NEURAL NETWORK BASED PREDICTIVE CONTROL FOR NONLINEAR SYSTEMS

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. **619**

Department of Automatic Control & Systems Engineering
The University of Sheffield

March 1996

Neural Network Based Predictive Control for Nonlinear Systems

G. P. Liu, V. Kadirkamanathan and **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

A neural network based predictive controller design algorithm is introduced fo nonlinear control systems. It is shown that the use of nonlinear programming techniques can be avoided by using a set of affine nonlinear predictors to predict the output of the nonlinear process. The new predictive controller based on this design is both simple and easy to implement in practice. An on-line weight learning algorithm based on neural networks is introduced and convergence of both the weights and estimation errors is established. Predictive controller design based on the new procedure is illustrated using a growing network example.

Keywords: Neural networks, nonlinear systems, predictive control, on-line learning.

1 Introduction

Predictive control is now widely used by industry and a large number of implementation algorithms, including generalized predictive control [4], dynamic matrix control [5], extended prediction self-adaptive control [9], predictive function control [19], extended horizon adaptive control [24] and unified predictive control [21], have appeared in the literature. Most predictive control algorithms are based on a linear model of the process. However, industrial processes usually contain complex nonlinearities and a linear model may be acceptable only when the process is operating around an equilibrium point. If the process is highly nonlinear, a nonlinear model will be necessary to describe the behaviour of the process.

Recently, neural networks have become an attractive tool in the construction of models of complex nonlinear processes. This is because neural networks have an inherent ability of learning and approximating nonlinear functions arbitrarily well and a large number of identification and control structures based on neural networks have been proposed (see, for example, [1] [3] [6] [12] [16] [20]). Neural networks have also been used in some predictive control algorithms that utilize nonlinear process models [8] [25]. Alternative design of nonlinear predictive control algorithms have also been studied ([14] [15] [18]). However, in most algorithms for nonlinear predictive control performance functions are minimized using nonlinear programming techniques to compute the future manipulated variables in the on-line optimization. This can make the realization of the algorithms very difficult for real time control.

This paper introduces neural network based affine nonlinear predictors so that the predictive control algorithm is simple and easy to implement. The use of nonlinear programming techniques to solve the on-line optimization problem is avoided and a neural network based on-line weight learning algorithm is developed for the affine nonlinear predictors. It is shown that using this algorithm, both the weights in the neural networks and the estimation error converge and never drift to infinity over time.

The paper is organized as follows. Section 2 presents the structure of the affine nonlinear predictors using neural networks. The predictive neural controller is given in Section 3. Section 4 develops the on-line weight learning algorithm for the neural networks used for the predictors and includes analysis of the properties of the algorithm. The design of nonlinear predictive control by growing neural networks is illustrated in Section 5. Finally, Section 6 gives a simulated example to show the operation of the neural network based predictive control.

2 Neural Network Based Predictors

The fundamental idea in predictive control is to predict the vector of future tracking errors and minimize its norm over a given number of future control moves. It is therefore clear that predictive controller design mainly consists of two parts: prediction and minimization. This section discusses the prediction part. The minimization part will be considered in the next section.

Only discrete-time affine nonlinear control systems will be considered with an input-output relation described by

$$y_t = F(\mathbf{x}_t) + G(\mathbf{x}_t)u_{t-d} \tag{1}$$

where $F(.)$ and $G(.)$ are nonlinear functions, y is the output and u the control input, respectively, the vector $\mathbf{x}_t = [\, y_{t-1} \quad y_{t-2} \quad \ldots \quad y_{t-n}\,]$, n is the order of $y(t)$ and d is the time-delay of the system. It is assumed that the order n and the time delay d are known but the nonlinear functions $F(.)$ and $G(.)$ are unknown.

Basically, there are two kinds of predictors which can be used to predict the future tracking errors of nonlinear systems. One is the recursive predictor and the other is the nonrecursive predictor. Here, the latter is used. According to the affine nonlinear system described by Eq.(1), we present some $(i+d)$-step ahead nonrecursive affine nonlinear predictors to compensate for the influence of the time-delay, for $i = 0, 1, ..., L$. These predictors use sequences of both past inputs and outputs of the process upto the sampling time t to construct the predictive models, which are of the following form:

$$\hat{y}_{t+d+i} = \hat{F}_i(\mathbf{x}_t) + \sum_{j=0}^{i} \hat{G}_{ij}(\mathbf{x}_t)u_{t+j} \tag{2}$$

for $i = 0, 1, ..., L$, where $\hat{F}_i(\mathbf{x}_t)$ and $\hat{G}_{ij}(\mathbf{x}_t)$ are nonlinear functions of the vector $\mathbf{x}_t$ to be estimated. It can be seen from Eq. (2) that linearized predictors for nonlinear system which are widely used in the literature (see, e.g., [22] [23]) are a special case of the above.

