Design of Optimal L1 Stable IIR Digital Filter using Real Coded Genetic Algorithm

Ranjit Kaur, Member, IAENG, Manjeet Singh Patterh, J.S. Dhillon

Abstract— A real coded genetic algorithm is applied for designing infinite impulse response (IIR) filter based on L1-approximation error criterion. The proposed real coded genetic algorithm, which is a technique for optimization inspired by genetics and natural evolution method, enhances the search capability and provides a fast convergence for calculating the optimal filter coefficients. The filter designed based on L1-approximation error possesses flat passbands and stopbands to that of the least square design. A comparison has been made with other design techniques, demonstrating that the real coded genetic algorithm obtains better or at least comparable results for designing digital IIR filters than the existing genetic algorithm based methods.

Index Terms— Digital IIR filters, Real Coded Genetic Algorithm, L1-approximation error, Stability.

I. INTRODUCTION

D IGITAL filters are broadly classified into two groups: infinite impulse response (IIR) and finite impulse response (FIR). An IIR filter requires less computation as compared to FIR filter for the same performance. IIR digital filters are effectual in wide range of applications, particularly where high selectivity and efficient processing of discrete signals are desirable. A reliable design method based on a global search procedure is required for designing IIR filter to overcome the problem of multi-modal error surface of IIR filter. The design task of IIR digital filters is to approximate a given ideal frequency response by a stable IIR digital filter under some design criterion. To implement optimization technique with some criteria, various optimization methods have been applied where p-error, mean-square-error and ripple magnitudes (tolerances) of both passband and stopband are used to measure performance for the design of digital IIR filters [1-3]. Due to non-linear and multimodal nature of error surface of IIR filters, conventional gradient-based design methods may easily get stuck in the local minima of error surface [4-5]. Therefore, researchers have developed design methods based on modern heuristics optimization algorithms such as genetic algorithms [6-14].

Genetic algorithm (GA) and Evolutionary Programming (EP) are general purpose search algorithms which use evolutionary ideas of natural selection and genetic dynamics [15-17] to find the global optimal solution of the problem. The ability of genetic algorithms to blend both exploration and exploitation in an optimal way is one of their main assets [18]. In many multidimensional and multimodal engineering design problems, the GA has been used as a robust and proficient search technique. GA requires less iterative computational equations as compared to traditional search algorithms like calculus based searches, dynamic programming, random searches and gradient methods, but it takes more convergence time to reach the solution. Out of various existing coding schemes used for coding of search space solutions, real coding technique seems particularly natural when tackling optimization problems. GA which uses real coding technique is called real coded genetic algorithm (RCGA) [19, 20]. Whenever a parameter is binary coded, there is always the danger of the reduced level of precision as it does not represent parameter values that produce the best solution values. The RCGA improves the final local tuning capabilities of a binary coded genetic algorithm, which is a must for high precision optimization problems. Real-coding of the genes eliminates concern that precision is not adequate to represent good values in the search space.

The magnitude response of IIR filter is more important than phase response in real time applications. The intent of this paper is to apply a RCGA with arithmetic-average-bound-blend crossover and wavelet mutation operator for the design of stable digital IIR filter. The values of the filter coefficients are optimized with RCGA approach to achieve minimum L1-norm approximation error criterion. The paper is structured as follows: Section II describes the IIR filter design problem statement, RCGA for designing the optimal digital IIR filters is described in Section III, the performance of the proposed method has been evaluated and achieved results are compared with the design results given by Tsai and Chou [7] and Tang et al. [13] for the low-pass (LP), high-pass (HP), band-pass (BP) and band-stop (BS) filters in Section IV, finally, the conclusions and discussions are outlined in Section V.

Manuscript received September 17, 2012.

Ranjit Kaur is with Department of Electronics, University College of Engineering, Punjab University Patiala-147002, Punjab, India. Phone: +91-94630-16064, Fax: +91-175-2283125, (e-mail: ranjit24_ucoe@pu.ac.in).

Manjeet Singh Patterh is with Department of Electronics, University College of Engineering, Panjab University Patiala-147002, Punjab, India. (e-mail: mspattar@yahoo.com).

J.S. Dhillon is with the Department of Electrical & Instrumentation Engineering, Sant Longowal Institute of Engineering & Technology, Longowal-148106, Punjab, India. (e-mail: jsdhillon@sliet.ac.in).

