Evolution Based Structure Design for Low-Pass IIR Digital Filters with Fault Tolerance

Lijia Chen, Mingguo Liu, Zhen Dai, Ying Chen

***Abstract*—Fault-tolerant digital filters are widely expected for use in modern electronic systems. In this paper, a structure evolution based optimization algorithm (SEOA) for designing an low-pass IIR digital filter is proposed. The filter structure is designed to alleviate the impact of concurrent permanent multiplier failures. A IIR digital filter structure is created by a new mechanism proposed in this paper, which provides not only structural validation and diversity but also a specific structure generation. Genetic algorithm (GA) is redesigned to integrate the mechanism for ensuring that structures are evolved validly and efficiently. In the improved GA, structures with different sizes are crossed by a single point way, which expands the solution space of structures. Desirable structures can be created by adjusting structural parameters, such as the structure size or the connection type. Simulation results show that filter performance is greatly affected by the concurrent short and open circuit faults of multipliers. Compared with the classic IIR digital filter implementations, SEOA improves the fault tolerance of the filter.**

***Index Terms*—IIR digital filters, structure design, fault tolerance, genetic algorithm.**

I. INTRODUCTION

DIGITAL filter has been widely used [1], [2] in electronic devices. In some areas, it is expected that the digital filter has a strong fault tolerance [3]. For example, the outer space circuit can tolerate a certain degree of cosmic particle impact [4]; critical computing, such as nuclear computing, can tolerate occasional failures, without affecting results. Some faults transiently change the states of a system [5], [6]. For the transient faults, fault-tolerant techniques have been proposed for digital filters [7], [8]. However, certain outer reasons may cause permanent damage to the filter hardware. For instance, environmental humidity, circuit aging, particle and electromagnetic interference, may cause insulation damage, resulting in circuit faults. In long-term unattended environments, some filter components may be destroyed permanently.

Some evolutionary algorithms, such as genetic algorithm (GA) [9], simulated annealing (SA) [10], differential evolutionary (DE) [11], particle swarm optimization (PSO) [12] and their variants, have been employed for the design of IIR digital filters. They optimize the coefficients of the transfer function to achieve desired frequency responses. But these algorithms can't deal with fault tolerance of digital filters because they don't design the filter structure. In this paper, fault tolerance of the digital filter is investigated against permanent circuit faults of multipliers. A structure evolution based optimization algorithm (SEOA) is proposed for designing the IIR low-pass digital filter. This approach creates and evolves IIR digital filter structures. GA is improved by integrating the structure generation so that it can validly and efficiently evolve the filter structure and improves fault tolerance of the filter. By comparing with classic implementations of digital filters, the simulation results show that the digital filter designed in this paper has excellent fault tolerance.

The organization of the paper is as follows. In section II, the structure design of IIR digital low-pass filters is presented. Experimental results are provided in section III. Finally, we give a conclusion in section IV.

II. ALGORITHM DESCRIPTION

SEOA is designed to generate and evolve IIR digital filter structures. A structurally automatic generation algorithm (SAGA) is proposed. SAGA allows the IIR digital filter structure to be created in a specific way that only valid structures are produced and at the same time that the structures are diversified. The evolution of digital filter structures is such a problem that filters perform distinctively when their structures have different scales. In SEOA, filter structures with different sizes are investigated and coded as chromosomes according to SAGA. GA has the potential of holding high performance for those problems with varied-length chromosomes [13, 14], which supports a single point crossover for filter structures. GA is improved to integrate SAGA with the code for validly and efficiently evolving filter structures. The code of GA is one of the key factors influencing its performance [15]. An elaborate combined code is designed to represent a filter structure. The fitness function is designed as a weighed attenuation error between the desired response and the designed response. The fitness decreases until a maximum generation is reached or it satisfies a given threshold.

SAGA starts the generation of filter structures with one node that is the input node of the filter. It grows by attaching the delay element and the multiplier to it one by one. The delay element and the multiplier are denoted separately by T_d and T_m. The adder is determined after the structure has been finished. Those nodes with more than one input are adders. Connections of digital elements in a structure are designed in four modes which comply with the following rules. First of all, an active node is defined as a node to which any new attached element must be linked. At the beginning of the algorithm, the input node of the filter is an active node. An element is attached to the active node in four modes. While its one terminal is connected to the active node, the other one is allowed to link itself with a newly generated active

Manuscript received September 7, 2017; revised June 13, 2018. This work was supported in part by the Foundation of Henan Province, China, with the grant numbers: 15A510018, 15A510019, 12A510002 and 142102210629. This work was also supported by the Foundation of Henan University, China, with the grant numbers: 2008YBZR028 and ZZJJ20140037.

L. Chen, M. Liu, Z. Dai and Y. Chen are with the Laboratory of Advanced Computation Methods and Intelligent Applications, School of Physics and Electronics, Henan University, China e-mail: (L. Chen: chenlijia just@163.com; M. Liu: mingguo@mail.ustc.edu.cn).

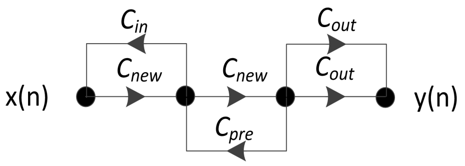

Fig. 1. An example of a structure generated by the four connections

node, a previous node, the system input node or the system output node. The four types of connection are denoted as C_{new}, C_{pre}, C_{in} and C_{out}. After a new active node appears, the old one will not be an active node any more. An example shows that elements are connected from $x(n)$ to $y(n)$ in the instruction order of C_{new}-C_{in}-C_{new}-C_{pre}-C_{out}-C_{out} in Fig. 1.

The system output node always exists during the structure generation. A special case is to attach the last element to the structure. The element is constrained to use C_{out} so that the system output node is definitely linked to the structure.

