Model structure selection for a discrete-time non-linear system using genetic algorithm

R Ahmad[1]***, H Jamaluddin**[1] and **M A Hussain**[2]
[1] Faculty of Mechanical Engineering, Universiti Teknologi Malaysia, Skudai, Johor, Malaysia
[2] Department of Chemical Engineering, Faculty of Engineering, Universiti Malaya, Kuala Lumpur, Malaysia

Abstract: In recent years, extensive works on genetic algorithms have been reported covering various applications. Genetic algorithms (GAs) have received significant interest from researchers and have been applied to various optimization problems. They offer many advantages such as global search characteristics, and this has led to the idea of using this programming method in modelling dynamic non-linear systems. In this paper, a methodology for model structure selection based on a genetic algorithm was developed and applied to non-linear discrete-time dynamic systems. First the effect of different combinations of GA operators on the performance of the model developed is studied. A proposed algorithm called modified GA, or MGA, is presented and a comparison between a simple GA and a modified GA is carried out. The performance of the proposed algorithm is also compared to the model developed using the orthogonal least squares (OLS) algorithm. The adequacy of the developed models is tested using one-step-ahead prediction and correlation-based model validation tests. The results show that the proposed algorithm can be employed as an algorithm to select the structure of the proposed model.

Keywords: model structure selection, system identification, evolutionary programming, genetic algorithms

NOTATION

a_i, b_i, c_i	coefficients of polynomial models
d	time delay
$e(t)$	random white noise
$F^l[.]$	a non-linear polynomial
l	degree of non-linearity
L	size of the regressor
M	number of terms of the regressors
n_y, n_u and n_c	output, input and noise lags
N	data length
p_c	crossover probability
p_m	mutation probability
$u(t)$	system input
$y(t)$	system output
$\hat{y}(t)$	one-step-ahead prediction
$\varepsilon(t)$	residual sequence
θ_i	unknown parameters
$\phi_i(t)$	regressors

1 INTRODUCTION

The identification of unknown dynamic systems has been studied and literature on system identification is extensive [**1**–**3**]. In system identification, a mathematical model is developed to simulate the actual system. System identification is then necessary to provide information for the analysis and design of an appropriate controller for a particular system. Procedures involved in system identification (SI) are the acquisition of data, definition of model structure, parameter estimation and model validation.

In conventional identification problems, a model structure is defined and the parameters of the model are estimated. The methods used in conventional algorithms, such as those based on least mean squares or maximum likelihood, estimate model parameters by optimizing the objective function based on gradient descent techniques. These methods suffer drawbacks, such as the solution becoming trapped in a local minima [**4**]. Since there is no particular algorithm being used to define the correct model structure, a trial-and-error method is adopted and the identification procedure becomes laborious in determining the correct model structure.

The MS was received on 3 April 2003 and was accepted after revision for publication on 8 October 2003.

* *Corresponding author: Faculty of Mechanical Engineering, Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia.*

Recently, evolutionary programming has been applied to various optimization problems [**5**] and offers many advantages, such as global search characteristics. One of these evolutionary programming approaches is the genetic algorithm (GA). Genetic algorithms were introduced by Holland [**6**] and extensively explored by Goldberg [**7**]. Such an algorithm evaluates multiple points in the solution space simultaneously and therefore has the potential to converge to the globally optimum solution [**7, 8**]. Some related works on the application of GA to system identification have been published [**9-14**]. Most of these assumed that the structures of the models were known and the genetic algorithm was then applied to estimate the parameters of those models, i.e. the GA is used as a parameter estimation algorithm.

The work by Kristinsson and Dumont [**9**] was considered as the first step in the direction of using a GA in system identification. They applied a GA to estimate parameters of both continuous and discrete time systems using an ARMAX (autoregressive moving average with exogenous inputs) model. They also showed that a GA could be used to identify physical parameters or poles and zeros. This work was then followed by several other researchers. Zibo and Naghdy [**10**] applied GAs not only to single-input–single-output (SISO) systems but also to multi-input–multi-output (MIMO) systems. Tan *et al.* [**11**] and Jeong and Lee [**12**] fine-tuned GAs with simulated annealing. The ARMAX model was widely used as a candidate model in their studies [**9-12**]. Sheta and De Jong [**13**] explored GAs as the key search procedures for parameter estimation and particular focus was on noisy environments and autonomous systems with varying initial conditions. Hong and Billings [**14**] introduced a new parameter-estimation algorithm based on stacked regression and evolutionary algorithms while the parsimony of the model structure was determined using the forward orthogonal least squares (OLS) algorithm.

Another important step in system identification is model structure selection and various approaches have been proposed. Haber and Unbehauen [**15**] reviewed several structure identification algorithms based on different classes of models such as block-oriented models, cascade models, semi-linear models with signal-dependent parameters, linear-in-parameter models and a group method of data handling (GMDH). Desrochers and Mohseni [**16**] proposed an algorithm based on projection matrices to model non-linear systems using a GMDH. Veres [**17**] defined model structure selection as a procedure of estimating the orders of lag y, u, e and time delay d (ARMAX systems) on the basis of the knowledge of the input and output sequences $u(t)$ and $y(t)$ respectively, where $e(t)$ is the noise sequence. The main task in model structure selection is to determine the significant terms to be included in the selected model. Among the techniques that have been developed are OLS [**18**] and genetic algorithms [**19**]. The OLS algorithm combines structure selection and parameter estimation and was shown to be much more efficient than ordinary least squares based algorithms [**20**]. The algorithm provides information regarding which terms in the model are significant using its by-product, called the error reduction ratio (ERR). However, the problem encountered using the OLS for the identification of an unknown model structure is the difficulty in deciding the ERR value at which to stop.

In this study, a genetic algorithm is used to determine the model structure of linear and non-linear systems that best represent the system from input–output data. The structure of the system considered in this study is obtained from difference equations of discrete-time systems. The performance of the GA with varying genetic operators is investigated to study the effect of those operators on the performance of the GA in system identification problems. An improved strategy called a modified GA (MGA) is proposed as the algorithm to select model structure. The model parameters are estimated using the least squares method. The identification results are compared with those from the simple GA (SGA) method and the OLS method. The parameters used were kept the same throughout for a fair comparison. The result of the study has shown that the proposed algorithm gives an efficient way of modelling structure selection and the modelling accuracy is as good or better when compared with the OLS method.

The paper is organized as follows. Section 2 reviews model structure selection in system identification including the OLS method. Section 3 reviews GAs and their application to model structure selection. Section 4 shows simulation results that include the effect of different GA operators on the applications, the comparison between the SGA and MGA and the comparison between the proposed algorithm and the OLS method. The conclusion summarizes the main contributions of the paper.

2 MODEL STRUCTURE SELECTION

A very common model structure for discrete systems used in control applications is the ARMAX model, where the system output can be predicted using the past input and output lags of the system. The ARMAX model is defined as [**21**]

$$\begin{aligned} y(t) + a_1 y(t-1) + \cdots + a_{n_y} y(t-n_y) \\ = b_1 u(t-1) + \cdots + b_{n_u} u(t-n_u) \\ + e(t) + c_1 e(t-1) + \cdots + c_{n_c} e(t-n_c) \end{aligned} \tag{1}$$

where $e(t)$ is a noise sequence with zero mean, $u(t)$ and $y(t)$ are the system input and output respectively while n_y, n_u and n_c are the orders of output, input and noise lags respectively.

mi00003103 15-01-04 11:52:56 Rev 14.05
The Charlesworth Group, Huddersfield 01484 517077

To represent the discrete non-linear system, the NARMAX model was introduced [22]

$$y(t) = F^l(y(t-1), \ldots, y(t-n_y), u(t-1), \ldots, u(t-n_u)) + e(t)l \qquad (2)$$

By defining $F^l(.)$ as a non-linear polynomial with l degree of non-linearity, the model belongs to the class of linear regression models

$$y(t) = \sum_{i=1}^{M} \theta_i \phi_i(t) + e(t), \qquad \leqslant t \leqslant N \qquad (3)$$

where $\phi_i(t)$ and θ_i are non-linear regressors and unknown parameters respectively, M is the number of terms of the regressors and N is the data length.

