# A Diversification Strategy for CSD Coefficient FIR Filter Design using GA 

Naoki Nakajima* and Kenji Suyama*<br>* School of Engineering, Tokyo Denki University, Japan


#### Abstract

In this paper, a design method of CSD (Canonic Signed Digit) coefficient FIR (Finite Impulse Response) filters using GA (Genetic Algorithm) is described. In GA, individuals tend to have similar genetic structure as the number of iterations increases. Therefore, a new strategy using multiple populations under different constraints was proposed to keep a successive diversification. However, a detail effectiveness of strategy is not verified. In this paper, several design examples are shown to present the effectiveness of multiple population GA for the CSD coefficient FIR filter design.


I. Introduction

Digital filters are used in various fields such as communication system. FIR filters can realize a perfect linear phase characteristic and ensure an absolute stability. However, in order to realize a sharp cutoff characteristic, a high order FIR filter is required. As a result, the power consumption increases as the circuit scale increases.
The circuit scale of FIR filters is dominated by multipliers which are consisted of shifters and adders. The number of shifters corresponds to the number of nonzero digits included in the binary number. Therefore, a reduction of nonzero digits is effective for reducing the circuit scale. It is well-known that the CSD representation is effective to reduce nonzero digits [1]-[3]. The CSD representation expresses each digit of the filter coefficient by 0,1 and $\overline{1}=-1$. Then, an allocation of two adjacent nonzero digits is forbidden. In addition, a total number of available nonzero digits is limited for reducing the circuit scale.

In general, a design problem of CSD coefficient FIR filters becomes a NP-hard problem, and thus it is difficult to design an optimal filter. Several heuristic approaches are used to solve the design problem of FIR filters [4]-[7] and those have been applied for solving the sub-optimal design fast instead of a strict design method [8]-[10]. Especially, GA has a high applicability to the combinatorial optimization problem [7].

GA is consisted of three operations which are the crossover, the mutation and the selection. However, the individuals in the new generation tend to have similar genetic structures as the number of iterations increases. For avoiding this difficulty, MPGA was proposed [11]. MPGA is consisted of three populations with different available number of nonzero digits to provide the different genetic property. Although a performance improvement of CSD coefficient FIR filter design by MPGA has been already revealed, the detail effect of crossover between individuals of each population is not verified. In this paper, the effectiveness of CSD coefficient FIR filter design by MPGA is verified through several design examples.
II. Design Problem
A. Design problem of FIR Filters

A magnitude response $H(\omega)$ of linear phase FIR filters when a filter order $N$ is even number and the impulse response is even symmetric, is described as follows,

$$
\begin{equation*}
H(\omega)=\sum_{n=0}^{N / 2} a_{n} \cos (n \omega) \tag{1}
\end{equation*}
$$

where $a_{n}$ is the filter coefficient. The design problem of FIR filters in the min-max criterion can be formulated as follows,

$$
\begin{equation*}
\min _{\left\{a_{n}\right\}} \max _{\omega \in \Omega}|D(\omega)-H(\omega)| \tag{2}
\end{equation*}
$$

where $D(\omega)$ is the desired magnitude response and $\Omega$ is an approximation frequency band.

## B. Design of CSD Coefficient FIR Filters

The multiplier consists of adders and shifters as shown in Fig.1. The CSD representation is effective to reduce the number of nonzero digits. In the CSD representation, each digit of the filter coefficient is represented by 0,1 and $\overline{1}=-1$, and the CSD representation has the restriction which the allocation of two adjacent nonzero digits is forbidden. For example, the coefficient $(0.0110111)_{2}$ in binary representation shown in Fig. 1 is represented as $(0.100 \overline{1} 00 \overline{1})_{\mathrm{CSD}}$ in the CSD representation shown in Fig.2. From those figures, it can be confirmed that the number of nonzero digits can be reduced and thus the circuit scale is also reduced. In this paper, in order to reduce the circuit scale, the total number of available nonzero digits is also limited.