Signal directions are dealt when those elements are appended to the structure. Every element has a signal direction. In a valid digital structure, the direction of two linked elements mustn't collide with each other. For example, if one circuit branch bears more than one element, these elements must have the same signal direction. Considering one node with d_i signals flowing into it and d_o signals flowing out of it, the direction constraint of digital structures can be

$$d_i > 0 \quad and \quad d_o > 0, \tag{1}$$

where every node must satisfy equation 1 except the input and output nodes of the system. We define such a node as a middle node. The system input node must have an output, and the system output node must have an input.

The structures created by SAGA are always valid, which is proved as follows. A valid structure requires that all the middle nodes in the structure must satisfy equation 1. According to the connection modes, a middle node gets at least one input from its previous active node, which creates the middle node by using C_{new}. The middle node is inevitable to advance to the next active node by C_{new} or to the output node of the system by C_{out}. Whether C_{new} or C_{out} produces an output from the middle node. Equation 1 holds for every middle node and therefore the structure is always valid.

A structure is coded as an instruction sequence in the way of structure generation. Each element attached to the structure is dictated by an instruction. An instruction is denoted as $T_{con}V_iV_oP_cT_{com}$, where T_{con} denotes the connection type. V_i and V_o are the nodes to which an attached element is linked. Signals always flow from V_i to V_o through the element. T_{com} indicates the element type which can be T_d or T_m. P_c is the element's parameter. An instruction sequence reflects the generation of a digital structure. The structure can reappear when elements are driven by following the instruction sequence. In SEOA, an instruction sequence is a chromosome of GA. Table I shows a 7-element instruction sequence, and Fig. 2 is the structure created by the sequence.

The sequence includes 7 instructions for forming the digital filter. Active nodes are created one by one, numbered by N_i incrementally. The final structure includes 5 multipliers, 2 delay elements and 3 adders. The adders are node N_1, N_2 and N_4.

Digital filter structures are evolved based on GA. GA is improved to integrate SAGA with code into the genetic operations. Crossover, mutation and selection genetic operators are improved and the fitness function is defined.

First, the crossover operator is elaborately designed for filter structures. The single-point crossover is adopted through the whole evolution. A crossing point divides an instruction sequence into two parts. The crossover causes an exchange of the front parts between the two sequences. Considering the generation process of a digital structure in SAGA, the

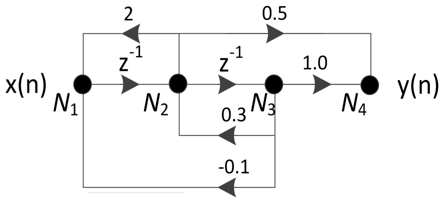

Fig. 2. The generated filter structure according to Table I

TABLE I
AN EXAMPLE FOR THE INSTRUCTION SEQUENCE

ID	T_{con}	V_i	V_o	P_c	T_{com}
1	C_{new}	N_1	N_2	-1	T_d
2	C_{in}	N_2	N_1	2.0	T_m
3	C_{out}	N_2	N_4	0.5	T_m
4	C_{new}	N_2	N_3	-1	T_d
5	C_{pre}	N_3	N_2	0.3	T_m
6	C_{in}	N_3	N_1	-0.1	T_m
1	C_{out}	N_3	N_4	1.0	T_m

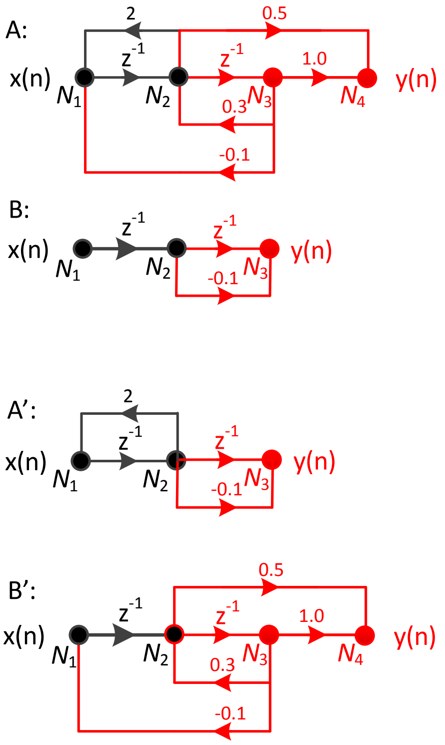

Fig. 3. Crossover of two structures with different sizes

new sequences are valid when the two crossing points have the same number of N_i. Under the condition, the front part of one structure grows in the way of the rear part of the other structure. Two structures produce valid descendants after they cross with each other in the single-point way. An example is shown in Fig. 3. Structure A corresponds to an instruction sequence of $|C_{new}N_1N_2 - 1T_d|C_{in}N_2N_1 2.0T_m|C_{out}N_2N_4 0.5T_m|C_{new}N_2N_3 - 1T_d|$ $C_{pre}N_3N_2 0.3T_m|C_{in}N_3N_1 - 0.1T_m|C_{out}N_3N_4 1.0T_m|$ and structure B corresponds to that of $|C_{new}N_1N_2 - 1T_d|C_{out}N_2N_3 - 1T_d|C_{out}N_2N_3 - 0.1T_m|$. The crossing point is N_2 which divides A and B into two parts as the black and red parts described in Fig. 3. A' and B' are the crossed structures from A and B. They are valid when A and B are valid.

Second, the mutation operator is designed to change a chromosome by replacing it with a newly created chromosome. The new chromosome has the same length with its predecessor. The mutation brings more changes than a single instruction mutation which only mutates an instruction of the chromosome.

Third, the selection operator is designed to sort chromosomes by their fitness values. P_s chromosomes with the smallest fitness values will be kept down as the next generation of chromosomes, where P_s is the population size.

