Monograph:

Liu, G.P., Kardikamanathan, V. and Billings, S.A. (1996) On-Line Identification of Nonlinear Systems Using Volterra Polynomial Basis Function Neural Networks. Research Report. ACSE Research Report 638 . Department of Automatic Control and Systems Engineering

ON-LINE IDENTIFICATION OF NONLINEAR SYSTEMS USING VOLTERRA POLYNOMIAL BASIS FUNCTION NEURAL NETWORKS

G.P. Liu, V. Kadirkamanathan and S.A. Billings

Department of Automatic Control & Systems Engineering
University of Sheffield, PO Box 600, Mappin Street, Sheffield S1 3JD, UK
Email: guoping/visakan/steve@acse.sheffield.ac.uk

Research Report No. **638**

Department of Automatic Control & Systems Engineering
The University of Sheffield

August 1996

On-Line Identification of Nonlinear Systems Using Volterra Polynomial Basis Function Neural Networks

G. P. Liu, V. Kadirkamanathanand S. A. Billings

Department of Automatic Control and Systems Engineering
University of Sheffield, PO Box 600, Mappin Street, Sheffield S1 3JD, U.K.
Tel: +44 1142825231 Fax: +44 1142731729
Email: guoping/visakan/steve@acse.sheffield.ac.uk

Abstract

An on-line identification scheme using Volterra polynomial basis function (VPBF) neural networks is considered for nonlinear control systems. This comprises of a structure selection procedure and a recursive weight learning algorithm. The orthogonal least squares algorithm is introduced for off-line structure selection and the growing network technique is used for on-line structure selection. An on-line recursive weight learning algorithm is developed to adjust the weights so that the identified model can adapt to variations of the characteristics and operating points in nonlinear systems. The convergence of both the weights and estimation errors is established using a Lyapunov technique. The identification procedure is illustrated using simulated examples.

Keywords: Neural networks, nonlinear system identification, recursive weighting learning, growing network, Volterra polynomials, orthogonal least squares algorithm.

1 Introduction

It is well known that in the past three decades linear models have been widely used in system identification for two major reasons. Firstly, the effects that different and combined input signals have on the output are easily determined. Secondly, linear systems are homogeneous. However, most control systems encountered in practice are nonlinear. In many cases, linear models are not suitable to represent these systems and nonlinear models have to be considered. Since there are nonlinear effects in practical systems, e.g., harmonic generation, intermodulation, desensitization, gain/expansion and chaos, neither of the above principles for linear models are valid for nonlinear systems. Therefore, nonlinear system identification is much more difficult than linear system identification.

The system identification procedure mainly consists of model structure selection and parameter estimation. The former is concerned with selecting which class of mathematical operator is to be used as a model. The latter is concerned with an estimation algorithm and usually requires input-output data from the process, a class of models to be identified and a suitable identification criterion. A number of techniques have been developed in recent years for model selection and parameter estimation of nonlinear systems. Forward and backward regression algorithms were analyzed in [21]. Stepwise regression was used in [4] and a class of orthogonal estimators were discussed in [18]. Algorithms with the objective of saving memory and allowing fast computation have also been proposed in [9] [40]. Methods to determine the a priori structural identifiability of a model have also been studied[24]. A survey of existing techniques of nonlinear system identification prior to the 1980s is given in [2], a survey of the structure detection of input-output nonlinear systems is given in [13] and a recent survey of nonlinear black-box modelling in system identification can be found in [35].

An area of rapid growth in recent years has been neural networks. This approach makes few restrictions on the type of input-output mapping that can be learnt. The application of neural network architectures to nonlinear system identification has been demonstrated by several studies in discrete time (see, for example, [3] [8] [16] [19] [29] [32] [39]) and in continuous time [31] [33] [34]. The majority of nonlinear identification techniques using neural network are off line which means the structure and parameters of the model are fixed after off-line identification based on a set of input-output data. However, if there is a change in the system operation or the real system input space is different from the one which was used for off-line identification, this will lead to changes in the parameters of the neural network based model, causing a deterioration in the performance of the identification. To avoid this, some neural network based identification schemes view the problem as deriving model parameter adjustment laws for the neural network. However, choosing the structure such as the number of basis functions (hidden units in a single hidden layer) in the model must be done *a priori*. This can often lead to an over- or under-determined network structure which in turn results in an identification model that is not optimal. In the discrete-time formulation, some approaches have been developed to determine the number of hidden units (or basis functions) using decision theory [1] and model comparison methods such as *minimum* *description length* [36] and *Bayesian methods* [28]. The problem with these methods is that they require all the observations to be available together and hence are not suitable for on-line identification. Therefore, in order to have good identification performance, both the structure and the parameters of the model need to be modified in response to variations of the plant characteristics and operating point. Recently, new algorithms have been developed which operates on a window of data and which can be used on-line to adaptively track the variations of both model structure [11] [25] or topology [26] [27] and update the estimated parameters or weights on-line.

This paper is mainly concerned with structure selection of nonlinear polynomials in the VPBF network and parameter estimation of the selected model. In order to obtain a proper sized network the orthogonal least squares algorithm is introduced for the off-line structure selection and this is then augmented by the growing network technique which is used for on-line structure selection. In the off-line selection stage, the orthogonal least squares technique is used to select a set of Volterra polynomial basis functions and to arrange the order according to their ability of reducing the approximation error. In the on-line selection, the growing network technique is used to approach gradually the appropriate complexity of the network that is sufficient to provide an approximation to the system to be identified that is consistent with the observations being received. For the parameter estimation, a new on-line recursive weight learning algorithm is developed using a Lyapunov synthesis approach. It is not necessary to assume that the approximation error is white noise or that its upper bound is known. The learning algorithm ensures that the weights and approximation accuracy converge to their required regions.

2 Problem Formulation

Consider the nonlinear discrete system described by

$$X_{t+1} = G(X_t, u_t) \tag{1}$$

$$y_t = h(X_t, u_t) \tag{2}$$

where $G(.)$ is a nonlinear function vector, $h(.)$ a nonlinear function, X_t the state vector, y_t the output and u_t the input. Based on the input and output relation of a system, the above nonlinear discrete system can also be expressed by a NARMA (Nonlinear Auto-Regressive Moving Average) model [20], that is,

$$y_t = f(y_{t-1}, y_{t-2}, \dots, y_{t-n}, u_{t-1}, u_{t-2}, \dots, u_{t-m}) \tag{3}$$

where $f(.)$ is some nonlinear function, n and m are the corresponding maximum delays.

It is well known that neural networks provide a good nonlinear function approximation techniques where the nonlinear function $f(.)$ in the NARMA model can be approximated by a single-layer neural network. This consists of a linear combination of basis functions.

$$\hat{f}(\mathbf{x}_t) = \sum_{k=1}^{N} w_k \phi_k(\mathbf{x}_t) \tag{4}$$

where $\mathbf{x}_t = [y_{t-1}, y_{t-2}, ..., y_{t-n}, u_{t-1}, u_{t-2}, ..., u_{t-m}]$, $\phi_k(\mathbf{x}_t)$ is the basis function and w_k the weight.

