A Genetic Algorithm Approach to Identification of Nonlinear Polynomial Models

ZI-JIANG YANG, TADASUKE FUJIMOTO and KATSUMI KUMAMARU

Faculty of Computer Engineering and System Science
Kyushu Institute of Technology
Iizuka, Fukuoka, 820-8502 Japan
TEL+81-948-29-7672, FAX+81-948-29-7651
E-Mail yang@cse.kyutech.ac.jp

Abstract— In the last two decades, linear-in-parameter nonlinear polynomial models for NARX (Nonlinear Auto-Regressive with eXogenous input) systems have received considerable attention. The keypoint of polynomial model identification is how to select a set of significant terms employed to approximate the NARX system under study, from a large number of candidates. To this end, the orthogonal least-squares method which is a local search procedure, and the genetic algorithm approach which has a high potential for global optimization have been proposed in the literature. However, it is considered that the methods reported so far in the literature still lack potential to identify the polynomial models with relatively high-order. This limits the applicability of the polynomial models to the real complex nonlinear systems. Motivated by this fact, in this paper, a new genetic algorithm approach to polynomial model identification is proposed. Our contribution in this paper is to introduce a novel hierarchical encoding technique which is considered to be suitable to the structure of the polynomial models. Simulation and application results are also included to verify the efficiency of the proposed identification algorithm. *Copyright ©2000 IFAC*

Keywords: NARX system, nonlinear polynomial model, genetic algorithm, system identification, structure determination, least-squares method.

1.INTRODUCTION

Nonlinearities are inherent in many real systems. It is an important task to construct a nonlinear dynamic system model for prediction, analysis, fault diagnosis and controller design of a real nonlinear system. Therefore, nonlinear system identification has being received more and more attentions in recent years. Especially, identification of an NARX system which describes the input-output behaviour of a nonlinear system has been conducted by a number of researchers [3, 16]. To mimic an NARX system, applications of sigmoidal neural networks, radial basis function networks, fuzzy models have been widely reported in the literature [3, 7, 16] since the last decade. On the other hand, linear-in-parameter nonlinear polynomial models which have shown great potential in their ability in approximating complex nonlinear input-output relationships, have also been studied since 1980s [3, 4, 6, 8, 9, 10, 11, 14]. In [9], it is proved theoretically that the polynomial models can be used to approximate the input-output behaviour of a large class of nonlinear systems over a finite period of time, and a state-affine realization algorithm from a polynomial model is also proposed. Examples of control system design based on polynomial models can be found in [6, 8].

For a polynomial model of order q, the number of possible terms grows drastically with increasing order q and maximum delays of the input and output signals. Fortunately, however, it is often observed that many of the terms do not contribute significantly to the overall goodness of fit, and in practice only a moderate number of terms are necessary to approximate the system behaviour under study. Therefore, a key point of polynomial model identification is how to select a moderate number of necessary monomials from a huge number of candidates. So far, the orthogonal least-squares method based on the Gram-Schmidt orthogonalization procedure which is considered as an effective approach to model structure determination and parameter estimation has extensively been reported in the literatuure [3, 4, 14]. However, it has been pointed out that this algorithm cannot guarantee that the resultant model is globally optimized [15]. Moreover, even with the forward regression orthogonal least-squares method, it is necessary at each step to compare the error reduction ratio of the data vector of a monomial candidate with a huge number of the others in a one-by-one manner. Therefore, in the cases where hundreads or thousands of candidate terms are involved, this approach will become prohibitively expensive as its cost increases super linearly with the number of candidates [10].

Most recently, the genetic algorithm (GA) [12] approach has also been reported in the literature [1, 10, 11]. The GA is a parallel, global, probabilistic search procedure based on the mechanics of natural selection and natural genetics. Because the GA simultaneously evaluates many points in the search space, it can in effect search many local optima and thereby increases the likelihood of finding the global optimum. In recent years, the GA has received considerable attention in various fields, because it has a high potential for global optimization. Studies on the control and identification problems by using the GA can be found for example in [12].

