



This is a repository copy of *Variable Neural Networks for Adaptive Control of Nonlinear Systems*.

White Rose Research Online URL for this paper:
<http://eprints.whiterose.ac.uk/80096/>

Monograph:

Liu, G.P., Kadiramanathan, V. and Billings, S.A. (1995) *Variable Neural Networks for Adaptive Control of Nonlinear Systems*. Research Report. ACSE Research Report 591 .
Department of Automatic Control and Systems Engineering

Reuse

Unless indicated otherwise, fulltext items are protected by copyright with all rights reserved. The copyright exception in section 29 of the Copyright, Designs and Patents Act 1988 allows the making of a single copy solely for the purpose of non-commercial research or private study within the limits of fair dealing. The publisher or other rights-holder may allow further reproduction and re-use of this version - refer to the White Rose Research Online record for this item. Where records identify the publisher as the copyright holder, users can verify any specific terms of use on the publisher's website.

Takedown

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.



eprints@whiterose.ac.uk
<https://eprints.whiterose.ac.uk/>

X
629
.8
(S)

VARIABLE NEURAL NETWORKS FOR ADAPTIVE CONTROL OF NONLINEAR SYSTEMS

G.P. Liu, V. Kadiramanathan 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. 591

Department of Automatic Control & Systems Engineering
The University of Sheffield



August 1995

Variable Neural Networks for Adaptive Control of Nonlinear Systems

G. P. Liu, V. Kadiramanathan 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

This paper is concerned with the adaptive control of continuous-time nonlinear dynamical systems using neural networks. A novel neural network architecture, referred to as a variable neural network, is proposed and shown to be useful in approximating the unknown nonlinearities of dynamical systems. In the variable neural networks, the number of basis functions can be either increased or decreased with time according to specified design strategies so that the network will not overfit or underfit the data set. Based on the Gaussian radial basis function variable neural network, an adaptive control scheme is presented. The location of the centres and the determination of the widths of the Gaussian radial basis functions in the variable neural network are analyzed to make a compromise between orthogonality and smoothness. The weight adaptive laws developed using the Lyapunov synthesis approach guarantee the stability of the overall control scheme, even in the presence of modelling error. The tracking errors converge to the required accuracy through the adaptive control algorithm derived by combining the variable neural network and Lyapunov synthesis techniques. The operation of an adaptive control scheme using the variable neural network is demonstrated using a simulated example.

Keywords: Neural networks, nonlinear systems, adaptive control, radial basis functions.

200303157



1 Introduction

Neural networks are capable of learning and reconstructing complex nonlinear mappings and have been widely studied by control researchers in the identification analysis and design of control systems. A large number of control structures have been proposed, including supervised control [51], direct inverse control [30], model reference control [35], internal model control [13], predictive control [14] [52], gain scheduling [12], optimal decision control [10], adaptive linear control [7], reinforcement learning control [1] [3], indirect adaptive control [35] and direct adaptive control [19] [41] [46] [47]. The principal types of neural networks used for control problems are the multilayer perceptron (MLP) neural networks with sigmoidal units [30] [35] [44] and the radial basis function (RBF) neural networks [37] [39] [43].

Most of the neural network based control schemes view the problem as deriving adaptation laws using a fixed structure neural network. However, choosing this structure such as the number of basis functions (hidden units in a single hidden layer) in the neural network must be done *a priori*. This can often lead to either an over-determined or an under-determined network structure. In the discrete-time formulation, some approaches have been developed to determine the number of hidden units (or basis functions) using decision theory [4] and model comparison methods such as *minimum description length* [50] and *Bayesian methods* [29]. The problem with these methods is that they require all observations to be available and hence are not suitable for on-line control tasks, especially adaptive control. In addition, the fixed structure neural networks often need a large number of basis functions even for simple problems.

Another type of neural network structure developed for learning systems is to begin with a larger network and then to prune this [28] [32] or to begin with a smaller network and then to expand this [9], [38] until the optimal network complexity is found. Amongst these dynamic structure models, the *resource allocating network* (RAN) developed by Platt [38] is an on-line or sequential identification algorithm. The RAN is essentially a growing Gaussian radial basis function (GRBF) network whose growth criteria and parameter adaptation laws have been studied and extended further [20], [21] and applied to time-series analysis [24] and pattern classification [23]. The RAN and its extensions addressed the identification of only autoregressive systems with no external inputs and hence stability was not an issue. Recently, the growing GRBF neural network has been applied to sequential identification and adaptive control of dynamical continuous nonlinear systems with external inputs [8] [22] [27]. Though the growing neural network is much better than the fixed neural network in reducing the number of basis functions, it is still possible that this network will induce an overfitting problem. There are two main reasons for this. It is difficult to know how many basis functions are really needed for the problem and secondly the nonlinearity of a nonlinear function to be modelled is different when its variables change their value ranges. Normally, the number of basis functions in the growing neural network may increase to the one that the system needs to meet the requirement for dealing with the most complicated nonlinearity (the worst case) of the nonlinear function. Thus, it may lead to a network which has the

same size as the fixed neural networks.

To overcome the above limitations, a new network structure, referred to as the variable neural network, is proposed in this paper. The basic principle of the variable neural network is that the number of basis functions in the network can be either increased or decreased over time according to a design strategy in an attempt to avoid overfitting or underfitting. In order to model unknown nonlinearities, the variable neural network starts with a small number of initial hidden units and then adds or removes units located in a variable grid. This grid consists of a number of subgrids composed of different sized hypercubes which depend on the novelty of the observation. Since the novelty of the observation is tested, it is ideally suited for on-line control problems. The objective behind the development is to gradually approach the appropriate network complexity that is sufficient to provide an approximation to the system nonlinearities and consistent with the observations being received. By allocating GRBF units on a variable grid, only the relevant state-space traversed by the dynamical system is spanned, resulting in considerable savings on the size of the network.

The parameters of the variable neural network are adjusted by adaptation laws developed using the Lyapunov synthesis approach. Combining the variable neural network and Lyapunov synthesis techniques, the adaptive control algorithm developed for continuous dynamical nonlinear systems guarantees the stability of the whole control scheme and the convergence of the tracking errors between the reference inputs and the outputs.

The remainder of the paper is organised as follows: In Section 2, the modelling of nonlinear dynamical systems by the GRBF network is discussed and a one-to-one mapping of the state-space to form a compact network input space is also introduced. In Section 3, a variable neural network is developed, based on a proposed variable grid. The selection of the Gaussian radial basis functions for the variable neural network is discussed. The adaptive control scheme using the variable neural networks and the Lyapunov synthesis techniques is developed in Section 4. The stability of the overall control scheme and the convergence of the tracking errors are also analysed. The operation of the adaptive control scheme is demonstrated by a simulated example in Section 5.

2 Nonlinear System Modelling

Consider a class of continuous nonlinear dynamical systems which can be expressed in the canonical form [18] [36] [49]:

$$y^{(n)}(t) + F(y^{(n-1)}(t), \dots, y^{(1)}(t), y(t)) = G(y^{(n-1)}(t), \dots, y^{(1)}(t), y(t))u(t) \quad (1)$$

where $y(t)$ is the output, $u(t)$ the control input, $y^{(i)}$ the i -th derivative of the output with respect to time, and $F(\cdot)$ and $G(\cdot)$ unknown nonlinear functions. The above system represents a class of continuous-time nonlinear systems, called affine systems. The above equation can

also be transformed to the state space form

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} - \mathbf{b}F(\mathbf{x}) + \mathbf{b}G(\mathbf{x})u \quad (2)$$

$$y = x_1 \quad (3)$$

where

$$\mathbf{A} = \begin{bmatrix} 0 & \mathbf{I}_{n-1} \\ 0 & 0 \end{bmatrix} \quad (4)$$

$\mathbf{b} = [0, 0, \dots, 1]^T$ and \mathbf{I}_{n-1} is an $(n-1) \times (n-1)$ identity matrix, and $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$ is the state vector.