According to the universal approximation theorem [14], there exist a finite number of basis functions so that the neural network can approximate the nonlinear function precisely. But, in practice, the problem is how to find these basis functions. Fortunately , it has been shown that the required approximation accuracy can be reached using an adequate number of independent nonlinear basis functions, for example, Volterra polynomial functions, radial functions, B-spline functions or wavelets. In this paper, a neural network which uses the Volterra polynomials as the basis functions will be studied. The representation of the nonlinear function $f(\mathbf{x}_t)$ is then given by

$$\begin{aligned} \hat{f}(\mathbf{x}_t) &= w_1 + w_2 y_{t-1} + w_3 y_{t-2} + ... + w_{n+1} y_{t-n} + w_{n+2} u_{t-1} + ... + w_{n+m+1} u_{t-m} \\ &\quad + w_{n+m+2} y_{t-1}^2 + w_{n+m+3} y_{t-1} y_{t-2} + ... + w_N u_{t-m}^l \\ &= \sum_{k=1}^{N} w_k \phi_k(\mathbf{x}_t) \end{aligned} \tag{5}$$

where

$$\begin{aligned} &[\phi_1, \phi_2, \phi_3, ..., \phi_{n+1}, \phi_{n+2}, ..., \phi_{n+m+1}, \phi_{n+m+2}, \phi_{n+m+3}, ..., \phi_N](\mathbf{x}_t) \\ &\quad = [1, y_{t-1}, y_{t-2}, ..., y_{t-n}, u_{t-1}, ..., u_{t-m}, y_{t-1}^2, y_{t-1}y_{t-2}, ..., u_{t-m}^l] \end{aligned} \tag{6}$$

$$N = \frac{(n+m+l)!}{l!(n+m)!} \tag{7}$$

is the set of the Volterra polynomial basis functions. Using the VPBF network, the nonlinear function $f(.)$ can be obtained by

$$f(\mathbf{x}_t) = \hat{f}(\mathbf{x}_t) + o(\mathbf{x}_t^l) \tag{8}$$

Increasing the order l, the number N of basis functions becomes larger and larger. Thus, the problem is how to estimate the function $\hat{f}(\mathbf{x}_t)$ using a proper sized neural network so that the approximation accuracy is within the required bound. The structure selection and the weight learning of the neural network are discussed in the following sections.

3 Structure Selection for Neural Networks

There are many ways to select the basis functions. Here, off-line structure selection using the orthogonal least squares algorithm [3] [5] and on-line structure selection using growing network techniques are introduced for the basis function selection of Volterra polynomial networks.

3.1 Off-Line Structure Selection

It is assumed that a set of input-output data ($y_t, u_t, t = 1, 2, ..., M$) of the system is given. Based on Eq.(5), the input-output relation may compactly be written in the following vector form:

$$Y = \Phi(\mathbf{x})W + O(\mathbf{x}^l) \tag{9}$$

where the input vector $Y \in \mathbb{R}^{M\times 1}$, the weight vector $W \in \mathbb{R}^{N\times 1}$, the approximation error vector $O(\mathbf{x}^l) \in \mathbb{R}^{M\times 1}$ and the basis function matrix $\Phi(\mathbf{x}) \in \mathbb{R}^{M\times N}$ are, respectively,

$$Y = [\, y_1 \quad y_2 \quad ... \quad y_M \,]^T \tag{10}$$

$$W = [\, w_1 \quad w_2 \quad ... \quad w_N \,]^T \tag{11}$$

$$O(\mathbf{x}^l) = [\, o(\mathbf{x}_1^l) \quad o(\mathbf{x}_2^l) \quad ... \quad o(\mathbf{x}_M^l) \,]^T \tag{12}$$

$$\Phi(\mathbf{x}) = \begin{bmatrix} \phi_1(\mathbf{x}_1) & \phi_2(\mathbf{x}_1) & \dots & \phi_N(\mathbf{x}_1) \\ \phi_1(\mathbf{x}_2) & \phi_2(\mathbf{x}_2) & \dots & \phi_N(\mathbf{x}_2) \\ \vdots & \vdots & \dots & \vdots \\ \phi_1(\mathbf{x}_M) & \phi_2(\mathbf{x}_M) & \dots & \phi_N(\mathbf{x}_M) \end{bmatrix} \tag{13}$$

An orthogonal decomposition of the matrix $\Phi(\mathbf{x})$ gives

$$\Phi(\mathbf{x}) = PQ \tag{14}$$

where $P = [P_1, P_2, ..., P_N]$ is an $M \times N$ matrix with orthogonal columns and Q is an $N \times N$ unit upper triangular matrix. Using the above, Eq. (9) can be written as

$$Y = PV + O(\mathbf{x}^l) \tag{15}$$

$$W = Q^{-1}V \tag{16}$$

where $V = [v_1, v_2, ..., v_N]^T \in \mathbb{R}^{N\times 1}$. It can be seen that the optimal estimate $V^+ = [v_1^+, v_2^+, ..., v_N^+]^T$ of the vector V is

$$v_i^+ = \frac{Y^T P_i}{P_i^T P_i}, \quad \text{for} \quad i = 1, 2, ..., N \tag{17}$$

so that $||Y - PV^+||_2$ is minimal. The corresponding weight vector is $W^+ = Q^{-1}V^+$. The error reduction ratio due to P_i may be expressed by [5]

$$r_i = \frac{v_i^+ P_i^T Y}{Y^T Y} \tag{18}$$

It is clear from Eqs. (17) and (18) that $r_i \geq 0$. Changing the order of the VPBFs will lead to a change in the error reduction ratio r_i. For N VPBFs, there are $N!$ sorting possibilities. Let the $r_i^{(j)}$ denote the error reduction ratio r_i corresponding to the j-th sorting of the VPBFs. The methods proposed in [5] [6] can be used to find the o-th sorting of the basis functions $\phi_1(\mathbf{x}_t), \phi_2(\mathbf{x}_t), ..., \phi_N(\mathbf{x}_t)$, which is the best sorting, such that

$$\sum_{i=1}^{k} r_i^{(o)} \geq \sum_{i=1}^{k} r_i^{(j)} \quad \text{for} \quad j \neq o, j = 1, 2, ..., N!, \quad k = 1, 2, ..., N \tag{19}$$

In this way, the priority of all candidates is determined. Thus, denote the best sorting of VPBFs by $\phi_1^o(\mathbf{x}_t), \phi_2^o(\mathbf{x}_t), ..., \phi_N^o(\mathbf{x}_t)$. The corresponding weight vector is denoted by W^o.

3.2 On-Line Structure Selection

For nonlinear systems, the system operation can change with time or the real system input space is different from the one which was used for off-line identification. In order to produce good identification performance, both the structure and the weights of the neural network model may need to be modified in response to variations in the plant characteristics. Here, the modification of the neural network structure will be taken into account. The adaptation of the weights will be discussed in the next section.

According to approximation theory, adding more independent basis functions to the network will improve approximation. In off-line structure selection, the VPBFs were reordered in terms of their priority. Here it is assumed that at time $t-1$ the basis functions of the VPBF network consist of the first L best candidates $\phi_1^o(\mathbf{x}_t), \phi_2^o(\mathbf{x}_t), ..., \phi_L^o(\mathbf{x}_t)$. To improve the approximation accuracy, the growing network technique [15] [17] [22] [23] is applied. This means that one more VPBF, which is chosen from and is of the highest priority in the remaining basis function candidates $\phi_{L+1}^o(\mathbf{x}_t), \phi_2^o(\mathbf{x}_t), ..., \phi_L^o(\mathbf{x}_t)$, needs to be added to the network. In this case, denote the structure of the VPBF network at time $t-1$ as $\hat{f}^{[t-1]}(\mathbf{x}_t)$ and the structure immediately after the addition of a basis function at time t as $\hat{f}^{[t]}(\mathbf{x}_t)$. Based on the growing network technique and the structure of the function $\hat{f}(\mathbf{x}_t)$ in Eq.(4), the structure of the VPBF network now becomes,

$$\hat{f}^{[t]}(\mathbf{x}_t) = \hat{f}^{[t-1]}(\mathbf{x}_t) + w_{L+1}^o \phi_{L+1}^o(\mathbf{x}_t) \tag{20}$$

where w_{L+1}^o is the weight corresponding to the new $(L+1)^{th}$ Volterra polynomial basis function $\phi_{L+1}^o(\mathbf{x}_t)$.

