Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2012, Article ID 560650, 15 pages
doi:10.1155/2012/560650

Research Article

CSD-Coded Genetic Algorithm on Robustly Stable Multiplierless IIR Filter Design

Shing-Tai Pan

Department of Computer Science and Information Engineering, National University of Kaohsiung, No. 700, Kaohsiung University Road, Nanzih District, Kaohsiung 811, Taiwan

Correspondence should be addressed to Shing-Tai Pan, stpan@nuk.edu.tw

Received 6 October 2011; Revised 5 December 2011; Accepted 5 December 2011

Academic Editor: Jyh Horng Chou

A Canonic-Signed-Digit-(CSD-) coded genetic algorithm (GA) is proposed to find the optimal design of robustly stable infinite impulse response digital filter (IIR). Under the characteristics of the CSD structure, the circuit of the filter can be simplified and also the calculation speed can be raised to increase the hardware's efficiency. However, the design of CSD has a big challenge: the CSD structure of the system parameters will be destroyed by an optimal design procedure. To solve this problem, in this research a CSD-coded GA is proposed so that the CSD structure can be maintained. Moreover, the robustly stable IIR filters design problem is included in this paper. The robustness of the IIR filters is achieved by ensuring that all poles of the filters are located inside a disk $D(\alpha, r)$ contained in the unit circle, in which α is the center, r is the radius of the disk, and $|\alpha| + r < 1$. Consequently, in this paper, a new and more efficient $D(\alpha, r)$-stability criterion will be derived and then embedded in GA for the design of robust IIR filters. It is worthwhile to note that to design an IIR filter simultaneously with CSD-structured parameters and robust stability is difficult and is not well explored so far. An example will be presented to show the efficiency of the proposed strategy for design of IIR filters.

1. Introduction

Among the digital filters, Infinite Impulse Response (IIR) digital filters are useful in various applications where high selectivity processing of discrete signals is desirable [1]. Consequently, IIR filters designs have been a hot topic in the research of digital signal processing [1–16]. Of all the design problems of IIR filters, stability is the major problem, which arises in the design of IIR filters. Recently, there have been many literatures concerning this topic. For example, in [8], stability is guaranteed by a linear inequality constraint. In that paper, the problem of minimizing the output error function with the stability constraint is solved by Quadratic Programming (QP). Similarly, in the papers [1, 12], the minimax design method is used to design stable IIR filters under the condition that the magnitude and phase response is prescribed. Moreover, adaptive IIR filtering is the challenging problem in the design of intrinsically stable filters [6]. Some convenient solutions have been developed by using gradient-based techniques as shown in the examples of [6, 10, 11].

As for the hardware implementation of IIR filters, in recent years, due to its characteristics the CSD code has been broadly applied to the simplification of hardware structure. The CSD code method can not only reduce the number of adders/subtractors and shift registers, but also accelerate the calculation process and achieve the performance of hardware circuit simplification. Therefore, in recent years, a lot of research has applied the CSD code structure to circuit design [17–21]. Recently, since the computation ability of computers has been enormously enhanced, evolutionary algorithm computation is widely applied in many areas of engineering; for a recent review of these applications, please see, for example, [14, 22–25]. The most popular evolutionary algorithm is GA. GA is based on the Darwinian “Survival of the Fittest” strategy. Each individual in the population represents a potential solution to the problem at hand. They compete and mate with each other in order to produce stronger individuals. Much recent research has pointed out that GA is efficient in solving optimization problems [2, 13–15, 26–30]. GA is adopted in this paper for the design of CSD-coded IIR filters due do to the advantage that each gene in the chromosome of GA can be taken care of individually and can be designed to keep the CSD structure. Other evolutionary algorithms, such as particle swarm optimization (PSO), differential evolution (DE), and steepest decent method (SDM), use real numbers to code the problem’s variables; therefore it is not possible for them to keep the special CSD coding.

So far, the research on CSD-coded filters has focused on the design of FIR filters. Examples can be found in [31–34]. In those researches, the GA based on the CSD code structure was studied for the design of FIR filters. There were two types of design methods. The first method was to check the inspection of the CSD structure after each generation of GA evolution. The second method was to transform all filter coefficients to the binary system code first and then transform the designed binary system code into a CSD structure code. However the CSD structure code can be only an approximation of the binary system code. Hence, errors unavoidably occur during the transformation.

In this paper, GA has been used to search for the optimum filter coefficient with the CSD code. Since the CSD code structure may be destructed during GA evolution process, this paper proposes a strategy to solve this problem. Moreover, the design of a robustly stable IIR filters is included in this CSD coded filter design. An algorithm is proposed in this paper to simultaneously solve the problem of the CSD code design and the robustly stable IIR filters design. In this paper, the robustness of the IIR filters is achieved by ensuring that all poles of the filters are located inside a disk $D(\alpha, r)$ contained in the unit circle, in which α is the center, r is the radius of the disk, and $|\alpha| + r < 1$. So, a $D(\alpha, r)$-stability criterion will be first derived and is then used to check the stability of the design filter in each generation of the evolution of GA. It is worthwhile to note that to design an IIR filter simultaneously with CSD-structured parameters and robust stability is difficult and is not well explored so far.

The structure of this paper is as follows. Section 2 introduces the IIR filters. Section 3 is on the CSD-coded evolution strategy of GA. In Section 4, a $D(\alpha, r)$-stability criterion will be derived and then be applied to the CSD-coded structured robustly stable IIR filters design according to the proposed strategy. Section 5 shows the simulation results with a low-pass IIR filter design example.

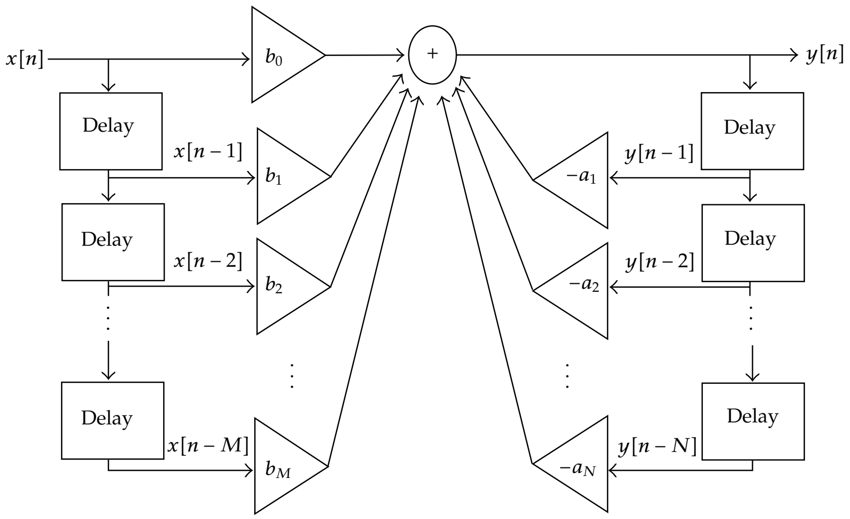


Figure 1: Architecture of IIR filters [14].

