Variable Selection in Nonlinear Systems Modelling

K.Z.Mao S.A.Billings

Department of Automatic Control and Systems Engineering

University of Sheffield

Mappin Street, Sheffield S1 3JD

United Kingdom

Research Report No. 658

November 27, 1996

University of Sheffield

Variable Selection in Nonlinear Systems Modelling

K.Z.Mao S.A.Billings
Department of Automatic Control and Systems Engineering
University of Sheffield
Sheffield S1 3JD, UK

Abstract

A new algorithm which preselects variables in nonlinear system models is introduced by converting the problem into a variable selection procedure for a set of linearised models. Based on this result an algorithm which consists of a cluster analysis linearisation sub-region division procedure, a linear subset selection routine using an all possible regression algorithm and a genetic algorithm is developed. This algorithm can be applied to the modelling of nonlinear systems using a wide class of model forms including the nonlinear polynomial model, the nonlinear rational model, artificial neural networks and others. Numerical simulations are included to demonstrate the efficiency of the new algorithm.

1 Problem statement

Consider a nonlinear function

$$y = f(x) + e \tag{1}$$

where $x = [x_1, x_2, \ldots, x_n]^T$ is an n-dimensional input vector, y and e denote a scalar output and white noise respectively, and $f(\bullet)$ is some nonlinear function. The objective is to construct a nonlinear model to approximate the underlying relationship $f(\bullet)$ using a series of output and input observations.

Several types of nonlinear model are available for nonlinear approximation. Typical models include nonlinear polynomial models, nonlinear rational models, artificial neural networks and others. But whatever kind of model form is used, the first problem encountered is how to determine which variables should be included in the model. Often many of the variables x_1, x_2, ..., x_n are redundant and only a subset of these variables is significant. Including redundant variables in the model induces at least two problems. First, the model complexity will increase because the size of nonlinear system models increases dramatically with the number of variables. For example, a nonlinear polynomial model with 8 variables and nonlinearity degree 3 contains 165 terms, but a model with 10 variables

and nonlinearity degree 3 comprises 286 terms. Second, including redundant variables leads to a large number of free parameters in the model, and as a consequence the model can tend to be oversensitive to training data and is likely to exhibit poor generalization properties. In the past few years, several approaches have been developed to address the variable selection problem including algorithms based on principal component analysis (PCA) (Oja 1992 and the references therein), mutual information (Battiti 1994, Zheng and Billings 1996) and others. But the disadvatage of using PCA is that the link back to the physical variables of the system is lost.

In linear system modelling, several methods can be used for variable selection, these include hypothesis testing, forward selection, backward elimination *etc.* However these approaches cannot simply be extended to the nonlinear case. For example if the hypothesis testing approach is used to select the input layer variables for a multilayered perceptron (MLP) neural network, this will involve training a large number of neural networks which include the various combinations of the given variables in the input layer. This is an enormous computational burden because the training of just one MLP neural network requires quite a lot of computations. Because of the nonlinear-in-the-parameter (weight) structure of an MLP neural network, forward selection and backward elimination algorithms are not applicable. Although the aforementioned algorithms can be used to detect the model structure of nonlinear polynomial and rational models which have or can be converted into a linear-in-the-parameters structure, these algorithms are in fact term selection and/or deletion methods rather than variable selection and/or deletion algorithms in the nonlinear system case. The distinction between variables and terms in nonlinear models is important and can be illustrated using the nonlinear system $y = a_1x_1^2 + a_2x_1x_2 + a_3x_2^2$. Here x_1 and x_2 are the variables, x_1x_2, x_1^2 and x_2^2 are the terms. Ideally the selection of variables and the determination of terms should be seperated. If the significant variables can be determined initially, the candidate term set, which is produced by performing a nonlinear search over the selected variables, will be reduced and hence the model structure detection procedure will be simplified.

In the present study it is shown that the variable selection problem for nonlinear models can be converted into the variable selection problem for a set of linear models. Based on this result, a novel algorithm which consists of a linearisation sub-region division procedure using a cluster analysis approach together with an all possible regression linear subset selection method and a genetic algorithm is developed. This algorithm can be used for variable selection for a wide range of nonlinear model forms including the nonlinear polynomial model, the nonlinear rational model, neural networks and others. Preselecting the variables in this way enables the terms selection methods to be much more focussed and

leads to simpler and more efficient model structure detection routines. Simulations are included to demontrate the application of the new algorithm.

2 Variable selection algorithm in nonlinear system modelling

2.1 Preliminaries

Consider again the nonlinear function described by

$$y = f(x_1, \ldots, x_n) + e \tag{2}$$

Notice that this is a general model which includes dynamic nonlinear models of the form

$$y(k) = f[y(k-1), \ldots, y(k-n_y), u(k-1), \ldots, u(k-n_u)] + e(k) \tag{3}$$

as a special case, where $y(k-i)$ and $u(k-i)$ are respectively the output and input at the time instant $(k-i)$. For the case of dynamic systems the variable selection problem becomes the lag selection problem.

To develop the new variable selection algorithm, consider initially model eqn (2) with the following assumptions

(i) The variables x_1, x_2, ..., x_n are bounded in an n-dimentional domain denoted by $\mathcal{D}$

$$(x_1, x_2, \ldots, x_n) \in \mathcal{D}$$

(ii) The output y is bounded.

(iii) f is a smooth function so that it has a Taylor expansion.