However, when we apply the GA to various engineering problems, it is an important issue to adopt an appropriate encoding method. From this view point, as explained later, the GA approaches reported so far still lack potential of identifying polynomial models of high-orders, since the existing encoding techiniques do not reflect the characteristics of plolynomial models actively. And therefore only low-order models are shown as examples [1, 10, 11]. In practice, however, when we identify a complex nonlinear system in a wide operating region, low-order polynomial models are not sufficient, as claimed in [14]. In this paper, a new GA approach to polynomial model identification is proposed. Our contribution in this paper is to introduce a novel hierarchical encoding technique which is considered to be suitable to the structure of the nonlinear polynomial models. Each layer of the hierarchical chromosome corresponds to the monomials of the same order. And thus genetic operations to the monomials of each order can be performed separately. It is often observed that for a polynomial model of finite order, the low-order monomials are more significant and are hence determined earlier than those of high-orders. Therefore, our proposed hierarchical chromosome is more flexible than a common chromosome in a binary string, so that the mutation probability can be chosen separately according to the monomial orders. Simulation and application examples show that the proposed identification algorithm is quite effective for identifying polynomial models of relatively high-order.

2.STATEMENT OF THE PROBLEM

Consider the following SISO NARX system [3, 4, 6, 8, 9, 10, 11, 14].

$$\begin{aligned} y(t) &= g_0(\phi(t)) + e(t) \\ \phi(t) &= [\phi_1(t), \cdots, \phi_{n_y}(t), \phi_{n_y+1}(t), \cdots, \phi_{n_y+n_u}(t)]^T \\ &= [y(t-1), \cdots, y(t-n_y), \\ &\qquad u(t-1), \cdots, u(t-n_u)]^T \end{aligned} \tag{1}$$

where, $u(t) \in R$, $y(t) \in R$, $e(t) \in R$ are the system input, output and a stochastic noise of zero-mean, respectively. $g_0 : R^d \to R$ is an unknown continuous function (black box) describing the dynamics of the system under study, and $\phi(t) \in R^{d=n_y+n_u}$ is the regressor vector composed of delayed input-output data. n_u and n_y are maximum delays of the input and output respectively.

For identification of the NARX system (1), we approximate the system model via the following qth order Kolmogorov-Gabor polynomial model instead of the well-known neural networks and fuzzy models [3, 4, 6, 8, 9, 10, 11, 14].

$$\begin{aligned} \widehat{g}(\phi(t), \theta) &= \theta_0 + \sum_{i_1=1}^{d} \theta_{i_1}\phi_{i_1}(t) \\ &+ \sum_{i_1=1}^{d}\sum_{i_2=i_1}^{d} \theta_{i_1 i_2}\phi_{i_1}(t)\phi_{i_2}(t) + \cdots \\ &+ \sum_{i_1=1}^{d} \cdots \sum_{i_q=i_{q-1}}^{d} \theta_{i_1 \cdots i_q}\phi_{i_1}(t)\cdots\phi_{i_q}(t) \end{aligned} \tag{2}$$

The polynomial model in equation (2) can be explained as a three-layered functional link network where the monomials 1, $\phi_{i_1}(t)$, $\phi_{i_1}(t)\phi_{i_2}(t)$, $\cdots$, $\phi_{i_1}(t)\cdots\phi_{i_q}(t)$ can be viewed as the hidden layer nodes [3]. The advantage of the polynomial model is that it is linear-in-parametr in contrast to the multilayered neural networks, and hence a number of application examples have been reported [3, 4, 8, 10, 11, 14]. Ususally, when we identify a nonlinear system via neural networks or fuzzy models, it is necessary to determine the maximum delays n_u and n_y of the system prior to starting the identification algorithm. However, systematic procedures for this are seldom studied in the literature. In contrast, in the polynomial model shown in equation (2), since the significant terms are usually automatically determined, the identification procedure still works effectly, if roughly guessed vlaues of n_u and n_y which are larger than their exact values are known in advance.

However, when the polynomial model order q and the dimension d of the regressor vector $\phi(t)$ increase, the maximum number of the candidate terms increases drastically as the following [10].

$$M = \sum_{i=0}^{q} n_i \tag{3}$$

where

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

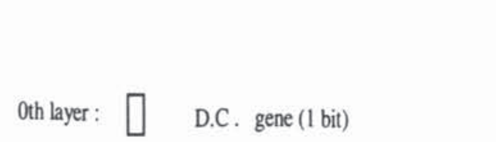

Figure 1: A hierarchical chromosome.

In practice, usually only a moderate number of terms are sufficient to approximate the system under study, and therefore the key point of identification of polynomial models is how to select the significant terms effectively from a huge number of candidates.

