Multiobjective Criteria for Nonlinear Model Selection and Identification with Neural Networks

G.P. Liu and V. Kadirkamanathan

Department of Automatic Control and Systems Engineering
University of Sheffield
P.O. Box 600
Mappin Street
Sheffield S1 4DU
United Kingdom

Research Report No. 508

March 1994

Multiobjective Criteria for Nonlinear Model Selection and Identification with Neural Networks

G.P. Liu and V. Kadirkamanathan
Department of Automatic Control and Systems Engineering,
University of Sheffield, Sheffield S1 4DU, U.K.

Abstract

This paper presents a new approach to model selection and identification of nonlinear systems via neural networks and genetic algorithms, based on multiobjective performance criteria. It considers three performance indices (or cost functions) in the objectives, which are the distance measurement and maximum difference measurement between the real nonlinear system and the nonlinear model, and the complexity measurement of the nonlinear model, instead of a single performance index. The Volterra polynomial basis function network and the Gaussian radial basis function network are applied to approximate the nonlinear system. A numerical algorithm for multiobjective nonlinear model selection and identification using neural networks and genetic algorithms is developed.

1 Introduction

Nonlinear system identification can be posed as a nonlinear functional approximation problem. From the famous Weierstrass Theorem (Powell, 1981) in approximation theory, which is a classical field of mathematics, it is known that the polynomial and many other approximation schemes can approximate arbitrarily well a continuous function. In recent years, many nonlinear system identification approaches, particularly identification using neural networks (Chen et al, 1990; Narendra and Parthasarathy, 1990; Kadirkamanathan, 1991; Chen and Billings, 1992; Qin, et al. 1992; Willis et al. 1992; Kuschewski, et al, 1993), are the application of a similar mathematic machinery.

For nonlinear system identification using approximation, two key questions are how to judge the accuracy for the nonlinear function being approximated and how to choose nonlinear function units to guarantee it. A lot of nonlinear system identification approaches fix the number of nonlinear function units and use only a single performance function, e.g. the 2-norm of the difference between the real nonlinear system and the nonlinear model which results in the well-known least squares algorithm, to measure and judge the accuracy of the identification and to optimize the approximation. However, in

nonlinear system identification there are often a number of objectives to be considered. The objectives are sometimes conflicting and no identification which can be considered best with respect to all objectives exists. Hence, there is an inevitable trade-off between objectives, for example, the distance measurement and maximum difference measurement between the real nonlinear system and the nonlinear model. These considerations lead to the study of multiobjective nonlinear system identification in this paper.
This paper presents a set of multiobjective performance functions to measure the approximation accuracy and complexity of the nonlinear model and uses genetic algorithms to select the nonlinear basis functions in the model to reach a simple nonlinear model. Two neural networks are applied for the model representation of the nonlinear systems. One is the Volterra polynomial basis function (VPBF) network and the other is the Gaussian radial basis function (GRBF) network. It also develops a numerical algorithm for multiobjective nonlinear model selection and identification using neural networks and genetic algorithms. Some examples demonstrate the operation of the algorithm.

2 Nonlinear Modelling by Neural Networks

The modelling of nonlinear systems has been implemented as the problem of selecting an approximate nonlinear function between the inputs and the outputs of the systems. For a single-input single-output system, it can be expressed by the NARMAX model (Nonlinear Auto-Regressive Moving Average model with eXogenous inputs) (Chen and Billings, 1989), i.e.

$$\begin{aligned} y(t) &= f(y(t-1), y(t-2), ..., y(t-n_y), u(t-1), u(t-2), ..., u(t-n_u), \\ &\quad e(t-1), e(t-2), ..., e(t-n_e)) + e(t), \end{aligned} \tag{1}$$

where $f(.)$ is a unknow nonlinear function, y is the output, u the control input and e the noise, respectively, n_y, n_u, n_e are the corresponding maximum delays.
The nonlinear function $f(.)$ in the above NARMAX model can be approximated by a single-layer neural network, i.e. a linear combination of the basis functions.

$$f^*(\mathbf{x}, \mathbf{p}) = \sum_{k=1}^{N} w_k f_k(\mathbf{x}, \mathbf{d}_k), \tag{2}$$

where

$$\begin{aligned} \mathbf{x} &= [y(t-1), y(t-2), ..., y(t-n_y), u(t-1), u(t-2), ..., u(t-n_u), \\ &\quad e(t-1), e(t-2), ..., e(t-n_e)] \end{aligned} \tag{3}$$

$f_k(\mathbf{x}, \mathbf{d}_k)$ $(k = 1, 2, ..., N)$ is the basis function with its parameter vector $\mathbf{d}_k$ and the parameter vector $\mathbf{p}$ containing the weights w_k and the basis function parameter vectors d_k. If the basis functions $f_k(\mathbf{x}, .\mathbf{d}_k)$ do not have the parameters d_k, then it is denoted by $f_k(\mathbf{x})$.

Two sets of basis functions are introduced in this paper: a set of the Volterra polynomial basis functions (VPBF) and a set of the Gaussian radial basis functions (GRBF).
Multivariate polynomial expansions have been suggested as a candidate for the nonlinear modelling using the NARMAX model. The Volterra polynomial expansion has been cast into the frameworks of nonlinear system approximations and neural networks. A network whose basis functions consist of the Volterra polynomials is named as the Volterra polynomial basis function network. Its functional representation is given by

$$\begin{aligned} f(\mathbf{x}) &= f^*(\mathbf{x},\mathbf{p}) + O(\mathbf{x}^3) \qquad (4)\\ f^*(\mathbf{x},\mathbf{p}) &= a + \mathbf{x}^T\mathbf{b} + \mathbf{x}^T\mathbf{C}\mathbf{x} \\ &= a + b_1x_1 + b_2x_2 + ... + c_{11}x_1^2 + c_{12}x_1x_2 + c_{22}x_2^2 + ... \\ &= [a, b_1, b_2, ..., c_{11}, c_{12}, c_{22}, ...][1, x_1, x_2, ..., x_1^2, x_1x_2, x_2^2, ...]^T \\ &= \sum_{k=1}^{N} w_k f_k(\mathbf{x}) \qquad (5) \end{aligned}$$

where

$$[w_1, w_2, w_3, ..., w_{n+2}, w_{n+3}, w_{n+4}, ..., w_N] = [a, b_1, b_2, ..., c_{11}, c_{12}, c_{22}..., c_{nn}], \qquad (6)$$