Due to the arbitrary approximation feature of neural networks, the nonlinear functions $\hat{F}_i(\mathbf{x}_t)$ and $\hat{G}_{ij}(\mathbf{x}_t)$ can both be approximated by single hidden layer networks. This is expressed by,

$$\hat{F}_i(\mathbf{x}_t) = \sum_{k=1}^{N_i} f_{i,k}\phi_{i,k}(\mathbf{x}_t) \tag{3}$$

$$\hat{G}_{ij}(\mathbf{x}_t) = \sum_{k=1}^{N_{ij}} g_{ij,k}\phi_{ij,k}(\mathbf{x}_t) \tag{4}$$

for $j \le i$ and $i, j = 0, 1, ..., L$, where $\phi_{i,k}(\mathbf{x}_t)$ and $\phi_{ij,k}(\mathbf{x}_t)$ are basis functions of the networks, N_i and N_{ij} denote the size of the networks. Define the weight and basis function vectors of the neural networks as

$$\bar{F}_i = [\, f_{i,1} \quad f_{i,2} \quad \cdots \quad f_{i,N_i} \,]^T \tag{5}$$

$$\bar{G}_{ij} = [\, g_{ij,1} \quad g_{ij,2} \quad \cdots \quad g_{ij,N_{ij}} \,]^T \tag{6}$$

$$\bar{\Phi}_i = [\, \phi_{i,1}(\mathbf{x}_t) \quad \phi_{i,2}(\mathbf{x}_t) \quad \cdots \quad \phi_{i,N_i}(\mathbf{x}_t) \,]^T \tag{7}$$

$$\bar{\Phi}_{ij} = [\, \phi_{ij,1}(\mathbf{x}_t) \quad \phi_{ij,2}(\mathbf{x}_t) \quad \cdots \quad \phi_{ij,N_{ij}}(\mathbf{x}_t) \,]^T \tag{8}$$

The neural network based predictors can then be rewritten as

$$\hat{y}_{t+d+i} = \bar{F}_i^T \bar{\Phi}_i + \sum_{j=0}^{i} \bar{G}_{ij}^T \bar{\Phi}_{ij} u_{t+j} \tag{9}$$

for $i = 0, 1, ..., L$.

It is well known from the universal approximation theory for neural networks that the modelling error of the predictor can be reduced arbitrarily by properly selecting the basis functions and adjusting the weights. There are many types of basis functions which can be selected, including radial functions, sigmoid functions, polynomial functions and so on. Section 5 will discuss the selection of basis functions using a radial basis function network. An on-line learning algorithm for the weight adjustment of the networks used in the predictors will be given in Section 4.

3 Predictive Neural Control

This section presents a predictive control strategy based on the neural predictor described in the previous section. In order to define how well the predicted process output tracks the reference trajectory, there are many cost functions used in predictive control. This section uses a cost function which is of the following quadratic form.

$$J_p = \frac{1}{2} \| R_{t+d+L} - Y_{t+d+L} \|_2^2 + \frac{1}{2} \alpha \| \Delta U_{t+L} \|_2^2 \tag{10}$$

where

$$R_{t+d+L} = [\, r_{t+d} \quad r_{t+d+1} \quad \cdots \quad r_{t+d+L} \,]^T \tag{11}$$

$$Y_{t+d+L} = [\, \hat{y}_{t+d} \quad \hat{y}_{t+d+1} \quad \cdots \quad \hat{y}_{t+d+L} \,]^T \tag{12}$$

$$U_{t+L} = [\, u_t \quad u_{t+1} \quad \ldots \quad u_{t+L} \,]^T \tag{13}$$

R_{t+d+L}, Y_{t+d+L} and U_{t+d+L} are the future reference input, predicted output and control input vectors, respectively, L is the control horizon, $L+d$ is the prediction horizon, and $\alpha > 0$ is the weight.

The optimal controller output sequence over the prediction horizon is obtained by minimizing the performance index J_p with respect to U_{t+L}. This can be carried out by setting

$$\frac{\partial J_p}{\partial U_{t+L}} = 0 \tag{14}$$

Taking the derivative of the performance function J_p with respect to the control input vector U_{t+L} results in

$$\frac{\partial Y^T_{t+d+L}}{\partial U_{t+L}}(Y_{t+d+L} - R_{t+d+L}) + \alpha\frac{\partial \Delta U^T_{t+L}}{\partial U_{t+L}}\Delta U_{t+L} = 0 \tag{15}$$

Using the neural network based predictors (9), the derivatives of Y_{t+d+L} with respect to the control input vector U_{t+L} are given by

$$\frac{\partial Y^T_{t+d+L}}{\partial U_{t+L}} = Q^T_L = \begin{bmatrix} \bar{G}^T_{00}\bar{\Phi}_{00} & & & \mathbf{0} \\ \bar{G}^T_{10}\bar{\Phi}_{10} & \bar{G}^T_{11}\bar{\Phi}_{11} & & \\ \vdots & \vdots & \ddots & \\ \bar{G}^T_{L0}\bar{\Phi}_{L0} & \bar{G}^T_{L1}\bar{\Phi}_{L1} & \dots & \bar{G}^T_{LL}\bar{\Phi}_{LL} \end{bmatrix}^T \tag{16}$$

Let

$$H_L = [\,\bar{F}^T_0\bar{\Phi}_0 \quad \bar{F}^T_1\bar{\Phi}_1 \quad \dots \quad \bar{F}^T_L\bar{\Phi}_L\,]^T \tag{17}$$

Eq.(15) can be compactly expressed by the following matrix equation:

$$Q^T_L(H_L + Q_L U_{t+L} - R_{t+d+L}) - \alpha I^T_1 u_{t-1} + \alpha D^T_L D_L U_{t+L} = 0 \tag{18}$$

where $I_1 = [1, 0, ..., 0]$ is an identity vector and the matrix D_L is of the form

$$D_L = \begin{bmatrix} 1 & & & \mathbf{0} \\ -1 & 1 & & \\ & \ddots & \ddots & \\ \mathbf{0} & & -1 & 1 \end{bmatrix} \tag{19}$$

It is clear from Eq.(18) that the controller input vector U_{t+L} can be calculated by

$$U_{t+L} = (Q^T_L Q_L + \alpha D^T_L D_L)^{-1}(Q^T_L R_{t+d+L} - Q^T_L H_L + \alpha I^T_1 u_{t-1}) \tag{20}$$

Thus, the control input u_t minimizing the performance function J_p is given by

$$u_t = I_1(Q^T_L Q_L + \alpha D^T_L D_L)^{-1}(Q^T_L R_{t+d+L} - Q^T_L H_L + \alpha I^T_1 u_{t-1}) \tag{21}$$

The predictive neural controller is therefore relatively simple and easy implement using the affine nonlinear predictors. There is no need to solve a nonlinear programming problem to obtain the optimal control input u_t unless additional constraints are imposed on the control signal and/or output of the system.