Assume that one of the operating points is $\mathcal{D}_0 = [x_{10}, \ldots, x_{n0}]^T \in \mathcal{D}$ and that there is a small domain $\Delta\mathcal{D}_0$ around $\mathcal{D}_0$. Linearising the nonlinear function f in the domin $\Delta\mathcal{D}_0 + \mathcal{D}_0 \in \mathcal{D}$, yields

$$\Delta y = \sum_{i=1}^{n} \frac{\partial f(x_1, \ldots, x_n)}{\partial x_i}\Big|_{x_1=x_{10}, \ldots, x_i=x_{i0}, \ldots, x_n=x_{n0}} \Delta x_i + \xi \tag{4}$$

where

$$\Delta y = y - y_0 \tag{5}$$

$$\Delta x_i = x_i + x_{i0} \qquad i = 1, 2, \ldots, n. \tag{6}$$

$$y_0 = f(x_{10}, x_{20}, \ldots, x_{n0})$$

and ξ is the modelling error caused by the linearisation and the noise e.

Substituting eqns (5)-(6) into eqn (4), gives

$$y = a_0 + \sum_{i=1}^{n} a_i x_i + \xi \tag{7}$$

where

$$a_i = \frac{\partial f(x_1, \ldots, x_n)}{\partial x_i}|_{x_1 = x_{10}, \ldots, x_i = x_{i0}, \ldots, x_n = x_{n0}} \qquad i = 1, 2, \ldots, n. \tag{8}$$

$$a_0 = f(x_{10}, \ldots, x_{n0}) - \sum_{i=1}^{n} a_i x_{i0} \tag{9}$$

Comparing eqn (7) with (2) shows that if the variable x_i is relevant to the output in the original nonlinear function, it will make a contribution $a_i x_i - a_i x_{i0}$ to the linearised model. Thus, detecting whether a variable is significant in the nonlinear model can be simplified to checking if it is significant in the linearised model. Consequently the variable selection problem for a nonlinear system can be converted to a variables selection problem for a linear model which can easily be implemented using forward selection and backward deletion algorithms *etc.* Eqns (8) and (9) clearly show that the parameters of the linearised model will be operating region dependent (Billings and Voon 1987), and consequently the significance of variables will also be operating region dependent. The overall significance of a certain variable will therefore have to be evaluated based on its significance in several operating regions.

The above consideration motivates the development of a new variable selection algorithm in the present study. The new algorithm consists of a two-step procedure. First, the whole operating region is divided into several linearisation sub-regions. Second, the significant variables are detected based on the linearised models in the linearisation sub-regions using a linear model subset selection algorithm. Details of the two-step procedure and the implementation of the new algorithm are discussed in §2.2, §2.3 and §2.4 respectively.

2.2 Linearisation sub-region division algorithm

To divide the operating region, each input variable is initially divided into sub-ranges over an appropriate interval. The operating sub-regions are the all possible combinations of these sub-ranges. This will be illustrated below.

First, equally divide the variable x_i into n_i parts

$$\Delta x_i = \frac{x_{imax} - x_{imin}}{n_i}, \quad i = 1, 2, \ldots, n.$$

and obtain the sub-ranges

$$\overbrace{[x_{imin},\ x_{imin} + \Delta x_i)}^{r_{i1}},\quad \overbrace{[x_{imin} + \Delta x_i,\ x_{imin} + 2\Delta x_i)}^{r_{i2}},\ \ldots,\ \overbrace{[x_{imax} - \Delta x_i,\ x_{imax}]}^{r_{in_i}}$$

where

$$x_{imax} = \max\{x_i\}, \quad i = 1, 2, \ldots, n.$$

$$x_{imin} = \min\{x_i\}, \quad i = 1, 2, \ldots, n.$$

Then, combining these sub-ranges determines the operating sub-regions

$$D_k = \{x_i \in r_{ij}, i = 1, 2, \ldots n,\ j = 1, 2, \ldots, n_i\}, \quad k = 1, 2, \ldots.$$

This region division approach is simple and intuitive, and has been successfully applied in the piecewise linear identification of nonlinear systems (Billings and Voon 1987). Because each variable is equally divided, the sub-range interval of each variable must be narrow enough in order to guarantee the modelling accuracy in all operating sub-regions. This is illustrated by a one-dimensional case shown in Fig.1, where the interval must be smaller than or equal to D_5. The narrow sub-range intervals can lead to a large number of sub-

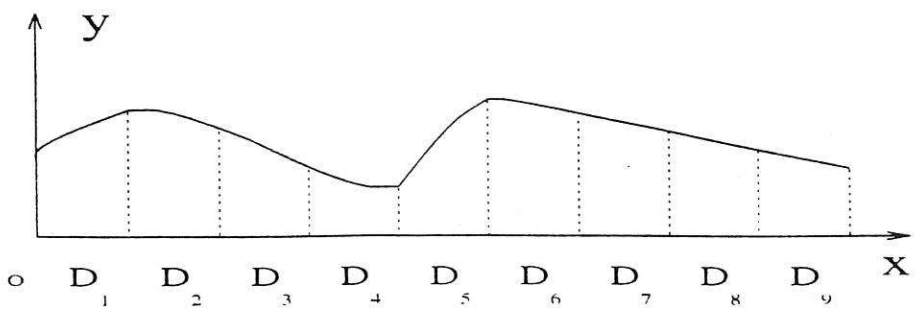

Figure 1: One-dimensional nonlinear system

regions. But a large number of sub-regions makes the samples very scattered and as a result the accuracy of the least squares estimate and the significance evaluation in each sub-regions will tend to degrade. In fact the linearisable regions do not necessarily have to be of the same size some operating sub-regions, for example D_2, D_3 and D_4 in Fig.1, can be merged.

Cluster analysis (Brain 1993) is a technique that partitions a collection of features into a number of subgroups or clusters where the features inside a cluster show a certain degree of similarity. Cluster analysis techniques therefore have the potential to address the sub-regions merging problem. Sub-regions that fit similar linear equations can be put into the same cluster.