So far, as an effective method of model selection and parameter estimation, the orthogonal least-squares method based on the Gram-Schmidt orthogonalization has been widely used for identification of polynomial models [3, 4, 14]. However, it has been pointed out through numerical example that the orthogonal least-squares cannot guarantee global solution [15]. And in recent years, the GA approaches have been reported in the literature [1, 10, 11]. In [10, 11], all the M possible candidates are simply encoded into a binary string of M bits, where 0 and 1 represent deletion and inclusion respectively. And in [1], all the M possible candidates are encoded into a chromosome composed of M integers, and the GA envolves by combining the genetic operations with the orthogonal least-squares method. However, for large polynomial order q and large maximum time delays n_u and n_y, the number M of the candidates may increase to hundreds or thousands, and hence the GA has to be performed based on extremely long chromosomes so that the efficiency of search may degenerate.

3.ENCODING OF THE POLYNOMIAL MODEL

In this section, we introduce a novel hierarchical encoding technique which is considered to be suitable to the structure of the nonlinear polynomial models. Inspection of equation (2) indicates that each monomial in the polynomial model is a combination of $\phi_i(t)(i = 1, \cdots, d)$. Therefore, we encode a monomial into a monomial gene block here. For example, a third-order monomial $\phi_1(t)\phi_2(t)\phi_3(t)$ is encoded into a monomial gene block composed of three genes as 123. Notice that since $\phi_1(t)\phi_3(t)\phi_2(t)$ is equal to $\phi_1(t)\phi_2(t)\phi_3(t)$, to ensure one-to-one encoding, all the genes in a monomial gene block is always sorted in an increasing order, i.e., 132 should be rewritten into 123.

Then the polynomial model (2) can be encoded into a hierachical chromosome as shown in figure 1, where the ith layer subchromosome corresponds to the ith order monomials. The 0th layer includes a D.C. gene of one binary bit, where 0 and 1 represent deletion and inclusion respectively of the constant term θ_0 in equation (2). In the other layers, the first gene in the left end is the L.C. (Length Control) gene, which indicates the number of the monomials of the corresponding order included into the polynomial model. And if the value of L.C. gene is zero, none of the monomials of the

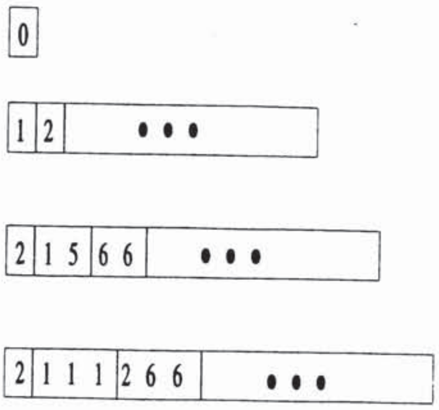

Figure 2: Coding of equation (5).

corresponding order is selected. Since each layer of the hierarchical chromosome corresponds to the monomials of the same order, genetic operations to the monomials of each order can be performed separately. It is often observed that for a polynomial model of finite order, the low-order monomials are more significant and are hence determined earlier than those of high-orders. Therefore, our proposed hierarchical chromosome is more flexible than a common chromosome in a binary string, so that the mutation probability can be chosen separately according to the monomial orders.

As an example, consider the following polynomial model.

$$\begin{aligned} y(t) &= -0.5y(t-2) \\ &+0.7u(t-1)y(t-1) + 0.6u^2(t-2) \\ &+0.2y^3(t-1) - 0.7u^2(t-2)y(t-2) + e(t) \end{aligned} \quad (5)$$

When the maximum delays are set as $n_u = n_y = 4$ (over-determined in this case), the hierarchical chromosome representing this polynomial model is shown in figure 2.

4.IDENTIFICATION ALGORITHM

The details of the concrete algorithm is described here together with some remarks associated with implementation issues.

Step 1:Record of the input-output data

Excite the nonlinear system via an appropriate sequence of input signal, and record the data series $\{u(t), y(t)\}(t = 0, 1, \cdots, t_s + N)$ where N is the data length for parameter estimation and prediction error evaluation. And t_s is chosen such that $t_s \geq \max(n_y, n_u)$.

Step 2 : Generation of the initial population

Generate an initial population of P individuals randomly. The maximum order q of the polynomial model, and the maximum numbers of monomial gene blocks of the subchromosomes in layers $1 \sim q$, i.e., the maximum value of the L.C. genes are determined in advance.