$$[f_1, f_2, f_3, ..., f_{n+2}, f_{n+3}, f_{n+4}, ..., f_N] = [1, x_1, x_2, ..., x_1^2, x_1x_2, x_2^2, ..., x_n^2] \qquad (7)$$

are the set of linear weights and the set of basis functions being linearly combined, respectively, and $\mathbf{x} \in \mathbb{R}^n$.
Radial basis functions were introduced as a technique for multivariable interpolation (Powell, 1987), which can be cast into an architecture similar to that of the multilayer perceptron. Radial basis function networks provide an alternative to the traditional neural network architectures and have excellent approximation properties. One of the most important radial basis function networks is the Gaussian radial basis function neural network, also called the localised receptive field network. The nonlinear function approximated by the GRBF network is expressed by

$$f^*(\mathbf{x},\mathbf{p}) = \sum_{k=1}^{N} w_k \exp(-(\mathbf{x}-\mathbf{d}_k)^T\mathbf{C}_k(\mathbf{x}-\mathbf{d}_k)) \qquad (8)$$

where $\mathbf{C}_k$ is the weighting matrix of the kth basis function whose centre is $\mathbf{d}_k$, which can trasnform the equidistant lines from being hyperpherical to the hyperellipsoidal, $\mathbf{C}_k = \mathbf{I}$ in this paper, and $\mathbf{p}$ is the parameter vector containing w_k and $\mathbf{d}_k$ $(k = 1, 2, ..., N)$.

3 Nonlinear Model Selection and Identification by Genetic Algorithms

Many different techniques are used today for optimizing the design space associated with various control systems. In recent years, the direct-search techniques, which are problem-independent, have been proposed as a panacea for the difficulties associated with the

traditional techniques. One of these techniques is the genetic algorithm (GA) (Goldberg, 1989; Davis, 1992). Genetic algorithms are search procedures based on the mechanics of natural genetics. They are different from normal search methods encountered in engineering optimization in following ways: a) The GA searches from a population of points, not a single point. b) The GA uses probabilistic and not deterministic transition rules. Recently, genetic algorithms is being applied to control system design (see, e.g., Davis, 1992; Bonseca and Fleming, 1993, Patton and Liu 1994). Thus, this paper applies the GA approach to the model selection and identification of nonlinear systems.

The model selection can be seen as a subset selection problem. For the model represented by the VPBF network, the principle of model selection using the genetic algorithms can be briefly explained as follows. For the vector $\mathbf{x} \in \mathbb{R}^n$, the maximum number of the model terms is given by $N = (n+1)(n+2)/2$. Thus, there are N basis functions which are the combination of 1 and the elements of the vector $\mathbf{x}$. Then there are 2^N possible models for selection. Each model is expressed by an N-bit binary model code $\mathbf{c}$, i.e. a chromosome representation in genetic algorithms. If some bits of the binary model code $\mathbf{c}$ are zeros, it means that the basis functions corresponding to these zero bits are not included in the model.

For example, if the vector $\mathbf{x} \in \mathbb{R}^3$, the maximum number of the model terms is 10. Then there are 1024 possible models. Each model can be expressed by a 10-bit binary model code. Thus the Volterra polynomial basis functions are

$$\mathbf{f} = [f_1, f_2, ..., f_{10}] = [1, x_1, x_2, x_3, x_1x_2, x_2x_3, x_1x_3, x_1^2, x_2^2, x_3^2]. \tag{9}$$

If the 10-bit binary model code is $\mathbf{c} = [1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0]$, the model can be written as

$$\begin{aligned} f^*(\mathbf{x}, \mathbf{p}) &= \mathbf{p}^T \mathrm{diag}(\mathbf{c})\mathbf{f} \\ &= [p_1, p_4, p_7, p_9][f_1, f_4, f_7, f_9]^T \\ &= p_1 + p_4 x_3 + p_7 x_1 x_3 + p_9 x_2^2. \end{aligned} \tag{10}$$

For the model represented by GRBF network, the maximum number of the model terms is given by N, the number of the Gaussian functions, and there are 2^N possible models for selection and also N possible radial basis functions with their centres $\mathbf{d}_k$. Thus a chromosome reprentation in genetic algorithms consists of an N-bit binary model code $\mathbf{c}$ and N real number basis function centres $\mathbf{d}_k$ $(k = 1, 2, ..., N)$, i.e.

$$[\mathbf{c}, \mathbf{d}_1^T, \mathbf{d}_2^T, \ldots, \mathbf{d}_N^T]. \tag{11}$$

For example, if $N = 5$, $x \in \mathbb{R}^2$ and the chromosome

$$[[0 \quad 1 \quad 0 \quad 0 \quad 1], [d_{11}, d_{12}], \ldots, [d_{51}, d_{52}]], \tag{12}$$

then the model is given by

$$f^*(\mathbf{x}, \mathbf{p}) = w_2 \exp(-\sum_{j=1}^{2}(x_j - d_{2j})^2) + w_5 \exp(-\sum_{j=1}^{2}(x_j - d_{5j})^2). \tag{13}$$

It shows from the above that only the basis functions corresponding to the non-zero bits of the binary model code $\mathbf{c}$ are included in the selected model. Given a parent set of binary model codes and basis function parameter vectors, a model satisfying a set of performance criteria is sought by the numerical algorithm in Section 5.

4 Multiobjective Performance Criteria

This section presents multiobjective performance criteria for nonlinear model selection and identification. Let us define the following performance functions:

$$\phi_1(\mathbf{p}) = \| f(\mathbf{x}) - f^*(\mathbf{x}, \mathbf{p}) \|_2 \tag{14}$$

$$\phi_2(\mathbf{p}) = \| f(\mathbf{x}) - f^*(\mathbf{x}, \mathbf{p}) \|_\infty \tag{15}$$

$$\phi_3(\mathbf{p}) = \sigma(\mathbf{c}) \tag{16}$$

where $\| . \|_2$ and $\| . \|_\infty$ are the 2- and ∞-norms of the function (.), $\sigma(\mathbf{c})$ is the number of the non-zero elements in the binary model code $\mathbf{c}$.

