Structure Detection for Nonlinear Rational Models Using Genetic Algorithms

S.A.Billings K.Z.Mao

Department of Automatic Control and Systems Engineering

University of Sheffield

Mappin Street, Sheffield S1 3JD

United Kingdom

Research Report No. 634

July 12, 1996

University of Sheffield

Structure Detection for Nonlinear Rational Models Using Genetic Algorithms

S.A.Billings K.Z.Mao

Department of Automatic Control and Systems Engineering

University of Sheffield

Sheffield S1 3JD, UK

Abstract

A new nonlinear rational model identification algorithm is introduced based on genetic algorithms. Compared with other rational model identification approaches, the new algorithm has two main advantages. First, this algorithm does not require a linear-in-the-parameters regression equation, and as a consequence the severe noise problems induced by multiplying out the rational model are avoided. Second, the new algorithm provides near-optimal global parameter estimation. Unfortunately, this is balanced by an enormous computational load even when identifying models which consist of modest parameter sets. Simulated examples are included to illustrate that the new algorithm works well on simple simulated examples but can fail when applied in more realistic situations.

1 Introduction

In the past few years, nonlinear polynomial models have received considerable attention (Leontaritis and Billings 1985a, 1985b, Billings and Chen 1989a, Chen and Billings 1989, Haber and Unbehauen 1990). Nonlinear polynomial models are fine for many applications (Billings *et al* 1988, Hernandez and Arkun 1993, Proll and Karim 1994, Sriniwas *et al* 1995, Thomson *et al* 1996) but alternative expansions may provide a more concise approximation for severely nonlinear systems (Billings and Chen 1989b). The nonlinear rational

model formulation, defined as the ratio of two nonlinear polynomial expansions, was recently introduced as an alternative to the polynomial model. Parameterization of systems with rational models offers substantial advantages compared with linear or nonlinear polynomial expansions (Ratkowsky 1987, Ponton 1993). However the rational model is much more difficult to identify because of the inherent nonlinear-in-the-parameters model expression. One of the first algorithms for nonlinear dynamic rational model identification was the prediction error estimation algorithm introduced by Billings and Chen (1989b). The main disadvantage of this approach was that nonlinear optimization was an inherent part of the algorithm. The second approach was the Rational Model Estimator (Billings and Zhu 1991, 1994a, Zhu and Billings 1993). This method involved multiplying out the rational model to form a linear-in-the-parameters regression equation so that linear least squares estimation algorithms can be applied to identify nonlinear rational models. But multiplying out the model induces a severe noise problem which will result in biased estimates even if the noise on the output is additive and white. This was overcome in the Rational Model Estimator (RME) by reformulating the least squares estimator using a new iterative procedure to essentially estimate and remove the bias terms induced by the noise. Structure detection was also incorporated to determine the significant terms in the model prior to final estimation. The third approach is the smoothed data identification algorithm derived by Billings and Mao (1996). This is based on smoothing the raw data using a NARMAX smoother and then performing the identification based on the smoothed data. If the smoother is properly constructed, preprocessing the data alleviates the severe noise problems which arise from multiplying out the model and allows all the well developed structure detection and parameter estimation algorithms for polynomial models to be applied to rational models.

In the present study, a new rational model identification algorithm based on genetic algorithms (GAs) is introduced. Genetic algorithms are a class of randomized search and optimization techniques motivated by the principle of natural evolution and population genetics (Goldberg 1989). Unlike other optimization methods, GAs can be applied to a very wide range of problems and the solution is less likely to be forced into a local

minimum because of the random multi-point parallel search mechanism. These two characteristics make GAs highly suitable for nonlinear rational model identification. There is no need to multiply out the rational model to apply GAs, and consequently the severe noise problems are avoided and the local minima problem which arises in the prediction error estimation algorithm is overcome. However, our simulation studies showed that these advantages were achieved at the expense of excessive computations. The new algorithm worked well on simple simulated systems with modest candidate term sets. But failed when applied to more realistic examples and to industrial data.

The paper is organized as follows. Previouly developed rational model identification algorithms are briefly reviewed and analysed in §2.1. The basics of genetic algorithms are introduced in §2.2.1. Genetic algorithms for nonlinear rational model parameter estimation and structure detection are discussed in §2.2.2. §3 presents several numerical simulation examples to demonstrate the application of GAs in dynamic nonlinear rational model identification.

2 Nonlinear rational model identification based on genetic algorithms (GAs)

2.1 Nonlinear rational model identification algorithms: a brief review and analysis

Consider the dynamic nonlinear rational model given by:

$$\begin{aligned} y(k) &= \frac{F_n[y(k-1)\dots y(k-n_{ny}), u(k-1)\dots u(k-n_{nu}), e(k-1)\dots e(k-n_{ne})]}{F_d[y(k-1)\dots y(k-n_{dy}), u(k-1)\dots u(k-n_{du}), e(k-1)\dots e(k-n_{de})]} + e(k) \\ &= \frac{\sum_{j=1}^{n_{num}} \varphi_{nj}(k)\theta_{nj}}{\sum_{j=1}^{n_{den}} \varphi_{dj}(k)\theta_{dj}} + e(k) \end{aligned} \tag{1}$$

where $F_n(\bullet)$ and $F_d(\bullet)$ are nonlinear polynomials functions of $(\bullet)$, $u(k)$ and $y(k)$ denote the input and output at sampling instant $k(k = 0, 1, \ldots)$ respectively, $\{e(k)\}$ is an unobservable independent noise sequence with zero mean and finite variance which accounts for uncertainties, possible noise and unmodeled dynamics, $n_\bullet$ denotes the order, $\varphi_\bullet(k)$ and $\theta_\bullet$ denote the regressor and the parameters respectively.