4 On-Line Weight Learning of Neural Networks

Here, we consider the on-line adjustment of the weights of the i-th predictor. The weight estimation of the other predictors are similar. It will be assumed that the basis functions of all the networks which are used in the predictors are given and the required prediction accuracy can be achieved by adjusting the corresponding weights to those functions.

Using the available output data $y_{t-d-i-1}, ..., y_{t-d-i-n}$ and the input data $u_{t-d}, ..., u_{t-d-i}$, the output of the i-th predictor at time t can be written as

$$y_t = (\bar{F}_i^*)^T \bar{\Phi}_i(\mathbf{x}_{t-d-i}) + \sum_{j=0}^{i} (\bar{G}_{ij}^*)^T \bar{\Phi}_{ij}(\mathbf{x}_{t-d-i}) u_{t-d-i+j} + \varepsilon_t \tag{22}$$

where $\bar{F}_i^*$ and $\bar{G}_{ij}^*$ are the optimal estimates of the weight vector $\bar{F}_i$ and $\bar{G}_{ij}$, for $j = 0, 1, ..., i$, respectively, ε_t is the approximation error of the predictor using the neural network and is assumed to be bounded by a positive number δ for all time, that is

$$\max_{t \in \mathbb{N}^+} |\varepsilon_t| \leq \delta. \tag{23}$$

The i-th estimated predictor can also be compactly expressed by

$$\hat{y}_t = W_t^T \Phi_{t-1} \tag{24}$$

where the weight vector W_t and the basis function vector Φ_t are

$$W_t = [\, \bar{F}_i^T \quad \bar{G}_{i0}^T \quad \bar{G}_{i1}^T \quad \ldots \quad \bar{G}_{ii}^T \,]^T \tag{25}$$

$$\Phi_{t-1} = \begin{bmatrix} \bar{\Phi}_i(\mathbf{x}_{t-d-i}) \\ \bar{\Phi}_{i0}(\mathbf{x}_{t-d-i}) u_{t-d-i} \\ \bar{\Phi}_{i1}(\mathbf{x}_{t-d-i}) u_{t-d-i+1} \\ \vdots \\ \bar{\Phi}_{ii}(\mathbf{x}_{t-d-i}) u_{t-d} \end{bmatrix} \tag{26}$$

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

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

In recent years, many estimation algorithms have been presented for fixed, growing and variable structure neural networks (see, for example, [10] [11] [13]). Based on the recursive least squares algorithm, an new on-line weight learning algorithm of neural networks is developed for the affine nonlinear predictors. The algorithm and its properties are given by the following theorem.

Theorem 1: Consider the i-th predictor and the learning algorithm:

$$W_t = W_{t-1} + \alpha_t \beta_t P_{t-1} \Phi_{t-1} e_t \tag{28}$$

$$P_t = P_{t-1} - \beta_t \gamma_t P_{t-1} \Phi_{t-1} \Phi_{t-1}^T P_{t-1} \tag{29}$$

$$\alpha_t = (1 - \delta |e_t|^{-1})(1 + \Phi_{t-1}^T P_{t-1} \Phi_{t-1})^{-1} \tag{30}$$

$$\gamma_t = (|e_t| - \delta)\left(|e_t| + (2|e_t| - \delta)\,\Phi_{t-1}^T P_{t-1}\Phi_{t-1}\right)^{-1} \tag{31}$$

$$e_t = y_t - W_{t-1}^T \Phi_{t-1} \tag{32}$$

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

Then

$$i) \qquad \lim_{t\to\infty} \frac{\beta_t (|e_t| - \delta)^2}{1 + \Phi_{t-1}^T P_{t-1}\Phi_{t-1}} = 0, \tag{34}$$

$$ii) \qquad \lim_{t\to\infty} |W_t - W_{t-1}| = 0, \tag{35}$$

$$iii) \qquad \left\| \tilde{W}_t \right\|_2 \le \sqrt{\frac{\lambda_{max}\left(P_0^{-1}\right)}{\lambda_{min}\left(P_0^{-1}\right)}} \left\| \tilde{W}_0 \right\|_2, \tag{36}$$

where

$$\tilde{W}_t = W^* - W_t \tag{37}$$

$\lambda_{max}(\cdot)$ and $\lambda_{min}(\cdot)$ denote the maximum and the minimum eigenvalues of the matrix $(\cdot)$, respectively, and W^* is the optimal estimate of the weight vector W_t.