The growing VPBF network is initialised with a small set of Volterra polynomial basis function units, which are normally obtained using the off-line structure selection so that the off-line approximation error is within the required accuracy. As observations are received, the network grows by adding new units. This is called the addition operation. The decision to add a new unit depends on two conditions. The first is that the following must be satisfied:

$$|y_t - \hat{f}(\mathbf{x}_t)| > \delta_{max} \tag{21}$$

where δ_{max} is chosen to represent the desired maximum tolerable accuracy. The above condition implies that the approximation error between the real output y_t and the output $\hat{f}(\mathbf{x}_t)$ of the VPBF network must be significant. The second is that the time period between two addition operations must not be less than the response time of the network to the addition operation. This is to limit the growing speed of the number of VPBFs in the network.

4 Recursive Weight Learning of Neural Networks

In the previous section, the structure selection of the VPBF network model was considered to reach a good approximation accuracy. This section takes into account the parameter adaptation laws which ensure that the estimation error converges to the desired range when the plant characteristics and the system operating point change. Here, it is assumed that the basis functions $\phi_1^o(\mathbf{x}_t), \phi_2^o(\mathbf{x}_t), ..., \phi_L^o(\mathbf{x}_t)$ are given. The estimated function $\hat{f}_t(\mathbf{x}_t)$ in the NARMA model can also be expressed by

$$\hat{f}_t(\mathbf{x}_t) = W_{t-1}^T \Phi_{t-1} \tag{22}$$

where the weight vector W_{t-1} and the basis function vector Φ_{t-1} are

$$W_{t-1} = [\, w_1(t-1) \quad w_2(t-1) \quad ... \quad w_L(t-1)\,]^T \tag{23}$$

$$\Phi_{t-1} = [\, \phi_1^o(\mathbf{x}_t) \quad \phi_2^o(\mathbf{x}_t) \quad ... \quad \phi_L^o(\mathbf{x}_t)\,]^T \tag{24}$$

and the inital weight vector is $W_0 = [\, w_1^o \quad w_2^o \quad ... \quad w_L^o\,]^T$. The output y_t of the system modelled using the VPBF network can be written in the form

$$y_t = \Phi_{t-1}^T W^* + \varepsilon_t \tag{25}$$

where W^* is the optimal estimate of the weight vector W_t in the network with L independent VPBF units and ε_t is the modelling error. In terms of the approximation ability of neural networks, the modelling error can be reduced arbitrarily by increasing the number L. Thus, it is reasonable to assume that the minimal upper bound of the modelling error $\varepsilon(t)$ is given by a constant δ_L, which represents the accuracy of the model and this is defined as

$$\delta_L = \sup_{t \in \Re^+} |\varepsilon_t| \tag{26}$$

The estimation problem is then to find a vector W belonging to the set defined by

$$\Xi(W) = \left\{ W : \left| y_t - W^T \Phi_{t-1} \right| \le \delta_L, \forall t \in \mathbb{N}^+ \right\}. \tag{27}$$

Although many algorithms have been proposed as a solution to the above estimation problem (see, for example, [7] [10] [30] [37] [38]), these are based on the assumption that the minimal upper bound δ_L of the estimation error is known. However, this assumption is not realistic in most practical applications. Following the recursive least squares algorithm, a recursive weight learning algorithm for the VPBF network is developed to remove this assumption. This algorithm is as follows.

$$W_t = W_t' - \alpha_t \beta_t \eta_t P_t \Phi_{t-1} e_t \tag{28}$$

$$W_t' = W_{t-1} + \alpha_t \beta_t P_t \Phi_{t-1} e_t \tag{29}$$

$$P_t = P_{t-1} - \beta_t \gamma_t P_{t-1} \Phi_{t-1} \Phi_{t-1}^T P_{t-1} \tag{30}$$

$$e_t = y_t - W_{t-1}^T \Phi_{t-1} \tag{31}$$

$$\alpha_t = (1 - \delta |e_t|^{-1})(1 + \Phi_{t-1}^T P_{t-1} \Phi_{t-1})^{-1} \tag{32}$$

$$\beta_t = \begin{cases} 1, & |e_t| > \delta \\ 0, & |e_t| \le \delta \end{cases} \tag{33}$$

$$\gamma_t = (|e_t| - \delta) \left(|e_t| + (2\,|e_t| - \delta)\, \Phi_{t-1}^T P_{t-1} \Phi_{t-1} \right)^{-1} \tag{34}$$

$$\eta_t \in \begin{cases} 0, & ||W_t'||_2 \le M \\ [s^-, \quad s^+], & ||W_t'||_2 > M \end{cases} \tag{35}$$

where the lower and upper bounds of η_t are given by

$$s^- = 1 + \frac{\alpha_t e_t \Phi_{t-1}^T P_t W_{t-1} - c_t}{\| \alpha_t e_t P_t \Phi_{t-1} \|_2^2} \tag{36}$$

$$s^+ = 1 + \frac{\alpha_t e_t \Phi_{t-1}^T P_t W_{t-1} + c_t}{\| \alpha_t e_t P_t \Phi_{t-1} \|_2^2} \tag{37}$$

with $c_t = \sqrt{(\alpha_t e_t \Phi_{t-1}^T P_t W_{t-1})^2 + \| \alpha_t e_t P_t \Phi_{t-1} \|_2^2 (M^2 - \| W_{t-1} \|_2^2)}$, M is the upper bound of the 2-norm of the weight vector W_t, and δ is the desired approximation error. The optimal design of η_t will be given later.

Next, the properties of the above learning algorithm are analysed using Lyapunov techniques. To ensure the convergence of the algorithm, consider the Lyapunov function:

$$V_t = \tilde{W}_t^T P_t^{-1} \tilde{W}_t \tag{38}$$

where $\tilde{W}_t = W^* - W_t$. Let

$$a_t = \alpha_t \beta_t P_t \Phi_{t-1} e_t \tag{39}$$

$$b_t = -\alpha_t \beta_t \eta_t P_t \Phi_{t-1} e_t \tag{40}$$

Then, the weight vector W_t in Eq.(28) can be written as

$$\begin{aligned} W_t &= W_{t-1} + \alpha_t \beta_t P_t \Phi_{t-1} e_t - \alpha_t \beta_t \eta_t P_t \Phi_{t-1} e_t \\ &= W_{t-1} + a_t + b_t \end{aligned} \tag{41}$$

The above implies that b_t is used to reduce the effect of a_t in the weight vector W_t if $||W_t'||_2 > M$. The Lyapunov function V_t in Eq.(38) is now extended to

$$\begin{aligned} V_t &= (\tilde{W}_{t-1} - a_t - b_t)^T P_t^{-1} (\tilde{W}_{t-1} - a_t - b_t) \\ &= (\tilde{W}_{t-1} - a_t)^T P_t^{-1} (\tilde{W}_{t-1} - a_t) + d_t \end{aligned} \tag{42}$$

where

$$d_t = -2 b_t^T P_t^{-1} (\tilde{W}_{t-1} - a_t) + b_t^T P_t^{-1} b_t \tag{43}$$

To make the algorithm converge fast, d_t should be as negative as possible. Two cases will be considered below, $\delta \geq \delta_L$ and $\delta < \delta_L$.

