Nonlinear Parameter Estimation via the Genetic Algorithm

Leehter Yao, *Member, IEEE,* and William A. Sethares

Abstract—**A modified genetic algorithm is used to solve the parameter identification problem for linear and nonlinear digital filters. Under suitable hypotheses, the estimation error is shown to converge in probability to zero. The scheme is also applied to feedforward and recurrent neural networks.**

I. Introduction

A VARIANT of the genetic algorithm (GA) [3] is used to solve the parameter estimation problems for linear and nonlinear digital filters and is applied to both feedforward and recurrent neural networks. Unlike steepest descent approaches (e.g., [1], [2]) to nonlinear parameter identification and filter design, the GA requires no calculation of the gradient and is not susceptible to local minimum problems that arise with multimodal error surfaces. The GA is a type of structured "random" search that mimics the process of biological evolution. The algorithm begins with a collection of parameter estimates (called a *chromosome*) and each is evaluated for its *fitness* in solving the given minimization task. At each *generation* (algorithm timestep), the most fit chromosomes are allowed to *mate* and bear offspring. These children (new parameter estimates) then form the basis for the next generation. The biological analogy suggests that such a procedure will be likely to lead to workable solutions for complex nonlinear problems, and we prove the asymptotic convergence of the estimation error under suitable hypotheses. The cost and system functions can be nonlinear and even discontinuous.

The GA was first introduced by Holland [3], and was extensively explored by Goldberg [4]. In [5], De Jong summarized several potential research directions, and the impact of tuning factors which affect the performance of the GA are studied in [6]. The GA is compared with another type of random search method (simulated annealing) in [7]. The GA has been successfully applied to a variety of optimization problems; for instance, it was applied to the delay estimation of sampled signals [8], to the parameter estimation of linear adaptive filters [9], and to the robotic trajectory planning problem [10]. The GA has also been applied to machine learning [11]–[13]. Because the GA tends to find the global optimum solution without becoming trapped at local minima, it is also a good tool to solve combinatorial problems. For instance, the GA has been applied to the linear transportation problem in [14] and to the floorplan design for VLSI design in [15]. The application in this paper is closest to the work of Etter *et al.* [9], where the GA was applied to the parameter estimation for linear autoregressive (AR) models. However, the particular form of the algorithm they implemented could not be guaranteed to yield an optimum estimation error and, in fact, often performed rather poorly. In this section, the GA has been modified to overcome this difficulty even when applied to the more complex nonlinear IIR setting.

Three modifications to the standard GA in [3] and [4] are proposed in this paper. First of all, parents are selected by rank, i.e., chromosomes that correspond to the least estimation errors are automatically chosen to mate. Passing the best chormosome unchanged into the next generation assures that the minimum estimation error is a monotonically decreasing process. This contrasts with the conventional approach in which parents are chosen based on a probability distribution defined by fitness values. Second, an operator named *extinction and immigration* is introduced. Extinction and immigration is applied when all chromosomes in the gene pool are identical or when there is no decrease in the estimation error over a certain number of generations. This plays a role analogous to mutation, yet has more dramatic effect. Finally, a small trick called reencoding of zeros is included to reduce the time wasted when the estimated parameters linger around zero.

Holland showed the power of the GA by relating it to the multiarmed bandit problem [3] and by guaranteeing that the minimal expected loss can be achieved. This falls short, however, of an actual proof of convergence. In this paper, we show that the estimation error converges in probability to zero.

People have applied the GA to the training of feedforward neural networks [16]–[17], and some comparisons between the GA and the backpropagation algorithm in [18] have been made empirically. The backpropagation algorithm has been extended to train recurrent neural networks as in [19] and [20]. In this paper, the modified GA will also be applied to train both feedforward and recurrent (feedback) neural network structures. Perhaps the most striking difference between the two approaches in that backpropagation is susceptible to problems with local minima, whereas the GA can be guaranteed to approach the global minimum under suitable circumstances.

This paper is constructed as follows. Section II explains how to apply the GA to the parameter estimation of IIR filters, and our modifications are discussed in detail. Section III contains proofs of the convergence for both Holland's and

Manuscript received February 9, 1991; revised February 24, 1993. The associate editor coordinating the review of this paper and approving it for publication was Prof. S. Y. Kung.

L. Yao is with the Department of Electrical Engineering, National Taipei Institute of Technology, Taipei, Taiwan, Republic of China.

W. A. Sethares is with the Department of Computer Engineering, University of Wisconsin, Madison 53706-1691.

IEEE Log Number 9215294.

our modified GA. Section IV gives several examples of the application of genetic algorithms to parameter estimation of linear and nonlinear, FIR and IIR filters and feedforward, and recurrent neural networks. Finally, concluding remarks are made in Section V.

II. Nonlinear IIR Filters and the GA

A. Problem Statement

Consider the general nonlinear IIR filter

$$y_k = f(W, Y, U) \tag{2.1}$$

where f is a nonlinear function such that (2.1) is stable in the sense of Lyapunov and where f is continuous from the right (or the left) at every point. Let $W = [w_1, \cdots, w_h]$ be the set of h fixed parameters, $Y = [y(k-1), \cdots, y(k-n)]$ be the set of n autoregressive terms, and $U = [u(k-1), \cdots, u(k-m)]$ be the set of m past input values.

The structure of f is assumed to be known to contain h terms with exactly one parameter associated with each term. Note that these terms need not enter linearly in f.

The IIR filter (2.1) is estimated by

$$\hat{y}(k) = f(\hat{W}, \hat{Y}, U) \tag{2.2}$$

where $\hat{W} = [\widehat{w_1}, \cdots, \widehat{w_h}]$ is a set of h parameter estimates, $\hat{Y} = [\hat{y}(k-1), \cdots, \hat{y}(k-n)]$ is a set of n estimated outputs, and U is defined as above.

