A GA-BASED REALIZATION METHOD OF OPTIMAL FINITE-PRECISION SYSTEM

Miki Haseyama and Daiki Matsuura

School of Engineering, Hokkaido University,
N-13 W-8 Kita-ku, Sapporo 060-8628, Japan
E-mail: mikich@media.eng.hokudai.ac.jp

ABSTRACT

A GA-based realization method of the optimal finite-precision system is proposed. The optimal realizations of the finite-precision systems are defined as those that minimize the error between the frequency characteristics of the original infinite-precision system and its finite-precision represented one and can be shown as the solutions of a nonlinear programming problem. Therefore, in this paper, GA-based optimization strategy is presented to provide an efficient method for solving this problem. The proposed realization method of the optimal finite-precision system is based on not only the GA but also an SA to prevent the GA from going into local minima. Some numerical examples and comparison simulations with the traditional quantization methods, such as rounding off, rounding up, and rounding down, and another SA-based method are given to verify the high performance of the proposed method.

1. INTRODUCTION

The recent advantages in digital system design methods have led to a need for the efficient and accurate hardware implementation. Although the number of system implementations using floating-point processor has been increasing; for reasons of cost, simplicity, speed, and memory space, the use of fixed-point processor is recently more desirable for many industrial and consumer applications. However, even if using floating-point processor, it is well-known that when the infinite-precision system is implemented as a finite-precision system, its frequency characteristic must worsen and may become unstable due to finite-word-length effects.

Therefore, this paper proposes a new realization method of the finite-precision systems to prevent the finite-word-length effects. The optimal realizations of the finite-precision system are defined as those that minimize the error between the frequency characteristic of the original infinite-precision system and its finite-precision represented one and can be shown as the solutions of a nonlinear programming problem. Therefore, we adopt a genetic algorithm (GA)[1] based optimization strategy to provide an efficient method for solving this problem. Also the GA used in the proposed realization method includes a simulated annealing (SA)[2] to prevent the GA from going into local minima. Consequently, it effectively searches for the optimal finite-precision IIR system, which retains the frequency characteristic of the original infinite-precision one, from a population of the finite-word-length IIR systems.

Some numerical examples and comparison simulations with not only the traditional quantization methods, but also another SA-based method are given to verify the high performance of the proposed method.

2. THE GA-BASED FINITE-PRECISION SYSTEM REALIZATION

In the GA for the realization of the optimal finite-precision system, a chromosome represents a finite-word-length IIR system and each gene represents a discrete-valued coefficient of the system. With these representations, the GA searches for a chromosome corresponding to the best finite-precision system from the population of all of the possible finite-precision systems, where the best finite-precision system is the one which can retain the original frequency characteristic most accurately. In addition, an SA is embedded into the GA in order to avoid being caught by local minima.

The detail implementation of the proposed method is described in the following:

(i) Initial population

Suppose the population has $N_{population}$ individuals. The coefficients of the original system or the system obtained by a model identification algorithm such as [7] from an observed signal are rounded off, up, and down. From these three finite-precision systems, the first three chromosomes are generated respectively by chaining together the coefficients to form an individual which contains $(p+q)*L$ genes, where (p, q) is the filter order and L is the word length of the binary representation of a coefficient, as shown in Fig.

This work is partially supported by a grant from the Japan Society for the Promotion of Science (Grant-in-Aid for Scientific Research (C) 14550343.

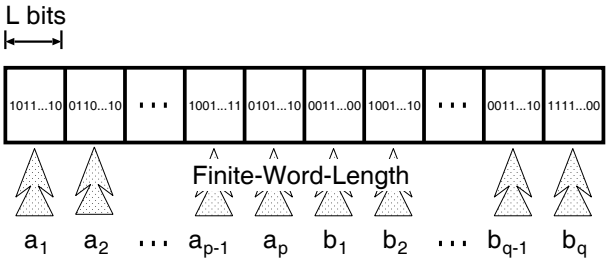

Fig. 1. An Example Chromosome representing a target IIR system, whose transfer function is $H(z^{-1}) = \frac{1+a_1z^{-1}+\cdots+a_Nz^{-N}}{1+b_1z^{-1}+\cdots+b_Mz^{-M}}$, with the finite-word-length L bit.