(Advance online publication: 21 November 2012)
II. FILTER DESIGN PROBLEM

Digital filter design problem involves the determination of a set of filter coefficients which meet performance specifications such as pass-band width and corresponding gain, width of the stop-band and attenuation, band edge frequencies and tolerable peak ripple in the pass band and stop-band. The transfer function of IIR can be represented by cascading first and second order sections to avoid the coefficient quantization problem which causes instability. In cascade realization coefficient range is limited. The structure of cascading type digital IIR filter is:

\[ H(\omega, x) = \beta \left( \prod_{i=0}^{M} \frac{1 + p_i e^{-j\omega}}{1 + q_i e^{-j\omega}} \right) \times \left( \prod_{i=1}^{N} \frac{1 + r_i e^{-j\omega} + r_{2i} e^{-2j\omega}}{1 + s_i e^{-j\omega} + s_{2i} e^{-2j\omega}} \right) \]

(1)

where \( \beta \) is the gain, \( p_i \) and \( q_i \) are the first order coefficients, \( r_i \), \( r_{2i} \), \( s_i \) and \( s_{2i} \) are the second order coefficients. To generalize, vector \( x \) represents the filter coefficients of dimension \( V \times 1 \) with \( V = 2M + 4N + 1 \). \( x \) vector is represented as:

\[ x = [p_1; q_{11}; \ldots; p_M; q_{1M}; r_{11}; r_{12}; \ldots; s_{11}; s_{12}; \ldots; s_{2N}; s_{1N}]^T \]

The IIR filter is designed by optimizing the coefficients such that the approximation error function in \( L_1 \)-norm for magnitude is minimized. The frequency response is specified at \( K \) equally spaced discrete points in pass-band and stop-band. The \( L_p \)-norm approximation error for the magnitude response is defined as [19]:

\[ e(x) = \left( \sum_{i=0}^{K} |H_d(\omega_i) - H(\omega_i, x)|^p \right)^{1/p} \]

(2)

In the IIR filter design problem fixed grid approach is used [19]. For \( p=1 \), the magnitude response error denotes the \( L_1 \)-norm error and is defined as given below:

\[ e(x) = \sum_{i=0}^{K} |H_d(\omega_i) - H(\omega_i, x)| \]

(3)

Desired magnitude response, \( H_d(\omega_i) \) of IIR filter is given as:

\[ H_d(\omega_i) = \begin{cases} 1, & \text{for } \omega_i \text{ in passband} \\ 0, & \text{for } \omega_i \text{ in stopband} \end{cases} \]

(4)

The design of causal recursive filters requires the inclusion of stability constraints. Therefore, the stability constraints given by Eq. (5a) to Eq. (5e) which are obtained by using the Jury method [22] on the coefficients of the digital IIR filter in Eq. (1) are included in the optimization process.

Mathematically, IIR filter problem is formulated as below:

Minimize \( f(x) = e(x) \)

Subject to: Stability constraints

1. \( 1 + q_i \geq 0 \) (i = 1, 2, ..., M) \hspace{1cm} (5a)
2. \( 1 - q_i \geq 0 \) (i = 1, 2, ..., M) \hspace{1cm} (5b)
3. \( 1 - s_i \geq 0 \) (k = 1, 2, ..., N) \hspace{1cm} (5c)
4. \( 1 + s_i + s_{2k} \geq 0 \) (k = 1, 2, ..., N) \hspace{1cm} (5d)
5. \( 1 - s_k + s_{2k} \geq 0 \) (k = 1, 2, ..., N) \hspace{1cm} (5e)

RCGA is applied and constraints are forced to satisfy by randomly updating them.

III. REAL CODED GENETIC ALGORITHM

RCGA searches for many points in the search space at once, and continually narrows the focus of the search to the areas of the observed best performance. RCGA possesses a lot of advantages than its binary coded counterpart when dealing with continuous search spaces with large dimensions and where great numerical precision is required. Using real values, the representation of the solutions is very close to the natural formulation of problem and it avoids the coding and decoding processes, thus increasing the GA’s speed, efficiency and precision. The basic elements of RCGA are reproduction, selection, crossover and mutation. In reproduction operation, the individuals possessing higher fitness values are selected from the existing population. In the crossover operation, two individuals are selected at random from the mating pool and a crossover is performed using mathematical relations. Mutation is an important part of genetic search, it helps to prevent the population from stagnating at any local optima. Mutation is intended to prevent the search falling into local optimum of the search space.