Because the nonlinear rational model is nonlinear in the parameters, ordinary least squares algorithms can not be applied directly. So far three types of algorithm have been developed to address the rational model identification problem.

2.1.1 Rational Model Estimator (RME) (Billings and Zhu 1991)

The motivation of RME is to multiply out the rational model to form a linear-in-the-parameters regression equation so that linear least squares algorithms can be applied to nonlinear rational model identification. Multiplying out equation (1) yields

$$\begin{aligned} Y(k) &= F_n(\bullet) - y(k)[F_d(\bullet) - \varphi_{d1}(k)\theta_{d1}] + F_d(\bullet)e(k) \\ &= \sum_{j=1}^{n_{num}} \varphi_{nj}(k)\theta_{nj} - \sum_{j=2}^{n_{den}} \varphi_{dj}(k)y(k)\theta_{dj} + \xi(k) \\ &= \phi(k)\Theta + \xi(k) \end{aligned} \tag{2}$$

where:

$$Y(k) = y(k)\varphi_{d1}(k)\theta_{d1} \mid_{\theta_{d1}=1}$$

$$\xi(k) = F_d(\bullet)e(k)$$

$$\Theta = [\theta_{n1} \ldots \theta_{nn_{num}} \ \theta_{d2} \ldots \theta_{dn_{den}}]^T$$

$$\phi(k) = [\varphi_{n1}(k) \ldots \varphi_{nn_{num}}(k), \ -\varphi_{d2}(k)y(k) \ldots - \varphi_{dn_{den}}(k)y(k)]$$

Notice that multiplying out the model induces a severe noise problem, because the residue

$\xi(k)$ becomes highly correlated with the regressor $\phi(k)$. This occurs even when the noise in eqn (1) is purely additive and white because of the term $F_d(\bullet)e(k)$ in eqn (2). This is unique to the rational model and does not apply to the linear and polynomial model forms. As a consequence estimates based on eqn (2) will be biased if ordinary least squares algorithms are applied directly to eqn (2). The Rational Model Estimator (RME) was derived as a reformulated least squares estimator to overcome this problem by iteratively estimating and removing the bias terms caused by the correlation between $\xi(k)$ and $\phi(k)$ (Billings and Zhu 1991, 1994, Zhu and Billings 1993).

2.1.2 Smoothed Data Based Estimation Algorithm (Billings and Mao 1996)

Eqn (2) can also be written as

$$\begin{aligned} Y(k) &= \sum_{j=1}^{n_{num}} \varphi_{nj}(k)\theta_{nj} - \sum_{j=2}^{n_{den}} \varphi_{dj}(k)y(k)\theta_{dj} + \sum_{j=1}^{n_{den}} \varphi_{dj}(k)\theta_{dj}e(k) \\ &= \sum_{j=1}^{n_{num}} \varphi_{nj}(k)\theta_{nj} - \sum_{j=2}^{n_{den}} \varphi_{dj}(k)[\overline{y}(k) + e(k)]\theta_{dj} + \sum_{j=1}^{n_{den}} \varphi_{dj}(k)\theta_{dj}e(k) \\ &= \sum_{j=1}^{n_{num}} \varphi_{nj}(k)\theta_{nj} - \sum_{j=2}^{n_{den}} \varphi_{dj}(k)\overline{y}(k)\theta_{dj} + \varphi_{d1}(k)e(k) \\ &= \phi_{free}(k)\Theta + \xi_1(k) \end{aligned} \tag{3}$$

where

$$\phi_{free}(k) = [\varphi_{n1}(k) \ldots \varphi_{nn_{num}}(k), \ -\varphi_{d2}(k)\overline{y}(k) \ldots - \varphi_{dn_{den}}(k)\overline{y}(k)]$$

$$\xi_1(k) = \varphi_{d1}(k)e(k)$$

$$\overline{y}(k) = y(k) - e(k) = \frac{F_n(\bullet)}{F_d(\bullet)}$$

and $\overline{y}(k)$ is the current time noise free part of the noisy output $y(k)$, which is uncorrelated with the residue $\xi_1(k)$. If $\overline{y}(k)$ is initially extracted from the raw signal and the estimator is then formulated based on eqn (3), most of the noise problems arising from multiplying out the model can be avoided, and all the well developed nonlinear polynomial model

identification algorithms can be applied directly to rational models. Billings and Mao (1996) showed that this idea works well in practice if a NARMAX smoothing algorithm is used to preprocess the raw data to yield the signal $\overline{y}(k)$ which can be used in eqn (3) above.

2.1.3 Prediction Error Estimation Algorithm (Billings and Chen 1989b)

The prediction error estimation algorithm does not require a linear-in-the-parameters regression equation and can be applied directly to eqn (1). The disadvantage of this algorithm is that the estimates are determined using a nonlinear optimization procedure and local minima can be a problem.

Structure detection, or determining from the raw data which are the significant terms in the model, is an important step in any nonlinear identification procedure and all the above algorithms incorporate some form of structure detection to ensure that parsimonious models are identified.