The number of possible terms in the ARMAX model is equal to the sum of allowed maximum lags, while the maximum number of terms in the NARMAX model of equation (2) is [20]

$$L = M + 1$$

where

$$M = \sum_{i=1}^{l} n_i, \qquad \text{for } l = \text{order of non-linearity}$$

and

$$n_i = \frac{n_{i-1}(n_y + n_u + i - 1)}{i}, \qquad n_0 = 1 \qquad (4)$$

A discrete-time system with $n_y = n_u = 2$, if expanded as a second degree of non-linearity model ($l = 2$), would contain 14 possible terms. In an NARMAX model the number of terms is extremely large; therefore it is important to select the significant terms for the model that will produce a parsimonious model that adequately represents the data set.

One method proposed for structure selection is to use the OLS method [18]. In OLS, the contributions of each term in the model are measured based on an error reduction ratio (ERR). The simple derivation of the ERR is calculated from the equation [18]

$$\text{ERR}_i = \frac{g_i^2 E[w_i^2(t)]}{E[y^2(t)]} \qquad (5)$$

where g_i is the coefficient of parameter estimates and $w_i(t)$ the orthogonal data set. However, it is found that the value of the ERR depends on the order of the terms in the regression equation, which sometimes may produce incorrect information for the significant terms [23]. These problems can be solved using the forward regression OLS [24]; however, the computation would be more expensive. Furthermore, the user needs to decide the ERR value but there is no specific criteria for the value, as discussed later in section 4.3.

The order of the input and output lags and the order of non-linearity chosen for the system also affect the model structure selection. Larger non-linearities will produce a more complex model and the search space would become very large and impractical. However, some of these terms can be excluded to simplify the structure detection and parameter estimation in order to produce a smaller model; therefore, a model with an accurate result can be determined. For example, for a system of $n_y = n_u = 3$ and $n_l = 2$, the complete polynomial has 27 terms. To determine the coefficients of each model structure combinations by solving normal calculations requires 134 217 726 of searching space. Employing the principle of survival of the fittest in the GA, this searching technique can avoid calculating all possible models and is capable of retaining a few significant terms, while at the same time maintaining the accuracy of the system. This leads to the superiority of the GA for model structure determination.

3 GENETIC ALGORITHMS FOR MODEL STRUCTURE SELECTION

A genetic algorithm is a search procedure that imitates the principle of natural evolution. It searches from a population of points or individuals where each individual represents a potential solution. These individuals, called chromosomes, evolve through the action of operators such as selection, crossover and mutation. In this study a genetic algorithm is used to determine the model structure of a system that best represents the system. The outline of an SGA is summarized in Fig. 1.

3.1 Model structure selection and parameter estimation

A genetic algorithm is used to select the model structure. A chromosome representing a potential solution is defined. The length of the chromosome, L, is equal to the number of regressors in the model. A chromosome is made up of genes. Each gene will indicate whether a regressor is to be included in the model. There are $2^L - 1$ possible models for selection and for each model the representation is expressed by an L-bit binary code of chromosome $\mathbf{c}$. By defining the initial population composed of a number of chromosomes, through the process of evolution a better population will be produced. Since the models are linear-in-the-parameter forms, the identified parameters are estimated using the least squares estimation algorithm [21].

3.2 GA control parameters

The control parameters of the GA include population size, crossover and mutation probability, and crossover and selection strategy. The choices of these parameters

```
Initialise parameters (e.g. population size, maximum generation) in GA
Generate initial random population
For generation := 1 to maximum generation
          Compute the fitness of each individual in the population
          While number of individual < population size
                Perform selection, crossover and mutation
                Place offspring to new population
          End
          Replace old population with new population
End
```

Fig. 1 Outline of a simple genetic algorithm

can affect the behaviour and performance of the GA and have been studied by researchers such as Grefenstette [**25**], Michalewicz [**8**] and Qingsheng *et al.* [**26**]. The choice of an appropriate population size is particularly important. If the population size is too small, the GA will converge too quickly and in many cases will cause premature convergence, while if the population size is too large, it will take a longer time to converge. Theoretically, it might be concluded that a larger population size will produce a better search and therefore will result in a better solution.

The mutation and crossover operators are the sources of exploration in the GA, and Grefenstette [**25**] discussed the probability rate for these operators. The individuals in a population are called chromosomes and are made up of genes. The different values of individual genes are called alleles. A schemata is a similarity template describing a set of chromosomes that match each other at certain positions. Crossover is an information exchange between chromosomes while mutation is applied to introduce a random change into the population. Disruption of schemata occurs when a crossover point between two alleles that are critical members of a certain solution is chosen. Variations of crossover techniques are used to reduce the probability of disruption due to crossover. Some researchers reported other types of crossover, such as Eshelman *et al.* [**27**], who experimented with two-point crossover, and segmented and shuffle crossover, and Syswerda [**28**] and Yuping *et al.* [**29**], who tried with uniform crossover.

Holland [**6**] and Goldberg [**7**] emphasized the fact that mutation is also important and serves as a background operator to support a crossover operator by ensuring that all the allele values are accessible to the search. A relatively small value of p_m ($p_\mathrm{m} \in [0.001, 0.01]$) is recommended for a canonical GA [**25**]. Recently, empirical and theoretical investigations have shown the benefits of emphasizing mutation as a search operator [**30**].

There are limitations encountered when using the SGA. The convergence is too rapid and the so-called superior chromosomes dominate the population. Therefore, the population diversity decreases due to the increase in selective pressure. In order to find a balance between population diversity and selective pressure, some modifications have been made, particularly in the selection procedure to the algorithm based on concepts found in the literature [**31**, **32**], which is called the modified genetic algorithm, or MGA. However, some modifications were made. To maintain the diversity of the population, ordinary chromosomes undergo a normal GA procedure and the worst performing individual does not reproduced but is replaced with new individuals. There is the possibility that the good individuals are prevented from reproduction, which results in poor exploration, and can therefore lead to a premature convergence. To preserve the potential solutions and the loss of critical alleles, the best chromosomes are selected for small alterations.

3.3 The modified algorithm

The flowchart of the model structure selection formulation using the MGA is shown in Fig. 2. The difference between the SGA and the MGA is in the selection procedure. After every generation, the individuals are categorized into three different groups based on their fitness values. Each group will undergo different recombination procedures.

Details of the procedures are as follows:

1. Representation. The individual or chromosome represents the model structure, and therefore each bit of the chromosome represents the term for the regressor in the equation. Bit string encoding is used because of its simplicity and traceability for this

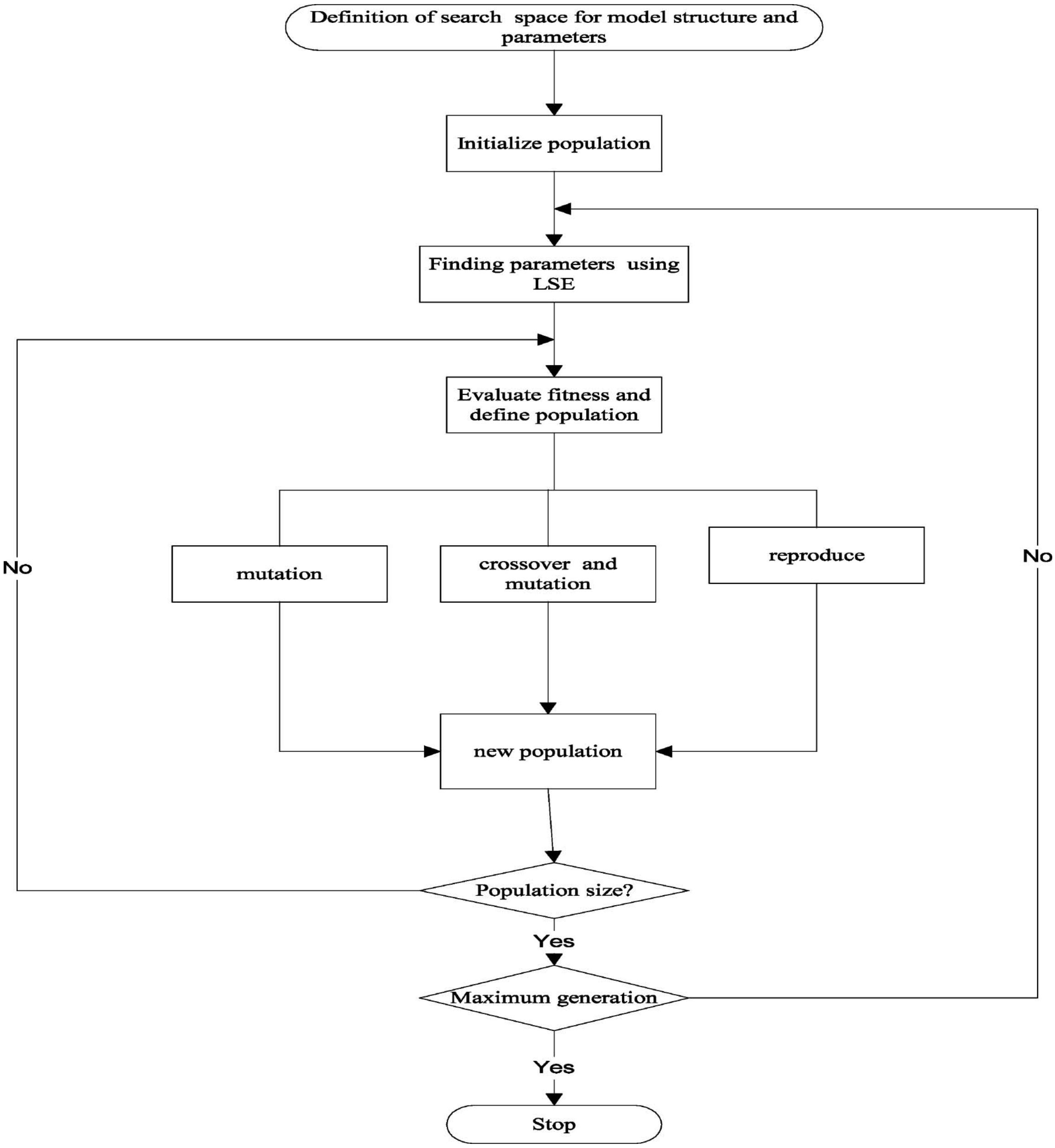
Definition of search space for model structure and parameters
Initialize population
Finding parameters using LSE
Evaluate fitness and define population
mutation
crossover and mutation
reproduce
new population
Population size?
No
Yes
Maximum generation
No
Yes
Stop

