Circuits Syst Signal Process (2009) 28: 899–911
DOI 10.1007/s00034-009-9128-1

Frequency Domain FIR Filter Design Using Fuzzy Adaptive Simulated Annealing

Hime A. Oliveira Jr. · Antonio Petraglia · Mariane R. Petraglia

Received: 8 March 2008 / Revised: 10 November 2008 / Published online: 21 August 2009

Abstract An alternative approach to digital filter design is presented. The overall technique is as follows: Starting from frequency domain constraints and a parameterized expression of the filter family under adaptation, a corresponding training set is created, an error function is synthesized and a global minimization process is executed. At the end, the point that minimizes globally the particular cost function at hand determines the optimal filter. The adopted numerical optimization algorithm is based upon the well-known simulated annealing paradigm and its implementation is known as fuzzy adaptive simulated annealing. Although it is used in this paper to fit FIR filters to frequency domain specifications, the method is suitable to application in other problems of digital filter design, where the matter under study can be stated as finding the global minimum of a numerical function of filter parameters. Design examples are shown to verify the effectiveness of the proposed approach.

Keywords Simulated annealing · Fuzzy logic · FIR digital filters

1 Introduction

A number of well succeeded and firmly established methods of digital FIR filter design and its complex associations have been proposed (e.g., [1, 2, 7, 10, 12, 14]).

H.A. Oliveira Jr. · A. Petraglia (✉) · M.R. Petraglia
Program of Electrical Engineering, COPPE/UFRJ, Federal University of Rio de Janeiro, Rio de Janeiro, Brazil
e-mail: antonio@pads.ufrj.br

H.A. Oliveira Jr.
e-mail: hime@engineer.com

M.R. Petraglia
e-mail: mariane@pads.ufrj.br

However, this is a research intensive area, aiming at obtaining more general and innovative techniques that are able to face new and complex engineering problems of great relevance today. Alternative approaches, such as neural networks, genetic algorithms, particle swarm optimization, and other tools related to computational intelligence [4, 8, 9, 11, 17], have been devoted to the synthesis of design methods capable of satisfying constraints which would be unattainable, if treated with the aforementioned conventional techniques. In some cases, such initiatives were successful and showed better performance indices than the conventional approaches. However, there are a few weak points associated to these new methods, as increased computational cost and nonexistence of theoretical proof of convergence to global optimum in sufficiently general conditions. Consequently, there is a need to search for more pervasive methods, capable of overcoming such weaknesses.

This paper presents an alternative approach that takes advantage of the power of the well-known stochastic global optimization technique called simulated annealing. Owing to the efficient implementation known as adaptive simulated annealing (ASA) [5] and its modified fuzzy controlled version, fuzzy ASA [13], promising results are shown here in approximating arbitrary magnitude and phase specifications. Although the algorithm is adequate to applications in any kind of parameterized filter family, we have chosen to focus on real-coefficient FIR filters, in view of their importance in engineering practice. The proposed approach follows the methodology applied in most filter design techniques: the imposed filter constraints are lumped into an objective or cost function of the parameters (in the present case, nonzero values of the filter impulse response), so that globally minimizing such a numerical function means to find the optimal filter parameters, compatible with the design conditions.

To validate the proposed method, we have applied it to the design problems presented in [8], and compared the respective results. Based upon a neural network framework, that paper presents a good and comprehensive set of results, and states arguments for the superiority of the algorithm therein described to the previous ones. As will be seen in Sect. 5, the new ASA-designed filters proposed here present better performance than those reported in [8], and hence to the other techniques therein considered for comparison.

2 Problem Statement

The frequency response of a linear and time-invariant real-coefficient FIR filter with impulse response $h(n)$, $n = 0, 1, \dots, N-1$, and order N is given by

$$\begin{aligned} H(\omega) &= \sum_{n=0}^{N-1} h(n) \exp\bigl(-j\omega n\bigr) \\ &= A(\omega) \exp\bigl(j\rho(\omega)\bigr), \end{aligned} \tag{1}$$

where $A(\omega)$ and $\rho(\omega)$ are the filter amplitude and phase responses, respectively.