The prototype-based algorithm is the most commonly used clustering approach (Hathaway and Bezdek 1995 and the references therein). This approach uses the distance (dis-similarity) measure as the clustering index

$$J = \sum_{i=1}^{m} \sum_{k=1}^{N} d^2[x(k), c_i]\mu_{ki} \tag{10}$$

where $x(k) = [x_1(k), x_2(k), \ldots, x_n(k)]^T$, and $c_i = [c_{i1}, \ldots, c_{im}]^T$ denotes the prototype of the i^{th} cluster. N is the number of samples, and m is the number of clusters. The membership of $x(k)$ belonging to the i^{th} cluster is denoted as μ_{ki} and is either 1 or 0 (in the non-fuzzy case), and $d^2[x(k), c_i]$ represents the Euclidean distance from $x(k)$ to prototype c_i. Distance computation is an important step in cluster analysis. If the prototype is a point, the distance $||x(k)-c_i||^2 = [x(k)-c_i]^T[x(k)-c_i]$ is very easy to compute. But in the present study the prototypes of clusters are linear equations, each of which represents a hypersurface. The computation of the exact distance from a point $x(k)$ to a hypersurface c_i is very difficult to express analytically and a numerical method is required. To overcome this problem a new cluster index is introduced as

$$J_{cluster} = \sum_{i=1}^{m} \sum_{k=1}^{N} [y(k) - \hat{y}(k)]^2 \mu_{ki} \tag{11}$$

where $\hat{y}(k)$ is the one-step ahead output prediction based on a local linear model. Because the samples in the same linearisation area can be represented by a local linear model it is reasonable to use the linear model one-step ahead prediction error as the cluster index in eqn (11).

The difficulty in minimising the cluster index eqn (11) is the determination of a proper

m. While a larger m can lead to a smaller $J_{cluster}$, this will distribute the samples to be more scattered. As a consequence the accuracy of the model parameter estimates and the variable significance evolution in each sub-region will be reduced. A small m is therefore required from the point of view of improving variable selection accuracy. To strengthen this requirement, the cluster index eqn (11) is converted to the following constraint optimization problem

$$min\,\{m\} \tag{12}$$

subject to

$$\frac{\sum_{k=1}^{N_i}[\hat{y}_i(k) - y_i(k)]^2}{\sum_{k=1}^{N_i} y^2(k)} < cutoff \qquad i = 1, 2, \ldots, m. \tag{13}$$

where $cutoff$ is a specified accuracy. $y_i(1)$, $y_i(2)$, ..., $y_i(N_i)$ are the samples in the i^{th} linearisation sub-region, and $\hat{y}_i(k)$ is the one-step ahead prediction of $y_i(k)$ based on the linear model fitted from the samples $y_i(1)$, $y_i(2)$, ..., $y_i(N_i)$.

At this stage it is assumed that the whole operating region has been divided into several linearisation sub-regions denoted by D_1, D_2, ..., D_p using the method in Billings and Voon (1987), and the samples are partitioned into corresponding sub-regions, the objective is to merge these sub-regions under the constraint eqn (13) so that the number of linearisation sub-regions is minimised. Assume that there are two sub-regions D_i and D_j, and the samples in the sub-regions are respectively

$$y_i(1), y_i(2), \ldots, y_i(N_i)$$

$$y_j(1), y_j(2), \ldots, y_j(N_j)$$

Define the merging index (MI)

$$MI(i,j) = \frac{\sum_{k=1}^{N_i}[y_i(k) - \hat{y}_{i(i+j)}(k)]^2 + \sum_{k=1}^{N_j}[y_j(k) - \hat{y}_{j(i+j)}(k)]^2}{\sum_{k=1}^{N_i} y_i^2(k) + \sum_{k=1}^{N_j} y_j^2(k)} \tag{14}$$

where $\hat{y}_{\bullet(i+j)}(k)$ denotes the one-step ahead prediction of $y_\bullet(k)$ based on the linear model fitted from the samples in both D_i and D_j. The two sub-regions D_i and D_j will be merged if and only if the merging index satisfies following merging condition

$$MI(i,j) < cutoff \tag{15}$$

For example assume that the operating region has been divided into linearisation sub-regions which are labeled D_1, D_2, ..., D_p, and samples have been partitioned into the

corresponding sub-regions. At the first step compute the merging index $M(1,j)$, $j = 2,3,\ldots,p$. Merge D_1 with the sub-region that meets the merging condition and has the smallest merging index. Relabel the merged sub-region as D_1, and the others as D_2, D_3, ... Repeat the procedure until D_1 can not merge with any other sub-regions. At the second step, compute $M(2,j)$, $j = 3,4,\ldots$, and merge D_2 with the sub-region that meets the merging condition and has the smallest merging index. Relabel the merged sub-regions as D_2, and the others as D_3, D_4, ... Repeat the procedure until D_2 can not merge with any other sub-regions. The final algorithm can be summarized as follows

(i) Divide each candidate variable into sub-ranges with small intervals and combine these small sub-ranges to produce sub-regions. Label these sub-regions as D_1, D_2, ..., D_p. Partition the samples into corresponding sub-regions and set the current step as $l = 1$.

(ii) Compute $MI(l,q)$ eqn (14), where $q = l+1, l+2, \ldots, p$. Merge D_l with the sub-region that satisfies the two conditions

$$MI(l,q) < cutoff$$

$$\min\{MI(l,q)\}$$

Set $p = p-1$, relabel the merged sub-regions as D_l, and the remainder as D_{l+1}, D_{l+2}, ...

(iii) Repeat step (ii) until D_l can not merge with any other sub-regions.

(iv) Set $l = l+1$, repeat Steps (ii)-(iii) until $l = p$.

(v) Take the sub-regions obtained in step (iv) as the initial sub-regions, repeat steps (ii)-(iv) until the linearisation sub-regions can not merge with each other.