Due to some desirable features such as local adjustment of weights and mathematical tractability, radial basis function networks (RBF) have recently attracted considerable attention (see, for example, [2] [5] [6] [26]). Their importance has also greatly benefited from the work of Moody and Darken [31] and, Poggio and Girosi [40] who explored the relationship between regularization theory and radial basis function networks. The good approximation properties of the radial basis functions in interpolation have been well studied by Powell and his group [43]. With the use of Gaussian activation functions, each basis function in the RBF network responds only to inputs in the neighbourhood determined by the centre and width of the function. It is also known that if the variables of a nonlinear function are in compact sets, the continuous function can be approximated arbitrarily well by GRBF networks [39]. Here, the GRBF networks are used to model the nonlinearity of the system.

If x_i is not in a compact set, we introduce the following one-to-one (1-1) mapping [27]:

$$\bar{x}_i = \frac{b_{xi}x_i}{|x_i| + a_{xi}} \quad \text{for } i = 1, 2, \dots, n, \quad (5)$$

where a_{xi}, b_{xi} are positive constants, which can be chosen by the designer (e.g., a_{xi}, b_{xi} are 1). Thus, it is clear from equation (5) that $\bar{x}_i \in [-b_{xi}, b_{xi}]$ for $x_i \in (-\infty, +\infty)$. The above one-to-one mapping shows that in the n -dimensional space the entire area can be transferred into an n -dimensional hypercube denoted by the compact set \mathcal{X} . Clearly, if \mathbf{x} is already in a compact set, we only need to set $\bar{\mathbf{x}} = \mathbf{x}$.

Thus the nonlinear part $G(\mathbf{x})u - F(\mathbf{x})$ of the system can be described by the following GRBF network:

$$G(\mathbf{x})u - F(\mathbf{x}) = (\mathbf{g}^*(K)u - \mathbf{f}^*(K))^T \Phi(\bar{\mathbf{x}}, K) + \varepsilon(K) \quad (6)$$

where

$$\Phi(\bar{\mathbf{x}}, K) = [\phi(\bar{\mathbf{x}}; \mathbf{c}_1, d_1), \phi(\bar{\mathbf{x}}; \mathbf{c}_2, d_2), \dots, \phi(\bar{\mathbf{x}}; \mathbf{c}_K, d_K)]^T \quad (7)$$

$$\phi(\bar{\mathbf{x}}; \mathbf{c}_i, d_i) = \exp \left\{ -\frac{1}{d_i^2} \|\bar{\mathbf{x}} - \mathbf{c}_i\|^2 \right\}, \quad \text{for } i = 1, 2, \dots, K \quad (8)$$

$\mathbf{f}^*(K) = [f_1^*, f_2^*, \dots, f_K^*]^T$ and $\mathbf{g}^*(K) = [g_1^*, g_2^*, \dots, g_K^*]^T$ are the optimal weight vectors, $\bar{\mathbf{x}} = [\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n]^T$ the variable vector, \mathbf{c}_i the i -th centre, d_i the i -th width, $\varepsilon(K)$ the modelling error, and K the number of the basis functions.

It is known from approximation theory that the modelling error can be reduced arbitrarily by increasing the number K , i.e., the number of the linear independent basis functions $\phi(\bar{\mathbf{x}}; \mathbf{c}_i, d_i)$ in the network model. Thus, it is reasonable to assume that the modelling error $\varepsilon(K)$ is bounded by a constant ε_K , which represents the accuracy of the model and this is defined as,

$$\varepsilon_K = \sup_{t \in \mathbb{R}^+} |\varepsilon(K)| \quad (9)$$

Although ε_K can be reduced arbitrarily by increasing the number of the independent basis functions, generally when the number is greater than a small value the modelling error ε_K is improved very little by increasing the number further. It also results in a large sized network even for a simple problem. In practice, this is not realistic. In most cases, the required modelling error can be given by considering the design requirements and specifications of the system. Thus, the problem now is to find a suitable sized network to achieve the required modelling error. In other words, it is how to determine the number, centres, widths and weights of the Gaussian radial basis functions in the network.

3 Variable Neural Networks

Two main neural network structures which are widely used in on-line identification and control are the fixed neural network and the growing neural network. The fixed neural network usually needs a large number of basis functions in most cases even for a simple problem. Though the growing network is much better than the fixed network in reducing the number of the basis functions for a number of problems, it is still possible that this network will lead to an overfitting problem for some cases and this is explained in Section 1. To overcome the above limitations of the fixed and growing neural networks, a new network structure, called the variable neural network, is proposed in this section.

3.1 Variable Grid

In GRBF networks, the very important parameter is the location of the centres of the Gaussian radial basis functions over the compact set \mathcal{X} , which is the approximation region. Usually, an n -dimension grid is used to locate all centres in the gridnodes [47]. Thus, the distance between the gridnodes affects the size of the networks and also the approximation accuracy. In other words, a large distance leads to a small network and a coarser approximation, while a small distance results in a large size of network and a finer approximation. However, even if the required accuracy is given, it is very difficult to know how small the distance should

be since the underlying function is unknown. Also, the nonlinearity of the system is not uniformly complex over the set \mathcal{X} . So, here a variable grid is introduced for the location of the centres of all GRBFs in the network.

The variable grid consists of a number of different subgrids. Each subgrid is composed of equally sized n -dimensional hypercuboids. It implies that the number of the subgrids can increase or decrease with time in the grid according to a design strategy. All the subgrids are named, the initial grid is named the 1-st order subgrid, then the 2-nd order subgrid and so on. In each subgrid, there are a different number of nodes, which are denoted by their positions. Let \mathcal{N}_i denote the set of nodes in the i -th order subgrid. Thus, the set of all nodes in the grid with m subgrids is

$$\mathcal{N} = \bigcup_{i=1}^m \mathcal{N}_i \quad (10)$$

To increase the density of the gridnodes, the edge lengths of the hypercubes of the i -th order subgrid will always be less than those of the $(i-1)$ -th order subgrid. Hence the higher order subgrids have more nodes than the lower order ones. On the other hand, to reduce the density of the gridnodes, always remove some subgrids from the grid until a required density is reached.

Let all elements of the set \mathcal{N} represent the possible centres of the network. So, the more the subgrids, the more the possible centres. Since the higher order subgrids probably have some nodes which are the same as the lower order subgrids, the set of the new possible centres provided by the i -th order subgrid is defined as

$$\mathcal{P}_i = \{c : c \in \mathcal{N}_i \text{ and } c \notin \mathcal{P}_j \text{ for } j = 1, 2, \dots, i-1\} \quad (11)$$

where \mathcal{P}_0 is an empty set. It shows that the possible centre set \mathcal{P}_i corresponding to the i -th subgrid does not include those which are given by the lower order subgrids, i.e.,

$$\bigcap_{j=1}^i \mathcal{P}_j = \emptyset. \quad (12)$$

For example, in the two-dimensional case, let the edge length of rectangulars on the i -th subgrid be half of the $(i-1)$ -th subgrid. The variable grid with three subgrids is shown in Figure 1.