Finally, the fitness function for evaluating digital filter structures is defined. The transfer function of a digital filter is derived from its structure as described in [16]. Each internal signal in the system is expressed as the output of a node.

$$\mathbf{W}(z) = (\mathbf{I} - \mathbf{Q}(z))^{-1}\mathbf{P}(z). \tag{2}$$

W records the expressions of internal signals. **I** is an identity matrix. **Q** is a connection matrix, which expresses the relations between internal signals. **P** denotes a vector which records gains or delay from the system input signal to internal signals. The transfer function of the structure is

$$H(z) = \mathbf{W}_{out}(z), \tag{3}$$

where out is the identified number of the system output node. In SEOA, $H(z)$ is discretized by substituting $e^{j\pi i/n}$ for z. $H(K_i)$ gets the ith value of n samples of the frequency response as shown in equation 4.

$$H(K_i) = H(z)|_{z=e^{j\pi i/n}}. \tag{4}$$

A chromosome is evaluated through calculating its frequency-response error. Fitness, defined as the error, is estimated in

$$fitness = \frac{1}{n}\sum w_i log^2(\frac{|H(K_i)|}{|D(K_i)|}), D(K_i) \neq 0, \tag{5}$$

where $D(K)$ is the desired frequency response in a discrete form. The n is the number of sampling points. w_i is the weight for sampling point i.

$$w_i = \begin{cases} ki/n + 1/2 & 0 \leq i < n/2 \\ \lambda(k(1 - i/n) + 1/2) & n/2 \leq i < n \end{cases}, \tag{6}$$

where k controls the sharpness of the transition band. λ makes a performance tradeoff between the stopband and the passband.

TABLE II
THE PARAMETER SETTINGS OF SEOA

Parameter	Value	Parameter	Value
crossover rate	0.7	mutation rate	0.1
population size	100	chromosome length	50
maximum generation	10000	seed	[0.0,1.0]
multiplier gain	[-2.0,2.0]	sampling rate	128
k	2.0	λ	0.01

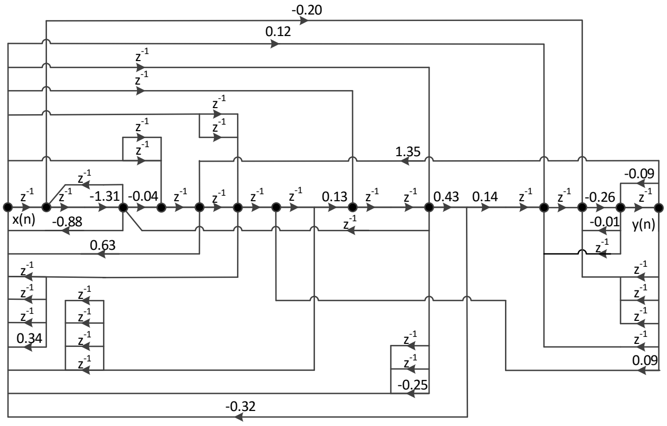

Fig. 4. The structure of the low-pass digital filter designed by SEOA

III. RESULTS AND DISCUSSION

In this section, a low-pass IIR digital filter structure is designed by SEOA and fault tolerance of the structure is compared with classic IIR digital filter implementations. The target filter is set as follows. Passband ranges from 0 rad to 0.45π rad, stopband covers from 0.6π rad to π rad, the minimum of passband attenuation is -1 dB and the maximum of stopband attenuation is -60 dB.

Simulation parameters are listed in Table II. The chromosome length is measured in instructions. The seed is used to create random digital structures for an instance. There are 30 random seeds in [0.0, 1.0] for 30 instances. The multiplier gain is a random real number and is confined within the range of [-2.0, 2.0]. Besides, the delay element is set to a unit delay. The desired frequency response and the designed frequency response are both sampled with 128 points. Each connection mode takes the chance of 0.25, and each element type takes that of 0.5.

The structure designed by SEOA is shown in Fig. 4. 100 chromosomes are evolved in 10000 iterations. The 50-element structure as shown in Fig. 4 is obtained, which consists of 13 adders, 33 delay elements and 17 multipliers. Its order is 9. The structure is produced by connections of 16 C_{new}s, 21 C_{in}s, 8 C_{out}s and 5 C_{pre}s. They are all randomly created and elements are connected according to them. Furthermore, the input node, the output node, the parameter and the element type are randomly generated to construct the filter structure.

We assume a circuit fault model as follows. Circuit faults are set to randomly happen on multipliers wherever the multipliers are located in a filter structure. Multiple multipliers can suffer the circuit faults at the same time and they aren't repaired after they are damaged. Three types of fault modes, the short circuit, the open circuit and the mixed circuit fault,

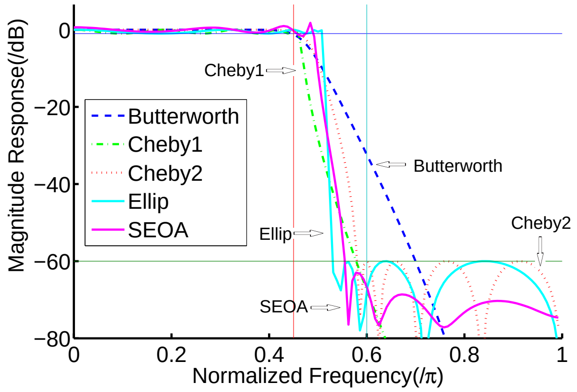

Fig. 5. The original magnitude frequency responses using different design methods and SEOA without any fault

are adopted. The mixed circuit fault is the mixture of the short circuit and the open circuit. The fault rate of multipliers is set to [1%, 7%] for the short circuit, which means that the quantity of the short-circuited multipliers is 1% - 7% of the total quantity of multipliers in a filter structure. The fault rates for the open circuit and the mixed circuit fault are [1%, 5%] and [2%, 10%], respectively. The structure in each case has 100 independent tests. Filter errors are composed of two parts, the error when the attenuation in passband is less than -1 dB or greater than 0 dB, and the error when the attenuation in stopband is greater than -60 dB. The errors are calculated based on equation 5.