1. The remaining $(N_{population}-3)$ chromosomes are then generated from these three based on probabilities of binomial distributions. First, one individual is randomly selected from the first three. Next, its genes are changed according to the following binomial probability $P(i)$; where $P(i)$ represents the probability that the gene corresponding to I_{i_c} is changed to another value I_i, and I_j ($j \in \{1,2,\ldots,L | I_1 < \cdots < I_L\}$) is a possible finite-word-length value:

$$P(i) = \begin{cases} P_R(i) & (i_R \geq i > i_c) \\ P_L(i) & (i_L \leq i < i_c) \\ 0 & otherwise, \end{cases}$$

where if $N_R \stackrel{\triangle}{=} i_R - i_c (>0)$ and $N_L \stackrel{\triangle}{=} i_c - i_L (>0)$, then

$$P_R(i) = \frac{(2N_R-1)!}{(N_R-i+i_c)!(N_R+i-i_c-1)!}\left(\frac{1}{2}\right)^{2N_R-1},$$

$$P_L(i) = \frac{(2N_L-1)!}{(N_L-i_c+i)!(N_L+i_c-i-1)!}\left(\frac{1}{2}\right)^{2N_L-1}.$$

The above probability functions are well-known as binomial distributions. The parameters i_R and i_L are integers minimizing $|I_{i_R} - I_{i_c} - L_{range}|$ and $|I_{i_c} - I_{i_L} - L_{range}|$, respectively, where L_{range} is a predefined parameter.

(ii) The fitness

Suppose the transfer function of a finite-precision system corresponding to a chromosome f is $H(\omega)$, its fitness is defined as follows:

a) If the finite-precision system represented by f is stable,

$$fitness(f) \stackrel{\triangle}{=} \frac{1}{E(f)} \tag{1}$$

where $E(f)$ denotes the difference between $H(\omega)$ and the transfer function of the original filter $H_{org}(\omega)$. For example,

$$E(f) \stackrel{\triangle}{=} \int_0^\pi W(\omega)\left||H_{org}(\omega)| - |H(\omega)|\right| d\omega \tag{2}$$

where $W(\omega)$ is a frequency-weighting function. The integral is practically computed by the summation with discrete ω. Or

b) If the finite-precision system represented by f is unstable,

$$fitness(f) = \frac{1}{m}\left(\sum_{k=N-m}^{N-1}|h(k)|\right)^{-1} \tag{3}$$

where $h(k)$ is the impulse response of the finite-precision system f; and m is the number of the data samples used for deciding whether the impulse response is converged.

The stability of the finite-precision system represented by each individual is judged as follows: if $\sum_{k=N-m}^{N-1}|h(k)| < C_{Th}$, then it is stable; otherwise it is unstable, where C_{Th} is usually set at $m|h(0)|$. While the GA iteration is proceeding, the unstable individuals gradually expire, and will not be finally selected as the optimization result. Since some unstable chromosomes possibly have good genes, our method does not make them lethal. The frequency-weighting function $W(\omega)$ in Eq. (2) can control the accuracy of the finite-precision system in an arbitrary frequency region[6][7].

(iii) Reproduction (Selection)

Reproduction is performed by the rank selection, which ranks the individuals in descending order according to their fitness. Furthermore, in our method, all of the individuals representing stable filters are ranked ahead of all the unstable individuals. Then k-th ranked individual receives the following selection probability:

$$P(k) = \frac{2(N_{population}-k+1)}{N_{population}(N_{population}+1)}. \tag{4}$$

DeJong's elitist model[8] is also utilized for the elite to survive.

(iv) Crossover

The uniform crossover[1] is used.

(v) Mutation

Mutation is performed on genes selected according to a probability of mutation. A selected gene is replaced by a value generated by the same manner as in (i), where both I_{i_u} and I_{i_d} are substituted with the original value of the selected gene, respectively.

(vi) Including a simple SA

The following simple SA is applied to all the chromosomes f's in an interval of $N_{interval}$ GA generations: Firstly, a gene I_{i_t} ($i_t \in \{1,\ldots,L\}$) of f is randomly selected and randomly changed to either I_{i_t+1} or I_{i_t-1}. Then, the chromosome with the new gene is denoted by f'. If f' is unsta-

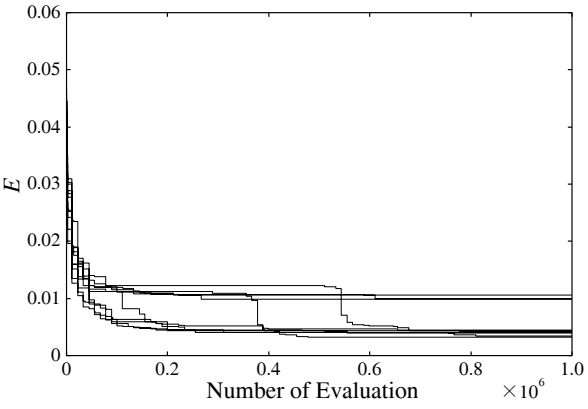

Fig. 2. Fitness versus generation. $fitness = 1/E(f)$. Ten trials with different initial populations generated randomly with different seeds. The fitness of the elite in each trial is plotted.