3.2 Variable Network

The variable neural network has the property that the number of the basis functions in the network can be either increased or decreased over time according to a design strategy.

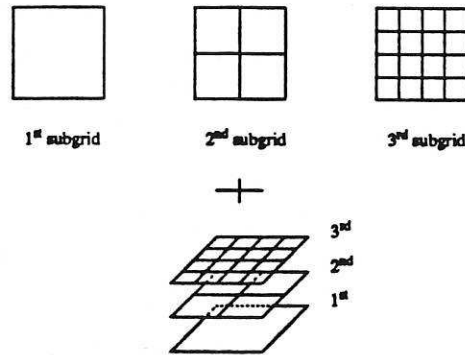


Figure 1: The variable grid with 3 subgrids.

For the problem of nonlinear modelling with neural networks, the variable network is initialised with a small number of basis function units. As observations are received, the network grows by adding some new basis functions or is pruned by removing some old ones.

To add new basis functions to the network the following two conditions must be satisfied: a) The modelling error must be greater than the required accuracy. b) The period between the two adding operations must be greater than the minimum response time to the adding operation.

To remove some old basis functions from the network, the following two conditions must be satisfied: a) The modelling error must be less than the required accuracy. b) The period between the two removing operations must be greater than the minimum response time of the removing operation.

It is known that if the grid consists of the same size n -dimension hypercubes with the edge length vector $\rho = [\rho_1, \rho_2, \dots, \rho_n]$, then the accuracy of approximating a function is in direct proportion to the norm of the edge length vector of the grid [42], i.e.,

$$\epsilon_K \propto \|\rho\| \quad (13)$$

Therefore, based on the variable grid, the structure of a variable neural network is proposed here. The network selects the centres from the node set \mathcal{N} of the variable grid. When the network needs some new basis functions, a new higher order subgrid (say, $(m+1)$ -th subgrid) is appended to the grid. The network chooses the new centres from the possible centre set \mathcal{P}_{m+1} provided by the newly created subgrid. Similarly if the network needs to be reduced, the highest order subgrid (say, m -th subgrid) is deleted from the grid. Meanwhile, the network removes the centres associated with the deleted subgrid. In this way, the network is kept to a suitable size. How to locate the centres and determine the widths of the GRBFs is discussed in the next section.

3.3 Selection of Basis Functions

It is also known that the Gaussian radial basis function has a localisation property that the influence area of the k^{th} basis function is governed by the centre \mathbf{c}_k and width d_k . In other words, once the centre \mathbf{c}_k and the width d_k are fixed, the influence area of the Gaussian radial basis function $\phi(\bar{\mathbf{x}}; \mathbf{c}_k, d_k)$ is limited in the state-space to the neighbourhood of \mathbf{c}_k .

On the basis of the possible centre set \mathcal{N} produced by the variable grid, there are large number of basis function candidates, denoted by the set \mathcal{B} . During the system operation, the state vector $\bar{\mathbf{x}}$ will gradually scan a subset of the state-space set \mathcal{X} . Since the basis functions in the GRBF network have a localised receptive field, if the neighborhood of a basis function $\phi \in \mathcal{B}$ is located 'far away' from the current state $\bar{\mathbf{x}}(t)$, its influence to the approximation is very small and could be ignored by the network. On the other hand, if the neighborhood of a basis function $\phi \in \mathcal{B}$ is near to or covers the the current state $\bar{\mathbf{x}}(t)$, it will play a very important role in the approximation. Thus it should be kept if it is already in the network or added into the network if it is not in.

Given any point $\bar{\mathbf{x}}$, the nearest node $\bar{\mathbf{x}}_i^+ = [\bar{x}_{i1}^+, \bar{x}_{i2}^+, \dots, \bar{x}_{in}^+]^T$ to it in the i -th subgrid can be calculated by,

$$\bar{x}_{ij}^+ = \text{round} \left(\frac{\bar{x}_j}{\delta_{ij}} \right) \delta_{ij} \quad (14)$$

for $j = 1, 2, \dots, n$, where $\text{round}(\cdot)$ is an operator for rounding the number (\cdot) to the nearest integer, for example, $\text{round}(2.51) = 3$, and δ_{ij} is the edge length of the hypercube corresponding to the j -th element of the vector $\bar{\mathbf{x}}$ in the i -th subgrid. Without lose of generality, let $\delta_i = \delta_{i1} = \delta_{i2} = \dots = \delta_{in}$.

Define m hyperspheres corresponding to the m subgrids, respectively,

$$\mathcal{H}_i(\bar{\mathbf{x}}_i^+, \sigma_i) = \left\{ \bar{\mathbf{x}} : \sum_{j=1}^n (x_{ij} - \bar{x}_{ij}^+)^2 \leq \sigma_i^2 \right\} \quad (15)$$

for $i = 1, 2, \dots, m$, where σ_i is the radius of the i -th hypersphere. In order to get a suitable sized variable network, choose the centres of the basis functions from the nodes contained in the different hyperspheres $\mathcal{H}_i(\bar{\mathbf{x}}_i^+, \sigma_i)$, which are centred in the nearest nodes $\bar{\mathbf{x}}_i^+$ to $\bar{\mathbf{x}}$ in the different subgrids with radius σ_i , for $i = 1, 2, \dots, m$. For the sake of simplicity, it is assumed that the basis function candidates whose centres are in the set \mathcal{P}_i have the same width d_i and $d_i < d_{i-1}$. Thus, for the higher order subgrids, use the smaller radius, i.e.

$$\sigma_m < \sigma_{m-1} < \dots < \sigma_1 \quad (16)$$

Usually, choose

$$\sigma_i = \gamma_1 \sigma_{i-1} \quad (17)$$

where γ_1 is a constant and less than 1. Thus, the chosen centres from the set \mathcal{P}_i are given by the set:

$$\mathcal{C}_i = \{ \mathbf{c} : \mathbf{c} \in \mathcal{P}_i \text{ and } \mathbf{c} \in \mathcal{H}_i(\bar{\mathbf{x}}_i^+, \sigma_i) \} \quad (18)$$

In order that the basis function candidates in the set \mathcal{P}_i which are less than an activation threshold to the nearest grid node \bar{x}_i^+ in the i -th subgrid are outside the set $\mathcal{H}_i(\bar{x}_i^+, \sigma_i)$, it can be deduced from (8) and (15) that the σ_i must be chosen to be

$$\sigma_i \geq \sqrt{\log(\delta_{min}^{-1})} d_i \quad (19)$$

for $i = 1, 2, \dots, m$, where $\delta_{min} \in (0, 1)$ represents the activation threshold.

Thus, the centre set of the network is given by the union of the centre sets \mathcal{C}_i , for $i = 1, 2, \dots, m$, that is,

$$\mathcal{C} = \bigcup_{i=1}^m \mathcal{C}_i \quad (20)$$

For example, in the 2-dimension case, the radii are chosen to be the same as the edge lengths of the squares in the subgrids, that is,

$$\sigma_i = \delta_i, \quad \text{for } i = 1, 2, \dots, m. \quad (21)$$

The chosen centres in the variable grid with four subgrids are as shown in Figure 2.