2.2 Rational model identification using genetic algorithms

2.2.1 Genetic algorithms: Basic principles and features

Genetic algorithms (GAs) are a class of randomized search procedure which were initially motivated by the principles of natural evolution and population genetics. Individuals which fit the environment better will survive and hand down their chromosomes to their descendants while less fit individuals will become extinct (Goldberg 1989, Davis 1989, Fonseca and Fleming 1995). Basic genetic algorithms involve three operators: *reproduction, crossover* and *mutation. Reproduction* is a process in which a new generation is produced by randomly selecting strings from an existing population according to their fit-

ness. This process means that individuals with a high fitness value obtain more copies in the next generation, while less fit individuals may become extinct. *Crossover* is the most dominant operator in GAs. Under this operation, a pair of strings are randomly selected, a random point is chosen, and the right parts of the parents' chromosomes are exchanged to produce two new offsprings. Although this does not add any new genetic material to the population, *crossover* allows recombining of good population chromosomes. *Mutation* is a local operator which acts with a very low probability. Its role is to produce new genes and to help the population evolution. Through successive applications of these three operators the solution should converge to the global minimum.

Genetic algoritms (GAs) are distinguished from most other gradient based optimization approaches by the following characteristics:

- GAs work with a coding of the parameter set, not the parameters themselves. Binary encoding is normally used and has been suggested as being optimal in some cases. But this is not a requirement of GAs and other encoding modes such as decimal encoding can also be used (Billings and Zheng 1995).
- GAs are a class of multi-solution parallel search methods. Instead of pushing one single candidate towards the optimal solution, GAs act on a candidate solution set to explore several zones of the search space simultaneously. Consequently GAs are capable of providing global near-optimal solutions.
- GAs use only the fitness function to guide the search, and can be applied to various problems with no specific requirements on the problem forms.
- GAs are randomized algorithms because the search mechanism uses a probabilistic operator.

2.2.2 GAs for dynamic rational model parameter estimation and structure detection

Two of the attractive features of GAs are the simplicity and the multisolution parallel optimization mechanism. Unlike ordinary least squares estimation algorithms which require linear-in-the-parameters regression equations, GAs are not restricted to specific forms of problems. Moreover, the solution is less likely to be forced into a local minimum because of the random multi-point parallel search mechanism. These two characteristics make GAs suitable for nonlinear rational model identification. If GAs are applied, there is no need to multiply out the rational model, and consequently the local minima problems which can arise in prediction error estimation, and the severe noise problems which arise if the model is multiplied out are both avoided.

(i) Cost function and fitness function

In system identification, a quadratic cost function is usually employed

$$J_1(N,\Theta) = \sum_{j=1}^{N}[y(k) - \hat{y}(k)]^2 \tag{4}$$

where $\hat{y}(k)$ is the one step ahead predicted output. The objective is to find an estimation $\hat{\Theta}$ so that this cost function is minimized. However, models obtained by minimizing the cost function $J_1(N,\Theta)$ might fail to capture the underlying dynamics in the data in cases where the data is noisy and/or the model structure is unknown. Recently, regularization was introduced to improve estimation accuracy (Hoerl and Kennard 1970, Barron and Xiao 1991, Bishop 1991, Orr 1995, Mao *et al* 1996) and this approach can be used to advantage in the new GA based algorithm for rational model structure detection. The regularized solution for Θ is defined as the value which minimises

$$J_2(N,\lambda,\Theta) = \frac{1}{N}\sum_{k=1}^{N}[y(k) - \hat{y}(k)]^2 + \Theta^T\lambda\Theta \tag{5}$$

where λ =diag$\{\lambda_i,\ i = 1, 2, \ldots, n_{num} + n_{den}\}$, $\lambda_i > 0$ is a scalar regularization parameter, and diag($\bullet$) denotes a diagonal matrix.

Regularization helps to improve the structure detection to some extent. It has been found, however, that using only this constraint does not yield a parsimonious model. To overcome this problem, the model size is constrained by including the number of model terms in the cost function

$$J_3(N, \lambda, \gamma, \Theta, n_T) = \frac{1}{N}\sum_{k=1}^{N}[y(k) - \hat{y}(k)]^2 + \Theta^T\lambda\Theta + \gamma n_T \tag{6}$$

where n_T denotes the number of terms included in the candidate models, and γ is a positive real number.

The GA search is guided by fitness, and therefore the value of the cost function should be mapped into a fitness function. The fitness value is a measure of the minimization of the cost function. The smaller the cost function value, the higher the fitness. Thus, the fitness function should be inversely proportional to the cost function

$$f(\bullet) \propto \frac{1}{J_3(\bullet)}$$

The *windowing mapping* scheme (Davis 1989) is employed in the present study to map the inverse relation. A zero or a constant minimum fitness value is initially assigned to the worst individual, then each individual of the population is assigned a fitness value proportional to the amount of the cost less the cost of the worst case

$$f(\bullet) = f_{max}(\bullet) - \kappa\frac{f_{max}(\bullet) - f_{min}(\bullet)}{J_{3max}(\bullet) - J_{3min}(\bullet)}[J_3(\bullet) - J_{3min}(\bullet)] \tag{7}$$

where $J_{3min}(\bullet)$, $J_{3max}(\bullet)$, and $f_{min}(\bullet)$, $f_{max}(\bullet)$ denote the minimum and maximum cost values, and the minimum and maximum fitness values respectively. κ is a positive parameter which tunes the mapping between cost function and fitness function and which

is typically set in the range [1 10].