2. Infinite Impulse Response Filters (IIRs)

An IIR filter is a filter with the property of output feedback. An architecture which is used generally is depicted in Figure 1 [14]. The governing equation of an IIR filter is then described as follows:

$$\sum_{k=0}^{N} a_k y[n-k] = \sum_{k=0}^{M} b_k x[n-k], \tag{2.1}$$

in which a_k, $k = 0, 1, \ldots, N$, and b_k, $k = 0, 1, \ldots, M$, are the coefficients of the IIR filter. It is obvious that the output of the filter depends on not only the current and past inputs but also on the past outputs. This leads to the dependence of the output on the infinite many past inputs due to the iteration process in (2.1). Hence, the filter is called Infinite Impulse Response (IIR) filter. According to (2.1), the transfer function of the IIR filter can be derived as

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3} + \cdots + b_M z^{-M}}{a_0 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3} + \cdots + a_N z^{-N}} = \frac{\sum_{k=0}^{M} b_k z^{-k}}{\sum_{k=0}^{N} a_k z^{-k}}. \tag{2.2}$$

Before further discussions, the following definitions will be helpful to describe the proposed design strategy.

Definition 2.1. A polynomial $d(z)$ is P-stable if all solutions of the equation $d(z) = 0$ lie inside the unit circle.

Definition 2.2. A polynomial $d(z)$ is $PD(\alpha, r)$-stable if all solutions of the equation $d(z) = 0$ are within the disk $D(\alpha, r)$ centered at α with radius r, in which $r > 0$ and $|\alpha| + r < 1$.

Definition 2.3. Let the transfer function of a IIR filter be described as $H(z) = \sum_{k=0}^{M} b_k z^{-k} / \sum_{k=0}^{N} a_k z^{-k}$. If the denominator of the $H(z)$, $a(z) = \sum_{k=0}^{N} a_k z^{-k}$, is P-stable, then the IIR filter is stable. Moreover, if $a(z)$ is $PD(\alpha, r)$-stable, then the IIR filter is $D(\alpha, r)$-stable.

However, without loss of generality, we adopt the function

$$H(z) = \frac{\sum_{k=0}^{m} b_k z^{-k}}{1 + \sum_{k=1}^{n} a_k z^{-k}} \tag{2.3}$$

as the transfer function of designed IIR filters [14]. In the following paragraphs, the parameters a_k and b_k will be found, under the CSD-coded format, via the proposed strategy so that the error between the frequency response of designed IIR filters and the desired frequency response is minimal. Moreover, all the poles of $H(z)$ will be placed in the disk $D(\alpha, r)$, that is, the IIR filters are $D(\alpha, r)$-stable.

3. Genetic Algorithm with CSD-Coded Evolution Strategy

The main feature of the CSD format is that two consecutive digits cannot be "1" simultaneously. That is, the product of every two consecutive digits must be zero, that is, $b_n \times b_{n+1} = 0$. With this feature, fewer bits of nonzero values will appear in the parameters of a system, and hence fewer shifting and adding operations are required for the computation of system output. Therefore, compared with the binary system, the CSD coding of system parameters can accelerate the operation speed [32]. The CSD-coded coefficient C is represented as follows. It is noted that a ternary coding for the CSD code is adopted here to shorten the bit length of the encoded number

$$C = \sum_{j=0}^{m} S_{k_j} \times 2^{j}, \tag{3.1}$$

where $S_{k_j} \in \{-1, 0, 1\}$ and m is the bit length of the encoded number C.

Since the CSD format is possibly destructed during the evolution of GA, in past research, the examination after each evolution for making sure whether the CSD structure remains or not is necessary. A new evolution process will be fired if the CSD format is destructed by the previous evolution. However, this method lengthens the search time for an optimum solution. Therefore, in this section, in order to decrease the time of examining the CSD structure, a new crossover and a new mutation are proposed so that the CSD code structure will remain during the evolution process of GA.

3.1. Definition of the Fitness Function

In order to identify the quality of a chromosome in GA, the fitness function is usually used to evaluate each chromosome. Different fitness functions are used for different environments. It is important to define the fitness function of a chromosome. In this paper, the definition of the fitness function for the IIR design is defined as follows.

Suppose the transfer function of IIR filters is expressed as (2.3). Let $z = e^{j\Omega}$, then the frequency response of IIR filters becomes

$$H(\Omega) = \frac{\sum_{k=0}^{m} b_k e^{-jk\Omega}}{1 + \sum_{k=1}^{n} a_k e^{-jk\Omega}} \tag{3.2}$$

or

$$H(\Omega) = \frac{Y(\Omega)}{X(\Omega)} = \frac{b_0 + b_1 e^{-j\Omega} + b_2 e^{-j2\Omega} + \cdots + b_m e^{-jm\Omega}}{1 + a_1 e^{-j\Omega} + a_2 e^{-i2\Omega} + \cdots + a_n e^{-jn\Omega}}, \tag{3.3}$$

in which $\Omega = 2\pi(f/f_s)$; Ω is the digital frequency; f is the analog frequency; f_s is the sampling frequency. The error function is then defined as

$$E_p \equiv \frac{\sum_{\Omega} |H(\Omega) - H_I(\Omega)|^2}{N_s}, \quad 0 \le \Omega \le \pi, \tag{3.4}$$

in which $H_I(\Omega)$ is the desired frequency response and N_s is the sampling point. The fitness function for a chromosome p is then defined as

$$\text{fitness}(p) = \frac{1}{E_p^2 + 1}. \tag{3.5}$$

3.2. Crossover Based on CSD Structure

For crossover of the chromosomes with CSD code format, it matters whether the offspring remains its CSD structure after the crossover. Usually the crossover results in a non-CSD structure and a renewal of the code is required. In past research, the structure of the parameters must be examined after crossover. If a non-CSD structure was found, the code was renewed. The renewed code must be close to the value of the original code. If the length of a code was long, it takes much time to examine the code format. Consequently, it cost enormous computation time for the methods in the research. Therefore, this paper proposes a new method to save the time for renewing a code. However, the proposed method must be operated only when the parent chromosomes are already in the CSD format, and then the offspring will be permanently maintained in the CSD format. The Proposed CSD-based crossover can be described as follows:

(1) if $C_{N_1}^{P_1} \times C_{N_1-1}^{P_2} \neq 0$ then

$$C_{N_1-1}^{P_2} = 0 \tag{3.6}$$

end,

(2) if $C_{N_2}^{P_1} \times C_{N_2+1}^{P_2} \neq 0$ then

$$C_{N_2}^{P_1} = 0 \tag{3.7}$$

end,

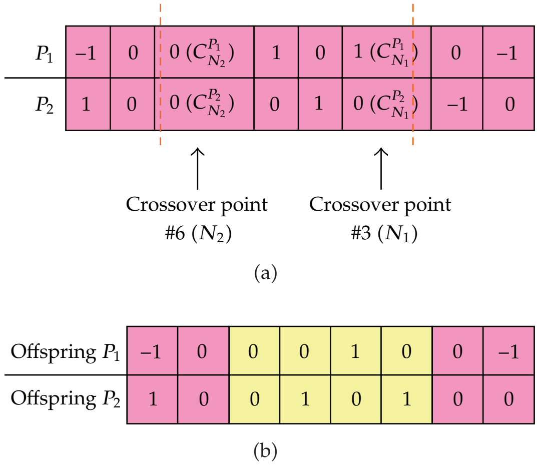

P_1 | −1 | 0 | 0 ($C^{P_1}_{N_2}$) | 1 | 0 | 1 ($C^{P_1}_{N_1}$) | 0 | −1
P_2 | 1 | 0 | 0 ($C^{P_2}_{N_2}$) | 0 | 1 | 0 ($C^{P_2}_{N_1}$) | −1 | 0
Crossover point #6 (N_2)
Crossover point #3 (N_1)
(a)
Offspring P_1 | −1 | 0 | 0 | 0 | 1 | 0 | 0 | −1
Offspring P_2 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0
(b)

Figure 2: (a) Before two-point crossover based on CSD structure. (b) After two-point crossover based on CSD structure.

(3) if $C^{P_2}_{N_1} \times C^{P_1}_{N_1-1} \neq 0$ then

$$C^{P_1}_{N_1-1} = 0 \tag{3.8}$$

end,

(4) if $C^{P_2}_{N_2} \times C^{P_1}_{N_2+1} \neq 0$ then

$$C^{P_2}_{N_2} = 0 \tag{3.9}$$

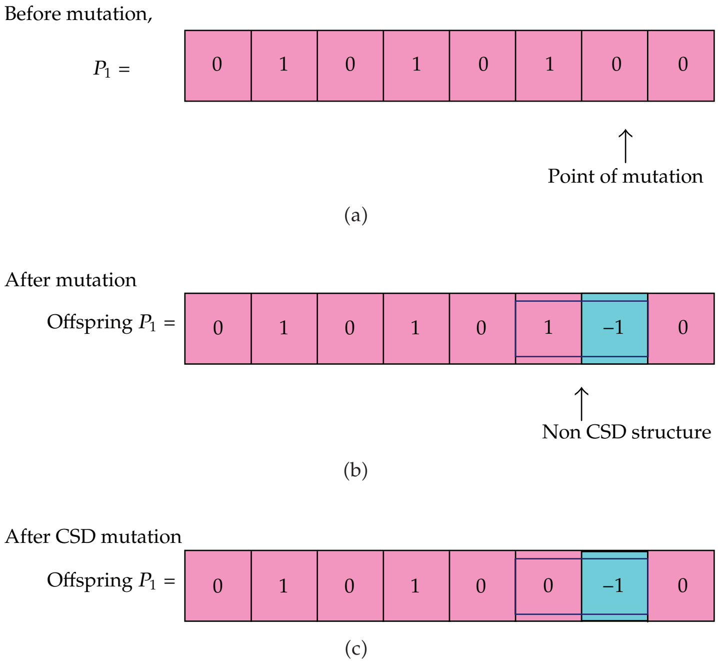
end,

(5) proceed the crossover process of the two points P_1 and P_2 with N_1 and N_2.

The notations $C^{P_i}_{N_i}$, $i = 1, 2$, represent the N_i^{th} bit of the P_i^{th} chromosome, and "×" represents the mathematical multiplication. Figure 2 gives an example for this CSD-based crossover. In the figure, we suppose the two points selected for crossover are points 3 and 6 of a chromosome. Figure 2(b) shows that the proposed crossover preserves the CSD structure in the offspring.

3.3. Mutation Based on CSD Structure

A mutation for a binary-coded chromosome always simply changes 0 to 1 or vice versa. However, for a CSD structure, the mutation will cause loss of the CSD format. Thus a method must be found to make the code remain in the CSD format after mutation. Otherwise, the effort to maintain the CSD structure in the previous section by the CSD-based crossover method would become futile. Therefore, in this paper, we propose a method for the mutation so that the CSD format will be maintained after mutation.



(a)

(b)

(c)

Figure 3: Illustration for CSD-based mutation.

In the CSD structure only the values 1 and −1 would influence the CSD structure; the CSD structure will be affected when a mutation transforms from 0 to 1 or −1, whereas the structure is not affected with the transform from 1 or −1 to 0. That is, the CSD structure might be destructed when the new value of the mutation point is 1 or −1. Consequently, the focus in this paper is only on the point of mutation and its two neighbor points. To check and solve this problem, the following steps are performed.

Step 1. First, inspect the value of the point of mutation. As mentioned before, if the original value of the mutation point is 1 or −1, the structure will not be influenced after mutation. However if it is 0, the mutation transforming from 0 to 1 or −1 might cause the structure to change, so the structure cannot further follow the CSD structure.

Step 2. This step is to inspect the values of the points adjacent to the mutational point. If the value of any of them is 1 or −1, suitable changes are required. If the adjacent values are both 0, no changes are required.