Proof: (i) Consider the Lyapunov function:

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

Combining (22) and (28) - (33) yields

$$\begin{aligned} V_t &= (\tilde{W}_{t-1} - \alpha_t\beta_t P_{t-1}\Phi_{t-1}e_t)^T P_t^{-1}(\tilde{W}_{t-1} - \alpha_t\beta_t P_{t-1}\Phi_{t-1}e_t) \\ &= V_{t-1} + \alpha_t\beta_t\left[\varepsilon_t^2 - \alpha_t^{-1}\gamma_t e_t^2\right]. \end{aligned} \tag{39}$$

Since it is assumed that the approximation error ε_t of the predictor satisfies $|\varepsilon_t| \le \delta$, it is known from the above that

$$\begin{aligned} \Delta V_t &\le \alpha_t\beta_t\left(\delta^2 - \alpha_t^{-1}\gamma_t e_t^2\right) \\ &= -\alpha_t\beta_t\left(\frac{\left(|e_t|^3 - 2|e_t|\delta^2 + \delta^3\right)\left(1 + \Phi_{t-1}^T P_{t-1}\Phi_{t-1}\right)}{|e_t| + (2|e_t| - \delta)\,\Phi_{t-1}^T P_{t-1}\Phi_{t-1}} + \gamma_t\delta^2\right) \\ &\le -\beta_t\gamma_t\left(|e_t|^3 - 2|e_t|\delta^2 + \delta^3\right)|e_t|^{-1} \end{aligned} \tag{40}$$

In addition, for $|e_t| \ge \delta$,

$$e_t^2 - 2\delta^2 + \frac{\delta^3}{|e_t|} \ge |e_t|\,(|e_t| - \delta). \tag{41}$$

It is straightforward to show that

$$\begin{aligned} V_t &\le V_{t-1} - \beta_t\gamma_t|e_t|(|e_t| - \delta) \\ &\le V_{t-1} - \frac{\beta_t(|e_t| - \delta)^2}{2\left(1 + \Phi_{t-1}^T P_{t-1}\Phi_{t-1}\right)}. \end{aligned} \tag{42}$$

It can therefore follow that

$$\lim_{t\to\infty} \frac{\beta_t \left(|e_t| - \delta\right)^2}{1 + \Phi_{t-1}^T P_{t-1} \Phi_{t-1}} = 0 \tag{43}$$

(ii) From (28),

$$\begin{aligned} \| W_t - W_{t-1} \|_2^2 &= \alpha_t^2 \beta_t \Phi_{t-1}^T P_{t-1}^2 \Phi_{t-1} e_t^2 \\ &\leq \frac{\beta_t \lambda_{max}(P_{t-1})}{1 + \Phi_{t-1}^T P_{t-1} \Phi_{t-1}} \left(|e_t| - \delta\right)^2 \end{aligned} \tag{44}$$

It is clear from (29) that $\lambda_{max}(P_t) \leq \lambda_{max}(P_{t-1}) \leq \ldots \leq \lambda_{max}(P_0)$. Then (44) can be written as

$$\| W_t - W_{t-1} \|_2^2 \leq \frac{\beta_t \lambda_{max}(P_0) \left(|e_t| - \delta\right)^2}{1 + \Phi_{t-1}^T P_{t-1} \Phi_{t-1}} \tag{45}$$

which, together with (43), proves (ii).

(iii) From the matrix inversion theorem [7], it follows that

$$P_t^{-1} = P_{t-1}^{-1} + \frac{\beta_t \Phi_{t-1} \Phi_{t-1}^T}{1 + \Phi_{t-1}^T P_{t-1} \Phi_{t-1}} \left(1 - \frac{\delta}{|e_t|}\right) \tag{46}$$

Then

$$\lambda_{min}(P_t^{-1}) \geq \lambda_{min}(P_{t-1}^{-1}) \geq \ldots \geq \lambda_{min}(P_0^{-1}) \tag{47}$$

Equation (42), together with the above, gives

$$V_t \leq V_0 \tag{48}$$

which results in

$$\lambda_{min}(P_0^{-1}) \left\| \tilde{W}_t \right\|_2^2 \leq \lambda_{max}(P_t^{-1}) \left\| \tilde{W}_0 \right\|_2^2 \tag{49}$$

Thus,

$$\left\| \tilde{W}_t \right\|_2^2 \leq \frac{\lambda_{max}(P_0^{-1})}{\lambda_{min}(P_0^{-1})} \left\| \tilde{W}_0 \right\|_2^2 \tag{50}$$

This establishes (iii).

Property (i) of the theorem above 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. Also, it can be seen from Property (ii) that the weights converge as time t approaches infinity. In addition, Property (iii) implies that the weights will never drift to infinity over time.

5 Predictive Controller Design Using Growing Networks

This section introduces growing neural networks. Consider the i-th predictor to show how to design the predictive control. For the sake of simplicity, the basis function vectors of the i-th predictor are assumed to be

$$\bar{\Phi}_i = \bar{\Phi}_{i0} = \bar{\Phi}_{i1} = \ldots = \bar{\Phi}_{ii} \tag{51}$$

This means all neural networks for the i-th predictor have the same basis functions. which are of the form

$$\bar{\Phi}_i = [\,\phi_{i1}(\mathbf{x}_t) \quad \phi_{i2}(\mathbf{x}_t) \quad \ldots \quad \phi_{iN_i}(\mathbf{x}_t)\,]^T \tag{52}$$

where N_i is the number of the basis functions and the basis functions $\phi_{ik}(\mathbf{x}_t)$ are the Gaussian radial basis functions (GRBF), i.e.

$$\phi_{ik}(\mathbf{x}_t) = \exp\{-r_{ik}^{-2} \| \mathbf{x}_t - \mathbf{c}_{ik} \|_2^2\} \tag{53}$$

r_{ik} is the width of the (ik)-th basis function and $\mathbf{c}_{ik}$ is its centre.