Fig. 2 Modified genetic algorithm for model structure selection

application. It consists of an L-bit binary code, where the length of L equals the size of the regressor. The maximum number of terms in the ARMAX model is equal to the input and output lags and the maximum number of terms in the NARMAX model is as in equations (4).

2. Create individuals. Initially, m individuals are created at random. Each individual that represents a possible model is expressed by the L-bit binary model code **c**. Binary bits represent genes. If some bits of the binary model of chromosome **c** equal zero, the terms are excluded from the model. For example, in an ARMAX model as in equation (1) for $n_y = n_u = 5$, the equation would become

$$\begin{aligned} y(t) + a_1 y(t-1) + a^2 y(t-2) + a_3 y(t-3) \\ + a_4 y(t-4) + a_5 y(t-5) \\ = b_1 u(t-1) + b_2 u(t-2) + b_3 u(t-3) \\ + b_4 u(t-4) + b_5 u(t-5) \end{aligned}$$

where L equals $n_y + n_u = 10$ and therefore there are 1024 possible models to be selected. Let the 10-bit binary model code be $c_1 = [1001001010]$; then the model can be expressed as

$$y(t) = a_1 y(t-1) + a_4 y(t-4) + b_2 u(t-2) + b_4 u(t-4)$$

For the NARMAX model, as in equation (2) with $n_y = n_u = n_l = 2$, there are 14 possible terms with 16 383 possible models and the equation for the full model is

$$\begin{aligned} y(t) = {} & a_1 y(t-1) + a_2 y(t-2) + a_3 u(t-1) \\ & + a_4 u(t-2) + a_5 y^2(t-1) + a_6 y(t-1) y(t-2) \\ & + a_7 y(t-1) u(t-1) + a_8 y(t-1) u(t-2) \\ & + a_9 y^2(t-2) + a_{10} y(t-2) u(t-1) \\ & + a_{11} y(t-2) u(t-2) + a_{12} u^2(t-1) \\ & + a_{13} u(t-1) u(t-2) + a_{14} u^2(t-2) \end{aligned}$$

If the binary model code is $c_2 = [10110011000100]$, the model can be expressed as

$$\begin{aligned} y(t) = {} & a_1 y(t-1) + a_3 u(t-1) + a_4 u(t-2) \\ & + a_7 y(t-1) u(t-1) + a_7 y(t-1) u(t-1) \\ & + a_{12} u^2(t-1) \end{aligned}$$

3. Parameter estimation θ_i. The selected term for the model is identified by the value 1 for each chromosome. Based on the identified model structure, the value of the parameters θ_i are calculated using the least squares estimation method. The values of parameters to be calculated for the above example are a_1, a_4, b_2 and b_4 respectively.
4. Fitness function. The objective of the algorithm is to minimize the error between the model and the system. In a genetic algorithm, the fitness function is defined as the quality of the model. The fitness function used in this study is the adaptive fitness function [**10**], where for every new generation, a positive scaling value (SV) is found before the objective function (OF) is calculated. The fitness function is defined as

   $$\text{Fitness function} = \text{SV} - \text{OF} \tag{6}$$

   where SV is a positive scaling value of the objective function and OF is defined by

   $$\text{OF} = \sum_{i}^{N} [y(t) - \hat{y}(t)]^2 \tag{7}$$

   where

   $$\hat{y}(t) = \sum_{i=1}^{M} \phi_i(t) \hat{\theta}_i \tag{8}$$

   is the one-step-ahead prediction of the model output using the selected terms and estimated parameters as determined in step 3.
5. Selection. Based on the fitness function, a new population is generated by dividing the chromosomes into three categories, namely the good, the second best and the bad chromosomes, based on the value of the objective function. The first and the second groups are selected and will undergo mutation and crossover. Solutions with a large sum-squared error falling into the third category are eliminated from the population.
6. Mutation. The good chromosomes have to undergo a mutation process in order to prevent schemata disruption as well as to preserve these potential populations. For a binary coded string, mutation is a process of replacing a bit of value 1 with 0 and vice versa, with a specified mutation rate of p_{m}.
7. Crossover. The second best chromosomes will have to undergo both crossover and mutation processes. In crossover, two chromosomes are selected from the mating pool and information between these individuals (parents) is exchanged to produce two offspring with a probability of p_{c}.
8. After a stopping criteria has been met (a specified number of generations), the final model will be selected among all the possible models based on the highest fitness value.

3.4 Model validity test

Model validation is the final procedure in system identification. The objective is to check whether the model fits the data adequately without any bias. The model validity test employed in this study includes the correlation test [**33**] based on the following conditions:

$$\begin{aligned} \phi_{\varepsilon\varepsilon}(\tau) &= \frac{E[\varepsilon(t)\varepsilon(t-\tau)]}{E[\varepsilon^2(t)]} = \delta(\tau), && \tau = 0 \\ \phi_{u\varepsilon}(\tau) &= \frac{E[u(\tau)\varepsilon(t-\tau)]}{\sqrt{E[u^2(\tau)\varepsilon^2(t)]}} = 0 && \text{for all } \tau \\ \phi_{\varepsilon(\varepsilon u)}(\tau) &= \frac{E[\varepsilon(t)\varepsilon(t-1-\tau)u(t-1-\tau)]}{\sqrt{E[\varepsilon^2(t)]E[\varepsilon^2(t)u^2(t)]}} = 0, && \tau \geqslant 0 \\ \phi_{u^{2\prime}\varepsilon}(\tau) &= \frac{E[(u^2(t) - \overline{u^2})\varepsilon(t-\tau)]}{\sqrt{E[(u^2(t) - \overline{u^2})^2]E[\varepsilon^2(t)]}} = 0, && \text{for all } \tau \\ \phi_{u^{2\prime}\varepsilon^2}(\tau) &= \frac{E[(u^2(t) - \overline{u^2})\varepsilon^2(t)]}{\sqrt{E[(u^2(t) - \overline{u^2})^2]E[\varepsilon^4(t)]}} = 0, && \text{for all } \tau \end{aligned} \tag{9}$$

where ϕ represents the standard correlation function, $\varepsilon(t)$ represents the residual sequence

$$\varepsilon = \hat{y} - y$$

and the overbar denotes the time average and is given as

$$\overline{u^2} = \frac{1}{N} \sum_{t=1}^{N} u^2(t)$$

The tests are able to indicate adequacy of the fitted model. Generally, if the correlation functions are within the 95 per cent confidence interval, i.e. $\pm 1.96\sqrt{N}$, the model is regarded as adequate where N is the number of data points. Another test includes the one-step-ahead (OSA) prediction given by

$$\hat{y}(t) = \hat{f}(y(t-1), \ldots, y(t-n_y), u(t-1), \ldots, u(t-n_u)) \tag{10}$$

where the predicted output is based on the previous input and output data. The performance of the model is also evaluated using an error index (EI), defined as

$$\mathrm{EI} = \sqrt{\frac{\sum [y(t) - \hat{y}(t)]^2}{\sum y^2(t)}} \tag{11}$$

4 SIMULATION STUDY

Based on the methodology discussed in the previous sections, simulation studies were conducted. The first part of the study investigated the effect of GA control parameters such as population size, crossover and mutation probabilities and the crossover mechanism used in the proposed algorithm as a benchmark for system identification problems. The performance of the algorithm based on those different values of GA parameters was studied. A wide range of values for population size as well as crossover and mutation probabilities was used. The goal of this study is to examine and compare these various strategies and help to obtain a better understanding of the conditions under which these operators are effective. The second part of the study compared the SGA and the MGA. In order to evaluate the performance of the MGA, four simulated non-linear systems were used for both cases and the effect on the convergence rate was investigated. The third part of the study is the comparison between the MGA and OLS with application to discrete-time non-linear systems identification. The model structures identified by those two methods were compared with respect to the accuracy of the models as well as the identified model structures determined by those two methods.