Our purpose in this work is to show how to approximate arbitrary responses and/or frequency-domain constraints, by applying specific global nonlinear minimization techniques and (not necessarily uniform) frequency sampling. So, we could approximate a given amplitude response, and, at the same time, a specific group delay curve, for instance. For that, it is necessary to define a finite set of frequencies $\{\omega_i,\ i = 1, 2, \ldots, L\}$ belonging to the interval $[0, \pi]$ and build a cost function reflecting the deviation between the desired and obtained quantities at each frequency ω_i, as a function of the filter parameters $\{h(n),\ n = 0, 1, \ldots, N - 1\}$. In this manner, the particular values of $h(n)$ that globally minimize the corresponding cost function will determine the optimal filter. Hence, the critical part of the whole task is the global minimization process, which, in this paper, will be carried out by means of the fuzzy ASA algorithm. Such a task is not trivial because the typical functions are not well-behaved in general.

To demonstrate the effectiveness of the suggested approach, we will show two design examples in Sect. 4. A general view of the fuzzy ASA algorithm is provided in the next section.

3 Fuzzy Adaptive Simulated Annealing

The task of global minimization of numerical functions has paramount importance in several areas of knowledge. In practical cases, the function to be minimized shows itself in the form of a cost measure that varies with several parameters and is subject to certain constraints, imposed by its environment. When the objective function is well-behaved, there are several methods to find the point at which it attains the minimum value, while satisfying the constraints [15].

The difficulties arise when the given function presents several local minima, thus making the final result dependent on the starting point. Unfortunately, most real problems lead to very complex objective functions that are nonlinear, discontinuous, multi-modal, and multi-dimensional among other undesired properties. To solve such a class of problems, stochastic methods seem to be a good, and sometimes the only, alternative. Genetic algorithms and simulated annealing are among the most popular approaches to stochastic global optimization.

The difficulty of the simulated annealing approach is related to the speed of convergence, whereas that of the genetic approach is its inability of reaching the global optimum under general conditions. Pure annealing methods, on the other hand, have achieved convergence to the global minimum with probability 1, but the performance presented by most implementations is not very encouraging. In spite of this drawback, researchers have found ways to overcome the limitations of original annealing schemes, leading to the so-called very fast simulated re-annealing (VFSR) approach, which is a sophisticated and rather effective global optimization method [5, 6]. The VFSR technique is particularly well suited to applications involving neuro-fuzzy systems and neural network training because of its superior performance and simplicity.

Adaptive simulated annealing is an implementation of VFSR that has the benefits of being publicly available, parameterized and well-maintained. Besides, the ASA approach shows itself as an alternative to genetic algorithms, according to the published benchmarks, which demonstrate its effectiveness [12]. Unfortunately, stochastic global optimization algorithms share a few bad characteristics, such as large periods of poor improvement in their way to a global extreme. In simulated annealing implementations, that is mainly due to the "cooling" schedule, whose speed is limited by the characteristics of probability density functions, which are employed with the purpose of generating new candidate points. Annealing is carried out through the simulation of several Markov chains, each of them associated to one temperature value. The aim is to scan the objective function landscape to devise promising regions that could be additionally explored [3].

In this manner, if we choose to employ the so-called Boltzmann annealing, the "temperature" has to be lowered at a maximum rate of $T(k) = T(0)/\ln(k)$. In the case of fast annealing, the schedule becomes $T(k) = T(0)/k$, if assurance of convergence with probability 1 is to be maintained, resulting in a faster schedule. The approach based on ASA has an even better default scheme given by

$$T_i(k) = T_i(0)\exp\left(-C_i k^{1/D}\right) \tag{2}$$

because of its improved generating distribution. The constant C_i is a user-defined parameter, and D is the number of independent variables of the function under minimization (dimension of the domain). Note that subscripts indicate independent evolution of temperatures for each parameter dimension. In addition, it is possible to take advantage of simulated quenching, resulting in

$$T_i(k) = T_i(0)\exp\left(-C_i k^{Q_i/D}\right), \tag{3}$$

where Q_i is termed the quenching parameter. By attributing to Q_i values greater than 1 we obtain a gain in speed, but the convergence to a global optimum is no longer assured [5]. Such a procedure could be used for higher-dimensional parameter spaces, when computational resources are scarce. The internal structure of a well-succeeded approach to accelerate the ASA algorithm, using a simple Mamdani fuzzy controller that dynamically adjusts certain user's parameters related to quenching, is described in [13]. It is shown that, by increasing the algorithm perception of slow convergence, it is possible to speed it up significantly, thereby reducing considerably (perhaps eliminating) the user's task of parameter tuning.

In the following sections, a design technique is advanced employing the above concept of fuzzy ASA to produce optimum FIR filters.