The i-th predictor is now given by

$$\hat{y}_{t+d+i} = \sum_{k=1}^{N_i} f_{i,k}\phi_{ik}(\mathbf{x}_t) + \sum_{j=0}^{i}\sum_{k=1}^{N_i} g_{ij,k}\phi_{ik}(\mathbf{x}_t)u_{t+j} \tag{54}$$

If the prediction error of the i-th predictor is greater than required, according to approximation theory more basis functions should be added to the networks to improve approximation. In this case, denote the structure of the i-th predictor at time $t-1$ as $\hat{y}^{(t-1)}_{t+d+i}$ and the structure immediately after the addition of a basis function at time t as $\hat{y}^{(t)}_{t+d+i}$. Based on the structure of the function $\hat{y}_{t+d+i}$ in equation (9), the structure of the i-th predictor now becomes,

$$\hat{y}^{(t)}_{t+d+i} = \hat{y}^{(t-1)}_{t+d+i} + \left(f_{i,N_i+1}\phi_{i(N_i+1)}(\mathbf{x}_t) + \sum_{j=1}^{i} g_{ij,N_i+1}\phi_{i(N_i+1)}(\mathbf{x}_t)u_{t+j} \right) \tag{55}$$

where f_{i,N_i+1} and g_{ij,N_i+1} are the weights corresponding to the new $(N_i+1)^{th}$ Gaussian radial basis function $\phi_{i(N_i+1)}(\mathbf{x}_t)$.

The growing network is initialised with no basis function units. As observations are received the network grows by adding new units. The decision to add a new unit depends on the observation novelty for which the following two conditions must be satisfied:

$$\begin{aligned} &\text{(i)} \qquad \min_{k=1,\ldots,N_i} \| \mathbf{x}_t - \mathbf{c}_{ik} \|_2 > \delta_c \\ &\text{(ii)} \qquad |e_i(t)| > \delta_{max} \end{aligned} \tag{56}$$

where $e_i(t)$ is the prediction error of the i-th predictor which may approximately be measured by e_t defined by (32), δ_c is the required distance between the basis functions and δ_{max} is chosen to represent the desired maximum tolerable accuracy of the predictor estimation. Criterion (i) says that the current observation must be far from existing centres. Criterion (ii) means that the approximation error in the network must be significant.

If the above conditions are satisfied, the new centre is set to be $\mathbf{c}_{i(N_i+1)} = \mathbf{x}_t$. In order to assign a new basis function $\phi_{i(N_i+1)}(\mathbf{x}_t)$ that is nearly orthogonal to all existing basis functions, the angle between the GRBFs should be as large as possible. The width $r_{i(N_i+1)}$ should therefore be reduced. However, reducing $r_{i(N_i+1)}$ increases the curvature of $\phi_{i(N_i+1)}(\mathbf{x}_t)$ which in turn gives a less smooth function and can lead to *overfitting* problems. Thus, to make a trade-off between orthogonality and smoothness, a good choice for the width $r_{i(N_i+1)}$, which ensures the angles between GRBF units are approximately equal to the required angle θ_{min}, is [11],

$$r_{i(N_i+1)} = \left(\frac{1}{2\log(1/\cos^2\theta_{min})}\right)^{\frac{1}{2}} \min_{k=1,\ldots,N_i}\{\|\mathbf{c}_{ik} - \mathbf{c}_{i(N_i+1)}\|_2\} \tag{57}$$

where θ_{min} being the required minimum angle between Gaussian radial basis functions.

The above assignments are the same as those for the resource allocating network (RAN) [17] which is defined based on observation novelty heuristics. When a new unit is added to the network at time t, the dimension of the vectors W_t and Φ_t and the matrix P_t should increase by 1. The on-line learning algorithm for the i-th predictor is still the same as that given in Section 4.

After the above consideration, the matrices Q_L and H_L are of the following form:

$$Q_L^T = \begin{bmatrix} \bar{G}_{00}^T\bar{\Phi}_0 & & \mathbf{0} & \\ \bar{G}_{10}^T\bar{\Phi}_1 & \bar{G}_{11}^T\bar{\Phi}_1 & & \\ \vdots & \vdots & \ddots & \\ \bar{G}_{L0}^T\bar{\Phi}_L & \bar{G}_{L1}^T\bar{\Phi}_L & \ldots & \bar{G}_{LL}^T\bar{\Phi}_L \end{bmatrix}^T \tag{58}$$

$$H_L = [\,\bar{F}_0^T\bar{\Phi}_0 \quad \bar{F}_1^T\bar{\Phi}_1 \quad \ldots \quad \bar{F}_L^T\bar{\Phi}_L\,]^T \tag{59}$$

The predictive controller are still given by the form (21). In this way, the design of the nonlinear predictive neural control is completed by growing networks.

6 Simulation

In this section, consider the following affine nonlinear system [2]:

$$y_t = \frac{2.5y_{t-1}y_{t-2}}{1 + y_{t-1}^2 + y_{t-2}^2} + 0.3cos(0.5(y_{t-1} + y_{t-2})) + 1.2u_{t-1} \tag{60}$$

The reference input $r(t) = sin(\pi t/500)$. The initial condition of the plant is $(y_{-1}, y_{-2}) = (0, 0)$.