Step 3. If any necessary change is detected in Step 2, find the last significant bit in a series of "1" or "−1" or alternation of "1" and "−1" change it to zero. Repeat this step until the CSD format is retained.

By following the above three steps, the chromosomes will stay in CSD. Therefore, the CSD format will maintain after crossover and mutation proposed in this paper. Consequently, reinspection or code renewal is not necessary. With this method, the computation time for evolution can be reduced. Figure 3 shows the procedure of a CSD-based mutation. Again, after the CSD-based mutation, the CSD format is preserved.

4. Stability Criterion and the Proposed Strategy

A theorem which is useful in the evolution of GA for a design of robust stable IIR filters is introduced in this section. Before proceeding, we introduce a useful lemma as follows.

Lemma 4.1 (see [35]). *If $f(z)$ is analytic in a bounded domain Ψ and continuous in the closure of Ψ, then $|f(z)|$ takes its maximum on the boundary of Ψ.*

One is now to derive the main theorem. The following theorem will provide a stability criterion for the $PD(\alpha, r)$-stability of a polynomial.

Theorem 4.2. *Consider the polynomial $d(z) \equiv \sum_{k=0}^{n} a_k z^{-k}$. If the following inequality (4.1) is satisfied, then all the solutions of $d(z) = 0$ will lie inside a disk $D(\alpha, r)$, that is, the polynomial $d(z)$ is $PD(\alpha, r)$-stable with $|\alpha| + r < 1$ and $|\alpha| \leq r$.*

$$\sum_{k=1}^{n} \frac{a_k}{a_0} (|r| - a)^{-k} < 1, \tag{4.1}$$

Proof. From Definition 2.1, it is obvious that if $|d(z)| = |\sum_{k=0}^{n} a_k z^{-k}| > 0$, for all $|z| \geq 1$, then the polynomial $d(z)$ is P-stable. Furthermore, according to Definition 2.2, if $|d(z)| = |\sum_{k=0}^{n} a_k z^{-k}| > 0$, for all $|z - \alpha| > r$ (i.e., $|(z - \alpha)/r| > 1$), then the polynomial $d(z)$ is $PD(\alpha, r)$-stable.

Let $\beta = (z - \alpha)/r$, that is, $z = r\beta + \alpha$. We have $|d(z)| = |\sum_{k=0}^{n} a_k z^{-k}| = |\sum_{k=1}^{n} a_k (r\beta + \alpha)^{-k}|$. Consequently, if

$$\left| \sum_{k=0}^{n} a_k (r\beta + \alpha)^{-k} \right| > 0, \quad \forall |\beta| \geq 1, \tag{4.2}$$

then the polynomial $d(z)$ is $PD(\alpha, r)$-stable from Definition 2.2.

Letting $\delta = \beta^{-1}$, the inequality (4.2) becomes

$$\left| \sum_{k=0}^{n} a_k \left(r\delta^{-1} + \alpha \right)^{-k} \right| > 0, \quad \forall |\delta| \leq 1. \tag{4.3}$$

Note that

$$\sum_{k=0}^{n} a_k \left(r\delta^{-1} + \alpha \right)^{-k} = a_0 \left[1 + \sum_{k=1}^{n} \frac{a_k}{a_0} \left(r\delta^{-1} + \alpha \right)^{-k} \right]. \tag{4.4}$$

Since the roots of the terms $(r\delta^{-1} + \alpha)^{-k}$, $k = 1, 2, \ldots, n$, are multiple roots at $\delta = -r/\alpha$ and $|\delta| = |r|/|\alpha| > 1$, $(r\delta^{-1} + \alpha)^{-k}$ is analytic for all $|\delta| \leq 1$. This implies that $\sum_{k=1}^{n} (a_k/a_0)(r\delta^{-1} + \alpha)^{-k}$ is analytic on the closed and bounded region $|\delta| \leq 1$. Hence, by

Lemma 4.1, $|\sum_{k=1}^{n}(a_k/a_0)(r\delta^{-1}+\alpha)^{-k}|$ will take its maximum on the boundary $|\delta| = 1$. Consider $\delta = e^{-j\theta}$, that is, the boundary $|\delta| = 1$, we have $|\sum_{k=1}^{n}(a_k/a_0)(re^{j\theta}+\alpha)^{-k}| \le \sum_{k=1}^{n}(a_k/a_0)|(re^{j\theta}+\alpha)|^{-k} \le \sum_{k=1}^{n}(a_k/a_0)(|r|-\alpha)^{-k}$. Consequently, if the inequality (4.1) is satisfied, then $|\sum_{k=1}^{n}(a_k/a_0)(r\delta^{-1}+\alpha)^{-k}| < 1$, for all $|\delta| = 1$. This means that

$$\left|\sum_{k=1}^{n}\frac{a_k}{a_0}\left(r\delta^{-1}+\alpha\right)^{-k}\right| < 1, \quad \forall|\delta| \le 1. \tag{4.5}$$

According to (4.4) and (4.5), we have $|\sum_{k=0}^{n}a_k(r\delta^{-1}+\alpha)^{-k}| > 0$, for all $|\delta| \le 1$. Consequently, from (4.2) and (4.3), the polynomial $d(z)$ is $PD(\alpha, r)$-stable. This completes the proof. □

Remark 4.3. The stability criterion proposed in Theorem 4.2 can avoid the boundary checking procedure which is necessary in the stability criterion proposed in the paper [2]. This means that the stability checking procedure by using the proposed stability criterion can save much computation time than that by using the stability criterion in [2].

Theorem 4.2 will be used to check whether the poles of an IIR filter all lie inside the disk $D(\alpha, r)$. This stability check should be performed in each generation of an evolutionary algorithm which is used to design IIR filters. Before proceeding, we first define the vector of the coefficients in the denominator of $H(z)$, $a = (1 a_1 a_2 \cdots a_n)$, as a chromosome of GA. The following definition is fundamental to understand this paper.

Definition 4.4. A chromosome $a = (a_0\ a_1 a_2 \cdots a_n)$ is stable if the corresponding polynomial $a(z) = \sum_{k=0}^{n} a_k z^{-k}$ is P-stable. Moreover, it is $D(\alpha, r)$-stable if the corresponding polynomial is $PD(\alpha, r)$-stable.