(ii) Encoding

Genetic algorithms work by coding the parameters, rather than with the parameters themselves. Binary coding is normally used and has been suggested to be optimal in many applications. A standard encoding is to concatenate the binary string of each real parameter (assume there are n parameters and each parameter is represented by m bits)

$$P_{n1}P_{n2}\dots P_{nm},\dots,P_{i1}P_{i2}\dots P_{im},\dots,P_{11}P_{12}\dots P_{1m}$$

where P_{ij} denotes the jth bit of the ith parameter, this is either 1 or 0. In nonlinear dynamic model identification, if the model structure is not known *a priori* , the number of candidate terms will be excessively large even for simple rational models. For example, a rational model with just two lagged outputs, inputs and noise terms will produce 56 candidate terms in a model of nonlinearity degree two. The presence of many insignificant terms can affect the selection of the significant model terms. This problem can be solved by using a variable length string coding mode and by constraining the model size. Variable length coding is equivalent to forcing some parameters to be zero

$$P_{n1}P_{n2}\dots P_{nm},\dots,\overbrace{00\dots0}^{m},\dots,P_{11}P_{12}\dots P_{1m}$$

How many and which parameters are zero are all unknown in the initial population set and hence the number and position of the zero parameters is uniformly and randomly distributed in the ranges $[0, n-1]$ and $[1, n-1]$ respectively.

(iii) Reproduction

The *roulette wheel* approach is employed to implement the production procedure in this study. Each string is allocated a slot of the roulette wheel subtending an angle proportional to its fitness to the center of the wheel. A random number in the range of 0 to 2π

is generated. A copy of a string goes to the mating pool if the random number falls in the slot corresponding to the string. For a population with size l, the process is repeated $l/2$ times and l strings go into the mating pool.

(iv) Crossover

Multiple point crossover is applied in this study. After a pair of parent strings are randomly selected, the parameter from which the two strings exchange the right parts should be determined first. If any of the determined parameters in the two strings is zero, the two strings exchange bits from the begining bit of the selected parameter in order not to change the zero value of that parameter. If both of the two selected parameters are not zero, the right parts of the two strings from a randomly selected bit in the selected parameter are exchanged. The above procedure is repeated several times. This can be brifely demonstrated by the following example.

First randomly selected a pair of parent strings

$$P_{n1}\dots P_{nm},\dots,P_{i+11}\dots P_{i+1m},\overbrace{P_{i1}\dots P_{ij-1}P_{ij}P_{ij+1}\dots P_{im}}^{m},\dots P_{11}\dots P_{1m}$$

$$P'_{n1}\dots P'_{nm},\dots,P'_{i+11}\dots P'_{i+1m},\overbrace{P'_{i1}\dots P'_{ij-1}P'_{ij}P'_{ij+1}\dots P'_{im}}^{m},\dots P'_{11}\dots P'_{1m}$$

Then randomly select the parameter from which the two strings exchange their bits, for example the i-th parameter P_i.

If P_i in one of the two strings is equal to zero, exchange the bits from the beginning bit of that parameter to produce two offsprings

$$P_{n1}\dots P_{nm},\dots,P_{i+11}\dots P_{i+1m},\overbrace{P'_{i1}\dots P'_{ij-1}P'_{ij}P'_{ij+1}\dots P'_{im}}^{m},\dots P'_{11}\dots P'_{1m}$$

$$P'_{n1}\dots P'_{nm},\dots,P'_{i+11}\dots P'_{i+1m},\overbrace{P_{i1}\dots P_{ij-1}P_{ij}P_{ij+1}\dots P_{im}}^{m},\dots P_{11}\dots P_{1m}$$

If P_i in both strings is not zero, randomly select a bit from the parameter, for example the j-th bit and exchange the bits from the j-th bit

$$P_{n1}\dots P_{nm},\dots,P_{i+11}\dots P_{i+1m},\overbrace{P_{i1}\dots P_{ij-1}P'_{ij}P'_{ij+1}\dots P'_{im}}^{m},\dots P'_{11}\dots P'_{1m}$$

$$P'_{n1}\dots P'_{nm},\dots,P'_{i+11}\dots P'_{i+1m},\overbrace{P'_{i1}\dots P'_{ij-1}P_{ij}P_{ij+1}\dots P_{im}}^{m},\dots P_{11}\dots P_{1m}$$

The aim is not to change the zero parameter terms. The objective in combining this crossover mode and the cost function $J_3(\bullet)$ is to try and ensure that the algorithm provides a parsimonious model.

(v) Mutation

Mutation is a local operator that transfers the bits of a string. If the string is binary encoded, 1 is replaced by 0, and 0 is replaced by 1. The operator is applied bit by bit according to a predetermined mutation probability. The mutation operator enables GAs to overcome local minima. The mutation probability is taken as 0.06 in the present study. In order not to change the zero parameters, the mutation operation does not act on these.

(vi) Termination

Usually the GA search procedure is terminated when the search arrives at a pre-specified generation. In dynamic system identification, another termination rule based on model validaty tests should also be satisfied. Two recently introduced tests (Billings and Zhu 1994b) based on higher order cross-correlation functions between the output, input and residual are given by

$$\Phi_{\epsilon e^2}(\tau)=\frac{\sum_{k=1}^{N}(\epsilon(k)-\overline{\epsilon})(e^2(k-\tau)-\overline{e^2})}{\left[\left(\sum_{k=1}^{N}(\epsilon(k)-\overline{\epsilon})^2\right)\left(\sum_{k=1}^{N}(e^2(k)-\overline{e^2})^2\right)\right]^{0.5}} \tag{8}$$

$$\Phi_{\epsilon u^2}(\tau) = \frac{\sum_{k=1}^{N}(\epsilon(k)-\bar{\epsilon})(u^2(k-\tau)-\overline{u^2})}{\left[\left(\sum_{k=1}^{N}(\epsilon(k)-\bar{\epsilon})^2\right)\left(\sum_{k=1}^{N}(u^2(k)-\overline{u^2})^2\right)\right]^{0.5}} \tag{9}$$

where

$$\epsilon(k) = y(k)e(k)$$

$$\bar{\epsilon} = \frac{1}{N}\sum_{k=1}^{N}\epsilon(k)$$

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

$$\overline{e^2} = \frac{1}{N}\sum_{k=1}^{N}e^2(k)$$

If the residues satisfy

$$\Phi_{\epsilon e^2}(\tau) = \begin{cases} k_1 > 0 & \text{if } \tau = 0 \\ 0 & \text{otherwise} \end{cases}$$

$$\Phi_{\epsilon u^2}(\tau) = 0 \quad \text{for any } \tau.$$

the estimated model is considered to be an unbiased representation of the system otherwise the search procedure should be continued. In practice, the 95% confidence limits at approximately $1.96/\sqrt{N}$ are used to determine if the model is valid.