The filter coefficient in the CSD representation is presented as follows,

$$
\begin{equation*}
a_{n}=\sum_{k=1}^{p} x_{n, k} 2^{-k} \tag{3}
\end{equation*}
$$

where $p$ is a word length and $x_{n, k} \in\{1,0, \overline{1}\}$. The constraint on the restriction for forbiding the allocation of the two adjacent nonzero digits is formulated as follows,

$$
\begin{equation*}
\left|x_{n, k}\right|+\left|x_{n, k+1}\right| \leq 1 \tag{4}
\end{equation*}
$$

In addition, the constraint for the total number of available nonzero digits is formulated as follows,

$$
\begin{equation*}
\sum_{n=0}^{N / 2} \sum_{k=1}^{p}\left|x_{n, k}\right| \leq \Lambda \tag{5}
\end{equation*}
$$



Fig. 1. The structure of multiplier (binary number)


Fig. 2. The structure of multiplier (CSD representation)
where $\Lambda$ is the maximum number of total available nonzero digits. The design problem of CSD coefficient FIR filters is formulated as follows,

$$
\begin{array}{ll}
\min & \delta \\
\text { sub to } & \left|D\left(\omega_{i}\right)-H\left(\omega_{i}\right)\right| \leq \delta \\
& \left|x_{n, k}\right|+\left|x_{n, k+1}\right| \leq 1 \\
& \sum_{n=0}^{N / 2} \sum_{k=1}^{p}\left|x_{n, k}\right| \leq \Lambda  \tag{6}\\
& i \in\{0,1, \cdots, S\},
\end{array}
$$

where $S$ is the number of divided frequencies and $\delta$ is the maximum absolute error between $D(\omega)$ and $H(\omega)$. This problem is one of the mixed integer programming problems.

## C. Objective Function

The objective function is defined as follows,

$$
\begin{equation*}
F(\boldsymbol{x})=W \delta+s_{1} \phi_{1}(\boldsymbol{x})+s_{2} \phi_{2}(\boldsymbol{x}) \tag{7}
\end{equation*}
$$

where $\boldsymbol{x}=\left[x_{0,1}, x_{0,2}, \cdots, x_{N / 2, p}\right]^{T}, W, s_{1}$ and $s_{2}$ are weight parameters, and $\phi_{1}(\boldsymbol{x})$ and $\phi_{2}(\boldsymbol{x})$ are the penalty functions. $\phi_{1}(\boldsymbol{x})$ is the penalty function which forbids the allocation of two adjacent nonzero digits and is defined as follows,

$$
\phi_{1}(\boldsymbol{x})=\left\{\begin{array}{ll}
0, & \text { if } B_{n, k} \leq 1  \tag{8}\\
\sum_{n=0}^{N / 2} \sum_{k=1}^{p-1} B_{n, k}, & \text { otherwise }
\end{array},\right.
$$

where

$$
\begin{equation*}
B_{n, k}=\left|x_{n, k}\right|+\left|x_{n, k+1}\right| . \tag{9}
\end{equation*}
$$



Fig. 3. The individuals in GA
$\phi_{2}(x)$ is the penalty function which limits the total number of available nonzero digits and is defined as follows,

$$
\phi_{2}(\boldsymbol{x})= \begin{cases}0, & \text { if } \lambda \leq \Lambda  \tag{10}\\ \lambda-\Lambda, & \text { otherwise }\end{cases}
$$

where

$$
\begin{equation*}
\lambda=\sum_{n=0}^{N / 2} \sum_{k=1}^{p}\left|x_{n, k}\right| . \tag{11}
\end{equation*}
$$

From the viewpoint of reducing the circuit scale, it is desirable that the number of nonzero digits per one coefficient is limited to at most one. However, it is not effective for the coefficient having the high sensitivity. Therefore, the design of CSD coefficient FIR filters is required that the number of nonzero digits per one coefficient is one or two.

## D. GA (Genetic Algorithm)

GA consists of three operations: crossover, mutation and selection. The simple structure of the individual in GA is shown in Fig.3. In the crossover, two individuals are chosen from the population and each gene of those individuals is exchanged according to the crossover probability. In the mutation, the gene in the individual is changed to the allele according to the mutation probability. In the selection, the good individuals which is chosen based on the value of objective function is selected from the population. Then, a new generation is created by the individuals selected. As a result, the crossover and the mutation prompt the diversification, and the selection prompts the intensification. For the good design, it is desired that a balance between the diversification and the intensification is always kept among the search process.

However, the individuals in the new population tend to have similar gene structure as the number of iterations increases. The mutation and the selection can essentially keep the diversification and the intensification, respectively. On the other hand, because the crossover depends on genetic structure, it requires the existence of the different gene structures among individuals for the successive search. An example of gene structures when several individuals indicate the similar structures is shown in Fig.4. In Fig.4, individual 1 and 4 or 2 and 3 have the same


Fig. 4. Crossover (similar structure of individuals)
genetic structures. Therefore, the possibility that the another individuals are newly generated is extremely low, and thus the diversification may be prevented.