Case 1: $\delta \geq \delta_L$ and $\eta_t = 0$

In this case, it is clear that $W_t = W_t'$ since $\eta_t = 0$. This also means $||W_t'||_2 \leq M$ for all time. It will be shown later that this is true if M satisfies a certain condition. Since $\eta_t = 0$ which results in $d_t = 0$, Eq.(42) becomes

$$\begin{aligned} V_t &= (\tilde{W}_{t-1} - a_t)^T P_t^{-1} (\tilde{W}_{t-1} - a_t) \\ &= (\tilde{W}_{t-1} - \alpha_t \beta_t P_t \Phi_{t-1} e_t)^T P_t^{-1} (\tilde{W}_{t-1} - \alpha_t \beta_t P_t \Phi_{t-1} e_t) \\ &= \tilde{W}_{t-1}^T P_t^{-1} \tilde{W}_{t-1} - 2 \alpha_t \beta_t e_t \Phi_{t-1}^T \tilde{W}_{t-1} + \alpha_t^2 \beta_t e_t^2 \Phi_{t-1}^T P_t \Phi_{t-1} \end{aligned} \tag{44}$$

which uses $\beta_t = \beta_t^2$. Using the matrix inverse theorem [12], the inverse of the matrix P_t is obtained by

$$P_t^{-1} = P_{t-1}^{-1} + \alpha_t\beta_t\Phi_{t-1}\Phi_{t-1}^T \tag{45}$$

Since $y_t = \Phi_{t-1}^T W^* + \varepsilon_t$ and $e_t = y_t - W_{t-1}^T\Phi_{t-1}$,

$$\tilde{W}_{t-1}^T\Phi_{t-1} = e_t - \varepsilon_t \tag{46}$$

From Eqs. (45) and (46), the first term on the right side of Eq.(44) is expressed by

$$\begin{aligned} \tilde{W}_{t-1}^T P_t^{-1}\tilde{W}_{t-1} &= \tilde{W}_{t-1}^T P_{t-1}^{-1}\tilde{W}_{t-1} + \alpha_t\beta_t\left(\tilde{W}_{t-1}^T\Phi_{t-1}\right)^2 \\ &= V_{t-1} + \alpha_t\beta_t(e_t^2 + \varepsilon_t^2 - 2e_t\varepsilon_t) \end{aligned} \tag{47}$$

By Eq.(46), the second term on the right side of Eq.(44) is given by

$$-2\alpha_t\beta_t e_t\Phi_{t-1}^T\tilde{W}_{t-1} = 2\alpha_t\beta_t e_t(\varepsilon_t - e_t) \tag{48}$$

Substituting Eqs.(47) and (48) into Eq.(44) yields

$$\begin{aligned} V_t &= V_{t-1} + \alpha_t\beta_t(\varepsilon_t^2 - (1 - \alpha_t\Phi_{t-1}^T P_t\Phi_{t-1})e_t^2) \\ &= V_{t-1} + \alpha_t\beta_t\left(\varepsilon_t^2 - \alpha_t^{-1}\gamma_t e_t^2\right) \end{aligned} \tag{49}$$

using the result

$$\begin{aligned} 1 - \alpha_t\Phi_{t-1}^T P_t\Phi_{t-1} &= 1 - \alpha_t\Phi_{t-1}^T(P_{t-1} - \gamma_t P_{t-1}\Phi_{t-1}\Phi_{t-1}^T P_{t-1})\Phi_{t-1} \\ &= \alpha_t^{-1}\gamma_t \end{aligned} \tag{50}$$

Since it is assumed that the approximation error ε_t satisfies $|\varepsilon_t| \leq \delta_L \leq \delta$, then from the above

$$\begin{aligned} V_t &\leq V_{t-1} + \alpha_t\beta_t\left(\delta_L^2 - \alpha_t^{-1}\gamma_t e_t^2\right) \\ &\leq V_{t-1} + \alpha_t\beta_t\left(\delta^2 - \alpha_t^{-1}\gamma_t e_t^2\right) \\ &\leq V_{t-1} - \beta_t\gamma_t\left(|e_t|^3 - 2|e_t|\delta^2 + \delta^3\right)|e_t|^{-1} \end{aligned} \tag{51}$$

For $|e_t| \geq \delta$, it is not difficult to show that $|e_t|^3 - 2\delta^2|e_t| + \delta^3 \geq |e_t|^2(|e_t| - \delta)$. Hence

$$\begin{aligned} V_t &\leq V_{t-1} - \beta_t\gamma_t|e_t|(|e_t| - \delta) \\ &\leq V_{t-1} - \frac{\beta_t(|e_t| - \delta)^2}{2(1 + \Phi_{t-1}^T P_{t-1}\Phi_{t-1})}. \end{aligned} \tag{52}$$

which leads to

$$\lim_{t\to\infty} \frac{\beta_t(|e_t| - \delta)^2}{1 + \Phi_{t-1}^T P_{t-1}\Phi_{t-1}} = 0 \tag{53}$$

It is known from (30) that $\lambda_{max}(P_t) \leq \lambda_{max}(P_{t-1}) \leq \ldots \leq \lambda_{max}(P_0)$. As a result,

$$\begin{aligned} \| W_t - W_{t-1} \|_2^2 &= \alpha_t^2 \beta_t e_t^2 \Phi_{t-1}^T P_t^2 \Phi_{t-1} \\ &\leq \frac{\beta_t \lambda_{max}(P_0)\,(|e_t| - \delta)^2}{1 + \Phi_{t-1}^T P_{t-1} \Phi_{t-1}} \end{aligned} \tag{54}$$

It is clear from Eq. (52) that $V_t \leq V_0$. Thus,

$$\lambda_{min}(P_0^{-1}) \left\| \tilde{W}_t \right\|_2^2 \leq \lambda_{max}(P_0^{-1}) \left\| \tilde{W}_0 \right\|_2^2 \tag{55}$$

since $\lambda_{min}(P_t^{-1}) \geq \lambda_{min}(P_{t-1}^{-1}) \geq \ldots \geq \lambda_{min}(P_0^{-1})$.

Clearly, Eq.(53) shows that if $1 + \Phi_{t-1}^T P_{t-1} \Phi_{t-1}$ is finite for all time, which is true if the closed-loop system is stable, the estimation error e_t converges to δ, that is,

$$\lim_{t \to \infty} e_t = \delta \tag{56}$$

Also, it can be seen from Eq.(54) that the weights converge as time t approaches infinity. In addition, Eq. (55) implies that the weights will never drift to infinity over time. Thus, if M is chosen to satisfy

$$M \geq \frac{\lambda_{max}(P_0^{-1})}{\lambda_{min}(P_0^{-1})} \| W^* - W_0 \|_2^2 + \| W^* \|_2^2 \tag{57}$$

then $\| W_t \|_2^2$ is not greater than M for all time. In practice, it is very difficult to determine the upper bound δ_L of the modelling error ε_t and the optimal weight vector W^*. Eq.(52) shows that if $\delta < \delta_L$, W_t will drift away because $\Delta V_t = V_t - V_{t-1}$ may be positive. Moreover, if (57) is not satisfied, $||W_t^{'}||_2 \leq M, \forall t$ may not hold. These problems will be considered next.