The successive application of these operators can be summarized as follows

(i) Generate an initial population set $\mathcal{P}$ consisting of l individuals, where l is the so called population size. Set the current generation number $i = 1$.

(ii) Calculate the fitness value of each individual in set $\mathcal{P}$. Form a mating pool $\mathcal{M}$ using all individuals in the population set $\mathcal{P}$ at the probabilties assigned to each individual according to the corresponding fitness value.

(iii) Randomly select a pair of parent strings from the mating pool $\mathcal{M}$. Choose a random crossover point and exchange the parent string bits to produce two offsprings and put the offsprings in the offspring set $\mathcal{O}$. Repeat the procedure $l/2$ times.

(iv) Mutate each bit of the non-zero parameters in each offspring in the set $\mathcal{O}$ with a predefined probability, and calculate the fitness value of each offspring.

(v) Select the l fitest individuals from sets $\mathcal{P}$ and $\mathcal{O}$ by comparing fitness values.

(vi) Reset the set $\mathcal{P}$ with the newly selected l individuals, reset the number of generation $i = i + 1$, and nullify the offspring set $\mathcal{O}$.

(vii) Steps (ii)-(vi) are repeated until the residue termination conditions are satisfied.

Remark 1

In almost all practical cases, the noise $e(k)$ in equation (1) will be coloured. Therefore a noise modelling procedure should be included as part of the parameter estimation algorithm in order to obtain unbiased parameter estimates. This can be done by using an iterative procedure (Billings and Zhu 1991). Notice that the noise estimation at the current iteration will be based on the noise sequence computed at the previous iteration.

In genetic algorithms, this iterative procedure can be carried out together with the population evolution. But there is a problem. When the gene of two parent strings is exchanged in a crossover operation, two pairs of new parameter estimates are created. It is difficult to clarify which of these the noise estimation should be based on to produce the new noise estimation corresponding to the newly created parameter estimate. Fortunately, genetic algorimths tend to be insensitive to the noise estimation error, and in the present study all the individuals at one generation have the same noise estimation as the best individual.

Remark 2

In the present study model size was introduced as a penealty term in the cost function to provide a balance between the prediction error and the model size. A proper choice of the weight γ will help to build a model of a suitable size and with a satisfactory accuracy. However, the selection of weight γ was found to be very difficult and in practice this was

done by a lot of trial and error.

The peneaty on model size helps to solve the non-parsimonious model problem to some extent but this kind of implicit method does not solve the problem completely. An explicit constraint on model size would be preferable. Hypothesis tests have proved to be an efficient tool for term deletion/selection for nonlinear polynomial models (Leontaritis and Billings 1987), but this approach cannot be applied in the present algorithm. In the case of polynomial models, the parameter estimates under a specific model structure can easily be obtained by using a least squares type estimation routine, but in the GA algorithm both the model structure and the parameter estimates are uncertain in the search process and it is impossible to determine if an unacceptable prediction error is caused by the wrong model structure or by incorrect parameter estimation. Methods of introducing a proper explicit term deletion procedure into the genetic algorithm is the subject of future research.

Remark 3

If the model structure is unknown, the number of candidate terms is usually excessively large. For example a model with 3 lagged inputs, outputs and noise terms will produce 440 candidate terms for a single-input-single-output (SISO) rational model of nonlinear degree 3, and 2660 candidate terms for a two-input-two-output system with the same specifications. Searching for the model structure and parameter estimates in high dimensional space is computational expensive. Even a search over only 56 candidate terms (Example 2) will typically take a few hours on a Sun Sparc Station 20. This is a major disadvantage of the GA approach which for rational model structure detection and parameter estimation is not as efficient as the forward regression orthogonal estimation algorithm (Billings and Zhu 1991, Zhu and Billings 1993, Billings and Mao 1996).

3 Simulation Examples

Three examples will be used to test the new GA algorithm for nonlinear rational model identification. In all the examples, the input $u(k)$ was a uniformly distributed random sequence with zero mean and amplitute ± 1, and the noise $e(k)$ was a normally distrubuted disturbance sequence with zero mean and variance 0.01. In Examples 1 and 2, 600 pairs of input/output data were produced, where the first 500 pairs were used for identification and the last 100 pairs for testing. In Example 3, 2000 pairs of input/output data were used for identification, and 100 pairs for testing.

Example 1

Consider the following dynamic nonlinear rational model

$$y(k) = \frac{y(k-1) + u(k-1)u(k-2) + u(k-1)}{1 + y^2(k-1) + u^2(k-1)} + e(k)$$

To initialise the algorithm it was assumed that the maximum lags in the input and output was 2, and the degree of nonlinearity was also 2. This model specification will produce 15 candidate terms for both the denominator and numerator. The working parameters of the GA algorithm were as follows: $\lambda_i = 0.05$ $(i = 1, 2, \ldots, 30)$, $\gamma = 4$, the population size was set to 70, and the mutation rate was equal to 0.06. Each parameter was represented by an 8 bit binary number, where 1 bit represents the sign (positive or negative), 3 bits are used for the integer part and 4 bits for the fractional part. At the 200^{th} generation the model structure and parameter estimation were as shown in Table 1, the model validity tests were as shown in Fig.1, and the one-step ahead predictions were as shown in Fig.2.