2.3 The variable selection algorithm

Once the linearisation sub-regions are obtained, a linear system variable selection algorithm can be applied to select the variables for the nonlinear system model. Several possible algorithms can be used at this stage including forward selection, backward elimination, stepwise selection, all possible regression *etc.* Among these approaches the all possible regression algorithm (Gunst 1980) is the most comprehensive method of variable selection because this approach provides the smallest number of variables for a defined approximation accuracy. The all possible regression approach is in fact a combinatoral

optimization strategy which selects the model from a model set which contains models of one variable and more. The full model set of a system with n candidate variables is shown in Fig.2, where $\mathcal{S}_i$ denotes the set of models containing i variables. For example the full model set of a system which has three variables labeled as x_1, x_2 and x_3 is

$$\mathcal{S} = \mathcal{S}_1 \cup \mathcal{S}_2 \cup \mathcal{S}_3$$

where

$$\mathcal{S}_1 = \{y = a_i x_i,\ i = 1, 2, 3\}$$

$$\mathcal{S}_2 = \{y = a_i x_i + a_j x_j,\ i = 1, 2,\ j = 2, 3,\ j > i\}$$

$$\mathcal{S}_3 = \{y = a_1 x_1 + a_2 x_2 + a_3 x_3\}$$

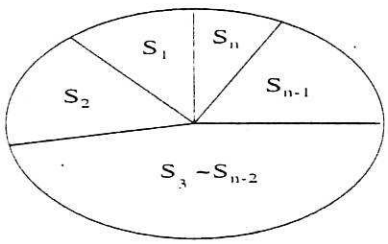

Figure 2: Full model set

Because complete information can only be attained from the use of all possible regression this approach is the only method that can find the best subset given a specified selection criteria. The major drawback of this approach is, however, that a large number of candidate models need to be evaluated. For a model with n variables the number of all possible model subsets amounts to $2^n - 1$, and because of this the all possible regression approach is seldom attempted in practice (Gunst 1980).

In the present study genetic algorithms (Goldberg 1989) will be used to address the combinatorial optimization problem. The main difference between the new algorithm and the standard all possible regression algorithm is that the new algorithm searches for the optimal solution using a random search mechanism rather than by performing an exhaustive search. As a consequence the near-optimal model subset can be found much more efficiently.

The genetic algorithm works with a coding of the parameters rather than with the parameters themselves and therefore the candidate model subset needs to be encoded. Binary

coding is employed in the present study where a 1 is used to represent that the corresponding variable is included in the model, and a 0 is used to denote that the corresponding variable is not included in the model. For example, assuming there are eight possible variables, the model

$$y = a_1 x_1 + a_2 x_5 + a_3 x_6 + a_4 x_8$$

can be represented by the following binary string

$$\overbrace{1}^{x_1}\ \overbrace{0}^{x_2}\ \overbrace{0}^{x_3}\ \overbrace{0}^{x_4}\ \overbrace{1}^{x_5}\ \overbrace{1}^{x_6}\ \overbrace{0}^{x_7}\ \overbrace{1}^{x_8}$$

Genetic algorithms (GAs) consist of three genetic operators, *reproduction*, *crossover* and *mutation*. The application of these operators can be summarised as follows

(i) *Population initialization.* Randomly generate an initial population consisting of l individuals, where each individual represents one model subset, l is a positive integer. Because the number of variables and which variables are significant are all unknown *a priori*, in the initial population set the number and position of the 0's is uniformly and randomly distributed in the ranges $[0, n-1]$ and $[1, n]$ respectively.

(ii) *Population evaluation.* Because the purpose of using the all possible regression algorithm is to find the least necessary variables for a specified accuracy the less the number of variables the better the selection is. Therefore the fitness value should be inversely proportional to the number of variables

$$f \propto 1/n \tag{16}$$

This reverse relation can be mapped using

$$f_i = f_{max} - \frac{f_{max}}{n_{max} - n_{min}}[n_i - n_{min}] \tag{17}$$

where n_{min}, n_{max}, and f_{max} denote the minimum and maximum number of variables and the maximum fitness value respectively. Individuals that do not satisfy the prespecified accuracy are given a zero fitness value.

(iii) *Reproduction.* In the present study the reproduction is implemented as a linear search through a roulette wheel. Each individual is allocated a slot on the roulette wheel subtending an angle proportional to its fitness. A random number in the range 0 to 2π is generated and a copy of a string goes to the mating pool if the random number falls in the slot corresponding to that string. For a population of

size l the reproduction process is repeated l times and l strings go into the mating pool.

(iv) *Crossover.* Two strings are randomly selected from the mating pool. For example consider the two randomly selected parent strings

$$\overset{x_1}{1}\ \overset{x_2}{0}\ \overset{x_3}{0}\ \overset{x_4}{1}\ \overset{x_5}{0}\ \overset{x_6}{0}\ \overset{x_7}{1}\ \overset{x_8}{0} \qquad \text{(A)}$$

$$\overset{x_1}{1}\ \overset{x_2}{0}\ \overset{x_3}{0}\ \overset{x_4}{0}\ \overset{x_5}{1}\ \overset{x_6}{1}\ \overset{x_7}{0}\ \overset{x_8}{1} \qquad \text{(B)}$$

Randomly select the bit from which the two strings are to exchange genes, for example x_5. Exchanging the two strings from the selected position yields

$$\overset{x_1}{1}\ \overset{x_2}{0}\ \overset{x_3}{0}\ \overset{x_4}{1}\ \overset{x_5}{1}\ \overset{x_6}{1}\ \overset{x_7}{0}\ \overset{x_8}{1} \qquad \text{(C)}$$

$$\overset{x_1}{1}\ \overset{x_2}{0}\ \overset{x_3}{0}\ \overset{x_4}{0}\ \overset{x_5}{0}\ \overset{x_6}{0}\ \overset{x_7}{1}\ \overset{x_8}{0} \qquad \text{(D)}$$

Often the above crossover procedure needs to be repeated several times.