Case 2: $\delta < \delta_L$ or $\eta_t \neq 0$

The analysis of the algorithm for Case 1 shows that if $\delta < \delta_L$, $W_t^{'}$ may be greater than the bound M. In addition, in the case where (57) is not satisfied, it can not be simply assumed that $\eta_t = 0$ since $W_t^{'}$ may also be greater than the bound M. So, $W_t = W_t^{'} - \eta_t \beta_t \alpha_t P_t \Phi_{t-1} e_t$ will be used for weight adjustment. This leads to

$$\begin{aligned} \| W_t \|_2^2 &= \| W_{t-1} + a_t + b_t \|_2^2 \\ &= \| W_{t-1} \|_2^2 + (a_t + b_t)^T (2W_{t-1} + a_t + b_t) \\ &= \| W_{t-1} \|_2^2 + (1 - \eta_t) a_t^T (2W_{t-1} + (1 - \eta_t) a_t) \end{aligned} \tag{58}$$

To ensure that the weight vector W_t does not drift, we require

$$(1 - \eta_t) a_t^T (2W_{t-1} + (1 - \eta_t) a_t) \leq M^2 - \| W_{t-1} \|_2^2 \tag{59}$$

where the solution to the inequality is given by

$$\eta_t \in [s^-, \quad s^+] \tag{60}$$

and s^- and s^+ are given by Eqs.(36) and (37). There are still an infinite number of possibilities for η_t. Hence, the question of what is the optimal solution of η_t arises. Consider Eq.(43). The first term on the right side of Eq.(43) can be calculated as

$$\begin{aligned}
-2b_t^T P_t^{-1}(\tilde{W}_{t-1} - a_t) &= 2\eta_t a_t^T P_t^{-1}(\tilde{W}_{t-1} - a_t) \\
&= 2\eta_t \left(\alpha_t\beta_t e_t \Phi_{t-1}^T \tilde{W}_{t-1} - \alpha_t^2\beta_t e_t^2 \Phi_{t-1}^T P_t \Phi_{t-1}\right) \\
&= 2\eta_t\alpha_t\beta_t e_t^2 - 2\eta_t\alpha_t\beta_t e_t\varepsilon_t - 2\eta_t\alpha_t^2\beta_t e_t^2\Phi_{t-1}^T P_t\Phi_{t-1}
\end{aligned} \tag{61}$$

The second term on the right side of Eq.(43) can be computed by

$$b_t^T P_t^{-1} b_t = \eta_t^2 \alpha_t^2 \beta_t e_t^2 \Phi_{t-1}^T P_t \Phi_{t-1} \tag{62}$$

Substituting Eqs.(61) and (62) into Eq.(43) yields

$$d_t = g_t(\eta_t) - 2\eta_t\alpha_t\beta_t e_t\varepsilon_t \tag{63}$$

where

$$g_t(\eta_t) = 2\eta_t\alpha_t\beta_t e_t^2 - (2 - \eta_t)\eta_t\alpha_t^2\beta_t e_t^2\Phi_{t-1}^T P_t\Phi_{t-1} \tag{64}$$

Now, d_t consists of two parts. The first is $2\eta_t\alpha_t\beta_t e_t\varepsilon_t$. This is the uncertain part because the modelling error is unknown. The second is $g(\eta_t)$ which is computable. It is also known from the Lyapunov technique that the more negative d_t is the faster is the reduction of the function V_t. Thus, the function $g(\eta_t)$ is used as the performance index for choosing the optimal solution of η_t.

The function $g_t(\eta_t)$ is a concave parabola and has only one minimum. The optimal η_t^* which minimizes g_t and the minimum $g_t(\eta_t^*)$ are given by

$$\eta_t^* = 1 - \frac{1}{\alpha_t \Phi_{t-1}^T P_t \Phi_{t-1}} \tag{65}$$

$$g_t(\eta_t^*) = -\frac{\beta_t(\alpha_t e_t^2 - \alpha_t^2 e_t^2 \Phi_{t-1}^T P_t \Phi_{t-1})^2}{\alpha_t^2 e_t^2 \Phi_{t-1}^T P_t \Phi_{t-1}} \tag{66}$$

Since $|\varepsilon_t| \le \delta_L$, it is clear from Eq.(63) that

$$d_t \le g_t(\eta_t^*) + 2|\eta_t^*|\alpha_t\beta_t|e_t|\delta_L \tag{67}$$

If $\eta_t^* \in [s^-, \quad s^+]$, then from Eqs.(42), (49) and (67)

$$\begin{aligned}
V_t &\le V_{t-1} + \alpha_t\beta_t\left(\delta_L^2 - \alpha_t^{-1}\gamma_t e_t^2\right) + g_t(\eta_t^*) + 2\alpha_t\beta_t|\eta_t^* e_t|\delta_L \\
&= V_{t-1} + \alpha_t\beta_t\left(\delta_L^2 - \alpha_t^{-1}\gamma_t e_t^2\right) + \alpha_t\beta_t|\eta_t^* e_t|(2\delta_L - |1 - \alpha_t\Phi_{t-1}^T P_t\Phi_{t-1}||e_t|)
\end{aligned} \tag{68}$$

It is clear from Eq. (68) that if

$$|e_t| > \max\{\sqrt{\alpha_t\gamma_t^{-1}}\delta_L, \quad 2|1-\alpha_t\Phi_{t-1}^T P_t\Phi_{t-1}|^{-1}\delta_L\} \tag{69}$$

the second and third terms on the right hand side of Eq. (68) will be negative. Using Eqs.(32) and (34) gives

$$\alpha_t\gamma_t^{-1} = (1+(2-\delta|e_t|^{-1})\Phi_{t-1}^T P_{t-1}\Phi_{t-1})(1+\Phi_{t-1}^T P_{t-1}\Phi_{t-1})^{-1} \leq 2 \tag{70}$$

As a result, if the following condition

$$|e_t| > \max\{\sqrt{2}\delta_L, \quad 2|1-\alpha_t\Phi_{t-1}^T P_t\Phi_{t-1}|^{-1}\delta_L\} \tag{71}$$

is satisfied, then the weights converge to their optimal values since $\Delta V_t \leq 0$. On the other hand, if the above condition is not satisfied, it is possible that $\Delta V_t > 0$. This implies that the weight vector W_t may drift away over time. In this case, the weight learning algorithm given by Eq.(28) avoids divergence of the weight vector because $||W_t||_2$ will not be greater than M for $\eta_t \in [s^-, \quad s^+]$. Thus the error $|e_t|$ always converges.

If $\eta_t^* \notin [s^-, \quad s^+]$, let

$$\eta_t^+ = \begin{cases} s^+ & g(s^+) \leq g(s^-) \\ s^- & g(s^+) > g(s^-) \end{cases} \tag{72}$$

Then

$$\begin{aligned} V_t &\leq V_{t-1} + \alpha_t\beta_t\left(\delta_L^2 - \alpha_t^{-1}\gamma_t e_t^2\right) + g_t(\eta_t^+) + 2\alpha_t\beta_t|\eta_t^+ e_t|\delta_L \\ &= V_{t-1} + \alpha_t\beta_t\left(\delta_L^2 - \alpha_t^{-1}\gamma_t e_t^2\right) \\ &\quad +2\beta_t\alpha_t|\eta_t^+ e_t|(\delta_L - (1-(1-0.5\eta_t^+)\alpha_t\Phi_{t-1}^T P_t\Phi_{t-1}) \text{ sgn } (\eta_t^+)|e_t|) \end{aligned} \tag{73}$$

Similarly, if the following condition is satisfied

$$|e_t| > \max\{\sqrt{2}\delta_L, \quad (1-(1-0.5\eta_t^+)\alpha_t\Phi_{t-1}^T P_t\Phi_{t-1})^{-1} \text{ sgn } (\eta_t^+)\delta_L\} \tag{74}$$

then the weights converge to their optimal values since $\Delta V_t \leq 0$. However, it is possible that $\Delta V_t > 0$ if the above condition is not satisfied. This indicates that the weight vector W_t may drift away over time. But, the weigh learning algorithm given by Eq.(28) constrains $||W_t||_2$ to be not greater than M. Thus the error $|e_t|$ always converges.

