Journal of University of Science and Technology Beijing
Vol.7(2000), No.1, p.58

Application of Genetic Algorithms in Identification of Linear Time-Varying System

Zhichun Mu[1], *Ke Liu*[1], *Zichao Wang*[1], *Datai Yu*[1], *D. Koshal*[2], *D. Pearce*[2]

1) Information Engineering School, University of Science & Technology Beijing, Beijing 100083, China
2) School of Engineering, University of Brighton, Brighton, UK
(Received 1999-01-04)

Abstract: By applying genetic algorithms (GA) to on-line identification of linear time-varying systems, a number of modifications are made to the Simple Genetic Algorithm to improve the performance of the algorithm in identification applications. The simulation results indicate that the method is not only capable of following the changing parameters of the system, but also has improved the identification accuracy compared with that using the least square method.

Key words: genetic algorithm; system identification; linear system

Genetic Algorithms (GA) has some excellent properties [1, 2]. It is a global optimization and parallel search method. These properties make it become an advantageous algorithm suitable for system identification [3, 4]. The most common method that has been fairly successfully used in linear system identification is the least square method. However, the least square method cannot satisfactorily trace the gradual changes of parameters of time-varying systems. By introducing GA into system identification, some application problems with the least square method can be solved and hence adaptive identification can be realized.

1 Genetic Algorithms

1.1 The overview of genetic algorithm

John Holland laid the foundation of GA with the goal to create an effective search algorithm by simulating the characteristics of a natural system. GA is loosely based upon the Darwinian principles of biological evolution. This approach has proved to be particularly effective in searching in irregular spaces, to which very limited prior knowledge is known. GA generally uses coded strings (chromosomes) of binary numbers (genes) to describe the search process, which is actually an analogy to the genetic coding in our own DNA structure where the coded chromosome is composed of many genes. Compared with natural evolution, the emulated process is more efficient, controllable and flexible for the applications of optimization and machine learning.

1.2 Primary procedure of GA implementation

(a) Encoding. Encode parameters into binary string.

(b) Generating initial population. Select randomly a set of individuals as initial population.

(c) Evaluation. Define the fitness function and evaluate fitness of every individual in the population.

(d) Reproduction. Select individuals on certain probability rate from the population and copy them into next generation without any change. The individuals with higher fitness value have more chance to be selected and copied.

(e) Crossover. Select two individuals as parents from population and choose bits randomly in their binary string and swap the selected sub-strings between the two individuals to generate new individuals in next generation (**figure 1**).

(f) Mutation. Select a number of individuals according to certain probability rate and determine randomly a bit in each selected individual, and turn the bit value to its reverse to generate new individual (**figure 2**).

(g) Iterating (c)–(f) until fitness of the population approaches a satisfied steady value.

1.3 Modifications to simple genetic algorithms

To use GA for system identification, some modifications need to be made to simple genetic algorithms to improve its performance.

One of the problems that often occur in GA's application is premature convergence. The abilities of global

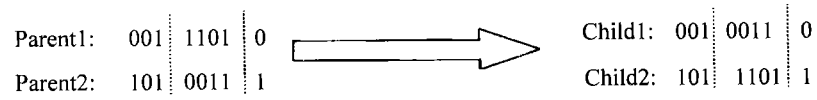

Figure 1 Crossover

Figure 2 Mutation

searching and optimization in GA mainly attribute to two aspects: the randomness of the generation of initial population and gene's recombination (crossover) as well as the operation of the mutation. However, limitation on the size of initial population will reduce the randomness, or in other words, reduce the diversity of initial individuals, which consequently restricts the ability of global searching. In addition, the operation of the mutation is usually performed according to a fixed probability. This makes the mutation effective at initial stage of evolution where individuals in population are diverse, but as the evolution process continues the mutation becomes less and less effective at the later stages. Once all individuals evolve towards homologous, the mutation operation with fixed rate has little effect to avoid premature convergence.