4.1 Genetic algorithm control parameters

As discussed previously, the performance of a GA is greatly influenced by the genetic operators. In this section, the effect of the performance of the proposed algorithm or MGA based on different values of GA parameters is investigated to understand better how it works. The control parameters used in the study are:

1. Population size. Population sizes of 5, 10, 50, 100 and 200 were used.
2. Crossover mechanism. Single crossover and double crossover.
3. Crossover probabilities. Probabilities of 0.05, 0.6 and 0.95 were used.
4. Mutation probabilities. Probabilities of 0.001, 0.01 and 0.1 were used.

The experimental studies included a wide range of values for these GA control parameters to bracket the best MGA performance.

The systems that are investigated are based on the following equations [**34**]

Model 1:

$$\begin{aligned} y(t) = {} & 0.954y(t-1) - 0.222y(t-3) + 0.123y(t-6) \\ & + 0.456u(t-1) + 0.096u(t-2) \\ & - 0.052y^2(t-1) + 0.231y(t-1)u(t-2) \\ & + 0.321u(t-2)u(t-3) + e(t) \end{aligned}$$

Model 2:

$$\begin{aligned} y(t) = {} & 0.797y(t-1) - 0.237y(t-3) + 0.441u(t-1) \\ & + 0.105y(t-4)u(t-4) + 0.333u(t-3)u(t-5) \\ & + e(t) \end{aligned}$$

where $u(t)$ is a zero mean random sequence and $e(t)$ is a random white noise. These models are selected since they have a large number of possible terms giving a chance of studying the effects of varying GA operators. In this study, 1000 data points, N, were generated. Models 1 and 2 are non-linear systems with an NARMAX model structure. The values of n_y, n_u, and n_l are 6, 3 and 2 respectively for model 1 and 4, 5 and 2 respectively for model 2. The total number of possible terms is 54; therefore there are a total of $10^{54} - 1$ possible forms or model structures. The number of generations used was 100, p_c was 0.6 and p_m was 0.01. Table 1 gives a comparison of the values of EI produced by the final model selected by the GA using different population sizes (n) in models 1 and 2 respectively.

The results in Table 1 showed that the lowest EI for model 1 is $n = 10$ and model 2 is $n = 50$ for generations up to 100. The EI values are larger after $n = 10$ for model 1 and $n = 50$ for model 2 because larger populations will need more generation for convergence. The results above indicate that the performance of the algorithm improves when the population size is increased. For a small

Table 1 Effect of population size on error index (EI)

Population size	EI for model 1 ($\times 10^{-2}$)	EI for model 2 ($\times 10^{-2}$)
5	2.425353	4.447845
10	1.272846	4.379760
50	1.372062	2.559774
100	1.527128	4.132735
200	1.774084	5.209113

mi00003103 15-01-04 11:52:56 Rev 14.05
The Charlesworth Group, Huddersfield 01484 517077

number of populations, most of the highly superior individuals will dominate the population towards a later generation, giving them the chance of being selected. However, the disadvantage is that there will be a chance of not selecting the better solution. On the other hand, a larger population gives more diversity in the population and better search, but requires more evaluation per iterations and takes a longer time to converge. Therefore, other GA operators such as crossover and mutation also play an important role for overall performance of the GA.

In the next simulation, by varying the crossover and mutation rates, the convergence of the algorithms is observed. The population size is set 30 and the number of generations is set 100. The effect on different crossover strategies is also included. These results are shown in Tables 2 and 3 and Figs 3, 4 and 5 respectively.

Table 2 Values of EI for varied crossover and mutation rates (model 1)

	$p_c = 0.05$	$p_c = 0.6$	$p_c = 0.95$
$p_m = 0.001$	0.0246	0.0127	0.0176
$p_m = 0.01$	0.0351	0.0147	0.0152
$p_m = 0.1$	0.0312	0.0199	0.0169

Table 3 Values of EI for varied crossover and mutation rates (model 2)

	$p_c = 0.05$	$p_c = 0.6$	$p_c = 0.95$
$p_m = 0.001$	0.0211	0.0426	0.0473
$p_m = 0.01$	0.0139	0.0318	0.0274
$p_m = 0.1$	0.0385	0.0440	0.0271

For model 1, a less significant effect is seen for a varied mutation rate at a very low crossover rate. The effect of a varied mutation rate can be seen at $p_c = 0.6$, as shown in Fig. 3b and the best performance can be seen at $p_m = 0.01$, with EI equal to 0.0127. For model 2, the best performance can be seen at $p_c = 0.05$ and $p_m = 0.01$ with the value of EI of 0.0139. In this simulation, a generalization cannot be made due to different results that have been obtained. These results indicate that there are no specific rules for optimal values for p_c and p_m and the optimal performance that can be achieved depends on the models to be studied. However, varying the crossover strategy had some effect on the performance of the algorithm for both cases in the examples and these effects are shown in Fig. 5. Using a probability rate of $p_c = 0.6$ and a mutation rate of $p_m = 0.01$, a double crossover strategy will give faster convergence for both model 1 and model 2.

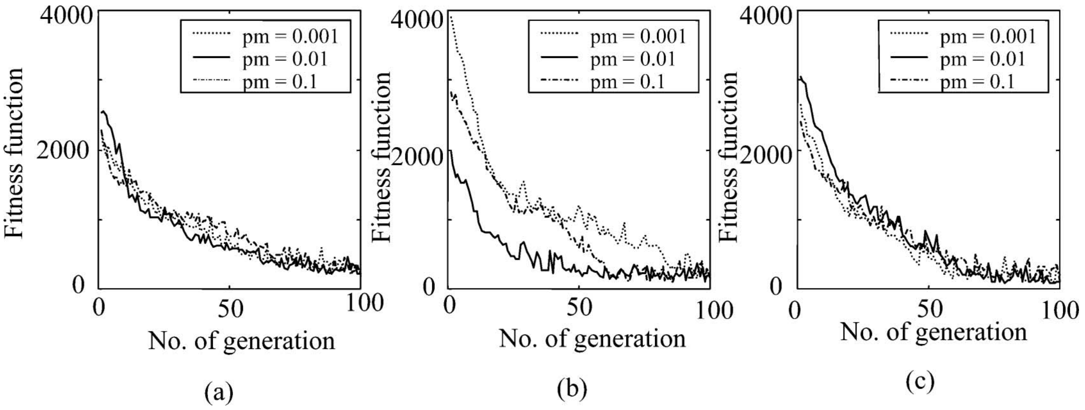

Fig. 3 Simulation results for model 1 on the effect of varied crossover and mutation rates: (a) $p_c = 0.05$, varied p_m; (b) $p_c = 0.6$, varied p_m; (c) $p_c = 0.95$, varied p_m