Example 2

Consider a more realistic system which is disturbed by coloured nonlinear noise

$$y(k) = \frac{y(k-1)y(k-2) + u(k-1)u(k-2) + u(k-1)e(k-1)}{1 + y^2(k-1) + y^2(k-2)} + e(k)$$

Assume that the maximum lags in the input, output and noise are all equal to 2. The element set

$$\{y(k-1), y(k-2), u(k-1), u(k-2), e(k-1), e(k-2)\}$$

will produce 28 terms in both the numerator and denominator for a rational model of nonlinear degree 2. The working parameters of the GA algorithm were defined as follows: $\lambda_i = 0.05$ $(i = 1, 2, \ldots, 56)$, $\gamma = 4$, the population size was set to 100, and the mutation rate was equal to 0.06. Each parameter was represented by an 8 bit binary number as defined in Example 1. At the 200^{th} generation the model structure and parameter estimation were as shown in Table 2, the model validity tests were as shown in Fig.3, and the one-step ahead prediction were as shown in Fig.4.

Example 3

Consider a more complex nonlinear dynamic system

$$y(k) = \frac{y^3(k-1) + u(k-1)u(k-2) + u^3(k-4) + u(k-5) + u(k-3)e(k-1)}{1 + y^3(k-1) + u^2(k-2)y(k-3)} + e(k)$$

It was assumed that the maximum lags in the input, output and noise were 5, 5 and 2 respectively. The element set

$$\begin{aligned} \mathcal{S} &= \{y(k-1), y(k-2), y(k-3), y(k-4), y(k-5)\} \\ &\quad \cup \{u(k-1), u(k-2), u(k-3), u(k-4), u(k-5)\} \cup \{e(k-1), e(k-2)\} \end{aligned}$$

produces a total of 910 terms for a rational model with nonlinear degree 3. The working parameters for the GA algorithm were defined as follows: $\lambda_i = 0.05$ $(i = 1, 2, \ldots, 910)$, $\gamma = 4$, the population size was set to 100, and the mutation rate was 0.06. Each parameter

was represented by an 8 bit binary number as defined in Example 1. At the 20000^{th} generation the model still had several hundred terms. The genetic algorithm failed to build a parsimonious model when the number of candidate terms was large. This result was typical of all the examples we tried with a large candidate set and seemed to be a result of the problems discussed in Remark 2.

Our experience of analysing real data shows that it is quite usual to have to search over quite a broad range of lags and to a lesser extent degree of nonlinearity. This inevitably leads to large candidate sets even for SISO systems. The genetic algorithm introduced in the present study requires significant computations in such cases and the extension of this to MIMO systems does not seem realistic at the present time. Indeed we tried to use the GA algorithm to identify a model of a diesel engine but failed despite several attempts. Yet this system can quickly be identified using the orthogonal estimator (Billings *et al* 1989c).

4 Conclusions

A new rational model identification algorithm based on genetic algorithms has been developed in this study. Details of nonlinear model structure detection and parameter estimation using GAs have been discussed. The algorithm works well on simple simulated examples where the range of allowed lags is small. But this is an unrealistic test since most analysis of real data sets often involves a search over quite a broad range of lags and degree of nonlinearity. In the latter case the GA routine involves an enormous computational load and despite several attempts we failed to identify a relatively simple simulated system and an industrial diesel engine. This suggests that more research is required before genetic algorithms can be successfully used to identify MIMO nonlinear models from real data sets.

5 Acknowledgements

The authors would like to thank Dr G.L.Zheng, Department of ACSE, University of Sheffield, for helpful discussions on genetic algorithms. SAB gratefully acknowledges that part of this work was supported by EPSRC.

References

[1] Barron, A.R. and X.Xiao, 1991, Discussion of multivariate adaptive regression splines, *Annuals of Statistics*, 19, 67-82.

[2] Billings, S.A., Fadzil, M.B., Sulley, J.L. and P.M.Johnson, 1988, Identification of a nonlinear difference equation model of an industrial diesel generator, *Mechanical Systems and Signal Processing* , 2, 59-76.

[3] Billings, S.A. and S.Chen, 1989a, Extended model set, global data and threshold model identification of severely nonlinear systems, *International Journal of Control*, 50, 1897-1923.

[4] Billings, S.A. and S.Chen, 1989b, Identification of nonlinear rational systems using a prediction error estimation algorithm, *International Journal of System Science*, 20, 467-494.

[5] Billings, S.A., S.Chen and R.J.Backhouse, 1989c, The identification of linear and nonlinear models of a turbocharged automotive diesel engine, *Mechanical Systems and Signal Processing*, 3, 123-142.

[6] Billings, S.A. and Q.M.Zhu, 1991, Rational model identification using an extended least square algorithm, *International Journal of Control* , 54, 529-564.

[7] Billings, S.A. and Q.M.Zhu, 1994a, A structure detection algorithm for nonlinear dynamic rational models, *International Journal of Control* , 59, 1439-1463.

[8] Billings, S.A. and Q.M.Zhu, 1994b, Nonlinear model validation using correlation tests, *International Journal of Control* , 60, 1107-1120.