For model selection and identification of nonlinear systems, there are excellent reasons for giving attention to the performance functions $\phi_i(\mathbf{p})$ $(i = 1, 2, 3)$. The practical reasons for considering the performance function $\phi_1(\mathbf{p})$ is even stonger than two other performance functions $\phi_2(\mathbf{p})$ and $\phi_3(\mathbf{p})$. Statistical considerations show that it is the most appropriate choice for data fitting when errors in the data have a normal distribution. Often the performance function $\phi_1(\mathbf{p})$ is prefered because it is known that the best approximation calculation is straightforward to solve. The performance function $\phi_2(\mathbf{p})$ provides the foundation of much of approximation theory. It shows that is small, then the performance function $\phi_1(\mathbf{p})$ is small also. But the converse statement may not be true. A practical reason for using the performance function $\phi_2(\mathbf{p})$ is that when in computer calculations, a complicated nonlinear function is estimated by one that is easy to calculate, then it is usually necessary to ensure that the greatest value of the error function is less than a fixed amount, which is just the required accuracy of the approximation. The performance function $\phi_3(\mathbf{p})$ is used to measure the complexity of the model. The small the performance function $\phi_3(\mathbf{p})$ and the simpler the model. Under the same other conditions, the simple model is always better than the complicated one.

In order to give a feel for the usefulness of the multi-objective approach as opposed to single-objective design techniques, let us consider the minimisation of the cost functions $\phi_i(\mathbf{p})$ $(i = 1, 2, 3)$. Let the minimum value of ϕ_i be given by ϕ_i^*, for $i = 1, 2, 3$, respectively. For these optimal values ϕ_i^* there exist corresponding values given by $\phi_j[\phi_i^*]$ $(j \neq i, j = 1, 2, 3)$, for $i = 1, 2, 3$, respectively, and the following relations:

$$\min\{\phi_1[\phi_2^*], \phi_1[\phi_3^*]\} \geq \phi_1^*, \tag{17}$$

$$\min\{\phi_2[\phi_1^*], \phi_2[\phi_3^*]\} \geq \phi_2^* \tag{18}$$

$$\min\{\phi_3[\phi_1^*], \phi_3[\phi_2^*]\} \geq \phi_3^* \tag{19}$$

If one of the performance functions ϕ_i $(i = 1,2,3)$ is minimized individually (single-objective approach), then unacceptably large values may result for other performance functions ϕ_j $(j \neq i,\ j = 1,2,3)$. Generally, there does not exist a solution for all performance function $\phi_i(\mathbf{p})$ for $i = 1,2,3$ to be minimized by the same parameter vector $\mathbf{p}$. Following the method of inequalities (Zakian and Al-Naib, 1973; Liu, 1992; Whidborne and Liu, 1993), we reformulate the optimization into a multiobjective problem as

$$\phi_i(\mathbf{p}) \leq \varepsilon_i, \quad \text{for} \quad i = 1,2,3 \tag{20}$$

where the positive real number ε_i represents the numerical bound on the performance function $\phi_i(\mathbf{p})$ and is determined by the designer.

5 Numerical Algorithm

As we are concerned with three objectives (or cost functions) for model selection and identification, this section develops the numerical algorithms combining genetic algorithm approaches and the method of inequalities to get a numerical solution satisfying the performance criteria.

Now, let us normalize the multiobjective performance functions as the following.

$$\psi_i(\mathbf{p}) = \begin{cases} \dfrac{\phi_i(\mathbf{p})}{\varepsilon_i}, & \text{for} \quad \varepsilon_i \neq 0 \\ \phi_i(\mathbf{p}) + 1, & \text{for} \quad \varepsilon_i = 0 \end{cases} \tag{21}$$

Let Γ_i be the set of parameter vectors $\mathbf{p}$ for which the ith performance criterion is satisfied:

$$\Gamma_i = \{\mathbf{p} : \psi_i(\mathbf{p}) \leq 1\}. \tag{22}$$

Then the admissible or feasible set of parameter vectors for which all the performance criteria hold is the intersection

$$\Gamma = \Gamma_1 \cap \Gamma_2 \cap \Gamma_3. \tag{23}$$

Clearly, $\mathbf{p}$ is an admissible parameter vector if and only if

$$\max\{\psi_1(\mathbf{p}), \psi_2(\mathbf{p}), \psi_3(\mathbf{p})\} \leq 1. \tag{24}$$

which shows that the search for an admissible $\mathbf{p}$ can be pursued by optimization, in particular by solving

$$\min_{\mathbf{p}}\{\max\{\psi_1(\mathbf{p}), \psi_2(\mathbf{p}),, \psi_3(\mathbf{p})\}\} \leq 1. \tag{25}$$

Now, let $\mathbf{p}^k$ be the value of the parameter vector at the kth step, and define

$$\Gamma_i^k = \{\mathbf{p} : \psi_i(\mathbf{p}) \leq \Delta^k\}, \quad \text{for} \quad i = 1,2,3, \tag{26}$$

where

$$\Delta^k = \max\{\psi_i(\mathbf{p}^k)\} \tag{27}$$

and also define

$$\Gamma^k = \Gamma_1^k \cap \Gamma_2^k \cap \Gamma_3^k, \tag{28}$$

$$E^k = \psi_1(\mathbf{p}^k) + \psi_2(\mathbf{p}^k) + \psi_3(\mathbf{p}^k). \tag{29}$$

Γ^k is the kth set of parameter vectors for which all performance functions satisfy

$$\psi_i(\mathbf{p}) \leq \Delta^k, \quad \text{for} \quad i = 1, 2, 3. \tag{30}$$

It is clear that Γ^k contains both $\mathbf{p}^k$ and the admissible set Γ. E^k is a combined measurement of all performance functions. If we find a new parameter vector $\bar{\mathbf{p}}^k$, such that

$$\bar{\Delta}^k < \Delta^k, \tag{31}$$

or

$$\bar{\Delta}^k = \Delta^k \quad \text{and} \quad \bar{E}^k < E^k, \tag{32}$$

where $\bar{\Delta}^k$ and $\bar{E}^k$ are defined similarly to Δ^k and E^k, then we accept $\bar{\mathbf{p}}^k$ as the next value of the parameter vector. Then, we set $\mathbf{p}^{k+1} = \bar{\mathbf{p}}^k$. We then have

$$\psi_i(\mathbf{p}^{k+1}) \leq \psi_i(\mathbf{p}^k), \quad \text{for} \quad i = 1, 2, 3 \tag{33}$$

and

$$\Gamma \subset \Gamma^{k+1} \subset \Gamma^k. \tag{34}$$

So that the boundary of the set in which the parameters are located has been moved towards the admissible set, or, rarely, has remained unaltered. The process of finding the optimization solution is terminated when both Δ_k and E^k cannot be reduced any further. But the process of finding an admissible parameter vector $\mathbf{p}$ is terminated when

$$\Delta^k \leq 1, \tag{35}$$

i.e. when the boundaries of Γ^k have converged to the boundaries of Γ. If the Δ^k persists in being larger than 1, this may be taken as an indication that the performance criteria may be inconsistent, whist their magnitude gives some measure of how closely it is possible to approach the objectives. In this case, some of the performance criteria should be relaxed until they are satisfied. From a practical viewpoint, the approximate optimal solution is also useful if the optimal solution is not achievable.