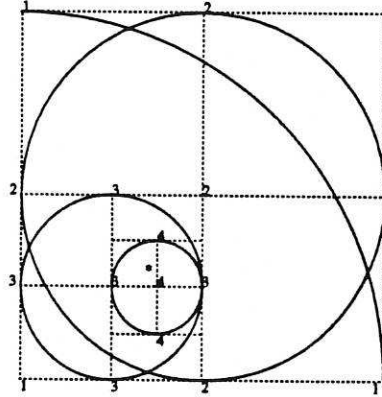


Figure 2: The location of centres in the variable grid with 4 subgrids. The number i ($i = 1, 2, 3, 4$) denotes the centres chosen from the i -th subgrid.

Now, consider how to choose the width d_k of the k^{th} basis function. The angle between the two GRBFs $\phi(\mathbf{x}; \mathbf{c}_i, d_i)$ and $\phi(\mathbf{x}; \mathbf{c}_j, d_j)$ is defined as

$$\theta_{ij} = \cos^{-1} \left(\frac{\langle \phi(\mathbf{x}; \mathbf{c}_i, d_i), \phi(\mathbf{x}; \mathbf{c}_j, d_j) \rangle}{\|\phi(\mathbf{x}; \mathbf{c}_i, d_i)\| \|\phi(\mathbf{x}; \mathbf{c}_j, d_j)\|} \right) \quad (22)$$

where $\langle \cdot, \cdot \rangle$ is the inner product in Hilbert space. The angle can be given by [20],

$$\theta_{ij} = \cos^{-1} \left(\left(\frac{2\sqrt{\xi}}{\xi + 1} \right)^{\frac{\eta}{4}} \phi(\mathbf{c}_j; \mathbf{c}_i, d_i)^{\frac{\xi}{2\xi+2}} \right). \quad (23)$$

where $\xi = d_i^2/d_j^2$. It shows from the above that the $\cos(\theta_{ij})$ depends on three factors: the dimension n , the width ratio ξ and the output of a basis function at the centre of the other basis function, $\phi(\mathbf{c}_j; \mathbf{c}_i, d_i)$.

If the centres of the two basis functions are chosen from the same subgrid, i.e., $\xi = 1$, it is clear from Eq.(23) that

$$\cos(\theta_{ij}) = \phi(\mathbf{c}_j; \mathbf{c}_i, d_i)^{\frac{1}{n}}. \quad (24)$$

On the other hand, if the centres of the two basis functions are from different subgrids, it is possible that their centres are very close. The worst case will be when $\phi(\mathbf{c}_j; \mathbf{c}_i, d_i)$ is near to 1. In this case, the angle between the two basis functions can be written as

$$\cos(\theta_{ij}) \leq \left(\frac{2\sqrt{\xi}}{\xi + 1} \right)^{\frac{n}{4}} \quad (25)$$

Given the centre \mathbf{c}_k , in order to assign a new basis function $\phi(\mathbf{x}; \mathbf{c}_k, d_k)$ that is nearly orthogonal to all existing basis functions, the angle between the GRBFs should be as large as possible. The width d_k should therefore be reduced. However, reducing d_k increases the curvature of $\phi(\mathbf{x}; \mathbf{c}_k, d_k)$ which in turn gives a less smooth function and can lead to *overfitting* problems. Thus, to make a trade-off between the orthogonality and the smoothness, it can be deduced from Eqs. (24) and (25) that the width d_k , which ensures the angles between GRBF units are not less than the required minimum angle θ_{min} , should satisfy

$$\xi \geq \left(\frac{1 + \sqrt{1 - \cos^{8/n}(\theta_{min})}}{\cos^{4/n}(\theta_{min})} \right)^2 \quad (26)$$

or

$$\xi \leq \left(\frac{1 - \sqrt{1 - \cos^{8/n}(\theta_{min})}}{\cos^{4/n}(\theta_{min})} \right)^2 \quad (27)$$

and

$$d_k \leq \sqrt{\log^{-4} \cos(\theta_{min})} \delta_k \quad (28)$$

For example, assume that the s_0 satisfies Eq.(26). If the width of the basis functions whose centres are located in the set \mathcal{C}_i , which corresponds to the i -th subgrid with $\delta_i = \xi_0 \delta_{i-1}$, is chosen to be $d_i = \xi_0 d_{i-1}$ and the width d_1 of the basis functions associated to the initial grid satisfies

$$d_1 \leq \sqrt{\log^{-4} \cos(\theta_{min})} \delta_1 \quad (29)$$

then the smallest angle between all basis functions are not less than the required minimum angle θ_{min} .

Therefore, based on a variable grid with m subgrids, the nonlinear function approximated by the GRBF network in Eq. (6) can also be expressed by

$$G(\mathbf{x})u - F(\mathbf{x}) = \sum_{i=1}^m \sum_{j=1}^{m_i} (f_{i+j}^* + g_{i+j}^* u) \phi(\bar{\mathbf{x}}, \mathbf{c}_{i+j}, d_i) + \varepsilon(K) \quad (30)$$

where

$$K = \sum_{i=1}^m m_i \quad (31)$$

\mathbf{c}_{i+j} is the j -th element of the set C_i , m_i is the number of its elements, and f_{i+j}^* and g_{i+j}^* are the optimal weights. So, the next step is how to obtain the estimates of the weights.

4 Adaptive Control

The stability of the overall control scheme is an important issue in the design of the system. The overall stability depends not only on the particular control approach that is chosen but also on the control laws that are used. In practice, one of the design objectives for a system is that the tracking error between inputs and outputs should converge to the required accuracy. Those problems are solved here by developing a stable adaptive control law based on Lyapunov stability techniques [25] and the variable GRBF network discussed in Section 3.

4.1 Adaptation Laws

We assume that the basis functions $\phi(\bar{\mathbf{x}}; \mathbf{c}_k, d_k)$ for $k = 1, 2, \dots, K$ are given. Section 4.2 will discuss how the basis functions of the network model are chosen.

The control objective is to force the plant state vector \mathbf{x} to follow a specified desired trajectory $\mathbf{y}_d = [y_d, y_d^{(1)}, \dots, y_d^{(n-1)}]^T$. The tracking error vector and the weight error vectors, respectively, are defined as,

$$\mathbf{e} = \mathbf{x} - \mathbf{y}_d \quad (32)$$

$$\tilde{\mathbf{f}}(K) = \mathbf{f}^*(K) - \mathbf{f}(K) \quad (33)$$

$$\tilde{\mathbf{g}}(K) = \mathbf{g}^*(K) - \mathbf{g}(K) \quad (34)$$

where $\mathbf{f}(K)$ and $\mathbf{g}(K)$ are the estimated weight vectors. From Eq.(1), it can be shown that

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{b}(\mathbf{g}^T(K)u - \mathbf{f}^T(K))\Phi(\bar{\mathbf{x}}, K) + \mathbf{b}(\tilde{\mathbf{g}}^T(K)u - \tilde{\mathbf{f}}^T(K))\Phi(\bar{\mathbf{x}}, K) + \mathbf{b}\varepsilon(K) \quad (35)$$

Hence, from Eqs.(32)-(35), the dynamical expression of the tracking error is,

$$\dot{\mathbf{e}} = \mathbf{A}\mathbf{e} - \mathbf{b}y_d^{(n)} + \mathbf{b}(\mathbf{g}^T(K)u - \mathbf{f}^T(K))\Phi(\bar{\mathbf{x}}, K) + \mathbf{b}(\tilde{\mathbf{g}}^T(K)u - \tilde{\mathbf{f}}^T(K))\Phi(\bar{\mathbf{x}}, K) + \mathbf{b}\varepsilon(K) \quad (36)$$

One approach to this problem is to take the control input satisfying

$$\mathbf{g}^T(K)\Phi(\bar{\mathbf{x}}, K)u = y_d^{(n)} + \mathbf{f}^T(K)\Phi(\bar{\mathbf{x}}, K) + \mathbf{a}^T \mathbf{e} \quad (37)$$

where the vector $\mathbf{a} = [a_1, a_2, \dots, a_n]^T$ makes the following matrix

$$\mathbf{A}_a = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ -a_1 & -a_2 & -a_3 & \dots & -a_n \end{bmatrix} \quad (38)$$

stable, i.e., all the eigenvalues are in the open left plane. The control input consists of a linear combination of the tracking errors $\mathbf{a}^T \mathbf{e}$, the adaptive part $\mathbf{f}^T(K)\Phi(\bar{\mathbf{x}}, K)$ which will attempt to estimate, and cancel, the unknown function $F(\cdot)$, and $y_d^{(n)}$ is a feedforward of the n -th derivative of the desired trajectory.