For comparison, Butterworth, Cheby I, Cheby II and Ellip are utilized to design the same desired IIR digital filter. Their orders are set to 9. The magnitude frequency responses of them and SEOA without any fault are shown in Fig. 5. The filter specifications are well catered by these design methods except Butterworth. Butterworth is gradually decreasing in magnitude and fails to reach a width of 0.15π for the transition band. Other methods satisfy the design target. Cheby I has ripples in the passband and is monotonous in the stopband, while Cheby II is monotonous in the passband and has ripples in the stopband. Ellip gives the steepest transition band among all design methods. SEOA comes with some ripples in the passband and stopband.

A. The short circuit tests

The magnitude frequency responses using Butterworth are shown in Fig. 6. The direct II, cascade, parallel and lattice structures are compared with our structure for fault tolerance. The fault rate is set to 1%, 4% and 7% in the three subfigures of Fig. 6, respectively. The results show that errors increase with the increment of the fault rate. The magnitude responses of the direct II and lattice structures depart rapidly from their original positions. Their frequency responses have been damaged. The response of the parallel structure is greatly affected, which is almost all above 20 dB. SEOA and Butterworth with the cascade structure, perform best among all implementations. Compared with the cascade structure, SEOA excels by a relatively better passband and stopband performance. SEOA meets the specification of the desired filter when the fault rate is equal to 1% as shown in Fig. 6 (a). The passband of the filter surpasses the maximum bound of 0 dB as shown in Fig. 6 (b) when the fault rate is 3%. Moreover, the passband and the stopband are both beyond the desired attenuation when the fault rate is up to 7% as shown in Fig. 6 (c).

Error comparisons of our structure and the classic implementations of Butterworth are shown in Fig. 7. The filter errors are counted with their median, lower quartile q_1, upper quartile q_3, the maximum and minimum non-outliers, and outliers. Outliers are those values which are larger than $q_3+1.5*(q_3-q_1)$ or smaller than $q_1-1.5*(q_3-q_1)$. The worst performance of the filter is highlighted by the largest value of the outliers. In the results, the direct II, parallel and lattice structures have bigger filter errors than our structure. The lattice structure produces many invalid frequency responses, resulting in the incomplete box in Fig. 7 (c). The cascade structure takes smaller median values but larger outliers than our structure.

The comparison of our method with the design methods of Butterworth, Cheby I, Cheby II and Ellip is listed in Table III - VI, where 'Inf' indicates an infinite number and '-' indicates the result incalculable. The tables display the maximum, mean and variance of the errors in implementation of the filter with different structures. As it can be seen from the statistical results, the average trend of the errors is increasing when the fault rate increases from 1% to 7%. The smallest values of the maximum error, the error variance and the mean error in their separate columns are in bold in Table III - VI. The maximum error and the error variance of SEOA are smallest in all structures. The mean error is the second smallest among all structures. The results show that SEOA performs best among all methods when filter structures suffer the most serious damage at the same fault rate.

B. The open circuit tests

The open circuit faults are set with different fault rates from 0.01 to 0.05. Open circuits cause breaks in a filter structure, where signals fail to pass through the broken parts of the structure. Some structures are sensitive to the breaks. For instance, the cascade structure has a unique multiplier, whose break will cut the whole structure into two segments. Other structures are also affected by the breaks of multipliers to different extents. As depicted in Fig. 8, magnitude responses of these structures are greatly subject to fault rates. In these methods, SEOA is more stable compared with the other methods. From the statistical data in Fig. 9, cascade structures and parallel structures suffer great damages under the open circuits. Cascade structures have many large outliers and parallel structures get into enormous bias. SEOA is prominent in the respect of the average error and the maximum error.

The error comparison of the classic implementations and SEOA is shown in Table VII. The best performance is emphasized in bold. Our structure obtains the optimums of 12 indexes in all 15 indexes. Under the open circuit faults, the performance of the filters shows different characteristics from those under the short circuit faults. The cascade structure is ruined with a high probability because the open circuit faults break the structure and make it divided into segments.

C. The mixed fault tests

The mixed circuit faults, including open and short circuit faults, are set with the same fault rate ranging from 0.01 to

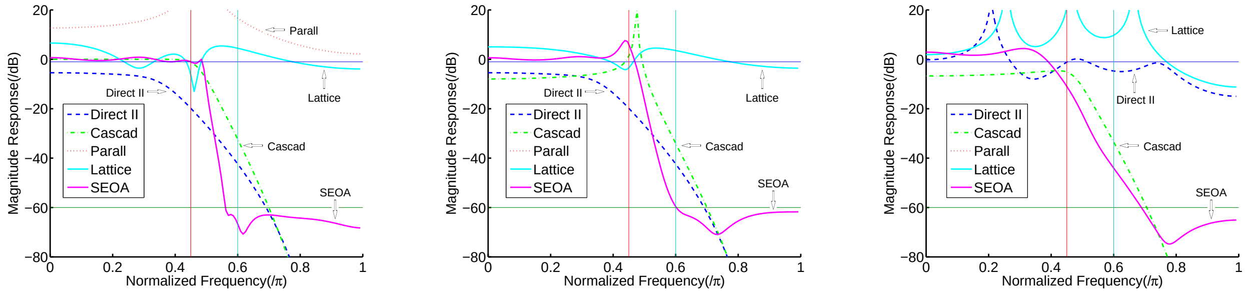

Fig. 6. The magnitude frequency response comparison with short circuits of multipliers using the Butterworth design method. (a)fault rate=1%, (b)fault rate=4%, (c)fault rate=7%