(v) *Mutation.* Mutation is a local operator that operates bit by bit which mutates a 1 to a 0 and a 0 to a 1. Consider for example the string

$$\overset{x_1}{1}\ \overset{x_2}{0}\ \overset{x_3}{0}\ \overset{x_4}{0}\ \overset{x_5}{1}\ \overset{x_6}{0}\ \overset{x_7}{1}\ \overset{x_8}{0}$$

If bit x_1 is supposed to mutate, replacing the 1 with a 0 yields the mutated string

$$\overset{x_1}{0}\ \overset{x_2}{0}\ \overset{x_3}{0}\ \overset{x_4}{0}\ \overset{x_5}{1}\ \overset{x_6}{0}\ \overset{x_7}{1}\ \overset{x_8}{0}$$

(vi) *Selection.* Steps (iv)-(v) are repeated $l/2$ times and l offsprings are obtained. These offsprings are evaluated together with the parent strings and the best l are selected as the population of the current generation.

(vii) *Termination.* Steps (ii)-(vi) are repeated until a pre-specified number of generations is reached.

2.4 Implementation of the nonlinear variable selection algorithm

The nonlinear model variable selection algorithm is the combination of the linerization sub-region division procedure and the all possible linear model variable selection procedure:

(i) Divide the whole operating regions into linerization sub-regions and partition the samples into corresponding sub-regions using the procedure developed in §2.2.

(ii) Select the significant variables based on the samples in the linearisation sub-regions using the new all possible regression algorithm developed in §2.3.

The implementation of these operations can be summarised using two functions *division* and *selection*:

$$subregions = division(y, x, part, cutoff)$$

$$variables = selection(y, x, subregions, gawp, cutoff)$$

The function *division* can be used to divide the operting regions into linearisation sub-regions and to partition the samples into corresponding sub-regions. The input parameters of this function are: $y = [y(1), y(2), \ldots, y(N)]^T$ a column vector which consists of observations of the output, $x = [x_1, x_2, \ldots, x_n]$ a matrix which consists of observations of candidate input variables ($x_i = [x_i(1), x_i(2), \ldots, x_i(N)]^T$ are the observations of variable x_i, $i = 1, 2, \ldots, n$), *part* is a scalar which specifies the number of sub-ranges that each variable is divided into, and $cutoff$ is a scalar which specifies the accuracy requirement. The output of the function *division* is the *subregions* which is a matrix.

Once the sub-regions have been obtained using the function *division*, the function *selection* can be used to select variables. The input parameters to the function *selection* are: y and x are the same as in *division*, *subregions* is the output of the function *division*, *gawp* is a row vector which specifies the working parameters of the genetic algorithm ($gawp = [population, mutation, cpn, generation]$, where *population* denotes the population size, *mutation* the mutation rate, *cpn* and *generation* specify the times that the crossover operation repeats and the number of generations that the genetic algorithm evolutes respectively), and $cutoff$ specifies the accuracy requirement. The output of the function *selection* is the significant variables. By successive application of the two functions, significant variables can be selected efficiently.

2.5 Remarks

i) The total linearisation sub-regions are the combination of all sub-ranges of all variables. Therefore the number of variables and the width of the sub-ranges are vitally important parameters that affect the linearisation sub-region division and sample partitions. In order to accurately approximate the nonlinear system using piecewise linear models, the width of each sub-range must be narrow enough and the initial variables must be large enough. But the narrower the sub-range widths and the more variables that are considered the more scattered the samples will be distributed. Although the cluster merging procedure can merge the samples distributed in different sub-regions the procedure will only merge the sub-regions that have more samples than the number of considered variables in order to guarantee the existence of least squares parameter estimates of linearised models. This indicates that many samples will be lost. Because of this the proposed algorithm may be difficult to apply to systems that have a small number of samples and/or have an excessively large number of candidate variables. In practice, the aforementioned two parameters are selected by trial and error.

ii) The new nonlinear model variable algorithm selects the variables based on part of the sampled data set. Model validation tests should be included in the model building procedure to test whether the selected variables are adequate to describe the underlying system.

iii) In §2.1, the nonlinear function is required to be smooth so that a Taylor expansion exists. Because the linearisation equations are fitted from samples in a sub-region not at a point, in practice this restriction can be lifted.

3 Simulation Examples

Example 1

Consider the following nonlinear model

$$y = 10\sin(x_1x_2) + 20(x_3 - 0.5)^2 + 10x_4 - 5x_5 + x_6x_7 + x_7^2 + 5\cos(x_6x_8) + \exp(-\|x_8\|)$$

where y and x_i are the output and input variables respectively, and $\| \bullet \|$ denotes the absolute value. In the simulation, it is assumed that x_1, x_2 and x_5 are independent

random variables uniformly distributed in the range (0 1). The variables x_4, x_6, x_7 and x_8 have the following relationship with the independent variables

$$\begin{cases} x_3 = x_2^2 + 0.5x_1^2 \\ x_4 = x_1 x_5 \\ x_6 = x_3 x_2 \\ x_7 = x_1 x_2 \\ x_8 = x_5^2 \end{cases}$$

The variable selection algorithm was applied to find the most significant variables. The function *division* was first used to divide the linearisation sub-regions. The number that each variable is divided into is an important parameter that affects the linearisation sub-region division. Consider the following four cases (i) *part*=2; (ii) *part*=3; (iii) *part*=4; (iv) *part*=5, where *part* denotes the number that each variable is divided into.