Consider the following Lyapunov function,

$$V(\mathbf{e}, \tilde{\mathbf{f}}, \tilde{\mathbf{g}}) = \mathbf{e}^T \mathbf{P} \mathbf{e} + \frac{1}{\alpha} \tilde{\mathbf{f}}^T(K) \tilde{\mathbf{f}}(K) + \frac{1}{\beta} \tilde{\mathbf{g}}^T(K) \tilde{\mathbf{g}}(K) \quad (39)$$

where \mathbf{P} is chosen to be a positive definite matrix so that the matrix $\mathbf{Q} = -\mathbf{P} \mathbf{A}_a - \mathbf{A}_a^T \mathbf{P}$ is also a positive definite matrix, and α and β are positive constants which will appear in the sequential adaptation laws, also referred to as the learning or adaptation-rates. Using Eq. (36), the derivative of the Lyapunov function V with respect to time is given by

$$\begin{aligned} \dot{V}(\mathbf{e}, \tilde{\mathbf{f}}, \tilde{\mathbf{g}}) &= -\mathbf{e}^T \mathbf{Q} \mathbf{e} + 2\mathbf{P}_n^T \mathbf{e} ((\tilde{\mathbf{g}}^T(K)u - \tilde{\mathbf{f}}^T(K))\Phi(\bar{\mathbf{x}}, K) + \varepsilon(K)) \\ &\quad + 2\alpha^{-1} \tilde{\mathbf{f}}^T(K) \dot{\tilde{\mathbf{f}}}(K) + 2\beta^{-1} \tilde{\mathbf{g}}^T(K) \dot{\tilde{\mathbf{g}}}(K) \end{aligned} \quad (40)$$

where the vector \mathbf{P}_n is the n -th row of the matrix \mathbf{P} , i.e., $\mathbf{P}_n = [p_{n1}, p_{n2}, \dots, p_{nn}]$.

Since \mathbf{f}^* is a constant vector, we have that $\dot{\tilde{\mathbf{f}}} = -\dot{\mathbf{f}}$, similarly, $\dot{\tilde{\mathbf{g}}} = -\dot{\mathbf{g}}$. If there is no modelling error, i.e., $\varepsilon(K) = 0$, the weight vectors \mathbf{f} and \mathbf{g} can simply be generated according to the following standard adaptation laws: $\dot{\mathbf{f}}(K) = -\alpha \mathbf{P}_n^T \mathbf{e} \Phi(\bar{\mathbf{x}}; K)$ and $\dot{\mathbf{g}}(K) = \beta \mathbf{P}_n^T \mathbf{e} u \Phi(\bar{\mathbf{x}}; K)$. In the presence of a modelling error $\varepsilon(K)$, to ensure the stability of the system, a lot of algorithms, e.g., the fixed or switching σ -modification [16] [17], ε -modification [33] and the dead-zone methods [34] [48], can be applied to modify the above standard adaptation laws.

Define the following sets:

$$\mathcal{F}_1 = \{\mathbf{f} : \|\mathbf{f}\| < M_1 \text{ or } (\|\mathbf{f}\| = M_1 \text{ and } \mathbf{P}_n^T \mathbf{e} \mathbf{f}^T \Phi(\bar{\mathbf{x}}) \geq 0)\} \quad (41)$$

$$\mathcal{F}_2 = \{\mathbf{f} : \|\mathbf{f}\| = M_1 \text{ and } \mathbf{P}_n^T \mathbf{e} \mathbf{f}^T \Phi(\bar{\mathbf{x}}) < 0\} \quad (42)$$

$$\mathcal{G}_1 = \{\mathbf{g} : \|\mathbf{g}\| < M_2 \text{ or } (\|\mathbf{g}\| = M_2 \text{ and } \mathbf{P}_n^T \mathbf{e} u \mathbf{g}^T \Phi(\bar{\mathbf{x}}) \leq 0)\} \quad (43)$$

$$\mathcal{G}_2 = \{\mathbf{g} : \|\mathbf{g}\| = M_2 \text{ and } \mathbf{P}_n^T \mathbf{e} u \mathbf{g}^T \Phi(\bar{\mathbf{x}}) > 0\} \quad (44)$$

where M_1 and M_2 are positive constants.

Here, in order to avoid parameter drift in the presence of modelling error, the application of the projection algorithm [11] [15] [41] gives the following adaptive laws for the parameter estimates \mathbf{f} and \mathbf{g} :

$$\dot{\mathbf{f}}(K) = \begin{cases} -\alpha \mathbf{P}_n^T \mathbf{e} \Phi(\bar{\mathbf{x}}; K) & \text{if } \mathbf{f}(K) \in \mathcal{F}_1 \\ -\alpha \mathbf{P}_n^T \mathbf{e} \Phi(\bar{\mathbf{x}}; K) + \alpha M_1^{-2} \mathbf{P}_n^T \mathbf{e} \mathbf{f}^T(K) \Phi(\bar{\mathbf{x}}; K) \mathbf{f}(K) & \text{if } \mathbf{f}(K) \in \mathcal{F}_2 \end{cases} \quad (45)$$

$$\dot{\mathbf{g}}(K) = \begin{cases} \beta \mathbf{P}_n^T \mathbf{e} u \Phi(\bar{\mathbf{x}}; K) & \text{if } \mathbf{g}(K) \in \mathcal{G}_1 \\ \beta \mathbf{P}_n^T \mathbf{e} u \Phi(\bar{\mathbf{x}}; K) - \beta M_2^{-2} \mathbf{P}_n^T \mathbf{e} u \mathbf{g}^T(K) \Phi(\bar{\mathbf{x}}; K) \mathbf{g}(K) & \text{if } \mathbf{g}(K) \in \mathcal{G}_2 \end{cases} \quad (46)$$

It is clear that if the initial weights are chosen such that $\mathbf{f}(K, 0) \in \mathcal{F}_1 \cup \mathcal{F}_2$ and $\mathbf{g}(K, 0) \in \mathcal{G}_1 \cup \mathcal{G}_2$, then the weight vectors \mathbf{f} and \mathbf{g} are confined to the sets $\mathcal{F}_1 \cup \mathcal{F}_2$ and $\mathcal{G}_1 \cup \mathcal{G}_2$, respectively. With use of the adaptive laws (45) and (46), Eq. (40) becomes,

$$\dot{V}(\mathbf{e}, \tilde{\mathbf{f}}, \tilde{\mathbf{g}}) \leq -\mathbf{e}^T \mathbf{Q} \mathbf{e} + 2 \sum_{i=1}^n |p_{ni}| |e_i| \varepsilon_K \quad (47)$$

For the sake of simplicity, the positive definite matrix \mathbf{Q} is assumed to be diagonal, i.e., $\mathbf{Q} = \text{diag}[q_1, q_2, \dots, q_n]$. Also define

$$\Theta(\zeta) = \left\{ \mathbf{e} : \sum_{i=1}^n q_i \left(|e_i| - \frac{|p_{ni}|}{q_i} \zeta \right)^2 \leq \sum_{i=1}^n \frac{p_{ni}^2 \zeta^2}{q_i} \right\} \quad (48)$$

where ζ is a positive variable, i.e., $\zeta \geq 0$.