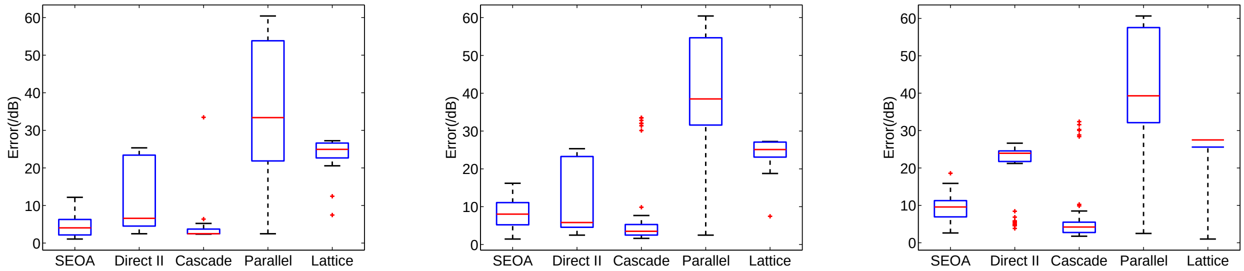

Fig. 7. The error comparison with short circuits of multipliers using the Butterworth design method. (a)fault rate=1%, (b)fault rate=4%, (c)fault rate=7%

TABLE III
ERROR COMPARISON OF FIVE STRUCTURES FOR IMPLEMENTING IIR DIGITAL FILTER USING BUTTERWORTH METHOD (DATA IN FORMAT: MAX(DB), MEAN(DB), VAR)

Fault rate	0.01	0.02	0.03	0.04	0.05	0.06	0.07
Direct II	25/14/96	25/15/93	25/13/92	25/13/93	25/115/93	27/18/79	27/20/60
Cascade	33/**3**/10	33/**5**/46	33/**4**/19	34/**6**/63	34/**5**/44	33/**5**/27	32/**6**/41
Parallel	60/33/376	60/29/274	60/29/372	60/39/315	60/41/214	61/44/188	61/43/206
Lattice	Inf/Inf/-	Inf/Inf/-	Inf/Inf/-	Inf/Inf/-	Inf/Inf/-	Inf/Inf/-	Inf/Inf/-
SEOA	**12**/4/**6**	**14**/6/**10**	**15**/6/**9**	**16**/8/**13**	**15**/7/**9**	**18**/9/**14**	**19**/9/**11**

TABLE IV
ERROR COMPARISON OF FIVE STRUCTURES FOR IMPLEMENTING IIR DIGITAL FILTERS USING CHEBY I METHOD (DATA IN FORMAT: MAX(DB), MEAN(DB), VAR

Fault rate	0.01	0.02	0.03	0.04	0.05	0.06	0.07
Direct II	24/19/48	24/19/45	24/20/41	24/19/36	24/19/44	26/21/15	26/21/10
Cascade	38/**3**/30	38/**4**/54	38/**4**/78	38/**4**/28	39/**8**/122	38/**6**/89	38/**7**/70
Parallel	29/18/112	29/29/110	29/19/95	31/23/60	33/24/44 7	32/24/57	31/24/45
Lattice	Inf/Inf/-	Inf/Inf/-	Inf/Inf/-	Inf/Inf/-	Inf/Inf/-	Inf/Inf/-	Inf/Inf/-
SEOA	**10**/4/**6**	**17**/6/**12**	**14**/7/**10**	**17**/7/**12**	**16/8/12**	**18**/9/**11**	**16**/9/**10**

TABLE V
ERROR COMPARISON OF FIVE STRUCTURES FOR IMPLEMENTING IIR DIGITAL FILTERS USING CHEBY II METHOD (DATA IN FORMAT: MAX(DB), MEAN(DB), VAR)

Fault rate	0.01	0.02	0.03	0.04	0.05	0.06	0.07
Direct II	26/11/100	26/11/108	26/13/105	26/13/102	26/11/100	28/19/85	28/19/80
Cascade	24/**3**/17	24/**3**/21	24/**3**/17	25/**4**/30	25/**4**/33	25/**5**/39	26/**5**/36
Parallel	50/29/251	50/26/270	50/28/243	50/35/122	51/35/126	51/35/133	50/36/118
Lattice	Inf/Inf/-	Inf/Inf/-	Inf/Inf/-	Inf/Inf/-	Inf/Inf/-	Inf/Inf/-	Inf/Inf/-
SEOA	**12**/4/**7**	**16**/6/**12**	**15**/6/**10**	**16**/8/**10**	**16**/8/**11**	**19**/9/**11**	**18**/9/**13**

0.05. So the total fault rates is doubled with [0.02, 0.10]. The mixed circuit faults make magnitude responses more deteriorate as depicted in Fig. 10. In this comparison, some independent tests often happen with totally destroyed filter structures, whose magnitude responses either reside above 20 dB or are nonexistent at all. Those absent magnitude responses in Fig. 10 belong to this case.

SEOA performs well in the aspects of the maximum error and the average error as shown in Fig. 11. Cascade structures go with more and bigger outliers than SEOA. SEOA's maximum error is the smallest among all implementations, which means the highest reliability of SEOA.

TABLE VI
ERROR COMPARISON OF FIVE STRUCTURES FOR IMPLEMENTING IIR DIGITAL FILTERS USING ELLIPTIC METHOD (DATA IN FORMAT: MAX(DB), MEAN(DB), VAR)

Fault rate	0.01	0.02	0.03	0.04	0.05	0.06	0.07
Direct II	22/13/51	22/13/44	22/13/55	22/13/51	22/13/51	24/17/33	24/16/33
Cascade	26/**3**/21	26/**3**/31	26/**3**/23	26/**6**/47	26/**5**/36	27/**5**/38	26/**6**/49
Parallel	29/18/107	29/16/112	29/16/128	34/23/48	31/25/32	31/23/58	34/24/48
Lattice	Inf/Inf/-	Inf/Inf/-	Inf/Inf/-	Inf/Inf/-	Inf/Inf/-	Inf/Inf/-	Inf/Inf/-
SEOA	**12**/5/**7**	**14**/6/**8**	**14**/6/**10**	**17**/8/**9**	**15**/8/**7**	**21**/9/**13**	**19**/9/**11**