To apply the GA, each estimated parameter $\hat{w}_i$ is encoded as a string of binary numbers called a *gene*. Genes are cascaded to form a longer string $\hat{W}$ called a chromosome. Each possible combination of estimated parameters is thus represented by a chromosome. The identification strategy is to apply the GA to search for the best chromosome $\hat{W}$ so that $\hat{y}(k) \to y(k)$. A collection of N chromosomes of estimated parameters (called the *gene pool*) are explored in each generation. Let the error associated with the jth chromosome at the ith generation be defined as

$$e_{ji} = \frac{1}{d}\sum_{k=1}^{d}(y(k) - \hat{y}_{ji}(k))^2 \tag{2.3}$$

where d is the window size over which the errors will be accumulated, and $\widehat{y_{ji}}(\cdot)$ is the estimated output associated with the jth chromosome of estimated parameters for the ith generation. At each generation i, the GA searches for the minimum estimation error

$$e^i_{\min} \equiv \min(e_{ji}) \quad \forall j \in [1 \cdots N] \tag{2.4}$$

over the entire space of parameters and attempts to drive $e^i_{\min}$ to zero over succeeding generation.

B. Features of the GA

This subsection describes the various elements of the GA, including the encoding mechanism, a method of initialization, the parent selection policy, details of the mating procedure, the introduction of mutation as a way to avoid local minima, our new policy of extinction and immigration, and the reencoding of zeros. Once these features have been described, the algorithm can be stated clearly and succinctly.

1) Encoding: The estimated parameters are encoded into genes and chromosomes as a string of binary digits using one's complement representation. Assuming that the parameters lie in some bounded region

$$|w_k| \le \eta_k \quad \text{for} \quad k = 1 \cdots h, \tag{2.5}$$

the length of the genes (and hence the chromosomes) can be calculated as the length of the binary string B_k needed to encode w_k based on η_k and the desired accuracy δ_k. In [11], the value of the individual weight is taken as a gene, and thus, the chromosome is encoded by a string of real numbers instead of binary numbers. This encoding scheme saves memory but appears to hinder crossover from playing its full role in the GA [21].

2) Initialization: The initial values of the estimated parameters are randomly assigned. Therefore, at the beginning of the estimation process, N chromosomes are generated as random binary strings.

3) Parent Selection: The selection of parent chromosomes is based on the notion of "fitness," which governs the extent to which an individual can influence future generations. For parameter estimation problems, the fitness of a particular chromosome is roughly proportional to the inverse of the error. In the conventional GA [3]–[4], chromosomes are selected for mating based on the ratio of their fitness value to the sum of total fitness value of all chromosomes in the generation. In [9], the fitness value for the jth chromosome in the ith generation is defined as $e^i_{\max} - e_{ji}$, where $e^i_{\max}$ is the maximum estimation error in generation i. The probability of the jth chromosome being selected for mating in the next generation is

$$P^i_j = \frac{e^i_{\max} - e_{ji}}{\sum_{k=1}^{i}(e^i_{\max} - e_{ki})}. \tag{2.6}$$

However, the estimated error can be quite sensitive to variations of the estimated parameters. In the event of an unstable estimated system, $e^i_{\max}$ is huge, and P^i_j in (2.6) is small even for chromosomes that are close to the ideal values. Using this fitness criterion, chromosomes representing parameter sets that are close to the optimal values might be discarded (not selected for mating), resulting in slow parameter convergence and poor performance.

Since improvement of the estimation error requires that the better parameter sets (chromosomes) be allowed to mate, the GA is modified to ensure that these chromosomes are mated in every generation. In each generation, the best D chromosomes corresponding to the least estimation errors among the N chromosomes of the present gene pool are selected for mating, and then some of them, say, $\rho \cdot D$ best chromosomes, are allowed to surive into the next generation. Because some of the D best parents are allowed to live into the next generation, the minimum estimation error in the current generation will be

always less than or equal to the minimum estimation error in the previous generation.

4) Mating: Among the D potential parents, pairs are randomly selected for mating, which is carried out via a crossover procedure that mimics biological mating. Basically, crossover combines the features of two parent chromosomes to form two new "children" chromosomes. Given two parents and a randomly assigned splice point, the crossover procedure can be illustrated as follows.

```
parents 1   xxxxxxxxxxxxx
parents 2   yyyyyyyyyyyyy
                  ↑ splice point
```

Two children are created in the crossover procedure. Child 1 contains the front part of parent 1 and the tail part of parent 2, whereas child 2 contains the front part of parent 2 and the tail part of parent 1.

```
child 1   xxxxxyyyyyyyy
child 2   yyyyyxxxxxxxx
```

Assume the schema xxxxx******** corresponds to low estimation error, where * represents an unspecified value (or a wild card). Then, child 1 might correspond to a lower estimation error than either parent does. It is this (biologically inspired) mating procedure, and the draconian annihilation of unfit chromosomes that separates the GA from truly random search style algorithms.

It appears to be more efficient to perform the mating at the parameter (gene) level rather than the chromosome level. In each generation, since $\rho \cdot D$ parents are passed from the previous generations, $N - \rho \cdot D$ new childrn are born; leaving the population constant throughout the procedure.

5) Mutation: Generally, over a period of several generations, the gene pool tends to become more and more homogeneous as one gene begins to dominate. A mutation feature is often introduced to guard against premature convergence (to a nonoptimal solution). Mutation randomly alters the gene from 0 to 1 or from 1 to 0 with probability P_m. The purpose of mutation is to introduce occasional perturbations to the estimated parameters to ensure that all points in the search space can ultimately be reached. Generally, if P_m is larger, the convergence rate is faster but a larger "steady state error" results. This is, in some respects, analogous to the stepsize parameter of gradient algorithms; larger stepsizes imply faster convergence but higher steady state error, whereas smaller stepsizes imply slower convergence but smaller final error. Some studies (such as [16]) have suggested changing the mutation probability P_m adaptively so that a faster converging rate and finer tuned estimated parameter can be both achieved.

6) Extinction and Immigration: Holland has shown in [3] that the number of chromosomes in the gene pool associated with smaller estimation errors grows exponentially. Therefore, after some generations, the D parent chromosomes chosen to mate are eventually the same. It is clear that if two parents are identical, their children will also be identical and no new information is obtained. The estimation thus tends to stagnate, and the only mechanism to generate better chromosomes is mutation. Since P_m is generally small (<0.1), the probability of further reducing estimation error is very small, especially for long chromosomes. We propose a drastic technique called extinction and immigration to bypass this difficulty.

Extinction eliminates all of the chromosomes in the current generation except the chromosome corresponding to the minimum estimation error. $N-1$ chromosomes are then randomly generated to fill out the population (a mass immigration). $D-1$ chromosomes associated with the least estimation errors among these immigrants are then selected as the parents. Together with the surviving chromosome, these are allowed to mate as usual to form the next generation. For convenience, we say that another *era* begins. Extinction and immigration is analogous to a particular time varying mutation rate in which P_m is close to 1 at the beginning of each era and then small for the remaining generations within the era.