If there is no modelling error (i.e., $\varepsilon_K = 0$), then from Eq.(47), \dot{V} is negative semidefinite. Hence the stability of the overall identification scheme is guaranteed and

$$\mathbf{e} \rightarrow 0, \quad \tilde{\mathbf{f}} \rightarrow 0, \quad \tilde{\mathbf{g}} \rightarrow 0. \quad (49)$$

On the other hand, in the presence of modelling error, if $\mathbf{e} \notin \Theta(\varepsilon_K)$, it is easy to show from (47) that \dot{V} is still negative and the tracking errors will converge to the set $\Theta(\varepsilon_K)$. But, if $\mathbf{e} \in \Theta(\varepsilon_K)$, it is possible that $\dot{V} > 0$, which implies that the weight vectors $\mathbf{f}(K)$ and $\mathbf{g}(K)$ may drift to infinity over time. The adaptive laws (45) and (46) avoid this drift by limiting the upper bounds of the weights. Thus the tracking error always converges to the set $\Theta(\varepsilon_K)$ and the overall control scheme will remain stable in the case of modelling error.

4.2 Adaptive Control Algorithm

From the set $\Theta(\varepsilon_K)$ which gives a relationship between the tracking and modeling errors, it can be shown that the tracking error depends on the modelling error. If the modelling error

ε_K is known, then the set $\Theta(\varepsilon_K)$ to which the tracking error will converge is also known. However, in most cases the upper bound ε_K is unknown.

In practice, control systems are usually required to keep the tracking errors within prescribed bounds, that is,

$$|e_i| \leq \varepsilon_{i0}, \quad \text{for } i = 1, 2, \dots, n \quad (50)$$

where ε_{i0} is the required accuracy. At the beginning, it is very difficult to know how many neural network units are needed to achieve the above control requirements. In order to find a suitable sized network for this control problem, first set lower and upper bounds for the tracking errors, which are functions of time t , and then try to find a variable network such that

$$|e_i| \in [\Delta_i^L(t), \Delta_i^U(t) + \varepsilon_{i0}], \quad \text{for } i = 1, 2, \dots, n \quad (51)$$

where $\Delta_i^L(t), \Delta_i^U(t)$ are monodecreasing functions of time t , respectively. Those bounds are usually defined as

$$\Delta_i^U(t) = \beta_U^t \Delta_i^U(0) \quad (52)$$

$$\Delta_i^L(t) = \beta_L^t \Delta_i^L(0) \quad (53)$$

where β_U, β_L are constants and less than 1, $\Delta_i^U(0), \Delta_i^L(0)$ are the initial values. It is clear that $\Delta_i^U(t), \Delta_i^L(t)$ decrease with time t . As $t \rightarrow \infty$, $\Delta_i^U(t), \Delta_i^L(t)$ approach 0. Thus, in this way the tracking errors reach the required accuracies given in Eq.(50).

According to the relationship between the modelling error and the tracking error, it is easy to know that given the lower and upper bounds $\Delta_i^U(t), \Delta_i^L(t) + \varepsilon_{i0}$ of the tracking errors the modelling error corresponding to the above should be

$$\varepsilon_K(t) \in [\varepsilon_L(t), \varepsilon_U(t)] \quad (54)$$

It is easy to know that the area that the set $\Theta(\zeta)$ covers is a hyperellipsoid with the center

$$\left(\frac{|p_{n1}|}{q_1} \zeta, \frac{|p_{n2}|}{q_2} \zeta, \dots, \frac{|p_{nn}|}{q_n} \zeta \right). \quad (55)$$

Thus, it can be deduced from the set $\Theta(\varepsilon_K(t))$ given by Eq.(48) that the upper bound $\varepsilon_U(t)$ and the lower bound $\varepsilon_L(t)$ are given by

$$\varepsilon_L(t) = \max_{i=1,2,\dots,n} \left(\frac{|p_{ni}|}{q_i} + \left(\sum_{j=1}^n \frac{p_{nj}^2}{q_i q_j} \right)^{0.5} \right)^{-0.5} \Delta_i^L(t) \quad (56)$$

$$\varepsilon_U(t) = \min_{i=1,2,\dots,n} \left(\frac{|p_{ni}|}{q_i} + \left(\sum_{j=1}^n \frac{p_{nj}^2}{q_i q_j} \right)^{0.5} \right)^{-0.5} (\Delta_i^U(t) + \varepsilon_{i0}) \quad (57)$$

Hence, if the tracking error $e \notin \Theta(\varepsilon_U(t))$, the network needs more basis functions. Add the $(m+1)$ -th order subgrid to the grid. The parameters associated with the GRBF units are then changed as follows:

$$\sigma_{m+1} = \gamma_1 \sigma_m \quad (58)$$

$$\delta_{m+1} = \gamma_2 \delta_m \quad (59)$$

$$d_{m+1} = \gamma_3 d_m \quad (60)$$

$$\mathcal{P} = \bigcup_{i=1}^{m+1} \mathcal{P}_i \quad (61)$$

$$\mathcal{C} = \bigcup_{i=1}^{m+1} \mathcal{C}_i \quad (62)$$

$$K = \sum_{i=1}^{m+1} m_i \quad (63)$$

where γ_i , for $i = 1, 2, 3$, is a constant and less than 1.

But, if the tracking error $e \in \Theta(\varepsilon_L(t))$, the network needs to remove some basis functions. Just remove the units associated with m -th subgrid. The parameters associated with the GRBF units are then changed as follows:

$$\mathcal{P} = \bigcup_{i=1}^{m-1} \mathcal{P}_i \quad (64)$$

$$\mathcal{C} = \bigcup_{i=1}^{m-1} \mathcal{C}_i \quad (65)$$

$$K = \sum_{i=1}^{m-1} m_i \quad (66)$$

In both above cases, the adaptive laws of the weights are still given in the form of (45) and (46), based on the above changed parameters. For the two dimensional case, the convergence area is shown in Figure 3. At the beginning, the convergence area of the tracking area is E_0 . Finally it approaches to the expected convergence area E_∞ , that is, $|e_i| \leq \varepsilon_{i0}$, for $i = 1, 2$.

5 Simulation Results

The dynamical system used in the simulation example is given in [47],

$$\ddot{y} - 4 \left(\frac{\sin(4\pi y)}{\pi y} \right) \left(\frac{\sin(\pi \dot{y})}{\pi \dot{y}} \right)^2 = (2 + \sin(3\pi y - 1.5\pi))u \quad (67)$$

which is a second order nonlinear system.