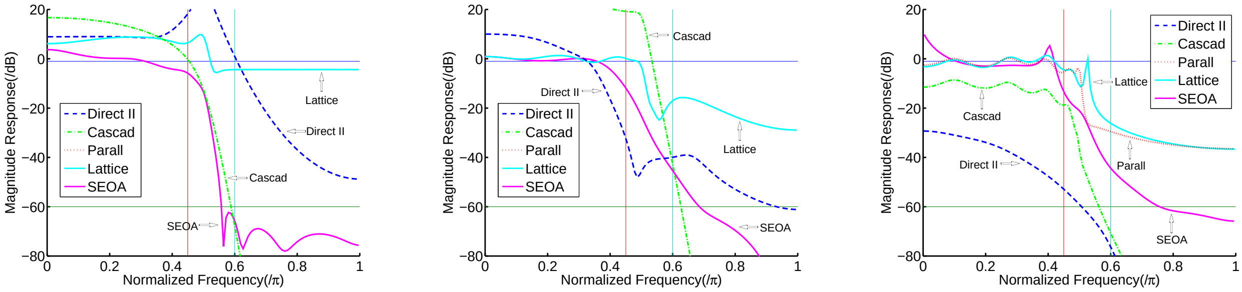

Fig. 8. The magnitude frequency response comparison with open circuits of multipliers using the Butterworth design method. (a)fault rate=1%, (b)fault rate=3%, (c)fault rate=5%

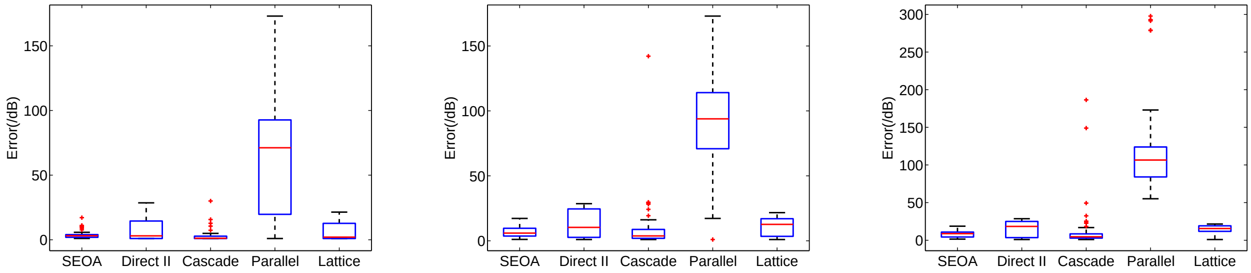

Fig. 9. The error comparison with open circuits of multipliers using the Butterworth design method. (a)fault rate=1%, (b)fault rate=3%, (c)fault rate=5%

TABLE VII
ERROR COMPARISON OF FOUR STRUCTURES USING DIFFERENT PROTOTYPE FILTERS WITH SEOA WITH OPEN CIRCUIT FAULTS (DATA IN FORMAT: MAX(DB), MEAN(DB), VAR)

Fault rate	0.01	0.02	0.03	0.04	0.05
Butterworth					
Direct II	20/8/40	20/7/37	20/8/43	20/8/44	20/8/42
Cascade	$Inf/Inf/-$	$Inf/Inf/-$	$Inf/Inf/-$	$Inf/Inf/-$	$Inf/Inf/-$
Parallel	$82/33/B$	$82/37/B$	$82/32/B$	$83/43/B$	$83/44/B$
Lattice	23/13/44	23/13/38	23/13/42	23/13/32	23/13/38
Cheby I					
Direct II	19/8/50	19/9/54	19/9/51	19/9/50	19/9/52
Cascade	$Inf/Inf/-$	$Inf/Inf/-$	$Inf/Inf/-$	$Inf/Inf/-$	$Inf/Inf/-$
Parallel	21/14/37	21/14/38	21/15/29	23/17/12	23/17/15
Lattice	14/7/24	14/7/23	14/7/26	**14**/7/23	**14/7**/27
Cheby II					
Direct II	25/11/89	25/12/99	25/12/96	25/11/91	25/11/99
Cascade	$Inf/Inf/-$	$Inf/Inf/-$	$Inf/Inf/-$	$Inf/Inf/-$	$Inf/Inf/-$
Parallel	$62/28/B$	$62/32/B$	$62/30/B$	$62/32/B$	$62/36/B$
Lattice	22/12/46	22/13/43	22/12/45	22/13/42	22/13/39
Elliptic					
Direct II	15/9/13	15/9/14	15/9/12	15/9/14	15/8/15
Cascade	$Inf/Inf/-$	$Inf/Inf/-$	$Inf/Inf/-$	$Inf/Inf/-$	$Inf/Inf/-$
Parallel	24/12/47	24/12/45	24/13/55	27/17/30	33/17/29
Lattice	19/9/42	19/9/41	19/10/41	19/11/36	19/9/39
SEOA					
SEOA	**12/4/8**	**12/5/7**	**12/6/10**	15/**7/8**	17/8/**11**

The error comparison of the classic implementations and SEOA is shown in table VIII. The mixed faults cause the performance deterioration of all the designed filters. The cascade and parallel structures are almost totally destroyed. The lattice structure is also ruined when it's designed by using Ellip prototype filter at the mixed fault rate of 0.10. SEOA reaches the optimums of 12 indexes in all 15 indexes in the mixed fault tests. The results show that filter structures