The goal is to control the plant (60) to track the reference input $r(t)$ using a predictive control strategy so that the following quadratic cost function is minimized.

$$J_p = \frac{1}{2}\left(\left\|\begin{bmatrix} r_{t+1} \\ r_{t+2}\end{bmatrix} - \begin{bmatrix}\hat{y}_{t+1} \\ \hat{y}_{t+2}\end{bmatrix}\right\|_2^2 + \left\|\begin{bmatrix}\Delta u_t \\ \Delta u_{t+1}\end{bmatrix}\right\|_2^2\right) \tag{61}$$

It is well known that if the variables of a nonlinear function are in compact sets, the nonlinear function can be approximated arbitrarily well by neural networks. To ensure y_t is in a compact set, we used the following one-to-one (1-1) mapping [13]:

$$\bar{y}_t = \frac{y_t}{|y_t| + 1}, \tag{62}$$

It is clear from equation (62) that $\bar{y}_t \in [-1, 1]$ for $y_t \in (-\infty, +\infty)$. Thus $\mathbf{x}_t$ is replaced by the vector $\bar{\mathbf{x}}_t = [\bar{y}_{t-1}, \bar{y}_{t-2}]$.

We used growing networks to model the one-step and two-step ahead predictors. The growing networks were initialised with no basis function units. As observations were received the network grew by adding new units. The required distance and minimum angle between the basis functions were set to be $\delta_c = 0.011$, $\theta_{min} = 20°$, respectively. The required and maximum tolerable estimation accuracy of the predictor were $\delta = 0.008$ and $\delta_{max} = 0.012$, respectively.

In the simulation, the performance of the neural network based predictive control is shown in Figs.1-5. Fig.1 shows the output y_t and the reference input r_t of the system. The tracking error $r_t - y_t$ is shown in Fig.2. The estimation errors of the one-step and two-step ahead predictors using growing networks are illustrated in Figs. 3 and 4, respectively. The number of basis functions in the neural networks with respect to time t for the two-step ahead predictor is given in Fig. 5.

It can be seen from the simulation results that the tracking error, the one-step and two-step prediction errors converge with time t. It is also clear that the growing neural network based prediction models grow gradually to approach the appropriate complexity of the predictors that are sufficient to provide the required approximation accuracy. Moreover, the nonlinear system is successfully controlled using the predictive neural controller.

7 Conclusions

This paper has developed a neural network based predictive controller design procedure for nonlinear systems. A set of affine nonlinear neural predictors was used to predict the output of the nonlinear process so that the difficulty of applying nonlinear programming techniques to minimize the performance function for nonlinear predictive control is avoided. The resulting predictive neural control algorithm is relatively simple and easy to implement in practice. Based on least squares techniques, an on-line weight learning algorithm for the neural networks based affine nonlinear predictors has been given. The properties of the algorithm have been studied and it has been shown that both the weights and the estimation error converge as time approaches infinity. An analysis of the stability of the closed-loop nonlinear predictive neural control system remains an open question for future research.

8 Acknowledgement

The authors are grateful for the support of EPSRC of UK under the grant GR/J46661.

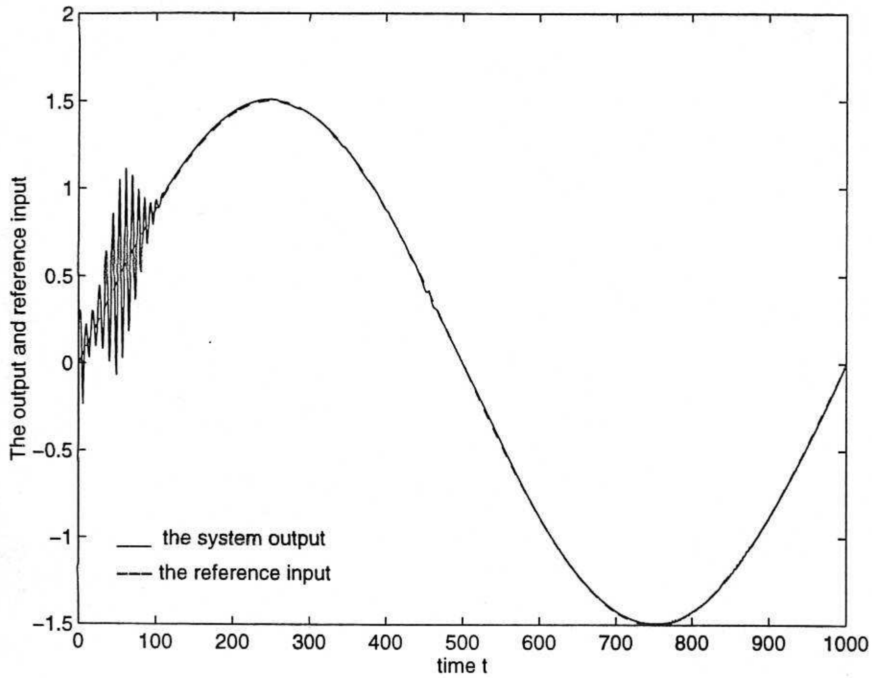

Figure 1: The output y_t and the reference input r_t of the system.

