process*. 1992;40(2):323-331.
6. Chen S, Istepanian RH, Luk BL. Digital IIR filter design using adaptive simulated annealing. *Digital Signal Process*. 2001;11(3):241-251.
7. Nambiar R, Tang CKK, Mars P. Genetic and learning automata algorithms for adaptive digital filters. *Proc ICASSP 1992*. 1992;4:41-44.
8. Wilson PB, Macleod MD. Low implementation cost IIR digital filter design using genetic algorithms. Workshop on Natural Algorithms in Signal Processing, Chelmsford, Essex, UK; 1993:4/1–4/8
9. White MS, Flockton SJ. Genetic algorithms for digital signal processing. *Lect Notes Comput Sci*. 1994;865:291-303.
10. Ng SC, Leung SH, Chung CY, Luk A, Lau WH. The genetic search approach: a new learning algorithm for adaptive IIR filtering. *IEEE Signal Process Mag*, November Issue. 1996;13(6):38-46.
11. Liang L, Ahmadi M, Ahmed M, Wallus K. Design of canonical signed digital filters using genetic algorithms. *IEEE Trans Signal Process*. 2003;3(1):2043-2047.
12. Ahmad SU, Antoniou A. Design of digital filters using genetic algorithms. *IEEE Trans Signal Process*. 2006;1(1):1-9.

13. Tang W, Shen T. Optimal design of FRM-based FIR filters by using hybrid Taguchi genetic algorithm. *Int Conf Green Circuits Syst (ICGCS-2010)*. 2010;392-397.
14. Ahmad SU, Antoniou A. A hybrid genetic algorithm for the design of IIR digital filters. 2007 Canadian conference on electrical and computer engineering, vol. 2012, no. August; 2007:286-292.
15. Kaur R, Patterh MS, Dhillon JS. Digital IIR filter design using real coded genetic algorithm. *Int J Inf Technol Comput Sci (IJITCS)*. 2013;5(7):27-35. https://doi.org/10.5815/ijitcs.2013.07.03
16. Krusienski DJ, Jenkins WK. Particle swarm optimization for adaptive IIR filter structures. *Proc Congr Evol Comput*, Jun. 2004;965-970.
17. Fang W, Sun J, Xu W-B. Design IIR digital filters using quantum-behaved particle swarm optimization. *Proc. 2nd Int. Conf. Natural Computation*, Xian, China, 24–28 September, Part II. 2006;637-640.
18. Das S, Konar A. A swarm intelligence approach to the synthesis of two-dimensional IIR filters. *Eng Appl Artif Intel*. 2007;20(8):1086-1096.
19. Chen S, Luk BL. Digital IIR filter design using particle swarm optimisation. *Int J Model Identif Control*. 2010;9(4):327-335.
20. Jiang S, Wang Y, Ji Z. A new design method for adaptive IIR system identification using hybrid particle swarm optimization and gravitational search algorithm. *Nonlinear Dyn*. 2015;79(4):2553-2576.
21. Karaboga N. A new design method based on artificial bee colony algorithm for digital IIR filters. *JFrankl Inst*. 2009;346(4):328-348.
22. Karaboga N, Cetinkaya MB. A novel and efficient algorithm for adaptive filtering: artificial bee colony algorithm. *Turk J Electr Eng Comput Sci*. 2011;19(1):175-190.
23. Dai C, Chen W, Zhu Y. Seeker optimization algorithm for digital IIR filter design. *IEEE Trans Ind Electron*. 2010;57(5):1710-1718.
24. Panda G, Pradhan PM, Majhi B. IIR system identification using cat swarm optimization. *Expert Syst Appl*. 2011;38(10):12671-12683.
25. Saha SK, Kar R, Mandal D, Ghoshal SP. A new design method using opposition-based BAT algorithm for IIR system identification problem. *Int J Bio-Inspired Comput*. 2013;5(2):99-132.
26. Kalinli A, Karaboga N. Artificial immune algorithm for IIR filter design. *Eng. Appl. Artif.Intell.* 2005;18(8):919-929.
27. Karaboga N. Digital IIR filter design using differential evolution algorithm. *EURASIP J Appl Signal Process*. 2005;8:1269-1276.
28. Mandal S, Ghoshal SP, Kar R, Mandal D. Differential evolution with wavelet mutation in digital FIR filter design. *J Optim Theory Appl*. 2012;155(1):315-324.
29. Rashedi E, Nezamabadi-Pour H, Saryazdi S. Filter modeling using gravitational search algorithm. *Eng Appl ArtifIntell*. 2011;24(1):117-122.
30. Saha S, Kar R, Mandal D, Ghoshal S. Optimal IIR filter design using gravitational search algorithm with wavelet mutation. *J King Saud Univ—Comput Inform Sci*. 2015;27(1):25-39.
31. Karaboga N, Kalinli A, Karaboga D. Designing IIR filters using ant colony optimization algorithm. *J Eng Appl ArtifIntell*. Apr. 2004;17(3):301-309.
32. Hlaing ZCSS, Khine MA. Solving traveling salesman problem by using improved ant colony optimization algorithm. *Int J Inf Educ Technol*. 2011;1(5):404-409.
33. Kalinli A, Karaboga N. A new method for adaptive IIR filter design based on Tabu search algorithm. *Int J ElectronCommun(AEÜ)*. 2005;59(2):111-117.
34. Upadhyay P, Kar R, Mandal D, Ghoshal SP. A new design method based on firefly algorithm for IIR system identification problem. *J King Saud Univ Eng Sci*. 2014. https://doi.org/10.1016/j.jksues.2014.03.0015
35. Kalinli A, Karaboga N. New method for adaptive IIR filter design based on tabu search algorithm," to appear in. *AEU Int J Electron Commun*. 2004.
36. Antoniou A. *Digital Signal Processing: Signals, Systems, and Filters*. New York: McGraw-Hill; 2005.
37. Askarzadeh A. A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm. *Comput Struct*. 2016;169:1-12. https://doi.org/10.1016/j.compstruc.2016.03.001
38. Antoniou A. Improved minimax optimisation algorithms and their application in the design of recursive digital filters. *Proc Inst Elect Eng, part G*. Dec. 1991;138:724-730.

How to cite this article: Ghibeche Y, Saadi S, Hafaifa A. Optimal design of IIR filters based on least ℓ_p-norm using a novel meta-heuristic algorithm. *Int J Numer Model*. 2018;e2480. https://doi.org/10.1002/jnm.2480