4 Proposed Method

Given a complex function $H_d(\omega) = A_d(\omega)\exp(j\rho_d(\omega))$, our target is to find the real coefficients $h(n),\ n = 0, \ldots, N-1$ of a linear, time-invariant FIR digital filter, whose frequency response approximates in an optimal manner a reference curve, by means of the following algorithm:

(1) Build the training set by computing the values of the function $H_d(\omega)$ at the frequencies $\omega_i,\ i = 1, \ldots, L$, where $\omega_i \in [0, \pi]$ and L is the number of samples. Note that the values of ω_i can be chosen non-uniformly spaced.

(2) Construct the cost function

$$J(h) = \sum_{i=1}^{L} W(i)\left(H_d(\omega_i) - H(\omega_i)\right)^2, \tag{4}$$

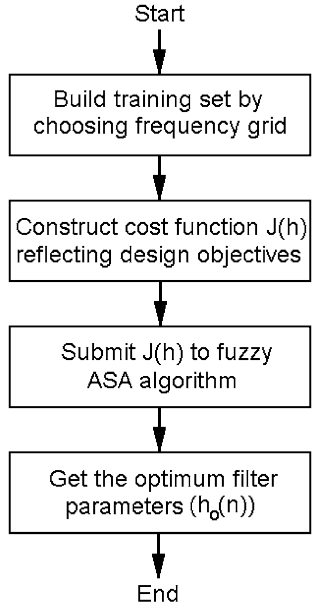


Fig. 1 Flowchart of the filter design approach using fuzzy ASA algorithm

where $H(\omega)$ is the frequency response of a generic LTI FIR filter with real coefficients $h(n),\ n = 0, 1, \ldots, N - 1$, $W(i)$ is a (time-varying) real-valued weighting function, used for numerical conditioning, not necessarily differentiable or continuous.

(3) Submit the function $J(h)$ to the global optimization algorithm (in our case, fuzzy ASA) and obtain the minimizing parameters $h_o(n),\ n = 0, \ldots, N - 1$. Such values will determine the optimal filter in the weighted L_2 sense.

A flowchart of the above design procedure is shown in Fig. 1.

Although the main global optimization (fuzzy ASA) algorithm is invariant, each design problem demands the construction of a specific cost function reflecting particular constraints of a given kind of filter—that is the role of the function $J(h)$. The global optimization of the objective function is carried out by considering the filter coefficients as coordinates of points in R^N, the Euclidean N-dimensional space, which contains the whole class of length-N FIR digital filters. The present optimization approach, just like many other evolutionary methods and even neural ones, regards $J(h)$ as the only "hint" to evolve coefficient vector h towards the desired end. The update of its components is realized through the generate-and-accept/reject cycle of simulating annealing. Hence, the main process repeatedly calls a module that returns values of $J(h)$ at different points of R^N (candidate filter parameters) and the fitness of particular realizations guides the optimization process. Just like other methods realizing the simulated annealing paradigm, the ASA implementation aims to approximate a very special probability distribution function—the one with modes coinciding with global minimizers of the cost function. In other words, the relationship between fuzzy ASA and $J(h)$ is that the first strives for globally minimizing the numerical value of the second, as in the other papers where genetic algorithms or PSO engines are used—fuzzy ASA is simply one algorithm aimed at minimizing generic functions. When fuzzy ASA stops, the resulting filter parameter vector (say, $h_o(n)$), is its best FIR filter design, taking into account that it has got the minimum value $J(h_o)$. To get there within an adequate time interval, the temperature schedule of (3) shows itself as an effective and efficient tool, leading to very satisfactory results.

For practical reasons, each simulated Markov chain was allowed to evolve in a compact subset of R^N and we had to determine its shape and boundary. Once more we are helped by specific information—we have chosen a hyper-rectangle, symmetric around the origin and formed by the N-dimensional Cartesian product of the $[-1.5, 1.5]$ closed interval, considering the fact that "interesting" filter coefficients are typically found in this range.