In this paper, a RCGA with genetic operators including arithmetic-average-bound-blend (AABBX) crossover and wavelet mutation is applied for optimizing the coefficients of digital IIR filter in order to minimize the magnitude approximation error in \( L_1 \)-norm by employing stability constraints. The arithmetic-average-bound-blend crossover operator combines the arithmetic, average, bound and blend crossover operators. The arithmetic crossover operation produces some children with their parent’s features; average crossover manipulates the genes of the selected parents and the minimum and maximum possible values of the genes and bound crossover is capable of moving the offspring near the domain boundary. The offspring thus obtained spreads over the domain so that a higher chance of reaching the global optimum can be obtained. The wavelet mutation operation based on wavelet theory [23] is a powerful tool for fine tuning of the genes to search the solution space locally. This property of wavelet mutation operation enhances the searching performance and provides a faster convergence than conventional RCGA.

Algorithm for Real Coded Genetic Algorithm

1. Generate initial population strings randomly.
2. Calculate fitness values of population members.
3. Search for solution among the population? If ‘yes’ then GOTO Step 8.
4. Using stochastic remainder roulette wheel selection choose highly fit member of population as parents and generate off-springs according to their fitness.
operator to introduce variations and generate offsprings.
6. Substitute existing offsprings with new offsprings by applying competition and selection.
7. GOTO Step 3 and repeat.
8. Stop.

A. Initialization
Random search is applied to record the starting point. Global search is applied to explore the starting point and then the starting point is perturbed in local search space to record the best starting point. The search process is started by initializing the variable \( x_i^0 \) using Eq. (6) which is used to calculate objective function Eq. (5).

\[
x_i^0 = x_{i_{\text{min}}} + \text{rand}(x_{i_{\text{max}}}-x_{i_{\text{min}}}) (i=1,2,...,V; j=1,2,...,NV)
\]  

(6)

where \( \text{rand} \) is a uniform random generated number having value between 0 and 1, \( i \) is number of generation, \( V \) is number of variables and \( NV \) is the population size.

B. Fitness Function
Expected fitness function, \( f \) is derived from the objective function and is used in successive genetic operations. The expected fitness function used to solve design of IIR filter is obtained from Eq. (3) and is stated as:

\[
f_j^r = \max \left\{ \frac{1}{1+e^r(x)} \right\}
\]

(7)

where \( e^r(x) \) is obtained from Eq. (3) and is stated as:

\[
e^r(x) = \sum_{i=1}^{k} \left[ H_d(\omega_i) - |H'_d(\omega_i,x)| \right]
\]

C. Constraint Handling
The stability constraints given by Eq. (5a) to Eq. (5e) have been forced to satisfy by updating the coefficients with random variation as given below. The variation is given as

\[
q_{i\ell} = \begin{cases}
(1-r)^2 ; & \text{if } (1+q_{i\ell}) < 0 \text{ or } (1-q_{i\ell}) < 0 \\
q_{i\ell} ; & \text{else}
\end{cases}
\]

(7a)

\[
s_{2k} = \begin{cases}
(1-r)^2 ; & \text{if } (1-s_{2k}) < 0 \text{ or } (1-s_{2k}) \geq 0 \\
s_{2k} ; & \text{else}
\end{cases}
\]

(7b)

\[
s_{1k} = \begin{cases}
(1-r)^2 ; & \text{if } (1+s_{1k}+s_{2k}) < 0 \text{ or } (1+s_{1k}+s_{2k}) \geq 0 \\
s_{1k} ; & \text{else}
\end{cases}
\]

(7c)

where \( r \) is any uniform random number which is varied between [0,1]. Square term gives small increment.

D. Reproduction
The initial and most important genetic algorithm operator is reproduction. In reproduction good members from population are selected to form a mating pool. The reproduction operator is also known as selection operator. Many reproduction operators exist and they all essentially pick the strings of above average from the current population and insert their multiple copies in the mating pool in a probabilistic manner. The commonly used reproduction operator is the proportionate reproduction operator where a string is selected for the mating pool with a probability proportional to its fitness. The basic roulette wheel selection method is stochastic sampling with replacement (SSR). The segment size and selection probability remain the same throughout the selection phase and individuals are selected accordingly. Stochastic sampling with partial replacement (SSPR) extends upon SSR by resizing an individual's segment if it is selected. After the selection of each individual, the size of its segment is reduced by one. If the segment size becomes negative, then it is set to zero. Remainder sampling methods involve two distinct phases. In the integral phase, the individuals are selected deterministically according to the integer part of their expected trials. The remaining individuals are then selected probabilistically from fractional part of the individual’s expected values. In this paper the stochastic remainder roulette wheel selection has been applied [24].