In the light of the above analysis, the design of η_t may be given by

$$\eta_t = \max(\min(\eta_t^*, s^+), s^-) \tag{75}$$

The analysis of the algorithm for the weight adaptation laws clearly shows that if the minimal upper bound δ_L of the approximation error is not known both the weights and the estimation error are still bounded.

5 Simulation Study

Two simulated examples will be used to illustrated the on-line identification. The first is a system described by an input-output model and the second is a system described by a state-space model.

Example 1

Consider the nonlinear system described by the input-output model [29]

$$y_t = \frac{y_{t-1}y_{t-2}y_{t-3}u_{t-2}(y_{t-3}-1)}{1+y_{t-2}^2+y_{t-3}^2} \tag{76}$$

The input u was set to be a random sequence between -0.5 and 0.5. Based on the input-output data, the orthogonal least algorithm was used in off-line structure selection of the VPBF network. Their order of selection and the corresponding weights are given in Table 1.

On-line structure selection was then applied and the recursive weight learning algorithm was used. The input was defined as

$$u_t = \begin{cases} \sin(2\pi t/250) & t \leq 500 \\ 0.8\sin(2\pi t/250) + 0.2\sin(2\pi t/25) & t > 500 \end{cases} \tag{77}$$

The simulation parameters were $M = 1.4$, $\delta = 0.02$. The growing VPBF network began with the fist five best VPBFs given in Table 1. The network grew until the number of VPBFs was 20. The simulation results are shown in Figs. 1-4.

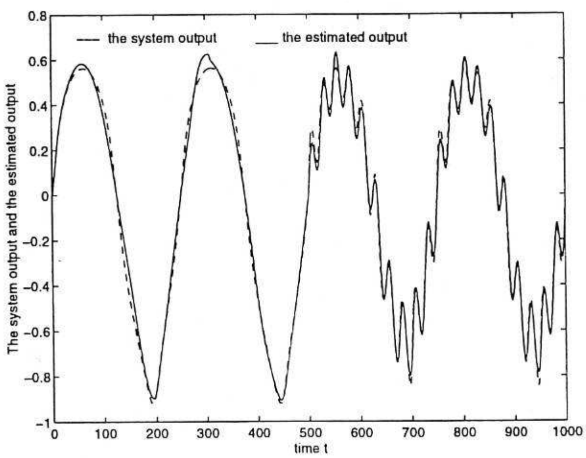

Figure 1: The system output y_t and estimated output $\hat{y}_t$ using on-line identification (Example 1).

Table 1

Priority Order i	VPBF ϕ_i^o	Weight w_i^o
1	u_{t-1}	0.9907
2	$y_{t-2}^2 u_{t-1}$	-0.6967
3	$y_{t-3}^2 u_{t-1}$	-0.7903
4	$y_{t-2} y_{t-3}$	-0.1437
5	$y_{t-2} u_{t-2}^2$	0.4680
6	$y_{t-1}^2 u_{t-3}$	-0.4705
7	$y_{t-2} u_{t-2} u_{t-3}$	0.3687
8	$y_{t-3} u_{t-3}$	0.0819
9	$y_{t-1} y_{t-2} u_{t-3}$	-0.3732
10	$y_{t-2} y_{t-3} u_{t-1}$	0.0244
11	$y_{t-2}^2 u_{t-3}$	-0.0304
12	$y_{t-1} y_{t-2}$	0.0052
13	$u_{t-1} u_{t-2} u_{t-3}$	0.2794
14	$y_{t-3} u_{t-2}^2$	-2.9104
15	$y_{t-3} u_{t-1}^2$	-0.0191
16	y_{t-2}	0.0052
17	$y_{t-3} u_{t-3}^2$	0.0177
18	$y_{t-3} u_{t-1}$	-0.0046
19	$y_{t-1}^2 y_{t-3}$	-4.2739
20	$y_{t-1} y_{t-3} u_{t-2}$	7.0854

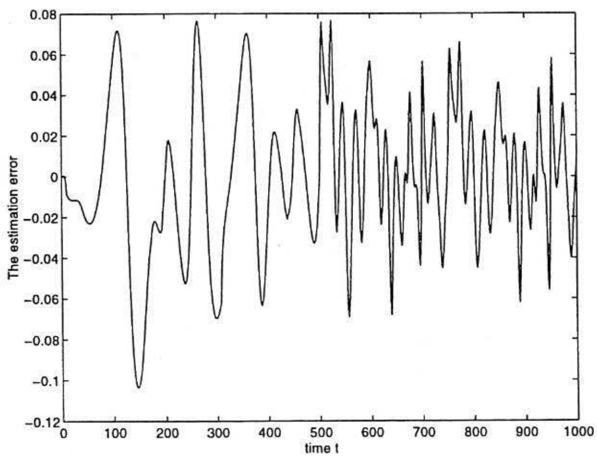

Figure 2: The on-line estimation error e_t (Example 1).

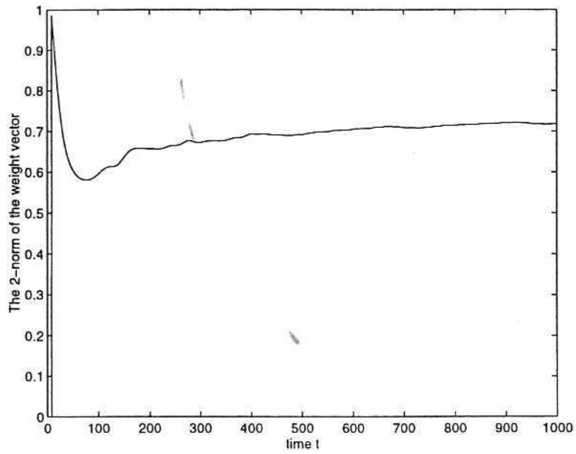

Figure 3: The 2-norm of the weigh vector W_t using on-line identification (Example 1).

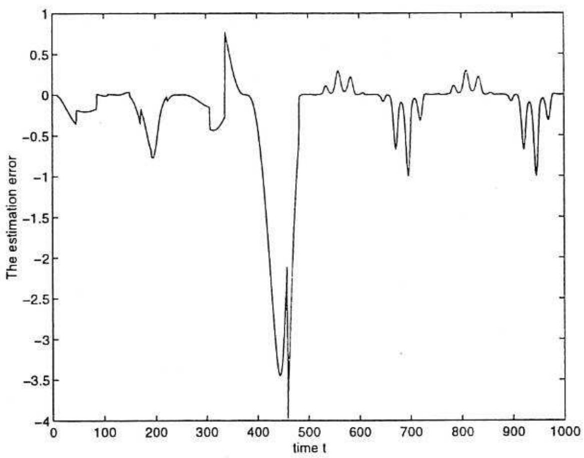

Figure 4: The estimation error $y_t - \hat{y}_t$ using off-line identification with 20 VPBFs (Example 1).