Step 3: Parameter estimation and calculation of the fitness function

Decode each individual into an polynomial model which is suitable for the least-squares method.

$$\begin{aligned} y(t) &= \boldsymbol{z}^T(t)\boldsymbol{c} + e(t) \\ \boldsymbol{z}(t) &= [z_1(t), \cdots, z_L(t)]^T \\ \boldsymbol{c} &= [c_1, \cdots, z_L]^T \end{aligned} \quad (6)$$

where each element $z_l(t)$ $(l = 1, \cdots, L)$ of vector $\boldsymbol{z}(t)$ represents a selected monomial and L is the number of selected monomials.

With the proposed encoding method, it is possible that several elements of vector $\boldsymbol{z}(t)$ share the same monomial gene block, i.e., it is possible that $z_j(t) = z_i(t)$ for $i \neq j$. This can cause numerical problem of the least-squares method. It is considerable to check out the repetitive terms and delete them. However, we have verified that this operation is quite time consuming. In this study, instead of deleting the repetitive terms, we adopt the Moore-Penrose pseudoinverse to avoid the numerical problem [5]. The repetitive terms are expected to disappear when the GA envolves based on the BIC (Bayesian Information Criterion) .

By using the least-squares method, the parameter vector is estimated as follows.

$$\hat{\boldsymbol{c}} = \left[\sum_{t=t_s+1}^{t_s+N} \boldsymbol{z}_t \boldsymbol{z}_t^T\right]^+ \left[\sum_{t=t_s+1}^{t_s+N} \boldsymbol{z}_t y_t\right] \quad (7)$$

where $\boldsymbol{A}^+$is the Moore-Penrose pseudoinverse of matrix $\boldsymbol{A}$.

To evaluate the goodness of the polynomial model, the BIC which takes an appropriate trade-off between the approximation error and model complexity is adopte here.

$$\mathrm{BIC} = N \cdot \ln\left(\frac{1}{N}\sum_{t=t_s+1}^{t_s+N} [\hat{y}(t) - y(t)]^2\right) + L\ln(N) \quad (8)$$

where $\hat{y}(t)$ is the one-step ahead output prediction as the following.

$$\hat{y}(t) = \boldsymbol{z}^T(t)\hat{\boldsymbol{c}} \quad (9)$$

Since the GA reproduces the inidividuals of higher fitness values with higher probabilities, we replace here the BIC by the following so that maximizing the fitness is equivalent to minimizing the BIC.

$$\mathrm{FIT} = Q - \mathrm{BIC} \quad (10)$$

where Q is a constant introduced to make FIT≥ 0. For this purpose, Q is chosen as the minimum BIC among all the individuals of the current generation.

Step 4: Reproduction

In this study, reproduction is performed as a linear search through roulette wheel slots weighted in proportion to the fitness value of each individual. Addtionally, the individual with the best BIC of the last generation is always reproduced.

Step 5: Crossover

Pick up $P/2$ pairs of individuals and perform a crossover operation for each according to the crossover probability P_c.

Crossover is performed between a pair of individuals represented by hierarachical chromosomes. For the D.C. gene of one binary bit, the crossover is carried out by exchange of the one-bit binary number. For the subchromosomes of layers $1 \sim q$, one-point crossover is performed. According to the commutative law of addition, in each subchromosome, no matter how the locations of the selected significant monomial gene blocks are, the decoded model is the same. Therefore, to let the selected significant monomial gene blocks succeed to the offsprings through crossover, prior to the one-point crossover operation between a pair of subchromosomes, common gene blocks (if there are) are relocated to the common positions from the left end. An example of crossover between a pair of subchromosomes is shown in figure 3.

Step 6 : Mutation

Perform the mutation operation on each gene according to the mutation probability $P_{mi}(i = 0, \cdots, q)$

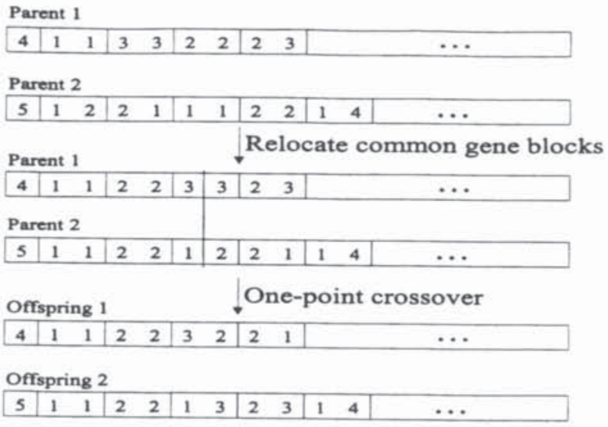

Figure 3: An example of crossover of two subchromosomes.

of each subchromosome. For the D.C. gene, alter the binary number (0 or 1). For the L.C. genes and monomial gene blocks, alter the integer number randomly within the prespecified range. Additionally, when the value of a L.C. gene increases, the number of the selected monomials belonging to the associated subchromosome increases accordingly. And when the value of a L.C. gene decreases, some of the selected monomials belonging to the associated subchromosome are deleted randomly.