There are two cases when extinction and immigration will occur. One is the case when all of the D parents are the same; the other is the case when no further decrease in the minimum estimation error has been detected for, say, L generations. Of course, these two cases are not independent since the first causes the second. However, when the number of parents D are large, it is often more efficient to stop the current mating and go through the extinction and immigration process without waiting for all parents to become identical. Continuing the analogy with linear gradient identification methods, one might think of the extinction and immigration policy as a type of "covariance reset" strategy.

6) Reencoding of Zeros: If the ideal value of one parameter is negative and the estimated value is initialized positive, there is the possibility that the parameter will linger around zero for many generations. This can be cured by reencoding the zeros, i.e., if the bit string for one parameter is $00\cdots00$,reencode it as $11\cdots11$, or vice versa. This is sensible since $1111\cdots = 0000\cdots$ in one's complement arithmetic.

C. Algorithm Statement

There are several tuning parameters to be set before running the GA. These parameters are as follows:

N	number of chromosomes in each generation
D	number of chromosomes chosen as parents for mating
L_t	number of generations tolerated for no improvement on the value of the minimum estimation error before the GA is terminated.
L_e	number of generations tolerated for no improvement on the value of the minimum estimation error before the operator extinction and immigration is applied. Note that $L_e \ll L_t$.
P_m	probability of mutation
ρ	the portion of chosen parents allowed to survive into next generation
ζ	desired accuracy of estimation

Two variants of the GA are stated (GA1 and GA2) without and with the extinction and immigration operator, respectively. The algorithm GA2 is summarized as follows:

1. Set the tuning parameters described as above. Encode the parameters to be estimated and refer them as the chromosomes. Set $i = 0, k = 0$, and $m = 0$.
2. Initialize N chromosomes, let $i = i + 1$. $k = 0$, and $m = 0$.
3. Decode the chromosomes and calculate the estimation error e_{ij} for every jth chromosome in the ith generation. Let $e^i_{\min} = \min_j (e_{ij})$.
4. Select D parents. Pass $\rho \cdot D$ parents into next generation.
5. Mate D parents and generate $N - \rho \cdot D$ children. Invoke mutation along with the crossover procedure. Reencode zero(s), if the best chromosome corresponds to some zero parameter(s).
6. If $e^i_{\min} = e^{i-1}_{\min}, k = k + 1$ and $m = m + 1$; otherwise, $k = 0$ and $m = 0$.
7. If D parents are all the same, apply the extinction and immigration operation by returning to step (1), saving only the chromosome corresponding to $e^i_{\min}$. Set $k = 0$.
8. If $k = L_e$, then apply the operator extinction and immigration as described in step (6). Set $k = 0$.
9. If $m = L_t$ (or $e^i_{\min} < \zeta$), then terminate the algorithm; otherwise, go to step (2).

□

Note that if the variance of the measurement noise is known *a priori*, then the GA is terminated if the estimation error is less than ζ. However, if no statistical information of measurement noise is known *a priori*, the GA is set to be terminated when no improvement on the estimation error has been detected for L_t generations. The algorithm GA1 is basically the same as GA2 as above except that steps (6) and (7) for the operator "extinction and immigration" are eliminated.

III. Convergence of the GA

Since the chromosome corresponding to the minimum estimation error in each generation always survives into the next generation, the estimation error is montonically decreasing sequence. Thus, the estimation error must converge since it is bounded below. In this section, the estimation error is further shown to converge in probability to zero.

Proposition 3.1: Suppose D parent chromosomes are selected from the present generation. Of all possible children that can be derived from the set of D parents through crossover, the probability of generating any particular child chromosome is

$$P = \frac{1}{D(D-1)} \prod_{i=1}^{h} \frac{1}{2(B_i - 1)} \tag{3.1}$$

assuming there is no mutation ($P_m = 0$).

Proof: Among D parent chromosomes, there are $D(D-1)$ possible ways to choose a pair of chromosomes for mating. Since the crossover is made parameterwise, $\Pi_{i=1}^{h} 2(B_i - 1)$ possible children can be generated by any given pair of parents, where B_i denotes the number of bits of ith parameter. Since the choice of parents is made independent of crossover, these probabilities multiply. □

Note that for identical chromosomes, the probability of generating each of those identical ones is also given as in (3.1).

Proposition 3.2: Given two arbitrary chromosomes C_i and C_j with the corresponding estimation errors e_i and e_j respectively, there exists a fixed constant $\tilde{\alpha} > 0$ such that if $e_i > e_j$, then

$$e_i - \tilde{\alpha} \geq e_j. \tag{3.2}$$

Proof: Let $\alpha_{ij} = |e_i - e_j|, \forall i, j$. Since every chromosome is represented by finite number of bits, there are only a finite number of possible chromosomes and, hence, a finite number of possible α_{ij}. Let $\tilde{\alpha}$ be the smallest nonzero α_{ij}. □

Since each gene consists of only a finite number of bits, the estimated parameters represented by these genes are actually quantized values. The smallest quantum for each estimated parameter is the desired encoding accuracy $\delta_k, k = 1 \cdots h$. If the ideal value of the estimated parameter is w^* and its quantized representation is $[w^*]_Q$, the quantization error will be

$$|w^* - [w^*]_Q \leq \delta_k, k = 1, \cdots, h. \tag{3.3}$$

The case when the quantization error of each estimated parameter is zero is called *ideal matching*. Under the ideal matching assumption, the following theorem shows that the minimum estimation error converges in probability to zero at a specific rate. This rate will be proportional to $\Sigma E(\theta_i)$, where θ_i is the number of all possible chromosomes in generation $i + 1$ that correspond to estimation errors less than $e^i_{\min}$. If the assumption of ideal matching fails, then the minimum estimation error will converge to a constant.

Theorem 3.1: Let $e^g_{\min}$ be the estimation error in generation g as in (2.4). If the ideal matching condition is satisfied for GA1, then

$$(1) \quad E(e^g_{\min}) \leq E(e^0_{\min}) - c \sum_{i=0}^{g-1} E(\theta_i) \quad \forall g \geq 1 \tag{3.4}$$

where c is a constant. Moreover

$$(2) \quad P(e^g_{\min} \geq \tilde{\alpha}) \to 0 \quad \text{as} \quad g \to \infty \tag{3.5}$$

where $\tilde{\alpha}$ is defined as in Proposition 3.2.