The error function weighting was performed by taking into account particular shapes assumed by approximation error curves along the frequency interval $[0, \pi]$. The adopted strategy was basically the following for all design cases:

(1) Let the underlying annealing process evolve without weighting at all, until stabilization of error curve shape—this procedure is important to guide the construction of the weight vector that will be used in the lowering of present error peaks. As this step is specific of each setting, it is advisable to be conservative and set up a sufficiently larger number of acceptances. In practice, our control parameter is the number of function evaluations (500,000 evaluations proved adequate in all cases).
(2) Sample the established error curve in all points of the training set.
(3) Normalize the above sampled data, dividing each element by the largest error.
(4) Proceed in the minimization process, activating the weighting mechanism by means of a fuzzy inference system that, at each training frequency, finds the adequate factor for that particular error value, tending to amplify large errors and attenuate small ones.
(5) In this manner, it is possible to shape the error curve without being limited by particular closed form functions, and to spread the error values more uniformly along the corresponding interval, thereby lowering the upper bound of the overall approximation error.
(6) It is worth noting that the proposed approach can be useful even in situations when we do not have any theoretical model, but just experimental data, be it in frequency or in time domain (this is a good property of black box modeling). Hence, this technique is able to handle nonlinear modeling, as well.

Convergence to a global minimum is guaranteed by results present in [9, 12, 14] and is inherent to the general simulated annealing pattern, considering that (4) satisfies conditions there imposed and minimization takes place over a compact subset of R^N. In addition, the time-variability of the weighting vector does not void the premises for theoretical convergence, taking into account the long periods between value changes, needed for stochastic stabilization. So, although we are proposing to solve a specific modeling problem, the general convergence issue is solved, thanks to the adequate "boundary conditions" and previous theoretical results.

The following examples show the results of the application of the proposed approach to two design problems, and comparisons with an alternative method based on neural networks.

5 Illustrative Design Examples

To illustrate the effectiveness of the proposed approach, we consider filter examples from [8], which present distinct and demanding design challenges. In what follows, we define the indexes therein adopted and compare our results to those presented in [8], using the performance figures originally employed there, namely the peak magnitude of complex error

$$\delta_p = \max\left\{\left|H_d(\omega) - H(\omega)\right| : \omega \in \Omega_p \cup \Omega_s\right\} \tag{5}$$

and the peak passband group delay error

$$\delta_\tau = \max\left\{\left|\tau_d(\omega) - \tau(\omega)\right| : \omega \in \Omega_p\right\}, \tag{6}$$

where Ω_p and Ω_s denote the passband and stopband, respectively, of the desired frequency response.

5.1 Example 1: Design of a Bandpass Filter

Here, the task is to design a bandpass FIR filter with length $N = 31$, amplitude response

$$A_d(\omega) = \begin{cases} 1, & \omega \in \Omega_p = [0.30, 0.56\pi], \\ 0, & \omega \in \Omega_s = [0, 0.20\pi] \cup [0.66\pi, \pi] \end{cases} \tag{7}$$

and phase response $\rho_d(\omega) = -12\omega$, which corresponds to a group delay $\tau_d(\omega) = 12$ samples in the passband. The coefficients of the designed filter are given in Table 1. Figures 2(a) and 2(b) display, respectively, the amplitude frequency response in dB, $A(\omega)$, and the group delay, $\tau(\omega)$, produced by the technique advanced in this paper.

Table 1 Coefficients of the bandpass filter of Example 1

$h(0) = -0.001837914476369$	$h(16) = 0.073708129357635$
$h(1) = 0.009935750560736$	$h(17) = 0.031666242086951$
$h(2) = -0.019251362346424$	$h(18) = 0.006345985585397$
$h(3) = -0.049861827100943$	$h(19) = 0.055649617634259$
$h(4) = 0.010528920995195$	$h(20) = 0.011587428546593$
$h(5) = 0.055915070756933$	$h(21) = -0.044244284103041$
$h(6) = 0.006393823781558$	$h(22) = -0.015580084869688$
$h(7) = 0.032228526486936$	$h(23) = 0.001475111214787$
$h(8) = 0.073932276963528$	$h(24) = -0.011019340846704$
$h(9) = -0.124760866532111$	$h(25) = 0.005664410859877$
$h(10) = -0.257499929241588$	$h(26) = 0.018058207255140$
$h(11) = 0.074193478705421$	$h(27) = 0.002728094197810$
$h(12) = 0.361484282182677$	$h(28) = -0.003688819770780$
$h(13) = 0.074213670077878$	$h(29) = -0.000193237176018$
$h(14) = -0.258049931324484$	$h(30) = -0.002536687445088$
$h(15) = -0.125161532135573$	