Empirical studies show that that for a polynomial model of finite order, the low-order monomials are more significant than those of high-orders and are hence determined earlier. Therefore, it is recommendable to perform the mutation operation with small probabilities for the genes associated with low-order monomials, and with relatively large probabiities for the genes associated with high-order monomials.

Step 7: Parameter estimation and calculation of the fitness function

As described in step 3, estimate the associated parameters and calculate the fitness for each individual of the current generation.

Step 8 : Deletion of some trivial monomial gene blocks

During the evolving process of the GA, besides the siginifant terms, some trivial ones may also be included. Although these trivial terms are expected to disappear as long as the evolving process continues, it is more straightforward to delete some trivial terms directly by intuition. If BIC of the elite individual was not improved in the last 10~20 generations, then try to delete some trivial terms in the elite individual. For each term with a parameter which is nearo zero, the following operation is performed[1] Remove this term and reestimate the parameters of the remained terms and recalculate the BIC. If BIC is improved, it is deleted, otherwise it is kept.

Step 8 : Repetition

Steps 4 $\sim$ 7 are repeated from generation to generation until the termination condition is satisfied.

5.SIMULATION AND APPLICATION RESULTS

Example 1

To show the efficiency of the proposed identification algorithm, we first consider the following nonlinear system governed by a polynomial model.

$$\begin{aligned} y(t) &= -0.5y(t-2) + 0.7u(t-1)y(t-1) + 0.6u^2(t-2) \\ &\quad + 0.2y^3(t-1) - 0.7u^2(t-2)y(t-2) + e(t) \end{aligned} \tag{11}$$

The input signal is $u(t)$ a white noise uniformly distributed in $[-1,1]$, and the noise $e(t)$ is a normally distributed white noise with zero-mean and variance $\sigma_e = 0.1^2$. The design parameters of the identification algorithm are given as follows.

Maximum generation: $G = 200$
Population: $P = 50$
Maximum order of the polynomial model: $q = 5$
Maximum values of the L.C. gene blocks in the subchromosomes (from layer 1 to layer q):
$\boldsymbol{LC}_{max} = [5, 5, 5, 5, 5]$
Crossover probability: $P_c = 0.8$
Mutation probabilities of the subchromosomes (from layer 0 to layer q):
$\boldsymbol{P}_m = [0.03, 0.03, 0.03, 0.1, 0.1, 0.1]$
Data length: $N = 500$
Maximum input-output delays: $n_u = 4$, $n_y = 4$.

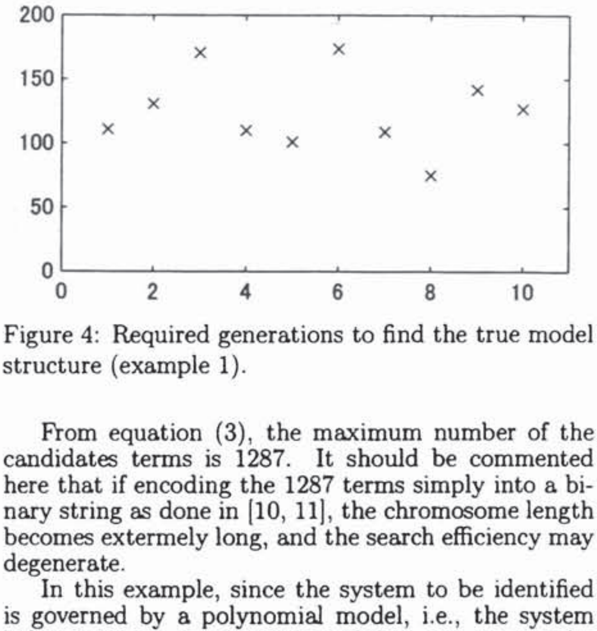

Figure 4: Required generations to find the true model structure (example 1).

From equation (3), the maximum number of the candidates terms is 1287. It should be commented here that if encoding the 1287 terms simply into a binary string as done in [10, 11], the chromosome length becomes extermely long, and the search efficiency may degenerate.