ble, the above procedures are repeated until either a stable f' is obtained, or the number of the repeat reaches the limit N_{repeat}. Secondly, f is replaced with f' based on the following acceptance probability:

$$P = \begin{cases} 1 & (\Delta E < 0) \\ e^{-\Delta E/T_i} & (\Delta E \geq 0) \end{cases} \quad (5)$$

where $\Delta E = E(f) - E(f')$, and T_i is a control parameter, which is generally called temperature. The above replacement is iterated starting with initial temperature T_0, and the temperature is decreased as $T_{i+1} = \alpha T_i$, and the iteration is terminated when $T_i = T_{end}$. Though the choice of good starting points is important for good performance in general SAs, our method is less sensitive to the initial starting points since the initial state, that is f before the iteration, is provided by the GA search.

(vii) The condition to terminate the GA iterations

When the number of the generation attains to $N_{termination}$, the GA iterations are terminated.

3. EXPERIMENTAL RESULTS

3.1. Performance Verification

The proposed method is applied to a 10th degree Butterworth filter, where the quantized coefficients are represented by sign 1 bit, mantissa 2 (n_f) bits, and exponent 3 (n_e) bits as follows:

$$(-1)^{g_s} \times \left(1 + \sum_{k=1}^{n_f} 2^{-k} c_k\right) \times 2^{e_u},$$

where $g_s, c_k \in \{1, 0\}$, and $e_u \in \{-7, -6 \ldots, 0\}$. The fitness function $E(f)$ in Eq. (2) is used. The frequency-weighting function is $W(\omega) = 1$ if $\omega < \frac{\pi}{2}$, otherwise $W(\omega) = 0.5$. It helps the proposed method to realize the finite-precision system more accurately in the low frequency band than in the other frequency bands. The parameters for the GA in the proposed method are: $N_{population}$ = 60; $L_{range} = 0.25$, which is the longest distance between neighboring quantized points; $m = 16$ and $N = 1000$ in Eq. (3); $C_{Th} = mh(0)$; the probability of mutation is 6 %; and the number of generations $N_{termination}$ in the termination condition is 10^6. The parameters for the embedded SA are: $T_0 = 10^{-3}$, $T_{end} = 10^{-6}$, $N_{repeat} = 42$, $N_{interval} = 50$, and $\alpha = 0.95$.

First, we demonstrate the behavior of the fitness of the elitist individual in Fig. 2. $E(f)$ of 10 trials (out of total 30) with different initial populations randomly generated by a random number generator with different seeds are plotted against generation. The smaller the $E(f)$, the better the fitness as defined in Eq. (1). They are all decreased with the growth of the generations. Among all 30 trials, the best, worst, and average quality of filter are -53.0 [dB], -38.1 [dB], and -46.7 [dB], respectively. These are all better than -23 dB which is given by the rounded filter. The frequency characteristics of the finite-precision systems obtained by rounding down, up, and off, and the proposed method are shown in 3. This figure also indicates that the proposed method can provide enough quality of the finite-precision system.

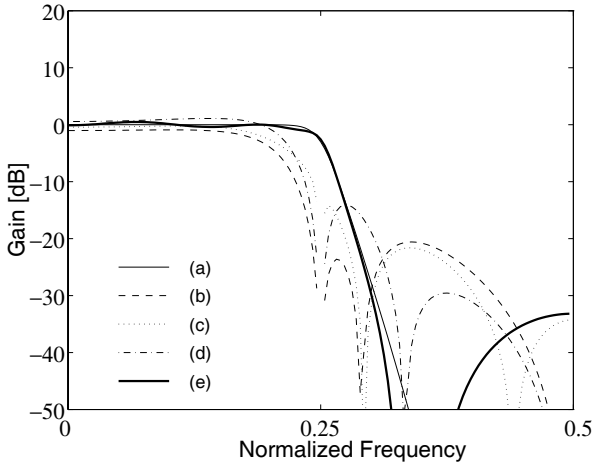

Fig. 3. The frequency characteristics of the finite-precision systems. (a) shows the reference frequency characteristic; (b) is the filter of which the coefficients are rounded down; (c) rounded up; (d) rounded off; and (e) the proposed method.