On the contrary to premature convergence, GA's convergence could be very slow in some other cases. The convergence relies on proper implementation of the mechanism of competition and gene's recombination and crossover. The mechanism works in accordance with Darwinian principle of the survival of the fittest, which makes the fittest gene remains and the genes of the population evolve towards the optimal. However, without modification it sometimes happens in the process that the fittest individual in a generation may disappear in next generation because of recombination with an inferior individual. This slows down the convergence.

Based on the above analysis, modifications are made accordingly in two aspects to improve the performance of the algorithm:

(a) To deal with the premature convergence problem, make the mutation probability Pm adjustable, *i.e.*, at the initial stage of evolution let Pm be a smaller value to start with and at later stages of evolution increase Pm to make individuals diverse. In addition, during the evolution process, keep generating a number of new individuals randomly according to a given probability and recombine them with the individuals selected from the generations evolved from the initial population so that the population can continue to take in new genes to guarantee that the population optimization is global.

(b) To prevent the fittest genetic information in a generation being "lost" during the evolution, copy certain number of individuals that have higher fitness value directly to the next generation.

2 Application of GA-based system identification

2.1 System model

In this paper, the system being investigated is shown in **figure 3**. It is treated as a "black box" because only

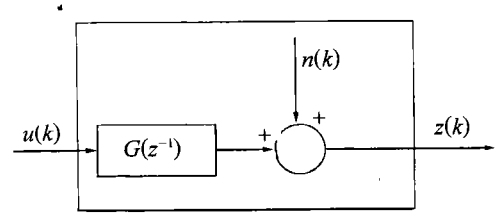

Figure 3 System structure

the dynamic relations between input /output are concerned. In figure 3, $u(k)$ is the input and $z(k)$ is the output, and both of them are assumed measurable. $G(z^{-1})$ denotes the plant dynamics. Since $G(z^{-1})$ could be generally expressed as

$$G(z^{-1}) = B(z^{-1}) / A(z^{-1}) \tag{1}$$

where

$$A(z^{-1}) = 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_n z^{-n} \tag{2}$$

$$B(z^{-1}) = b_1 z^{-1} + b_2 z^{-2} + \cdots + b_m z^{-m} \tag{3}$$

Therefore, the system model to be identified can be written in the following form:

$$z(k) = a_1 z(k-1) + a_2 z(k-2) + \cdots + a_n z(k-n) + b_1 u(k-1) + b_2 u(k-2) + \cdots + b_m u(k-m) + n(k) \tag{4}$$

where $n(k)$ denotes noise.

2.2 Encoding

All the parameters to be estimated are encoded into

binary strings in this practice to form chromosomes. The length of a binary string is decided according to the required precision of the parameters to be estimated. For example, if the range of the parameter to be estimated is $(x_{max} - x_{min})$, and the required precision is ε, then the length of the binary code l is determined by the following formula:

$$\frac{x_{max} - x_{min}}{2^l} < \varepsilon \tag{5}$$

Combining all the sub-strings of the parameters together, an individual G in the population is represented in a form of binary code chain, *i.e.*

$$G = \{\underbrace{\#\#\cdots\#}_{a_1}\ \underbrace{\#\#\cdots\#}_{a_2}\ \cdots\ \underbrace{\#\#\cdots\#}_{a_n}\ \underbrace{\#\#\cdots\#}_{b_1}\ \underbrace{\#\#\cdots\#}_{b_2}\ \cdots\ \underbrace{\#\#\cdots\#}_{b_m}\} \tag{6}$$

where # denote '0' or '1'.

2.3 Fitness function and evaluation of population

The fitness function of an individual F_R (R=1, 2, $\cdots L$) is defined as

$$F_R = F_M - \sum_{k=1}^{N} \lambda_k e(k) \tag{7}$$

Where L is the size of the population, N is the number of samples of input and output data $(u(k), y(k))$ taken from the system investigated, and F_M is a positive number which is chosen to ensure that F_R is positive. $e(k)$ is the error between system measurement $y(k)$ and the model output $z(k)$ as defined in equation (4), *i.e.*

$$e(k) = y(k) - z(k),$$

λ_k is an adjustable weight, which varies according to the creditability assigned to corresponding $e(k)$. For example, if λ_k is defined as

$$\lambda_k = u^{N-k} \quad (0<u<1) \tag{8}$$

i.e. λ_k=1 when $k = N$, and $\lambda_k << 1$ when k=0. It means, by definition (8), that more confidence has been assigned to the current data which would be more influential to the evolution process.