E. Crossover Operators
The arithmetic-average-bound-blend crossover operator has been used for the selection of chromosomes and is based on the stochastic remainder Roulette-wheel mechanism [24]. The AABBX operator is the combination of arithmetic crossover, average crossover, bound and blend crossovers. Suppose, two vectors are selected chromosomes in the \( i^{th} \) iteration of the RC\text{G}A execution. Each chromosome has \( V \) genes, which are real numbers. The AABBX operator creates ten children from the parents \( x_i \) and \( x_{i'} \) as follows:

(i) Arithmetic crossover

\[
x_{i1} = w_x x_{i'} + (1-w_x) x_i \quad (i = 1,2,...,V)
\]

(8)

\[
x_{i2} = (1-w_x) x_{i'} + w_x x_i \quad (i = 1,2,...,V)
\]

(9)

\[
x_{i3} = \text{Min} \{x_{i'}, x_i\} \quad (i = 1,2,...,V)
\]

(10)

\[
x_{i4} = \text{Max} \{x_{i'}, x_i\} \quad (i = 1,2,...,V)
\]

(11)

(ii) Average crossover

\[
x_{i5} = \frac{1}{2} (x_{i'} + x_i) \quad (i = 1,2,...,V)
\]

(12)

\[
x_{i6} = \frac{1}{2} (w_{x} (x_{i'} + x_i) + (1-w_{x}) (x_{i'} + x_i)) (i = 1,2,...,V)
\]

(13)

(iii) Bound crossover

\[
x_{i7} = w_x \text{Min} \{x_{i'}, x_i\} + (1-w_x) x_i \quad (i = 1,2,...,V)
\]

(14)

\[
x_{i8} = w_x \text{Min} \{x_{i'}, x_i\} + (1-w_x) x_{i'} \quad (i = 1,2,...,V)
\]

(15)

(iv) Blend crossover

\[
x_{i9} = w_x x_i + (1-w_x) x_{i'} \quad (i = 1,2,...,V)
\]

(16)
\[ x_{i,j}^{\text{up}} = (1 - w_a) x_{i,j}^w + w_a x_{i,j}^w \quad (i = 1, 2, ..., V) \] (17)

where \( x_{i,j}^w \) and \( x_{i,j}^u \) are constant weights. The values are adjusted such that \( 0 < w_a, w_b, w_c < 1 \). \( w_a \) is also constant weight such that \( 1 < w_a < 2 \). Two children having the highest fitness values are selected as the offspring chromosomes for the crossover operation. These two offspring chromosomes are added to the previous population including the parents. The enlarged population formed after the execution of the crossover operator is considered for the mutation.

**F. Mutation Operator**

Mutation is a genetic operator used to maintain genetic diversity from one generation of population of chromosomes to next. Mutation alters one or more gene values in a chromosome from its initial state. This can result in entirely new gene values being added to the gene pool. With the new gene values, the genetic algorithm may be able to arrive at better solution than was previously available. Each gene of the chromosome is given an opportunity to mutate, governed by the probability of the mutation \( pm \). For each gene of the chromosome, a random number in the range of \([0, 1]\) is generated. If the random number is less than \( pm \) that gene is selected for the mutation, otherwise it is not selected. In this algorithm, \( pm \) is set at 0.2. The new gene, \( x_{ij} \) after mutation will be as follows:

\[
x_{ij} = \begin{cases} 
  x_{ij}^1 + \Delta(\phi, t, \text{MAXIT}) \left( x_{ij}^{\text{max}} - x_{ij}^1 \right) & \text{if } \Delta \geq pm \\
  x_{ij}^1 + \Delta(\phi, t, \text{MAXIT}) \left( x_{ij}^{\text{max}} - x_{ij}^1 \right) & \text{if } \Delta < pm 
\end{cases}
\] (18)

where \((i = 1, 2, ..., V; j = 1, 2, ..., NV)\) \( \text{MAXIT} \) is the maximum number of iterations of the RCGA and \( t \) is the current iteration number.