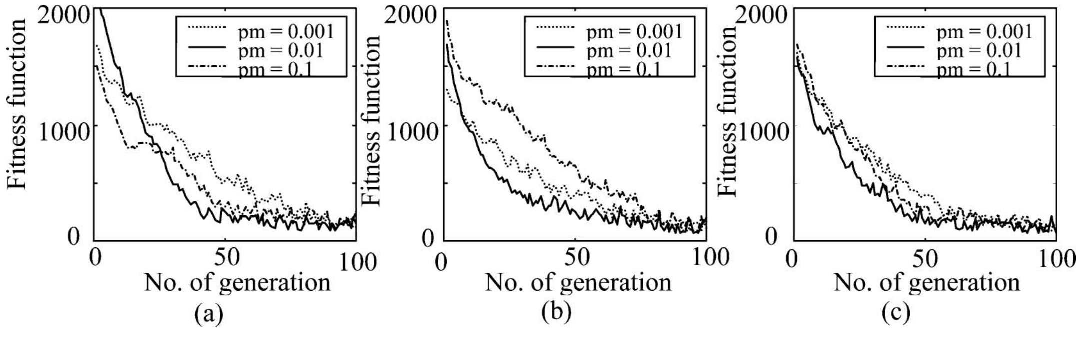

Fig. 4 Simulation results for model 2 on the effect of crossover and mutation rates: (a) $p_c = 0.05$, varied p_m; (b) $p_c = 0.6$, varied p_m; (c) $p_c = 0.95$, varied p_m

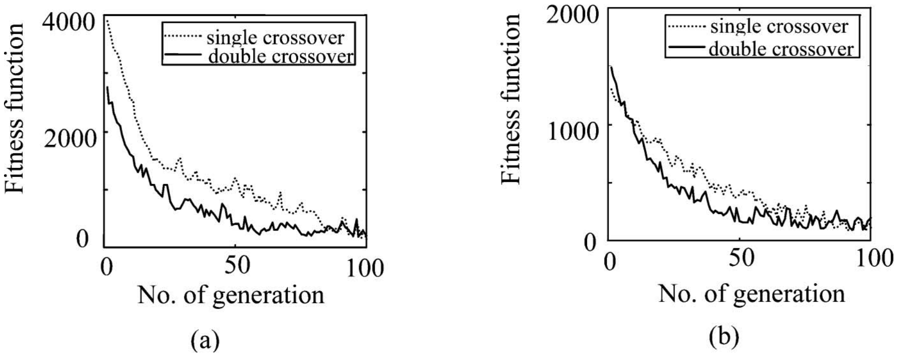

Fig. 5 Single crossover versus double crossover for (a) model 1 and (b) model 2

4.2 The comparison between the SGA and the MGA

The comparison between SGA and MGA is investigated with linear and non-linear discrete-time models. The models investigated were based on the following equations:

Model 3:

$$y(t) = 1.5y(t-1) - 0.7y(t-2) + u(t-5) + 0.5u(t-6) + e(t)$$

Model 4:

$$y(t) = 0.5y(t-1) + u(t-2) + 0.1u^2(t-1) + e(t)$$

Model 5:

$$y(t) = 0.5y(t-1) + u(t-2) + 0.2u(t-3) - 0.1u^2(t-1) + e(t)$$

Model 6:

$$y(t) = 0.8y(t-1) + 0.5u^2(t-1)y(t-1) + u^3(t-1)$$

where $u(t)$ is a zero mean random sequence, $e(t)$ is random white noise and N equals 500. For model 3, the system is a linear system using an ARMAX model representation with values of n_y and n_u being 2 and 6. For a non-linear system with the NARMAX model representation, the values for model 4 are $n_y = 1$, $n_u = 2$ and $n_l = 2$, the values for model 5 are $n_y = 1$, $n_u = 3$ and $n_l = 2$ and for model 6 are $n_y = 1$, $n_u = 1$ and $n_l = 3$. These linear and non-linear models were used in this study in order to investigate the efficiency of the GA in selecting the structure for a wider range of system. The results are shown in Figs 6 and 7 respectively. Similar trends of the convergence rate for both the SGA and the MGA are observed from these figures where for all cases the SGA has a faster convergence rate than the MGA; however the MGA produces smaller variations towards later generations.

A faster convergence can be seen using the SGA for these models as compared with the MGA. Larger variations in the value of the fitness function over many generations suggest that it is good for exploring the potential region for the solution, but still lacks in finding the optimum solution. In a few generations, the SGA has located the possible region of the solution, but is unable to converge to a global solution. For a similar number of generations using the MGA, the convergence curve looks more stable and smooth and appears to be more promising in finding solutions with lower costs.

4.3 Model structure selection using the OLS and the MGA

In this section, the adequacy of model structures of non-linear systems selected and fitted by two algorithms, the OLS and the MGA, are compared. For this purpose, the algorithms are tested for two different non-linear

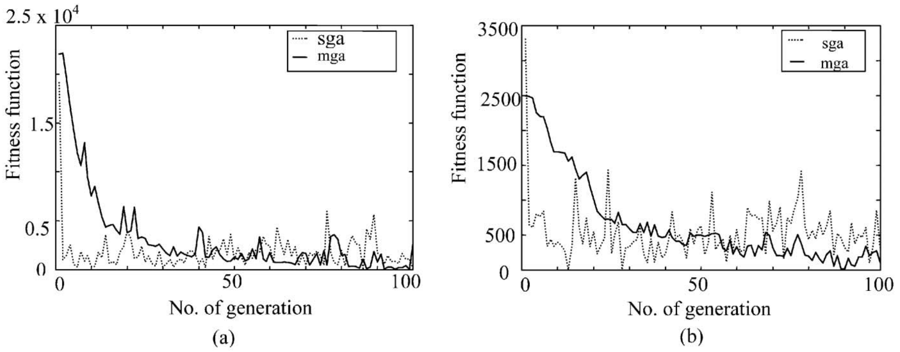

Fig. 6 Convergence of the SGA and MGA for (a) model 3 and (b) model 4

mi00003103 15-01-04 11:52:56 Rev 14.05
The Charlesworth Group, Huddersfield 01484 517077

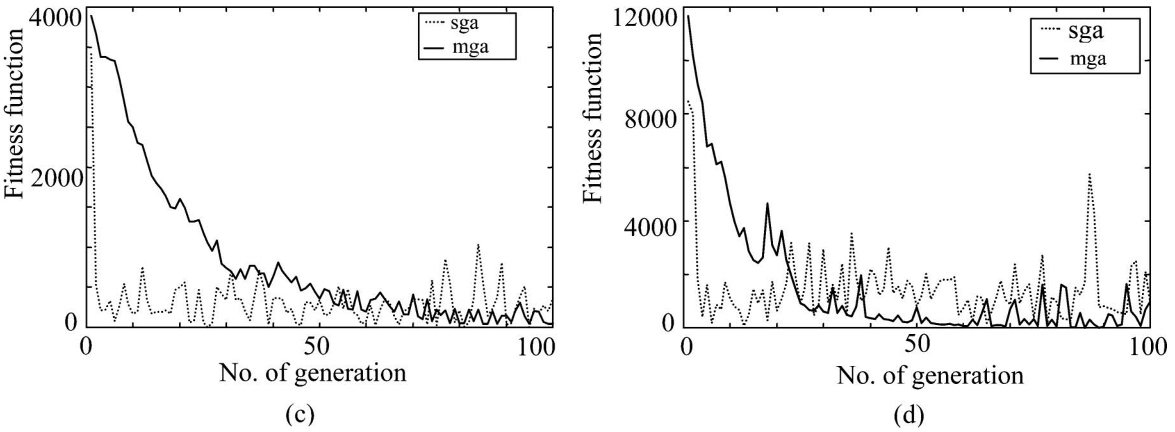

Fig. 7 Convergence of the SGA and MGA for (a) model 5 and (b) model 6

dynamic systems. The seventh example is a simulated system governed by the difference equation

$$y(t+1) = \frac{0.3y(t) + 0.6y(t-1) + 0.6\sin[\pi u(t)] + 0.4\sin[3\pi u(t)]}{5.5}$$

where the input $u(t)$ is chosen to be

$$u(t) = \sin\left(\frac{2\pi t}{250}\right)$$

with 500 pairs of input–output data. The input and output lags chosen for this model are $n_u = 2$ and $n_y = 3$, the non-linearity l is equal to 2, the maximum number of model terms is 20 and there are 1 048 575 possible models to be selected from.

The eighth example is based on input–output data originating from Box and Jenkins [**35**] concerning the identification of a gas oven. There are 296 pairs of input–output data. The input $u(t)$ of the plant is the methane gas flowrate into the furnace and the output $y(t)$ is the %CO$_2$ concentration in the outlet gas with the sampling interval equal to 9 s. In this example, the MGA is used to identify a non-linear system based on an NARMAX model structure with $n_y = 2$, $n_u = 2$ and the non-linearity l is equal to 2. The maximum number of model terms is 14 and there are 16 384 possible models to be selected from.