2.4 Implementation procedures

In GA's application, there are no general guidelines available to choose the size of population L, probability of crossover P_c and probability of mutation P_m. Normally they are assumed respectively in the following ranges:

L = 60–100 ; P_c = 0.6–0.8; P_m = 0.02–0.1.

As stated in section 2.3, some modifications have been made in this paper to the genetic algorithms regarding its application in system identification.The flow chart of the application is shown in **figure 4**.

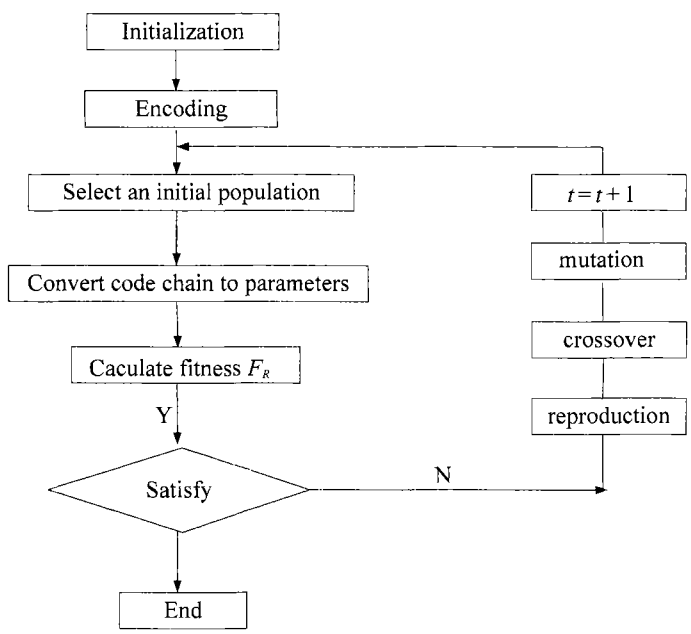

Figure 4 Flow chart of the GA-based identification

3 Simulation

The algorithm described above has been verified by simulation on the following system:

$$y(k) = 0.3y(k-1) - 0.4y(k-2) + 0.8u(k-1) + 0.7u(k-2) + n(k) \tag{9}$$

where $n(k)$ is white noise, $u(k)$ is input signal represented by a pseudo random sequence. The model to be identified is in the form of

$$z(k) = a_1 z(k-1) + a_2 z(k-2) + a_3 z(k-3) + b_1 u(k-1) + b_2 u(k-2) + b_3 u(k-3) \tag{10}$$

According to the prior knowledge, the range of parameters and required accuracy, the encoding length of each parameter is decided by equation (5) as 10, *i.e.* the code chain of an individual is made up with six ten-bit binary string.

The size of initial population L is set to 80 and the number of effective sequential samples is 100 (*i. e.* N=100). To keep this number at 100, the sequential data has to be refreshed in the process of on-line identification, taking in the new measurement and dropping out the oldest one at every sampling time.

Three cases have been investigated by simulation to evaluate the performance of the GA-based identification method.

Case 1: system with constant parameters. The system to be identified is assumed to have constant parameters. The output of the model identified is shown in **figure 5**, compared with the system output, and the parameters identification at 250th generation are listed in **table 1** against their true value.