Morlet wavelet as the mother wavelet shown in Fig. 1 can be rewritten as

\[
\Delta(\phi, t, \text{MAXIT}) = \psi_d(\phi) = \frac{1}{\sqrt{d}} e^{-\frac{\phi^2}{2}} \cos \phi 
\] (19)

where \( \phi \) is randomly generated in the range of \([-4, 4]\) because this wavelet has \([-4, 4]\) as its effective support.

The dilation parameter \( d \) is set to vary with the value of \((t/\text{MAXIT})\), giving the adaptive search capability to the proposed real coded genetic algorithm.

\[
d = \exp \left( \ln g \left( 1 - \left( \frac{t}{\text{MAXIT}} \right)^{\xi} \right) \right) 
\] (20)

where \( \xi \) is the shape parameter of the monotonic increasing function of \( d \). \( g \) is the upper limit of the dilation parameter. The dilation parameter \( d \) is a function of \( t \) and \( \text{MAXIT} \), and so \( \Delta \) is really a function of \( \phi \), \( t \) and \( \text{MAXIT} \).

**G. Competition and Selection**

Each individual in the combined population has to compete with some other individuals to have a chance to be copied to the next iteration. The score for each trial vector after stochastic competition is given by

\[
w_i = \sum_{m=1}^{NV} w_{i,m} 
\] (21)

where

\[
w_{i,m} = \begin{cases} 
  1 & \text{if } u_i < \frac{f_i^m}{f_i + f_m^m} \\
  0 & \text{otherwise} 
\end{cases} 
\] (22)

NV is the population size or the number of competitors. \( f_i^m \) is the fitness value of the randomly selected competitor from the combined population. \( f_i^m \) is the fitness value of \( x_i^m \). \( u_i \) and \( u_m \) are randomly selected numbers from a uniform distribution set \( u (0, 1) \) and \( m = \text{int} (2 \times NV \times u_i + 1) \). After competing, the trial 2NV solutions, including the parents and the offspring, are ranked in descending order of the score obtained in Eq. (21). The first NV trial solutions survive and are copied along with their objective functions into survivor set as the individuals of the next iteration.

**IV. DESIGN EXAMPLES AND COMPARISONS**

Low-pass, high-pass, band-pass and band-stop digital IIR filters have been designed by taking 200 equally spaced points within the frequency domain \([0, \pi]\). For the purpose of comparison, the lowest order of digital IIR filters is set exactly the same as that given by Tang et al. [13]. The intent of designing the digital IIR filters is to minimize the objective function given by Eq. (5) subject to stability constraints stated by Eq. (5a) to Eq. (5c). The prescribed design conditions for IIR filters are given in Table I.

It is known fact that the values of control parameters such as population size, crossover and mutation rate have significant effect on the performance of RCGA. The control parameter values employed for the RCGA algorithm are given in Table II.

The \( L_1 \)-norm approximation error in terms of magnitude response, pass-band ripples and stop-band ripples obtained with proposed RCGA approach for the LP, HP, BP and BS filters are presented in Tables III, Tables IV, Tables V and Tables VI, respectively. The magnitude response diagrams of LP, HP, BP and BS filters designed with RCGA approach are presented in figure 2, figure 3, figure 4 and figure 5 respectively. It can be observed from the aforementioned results that the IIR filters designed by the RCGA approach have better performances than those designed by the genetic algorithm based methods given by Tsai and Chou [7] and Tang et al. [13].

(Advance online publication: 21 November 2012)
The pole-zero diagrams for LP, HP, BP and BS filters are presented in Figure 6. The maximum pole radius values of designed digital IIR filters are given in Table VII. It can be observed that the designed filters follow the stability constraints imposed in the design procedure as all the poles lie inside the unit circle. The position of poles away from the origin of the designed filters are given by (0.8655, ± 0.6293), (0.6655, 0) for LP, (0.8547, ± 2.4829), (0.6197, 3.1416) for HP, (0.8751, ± 1.2018), (0.8780, ± 1.9459), (0.7333, ± 1.5778) for BP, and (0.7352, ± 0.9760), (0.7312, ± 2.1594) for BS. The first number in parentheses is the radius, and the second number is the angle in radians. The designed IIR filter models obtained by the RCGA approach for LP, HP, BP and BS are given by Eq. (24), Eq. (25), Eq. (26) and Eq. (27), respectively.