The steps of the above algorithm to be executed for the GA implementation are as follows:

Step 1: Chromosomal representation
Each chromosome in the population consists of an N-bit binary model code $\mathbf{c}$ and a real number basis function parameter vector $\mathbf{D}$, where N is the number of the basis functions for the nonlinear model selection. For example, for the VPBF network there is not the vector $\mathbf{D}$ and for the GRBF network the vector $\mathbf{D}$ contains all basis function centres $\mathbf{d}_k$ $(k = 1, 2, ..., N)$, i.e. $\mathbf{D} = [\mathbf{d}_1^T, \mathbf{d}_2^T, ..., \mathbf{d}_N^T]$

Step 2: Generation of the initial population
The M chromosomes $[\mathbf{c}, \mathbf{D}]$ for the initial population are randomly generated, where M is an odd number.

Step 3: Evaluation of the performance functions
Given the j-th binary model code $\mathbf{c}_j$ and basis function parameter vector $\mathbf{D}_j$, then the j-th nonlinear model is known. Using the least squares algorithm, the j-th weight vector $\mathbf{w}_j$ can be computed easily, based on the datum of the vector $\mathbf{x}$, the binary model code $\mathbf{c}_j$ and the basis function parameter vector $\mathbf{D}_j$. Then evaluate the normalized performance functions $\psi_i(\mathbf{p}_j)$ $(i = 1, 2, 3)$, where $\mathbf{p}_j = [\mathbf{w}_j, \mathbf{c}_j, \mathbf{D}_j]$, and

$$\Delta_j = \max_{i=1,2,3} \psi_i(\mathbf{p}_j), \tag{36}$$

$$E_j = \sum_{i=1}^{3} \psi_i(\mathbf{p}_j), \tag{37}$$

These above computation are completed for all M sets of chromosomes, i.e. $j = 1, 2, ..., M$.

Step 4: Selection
According to the fitness of the performance functions for each chromosome, delete the $(M-1)/2$ weaker members of the population and reorder the chromosomes. The fitness of the performance functions is measured by

$$F_j = \frac{1}{\Delta_j}, \quad \text{for} \quad j = 1, 2, \ldots, M. \tag{38}$$

Step 5: Crossover
Offspring binary model codes are produced from two parent binary model codes so that their fist half elements are preserved. The second half elements in each parent are exchanged. The average crossover operator is used to produce offspring basis function parameter vectors. The average crossover function is defined as

$$\frac{\mathbf{D}_{j+1} + \mathbf{D}_j}{2}, \quad \text{for} \quad j = 1, 2, \ldots, \frac{M-1}{2}. \tag{39}$$

Then the $(M-1)/2$ offsprings are produced.

Step 6: Mutation

A mutation operator, called a creep, is used. For the binary model codes, it randomly replaces one bit in each offspring binary model code with a random number 1 or 0. For the offspring basis function parameter vectors, the mutation operation is defined as

$$\mathbf{D}_j + \beta\xi_j, \quad \text{for} \quad j = 1, 2, \ldots, \frac{M-1}{2}, \tag{40}$$

where β is the maximum to be altered and $\xi_j \in [-1, 1]$ is a random variable with zero mean.

Step 7: Elitism
The elitist strategy copies the best chromosome into the succeeding generation. It prevents the best chromosome from loss in the next generation. It may increase the speed of domination of a population by a super individual, but on balance it appears to improve genetic algorithm performance. The best chromosome is defined as one satisfying

$$E_b = \min_{l \in \{1,2,\ldots,M\}} \{E_l : E_l \leq E_m - \alpha(\Delta_l - \Delta_m) \quad \text{and} \quad \Delta_l \leq \Delta_m + \delta\}, \tag{41}$$

where

$$\Delta_m = \min_{j=1,2,\ldots,M} \{\Delta_j\}, \tag{42}$$

E_m and E_l are corresponding to Δ_m and Δ_l, which are defined in Eq.(?) and (?), $\alpha > 1$ and δ is a positive number, which are given by the designer (e.g. $\alpha = 1.1$ and $\delta = 0.1$).

Step 8: New offsprings
Add the $(M-1)/2$ new offsprings to the population which are generated in a random fashion. Actually, the new offsprings are formed by replacing randomly some elements of the best binary model code and mutating the best basis function parameter vector with a probability

Step 9: Stop check
Continue the cycle initiated in Step 3 until convergence is achieved. The population is considered to have converged when

$$\Delta_j - \Delta_b \leq \varepsilon \quad \text{for} \quad j = 1, 2, \ldots, (M-1)/2, \tag{43}$$

where Δ_b is corresponding to E_b, and ε is a positive number.

Take the best solution in the converged generation and place it in a second "initial generation". Generate the other $M-1$ chromosomes in this second initial generation at random and begin the cycle again until a satisfactory solution is obtained or Δ_b and E_b cannot be reduced any further.

6 Examples

We use the data generated by a large pilot scale liquid level nonlinear system with zero mean Gaussian input signal (Voon, 1984; Fonseca et al. 1993). 1000 pairs of input-output data were collected. The first 500 pairs were used in the model selection and identification of the system, while the remaining 500 pairs for validation test. The Volterra polynomial basis function network and the Gaussian radial basis function network were applied to select and identify the model of the system by the numerical algorithm developed in Section 5. The parameters for the algorithm are as follows:

Parameter Name	VPBF Network	GRBF Network
model term number N	45	10
chromosome length	45	50
variable vector x	$\begin{bmatrix} y(t-1) \\ y(t-2) \\ y(t-3) \\ y(t-4) \\ u(t-1) \\ u(t-2) \\ u(t-3) \\ u(t-4) \end{bmatrix}$	$\begin{bmatrix} y(t-1) \\ y(t-2) \\ u(t-1) \\ u(t-2) \end{bmatrix}$
ε_1	1.5	1.5
ε_2	0.3	0.3
ε_3	7	7

The VPBF Network

Since the maximum number of the model terms is 45, there are 2^{45} possible models for selections. But, after 210 generations the optimal results has been found approximately by the algorithm. The performance functions are

$$\phi_1(\mathbf{p}) = 1.8000, \quad \phi_2(\mathbf{p}) = 0.3965, \quad \phi_3(\mathbf{p}) = 3. \tag{44}$$

The model represented by the VPBF network is

$$y(t) = 1.3234y(t-1) - 0.3427y(t-2) + 0.075y(t-4)u(t-2). \tag{45}$$

The convergence of the performance functions with respect to generations are given in Figure 1. The measured and estimated outputs, and estimation error of the system are shown in Figure 2. The validity test of the model identified via the VPBF network is illustrated in Figure 3.