3.2. Comparison of convergence performance with another SA-based method

Next, we show the convergence performance of the proposed method by comparing it with the SA-based method in [5]. For better comparisons, we use the same criterion that is used by [5] to replace $E(f)$ in Eq. (2) as follows:

$$E(f) = \max_{\omega} \left\{ \left| W(\omega) \left(G^{-1} H(\omega) - H_{org}(\omega) \right) \right| \right\} \quad (6)$$

where G is the filter gain. Unlike in [5], the continuous value G needs to be searched for, in our algorithm, it can be computed by $H(\omega)$ represented by the individual f as follows:

$$G(f) = \begin{cases} \frac{p_{max}+p_{min}}{2}, & \frac{p_{max}-p_{min}}{2} \leq s_{max} \\ p_{min} + s_{max}, & \text{otherwise} \end{cases}$$

where p_{max}, p_{min}, and s_{max} are the maximum, minimum values in its passband, and the maximum value in its stopband, respectively.

The target system is a linear phase FIR system of order N (odd) of which the frequency characteristic is with normalized passband $0 \sim 0.3$ and stopband $0.5 \sim 1$. $W(\omega) = 1$ for these bands, and otherwise $W(\omega) = 0$ for the others. The finite-word-length coefficients are represented by

$$\sum_{k=1}^{2} d_k 2^{-g_k}, \qquad d_k \in \{-1, 0, -1\} \text{ and } g_k \in \{1, \ldots, B\}$$

where $B = 10$ in this experiment. Other parameters in the proposed method are the same as those for the experiments in 3.1.

Table 1 shows the computational costs before the convergence which are reported as the total number of function evaluations. Where the quality of the obtained finite-precision system by the proposed method is equal or better than [5]. Clearly, the proposed method can provide a significant saving in computational cost.

In these simulations several restrictions have to be included for the execution of [5]. However, the proposed method can be applied to not only this kind of system but also any other ones.

Table 1. Convergence performances of the proposed method and [5].

	Proposed Method	SA-based Method in [5]
Filter Length N	Number of Evaluation^{*1} ($\times 10^5$)	Number of Evaluation^{*2} ($\times 10^5$)
27	0.34	8.49
29	0.67	9.39
31	1.06	10.6
33	2.39	10.26
35	2.79	11.05

$*1$: The number of evaluation includes the GA and SA evaluations both before the convergence. It is the average of 50 trials.

$*2$: The number of evaluation in the SA before the convergence.

4. CONCLUSION

The GA-based realization method of the optimal finite-precision system has been proposed. By using the proposed method, we can obtain the optimal quantized filter, which retains the reference frequency characteristic most accurately.

Some numerical examples have been given to verify its higher performance than the traditional quantization methods', such as rounding off, rounding up, and rounding down. And also the comparison simulations with the SA-based method have indicated that the proposed method can reaches the desired accuracy faster than SA-based method does.

5. REFERENCES

[1] David E. Goldberg, *Genetic algorithms in search, optimization, and machine learning.* Addison Wesley Publishing Company, 1989.

[2] R. H. J. M. Otten and L. P. P. P. van Ginneken, *The Annealing Algorithm.* Boston: Kluwer, 1989.

[3] Gabor C. Temes and Sanjit K. Mitra, *Modern filter theory and design.* John Wiley & Sons, 1973.

[4] Ioannis Pitas, "Optimization and adaptation of discrete-valued digital filter parameters by simulated annealing," *IEEE Trans. Signal Processing*, vol 42, no. 4, pp.860-866, April 1994.

[5] Nevio Benvenuto, Michele Marchesi and Aurelio Uncini, "Application of Simulated Annealing for the design of special digital filters," *IEEE Trans. Signal Processing*, vol 40, no. 2, pp.323-332, Feb. 1992.

[6] Lennart Ljung, *System identification theory for the user.* Englewood Cliffs, NJ: Prentice Hall, 1987.

[7] Miki Haseyama, Nobuo Nagai, and Nobuhiro Miki, "An adaptive ARMA four-line lattice filter for spectral estimation with frequency weighting," *IEEE Trans. Signal Processing*, vol. 41, no. 6, pp. 2193-2207, June 1993.

[8] K. DeJong, *An analysis of the behavior of a class of genetic adaptive systems.* Ph. D Thesis, University of Michigan, 1975.