The parameters used in the algorithms are 50, 0.6 and 0.01 for population size, p_c and p_m respectively. The values of EI for both MGA and OLS methods are then compared. The results for structure selection using the OLS and MGA are shown in Tables 4, 5, 6 and 7. The seventh example is a simulated example and the eighth example is a real experimental data. Both of these examples are not NARMAX model representations and are fitted using NARMAX models. The purpose of this study is to investigate whether the MGA would yield more compact model structures than the OLS.

Table 4 Estimated model using the OLS for model 7

Terms	Estimate	ERR$_i$
$y(t-1)$	7.1450×10^{-1}	9.829×10^{-1}
$y(t-2)$	3.2473×10^{-1}	1.140×10^{-3}
$u(t-1)$	4.1198×10^{-1}	5.006×10^{-4}
$u(t-2)$	-3.8389×10^{-1}	1.769×10^{-4}
$y(t-3)$	-6.7701×10^{-2}	7.214×10^{-5}
$y(t-3)y(t-1)$	-1.1087×10^{-1}	2.313×10^{-5}
$u(t-2)u(t-1)$	-1.4331×10	2.099×10^{-5}
$u(t-2)u(t-2)$	1.8335×10	1.958×10^{-5}
$u(t-2)y(t-1)$	8.4943×10^{-2}	1.498×10^{-5}
$y(t-2)y(t-1)$	2.0637×10^{-1}	7.591×10^{-6}
$y(t-3)u(t-2)$	-2.1958×10	4.656×10^{-6}
$y(t-3)y(t-3)$	6.7980×10^{-2}	3.354×10^{-6}
$y(t-3)y(t-2)$	6.4252×10^{-2}	3.060×10^{-6}
$y^2(t-1)$	-2.9870×10^{-2}	2.993×10^{-6}
$y(t-1)u(t-1)$	-8.0519×10^{-2}	2.134×10^{-6}
$y(t-2)y(t-2)$	-1.3626×10^{-1}	1.028×10^{-6}
$y(t-2)u(t-1)$	3.4280×10^{-1}	6.984×10^{-7}
$y(t-3)u(t-1)$	2.1021×10	1.829×10^{-7}
$y(t-2)u(t-2)$	-3.2076×10^{-1}	1.526×10^{-7}
$u(t-1)u(t-1)$	-3.8951×10^{-1}	9.459×10^{-9}

Table 5 Estimated model using the GA for model 7

Terms	Estimate
$y(t-1)$	7.072627×10^{-1}
$y(t-2)$	2.557947×10^{-1}
$u(t-2)$	3.633663×10^{-2}
$y^2(t-1)$	2.863361×10^{-2}
$y(t-1)y(t-2)$	-3.614872×10^{-2}
$y(t-1)u(t-1)$	2.218746×10^{-2}
$y(t-2)u(t-2)$	3.277503×10^{-2}
$y(t-3)u(t-2)$	-3.759731×10^{-2}
$u(t-2)u(t-1)$	-8.189368×10^{-3}

For model 7, the EI produced by the GA is 0.1237 as compared with the EI produced by the OLS of 0.1245, as given in Table 8. However, the number of terms for the GA is less, which is 9 compared with the OLS which

Table 6 Estimated model using the OLS for model 8

Terms	Estimate	ERR$_i$
$y(t-1)$	-5.6095×10	9.998×10^{-1}
$y(t-2)$	6.7452×10	1.339×10^{-4}
$u(t-2)$	-2.2006×10	2.642×10^{-5}
$y(t-2)y(t-1)$	-8.4325×10^{-2}	9.156×10^{-6}
$y(t-1)u(t-1)$	4.2536×10^{-1}	1.468×10^{-6}
$u^2(t-2)$	-1.8162×10^{-1}	1.132×10^{-6}
$y(t-2)u(t-1)$	-3.8277×10^{-1}	9.714×10^{-7}
$u^2(t-1)$	-1.7146×10^{-1}	8.287×10^{-7}
$y^2(t-2)$	-2.6074×10^{-2}	6.472×10^{-7}
$y^2(t-1)$	1.0787×10^{-1}	5.024×10^{-7}
$u(t-1)u(t-2)$	5.2008×10^{-1}	3.942×10^{-7}
$u(t-1)$	-1.8454×10	3.275×10^{-7}
$y(t-2)u(t-2)$	6.5491×10^{-2}	1.789×10^{-7}
$y(t-1)u(t-2)$	-3.9876×10^{-2}	1.613×10^{-7}

Table 7 Estimated model using the GA for model 8

Terms	Estimate
$y(t-1)$	1.055281×10
$y(t-2)$	7.797739×10^{-2}
$u(t-1)$	-8.263638×10^{-1}
$y^2(t-1)$	4.842900×10^{-2}
$y(t-1)y(t-2)$	-8.960925×10^{-2}
$y(t-1)u(t-1)$	2.050143×10^{-2}
$y(t-1)\ u(t-2)$	5.958990×10^{-2}
$y^2(t-2)$	3.868690×10^{-2}
$y(t-2)u(t-2)$	-7.221528×10^{-2}
$u^2(t-1)$	7.702835×10^{-3}
$u(t-1)u(t-2)$	1.523536×10^{-2}

Table 8 Simulation results for models 7 and 8

Example	Methods	EI	Number of terms
7	OLS	0.1245	20
	MGA	0.1237	9
8	OLS	0.0064	14
	MGA	0.0048	11

is 20. The result shows that the proposed algorithm is able to determine a more compact model of the system than using the OLS and still gives models with good predictive accuracy. Using the OLS, the ERR value is used as a criterium to select the significant terms to be included in the model. However, based on the result in Table 4, it is difficult to decide on the ERR value to stop regression.

For model 8, the value of EI given by the MGA is 0.0048, which is smaller compared with the EI determined by the OLS, which is 0.0064, as given in Table 8. This result indicates that the proposed algorithm outperformed the OLS algorithm in terms of execution time, number of terms and accuracy.

4.4 Correlation tests

The correlation tests for models 7 and 8 identified using the OLS and the MGA are shown in Figs 8, 9, 10 and 11 respectively. It is observed that almost all the correlation tests were within the 95 per cent confidence bands except for $\phi_{\varepsilon\varepsilon}$ and $\phi_{\varepsilon(\varepsilon u)}$ tests using the OLS for model 8; $\phi_{\varepsilon\varepsilon}$ outside the confidence band can be rectified by introducing the noise model. The tests also indicate that the model developed using the OLS is not yet adequate and extra terms might be needed. Generally, it can be concluded that the tests reveal that the models using the MGA for both examples are adequate.