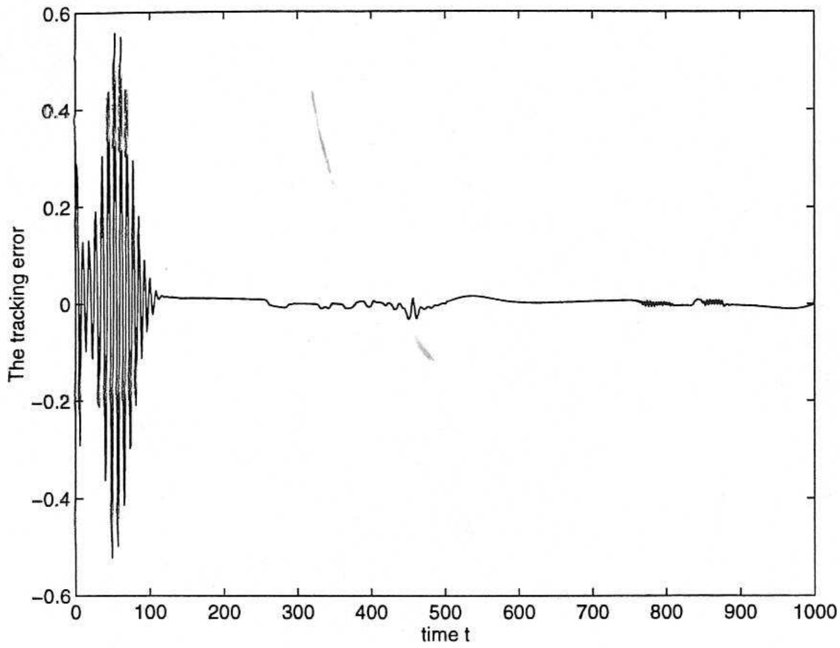

Figure 2: The tracking error $y_t - r_t$ of the system.

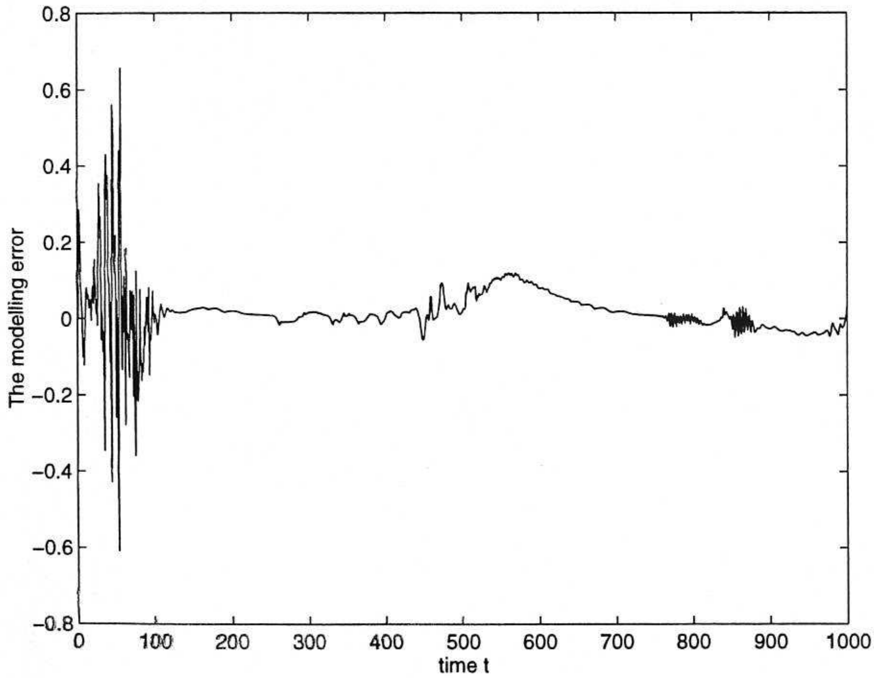

Figure 3: The modelling error of the one-step ahead predictor.

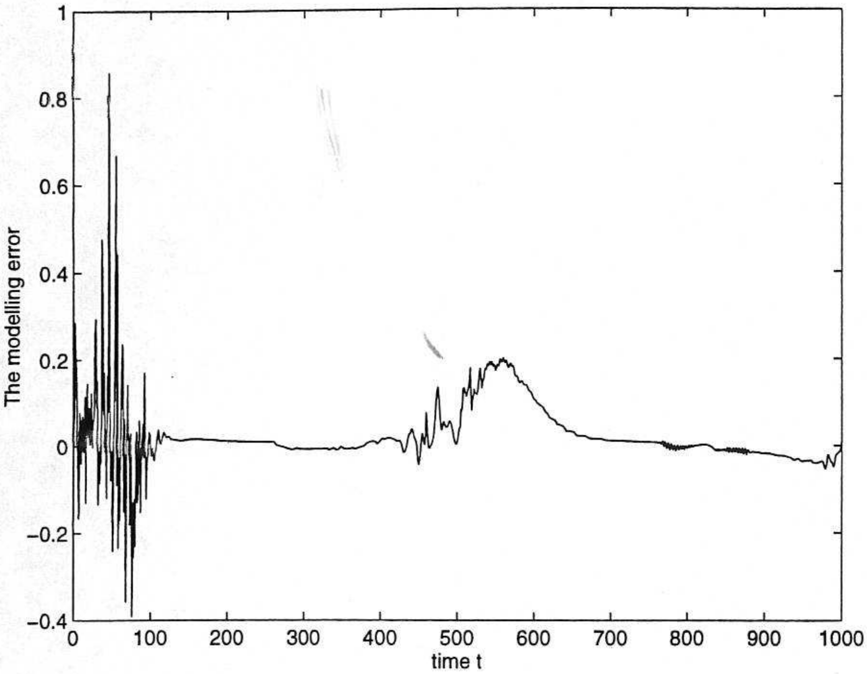

Figure 4: The modelling error of the two-step ahead predictor.