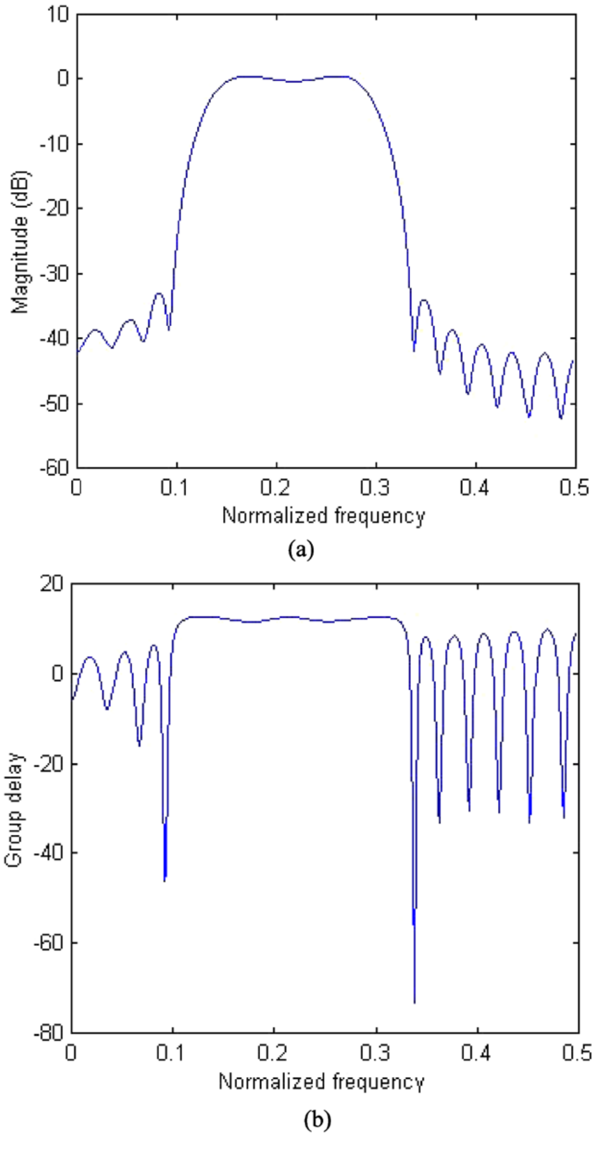


Fig. 2 Frequency responses for the bandpass filter of Example 1: (**a**) Amplitude; (**b**) Group delay

Table 2 Comparison of performance indexes—Example 1

Method	Peak magnitude (δ_p)	Group delay error (δ_τ)
This paper	6.089e−02	0.5489
Ref. [8]	6.266e−02	1.080

Table 2 summarizes the achieved performance indexes and compares them to those obtained in [8]. These results show the improvement introduced by the proposed approach with respect to other well-succeeded techniques.

Table 3 Coefficients of the allpass equalizer of Example 2; $h(n) = 0$ for n odd (see text)

$h(0) = -4.203805455009930e-04$	$h(32) = 8.523424042303883e-02$
$h(2) = -3.685652615134307e-04$	$h(34) = 3.359228009317205e-02$
$h(4) = -3.502164230926212e-04$	$h(36) = 1.454205724613560e-02$
$h(6) = -3.666319672442465e-04$	$h(38) = 7.094900560162374e-03$
$h(8) = -4.066932095494393e-04$	$h(40) = 3.842872804071504e-03$
$h(10) = -4.685477427932168e-04$	$h(42) = 2.289897391547230e-03$
$h(12) = -7.014383800246077e-04$	$h(44) = 1.463033805840513e-03$
$h(14) = -9.896010677664139e-04$	$h(46) = 9.926360012364303e-04$
$h(16) = -1.570226138650236e-03$	$h(48) = 7.018642536107877e-04$
$h(18) = -5.275019359728725e-04$	$h(50) = 4.714267850738611e-04$
$h(20) = -2.412287634084593e-02$	$h(52) = 4.106181303323064e-04$
$h(22) = 1.395102919207290e-01$	$h(54) = 3.732036513588636e-04$
$h(24) = -5.699249973981046e-01$	$h(56) = 3.512385395700095e-04$
$h(26) = 5.977756217915143e-01$	$h(58) = 3.672223488506539e-04$
$h(28) = 4.905200826608643e-01$	$h(60) = 4.066262600110443e-04$
$h(30) = 2.202753879678499e-01$	

Table 4 Comparison of performance indexes—Example 2

Method	Peak magnitude (δ_p)	Group delay error (δ_τ)
This paper	1.467e−03	8.873e−02
Ref. [8]	1.588e−03	9.830e−02