Proof: Let Φ be the set of all possible D chromosomes chosen as the parents in generation g and $\phi \in \Phi$ be the best D chromosomes. Assume that splice points occur at $s_1, \cdots, s_h$ for $\hat{w}_1, \cdots, \hat{w}_h$, where $s_i \in [1 \cdots B_i - 1], i = 1 \cdots h$. Each bit of the chromosome is mutated with probability P_m. Let U be the set of all possible patterns of mutation including the pattern of no mutation, and let $e^{g+1}_{ijs_1 \cdots s_h u}$ be the estimation erorr at generation $g+1$ associated with the chromosome which is a child of ith and jth chromosome chosen from previous generation with splice points at $s_1, \cdots, s_h$ and mutation pattern u. Then, if the best D chromosomes, which are denoted by ϕ are selected at generation g

$$\begin{aligned} E(e^{g+1}_{\min} | \phi) &= \sum_{i \in D} \sum_{j \in D, i \neq j} \sum_{s_1} \cdots \sum_{s_h} \sum_{u \in U} \\ &\quad \cdot P_{ijs_1 \cdots s_h} P_u \min (e^g_{\min}, e^{g+1}_{ijs_1 \cdots s_h u}) \\ &= P \sum_{ijs_1 \cdots s_h u} P_u \min (e^g_{\min}, e^{g+1}_{ijs_1 \cdots s_h u}) \end{aligned} \tag{3.6}$$

where $P_{ijs_1\cdots s_h}$ is the constant P from Proposition 3.1, and $e^{g+1}_{ijs_1\cdots s_h u}$ is the associated estimation error for any possible chromosome in generation $g+1$. Let

$$\alpha_{ijs_1\cdots s_h u} = e^g_{\min} - e^{g+1}_{ijs_1\cdots s_h u}, \quad \forall e^{g+1}_{ijs_1\cdots s_h u} < e^g_{\min} \tag{3.7}$$

then (3.6) can be written as

$$\begin{aligned} E(e^{g+1}_{\min}|\phi) &= P \sum_{\substack{ijs_1\cdots s_h u \\ e^{g+1}_{ijs_1\cdots s_h u} \geq e^g_{\min}}} P_u e^g_{\min} + P \\ &\quad \cdot \sum_{\substack{ijs_1\cdots s_h u \\ e^{g+1}_{ijs_1\cdots s_j u} < e^g_{\min}}} P_u (e^g_{\min} - \alpha^g_{ijs_1\cdots s_h u}) \\ &= P \sum_{ijs_1\cdots s_h u} P_u e^g_{\min} - P \sum_{\substack{ijs_1\cdots s_h u \\ e^{g+1}_{ijs_1\cdots s_h u} < e^g_{\min}}} P_u \alpha^g_{ijs_1\cdots s_h u} \\ &= e^g_{\min} - PF_g \end{aligned} \tag{3.8}$$

where

$$F_g = \sum_{\substack{ijs_1\cdots s_h u \\ e^{g+1}_{ijs_1\cdots s_h u} < e^g_{\min}}} P_u \alpha^g_{ijs_1\cdots s_h u}. \tag{3.9}$$

Due to mutation, all chromosomes are possible, albeit with small probability. Therefore, if $e^{g+1}_{\min} > \tilde{\alpha}$ then $\theta_g \geq 1$. Let $\tilde{P}$ be the probability of mutating every bit in the chromosome, and then

$$\tilde{P} = P_m^{\sum_{i=1}^h B_i}. \tag{3.10}$$

The probability of mutating j specific bits in the chromosome is given by

$$\overline{P} = (1-P_m)^{\sum_{i=1}^h B_{i-j}} P_m^j; \qquad j = 0, 1, 2, \cdots \tag{3.11}$$

Equations (3.10) and (3.11) imply that

$$\tilde{P} \leq P_u \quad \forall u \in U \tag{3.12}$$

since P_m is usually set to be less than 0.1.

Proposition 3.2 and (3.12) show that

$$F_g \geq \theta_g \tilde{P} \min(\alpha^g_{ijs_1\cdots s_h}) \geq \theta_g \tilde{P}\tilde{\alpha}. \tag{3.13}$$

Recall that θ_g is the number of all possible chromosomes in generation $g+1$ that correspond to estimation errors less than $e^g_{\min}$.

Substituting (3.13) into (3.8) yields

$$E(e^{g+1}_{\min}|\phi) \leq e^g_{\min} - P\theta_g \tilde{P}\tilde{\alpha}. \tag{3.14}$$

$$\begin{aligned} E(E(e^{g+1}_{\min}|\phi)) &= \sum_{\phi\in\Phi} P_\phi E(e^{g+1}_{\min}|\phi) \\ &\leq \sum_{\phi\in\Phi} P_\phi e^g_{\min} - P\alpha\tilde{P} \sum_{\phi\in\Phi} P_\phi \theta_g \\ &= E(e^g_{\min}) - P\alpha\tilde{P}E(\theta_g). \end{aligned} \tag{3.15}$$

By the theorem of total expectation [27], $E(E(e^{g+1}_{\min}|\phi)) = E(e^{g+1}_{\min})$. Consequently, (3.15) implies

$$E(e^g_{\min}) \leq E(e^0_{\min}) - P\tilde{P}\tilde{\alpha}\sum_{i=0}^{g-1} E(\theta_i) \quad \forall g \geq 1. \tag{3.16}$$

Thus, (3.4) holds with the constant $c = P\tilde{P}\tilde{\alpha}$.

Since $\theta_i \geq 1, \forall i \geq 0, E(e^g_{\min}) \to 0$ as $g \to \infty$. By the extension of the Chebyshev inequality in [22]

$$\lim_{g\to\infty} P(e^g_{\min} \geq \tilde{\alpha}) \leq \lim_{g\to\infty} \frac{E(e^g_{\min})}{\tilde{\alpha}} = 0 \tag{3.17}$$

which implies (3.5). □

Theorem 3.1 proves the convergence of the modified GA within one era. The proof of convergence for the modified GA, which consists of the "extinction and immigration" operator, is shown in the following corollary.