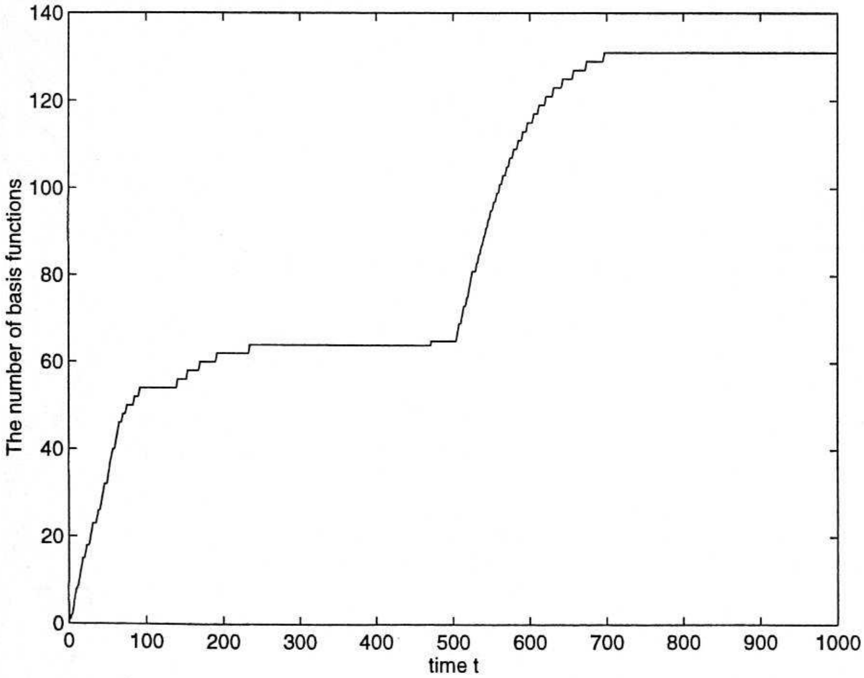

Figure 5: The number of the basis functions of the growing neural networks for the two-step ahead predictor.

References

[1] 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.

[2] F.C. Chen and H.K. Khalil, "Adaptive control of a class of nonlinear discrete-time systems using neural networks," *IEEE Trans. on Automatic Control*, vol. 40, no. 5, pp. 791-801, 1995.

[3] 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.

[4] D.W. Clarke, C. Mohtadi and P.S. Tuffs, "Generalized predictive control– Part I. The basic algorithm," *Automatica*, vol. 23, no. 2, pp. 149-160, 1987.

[5] C.R. Cutler and B.L. Ramaker, "Dynamic matrix control – A computer control algorithm," *Proceedings of the American Control Conference*, San Francisco, U.S.A., 1980.

[6] S. Fabri, and V. Kadirkamanathan, "Dynamic structure neural networks for stable adaptive control of nonlinear systems", to appear in the *IEEE Trans. on Neural Networks*, 1996.

[7] G. C. Goodwin and K. S. Sin, Adaptive Filtering Prediction and Control, *Prentice-Hall, Englewood Cliffs*, NJ, 1984.

[8] K. J. Hunt, D. Sbararo, R. Zbikowski and P. J. Gawthrop, "Neural networks for control systems — a survey," *Automatica*, Vol. 28, No. 6, pp. 1083-1112, 1992.

[9] R.M.C. De Keyser and A.R. van Cauwenberghe, "Extended prediction self-adaptive control," *Proceedings of the 7th IFAC Symposium on Identification and System Parameter Estimation*, York, UK, pp. 1255-1260, 1985.

[10] 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. 305-310, 1995.

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

[12] G.P. Liu, V. Kadirkamanathan and S.A. Billings, "Variable neural networks for nonlinear adaptive control," to be presented in the *13th IFAC Congress*, USA, 1996.

[13] 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.

[14] A.R. McIntosh, S.L. Shah and D.G. Fisher, "Analysis and tuning of adaptive generalized predictive control," *The Canadian Journal of Chemical Engineering*, vol. 69, pp. 97-110, 1991.

[15] J.D. Morningred, D.A. Mellichamp and D.E. Seborg, "A multivariable adaptive nonlinear predictive controller," *Proceedings of the American Control Conference*, Boston, pp. 2776-2781, 1991.

[16] 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.

[17] J. Platt. "A resource allocating network for function interpolation", *Neural Computation*, Vol.4, No.2, pp.213-225, 1991.

[18] T. Proll and M.N. Karim, "Real-time design of an adaptive nonlinear predictive controller," *Int. J. Control*, vol. 59, no. 3, pp. 863-889, 1994.

[19] J. Richalet, S. Abu el Ata-Doss, Ch. Arber, H.B. Kuntze, A. Jacubasch and W.Schill, "Predictive functional control. application to fast and accurate robots," *Proceedings of the 10th IFAC Congress*, Munich, Germany, 1987.

[20] 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.

[21] A.R.M. Soeterboek, H.B. Verbruggen, P.P.J. Van den Bosch and H.Bulter, "On the unification of predictive control algorithms," *Proceedings of the 29th IEEE Conference on Decision and Control*, Honolulu, U.S.A., 1990.

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

[23] X. Xie and R. Evans, "Discrete-time adaptive control for deterministic time-varing systems," Automatica, vol. 20, pp. 309-319, 1984.

[24] B.E. Ydstie, "Extended horizon adaptive control," *Proceedings of the 9th IFAC World Congress*, Budapest, Hungary, 1984.

[25] 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.