The parameter values used in this example are as follows: the reference input $y_d = \sin(t)$; the initial value of the output $y(0) = 0.5$; the initial value of the output derivative $\dot{y}(0) = 0$;

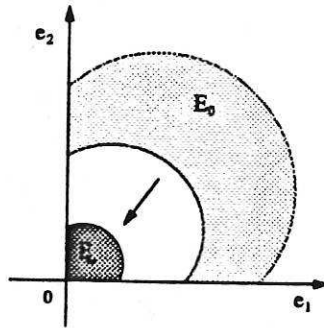


Figure 3: The two-dimensional convergence area.

the required accuracy of the tracking error vector $[\varepsilon_{10}, \varepsilon_{20}] = [0.05, 0.1]$; the constants $\beta_U = \beta_L = 0.96$; the initial values $\Delta_i^U(0) = 0.005, \Delta_i^L(0) = 0.05$, for $i = 1, 2$; the required minimum angle between the GRBFs $\cos(\theta_{min}) = 0.951$; the edge length of the rectangles in the 1-st subgrid is $\delta_1 = 0.5$; the radius of centre selection in the 1-st subgrid $\sigma_1 = 0.99$; the width of the GRBF units corresponding to the 1-st subgrid $d_1 = 1.11$; the activation threshold $\delta_{min} = 0.45$; the initial number of the variable networks is 45; the vector $a = [1, 1]$; the matrix $P = [[0.75, 0.5]^T, [0.5, 1]^T]$; the adaptation rates $\alpha = 1.5$ and $\beta = 3$.

The parameters associated with the variable network are

$$\delta_i = 0.618\delta_{i-1}, \quad \sigma_i = 0.618\sigma_{i-1}, \quad d_i = 0.618d_{i-1} \quad (68)$$

for $i = 2, 3, \dots, m$. The maximum of m (the number of the subgrids) is limited to be 11.

The weights are adaptively adjusted by the laws in Eqs.(45) and (46). The adaptive control law is given by Eq.(37). The results of the simulation are shown in Figures 4-6. Though the difference between the system output and the desired output is very large at the beginning, the system is still stable and the tracking error asymptotically converges to the expected range, which is also shown in Fig.5. As it can be seen from Fig. 6, the number of GRBF units in the neural network also converges in a period of time.

6 Conclusions

A variable neural network structure has been proposed, where the number of the basis functions in the network can be either increased or decreased over time according to some design strategy to avoid either overfitting or underfitting. In order to model unknown nonlinearities of nonlinear systems, the variable neural network starts with a small number of initial hidden units, then adds or removes units on a variable grid consisting of a variable number of subgrids with different sized hypercubes, based on the novelty of observation. The adaptive

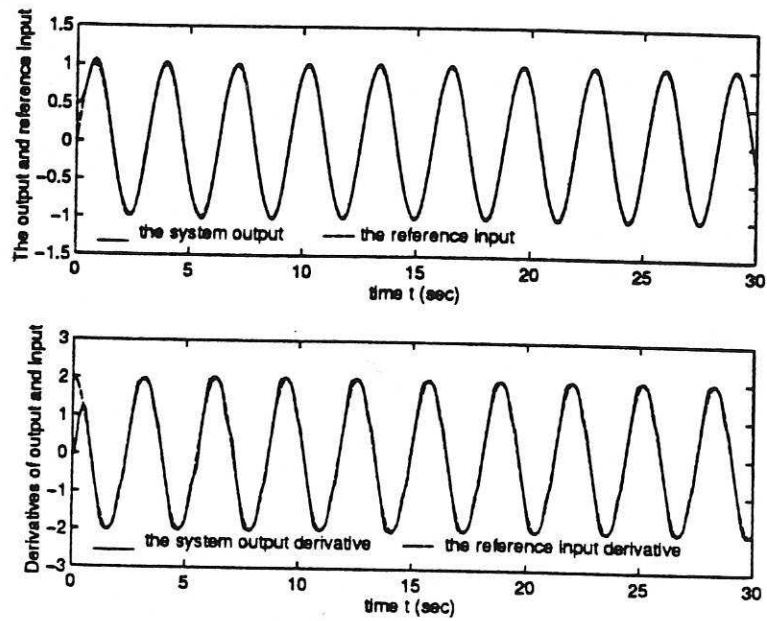


Figure 4: The reference input $y_d(t)$, output $y(t)$, reference input derivative $\dot{y}_d(t)$ and output derivative $\dot{y}(t)$ of the system.

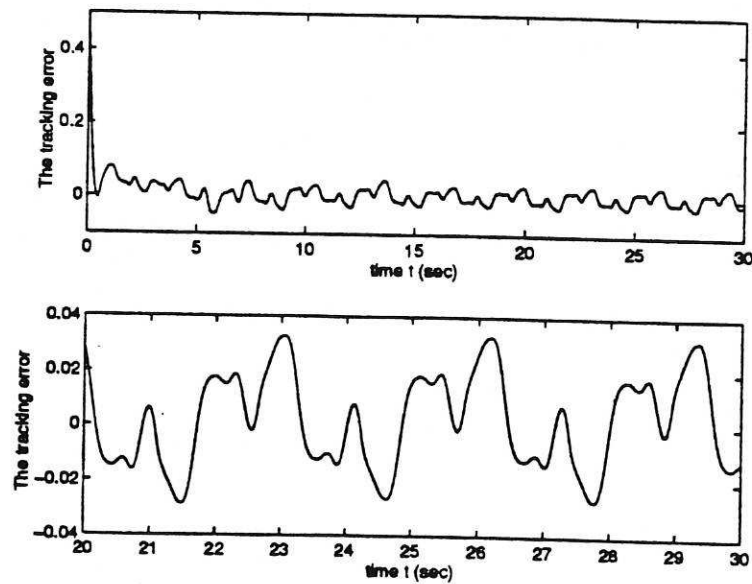


Figure 5: The tracking error $y(t) - y_d(t)$ of the system.

control algorithm, developed by combining the variable Gaussian radial basis function network and Lyapunov synthesis techniques, guarantees the stability of the control system and the convergence of the tracking errors. The number of GRBF units in the neural network also converges by introducing the monodecreasing upper and lower bounds of the tracking errors.

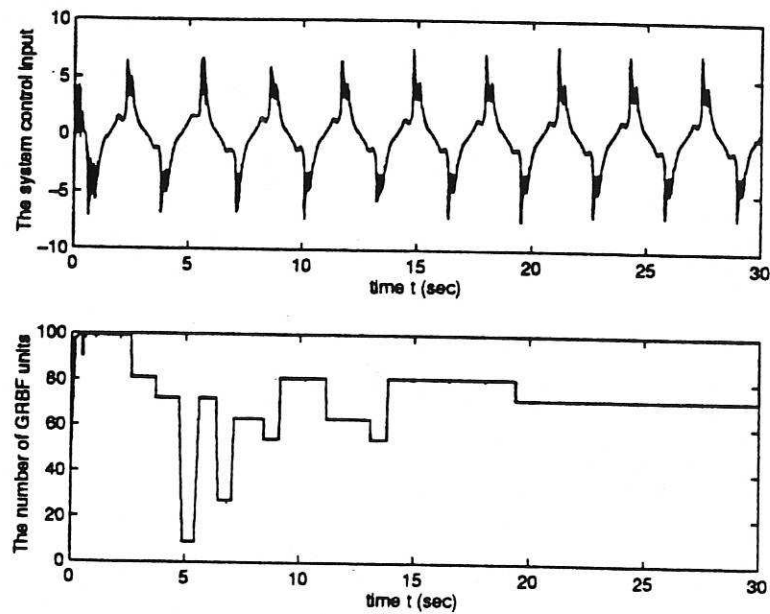


Figure 6: The control input $u(t)$ of the system and the number K of GRBF units in the variable neural network.

The results of the simulation example illustrate the operation of the variable neural network for adaptive nonlinear system control.