Corollary 3.1: Let era $1, 2, \cdots, m$ consist of $T_1, T_2, \cdots, T_m$ generations, respectively, and

$$g = T_1 + T_2, \cdots T_m + j. \tag{3.18}$$

If the ideal matching assumption is satisfied for GA2, then (3.4) and (3.5) both hold.

Proof: From (3.16)

$$E(e^{T_1}_{\min}) \leq E(e^0_{\min}) - P\tilde{P}\tilde{\alpha}\sum_{i=0}^{T_1-1} E(\theta_i). \tag{3.19}$$

$$\vdots$$

$$E(e^{T_1+\cdots T_m+j}_{\min}) \leq E(e^{T_1+\cdots T_m}_{\min}) - P\tilde{P}\tilde{\alpha}\sum_{i=T_1+\cdots T_m}^{T_1+\cdots T_m+j-1} E(\theta_i). \tag{3.20}$$

Therefore

$$E(e^{T_1+\cdots T_m+j}_{\min}) \leq E(e^0_{\min}) - P\tilde{P}\tilde{\alpha}\sum_{i=0}^{T_1+\cdots T_m+j-1} E(\theta_i). \tag{3.21}$$

Referring to (3.18) and (3.21), (3.4) thus holds. Consequently, (3.5) also holds. □

Theorem 3.1 and its corollary imply that if the ideal matching condition is satisfied, then the estimation error converges in probability to zero. Note that this does not necessarily imply convergence of the parameter estimates $\hat{w}_i$ to their true values w_i^* unless additional "persistance of excitation" conditions [1], [23] are invoked on the input vector U of (2.1). This is tantamount to an identifiability condition and is dependant on the structure of the nonlinearity $f(\cdot)$ of (2.1). Deriving such conditions for interesting classes of f's is an important topic that deserves further attention.

Referring to (2.2) and (2.3), the quantized steady estimation error is defined as

$$[e^*_{\min}]_Q = \frac{1}{d}\sum_{k=1}^{d} (y^*(k) - [y^*(k)]_Q)^2 \tag{3.22}$$

where $y^*(k)$ is the output due to ideal values of estimated parameters $w_1^*, \cdots, w_h^*$ and $[y^*(k)]_Q$ is the output due to $[w_1^*]_Q, \cdots, [w_h^*]_Q$. The quantized estimation eror $[e_{\min}^*]_Q$ is therefore a (finite) constant that depends on the quantization error δ_k, the size of the input U, and the "gain" of the system. Theorem 3.1 and its corollary thus imply that as $g \to \infty, \hat{y}(k) - [y^*(k)]_Q$ converges in probability to zero. Hence, $[e_{\min}^*]_Q$ provides an upper bound on the asymptotic error. The implication of this is that the GA is robust not only to suitably small mismodelling errors and to noisy data but also to quantization errors.

Let era 1, 2, 3, $\cdots$ consist of $T_1, T_2, T_3, \cdots$ generations. Let $e_{\min}^{'g}$ and $e_{\min}^{g}$ denote the minimum estimation error due to GA1 and GA2 in generation g. There is no difference between GA1 and GA2 in the first era. Once the parent chromosomes are all identical at $g = T_1$, the only way for GA1 to further decrease the estimation error $e_{\min}^{'g}$ is through mutation. In this case, the parameter estimation is like a simple random search. It has been shown in [4], [24] that N^3 schemata are implicitly searched by GA in each generation. Since the parent chromosomes are identical at $g = T_1$, most of the schemata searched by GA1 are actually the same: $\forall g \geq T_1$. However, for GA2, many diverse immigrants are introduced into the gene pool, and for $g \geq T_1 + 1$, GA2 searches many more schemata than GA1.

The best chromosome from era 1 (C_{best}) survives into era 2. Due to the quick dominance of C_{best} in succeeding generations, a large number of schemata that are close to C_{best} are examined. Thus

$$E(e_{\min}^{T_1} - e_{\min}^{T_1+T_2}) \leq E(e_{\min}^{'T_1} - e_{\min}^{'T_1+T_2}). \tag{3.23}$$

But $E(e_{\min}^{T_1}) = E(e_{\min}^{'T_1})$, hence

$$E(e_{\min}^{T_1+T_2}) \leq E(e_{\min}^{'T_1+T_2}). \tag{3.24}$$

Applying the same arguments iteratively to other eras shows

$$E(e_{\min}^{k}) \leq E(e_{\min}^{'k}) \quad \forall k = T_1+T_2, T_1+T_2+T_3, \cdots \tag{3.25}$$

To gain a better idea of the performance of the modified GA, we may also look at its efficiency. Recall that there are $B_1, \cdots, B_h$ bits used to encode the estimated parameters $\hat{w}_1 \cdots \hat{w}_h$. The number of different values to be searched using a truly random searching method would be

$$(2^{B_1} - 1) \cdots (2^{B_h} - 1) \approx 2^{B_1 + \cdots B_h} \tag{3.26}$$

Suppose G generations are needed to achieve the estimation error $e_{\min} \leq \zeta$; then, the total number of values the modified GA has searched is hNG. Define the efficiency index λ for this algorithm

$$\lambda = -\log\left(\frac{hNG}{2^{B_1+\cdots B_h}}\right). \tag{3.27}$$

It is clear that λ is usually a large number. As the number of estimated parameters increases, the values in the denominator and numerator of (3.27) both increase. Since the crossover is made parameterwise, the GA carries out multiple operations in parallel. Therefore, the number of generations for GA to achieve the estimation error $e_{\min} \leq e_b$ is not greatly increased as the number of estimated parameters increases. Empirically, G tends to increase linearly while the number of different values to be searched by truly random searching method increases exponentially. Therefore, the efficiency λ tends to increase as the number of estimated parameters increases.

IV. Examples

In this section, numerical examples are given to show applications to the GA to parameter estimations of linear and nonlinear IIR filters, feedforward and recurrent neural networks, and frequency modulated sinusoidal signals. The reason FIR filters are not simulated is because they are simply a special case of IIR filters.

For the following examples, the tuning parameters are set as follows.

$$N = 240, \quad D = 60, \quad L_e = 15, \quad P_m = 0.01, \quad \rho = 0.5.$$

Note that in Examples 4.1 and 4.2, every parameter to be estimated is encoded by 6 b for the integer part and 7 b for the decimal part, whereas in Example 4.3, every parameter is encoded by 9 and 3 b for the integer and decimal parts, respectively. The desired accuracy ζ varies with the different examples.