[9] Billings, S.A. and G.L.Zheng, 1995, Radial basis function network configuration using genetic algorithm, *Neural Networks*, 8, 877-890.

[10] Billings, S.A. and K.Z.Mao, 1996, Rational model data smoothers and identification algorithms (submitted for publication).

[11] Bishop, C., 1991, Improving the generalization properties of radial basis function neural networks, *Neural Computation*, 3, 579-588.

[12] Chen, S. and S.A.Billings, 1989, Representations of nonlinear systems:the NARMAX model, *International Journal of Control* , 48, 1013-1032.

[13] Davis, L., Adaptive operator probabilities in genetic algorithms, *Proceedings of the Third International Conference on Genetic Algorithms*, Morgan Kauffman Publisher, Cambridge, Massachusetts, 1989, 61-69.

[14] Fonseca, C.M. and P.J.Fleming, 1995, An overview of evolutionary algorithms in multiobjective optimization, *Evolutionary Computation*, 3, 1-16.

[15] Goldberg, D.E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Reading, Massachusetts: Addison-Wesley, 1989.

[16] Haber, R. and H.Unbehauen, 1990, Structure identification of non-linear dynamic system-a survey on input/output approaches, *Automatica*, 26, 651-677.

[17] Hernandez, E. and Y.Arkun, 1993, Control of nonlinear system using polynomial ARMA models, *AIChe Journal* , 39, 446-462.

[18] Hoerl, A. and R.Kennard, 1970, Ridge regression:application to nonorthogonal problems, *Technometrics*, 12, 55-67.

[19] Leontaritis, I.J. and S.A. Billings, 1985a, Input-output parametric models for non-linear systems, Part One, *International Journal of Control*, 41, 303-328.

[20] Leontaritis,I.J. and S.A. Billings, 1985b, Input-output parametric models for nonlinear systems, Part Two, *International Journal of Control*, 41, 329-344.

[21] Leontaritis,I.J. and S.A. Billings, 1987, Model selection and validation methods for nonlinear systems, *International Journal of Control*, 45, 311-341.

[22] Mao, K.Z., S.A.Billings and Q.M.Zhu, 1996, A regularized least squares algorithm for nonlinear rational model identification (submitted for publication).

[23] Orr, M.J.L., 1995, Regularization in the selection of radial basis function center, *Neural Computation* , 7, 606-623.

[24] Ponton, J.W., 1993, The Use of multivariable rational functions for nonlinear data presentation and classification, *Computers and Chemical Engineering*, 17, 1047-1052.

[25] Proll, T. and M.N. Karim, 1994, Model-predictive PH control using real-time NARX approach, *AIChe Journal*, 40, 269-184.

[26] Ratkowsky, D.A., 1987, The statistical properties and utility of rational functions models, *The Canadian Journal of Chemical Engineering*, 65, 845-851.

[27] Sriniwas, G.R., Y.Arkun,.I.Chien, and B.A. Ogunnaike, 1995, Nonlinear identification and control of a high-purity distillation collumn: a case study, *Journal of Process Control*, 5, 149-162.

[28] Thomson, M., S.P.Schooling and M.Soufian, 1996, The practical application of a nonlinear identification methodology, *Control Engineering Practice*, Vol.4, 295-306.

[29] Zhu, Q.M. and S.A.Billings, 1993, Parameter estimation for stochastic nonlinear rational models, *International Journal of Control* , 57, 309-333.

numerator polynomial	true value	estimate
$y(k-1)$	1	1.0000
$u(k-1)u(k-2)$	1	1.0625
$u(k-1)$	1	1.0000
denominator polynomial	true value	estimate
$y^2(k-1)$	1	1.0000
$u^2(k-1)$	1	0.8725

Table 1: Model structure and parameter estimates for Example 1

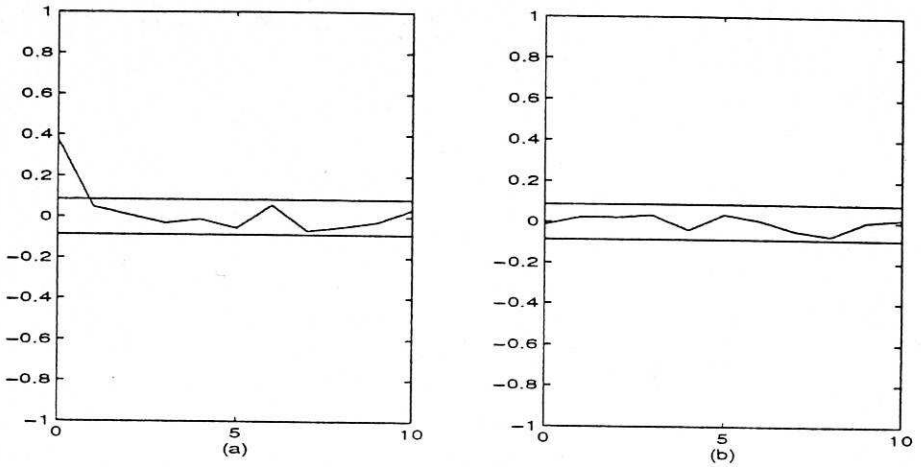

Figure 1: Model validity tests for Example 1 (a —$\Phi_{\epsilon e^2}$ b—$\Phi_{\epsilon u^2}$)