The GRBF Network

Although the maximum number of the model terms is only 10 (i.e. 1024 possible models for selection), the search dimension of the basis function centre parameters is 40 in real number space (i.e. infinity possibilities for selection). After 700 generations the performance criteria are almost satisfied. At this stage, $\phi_1(\mathbf{p}) = 1.5643$, $\phi_2(\mathbf{p}) = 0.2511$, $\phi_3(\mathbf{p}) = 5$. In order to obtain the better performance, the basis function parameter vector was searched for another 100 generations using the algorithm with the fixed number of the model terms, i.e. let $\phi_3(\mathbf{p}) = 5$ for this case. Finally, the performance functions are

$$\phi_1(\mathbf{p}) = 1.2957, \quad \phi_2(\mathbf{p}) = 0.1724, \quad \phi_3(\mathbf{p}) = 5. \tag{46}$$

The model represented by the GRBF network is

$$y(t) = \sum_{i=1}^{5} w_i \exp(-\sum_{j=1}^{2}(y(t-j) - d_{ij})^2 - \sum_{j=3}^{4}(u(t-j+2) - d_{ij})^2), \tag{47}$$

where

$$\begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \end{bmatrix} = \begin{bmatrix} -2.6363 \\ -1.2470 \\ -1.7695 \\ 0.9437 \\ -0.5341 \end{bmatrix}, \quad \{d_{ij}\} = \begin{bmatrix} -2.1577 & -1.8855 & -0.8975 & -0.2841 \\ -1.2717 & -2.2730 & 0.3445 & 0.3315 \\ -0.6345 & -1.1223 & -1.1615 & -0.3666 \\ 0.7344 & 1.0223 & 0.5469 & 0.1989 \\ -1.2336 & -0.5928 & 0.3212 & 0.5754 \end{bmatrix} \tag{48}$$

The performance of the GRBF network is shown in Figures. 4-6. Figure 4 shows the convergence of the performance functions with respect to generations. The measured and estimated outputs, and estimation error of the system are given in Figure 5. The validation test of the model identified via the GRBF network is demonstrated in Figure 6.

The selection, identification and validity test results for the large pilot scale liquid level nonlinear system shows that the VPBF network is simpler than the GRBF network, but the performance of the latter is better than that of the former.

7 Conclusions

This paper has been concerned with model selection and identification of nonlinear systems on the basis of neural networks, genetic algorithms and multiobjective optimization techniques. It has proposed a set of multiobjective performance functions to measure the approximation accuracy and complexity of the nonlinear model and uses genetic algorithms to select the nonlinear function units to reach a simple nonlinear model. The Volterra polynomial basis function (VPBF) network and the Gaussian radial basis function (GRBF) are applied for the model representation of the nonlinear systems. It has

developed a numerical algorithm for multiobjective nonlinear model selection and identification. The examples have demonstrated the operation of the algorithm.

Acknowledgements

The authors would like to thank Professor S.A. Billings for his helps and gratefully acknowledge the support of Science and Engineering Research Council (SERC) of UK under the contract GR/J46661.

References

[1] Chen, S. and S.A. Billings," Representations of non-linear systems : the NARMAX model," Int. J. Control, vol. 49, no. 3, pp. 1013-1032, 1989.

[2] Chen, S., S.A. Billings and P.M. Grant, "Nonlinear system identification using neural networks," Int. J. Control, vol. 51, no. 6, pp. 1191-1214, 1990.

[3] Chen, S. and S.A. Billings, "Neural networks for nonlinear dynamic system modelling and identification" Int. J. Control, vol. 56, no. 2, pp. 319-346, 1992.

[4] Davis, L. (Eds.), Handbook of Genetic Algorithms, Van Nostrand Reinhold, New York, 1991.

[5] Fonseca, C.M. and P.J. Fleming, "Multiobjective genetic algorithms," The IEE Colloquium on Genetic Algorithms for Control System Engineering, London, May 1993.

[6] Fonseca, C.M., E.M. Mendes, P.J. Fleming and S.A. Billings, "Nonlinear model term selection with genetic algorithms," Research Report of Dept. of Automatic Control and Systems Engineering, University of Sheffield, U.K., 1993.

[7] Goldberg, D.E., Genetic Algorithm in Search, Optimization, and Machine Learning, Addison Wesley Publishing Company, 1989.

[8] Kadirkamanathan, V., Sequential Learning in Artificial Neural Networks, Ph.D Thesis, University of Cambridge, U.K., 1991.

[9] Kuschewski, J.G., S. Hui and S.H. Zak, "Application of feedforward neural networks to dynamical system identification and control," IEEE Trans. on Control Systems Technology, vol. 1, no. 1, pp. 37-49, 1993.

[10] Liu, G.P., Theory and Design of Critical Control Systems, Ph.D Thesis, University of Manchester Institute of Science and Technology, U.K., 1992.

[11] Narendra, K. S. and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," IEEE Trans. on Neural networks, vol. 1, no.1, pp. 4-27, 1990.

[12] Patton, R.J. and G.P. Liu, "Robust control design via eigenstructure assignment, genetic algorithms and gradient-based optimization," to appear in the IEE Proceedings, Part D, in 1994.

[13] Powell, M.J.D., Approximation Theory and Methods, Cambridge University Press, Cambridge, 1981.

[14] Powell, M.J.D., "Radial basis functions for multivariable interpolation: A review," In J.C. Mason and M.G. Cox, editors, Algorithms for Approximation, pp. 143-167, Oxford University Press, Oxford, 1987.

[15] Qin, S.Z., H.T. Su and T.J. McAvoy, "Comparison of four net learning methods for dynamic system identification," IEEE Trans. on Neural networks, vol. 3, no. 1, pp. 122-130, 1992.

[16] Voon, W.S.F., Nonlinear Parameter Estimation Techniques, PhD Thesis, Dept. Automatic Control and Systems Engineering, University of Sheffield, U.K., 1984.