In this example, since the system to be identified is governed by a polynomial model, i.e., the system belongs to the set of model candidates represented by the hierarchical chromosomes, it is expected here that the identification algorithm can find the exact model structure. As expected, the following identified model is obtained.

$$\begin{aligned} y(t) &= -0.500594y(t-2) \\ &\quad + 0.703661y(t-1)u(t-1) + 0.606532u^2(t-2) \\ &\quad + 0.199926y^3(t-1) - 0.694277y(t-2)u^2(t-2) \end{aligned} \tag{12}$$

Next, to show the search efficiency of the identification algorithm, the proposed genetic algorithm is performed 10 times with different randomly generated initial populations, and the required generations to find the true model structure for each simulation are shown in figure 4. It can be verified that the proposed genetic algorithm can find out the true model structure below 200 generations in each case. This fact implies that the proposed algorithm is quite reliable.

[1] In the case where $|y(t)| \gg 1$ and/or $|u(t)| \gg 1$. the contributions of some high-order monomials are still significant even when their parameters approach zero. In this case, it is recommendable to scale the input-output data prior to implement the identification algorithm. See example 3 that appears later.

Example 2

Consider the following system of a nonlinear rational model studied in [13].

$$y(t) = g_0(\boldsymbol{\phi}(t)) + e(t) \tag{13}$$

where

$$\begin{aligned} g_0[\phi_1, \phi_2, \phi_3, \phi_4, \phi_5] &= \frac{\phi_1\phi_2\phi_3\phi_5(\phi_3 - 1) + \phi_4}{1 + {\phi_2}^2 + {\phi_3}^2} \\ \boldsymbol{\phi}(t) &= [\phi_1, \phi_2, \phi_3, \phi_4, \phi_5] \\ &= [y(t-1), y(t-2), y(t-3), u(t-1), u(t-2)] \end{aligned} \tag{14}$$

It has been believed that this is a very complex, strongly nonlinear system and in [13], a four-layered neuralnetwork $\mathcal{N}_{5,20,10,1}$ with $n_u = 2$ and $n_y = 3$ is employed to approximate this system. The network contains 341 parameters and the backpropagation learning algorithm requires 100,000 epochs.

In this study, the input signal $u(t)$ for identification is a white noise uniformly distributed in $[-1.5, 1.5]$, and the noise $e(t)$ is a normally distributed white noise with zero-mean and variance $\sigma_e = 0.05^2$. The design parameters of the identification algorithm are given as follows.

Maximum generation: $G = 200$
Population: $P = 50$
Maximum order of the polynomial model: $q = 8$
Maximum values of the L.C. gene blocks in the subchromosomes (from layer 1 to layer q):
$\boldsymbol{LC}_{max} = [6, 6, 6, 6, 6, 6, 6, 6]$
Crossover probability: $P_c = 0.8$
Mutation probabilities of the subchromosomes (from layer 0 to layer q):
$\boldsymbol{P}_m = [0.03, 0.03, 0.03, 0.06, 0.06, 0.06, 0.06, 0.06, 0.06]$
Data length: $N = 800$
Maximum input-output delays: $n_u = 3, \;\; n_y = 4$.

From equation (3), the maximum number of the candidate terms is 6435. And the following approximated model is obtained.

$$\begin{aligned} y(t) &= 0.867108u(t-1) \\ &-0.28615y^2(t-3)u(t-1) \\ &-0.24383y(t-2)u(t-1)u(t-3) \\ &+0.0886779y(t-2)u^2(t-2) \\ &-0.424154y(t-1)y(t-2)y(t-3)u(t-2) \end{aligned} \tag{15}$$

Very suprisingly, the above identified model is much more parsimonious than the neural network trained in [13]. Furthermore, to evaluate the generalization performance of the identified model, we use the following input to generate the validation data.

$$u(t) = \begin{cases} \sin(2\pi t/250), & t \le 500 \\ 0.8\sin(2\pi t/250) + 0.2\sin(2\pi t/25), & t > 500 \end{cases} \tag{16}$$

Simulations are performed on the exact system model and identified model as follows.

$$y(t) = g_0[y(t-1), y(t-2), y(t-3), u(t-1), u(t-2)] \tag{17}$$

$$\begin{aligned} \widehat{y}(t) &= 0.867108u(t-1) \\ &-0.28615\widehat{y}^2(t-3)u(t-1) \\ &-0.24383\widehat{y}(t-2)u(t-1)u(t-3) \\ &+0.0886779\widehat{y}(t-2)u^2(t-2) \\ &-0.424154\widehat{y}(t-1)\widehat{y}(t-2)\widehat{y}(t-3)u(t-2) \end{aligned} \tag{18}$$