Example 4.1: (IIR Filters) In this example, the GA is applied to estimate the parameters of the "unknown" linear IIR filter

$$\begin{aligned} y(k) = &-0.3y(k-1) + 0.4y(k-2) \\ &+ 1.25u(k-1) - 2.5u(k-2) + n(k) \end{aligned} \tag{4.1}$$

and the "unknown" nonlinear IIR filter

$$\begin{aligned} y(k) = &\left(\frac{3 - 0.3y(k-1)u(k-2)}{5 + 0.4y(k-2)u^2(k-1)}\right)^2 \\ &+ (1.25u^2(k-1) - 2.5u^2(k)) \ln(|1.25u^2(k-2) \\ &- 2.5u^2(k)|) + n(k) \end{aligned} \tag{4.2}$$

where the input signal is uniformly distributed between -2.5 and 2.5, and the measurement noise $n(\cdot)$ is uniformly distributed between -0.25 and 0.25. In both (4.1) and (4.2), the parameters -0.3, 0.4, 1.25, and -2.5 are those to be estimated. Note that the parameter -0.3 and 0.4 cannot be encoded exactly by the assumed 7 b for decimal part, demonstrating the robustness of the GA to quantization errors. With window size $d = 200$, the convergence of estimation error due to GA1 and GA2 are compared in Figs. 1 and 2 for the linear and nonlinear IIR filter, respectively. Clearly, GA2 with the "extinction and immigration" operator substantially outperforms the GA1 without this operator. It can be inferred from both figures that the gene pool in GA1 is eventually dominated by the best chromosome. As a result, the convergence of the estimation error stagnates. However, the gene pool of GA2 is rejuvenated from time to time by the "extinction and immigration" operator, and the convergence of the estimation error is faster than with GA1. □

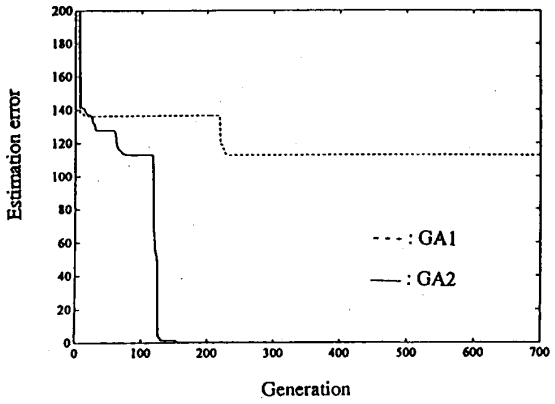


Fig. 1. Estimation error convergence of the linear IIR filter, Example 4.1.

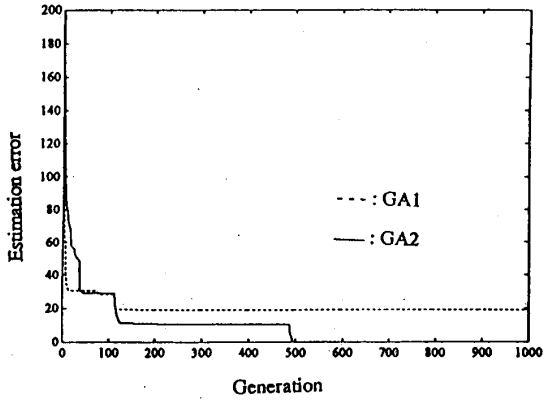


Fig. 2. Estimation error convergence of the nonlinear IIR filter, Example 4.1.

Example 4.2: (Neural Networks) Since feedforward and recurrent neural networks are particular parameterizations of nonlinear FIR filters and IIR filters, respectively, the modified GA can also be directly applied to the estimation of weights and biases of the units (neurons) in such network structures. Suppose that the activation function in each unit j is the logistic function

$$f = \frac{1}{1 + \exp\left(-\sum_i w_{ji}u_i + b_j\right)} \tag{4.3}$$

where u_i are the inputs to the jth unit (which may be inputs to the network, outputs from units in the previous layers, outputs from the unit itself, or outputs from units in succeeding layers). At each unit, the weight w_{ji} and the bias b_j are to be estimated.

In this example, the GA is used to train the neural network model to perform bit rotations of input bit strings. For instance, if the input string is b_1, b_2, b_3, and the task is to rotate the string to the left by two places, then the desired output is b_3, b_1, b_2. A feedforward and a recurrent network are trained to perform this task. The structures of feedforward and recurrent network are shown in Figs. 3(a) and (b), respectively. A total of 64 randomly shuffled training patterns are generated in which the output patterns are contaminated by uniformly distributed

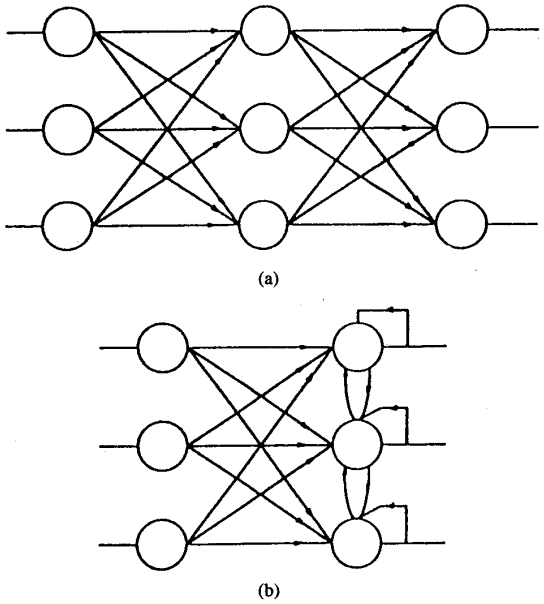


Fig. 3. (a) Feedforward neural network, Example 4.2; (b) recurrent neural network, Example 4.2.