[17] Whidborne, J.F. and G.P. Liu, Critical Control Systems: Theory, Design and Applications, Research Studies Press Limited, U.K., 1993.

[18] Willis, M.J., G.A. Montague, C. Di. Massimo, M.T. Tham and A.J. Morris, "Artificial neural networks in process estimation and control," Automatica, vol. 28, no. 6, pp. 1181-1187, 1992.

[19] Zakian, V., and U. Al-Naib, "Design of dynamical and control systems by the method of inequalities," Proc. Inst. Elec. Eng., vol. 120, pp. 1421-1427, 1973.

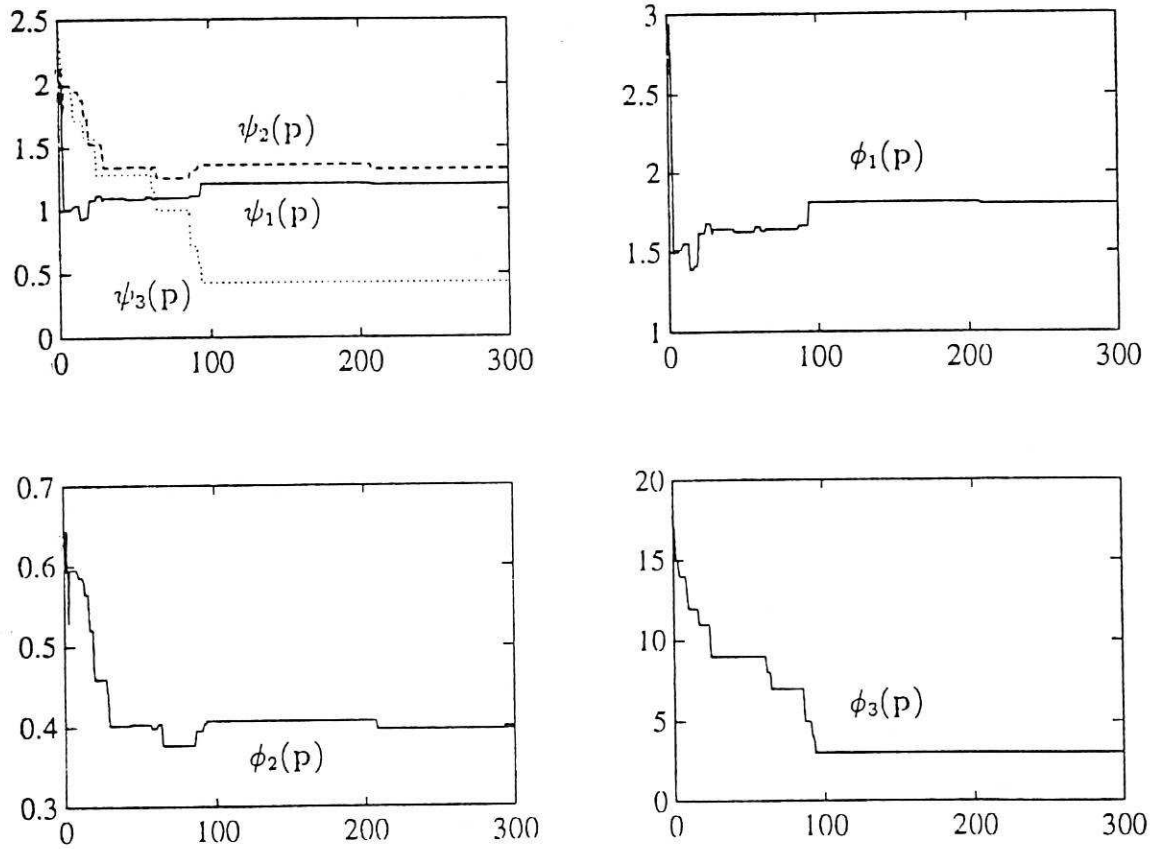
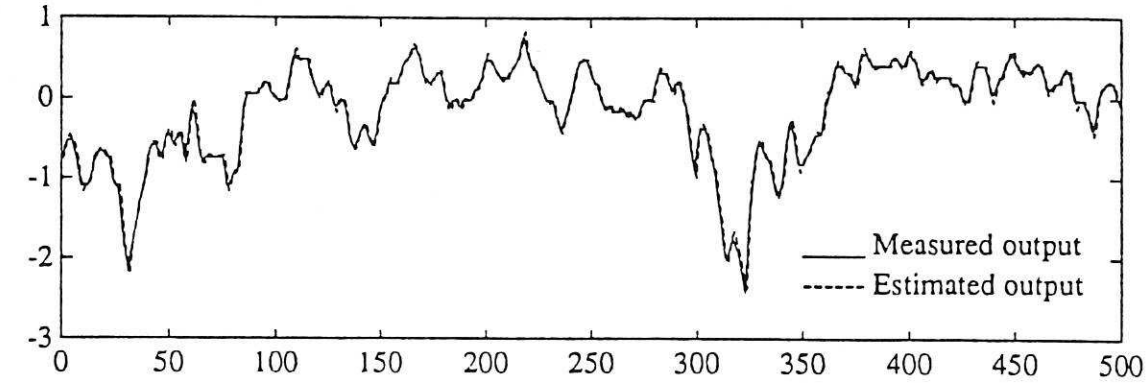
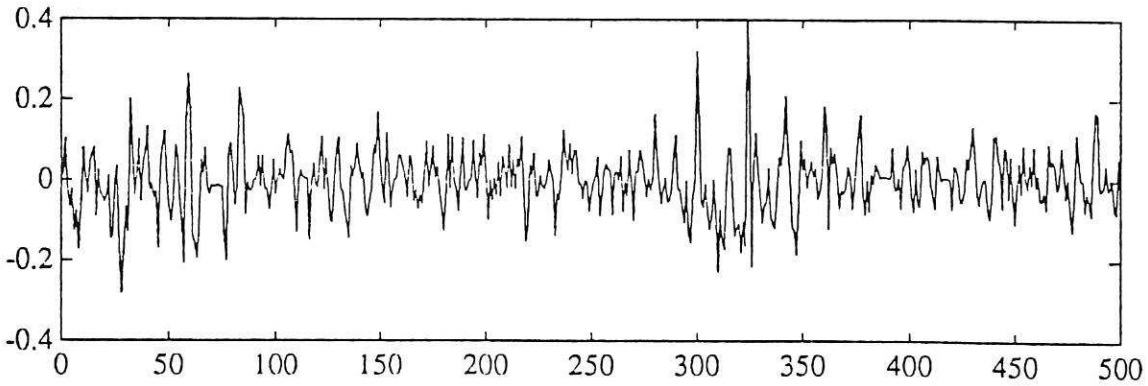


Figure 1: The convergence of the performance functions using VPBF network.