Example 2

Consider the nonlinear system described by the state-space model

$$x_1(t+1) = \frac{8.6x_1(t)}{1+u^2(t)} - \frac{1.5x_2^3(t)}{1+x_1^2(t)+x_2^2(t)} \tag{78}$$

$$x_2(t+1) = 1.4u^3(t) - \frac{1.8x_1(t)x_2(t)}{1+x_1^2(t)} \tag{79}$$

$$y(t) = 5x_1(t)u(t) - x_2(t)\sin(x_1(t)) \tag{80}$$

The input u was set to be a random sequence between -0.5 and 0.5, as in Example 1. Using the input-output data, the priority of the VPBFs was obtained using the orthogonal least squares algorithm. The order of the VPBFs and the corresponding weights were given in Table 2.

The on-line structure selection technique and the recursive weight learning algorithm were applied with the input given by

$$u(t) = 0.19\sin(2\pi t/50) + 0.095\sin(2\pi t/20) \tag{81}$$

The parameters in the simulation were $M = 7$, $\delta = 0.01$. The growing VPBF network started with the first 15 best VPBFs, and the network stopped growing when the number of the VPBFs reached 30. The simulation results are depicted in Figs. 5-8.

Table 2

Priority Order i	VPBF ϕ_i^o	Weight w_i^o
1	u_{t-2}^2	-4.4882
2	y_{t-1}^3	-0.0710
3	$y_{t-1}y_{t-2}u_{t-2}$	0.2592
4	$u_{t-2}u_{t-3}$	-2.8018
5	$y_{t-2}u_{t-1}$	0.5594
6	$y_{t-2}u_{t-1}u_{t-2}$	-0.5614
7	$u_{t-1}u_{t-2}$	0.5816
8	$y_{t-1}^2y_{t-2}$	0.2685
9	$y_{t-2}u_{t-1}^2$	1.1335
10	$y_{t-3}^2u_{t-2}$	-1.1162
11	$y_{t-1}u_{t-1}u_{t-2}$	0.7620
12	$y_{t-1}y_{t-2}$	1.6789
13	$y_{t-1}y_{t-3}^2$	-0.3676
14	u_{t-2}^3	-0.8687
15	u_{t-2}	1.3205
16	y_{t-1}	-0.2783
17	$u_{t-2}u_{t-3}^2$	0.1529
18	$y_{t-2}u_{t-2}^2$	-1.0620
19	$y_{t-3}u_{t-2}u_{t-3}$	-0.6005
20	$y_{t-1}u_{t-2}^2$	-1.0152
21	$y_{t-1}y_{t-2}^2$	0.4841
22	$y_{t-1}^2u_{t-3}$	-0.7660
23	$u_{t-2}u_{t-3}$	0.8791
24	u_{t-3}	0.8206
25	$u_{t-1}u_{t-3}$	-2.1835
26	$y_{t-1}u_{t-1}^2$	0.2744
27	u_{t-3}^3	0.5547
28	u_{t-3}^2	0.5798
29	1	0.0642
30	$y_{t-2}u_{t-3}^2$	-0.9806

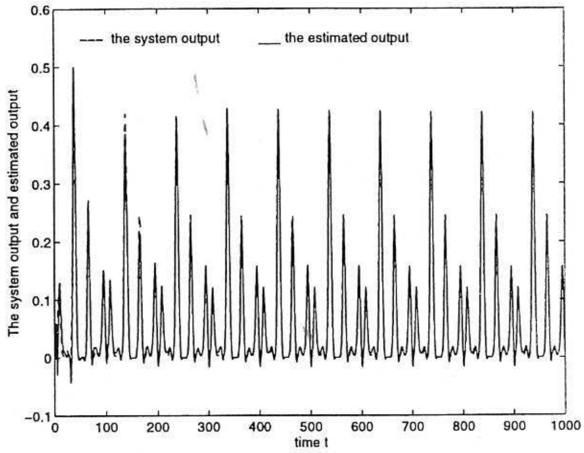

Figure 5: The system output y_t and estimated output $\hat{y}_t$ using on-line identification (Example 2).

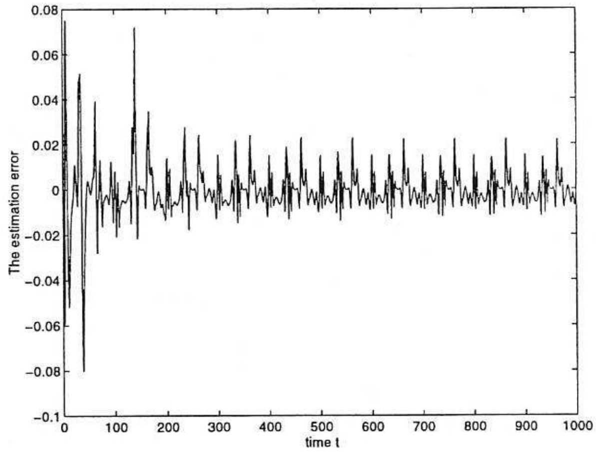

Figure 6: The on-line estimation error e_t (Example 2).

The results of the above two examples show that in terms of the estimation error the performance of the proposed on-line identification scheme is much better than an off-line approach. Although the minimal upper bound of the approximation error is unknown, the 2-norm of the weight vectors is bounded by M and the estimation errors converge to the required bounds.

6 Conclusions

An on-line nonlinear identification scheme based on VPBF networks together with an orthogonal least squares and a growing network algorithms has been presented. The structure

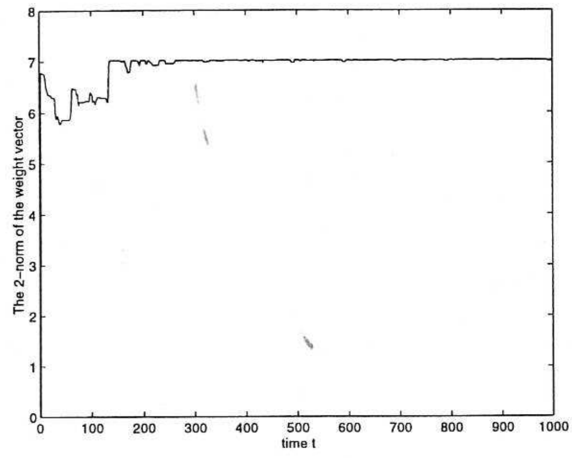

Figure 7: The 2-norm of the weigh vector W_t using on-line identification (Example 2).

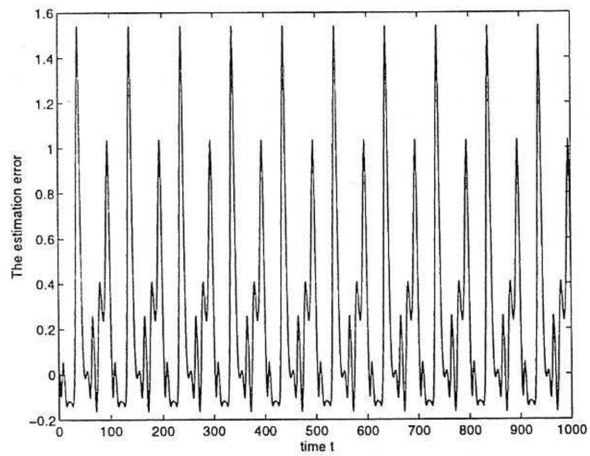

Figure 8: The estimation error $y_t - \hat{y}_t$ using off-line identification with 30 VPBFs (Example 2).

selection of nonlinear polynomials in the VPBF network and parameter estimation of the selected model were discussed. The orthogonal least squares algorithm was used for the off-line structure selection to find a initial set of VPBF candidate terms which were ranked according to the reduction in the approximation error. A growing network technique was then applied for on-line structure selection to obtain an appropriately sized network. An on-line recursive weight learning algorithm was developed for the parameter estimation and the properties of this were also analysized using Lyapunov methods. The learning algorithm ensures that the weights and approximation error converge to required bounds without assuming the approximation error is a white noise or that the upper bound of this is known.