\[ H_{LP}(z) = 0.041885 \times \frac{(z + 1.001136)(z^2 - 0.600598z + 0.996026)(z - 0.665484)(z^2 - 1.399500z + 0.749144)}{} \]  
\[ H_{HP}(z) = 0.055776 \times \frac{(z - 0.998962)(z^2 + 0.691867z + 0.983476)(z + 0.619709)(z^2 + 1.351760z + 0.730540)}{} \]  
\[ H_{BP}(z) = 0.022712 \times \frac{(z^2 - 0.331734z - 1.505514)(z^2 + 0.238150z - 0.749302)(z^2 - 0.631279z + 0.765823)(z^2 + 0.010281z + 0.537769)}{} \times \frac{(z^2 - 0.091838z - 1.107565)(z^2 + 0.643415z + 0.770893)}{} \]  
\[ H_{BS} = 0.426811 \times \frac{(z^2 + 0.430594z + 1.027871)(z^2 - 0.441609z + 0.996931)(z^2 - 0.823978z + 0.540532)(z^2 + 0.811903z + 0.534672)}{} \]

### Table I
**Prescribed Design Conditions on LP, HP, BP and BS Filters.**

| Filter type     | Pass-band (\(\omega\)) | Stop-band (\(\omega\)) | Maximum Value of \(|H(\omega, x)|\) |
|-----------------|-------------------------|-------------------------|-----------------------------------|
| Low-Pass        | \(0 \leq \omega \leq 0.2\pi\) | \(0.3\pi \leq \omega \leq \pi\) | 1                                 |
| High-Pass       | \(0.8\pi \leq \omega \leq \pi\) | \(0 \leq \omega \leq 0.7\pi\) | 1                                 |
| Band-Pass       | \(0.4\pi \leq \omega \leq 0.6\pi\) | \(0 \leq \omega \leq 0.25\pi\) | \(0.75 \leq \omega \leq \pi\) | 1                                 |
| Band-Stop       | \(0 \leq \omega \leq 0.25\pi\) | \(0.4\pi \leq \omega \leq 0.6\pi\) | \(0.75 \leq \omega \leq \pi\) | 1                                 |

### Table II
**Value of Control Parameters**

<table>
<thead>
<tr>
<th>Population Size</th>
<th>40</th>
</tr>
</thead>
<tbody>
<tr>
<td>Representation</td>
<td>Real number representation</td>
</tr>
<tr>
<td>Crossover</td>
<td>Arithmetic-average-bound-blend crossover</td>
</tr>
<tr>
<td>Crossover Rate</td>
<td>0.9</td>
</tr>
<tr>
<td>Mutation</td>
<td>Wavelet Mutation</td>
</tr>
<tr>
<td>Mutation Rate</td>
<td>0.01</td>
</tr>
</tbody>
</table>

(Advance online publication: 21 November 2012)
### TABLE III
**DESIGN RESULTS FOR LOW PASS FILTER**

<table>
<thead>
<tr>
<th>Method</th>
<th>$L_1$-norm error</th>
<th>Pass-band performance (Ripple magnitude)</th>
<th>Stop-band performance (Ripple magnitude)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$0.9220 \leq</td>
<td>H(e^{j\omega})</td>
</tr>
<tr>
<td>RCGA Approach</td>
<td>3.7578</td>
<td>(0.0975)</td>
<td>(0.1522)</td>
</tr>
<tr>
<td>TIA Approach[7]</td>
<td>3.8157</td>
<td>$0.8914 \leq</td>
<td>H(e^{j\omega})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0.1086)</td>
<td>(0.1638)</td>
</tr>
<tr>
<td>Method of Tang et al. [13]</td>
<td>4.3395</td>
<td>$0.8870 \leq</td>
<td>H(e^{j\omega})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0.1139)</td>
<td>(0.1802)</td>
</tr>
</tbody>
</table>