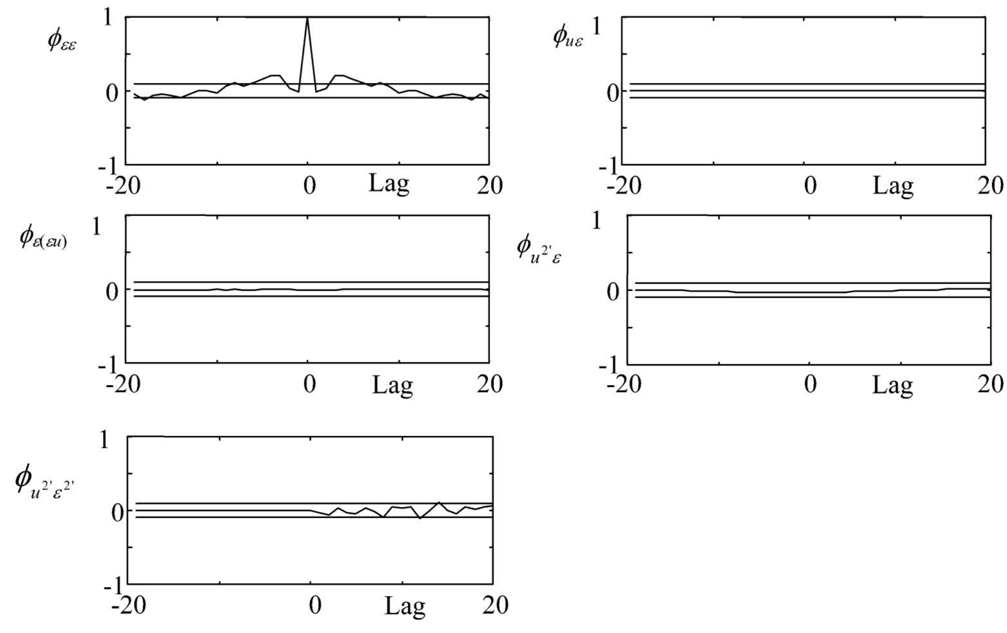

Fig. 8 Model validity test for model 7 identified using the OLS

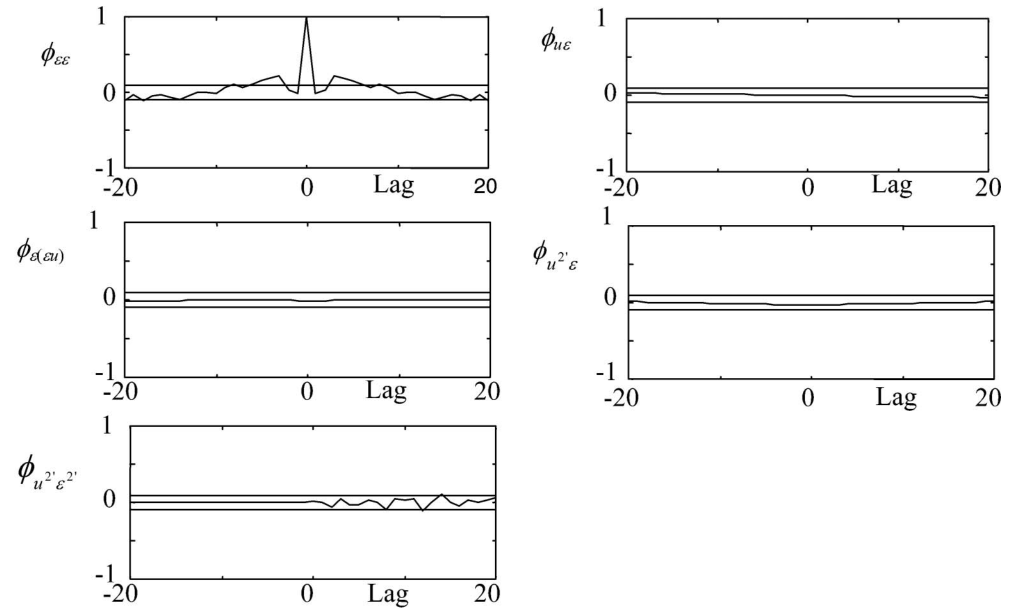

Fig. 9 Model validity test for model 7 identified using the MGA

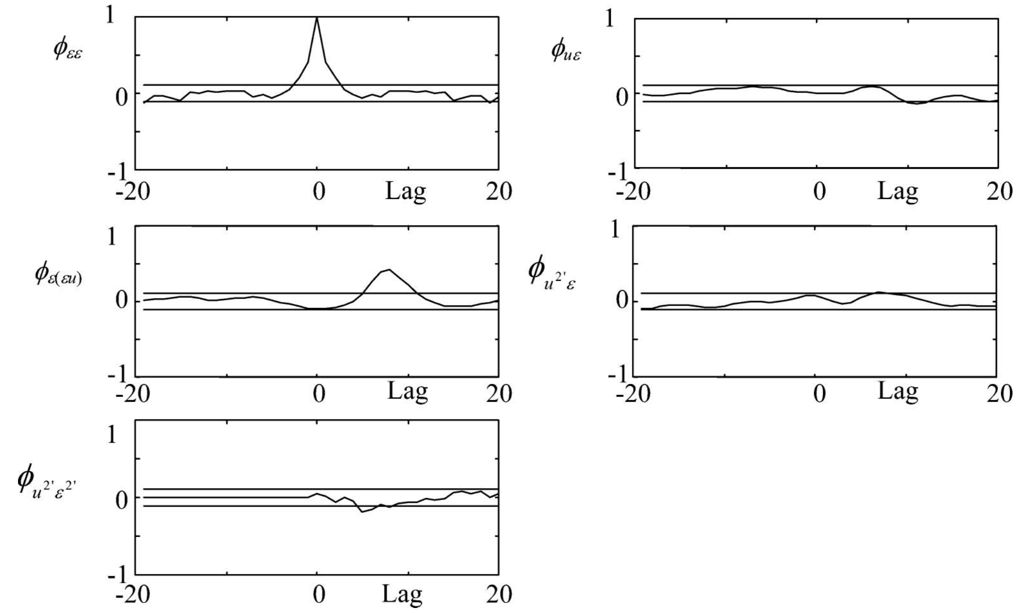

Fig. 10 Model validity test for model 8 identified using the OLS

4.5 Discussion

A study on the influence of the control parameters on the genetic search was presented. These empirical studies have shown that there is no specific guidance on how to choose these recombination operators. The best setting of the GA operators is also dependent on other factors such as population size, fitness function and problem definition. In relation to population size, as discussed previously, it was suggested that more disruptive recombination (higher p_c and p_m) should be used for a smaller population size.

There are some limitations to the SGA discovered in the work, such as the premature convergence of the GA to a non-optimal solution due to loss of some important characteristics (genes) in the chromosome. The results indicated that (a) the loss of critical alleles (bits/positions) occur where the chromosomes carrying those alleles are not chosen for reproduction and (b) selection directs the GA towards promising regions; however, too strong selection will lead the highly fit individuals to take over the population, causing it to miss much better solutions elsewhere in the search.

To further improve the search, the SGA needs to be modified. The success of the genetic search depends on balancing the aspects of population diversity where the different regions of the search space are explored;

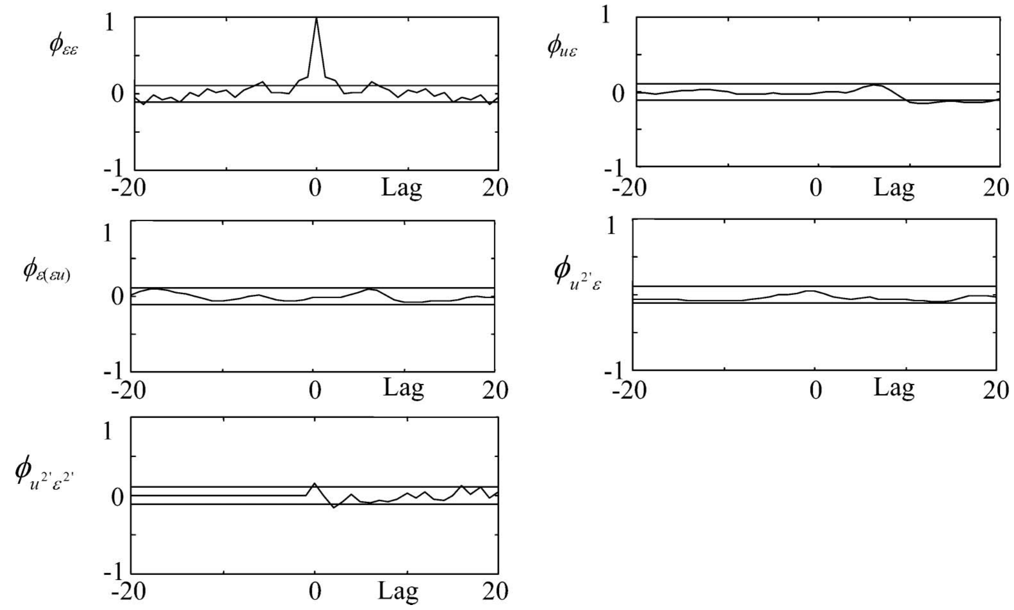

Fig. 11 Model validity test for model 8 identified using the MGA

therefore, it is necessary to tune the search. In the MGA, crossover and mutation are positioned where they should occur so that the best few chromosomes are disrupted with less probability than those with weaker fitness.

The ERR values given in the OLS algorithm indicate the significance of the terms to the model; therefore only the significant terms will be chosen, eliminating the insignificant terms. However, the values of ERR in the empirical studies (Tables 4 and 6) showed that these values reduce gradually and therefore lead to the difficulty in deciding when to stop the regression ending up choosing all the terms. However, in the GA the different choices of structures are explored and the best one is selected among them, giving the model with the minimum error and a smaller number of terms. Furthermore, the GA has the advantage of computational time compared to the OLS. For example, there are 1 048 575 possible models for the identification of the seventh example and with a population size of 50, results will be obtained after the hundredth generation. There are therefore only 5000 states to be computed for the GA to reach a viable solution.