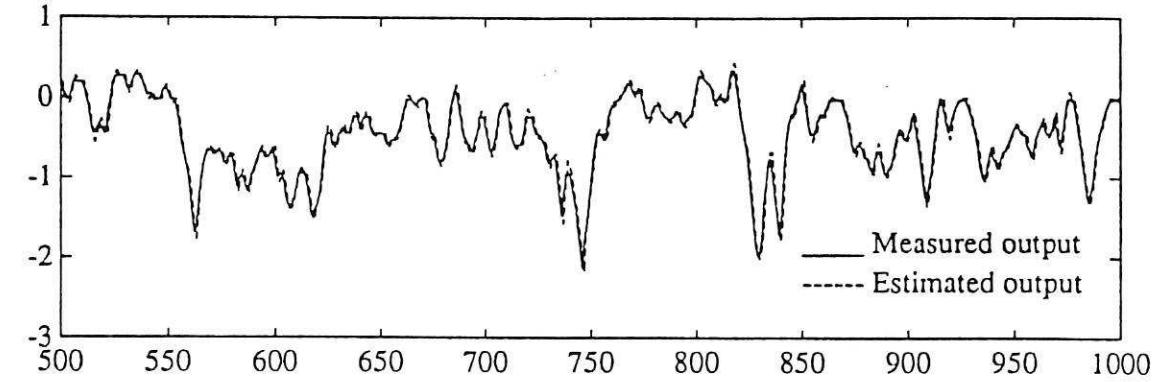


The measured and estimated outputs of the system.

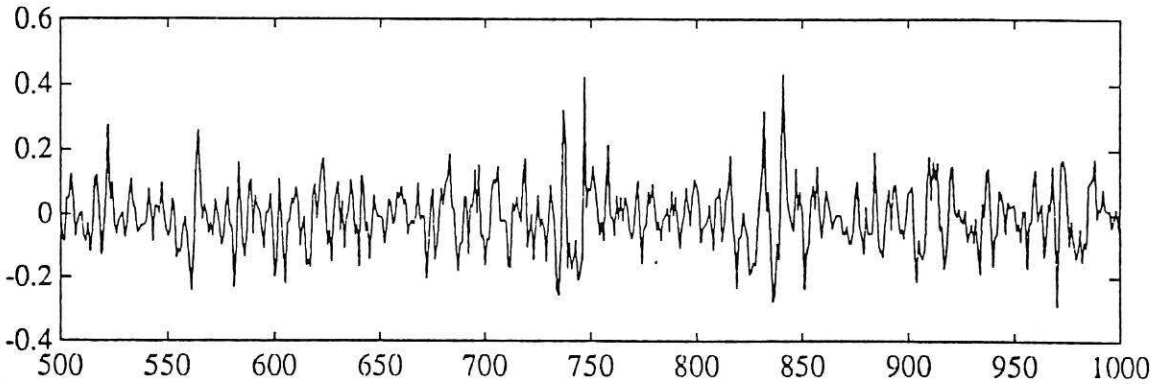


The estimation error of the system.

Figure 2: The selection and identification results of the system using VPBF network.



The measured and estimated outputs of the system.



The estimation error of the system.

Figure 3: The validation results of the system using VPBF network.