### TABLE IV
**DESIGN RESULTS FOR HIGH PASS FILTER**

<table>
<thead>
<tr>
<th>Method</th>
<th>$L_1$-norm error</th>
<th>Pass-band performance (Ripple magnitude)</th>
<th>Stop-band performance (Ripple magnitude)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$0.9608 \leq</td>
<td>H(e^{j\omega})</td>
</tr>
<tr>
<td>RCGA Approach</td>
<td>4.1519</td>
<td>(0.0633)</td>
<td>(0.1421)</td>
</tr>
<tr>
<td>TIA Approach[7]</td>
<td>4.1819</td>
<td>$0.9229 \leq</td>
<td>H(e^{j\omega})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0.0771)</td>
<td>(0.1424)</td>
</tr>
<tr>
<td>Method of Tang et al. [13]</td>
<td>14.5078</td>
<td>$0.9224 \leq</td>
<td>H(e^{j\omega})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0.0779)</td>
<td>(0.1819)</td>
</tr>
</tbody>
</table>

### TABLE V
**DESIGN RESULTS FOR BAND-PASS FILTER**

<table>
<thead>
<tr>
<th>Method</th>
<th>$L_1$-norm error</th>
<th>Pass-band performance (Ripple magnitude)</th>
<th>Stop-band performance (Ripple magnitude)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$0.9924 \leq</td>
<td>H(e^{j\omega})</td>
</tr>
<tr>
<td>RCGA Approach</td>
<td>1.4188</td>
<td>(0.0170)</td>
<td>(0.0606)</td>
</tr>
<tr>
<td>TIA Approach[7]</td>
<td>1.5204</td>
<td>$0.9681 \leq</td>
<td>H(e^{j\omega})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0.0319)</td>
<td>(0.0679)</td>
</tr>
<tr>
<td>Method of Tang et al. [13]</td>
<td>5.2165</td>
<td>$0.8956 \leq</td>
<td>H(e^{j\omega})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0.1044)</td>
<td>(0.1772)</td>
</tr>
</tbody>
</table>

### TABLE VI
**DESIGN RESULTS FOR BAND-STOP FILTER**

<table>
<thead>
<tr>
<th>Method</th>
<th>$L_1$-norm error</th>
<th>Pass-band performance (Ripple magnitude)</th>
<th>Stop-band performance (Ripple magnitude)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$0.9385 \leq</td>
<td>H(e^{j\omega})</td>
</tr>
<tr>
<td>RCGA Approach</td>
<td>3.0306</td>
<td>(0.0735)</td>
<td>(0.1319)</td>
</tr>
<tr>
<td>TIA Approach[7]</td>
<td>3.4750</td>
<td>$0.9259 \leq</td>
<td>H(e^{j\omega})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0.0741)</td>
<td>(0.1178)</td>
</tr>
<tr>
<td>Method of Tang et al. [13]</td>
<td>6.6072</td>
<td>$0.8920 \leq</td>
<td>H(e^{j\omega})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0.1080)</td>
<td>(0.1726)</td>
</tr>
</tbody>
</table>
Table VII

Maximum Radius of Poles

<table>
<thead>
<tr>
<th>Filter Type</th>
<th>Maximum Radius</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low-Pass</td>
<td>0.8655</td>
</tr>
<tr>
<td>High-Pass</td>
<td>0.8547</td>
</tr>
<tr>
<td>Band-Pass</td>
<td>0.8780</td>
</tr>
<tr>
<td>Band-Stop</td>
<td>0.7352</td>
</tr>
</tbody>
</table>

Fig. 2 Frequency responses of low pass filter using the RCGA approach and the method given in [7] and [13], respectively.

Fig. 3 Frequency responses of high pass filter using the RCGA approach and the method given in [7] and [13], respectively.
Fig. 4 Frequency responses of band pass filter using the RCGA approach and the method given in [7] and [13], respectively.

Fig. 5 Frequency responses of band stop filter using the RCGA approach and the method given in [7] and [13] respectively.
This paper proposes a RCGA approach for the design of digital IIR filters based on \(L_1\)-norm approximation error. As shown through experimental results, RCGA approach works well with an arbitrary random initialization and the designed digital IIR filters satisfy the prescribed amplitude specifications consistently. Therefore, the proposed algorithm is a useful tool for the design of IIR filters with complicated stability constraints. The design results for LP, HP, BP and BS digital IIR filters clearly depict that the proposed RCGA approach possesses the capacity for the local tuning of the solutions and is an efficient optimizer with fast convergence rate and robustness. It can be further implemented to optimize the structure and phase response of the digital IIR filters.

**REFERENCES**


(Advance online publication: 21 November 2012)