The results obtained in this study indicated that the algorithm is capable of determining the model structure of an unknown dynamic system with a fewer number of terms as compared with the OLS and is also able to identify the models with a faster convergence rate accurately. The simulation results are compared as shown in Table 8 and the models are tested with correlation tests. The validation tests have all shown good results and the values of EI for different case studies are also provided, showing the capabilities of this proposed MGA.

5 CONCLUSION

The study has focused on model structure selection for linear-in-the-parameters dynamic systems using genetic algorithms. It is shown that the proposed algorithm provides an efficient way of determining the model structure of unknown linear and non-linear systems. The algorithm is able to select and explore potential solutions and makes it easier to determine the optimal solution. The case studies demonstrated that it is possible to find an effective search space in the GA and the simulated examples in section 4.2 seem to confirm the conclusion that was made. Optimal tuning of the GA operators remains an open issue. The proper selection of the parameters much depends on the characteristic of the solution space. However, the results indicated that different values of these parameters affect the performance of the algorithm. In conclusion, the simulation results compared to OLS algorithms showed that the modelling accuracy is as good or may be better than the OLS algorithm, but with a more compact model. Furthermore, the problem encountered when using the OLS is when to stop the regression, which is not inherent when using the GA.

REFERENCES

1 **Astrom, K. J.** and **Eykhoff, P.** System identification—a survey. *Automatica*, 1971, **7**, 123–162.

2 **Billings, S. A.** Identification of nonlinear systems: a survey. *Proc. Instn Electl Engrs*, *Part D*, 1980, **127**(6), 272–284.

3 **Haber, R.** and **Keviczky, L.** *Identification of Nonlinear Dynamic Systems, Identification and System Parameter Estimation* (Ed. Rajbman), 1978 (North-Holland, Amsterdam, The Netherlands).

4 **Jacoby, S. L. S., Kowalik, J. S.** and **Pizzo, J. T.** *Iterative Methods for Non-linear Optimization Problem*, 1972 (Prentice-Hall, Englewood Cliffs, New Jersey).

5 **Back, T., Hammel, U.** and **Schwefel, H. P.** Evolutionary computation: comments on the history and current state. *IEEE Trans. Evolutionary Comput.*, 1997, **1**(1), 15–28.

6 **Holland, J. H.** *Adaptation in Natural and Artificial Systems*, 1975 (University of Michigan Press, Ann Arbour, Michigan).

7 **Goldberg, D. E.** *Genetic Algorithms in Search, Optimization and Machine Learning*, 1989 (Addison-Wesley, Reading, Massachusetts).

8 **Michalewicz, Z.** *Genetic Algorithms + Data Structure = Evolution Programs*, 1994 (Springer-Verlag, Berlin, Germany).

9 **Kristinsson, K.** and **Dumont, G. A.** System identification and control using genetic algorithms. *IEEE Trans. Systems, Man and Cybernetics*, 1992, **22**(5), 1033–1046.

10 **Zibo, Z.** and **Naghdy, F.** Application of genetic algorithms to system identification. In Proceedings of IEEE International Conference on *Evolutionary Computation*, 1995, Vol. 2, pp. 777–787.

11 **Tan, K. C., Li, Y., Murray-Smith, D. J.** and **Sharman, K. C.** System identification and linearisation using genetic algorithms with simulated annealing. In Proceedings of the First IEE/IEEE International Conference on *Genetic Algorithms in Engineering System: Innovations and Application*, 1995, pp. 164–169.

12 **Jeong, I. K.** and **Lee, J. J.** Adaptive simulated annealing genetic algorithm for system identification. *Enging Applic. Artif. Intell.*, 1996, **9**(5), 523–532.

13 **Sheta, A. F.** and **De Jong, K.** Parameter estimation of nonlinear systems in noisy environments using genetic algorithms. In Proceedings of the IEEE International Symposium on *Intelligent Control*, Dearborn, Michigan, 15–18 September, 1996, pp. 360–365.

14 **Hong, X.** and **Billings, S. A.** Parameter estimation based on stacked regression on evolutionary algorithms. *IEE Proc. Control Theory and Applic.*, 1999, **146**(5), 406–414.

15 **Haber, R.** and **Unbehauen, H.** Structure identification of nonlinear dynamics systems—a survey on input/output approaches. *Automatica*, 1990, **26**(4), 651–677.

16 **Desrochers, A.** and **Mohseni, S.** On determining the structure of a non-linear system. *Int. J. Control*, 1984, **40**(5), 923–938.

17 **Veres, S. M.** *Structure Selection of Stochastic Dynamic Systems*, 1990 (Gordon and Breach, New York).

18 **Chen, S., Billings, S. A.** and **Luo, W.** Orthogonal least square methods and their application to non-linear system identification. *Int. J. Control*, 1989, **50**(5), 1873–1896.

19 **Li, C. J.** and **Jeon, Y. C.** Genetic algorithm in identifying nonlinear auto regressive with exogenous input models for nonlinear input models for nonlinear systems. In Proceedings of the American Control Conference, San Francisco, California, 1993, pp. 2305–2309.

20 **Korenberg, M., Billings S. A., Liu, Y. P.** and **McIlroy, P. J.** Orthogonal parameter estimation algorithm for non-linear stochastic systems. *Int. J. Control*, 1988, **48**(1), 193–210.

21 **Ljung, L.** *System Identification Theory for the User*, 1999 (Prentice-Hall, Englewood Cliffs, New Jersey).

22 **Chen, S.** and **Billings, S. A.** Representations of non-linear systems: the NARMAX model. *Int. J. Control*, 1989, **49**(3), 1013–1032.

23 **Tsang, K. M.** Spectral analysis for nonlinear systems. PhD thesis, University of Sheffield, 1988.

24 **Chen, S., Billings, S. A.** and **Luo, W.** Orthogonal least square methods and their application to non-linear system identification. *Int. J. Control*, 1989, **50**(5), 1873–1896.

25 **Grefenstette, J. J.** Optimization of control parameters for genetic algorithms. *IEEE Trans. Systems, Man and Cybernetics*, 1986, **16**(1), 122–128.

26 **Qingsheng, R., Jin, Z.** and **Zhongxing, Y.** Analysis of common genetic operators. In Proceedings of the 5th International Conference on *Signal Processing*, 2000, 1684–1687.

27 **Eshelman, L. J., Caruna, R. A.** and **Shaffer, J. D.** Biases in crossover landscape. In Proceedings of the Third International Conference on *Genetic Algorithms* (Ed. J. Schaffer), 1989, pp. 10–19 (Morgan Kaufman, Los Altos, California).

28 **Syswerda, G.** Uniform crossover in genetic algorithms. In Proceedings of the Third International Conference on *Genetic Algorithms* (Ed. J. Schaffer), 1989, pp. 2–9 (Morgan Kaufman, Los Altos, California).

29 **Yuping, W., Hailin, L.** and **Leung, Y.-W.** An effective uniform genetic algorithm for hard optimization problems. In Proceedings of the 3rd World Congress on *Intelligent Control and Automation*, 28 June–2 July, 2000, pp. 656–660.

30 **Fogel, D. B.** Evolutionary optimization. In Proceedings of the 1992 Conference Record of the Twenty-Sixth Asilomar Conference on *Signals, Systems and Computers*, 1992, Vol. 1, No. 1, pp. 409–414.

31 **Hussein, S. B., Jamaluddin, H., Mailah, M.** and **Zalzala, A. M. S.** A hybrid intelligent active force controller for robot arms using evolutionary neural networks. In Proceedings of the 2000 Congress on *Evolutionary Computation*, 2000, Vol. 1, pp. 117–124.

32 **Papadapoulous, S., Mack, R. J.** and **Massara, R. E.** A hybrid genetic algorithms method for optimizing analog circuits. In Proceedings of the 43rd IEEE Symposium on *Circuits and Systems*, Lansing, Michigan, 8–11 August 2000, pp. 140–143.

33 **Billings, S. A.** and **Voon, W. S. F.** Correlation based model validity tests for non-linear models. *Int. J. Control*, 1986, **44**(1), 235–244.

34 **Luh, G. C.** and **Wu, C. Y.** Non-linear system identification using genetic algorithms. *Proc. Instn Mech. Engrs, Part I: J. systems and control engineering*, **213**(2), 105–118.

35 **Box, G. E. P., Jenkins, G. M.** and **Reinsel, G. C.** *Time Series Analysis Forecasting and Control*, 1994 (Prentice-Hall, Englewood Cliffs, New Jersey).

mi00003103 15-01-04 11:52:56 Rev 14.05
The Charlesworth Group, Huddersfield 01484 517077