In case (i), of 1000 samples 913 were clustered into 5 linearisation sub-regions. Applying the new all possible regression linear model variable selection algorithm developed in §2.3 to find the significant variables based on the 913 samples three solutions were obtained

$$\text{solution } 1 = \{x_1, x_2, x_5, \text{constant}\}$$

$$\text{solution } 2 = \{x_2, x_5, x_7, \text{constant}\}$$

$$\text{solution } 3 = \{x_4, x_5, x_7, \text{constant}\}$$

In case (ii), of 1000 samples 664 were clustered into 17 sub-regions. Applying the new all possible regression algorithm based on the 664 samples in the 17 sub-regions, the solutions were the same as those in case (i).

In case (iii), of 1000 samples 482 were clustered into 18 sub-regions. The selected variables based on these 482 sample were

$$\text{solution } 4 = \{x_4, x_7, \text{constant}\}$$

In case (iv), of 1000 samples 100 were partitioned into 3 sub-regions. Variable selection based on such a small number of samples is not reliable and this case was therefore not considered.

Using the four solutions as the inputs to four different RBF neural networks each with 20

hidden layer nodes and multiquadratic radial basis functions of unity width four different RBF neural networks were trained using the 1000 samples. The one-step ahead predictions of the four networks were all virtually coincident with the data. The mean squared errors associated with the four neural networks were 0.4910, 0.4265, 1.2919 and 2.3535 respectively. All these show that the solutions in cases (i), (ii) and (iii) provide adequate information.

Example 2

Consider a nonlinear rational model

$$y(k) = \frac{x_2(k) + x_3(k)}{1 + x_1^2(k)} + e(k) \tag{18}$$

where $x_1(k)$, $x_2(k)$ and $x_3(k)$ are the input variables, $y(k)$ and $e(k)$ are the output and a noise sequence respectively. Besides x_1, x_2 and x_3, other four measurable variables are also involved in the system and have following relationship with x_1, x_2 and x_3

$$\begin{cases} x_4(k) = \ln[0.5 + x_3^2(k)] \\ x_5(k) = \sin[x_3(k)] \\ x_6(k) = \exp[-\|x_2(k)\|] \\ x_7(k) = \cos[x_1(k) + x_2(k)] \end{cases}$$

This example is included to illustrate the application of the variable selection algorithm to systems where many variables are involved but where the output is only a function of part of these variables. In the simulation $x_1(k)$, $x_2(k)$ and $x_3(k)$ were independent random variables uniformly distributed in the range (-0.5 0.5), $\{e(k)\}$ was a normally distributed random sequence with zero mean and variance 0.0025. When each variable was initially divided into two sub-ranges, out of 1000 samples 688 were clustered into ten linear sub-regions. When each variable was initially divided into three sub-ranges, out of 1000 samples 161 were clustered into 9 sub-regions. When each variable was initially divided into four sub-ranges, out of 1000 samples only 17 were clustered into 2 linearisation sub-regions with 8 and 9 samples respectively. Because only a small part of the data samples were clustered into linearisation sub-regions in the latter two cases, each variable was initially divided into just two sub-ranges. Applying the new all possible regression variable selection algorithm developed in §2.3 to find the significant variables, two solutions were obtained

$$\text{solution } 1 = \{x_1(k), x_2(k), x_3(k)\}$$

$$\text{solution } 2 = \{x_1(k), x_2(k), x_5(k)\}$$

The two solutions can be used to build the nonlinear rational model. It was assumed that the degree of nonlinerity was 3, a nonlinear search over all the seven variables produced 165 candidate terms for both the numerator and denominator polynomial, but a nonlinear search over the selected variables produced only 20 candidate terms for both the numerator and denominator polynomial. Applying the rational model identification algorithm (Mao and Billings 1996), two models corresponding to the two solutions were obtained

$$y(k) = \frac{0.9919x_2(k) + 0.9977x_3(k)}{1 + 1.0191x_1^2(k)} \tag{19}$$

$$y(k) = \frac{0.9844x_2(k) + 1.0096x_5(k)}{1 + 1.0039x_1^2(k) - 0.148x_5^2(k)} \tag{20}$$

The one-step ahead output predictions of the two models were both excellent. The global model validation tests (Billings and Zhu 1995) are shown in Fig.3 and Fig.4 respectively. The predictions and the model validation tests show that both models provide an excellent representation of the system and that the proposed variable selection algorithm is reliable.

Example 3

Consider a multi-input-multi-output nonlinear system

$$\begin{aligned} y_1(k) &= 0.8y_1(k-1) + u_1(k-2) - 1.2u_1(k-1)u_2(k-2) + 0.4u_1^2(k-2) \\ &\quad -0.1y_2(k-1) + e_1(k) \end{aligned} \tag{21}$$

$$y_2(k) = 0.5y_2(k-1) + u_2(k-2) + u_1^2(k-1) + 0.5y_2(k-2)u_2^2(k-1) + e_2(k) \tag{22}$$

where y_i, u_i and e_i denote the output, input and noise respectively. In this simulation, the inputs $u_1(k)$ and $u_2(k)$ were independent random variables uniformly distributed in the range (0 0.5) and (0 1) respectively, the noise $\{e_1(k)\}$ and $\{e_2(k)\}$ were normally distributed random variables with zero mean and variance 0.01 and 0.04 respectively.