## E. MPGA (Multiple Population GA)

MPGA has three populations having different number of available nonzero digits; $\Lambda, \Lambda-1$ and $\Lambda+1$. The initial values of each population are given by simply rounded continuous coefficient [12] to the nearest CSD numbers. The three populations are defined by different penalty functions. The crossover between $\Lambda$ and $\Lambda-1$ works to prompt for removing nonzero digits. And the crossover between $\Lambda$ and $\Lambda+1$ works to prompt for adding nonzero digits. As a result, MPGA posseses the individuals with different genetic structures until the end of the search. Therefore, the preservation of diversification of search can be expected.

A procedure of MPGA consists of three operations like a standard GA: the crossover, the mutation, the selection. In the crossover, $\boldsymbol{x}_{\text {best }}$ and $\boldsymbol{x}_{r a n d}$ are chosen from three populations, where $x_{\text {best }}$ is the best solution among all populations and $\boldsymbol{x}_{\text {rand }}$ are randomly selected individual from all populations. The gene between two individuals are changed according to the crossover possibility. Then, the uniform crossover is applied as the crossover law. In the mutation, the gene of the individual is randomly selected according to the mutation rate and those are exchanged to the allele according to the mutation possibility. The CSD representation has three gene: $x_{n, k} \in\{0,1, \overline{1}\}$. Therefore, the probability of exchanging to the allele is set to $50 \%$. In the selection, the individuals of the new generation are selected from all individuals. Then, the ranking selection is applied as the selection law.

TABLE I
Design conditions

|  | Ex.1 | Ex. 2 | Ex.3 | Ex.4 |
| :---: | :---: | :---: | :---: | :---: |
| $N$ | 100 | 150 | 200 | 300 |
| $p$ | 16 | 16 | 16 | 16 |
| $f_{p}$ | 0.21 | 0.2 | 0.15 | 0.21 |
| $f_{s}$ | 0.23 | 0.215 | 0.16 | 0.217 |
| $S$ | 500 | 750 | 1000 | 1500 |
| $\Lambda$ | 100 | 150 | 200 | 300 |



Fig. 5. Verification result of standard GA (Ex.1)

## III. VERIFICATION of SEARCH PROCESS DESIGN EXAMPLES

Several design examples are shown to present the effectiveness of the proposed method. The desired magnitude response $D(\omega)$ was defined as follows,

$$
D(\omega)= \begin{cases}1, & 0 \leq \omega \leq 2 \pi f_{p}  \tag{12}\\ 0, & 2 \pi f_{s} \leq \omega \leq \pi\end{cases}
$$

where $f_{p}$ is the normalized passband edge frequency, $f_{s}$ is the normalized stopband edge frequency.

The design conditions are listed in Table I. The number of individuals per a population was set to 60 , the number of generations was set to 800 and the number of trials was set to 50. $W, s_{1}$ and $s_{2}$ were set to 1 . The crossover rate was set to 0.95 and the mutation rate was set to 0.3 . The updating curve of standard GA and the crossover points between $\Lambda$ and $\Lambda$ are shown from Fig. 5 to Fig.8. The updating curve of MPGA and the crossover points between $\Lambda$ and $\Lambda, \Lambda$ and $\Lambda-1$ and $\Lambda$ and $\Lambda+1$ are shown from Fig. 9 to Fig.12. The filter coefficients obtained by the standard GA and MPGA in Ex. 1 are listed in Table II.

From those verification results, it can be confirmed that MPGA has many crossover points and its average value of $F(\boldsymbol{x})$ is small. Terefore, it can be verified that MPGA has the individuals with dfferent genetic structures until the end of the search and can keep a successive diversification. From Table II, it is shown that the total number of nonzero digits is equal to $\Lambda$ and the allocation of adjacent nonzero digits does not exist.


Fig. 6. verification result of standard GA (Ex.2)


Fig. 7. verification result of standard GA (Ex.3)

## IV. CONCLUSIONS

In this paper, the search performance of MPGA for CSD coefficient FIR filter design was verified. From the several design examples, it was shown that the search by MPGA could be diversified by the crossover.