Acknowledgements

The authors gratefully acknowledge the support of the Engineering and Physical Sciences Research Council (EPSRC) of UK under the contract GR/J46661.

References

[1] E. Baum and D. Haussler, "What size net gives valid generalization", *Neural Computation*, Vol.1, No.1, 1989.

[2] S.A. Billings, "Identification of nonlinear systems – a survey," *IEE Proceedings, Part D*, Vol. 127, pp.272-285, 1980.

[3] S. A. Billings and S. Chen, "Neural Networks and System Identification", In K. Warwick *et al.*, (eds.) *Neural networks for systems and control*, pp.181-205, 1992.

[4] S.A. Billings and W.S.F. Voon, "A prediction-error and stepwise-regression estimation algorithm for non-linear systems," *International Journal of Control*, Vol.44, No.1, pp.803-822, 1986.

[5] S.A. Billings, S. Chen and M.J. Korenberg, "Identification of MIMO nonlinear systems using a forward-regression orthogonal estimator," *International Journal of Control*, Vol.49, pp.2157-2189, 1988.

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

[7] C. Canudas de Wit and J. Carrillo, "A modified EW-RLS algorithm for systems with bounded disturbances," *Automatica*, vol. 26, pp. 599-606, 1990.

[8] S. Chen, S. A. Billings and P. M. Grant, "Nonlinear system identification using neural networks", *International Journal of Control*, Vol.51, No.6, pp.1191-1214, 1990.

[9] S. Chen and J. Wigger, " A fast orthogonal least squares algorithm for efficient subset model selection," *IEEE Trans. on Signal Processing*, Vol. 43, No. 7, pp. 1713-1715, 1995.

[10] E. Fogel E. and Y. F. Huang, "On the value of information in system identification – bounded noise case", *Automatica*, vol. 18, no. 2, pp. 229-238, 1982.

[11] C. Fung, S.A. Billings and W. Luo, "On line supervised adaptive tracking using radial basis function neural networks," Neural Networks, to appear.

[12] G. C. Goodwin and D. Q. Mayne, " A parameter estimation perspective of continuous time model reference adaptive control," *Automatica*, Vol. 23, No. 1, pp. 57-70, 1987.

[13] H. Haber and H. Unbehauen, "Structure identification of nonlinear dynamic systems – A survey on input/output approaches," *Automatica*, Vol.26, No.4, pp.651-677, 1990.

[14] S. Haykin, Neural Networks: A Comprehensive Foundation, Macmillan College Publishing Company, New York, 1994.

[15] V. Kadirkamanathan, *Sequential Learning in Artificial Neural Networks*, Ph.D Thesis, University of Cambridge, UK, 1991.

[16] V. Kadirkamanathan and G.P. Liu, "Robust identification with neural networks using multiobjective criteria," *Preprints of 5th IFAC Symposium on adaptive Systems in Control and Signal Processing*, Budapest, Hungary, pp. 237-242, 1995.

[17] V. Kadirkamanathan and M. Niranjan, "A function estimation approach to sequential learning with neural networks", *Neural Computation*, Vol.5, pp.954-957, 1993.

[18] M. Korenberg, S.A. Billings, Y.P. Liu and P.J. Mcllroy, "Orthogonal parameter estimation algorithm for non-linear stochastic systems," *International Journal of Control*, Vol.48, No.1, pp.193-210, 1988.

[19] J. G. Kuschewski, 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.

[20] I.J. Leontaritis and S.A. Billings, "Input-output parametric models for nonlinear systems, Parts I and II," *International Journal of Control*, Vol.41, No.1, pp.303-344, 1985.

[21] I.J. Leontaritis and S.A. Billings, "Model selection and validation methods for non-linear systems," *International Journal of Control*, Vol.45, No.3, pp.311-341, 1987.

[22] G. P. Liu, V. Kadirkamanathan and S.A. Billings, "Sequential identification of nonlinear systems by neural networks," Proceedings of the *3rd European Control Conference*, Rome, pp. 2408-2413, 1995.

[23] G. P. Liu, V. Kadirkamanathan and S.A. Billings, "Stable sequential identification of continuous nonlinear dynamical systems by growing RBF networks," to appear in the *International Journal of Control*, 1996.

[24] L. Ljung and T. Glad, Modelling of Dynamic Systems, Information and System Sciences Series, Prentice Hall, Englewood Cliffs, NJ., 1994.

[25] W. Luo and S.A. Billings, "Adaptive model selection and estimation for nonlinear systems using a sliding data window," Signal Processing, vol. 46, pp. 179-202, 1995.

[26] W. Luo and S.A. Billings, "Structure selective updating for nonlinear models and radial basis function neural networks," submitted for publication.

[27] W. Luo, S.A. Billings and K.M. Tsang, "On line structure detection and parameter estimation with exponential windowing for nonlinear systems," submitted for publication.

[28] D. J. C. MacKay, "Bayesian interpolation", *Neural Computation*, Vol.4, No.3, pp.415-447, 1992.

[29] K. S. Narendra 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.

[30] R. Ortega and R. Lozano-Leal, "Reformulation of the parameter identification problem for systems with bounded disturbances," Automatica, vol. 23, no. 2, pp. 247-251, 1987.

[31] M. M. Polycarpou and P. A. Ioannou, "Identification and control of nonlinear systems using neural network models: design and stability analysis", Technical Report 91-09-01, Department of Electrical Engineering-Systems, University of Southern California, USA, 1991.

[32] S. Z. Qin, 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.

[33] N. Sadegh, "A perceptron network for functional identification and control of nonlinear systems", *IEEE Transactions on Neural Networks*, Vol.4, No.6, pp.982-988, 1993.

[34] R. M. Sanner and J. J. E. Slotine, "Gaussian networks for direct adaptive control", *IEEE Trans. on Neural Networks*, Vol.3, No.6, pp.837-863, 1992.

[35] J. Sjoberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P. Glorennec, H. Hjalmarsson and A. Juditsky, "Nonlinear black-box modelling in system identification: a unified overview," Automatica, Vol.31, No.12, pp.1691-1724, 1995.

[36] P. Smyth, "On stochastic complexity and admissible models for neural network classifiers", In R. P. Lippmann, J. Moody and D. S. Touretzky (eds.), *Advances in Neural Information Processing Systems 3*, Morgan Kaufmann, CA, San Mateo, 1991.

[37] H. Wang, G.P. Liu, C.J. Harris and M. Brown, Advanced Adaptive Control, *Pergamon Press Ltd*, Oxford, 1995.

[38] J.F. Whidborne and G.P. Liu, Critical Control Systems: Theory, Design and Applications, *Research Studies Press*, UK., 1993.

[39] M. J. Willis, 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.

[40] Q. Zhu and S.A. Billings, " Fast orthogonal identification of nonlinear structure models and radial basis function neural networks," submitted for publication.