After the derivation of the stability criterion for the design of an IIR filters, a design strategy of robust CSD-coded stable IIR filters is then proposed as follows. Moreover, the flowchart of the design procedure is depicted in Figure 4 for clarification.

Step 1 (initial generation). Set the coefficients of the denominator of the $H(z)$, $a(z) = \sum_{k=0}^{N} a_k z^{-k}$, as a chromosome $a = (a_0 a_1 a_2 \cdots a_n)$. Generate the initial generation of the chromosomes based on the CSD code format.

Step 2 (check the stability property). Check the stability of the chromosome according to Theorem 4.2 and Definition 4.4. If a chromosome does not satisfy the criterion in Theorem 4.2, then regenerate a new chromosome based on CSD code format.

Step 3 (evaluate the fitness value of the chromosomes). Evaluate the fitness value of the chromosomes according to (3.4).

Step 4 (check whether the result is acceptable). If the result is acceptable or the number of iterations is larger than an assigned maximum number, go to the end of this procedure (Step 7), or go to next step.

Step 5 (generate offspring). Generate new chromosomes by the crossover and mutation based on the CSD format which are proposed in Section 4.

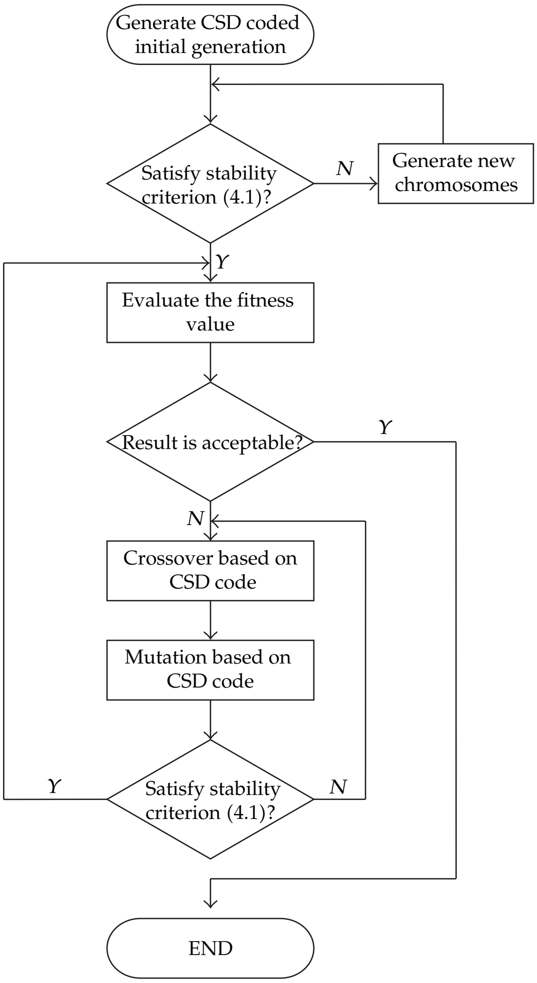

Generate CSD coded initial generation
Satisfy stability criterion (4.1)?
N
Generate new chromosomes
Y
Evaluate the fitness value
Result is acceptable?
Y
N
Crossover based on CSD code
Mutation based on CSD code
Y
Satisfy stability criterion (4.1)?
N
END

Figure 4: The design procedure of the proposed strategy.

Step 6 (check the stability criterion). Check the new chromosomes generated from Step 5 to see whether they satisfy the stability criterion in Theorem 4.2. Go to Step 3 if it does or go back to Step 5.

Step 7 (end of this procedure).

5. Simulation Results

Suppose that the transfer function of a CSD-coded IIR filter is described as $H(z) = (b_0 + b_1z^{-1})/(a_0 + a_1z^{-1} + a_2z^{-2})$, in which the coefficients a_i, $i = 0, 1, 2$, and b_i, $i = 0, 1$, are all designed in the CSD format. The desired frequency response is $H_I(\Omega)$ as depicted in Figure 5. The sampling rate, pass band edge frequency, and stop-band edge frequency of $H_I(\Omega)$ are set to be 9600 Hz, 1650 Hz, and 2800 Hz, respectively. The CSD-coded IIR filter is designed to

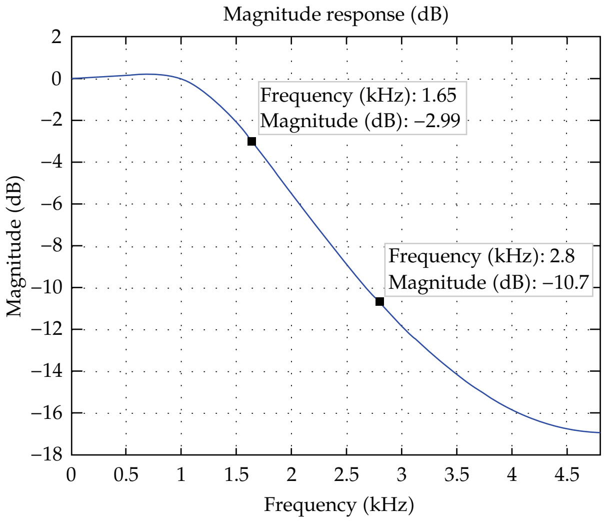


Figure 5: The desired frequency response.

Table 1: Parameters for GA.

Selection method	Roulette wheel
Crossover	CSD-based crossover
Mutation	CSD-based mutatiom
Chromosomes for each generation	40
No. of bits for each chromosome	90
Probability for crossover	0.9
Probability for mutation	0.08
No. of generations	35000

minimize the mean square error between $H(\Omega)$ and $H_I(\Omega)$ and to have the poles and zeros of $H(z)$ lie inside the disk $D(0.3, 0.7)$. That is, the IIR filter will be robustly $D(0.3, 0.7)$-stable.