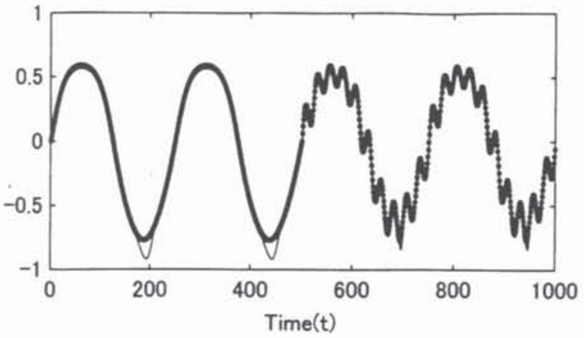

Figure 5: Simulated output (example 2).

The results are shown in figure 5. It can be verified that the identified model while being very simple, achieves similar generalization performance as the much more complex neural network in [13].

Example 3

Finally, as an application example to the real systems, we consider the prediction problem of Box and Jenkins gas furnace data [2]. The input and output data are shown in figures 6 and 7 respectively. Prior to identification, the input and output data are scaled by $\overline{u}(t) = u(t)/|u(t)|_{max}$, $\overline{y}(t) = y(t)/|y(t)|_{max}$, where $|u(t)|_{max}$ and $|y(t)|_{max}$ are maximum values of $|u(t)|$ and $|y(t)|$ respectively. The scaling operation is employed here to prevent the parameters of high-order monomials becoming extremely small. Only the first 200 points of data are used for identification. The design parameters of the identification algorithm are given as follows.

Maximum generation: $G = 200$
Population: $P = 50$
Maximum order of the polynomial model: $q = 6$
Maximum value of the L.C. gene in subchromosome (from layer 1 to layer q):
$\boldsymbol{LC}_{max} = [5, 5, 5, 5, 5, 5]$
Crossover probability: $P_c = 0.8$
Mutation probabilities of the subchromosomes (from layer 0 to layer q):
$\boldsymbol{P}_m = [0.03, 0.03, 0.03, 0.06, 0.06, 0.06, 0.06]$
Data length: $N = 200$
Maximum input-output delays: $n_u = 3, \;\; n_y = 3$.

From equation (3), the maximum number of candidate terms is 924. And the following terms are selected

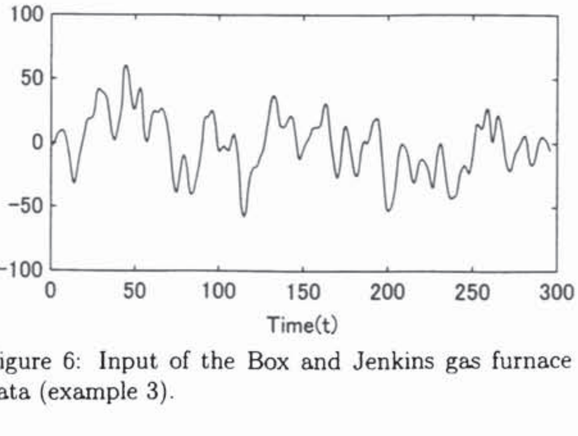

Figure 6: Input of the Box and Jenkins gas furnace data (example 3).

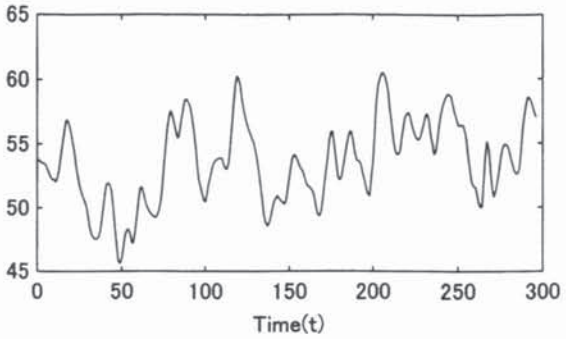

Figure 7: Output of the Box and Jenkins gas furnace data (example 3).