The maximum lag of the inputs and outputs was all initially set to three and each input and output variable was divided into two sub-ranges. Applying the linearisation sub-

regions division algorithm developed in §2.2, of 1000 samples 617 were clustered into 10 sub-regions for subsystem one, and of 1000 samples 456 were clustered into 12 sub-regions for subsystem two. Applying the new all possible regression linear subset selection algorithm developed in §2.3 to the two subsystems respectively produced the following significant lags

$$\text{solution}_{subsystem1} = \{y_1(k-1), u_1(k-1), u_1(k-2), y_2(k-1), u_2(k-2), \text{constant}\}$$

$$\text{solution}_{subsystem2} = \{u_1(k-1), y_2(k-1), y_2(k-2), u_2(k-1), u_2(k-2), \text{constant}\}$$

Assuming a nonlinerity degree of three for both subsystems, a nonlinear search over three lagged variables in all the inputs and outputs produced 455 candidate terms for each subsystem, but a nonlinear search over the preselected lags produced only 56 candidate terms. This shows that variable preselection can considerably reduce the nonlinear term selection problem. Applying the forward regression orthogonal algorithm (Billings *et al* 1988) produced the final models

$$\begin{aligned} y_1(k) &= 0.8041y_1(k-1) + 0.9897u_1(k-2) - 1.2179u_1(k-1)u_2(k-2) \\ &\quad +0.4228u_1^2(k-2) - 0.1006y_2(k-1) \end{aligned} \tag{23}$$

$$\begin{aligned} y_2(k) &= 0.4911y_2(k-1) + 1.0141u_2(k-2) + 0.9494u_1^2(k-1) \\ &\quad +0.5152y_2(k-2)u_2^2(k-1) \end{aligned} \tag{24}$$

A comparison with the original model clearly shows that both the correct model structure and parameter estimates have been obtained and this is confirmed by the one-step ahead output predictions over the first 100 samples shown in Fig.5, and the global model validation tests (Billings and Zhu 1995) shown in Fig.6.

4 Conclusions

Variable selection for nonlinear models is an important and difficult problem in nonlinear system identification. In the present paper it has been shown that this problem can be converted into a variable selection procedure for linearised models. The advantage of this approach is that the significant system variables can be determined and used as candidates for nonlinear model term composition and selection. This reduces the dimensionality of the term selection space but the disadvantage is that sufficient data samples are required

in each of the linear sub-regions and this can mean that the algorithm is difficult to apply if the number of samples is small.

5 Acknowledgement

SAB gratefully acknowledges that part of this work was supported by EPSRC.

References

[1] Battiti, R., 1994, Using mutual information for selecting features in supervised neural net learning, *IEEE Transaction on Neural Networks*, 5(4), 537-550.

[2] Billings, S.A. and W.S.F.Voon, 1987, Piecewise linear identification of non-linear systems, *International Journal of Control*, 46, 215-235.

[3] Billings, S.A., M.J.Korenberg and S.Chen, 1988, Identification of output-affine systems using an orthogonal least squares algorithm, *International Journal of Systems Science*, 19, 1559-1568.

[4] Billings, S.A. and Q.M.Zhu, 1995, Model validation tests for multivariable nonlinear models including neural networks, *International Journal of Control*, 62, 749-766.

[5] Brian, S.E., 1993, *Cluster Analysis*, Third Edition, Edward Arnold, London.

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

[7] Hathaway, R.J. and J.C.Bezdek, 1995, Optimization of cluster criteria by reformulation, *IEEE Transaction on Fuzzy Systems*, 3(2), 241-245.

[8] Korenberg, M.J., S.A.Billings, Y.P.Liu and P.J.Mcilroy, 1988, Orthogonal parameter estimation algorithm for nonlinear stochastic systems, *International Journal of Control*, 48, 193-210.

[9] Mao, K.Z. and S.A.Billings (1996), Algorithms for minimal model structure detection in nonlinear dynamic system identification, submitted for publication.

[10] Gunst, R.F., 1980, *Regression Analysis and Its Application: A Data-oriented Approach*, New York: Dekker, USA.

[11] Oja, E., 1992, Principal components, minor components and linear neural networks, *Neural Networks*, 5(6), 927-935.

[12] Zheng, G.L. and S.A.Billings, 1996, Radial basis function network configuration using mutual information and orthogonal least squares algorithm, *Neural Networks*, in print.

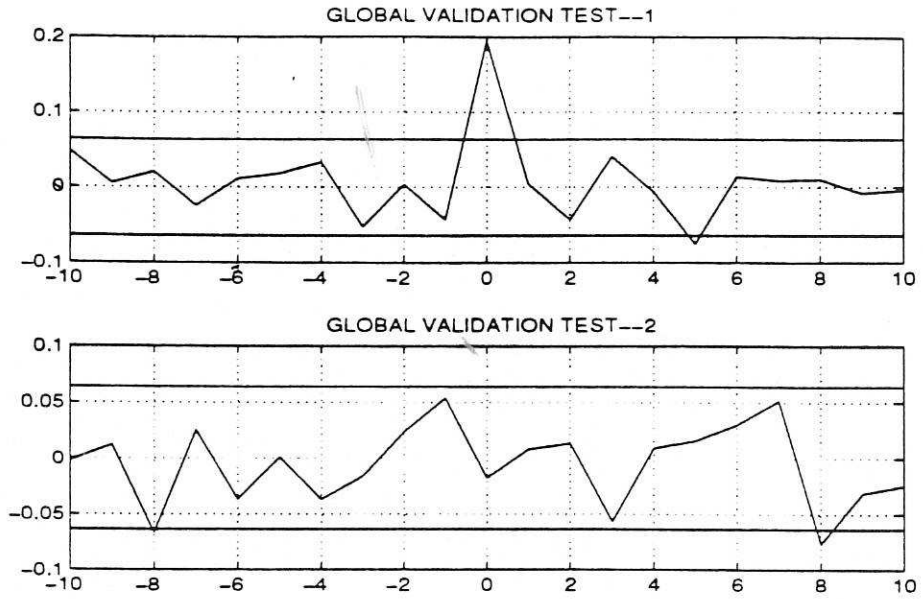

Figure 3: Global model validation tests for Example 2 (solution 1)