## References

[1] Yassin M. Hasan, Lina J. Karam, M. Falkinburg, A. Helwig, and M. Ronning, "Canonic Signed Digit FIR Filter Design," Proc. Thirty-Fourth Asilomar Conference on Signals, Systems and Computers, pp.1653-1656, November, 2000.
[2] R. Ito, T. Fujie, K. Suyama, and R. Hirabayashi, "A Powers-of-two Terrn Allocationalgorithrn for Designing FIR Filters with CSD Coefficients in A Min-max Sense," Proc. 12th European, Signal Processing Conference pp.987-990, September, 2004.
[3] R. Jia, F. Wang, R. Chen, Xing-Gang Wang, D. Shang, and Hai-Gang Yang, "High-OrderReconfigurable FIR Filter Design Based on Statistical Analysis of CSD Coefficients," Proc. Field-Programmable Technology, pp.402-405, December, 2013.
[4] Poulami Das, Sudip Kumar Naskar, and Sankar Narayan Patra, "Adaptive global best steered Cuckoo search algorithm for FIR filter design," Third International Conference on Research in Computational Intelligence and Communication Networks, pp.15-20, November, 2017.
[5] Adrika Mukherjee, Niloy Chakraborty, and Badhan Kumar Das, "Whale Optimization Algorithm:An Implementation to design low-pass FIR Filter," Innovations in Power and Advanced Computing Technologies, pp.15, April, 2017.


Fig. 8. verification result of standard GA (Ex.4)


Fig. 9. Verification result of MPGA (Ex.5)
[6] Rija Raju, and Hon keung Kwan, "FIR filter design using Multiobjective Artificial Bee Colony algorithm," IEEE 30 th Canadian Conference on Electical and Computer Engineering, April, pp.1-4, 2017.
[7] D. Suckley, "Genetic Algorithm in The Design of FIR Filters," Proc. of IEEE Circuits, Devices and Systems, Vol.138, No.2, pp.234-238, April, 1991.
[8] Q. Yue, C. Ma, and X. Wang, "Canonical Signed Digit Encoding Based Optimal Design for FIR Filters," Electronic and Mechanical Engineering and Information Technology, Vol.2, pp.729-732, August, 2011.
[9] T. Williams, M. Ahmadi, and W. C. Miller, "Genetic Algorithms for The Design of Digital Filters Using Canonic Signed Digit Coefficients," Proc. 7th International Conference on Signal Processing, pp.9-12, September, 2004.
[10] An-Xin Zhao, Xiao-Jun Tang, Zhong-Hua Zhang, and Jun-Hua Liu, "The Optimal Design Method of FIR Filter Using the Improved Genetic Algorithm," IEEE Conference on Industrial Electronics and Applications, pp.452-455, June 2014.
[11] Y. Arie and K. Suyama, "Evolutionary stagnation avoidance for design of CSD coefficient FIR filters using GA," Proc. 2016 International Technical Conference on Circuits/Systems, Computers and Communications, pp.823-826, July, 2016.
[12] J. McClellan, T. Parks, and L. Rabiner, "A Computer Program for Designing Optimum FIR Linear Phase Digital Filters," IEEE Transactions on Audio and Electroacoustics, Vol.21, pp.506-526, December, 1973.


Fig. 10. verification result of MPGA (Ex.6)


Fig. 11. verification result of MPGA (Ex.7)


Fig. 12. verification result of MPGA (Ex.8)

TABLE II
The filter coefficients (Ex.1)