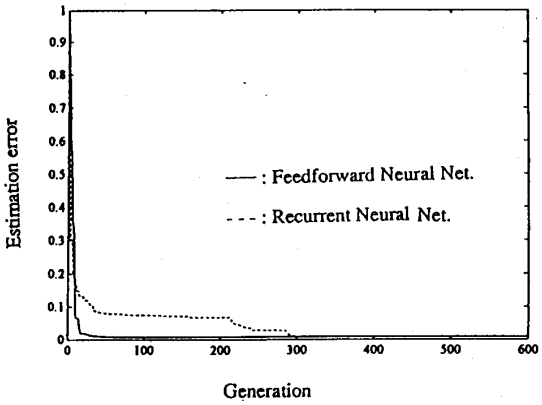


Fig. 4. Estimation error convergence of feedforward and recurrent neural networks, Example 4.2.

noise between -0.1 and 0.1. For the feedforward and recurrent neural network, there are 24 parameters to be trained, whereas for the recurrent network, there are 19 parameters to be trained. Note that there are no hidden units in the recurrent network. The convergnce of the estimation error for both structures are shown in Fig. 4. □

Example 4.3—(Modulated Sinusoidal Signals) In [25] and [26], it is shown that musical waveforms can be approximated based on a frequency modulation scheme. Let the music generation unit be given by

$$m_u(t) = A\sin\left(\omega_c t + I\sin\left(\omega_m t\right)\right) \tag{4.4}$$

where A is a constant denoting the magnitude of the waveform, ω_c is the carrier frequency, I is the modulation index, and ω_m is the modulating frequency. The musical waveform can be approximated by cascaded or parallel music generation units.

Refer to [25] and [26] for further details. In this example, the GA is applied to estimate the carrier, modulating frequencies, and modulation indices based on the modulated sinusoidal signals generated by the cascaded form

$$y_1(t) = \sin(\omega_{11}t + I_{11}\sin(\omega_{12}t + I_{12}\sin(\omega_{13}t))) + n(t) \quad (4.5)$$

and the parallel form

$$y_2(t) = \sin(\omega_{21}t + I_{21}\sin(\omega_{22}t)) + \sin(\omega_{23}t + I_{22}\sin(\omega_{24}t)) + n(t) \quad (4.6)$$

where $n(\cdot)$ is the measurement noise uniformly distributed between -0.25 and 0.25. The parameters are set to be

$$\omega_{11} = 30, \quad \omega_{12} = 200, \quad \omega_{13} = 150,$$
$$\omega_{21} = 40, \quad \omega_{22} = 50, \quad \omega_{23} = 45,$$
$$I_{22} = 30.$$
$$I_{11} = 20, \quad I_{12} = 50,$$
$$\omega_{24} = 45, \quad I_{21} = 50.$$

Note that (4.5) and (4.6) can be considered to be nonlinear FIR filters so that the parameters can be estimated in the same way as in previous examples. The only difference is that all parameters to be estimated in this example are positive, which simplifies the coding of the chromosome.

Let the sampling rate be 10^7 samples/s for (4.5) and 10^5 samples/s for (4.6). Accumulating the estimation errors for the 1000 samples (i.e., $d = 1000$), the estimated parameters by the GA are

$$\hat{\omega}_{11} = 33, \quad \hat{\omega}_{12} = 447.63, \quad \hat{\omega}_{13} = 97.5,$$
$$\hat{\omega}_{21} = 48, \quad \hat{\omega}_{22} = 47.75, \quad \hat{\omega}_{23} = 2.75,$$
$$\hat{I}_{22} = 31.75.$$
$$\hat{I}_{11} = 20, \quad \hat{I}_{12} = 74.38,$$
$$\hat{\omega}_{24} = 43.75, \quad \hat{I}_{21} = 51.75.$$

The modulated sinusoidal signals in (4.5) and (4.6) are compared in Figs. 5 and 6, respectively, with the signals $\hat{y}_1(t)$ and $\hat{y}_2(t)$, which are regenerated based on the estimated parameters. It is interesting to note that the parameter estimates are fundamentally different from their true values yet the regenerated signals based on these estimates match very well with the original signals. This demonstrates a fundamental lack of identifiability of the parameters in both (4.5) and (4.6). Equivalently, the system must be failing to satisfy some set of "persistance of excitation" conditions since the estimation errors converge towards zero even though the parameter values are not the same as used to generate the waveforms.

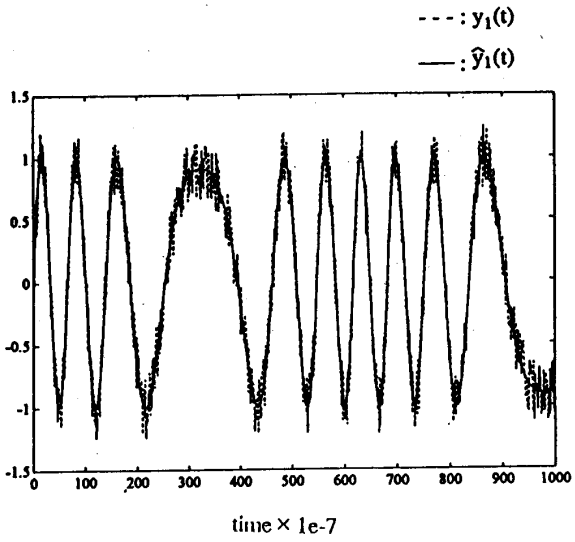


Fig. 5. Comparison of $y_1(t)$ and $\hat{y}_1(t)$, Example 4.3.

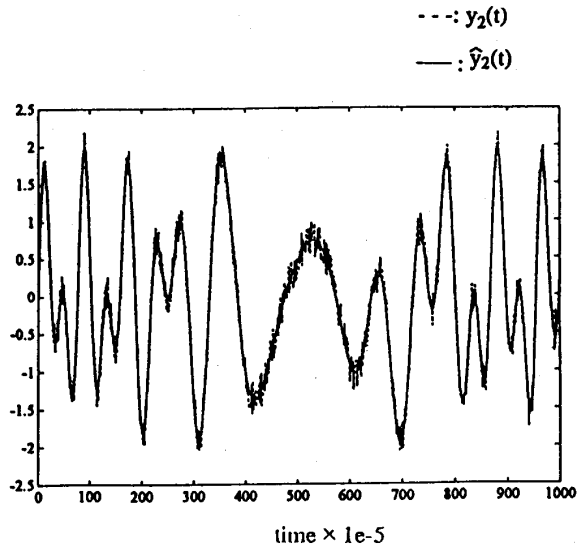


Fig. 6. Comparison of $y_2(t)$ and $\hat{y}_2(t)$, Example 4.3.