5.2 Example 2: Design of a Sine-Delay Allpass Phase Equalizer

In this case, the task is to design an all-pass FIR filter of length 61, amplitude response $A_d(\omega) = 1$ for all ω (no stopband), and phase response $\rho_d(\omega) = -30\omega - 2\pi\cos(\omega)$, or equivalently, a group delay $\tau_d(\omega) = 30 - 2\pi\sin(\omega)$. The filter coefficients are listed in Table 3, and the amplitude (in dB) and group delay responses are shown in Figs. 3(a) and 3(b), respectively. As indicated in Table 3, half of the filter coefficients are zero, that is, $h(n) = 0$ for n odd, because the phase is an odd function of the frequency about $\omega = \pi/2$. The performance indexes are presented in Table 4, along with those obtained in [8], for comparison, and indicate the better overall performance of the fuzzy ASA approach.

5.3 Example 3: Design of Hilbert Transformer

This final example deals with the design of a wide-band Hilbert transformer having length 31, normalized low passband and stopband edges at 0.05 and 0.5, respectively, and a group delay of 11.5 samples. The filter coefficients and are presented in Table 5, and the amplitude and group delay responses are shown in Figs. 4(a) and 4(b), respectively. Again, as can be noticed in Table 6, the proposed method yields better results than what has been obtained with other approaches.

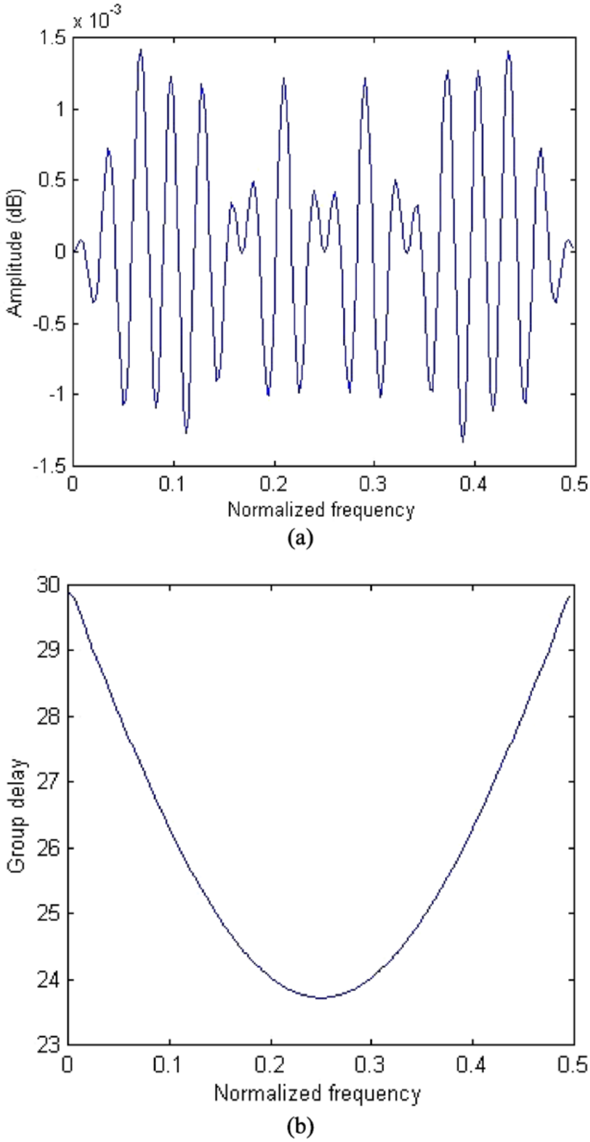


Fig. 3 Frequency responses for the allpass filter of Example 2: (**a**) Amplitude; (**b**) Group delay

5.4 Comments

In the first example, the training set had 110 elements, in the second one, 100 elements, and in the third one, 81 elements. The frequency grid was uniformly distributed along the $[0, \pi]$ interval. By contrast, in [8] the recommended cardinalities were 310, 610 and 310, respectively (31×10, 61×10 and 31×10). The proposed method also presented superior performance when applied to the other examples considered in [8].