|  | GA | multiple population GA |
| :---: | :---: | :---: |
| $a_{0}$ | $0.100 \overline{1} 000010000000$ | $0.100 \overline{1} 000010010000$ |
| $a_{1}$ | 0.1010000000000000 | 0.1010000000000000 |
| $a_{2}$ | $0.001000 \overline{1} 000000000$ | $0.001000 \overline{1} 000000000$ |
| $a_{3}$ | $0.0 \overline{1} 01001001000000$ | $0.0 \overline{1} 01001001000000$ |
| $a_{4}$ | $0.00 \overline{1} 0010001000000$ | $0.00 \overline{1} 0010001000000$ |
| $a_{5}$ | $0.0001010 \overline{1} 00000000$ | $0.0001010 \overline{1} 00000000$ |
| $a_{6}$ | $0.0010 \overline{1} 00000100000$ | $0.0010 \overline{1} 00001000000$ |
| $a_{7}$ | $0.0000 \overline{1} 01010000000$ | $0.0000 \overline{1} 01010000000$ |
| $a_{8}$ | $0.000 \overline{1} 0 \overline{1} 0001000000$ | $0.000 \overline{1} 0 \overline{1} 0001000000$ |
| $a_{9}$ | $0.000000 \overline{1} 00 \overline{1} 000000$ | $0.000000 \overline{1} 000000000$ |
| $a_{10}$ | $0.0001000 \overline{1} 00 \overline{1} 00000$ | $0.0001000 \overline{1} 0 \overline{1} 000000$ |
| $a_{11}$ | $0.0000100 \overline{1} 00 \overline{1} 00000$ | $0.0000100 \overline{1} 0 \overline{1} 000000$ |
| $a_{12}$ | $0.0000 \overline{1} 0 \overline{1} 001000000$ | $0.0000 \overline{1} 0 \overline{1} 001000000$ |
| $a_{13}$ | $0.0000 \overline{1} 00 \overline{1} 00000000$ | $0.0000 \overline{1} 00 \overline{1} 00000000$ |
| $a_{14}$ | 0.0000010100000000 | 0.0000010100000000 |
| $a_{15}$ | 0.0000100100000000 | 0.0000100101000000 |
| $a_{16}$ | $0.0000000 \overline{1} 00000000$ | $0.0000000 \overline{1} 00000000$ |
| $a_{17}$ | $0.0000 \overline{1} 000 \overline{1} 0100000$ | $0.0000 \overline{1} 0000 \overline{1} 000000$ |
| $a_{18}$ | $0.000000 \overline{1} 000000000$ | $0.000000 \overline{1} 000000000$ |
| $a_{19}$ | $0.000010 \overline{1} 010000000$ | $0.000010 \overline{1} 010000000$ |
| $a_{20}$ | 0.0000010000000000 | 0.0000010000000000 |
| $a_{21}$ | $0.00000 \overline{1} 000 \overline{1} 000000$ | $0.00000 \overline{1} 000 \overline{1} 000000$ |
| $a_{22}$ | $0.00000 \overline{1} 0 \overline{1} 00000000$ | $0.00000 \overline{1} 0 \overline{1} 00000000$ |
| $a_{23}$ | 0.0000001000000000 | 0.0000001000000000 |
| $a_{24}$ | 0.0000010100000000 | 0.0000010100000000 |
| $a_{25}$ | 0.0000000000000000 | 0.0000000000000000 |
| $a_{26}$ | $0.00000 \overline{1} 0000000000$ | $0.00000 \overline{1} 00 \overline{1} 0000000$ |
| $a_{27}$ | $0.000000 \overline{1} 010000000$ | $0.000000 \overline{1} 010000000$ |
| $a_{28}$ | $0.00000100 \overline{1} 0001000$ | $0.00000100 \overline{1} 00 \overline{1} 0000$ |
| $a_{29}$ | 0.0000001010000000 | 0.0000001010000000 |
| $a_{30}$ | $0.000000 \overline{1} 000000000$ | $0.000000 \overline{1} 000000000$ |
| $a_{31}$ | $0.00000 \overline{1} 0100000000$ | $0.00000 \overline{1} 0101000000$ |
| $a_{32}$ | $0.0000000010 \overline{1} 00000$ | 0.0000000010000000 |
| $a_{33}$ | $0.0000010 \overline{1} 00000000$ | $0.0000010 \overline{1} 00000000$ |
| $a_{34}$ | 0.0000000010000000 | 0.0000000010000000 |
| $a_{35}$ | $0.000000 \overline{1} 0 \overline{1} 0000000$ | $0.000000 \overline{1} 0 \overline{1} 0000000$ |
| $a_{36}$ | $0.0000000 \overline{1} 00000000$ | $0.0000000 \overline{1} 00000000$ |
| $a_{37}$ | $0.000000100 \overline{1} 000000$ | $0.000000100 \overline{1} 000000$ |
| $a_{38}$ | $0.00000010 \overline{1} 0000000$ | $0.00000010 \overline{1} 0000000$ |
| $a_{39}$ | $0.0000000 \overline{1} 00000000$ | $0.0000000 \overline{1} 01000000$ |
| $a_{40}$ | $0.000000 \overline{1} 001000000$ | $0.000000 \overline{1} 001000000$ |
| $a_{41}$ | 0.0000000001000000 | 0.0000000001000000 |
| $a_{42}$ | $0.00000010 \overline{1} 0000000$ | $0.00000010 \overline{1} 0000000$ |
| $a_{43}$ | 0.0000000010000000 | 0.0000000010000000 |
| $a_{44}$ | $0.0000000 \overline{1} 0 \overline{1} 000000$ | $0.0000000 \overline{1} 0 \overline{1} 000000$ |
| $a_{45}$ | $0.0000000 \overline{1} 01000000$ | $0.0000000 \overline{1} 01000000$ |
| $a_{46}$ | $0.000000010 \overline{1} 000000$ | 0.0000000100000000 |
| $a_{47}$ | 0.0000000100000000 | 0.0000000100000000 |
| $a_{48}$ | $0.00000000 \overline{1} 0100000$ | $0.00000000 \overline{1} 0000000$ |
| $a_{49}$ | $0.000000 \overline{1} 000000000$ | $0.000000 \overline{1} 000000000$ |
| $a_{50}$ | 0.0000000001000000 | 0.0000000000001000 |