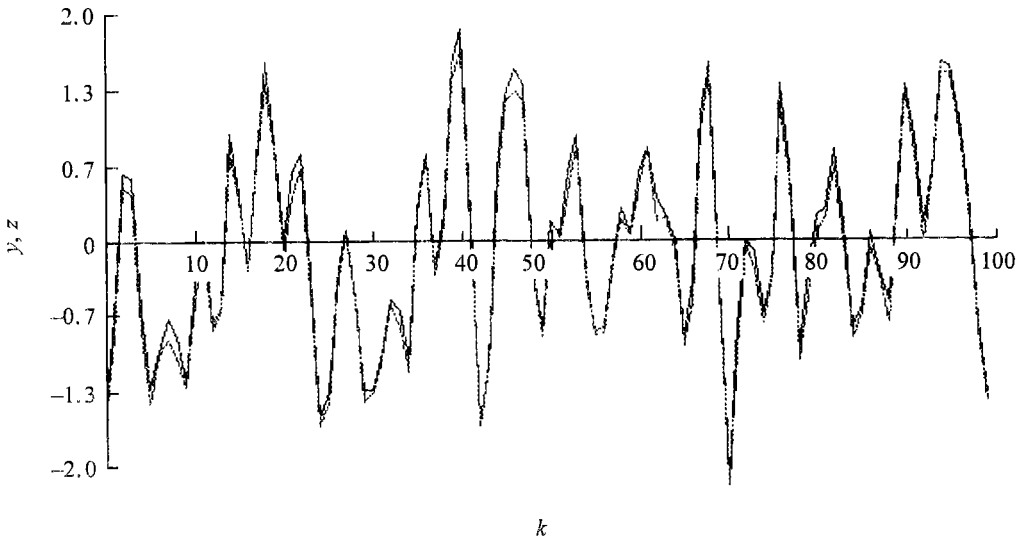

Figure 5 System output $y(k)$ (solid line) and model output $z(k)$ (dotted line)

Table 1 Comparison between system parameters and the parameters identified at 250th generation

Parameter	a_1	a_2	a_3	b_1	b_2	b_3
True value	0.3	−0.4	0	0.8	0.7	0
Identified value	0.304 6	−0.398 4	0.005 2	0.805 6	0.703 1	0.007 1

Case 2: System with sudden parameter changes. In this case, we assume that system parameter a_2 suddenly changes from −0.4 to −0.7 and b_2 jumps from 0.7 to 0.4. Two identification methods (*i.e.* GA-based method and Least Square method) are used for comparison. The corresponding results are shown in **figures 6** and **7**. It can be seen clearly that the model identified by GA-based method has better dynamic response to the change than that with the Least Square method.

Case 3: System with gradual time-varying parameters. We assume that system parameter b_2 is time-varying. The simulation result shows that the GA-based identification method can trace satisfactorily the time-varying parameter (**figure 8**).