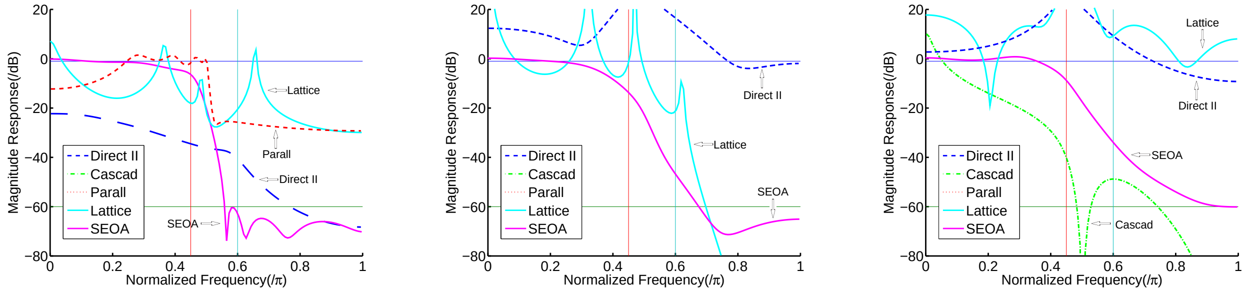

Fig. 10. The magnitude frequency response comparison with mixed circuit faults of multipliers using the Butterworth design method. (a)fault rate=2%, (b)fault rate=6%, (c)fault rate=10%

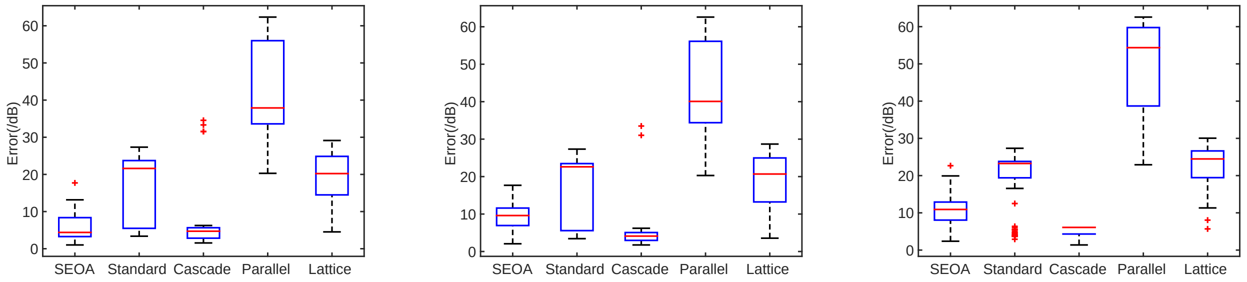

Fig. 11. The error comparison with mixed circuit faults of multipliers using the Butterworth design method. (a)fault rate=2%, (b)fault rate=6%, (c)fault rate=10%

TABLE VIII
ERROR COMPARISON OF FOUR STRUCTURES USING DIFFERENT PROTOTYPE FILTERS WITH SEOA WITH MIXED CIRCUIT FAULTS (DATA IN FORMAT: MAX(DB), MEAN(DB), VAR)

Fault rate	0.02	0.04	0.06	0.08	0.10
Butterworth					
Direct II	29/17/87	28/15/90	28/19/48	29/20/31	28/21/24
Cascade	$Inf/Inf/-$	$Inf/Inf/-$	$Inf/Inf/-$	$Inf/Inf/-$	$Inf/Inf/-$
Parallel	$B/92/B$	$B/95/B$	$B/B/B$	$Inf/Inf/-$	$Inf/Inf/-$
Lattice	28/16/83	30/22/42	31/25/27	32/25/24	32/26/20
Cheby I					
Direct II	26/10/20	24/19/**10**	25/19/14	24/20/**11**	26/20/**10**
Cascade	$Inf/Inf/-$	$Inf/Inf/-$	$Inf/Inf/-$	$Inf/Inf/-$	$Inf/Inf/-$
Parallel	27/21/41	$Inf/Inf/-$	$Inf/Inf/-$	$Inf/Inf/-$	$Inf/Inf/-$
Lattice	27/13/69	27/14/66	27/15/77	32/19/51	30/22/39
Cheby II					
Direct II	33/20/88	33/19/B	32/20/B	33/20/B	33/24/35
Cascade	$Inf/Inf/-$	$Inf/Inf/-$	$Inf/Inf/-$	$Inf/Inf/-$	$Inf/Inf/-$
Parallel	$Inf/Inf/-$	$Inf/Inf/-$	50/39/68	$Inf/Inf/-$	$Inf/Inf/-$
Lattice	28/21/35	30/20/27	28/21/26	31/24/21	31/24/19
Elliptic					
Direct II	24/17/22	23/16/22	23/17/23	**23**/16/24	23/17/22
Cascade	$Inf/Inf/-$	$Inf/Inf/-$	$Inf/Inf/-$	$Inf/Inf/-$	$Inf/Inf/-$
Parallel	$Inf/Inf/-$	27/22/36	$Inf/Inf/-$	$Inf/Inf/-$	$Inf/Inf/-$
Lattice	27/16/53	27/18/46	27/17/48	26/17/54	$Inf/Inf/-$
SEOA					
SEOA	**16/6/13**	**19/9**/14	**21/9/14**	**23/11**/15	**22/11**/14

are quite different in the aspect of fault tolerance and SEOA has an excellent property of tolerating circuit faults.

D. Result discussion

In this section, a comparative analysis between different fault modes is drawn from Table III - VIII. SEOA performs more stable than other methods when the fault rate is increasing, which makes a remarkable advantage in the maximum error, the mean error and the error variance. Short circuits and open circuits result in comparable effects on SEOA filter structures. Mixed faults cause a bigger damage to the structure at the same fault rate.

Traditional implementation methods contribute a little to tolerating these circuit faults. Lattice structures are sensitive to short circuits; cascade structures are sensitive to open circuits. Mixed faults lead to additional damages on magnitude responses.

Structure formation is an important factor for tolerating short and open circuits. Some multipliers whose failure results in structural serious defects are key elements for fault tolerance. Once they suffer damage, the filter structure is destroyed or leads to a huge bias. Such a kind of structures is vulnerable to element failures. SEOA is more robust to resist these faults because backup branches are included in the structure which is randomly generated but is optimized through the evolution technique. Fault tolerance is fulfilled by the network topology of the designed structure. The performance of fault tolerance is subject to the structure generation method of SAGA. SAGA includes four connection modes. One of them, the connection to a new active node extends the structure, for which no other branches make a backup. However, the other three connections can produce new backup branches between two linked nodes in the structure. Therefore, the probability that the four connections are adopted in SAGA, causes different fault tolerance of the designed structure. In the test, we set the same probability for the four connections, which provides 75% of the consideration for fault tolerance.