According to Steps 1 and 7 of the design strategy proposed in Section 5, in which the parameters of GA are listed in Table 1, the coefficients of the IIR filter are designed after 35000 generations of GA as CSD-coded format. The transfer function of designed CSD-coded IIR filter is then described as

$$H(z) = \frac{-4990 - 1250z^{-1}}{-12505 + 10000z^{-1} - 3750z^{-2}} = \frac{-0.499 - 0.125z^{-1}}{-1.2505 + 1z^{-1} - 0.375z^{-2}}. \tag{5.1}$$

Table 2 lists the designed CSD-coded coefficients and the numbers of operations for the CSD-coded coefficients while Table 3 lists the binary-coded coefficients and the numbers of operations for the binary-coded coefficients. In order to make the representation of the tables more compact, the value −1 in these tables is replaced by the notation $\overline{1}$. It is obvious that the numbers of operations for CSD-coded coefficients are less than those of binary-coded coefficients. This means that the CSD-coded coefficients have less computational cost than

Table 2: Numbers of operations for the designed CSD code of the coefficients.

Coefficient symbols	Scaled coeffiicients	CSD code	No. of add operations	No. of subtract operations	Total operations
a_0	−12505	0000 0$\bar{1}$01 000$\bar{1}$ 0010 100$\bar{1}$	3	3	6
a_1	10000	0000 0010 100$\bar{1}$ 0001 0000	3	1	4
a_2	−3750	0000 000$\bar{1}$ 0010 $\bar{1}$0$\bar{1}$0 $\bar{1}$010	2	4	6
b_0	−4990	0000 000$\bar{1}$ 0$\bar{1}$00 1000 0010	2	2	4
b_1	−1250	0000 0000 0$\bar{1}$0$\bar{1}$ 0001 00$\bar{1}$0	1	3	4

Table 3: Numbers of operations for the binary code of the coefficients.

Coefficient symbols	Scaled coefficients	Binary code	Total operations
a_0	−12505	1000 0011 0000 1101 1001	8
a_1	10000	0000 0010 0111 0001 0000	5
a_2	−3750	1000 0000 1110 1010 0110	8
b_0	−4990	1000 0001 0011 0111 1110	10
b_1	−1250	1000 0000 0100 1110 0010	6

Table 4: Precision and optimality of the designed coefficients.

Simplified coefficients	Trained by the proposed CSD-coded GA	Optimal solution without CSD coded	Differences between the designed and optimal coefficients
a_0	−1.2505	−1.2505	0
a_1	1	1.0004	0.0004
a_2	−0.375	−0.37505	0.00005
b_0	−0.499	−0.49895	0.00105
b_1	−0.125	−0.12505	0.00005

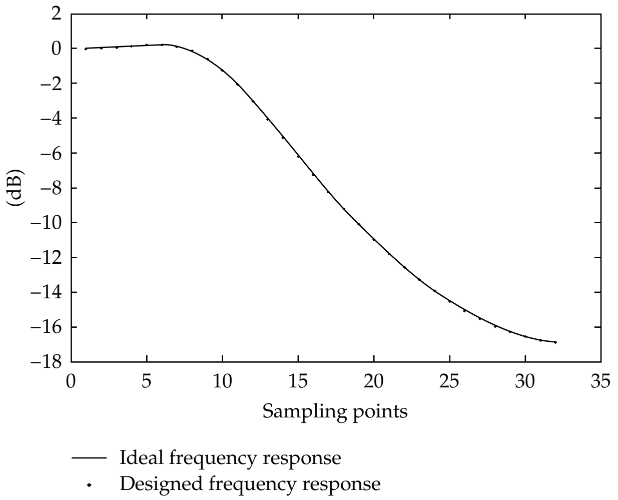


Figure 6: Comparison between the frequency responses of idea filter and designed filter.

Table 5: Precision and optimality of the designed poles.

Poles designed by the proposed CSD-coded GA	Optimal poles locations	Distance between the designed and optimal poles
$0.3998 + j0.3741$	$0.4 + j0.374$	0.000224
$0.3998 - j0.3741$	$0.4 - j0.374$	0.000224

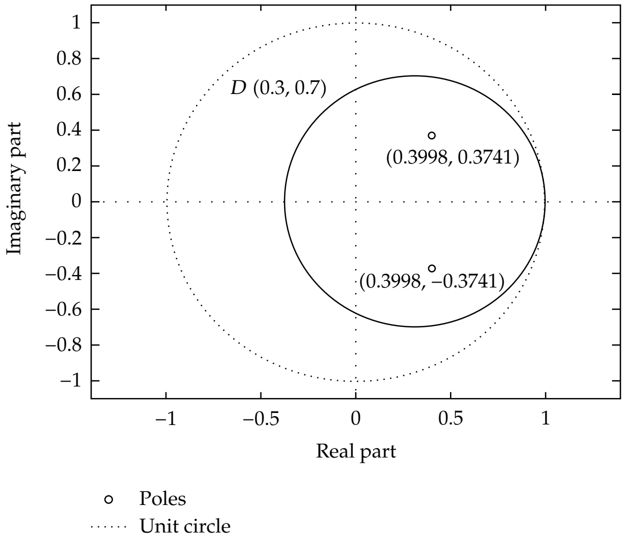


Figure 7: Poles of the designed IIR filter.

binary coded coefficients. Moreover, the precision and optimality of the designed CSD-coded coefficients and designed poles are illustrated in Tables 4 and 5, respectively. These two tables reveal that the designed coefficients and poles of the IIR filter are precise and are near optimal, since they are all very close to the optimal coefficients and poles for this design problem. Besides, the frequency response of the desired IIR filter and the designed IIR filter is depicted in Figure 6. It can be seen that the designed frequency response is very close to that of the desired filter. Moreover, the poles of the transfer function of the IIR filter are depicted in Figure 7. It is also obvious that all the poles lie inside the disk $D(0.3, 0.7)$. That is, the CSD-coded IIR filter is robustly $D(0.3, 0.7)$-stable.

6. Conclusions

A robust stability criterion is derived for the design of a robust IIR filters. A strategy is proposed based on GA to design an IIR filter so that its coefficients satisfy the robust stability criterion and the CSD code format simultaneously. This proposed CSD-based GA wouldnot destruct the filter coefficient format and hence can maintain the CSD code format after each generation of chromosome evolution. The recursive try-and-error step is not required and consequently can save a lot of time. Moreover, a more efficient stability criterion than that in [12] is proposed to be checked in the evolution of GA to ensure that stability of the designed

IIR filter. According to the final simulated results in the paper, it is shown that the proposed algorithm has a quite good efficiency.

Acknowledgement

This research work was supported by the National Science Council of the Republic of China under contract NSC 100-2221-E-390-025-MY2.