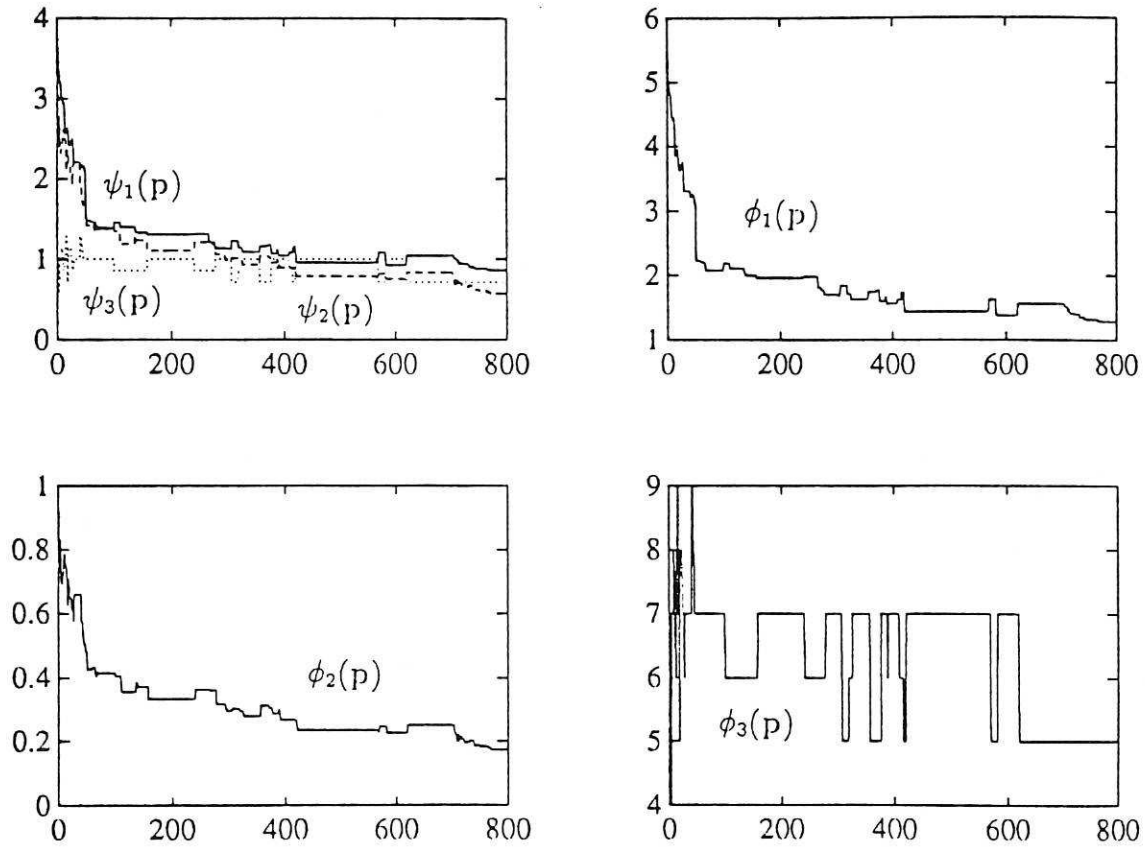
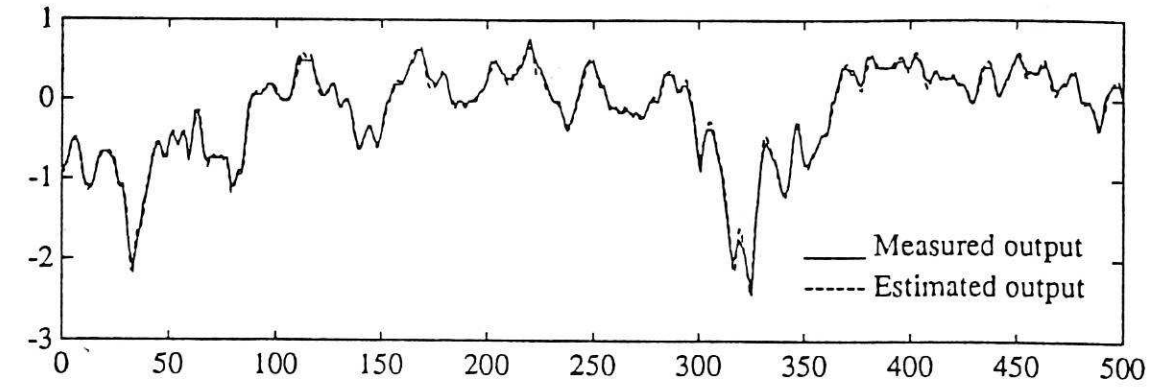
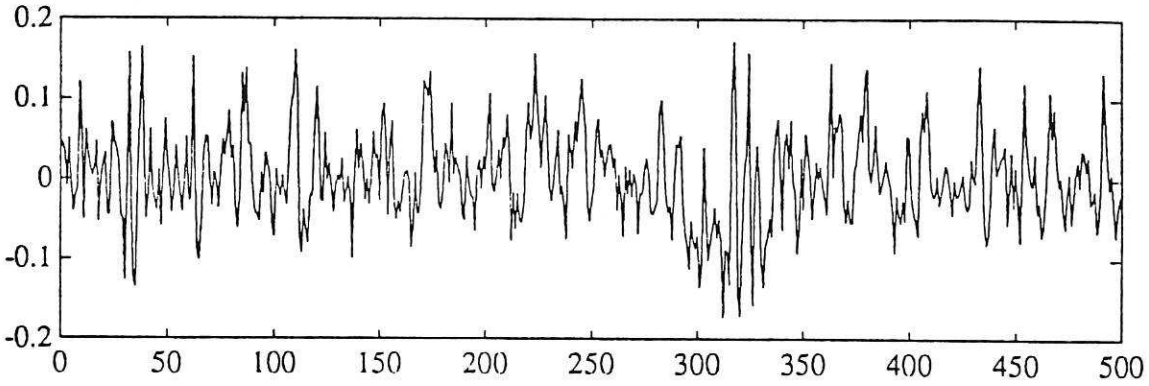


Figure 4: The convergence of the performance functions using GRBF network.

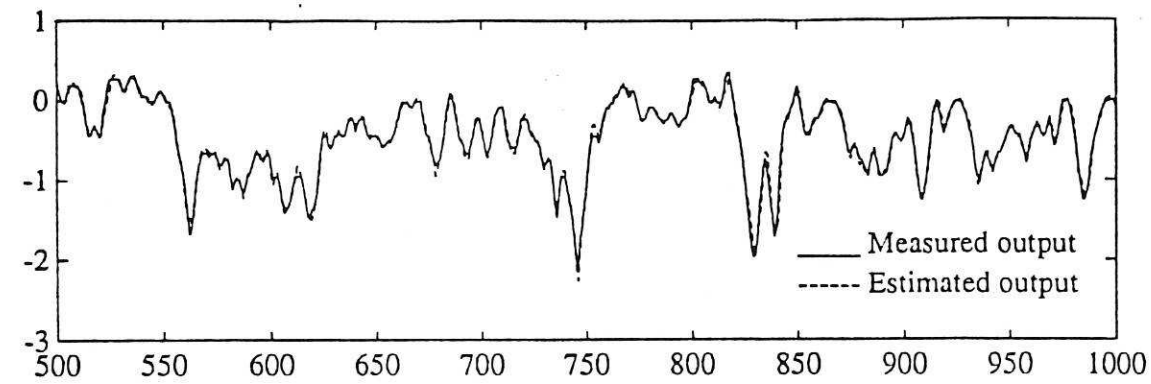


The measured and estimated outputs of the system.

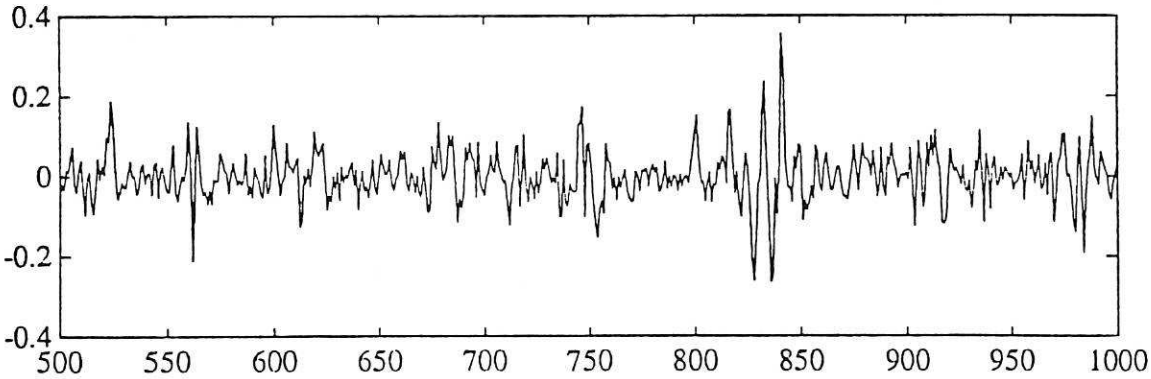


The estimation error of the system.

Figure 5: The selection and identification results of the system using GRBF network.



The measured and estimated outputs of the system.



The estimation error of the system.

Figure 6: The validation results of the system using GRBF network.