Table 5 Coefficients of the Hilbert transformer of Example 3

$h(0) = -1.067498402126269\text{e}{-03}$	$h(16) = 8.088234899873083\text{e}{-02}$
$h(1) = -2.292876910821742\text{e}{-03}$	$h(17) = 6.443986102801079\text{e}{-02}$
$h(2) = -4.205474523564142\text{e}{-03}$	$h(18) = 5.200523688763421\text{e}{-02}$
$h(3) = -7.067728770631233\text{e}{-03}$	$h(19) = 4.207804646689279\text{e}{-02}$
$h(4) = -1.124070199496255\text{e}{-02}$	$h(20) = 3.390073951622692\text{e}{-02}$
$h(5) = -1.727424020748538\text{e}{-02}$	$h(21) = 2.705186589338856\text{e}{-02}$
$h(6) = -2.605776712608001\text{e}{-02}$	$h(22) = 2.128647013965200\text{e}{-02}$
$h(7) = -3.921742220559379\text{e}{-02}$	$h(23) = 1.644387319876482\text{e}{-02}$
$h(8) = -6.017734383381798\text{e}{-02}$	$h(24) = 1.241476971640638\text{e}{-02}$
$h(9) = -9.777335594167726\text{e}{-02}$	$h(25) = 9.108423876203203\text{e}{-03}$
$h(10) = -1.843389043046776\text{e}{-01}$	$h(26) = 6.447583616208436\text{e}{-03}$
$h(11) = -6.108838520361685\text{e}{-01}$	$h(27) = 4.359358816587151\text{e}{-03}$
$h(12) = 6.598137962219037\text{e}{-01}$	$h(28) = 2.771492663960441\text{e}{-03}$
$h(13) = 2.324983196950823\text{e}{-01}$	$h(29) = 1.608869923657880\text{e}{-03}$
$h(14) = 1.444211798341833\text{e}{-01}$	$h(30) = 8.035104048487328\text{e}{-04}$
$h(15) = 1.046318500977753\text{e}{-01}$	

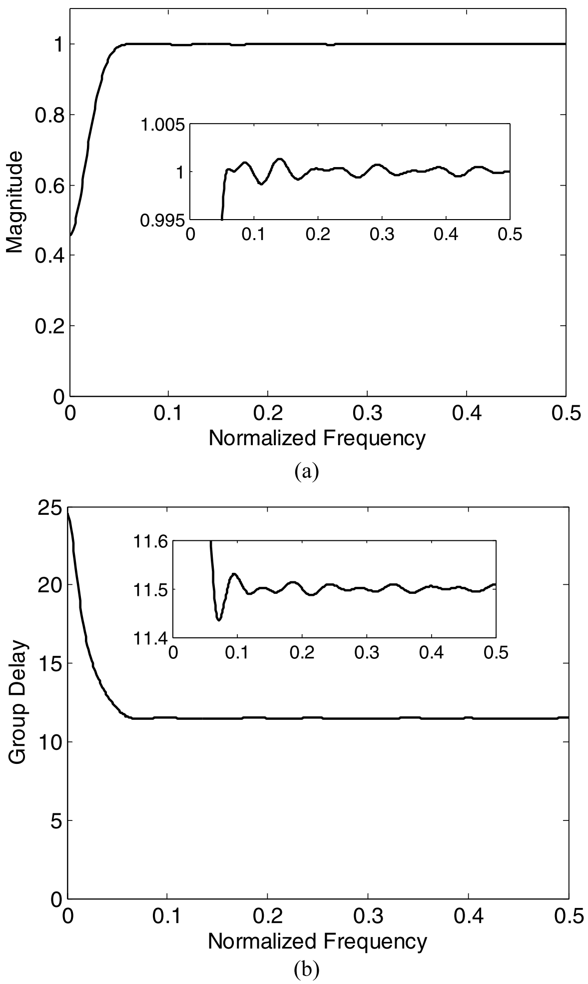
Table 6 Comparison of performance indexes—Example 3

Method	Peak magnitude (δ_p)	Group delay error (δ_τ)
This paper	4.857e−03	6.505e−02
Ref. [8]	1.050e−02	7.560e−02

In the above examples, each cost function consisted of the weighted sum of complex errors at the sampling points, with the weighting function tending to magnify large errors and reduce small ones by means of a simple Mamdani fuzzy inference system [16]. This fuzzy logic based weighting mechanism can help to improve dramatically final results, and is particularly appropriate whenever initial error curves present maximum to minimum ratio values substantially greater than 1 (say 3, or more).

Another notable aspect of the proposed method is its adaptive behavior—differently from the majority of existing approaches, the (discrete) weighting function is built "on the fly" according to the configuration presented by the specific error graph during the initial stage of the optimization process and without direct human intervention. Traditional practice consists in pre-establishing certain weighting mechanisms before starting the fitting procedure, keeping it fixed until its end and, in case of unsatisfactory results, tuning the prescribed weighting component until reaching the desired performance. With the approach advanced in this paper, on the other hand, the designer's task is not to synthesize particular weighting functions anymore, but to build instead adequate weighting strategies inside the fuzzy inference system, so as to favor specific and desired performance indexes, as implemented in the above design examples.