References

[1] W.-S. Lu and T. Hinamoto, “Optimal design of IIR digital filters with robust stability using conic-quadratic-programming updates,” *IEEE Transactions on Signal Processing*, vol. 51, no. 6, pp. 1581–1592, 2003.

[2] S.-T. Pan, “Design of robust D-stable IIR filters using genetic algorithms with embedded stability criterion,” *IEEE Transactions on Signal Processing*, vol. 57, no. 8, pp. 3008–3016, 2009.

[3] S.-T. Pan, “Evolutionary computation on programmable robust IIR filter pole-placement design,” *IEEE Transactions on Instrumentation & Measurement*, vol. 60, no. 4, pp. 1469–1479, 2011.

[4] M. D. Lutovac, J. D. Ćertić, and L. D. Milić, “Digital filter design using computer algebra systems,” *Circuits, Systems, and Signal Processing*, vol. 29, no. 1, pp. 51–64, 2010.

[5] G. Li, J. Hua, and G. Bi, “An efficient and robust orthogonal structure for linear discrete-time systems,” *Circuits, Systems, and Signal Processing*, vol. 29, no. 4, pp. 727–743, 2010.

[6] S. Fiori, “Blind adaptation of stable discrete-time IIR filters in state-space form,” *IEEE Transactions on Signal Processing*, vol. 54, no. 7, pp. 2596–2605, 2006.

[7] J. Konopacki, “The frequency transformation by matrix operation and its application in IIR filters design,” *IEEE Signal Processing Letters*, vol. 12, no. 1, pp. 5–8, 2005.

[8] W. S. Lu, “Design of stable IIR digital filters with equiripple passbands and peak-constrained least-squares stopbands,” *IEEE Transactions on Circuits and Systems II*, vol. 46, no. 11, pp. 1421–1426, 1999.

[9] L. Rosenbaum, P. Löwenborg, and H. Johansson, “Two classes of cosine-modulated IIR/IIR and IIR/FIR NPR filter banks,” *Circuits, Systems, and Signal Processing*, vol. 29, no. 1, pp. 103–133, 2010.

[10] S. Fiori, “Blind intrinsically-stable 2-pole IIR filtering,” *Electronics Letters*, vol. 38, no. 23, pp. 1482–1483, 2002.

[11] P. Hao, W. Zanji, and C. Jianye, “A measuring method of the single-phase AC frequency, phase, and reactive power based on the Hilbert filtering,” *IEEE Transactions on Instrumentation and Measurement*, vol. 56, no. 3, pp. 918–923, 2007.

[12] C.-C. Tseng, “Minimax design of stable IIR digital filter with prescribed magnitude and phase responses,” *IEEE Transactions on Circuits and Systems I*, vol. 49, pp. 547–551, 2002.

[13] K. Uesaka and M. Kawamata, “Synthesis of low-sensitivity second-order digital filters using genetic programming with automatically defined functions,” *IEEE Signal Processing Letters*, vol. 7, no. 4, pp. 83–85, 2000.

[14] K. Uesaka and M. Kawamata, “Evolutionary synthesis of digital filter structures using genetic programming,” *IEEE Transactions on Circuits and Systems II*, vol. 50, no. 12, pp. 977–983, 2003.

[15] Y. Yang and X. Yu, “Cooperative coevolutionary genetic algorithm for digital IIR filter design,” *IEEE Transactions on Industrial Electronics*, vol. 54, no. 3, pp. 1311–1318, 2007.

[16] J. Zhou, G. Li, and M. Sun, “A structural view of stability in adaptive IIR filters,” *IEEE Transactions on Signal Processing*, vol. 54, no. 12, pp. 4828–4835, 2006.

[17] Y. Cao, K. Wang, W. Pei, Y. Liu, and Y. Zhang, “Design of high-order extrapolated impulse response FIR filters with signed powers-of-two coefficients,” *Circuits, Systems, and Signal Processing*, vol. 30, no. 5, pp. 963–985, 2011.

[18] E. Cabal-Yepez, T. D. Carozzi, R. D. J. Romero-Troncoso, M. P. Gough, and N. Huber, “FPGA-based system for frequency detection of the main periodic component in time series information,” *Digital Signal Processing*, vol. 18, no. 6, pp. 1029–1044, 2008.

[19] Z. Tang, J. Zhang, and H. Min, “A high-speed, programmable, CSD coefficient FIR filter,” *IEEE Transactions on Consumer Electronics*, vol. 48, no. 4, pp. 834–837, 2002.

[20] W. Gass, “Automatic generation of high-performance ASICs,” *Digital Signal Processing*, vol. 1, no. 1, pp. 15–16, 1991.

[21] A. T. G. Fuller, B. Nowrouzian, and F. Ashrafzadeh, “A novel modified branch-and-bound technique for discrete optimization over canonical signed-digit number space,” in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS ’98)*, vol. 5, pp. 391–394, June 1998.

[22] R. L. Haupt and S. E. Haupt, *Practical Genetic Algorithms*, John Wiley & Sons, New York, NY, USA, 1998.

[23] F. H. F. Leung, H. K. Lam, S. H. Ling, and P. K. S. Tam, “Tuning of the structure and parameters of a neural network using an improved genetic algorithm,” *IEEE Transactions on Neural Networks*, vol. 14, no. 1, pp. 79–88, 2003.

[24] M. Gulek and I. H. Toroslu, “A genetic algorithm for maximum-weighted tree matching problem,” *Applied Soft Computing Journal*, vol. 10, no. 4, pp. 1127–1131, 2010.

[25] S. Grzegorz Zurek, “Genetic algorithm with peaks adaptive objective function used to fit the EPR powder spectrum,” *Applied Soft Computing Journal*, vol. 11, no. 1, pp. 1000–1007, 2011.

[26] J. T. Tsai, J. H. Chou, and T. K. Liu, “Optimal design of digital IIR filters by using hybrid Taguchi genetic algorithm,” *IEEE Transactions on Industrial Electronics*, vol. 53, no. 3, pp. 867–879, 2006.

[27] K. S. Tang, K. F. Man, S. Kwong, and Z. F. Liu, “Design and optimization of IIR filter structure using hierarchical genetic algorithms,” *IEEE Transactions on Industrial Electronics*, vol. 45, no. 3, pp. 481–487, 1998.