IV. Conclusion

In this paper, a structure evolution based design method of SEOA for IIR low-pass digital filters with fault tolerance is proposed. SEOA makes two contributions. The first one is that a new generating method of SAGA for IIR digital structures is proposed. SAGA is always creating valid and diversified structures. The second one is that GA is improved by integrating SAGA with code for the evolution of digital filter structures. The improvement allows the filter structures with different scales to be searched and evaluated throughout the structure space.

SEOA searches the structure space to optimize a low-pass IIR digital filter structure which can tolerate concurrent circuit faults of multipliers. Compared to the classic filter structures and design methods, SEOA can effectively improve fault tolerance of the filter.

SEOA offers a new routine to optimize the filter structure. Its applications can expand to adaptive filter design, intelligent system design and their fault tolerance.

References

[1] J. G. Proakis and D. G. Manolakis, Digital Signal Processing - Principles, Algorithms, and Applications, 4th ed. Prentice Hall, 2007.

[2] Z. Milivojevic, Digital filter design, 1st ed. mikroElektronika, 2009.

[3] V. B. Prasad, "Fault tolerant digital systems," *Potentials, IEEE*, vol. 8, no. 1, pp. 17-21, 1989.

[4] G. Zhen, P. Reviriego, Z. Ming, W. Jing, and J. A. Maestro, "Efficient single event upset-tolerant FIR filter design based on residue number for obp satellite communication systems," *China Communications*, vol. 10, no. 8, pp. 55-67, 2013.

[5] G. R. Redinbo, "Finite field fault-tolerant digital filtering architectures," *IEEE Transactions on Computers*, vol. 36, no. 10, pp. 1236-1242, 1987.

[6] B. Shim and N. R. Shanbhag, "Energy-efficient soft error-tolerant digital signal processing," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 14, no. 4, pp. 336-348, 2006.

[7] P. Reviriego, O. Ruano, and J. A. Maestro, "Implementing concurrent error detection in infinite-impulse-response filters," *IEEE Transactions on Circuits and Systems-II: Express Briefs*, vol. 59, no. 9, pp. 583-586, 2012.

[8] Z. Gao, P. Reviriego, Z. Xu, X. Su, J. Wang, and J. A. Maestro, "Efficient coding schemes for fault-tolerant parallel filters," *IEEE Transactions on Circuits and Systems-II: Express Briefs*, vol. 62, no. 7, pp. 666-670, 2015.

[9] V. Manoj and E. Elias, "Design of multiplierless nonuniform filter bank transmulti-plexure using genetic algorithm," *Signal Processing*, vol. 89, no. 11, pp. 2274-2285, 2009.

[10] S. Chen, R. Istepanian, and B. Luk, "Digital IIR filter design using adaptive simulated annealing," *Digital Signal Processing*, vol. 11, no. 2, pp. 241-251, 2001.

[11] N. Karaboga, "Digital IIR filter design using differential evolution algorithm," *EURASIP J. Appl. Signal Process*, vol. 2005, no. 8, pp. 1269-1276, 2005.

[12] C. R. Singh and S. K. Arya, "An optimal design of IIR digital filter using particle swarm optimization," *Applied Artificial Intelligence*, vol. 27, no. 6, pp. 429-440, July 2013.

[13] K. Boudjelaba, F. Ros, and D. Chikouche, "Potential of particle swarm optimization and genetic algorithms for FIR filter design," *Circuits, Systems, and Signal Processing*, vol. 33, no. 10, pp. 3195-3222, October 2014.

[14] S. D Dao, K. Abhary, R. Marian, "Maximising Performance of Genetic Algorithm Solver in Matlab," *Engineering Letters*, vol. 24, no. 1, pp. 75-83, 2016.

[15] X. Zhang, C. Wu "Energy cost minimization of a compressor station by modified genetic algorithms," *Engineering Letters*, vol. 23, no. 4, pp. 258-268, November 2015.

[16] H. Chen, "The matrix expression of signal flow graph and its application in system analysis software," *Chinese Journal of Electronics*, vol. 11, no. 3, pp. 361-363, 2002.

Lijia Chen is an Associate Professor of Communication Engineering in the School of Physics and Electronics, Henan University, Henan, China. In addition, he is serving as Director of Graduate Studies in Detection Technology and Automation, head of Communication Engineering, and member of the research group in the Laboratory of advanced computation methods and intelligent applications. He is also a council member of Henan communication academy. His research interests are in Evolutionary Computation, Communication Network, Digital Signal Processing and Evolutionary Design Methods on Electronic Circuits and Systems. He has published over 10 journal papers, over 10 conference papers and 3 granted patents.

Mingguo Liu is an Associate Professor of Electronic Science and Technology in the School of Physics and Electronics, Henan University, Henan, China. In addition, he is serving as Director of Graduate Studies in Circuits and systems and member of the research group in the Laboratory of advanced computation methods and intelligent applications. His research interests are in Evolutionary Computation, System Optimization, Deep Neural Networks and Evolutionary Design Methods on Electronic Circuits and Systems. He has published over 10 papers, include papers in IEEE Transactions on Evolutionary Computation, Soft Computing, and some conference proceedings.

Zhen Dai was a Bachelor of Electronic Information Engineering from Xinyang Normal University and a Master of Communication and information system from Hunan University in 2006. Now he is a lecturer at Henan University. His research interests are wireless communications, networking and filter design.

Ying Chen a Bachelor of Science in electronic and information engineering in Zhengzhou University, a Master of Science in electronic circuit and system in Zhengzhou University, is a lecturer at Henan University. Her research interest focuses on the embedded system and signal processing.