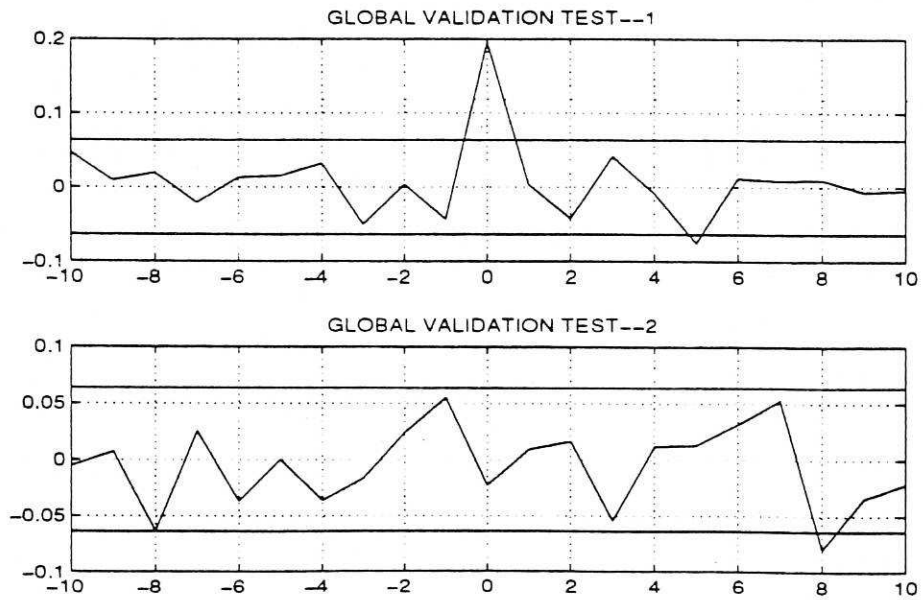

Figure 4: Global model validation tests for Example 2 (solution 2)

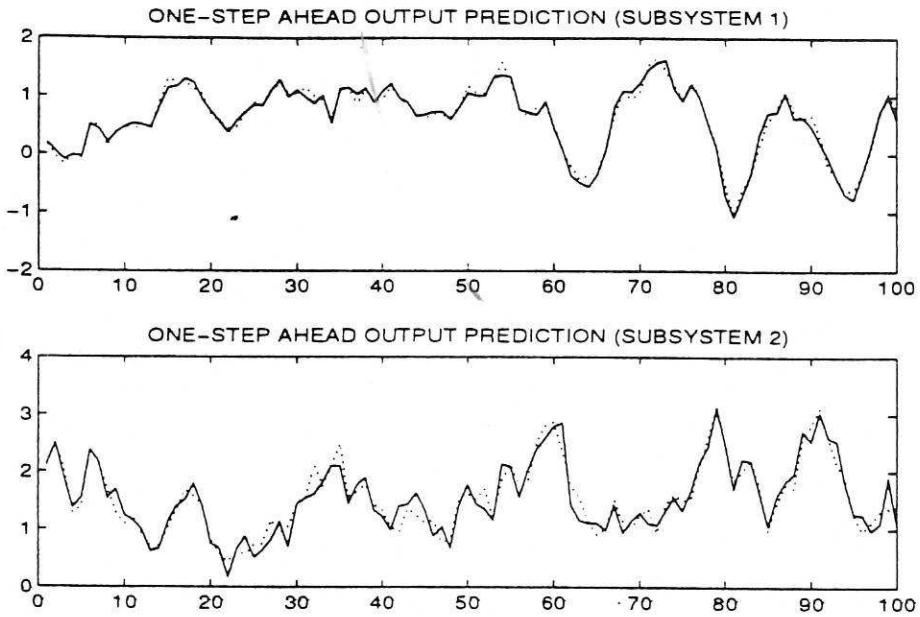

Figure 5: One-step predictions for Example 3 (solid–measurement, dotted–prediction)

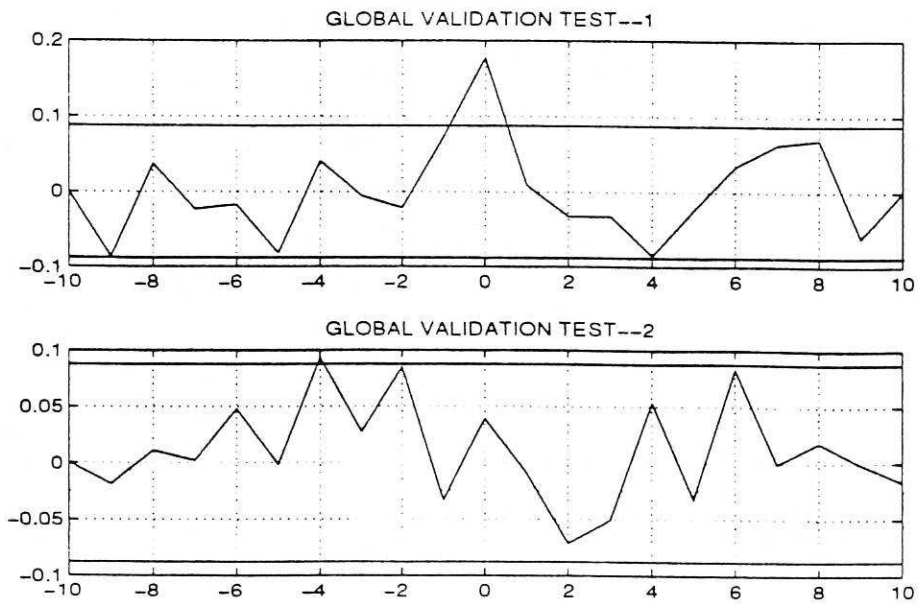

Figure 6: Global model validation tests for Example 3