[28] P. D. Karamalis, A. G. Kanatas, and P. Constantinou, “A genetic algorithm applied for optimization of antenna arrays used in mobile radio channel characterization devices,” *IEEE Transactions on Instrumentation and Measurement*, vol. 58, no. 8, pp. 2475–2487, 2009.

[29] S.-T. Pan and C. C. Lai, “Identification of chaotic systems by neural network with hybrid learning algorithm,” *Chaos, Solitons and Fractals*, vol. 37, no. 1, pp. 233–244, 2008.

[30] M. Haseyama and D. Matsuura, “A filter coefficient quantization method with genetic algorithm, including simulated annealing,” *IEEE Signal Processing Letters*, vol. 13, no. 4, pp. 189–192, 2006.

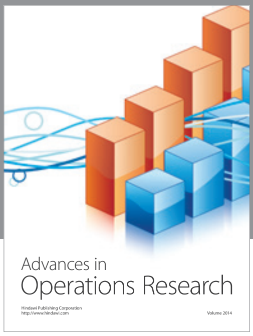
[31] Q. Zhao and Y. Tadokoro, “A simple design of FIR filters with powers-of-two coefficients,” *IEEE Transactions on Circuits and Systems*, vol. 35, no. 5, pp. 566–570, 1988.

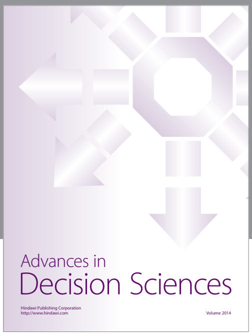
[32] K. Y. Khoo, A. Kwentus, and A. N. Willson, “A programmable FIR digital filter using CSD coefficients,” *IEEE Journal of Solid-State Circuits*, vol. 31, no. 6, pp. 869–874, 1996.

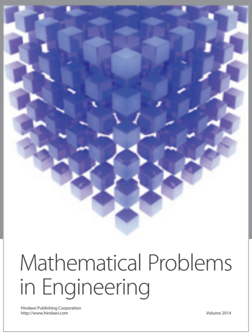
[33] H. R. Lee, C. W. Jen, and C. M. Liu, “A new hardware-efficient architecture for programmable FIR filters,” *IEEE Transactions on Circuits and Systems II*, vol. 43, no. 9, pp. 637–644, 1996.

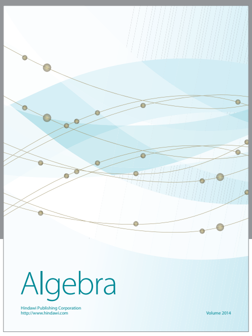
[34] I. C. Park and H. J. Kang, “Digital filter synthesis based on an algorithm to generate all minimal signed digit representations,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 21, no. 12, pp. 1525–1529, 2002.

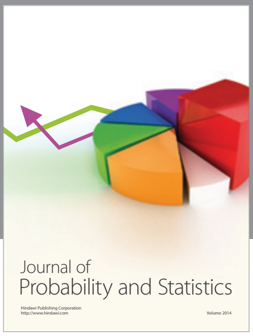
[35] W. D. John, *Applied Complex Variables*, The Macmillan, New York, NY, USA, 1967.


Advances in
Operations Research

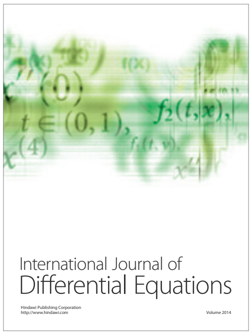

Advances in
Decision Sciences

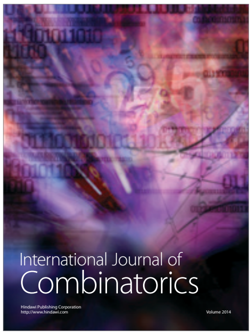

Mathematical Problems
in Engineering

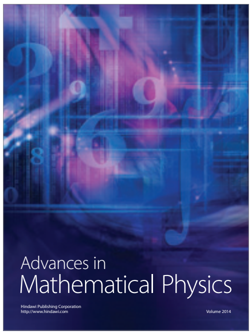

Algebra

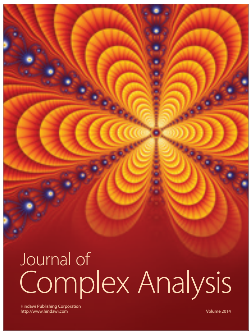

Journal of
Probability and Statistics

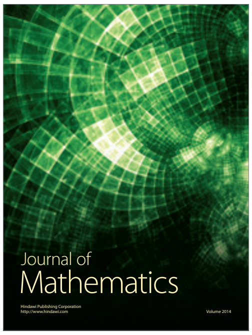
The Scientific
World Journal

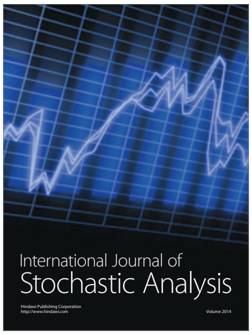

International Journal of
Differential Equations

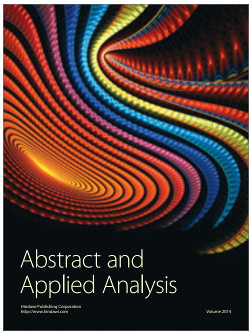

International Journal of
Combinatorics

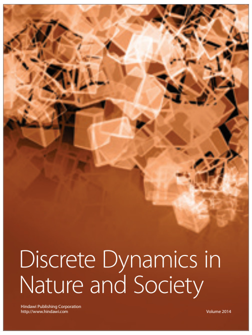

Advances in
Mathematical Physics

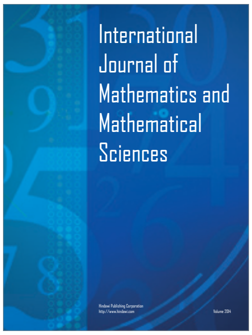

Journal of
Complex Analysis


Journal of
Mathematics


International Journal of
Stochastic Analysis

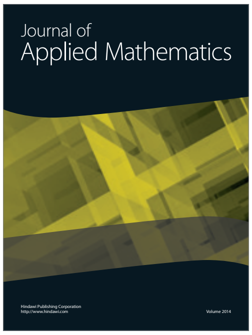

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


Journal of
Applied Mathematics

Journal of
Optimization