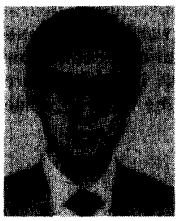
V. Concluding Remarks

The genetic algorithm is a good tool for the optimization of nonlinear functions. When the GA is applied to parameter estimation problems, it is especially powerful for nonlinear IIR filters since it can be applied in situations where gradient methods fail and is not susceptible to problems with local minima. The disadvantages of the GA stem from its computational complexity (when compared to a gradient approach) since it must process N different estimated parameter sets in each iteration. Thus, the GA should be seen primarily as a method for off-line identification, estimation, and optimization. Finally, it is not always obvious how to choose the user tunable parameters in an optimal fashion.

References

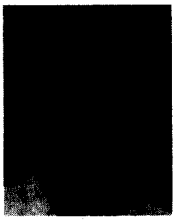
[1] C. R. Johnson, *Lectures on Adaptive Parameter Estimation*. Englewood Cliffs, NJ: Prentice-Hall, 1988.
[2] B. Widrow and S. D. Sterns, *Adaptive Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1985.
[3] J. H. Holland *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: University of Michigan Press, 1975.

[4] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. New York: Addison-Wesley, 1989.
[5] K. De Jong, "A 10 year perspective," in *Proc. Int. Conf. Genetic Algorithms Their Applications*, 1985, pp. 169–177.
[6] J. J. Grefenstette, "Optimization of control parameters for genetic algorithms," *IEEE Trans. Syst. Man. Cybern.*, vol. SMC-16, no. 1, pp. 122–128, Jan.-Feb. 1986.
[7] L. Davis and M. Steenstrup, "Genetic Algorithms and simulated annealing," in *Genetic Algorithms and Simulated Annealing*, L. Davis, Ed. London: Pitman, 1987, pp. 1–11.
[8] D. M. Etter and M. M. Masukawa, "A comparison of algorithms for adaptive estimation of the time delay between sampled signals," in *Proc. IEEE Int. Conf. ASSP*, 1981, pp. 1253–1256.
[9] D. M. Etter, M. J. Hicks, and K. H. Cho, "Recursive adaptive filter design using an adaptive genetic algorithm," in *Proc. IEEE Int. Conf. ASSP*, 1982, pp. 635–638.
[10] Y. Davidor, *Genetic Algorithms and Robotics, A Heuristic Strategy for Optimization*. Singapore: World Scientific, 1991.
[11] S. Matwin, T. Szapiro, and K. Haigh, "Genetic algorithms approach to a negotiation support system," *IEEE Trans. System Man Cybern.*, vol. 21, no. 1, pp. 102–114, Jan./Feb. 1991.
[12] R. Axelrod, "The evolution of strategies in the iterated prisoner's dilemma," in *Genetic Algorithms and Simulated Annealing* (L. David, Ed.). London: Pitman, 1987, pp. 32–41.
[13] K. De Jong, "Learning with the genetic algorithm: An overview," *Machine Learning*, vol. 3, pp. 121–137, Oct. 1988.
[14] G. A. Vignaux and Z. Michalewicz, "A genetic algorithm for the linear transportation problem," *IEEE Trans. Syst. Man Cybern.*, vol. 21, pp. 445–452, Mar./Apr. 1991.
[15] J. P. Cohoon, S. U. Hegde, W. N. Martin, and D. S. Richards, "Distributed genetic algorithm for the floorplan design problem," *IEEE Trans. Computer-Aided Des.*, vol. 10, pp. 483–491, Apr. 1991.
[16] D. J. Montana and L. Davis, "Training feedfoward neural networks using genetic algorithms," in *Proc. Int. Joint Conf. Artificial Intell.*, (Detroit), 1989, pp. 762–767.
[17] H. Kitano, "Empirical studies on the speed of convergence of neural network training using genetic algorithms," in *Proc. Eighth Nat. Conf. Artificial Intell.* (Boston), 1990, pp. 789–795.
[18] J. L. McClelland and D. E. Rumelhart, *Parallel Distributed Processing*. Cambridge, MA: MIT Press, 1986.
[19] F. J. Pineda, "Generalization of backpropagation to recurrent and higher-order networks," in *Proc. IEEE Conf. Neural Inform. Processing Syst.*, 1987, pp. 602–611.
[20] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Comput.*, 1989, pp. 270–280.
[21] K. De Jong, "Using experience-based in game playing," in *Proc. 5th Int. Conf. Machine Learning* (Ann Arbor, MI), 1988, pp. 284–290.
[22] H. Stark and J. W. Woods, *Probability, Random Processes and Estimation Theory for Engineers*.Englewood Cliffs, NJ: Prentice-Hall, 1986.
[23] K. S. Narendra and A. M. Annaswamy, *Stable Adaptive Systems*.Englewood Cliffs, NJ: Prentice-Hall, 1989.
[24] J. M. Fitzpatrick and J. J. Grefenstette, "Genetic algorithms in noisy environments," *Machine Learning*, vol. 3, pp. 101–120, Oct. 1988.
[25] J. Chowning, "The synthesis of complex audio spectra by means of frequency modulation," *J. Audio Eng. Soc.*, vol. 21, no. 7, pp. 526–534, 1973.
[26] J. Chowning and D. Bristow, *FM Theory and Applications, By Musicians for Musicians*. Tokyo: Yamaha, 1986.
[27] R. B. Ash, *Basic Probability Theory*. New York: Wiley, 1970.



Leehter Yao (S'86–M'92) received the Ph.D. degree in electrical engineering from University of Wisconsin–Madison, in 1992.

From 1988 to 1992, he was a research assistant in the Center for Health Systems Research and Analysis, University of Wisconsin. Since 1992, he has been with the Department of Electrical Engineering at the National Taipei Institute of Technology, Taiwan, R.O.C., where he is currently an associate professor. His research interests include adaptive systems in signal processing and automatic control, artificial intelligence, and pattern recognition.



William A. Sethares received the B.A. degree in mathematics from Brandeis University, Waltham, MA, and the M.S. and Ph.D. degrees in electrical engineering from Cornell University, Ithaca, NY.

He has worked at the Raytheon Company as a Systems Engineer and is currently on the faculty of the Department of Electrical and Computer Engineering at the University of Wisconsin in Madison. His research interests include adaptive systems in signal processing, communications and control, electronic music, and other fashionable topics. He especially enjoys writing brief biographical sketches.