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] C. W. Anderson, "Learning to control an inverted pendulum using neural networks," *IEEE Control Systems magazine*, Vol. 9, pp. 31-37, 1989.
- [2] P. J. Antsaklis, (ed.), Special Issue on Neural Networks in Control Systems, *IEEE Control Systems magazine*, Vol.10, No.3, 1990.
- [3] A. G. Barto, *Neural Network for Control*, MIT Press, Cambridge, MA, 1990.
- [4] E. Baum and D. Haussler, "What size net gives valid generalization", *Neural Computation*, Vol.1, No.1, 1989.
- [5] 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.
- [6] 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.

- [7] S. R. Chi, R. Shoureshi and M. Tenorio, "Neural networks for system identification," *IEEE Control Systems Magazine*, Vol. 10, pp. 31-34, 1990.
- [8] S. Fabri, and V. Kadiramanathan, "Dynamic structure neural networks for stable adaptive control of nonlinear systems", to appear in the *IEEE Trans. on Neural Networks*, 1996.
- [9] S. E. Fahlman and C. Lebiere, "The cascade-correlation architecture", In D. S. Touretzky (ed.), *Advances in Neural Information Processing Systems 2*, Morgan Kaufmann, CA: San Mateo, 1990.
- [10] K. S. Fu, "Learning control systems — review and outlook," *IEEE Trans. on Aut. Control*, Vol. 16, pp.210-221, 1970.
- [11] 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.
- [12] A. Guez, J. L. Elibert and M. Kam, "Neural network architecture for control," *IEEE Control Systems Magazine*, Vol. 8, pp. 22-25, 1988.
- [13] K. J. Hunt and D. Sbararo, "Neural networks for nonlinear internal model control," *Proc. IEE Pt. D*, Vol. 138, pp. 431-438, 1991.
- [14] 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.
- [15] P. A. Ioannou and A. Datta, "Robust adaptive control: design, analysis and robustness bounds," in *Foundations of Adaptive Control*, P.V. Kokotovic (ed.), pp. 71-152, Springer-Verlag, Berlin, 1991.
- [16] P. A. Ioannou and P. V. Kokotovic, *Adaptive Systems with Reduced Models*, Springer-Verlag, New York, NY, 1983.
- [17] P. A. Ioannou and Tsakalis, "A robust direct adaptive control," *IEEE Trans. Aut. Control*, Vol. 31, pp. 1033-1043, 1986
- [18] A. Isidori, *Nonlinear Control Systems: An Introduction*, Springer-Verlag, Berlin, 1989.
- [19] A. Karakasoglu, S. L. Sudharsanan and M. K. Sundareshan, "Identification and decentralized adaptive control using neural networks with application to robotic manipulators," *IEEE Trans. on Neural Networks*, Vol. 4, No. 6, pp. 919-930, 1993.
- [20] V. Kadiramanathan, *Sequential Learning in Artificial Neural Networks*, Ph.D Thesis, University of Cambridge, UK, 1991.
- [21] V. Kadiramanathan, "A statistical inference based growth criteria for the RBF network," In *Proc. IEEE Workshop on Neural Networks for Signal Processing*, pp. 12-21, 1994.
- [22] V. Kadiramanathan 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.
- [23] V. Kadiramanathan and M. Niranjana, "Application of an architecturally dynamic network for speech pattern classification", In *Proceedings of the Institute of Acoustics*, Vol.14, Part 6, pp.343-350, 1992.

- [24] V. Kadiramanathan and M. Niranjan, "A function estimation approach to sequential learning with neural networks", *Neural Computation*, Vol.5, pp.954-957, 1993.
- [25] I. D. Landau, *Adaptive Control — The Model Reference Approach*, Marcel Dekker, Inc., New York, 1979.
- [26] G. P. Liu and V. Kadiramanathan, "Learning with Multiobjective Criteria," *Proceedings of Fourth International Conference on Artificial Neural Networks*, Cambridge, U.K., pp.35-40, 1995.
- [27] G. P. Liu V. Kadiramanathan and S. A. Billings, "Identification of nonlinear systems using growing RBF networks," *Proceedings of the Third European Control Conference*, Rome, 1995.
- [28] Y. LeCun, J.S. Denker and S.A. Solla, "Optimal brain damage." *In advances in Neural Information Processing Systems 2* (D.S. Touretzky, ed.) pp. 598-605. San Mateo, CA: Morgan Kaufmann, 1990.
- [29] D. J. C. MacKay, "Bayesian interpolation", *Neural Computation*, Vol.4, No.3, pp.415-447, 1992.
- [30] T. M. Miller, R. S. Sutton III and P. J. Werbos (eds), *Neural Networks for Control*, MIT Press, Cambridge, MA, 1990.
- [31] J. Moody and C. J. Darken, "Fast learning in networks of locally-tuned processing units", *Neural Computation*, Vol.1, pp.281-294, 1989.
- [32] M. C. Mozer and P. Smolensky, "Skeletonization: A technique for trimming the fat from a network via relevance assignment", In D. S. Touretzky (ed.), *Advances in Neural Information Processing Systems 1*, Morgan Kaufmann, CA: San Mateo, 1989.
- [33] K. S. Narendra and A. M. Annaswamy, "A new adaptive law for robust adaptation with persistent excitation," *IEEE Trans. Aut. Control*, Vol. 32, No.2 pp. 134-145, 1987.
- [34] K. S. Narendra and A. M. Annaswamy, *Stable adaptive systems*, Prentice-Hall, Engelwood Cliffs, New Jersey, 1989.
- [35] 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.
- [36] H. Nijmeijer and A.J. van der Schaft, *Nonlinear Dynamical Control Systems*, Springer-Verlag, New York, 1990.
- [37] M. Niranjan and F. Fallside, "Neural networks and radial basis functions for classifying static speech patterns," *Computer Speech and Language*, Vol.4, pp.275-289, 1990.
- [38] J. Platt, "A resource allocating network for function interpolation", *Neural Computation*, Vol.4, No.2, pp.213-225, 1991.
- [39] T. Poggio and F. Girosi, "Networks for approximation and learning", *Proceedings of the IEEE*, Vol. 78, No. 9, 1990.
- [40] T. Poggio and F. Girosi, "Regularization algorithms for learning that are equivalent to multilayer networks", *Science*, Vol.247, pp.978-982, 1990.
- [41] 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.

- [42] M. J. D. Powell, *Approximation Theory and Methods*, Cambridge University Press, Cambridge, 1981.
- [43] M. J. D. Powell, "Radial basis functions for multivariable interpolation: A review", In J. C. Mason and M. G. Cox, (eds), *Algorithms for Approximation*, pp.143-167, Oxford University Press, Oxford, 1987.
- [44] D. Psaltis, A. Sideris and A.A. Yamamura, "A multilayered neural network controller," *IEEE Control Systems Magazine*, Vol. 8, pp. 17-21, 1988
- [45] 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.
- [46] 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.
- [47] 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.
- [48] S. Sastry and M. Bodson, *Adaptive Control: Stability, Convergence and Robustness*, Prentice-Hall, Englewood Cliffs, 1989.
- [49] J. J. E. Slotine and W. Li, *Applied Nonlinear Control*. Prentice-Hall International, Englewood Cliffs, New Jersey 1991.
- [50] 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.
- [51] R. J. Werbos, "Backpropagation through time: what it does and how to do it", In *Proceedings of the IEEE*, Vol.78, pp.1550-1560, 1990.
- [52] 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.