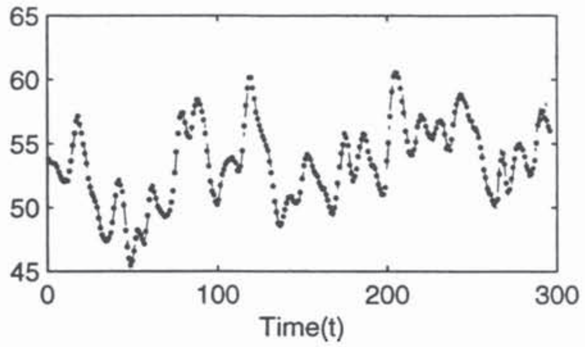

Figure 8: Predicted Output of the Box and Jenkins gas furnace data (example 3).

by the identification algorithm.

$$\begin{aligned}&\overline{y}(t) = 0.302518\overline{y}(t-2) + 0.996265\overline{y}(t-1)\\&-0.424836\overline{y}^2(t-2) + 0.0969916\overline{y}^2(t-1)\overline{y}(t-2)\\&-0.0616081\overline{y}(t-1)\overline{y}(t-3)u(t-3)\\&+0.401202\overline{y}^3(t-2)u(t-2)u(t-3)\\&+0.0270595\overline{y}^2(t-2)\overline{y}^2(t-3)u(t-3)\\&-0.40445\overline{y}^2(t-1)\overline{y}(t-2)u(t-2)u(t-3)\\&-0.0232915\overline{y}^2(t-2)u^2(t-2)u(t-3)\end{aligned} \tag{19}$$

When the one-step ahead prediction $\widehat{\overline{y}}(t)$ of the above identified model is calculated, the prediction of the real output is recovered as $\widehat{y}(t) = |y(t)|_{max}\widehat{\overline{y}}(t)$. The prediction performance of the identified model is shown in figure 8, for not only the first 200 points of data which are used for identification, but also for the other 196 points.

6.CONCLUSIONS

In this paper, a new genetic algorithm approach to polynomial model identification is proposed by a novel hierarchical encoding technique which is considered to be suitable to the structure of the polynomial models. And the efficiency of the proposed identification algorithm is verified through simulation and application examples.

References

[1] S. A. Billings and K. Z. Mao: Algorithms for minimal model structure dection in nonlinear dynamic system identification, International Journal of Control, **68**-2 311/330(1997).

[2] G. E. P. Box and G. M. Jenkins: Time Series Analysis, Forecastingn and Control, San Francisco, CA, Hoilden Day(1970).

[3] S. Chen and S. A. Billings: Neural nerworks for nonlinear dynamic system modelling and identification, International Journal of Control, **56**-2, 319/346(1992).

[4] S. Chen, S. A. Billings and W. Luo: Orthogonal least squares methods and their application to non-linear system identification, International Journal of Control, **50**-5, 1873/1896 (1989).

[5] G. H. Golub and C. F. Van Loan: Matrix Computation, The Johns Hopkins University Press(1989).

[6] J. W. Glass and M. A. Franchek: NARMAX modelling and robust control of internal combustion engines, International Journal of Control, **72**-4, 289/304(1999).

[7] H. Hellendoorn and D. Driankov(Eds.): Fuzzy Model Identification, Springer(1997).

[8] E. Hernandez and Y. Arkun: Control of nonlinear systems using polynomial ARMA models, AIChE Journal, **39**-3, 446/349(1993).

[9] D. Hernando and A. A. Desrochers: Modeling of nonlinear discrte-time systems from input-output data, Automatica, **24**-5, 629/641(1988).

[10] C. J. Li and Y. C. Jeon: Genetic algorithm in identifying nonlinear auto regressive with exogenou input models for monlinear systems, Proceedings of the American Control Conference, 2305/2309(1993).

[11] C. J. Li and Y. C. Jeon: A learning controller based on nonlinear ARX inverse model identified by a genetic algorithm, International Journal of Systems Science, **28**-8, 847/855(1993).

[12] K. F. Man, K. S. Tang, S. Kwong and W. A. Halang: Genetic Algorithms for Control and Signal Processing, Springer (1997).

[13] K. S. Narendra and K. Parthasathy: Identification and control of dynamical systems using neural networks, IEEE Transactions on Neural Networks, **1**-1 4/27(1990).

[14] M. Pottmann, H. Unbehauen and D. E. Seborg: Application of a general multi-model approach for identification of highly nonlinear processes-a case study, International Journal of Control,**57**-1,97/120(1992).

[15] A. Shertinsky and R. W. Picard: On the efficiency of the orthogonal least squares training method for radial basis function networks, IEEE Transactions on Neural Networks, **7**-2, 195/200 (1996).

[16] J. Sjoberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P. Y. Glorennec, H. Hjalmarsson and A. Juditsky: Nonlinear black-box modeling in system identification: a unified overview, Automatica, **31**-12, 1691/1724(1995).