Fig. 4 Frequency responses for the Hilbert transformer of Example 3: (**a**) Magnitude; (**b**) Group delay



6 Computational Requirements

All simulations were carried out using microcomputers with Core 2 CPUs (6600 @ 2.40 GHz) and 512 Mb of RAM, the algorithms were coded in C++ and compiled with Borland C++ 5.5. Example 1 took 1 minute and 39 seconds to complete, Example 2 took 4 minutes and 42 seconds, and Example 3 took 5 minutes and 17 seconds.

Taking into account that we have used a stochastic global optimization method to accomplish the design tasks, the results obtained are considered very good in terms of computational efficiency, given the complexity of the cost function, the dimensionality of parameter space and the peculiarities of simulated annealing processes.

7 Conclusion

This work presented a new approach for designing linear FIR filters by using nonlinear stochastic global optimization based on simulated annealing techniques conjugated to a fuzzy logic based method for weighting function synthesis. When compared to previously published techniques, the method showed good performance and is adequate for use in other related design problems, be them linear or not, where it is feasible to express problem constraints as an overall cost function. This was made possible by the large applicability and effectiveness of the fuzzy ASA implementation, which demanded just a few theoretical conditions for assuring its convergence in probability to the global minimum of numerical multivariable functions. The adopted approach was black box modeling, where experimental data and functional approximation methods take paramount roles, in contrast to white box modeling [1, 2, 10, 12, 14], where analytical results predominate.

References

1. T. Bose, *Digital Signal and Image Processing* (Wiley, New York, 2004)
2. C.K. Goh, Y.C. Lim, C.S. Ng, Improved weighted least squares algorithm for the design of quadrature mirror filters. IEEE Trans. Signal Process. **47**(7), 1866–1877 (1999)
3. O. Häggström, *Finite Markov Chains and Algorithmic Applications* (Cambridge University Press, Cambridge, 2002)
4. W.-P. Huang, L.-F. Zhou, J.-X. Qian, FIR filter design: frequency sampling filters by particle optimization algorithm, in *Third Int. Conf. on Machine Learning and Cybernetics*, Shanghai, Aug. 2004, pp. 26–29
5. L. Ingber, Adaptive simulated annealing (ASA): Lessons learned. Control Cybern. **25**(1), 33–54 (1996)
6. L. Ingber, Very fast simulated re-annealing. Math. Comput. Model. **12**(8), 967–973 (1989)
7. J.D. Johnston, A filter family designed for use in quadrature mirror filter banks, in *IEEE Int. Conf. Acoustics, Speech and Signal Process* (1980), pp. 291–294
8. Y.D. Jou, Design of real FIR filters with arbitrary magnitude and phase specifications using a neural-based approach. IEEE Trans. Circuits Syst. II, Express Briefs **53**(10), 1068–1072 (2006)
9. Y.D. Jou, Least-squares design of digital differentiators using neural networks with closed-form derivations. IEEE Signal Process, Lett. **12**(11), 760–763 (2005)
10. S.S. Kidambi, R.P. Ramachandran, Design of nonrecursive filters satisfying arbitrary magnitude and phase specifications using a least-squares approach. IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process. **42**(11), 711–716 (1995)
11. M.-I. Lai, S.-K. Jeng, Compact microstrip dual-band bandpass filters design using genetic-algorithm techniques. IEEE Trans. Microwave Theory Tech. **54**(1), 160–168 (2008)
12. T. Nguyen, The design of arbitrary FIR digital filters using the eigenfilter method. IEEE Trans. Signal Process. **41**(3), 1128–1139 (1993)
13. H.A. Oliveira Jr., Fuzzy control of stochastic global optimization algorithms and VFSR. Nav. Res. Mag. **16**, 103–113 (2003)
14. S.C. Pei, J.J. Shyu, Eigen-approach for designing FIR filters and all-pass phase equalizer with prescribed magnitude and phase response. IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process. **39**(3), 137–146 (1992)
15. J. Pintér, *Global Optimization in Action* (Kluwer Academic, Dordrecht, 1996)
16. T. Ross, *Fuzzy Logic with Engineering Applications* (McGraw-Hill, New York, 1995)
17. T. Schnier, X. Yao, P. Liu, Digital design using multiple Pareto fronts. Soft Comput. **8**, 332–343 (2004)