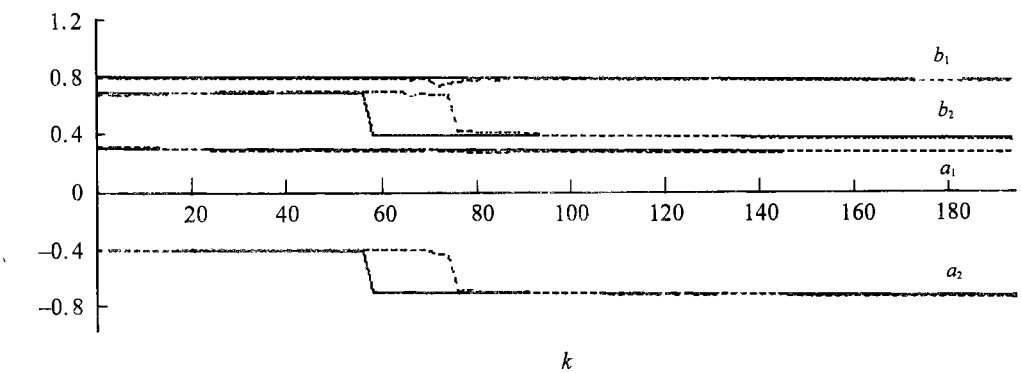

Figure 6 Identification process with GA-based method

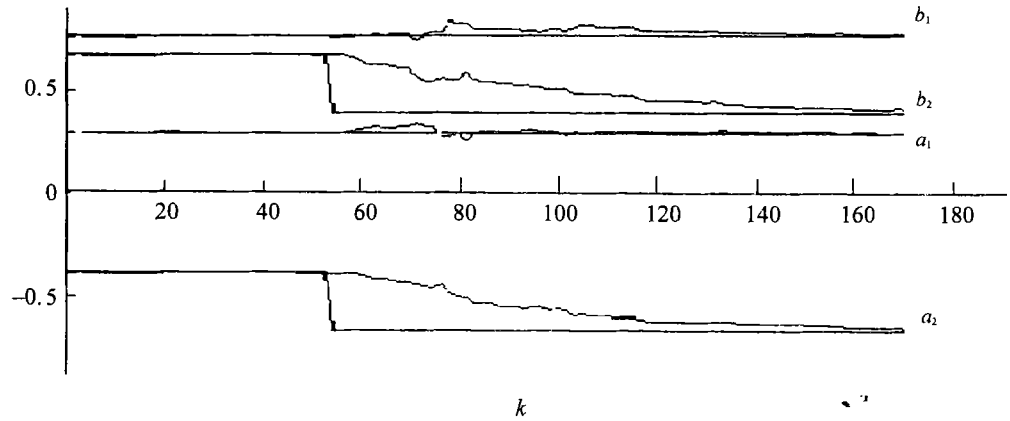

Figure 7 Identification process with least square method

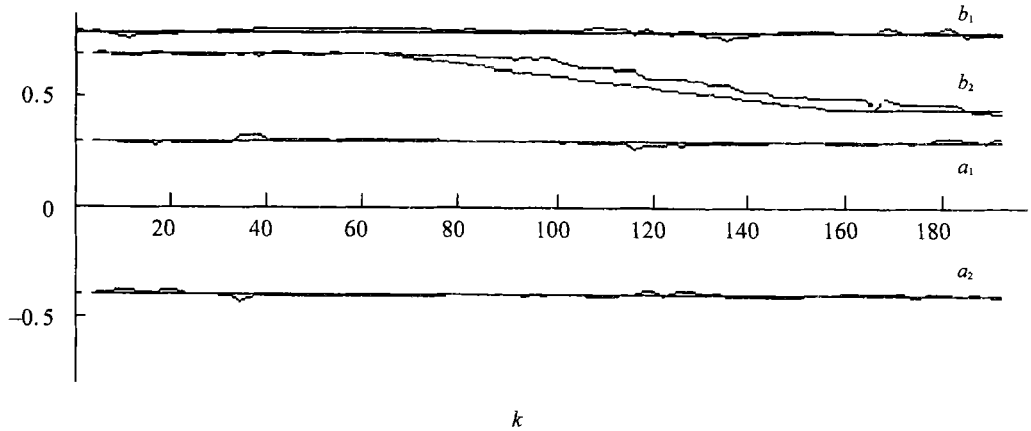

Figure 8 Identification process of the system with time-varying parameters

4 Conclusions

This paper has presented an implementation of GA-based identification. Some modifications have been made to improve the performance of the algorithm. The simulation results show that GA has great potential in system identification, which comes from its distinct characteristics such as less prior knowledge needed, robustness and adaptability. When it is applied to identification of the time-varying system, this method makes the model identified capable of following satisfactorily the changes in the system parameters.

Acknowledgment

The authors would like to thank the British Council for their ALCS project sponsorship (PEK\0992\290).

References

[1] Chengmin Ding, Chuansheng Zhang, Hui Liu: *Information and Control*, 26 (1997), No.1. p. 40.

[2] Liangjie Zhang, Yanda Li, Huimei Chen: *Acta Electronic Sinica*, 24 (1996), No.11. p. 6.

[3] Hitoshi Iba, Takio kurita, Hugo de Gares, Taisuke Sato: [In:] *Proceeding of Fifth International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers, USA, 1993.

[4] Stuart J. Flockton, Michael S. White: [In:] *Proceeding of Fifth International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers, USA, 1993.