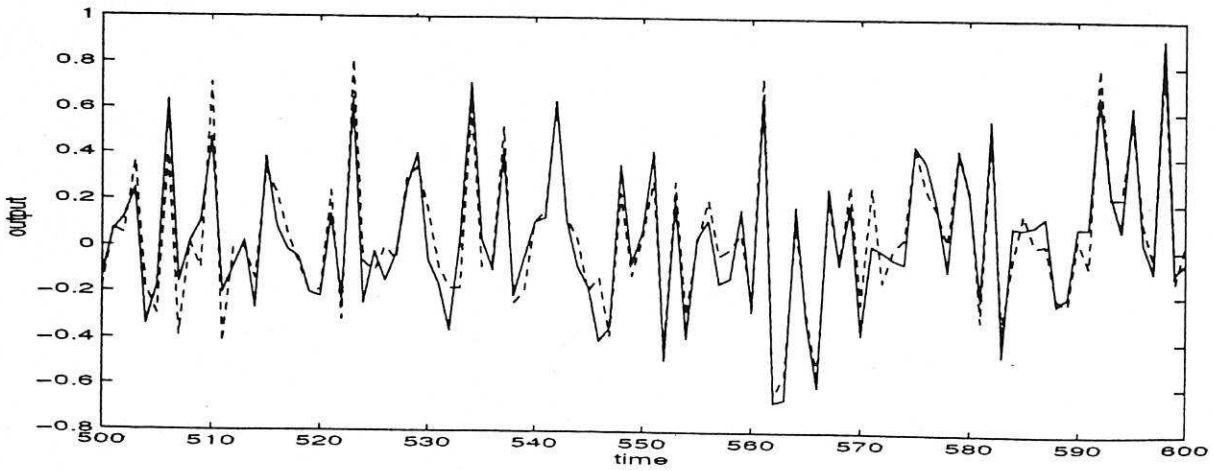

Figure 2: The one-step ahead prediction for Example 1 (dashed line—prediction solid line—measurement)

numerator polynomial	true value	estimate
$y(k-1)y(k-2)$	1	1.0000
$u(k-1)u(k-2)$	1	1.0625
$u(k-1)e(k-1)$	1	1.0000
denominator polynomial	true value	estimate
$y^2(k-1)$	1	1.0000
$y^2(k-2)$	1	1.0000

Table 2: Model structure and parameter estimates for Example 2

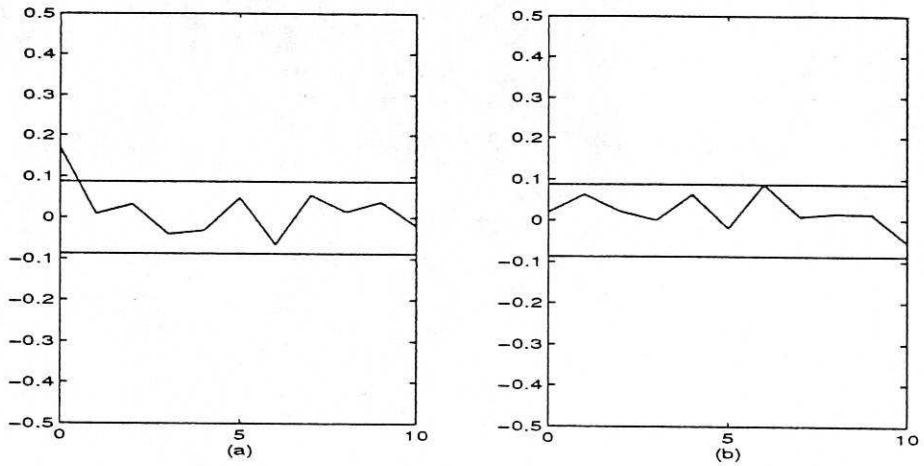

Figure 3: Model validity tests for Example 2 (a —$\Phi_{\epsilon e^2}$ b—$\Phi_{\epsilon u^2}$)

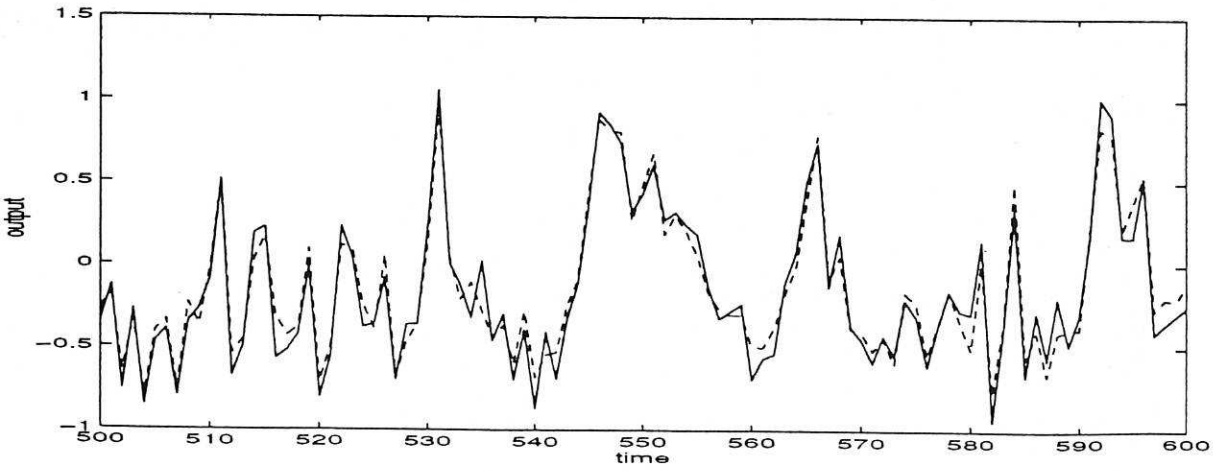

Figure 4: The one-step ahead prediction for Example 2 (dashed line—prediction solid line—measurement)