

Dynamic recurrent neural network for system identification and control

A. Delgado
C. Kambhampati
K. Warwick

Indexing terms: Identification, Hopfield neural networks, Nonlinear modelling, Neural networks

Abstract: A dynamic recurrent neural network (DRNN) that can be viewed as a generalisation of the Hopfield neural network is proposed to identify and control a class of control affine systems. In this approach, the identified network is used in the context of the differential geometric control to synthesise a state feedback that cancels the nonlinear terms of the plant yielding a linear plant which can then be controlled using a standard PID controller.

1 Introduction

Recent advances in the understanding of the working principles of artificial neural networks (ANNs) has given a tremendous boost to the applications of these modelling tools for the control of nonlinear systems [1, 2]. Most of the current applications rely on the classical NARMA approach; here a feedforward network is used to synthesise the nonlinear map [3–5]. This approach, powerful in itself, has some disadvantages:

(a) the network has as its inputs a number of past system inputs and outputs, so to find out the optimum number of past values a trial and error process must be carried on

(b) the model is naturally formulated in discrete time with fixed sampling period, so that if the sampling period is changed the network must be trained again

(c) the problems associated with the stability of these networks are not clearly understood

(d) the problem of discrete-time nonlinear control is not fully understood and there is not a framework available for analysis.

Along with the developments in the area of neural networks, tremendous strides have been made in the area of nonlinear control analysis using differential geometric and algebraic techniques [6, 7]. The tools for analysis provided form a natural basis for the analysis of dynamic neural controllers; it is here that dynamic recurrent neural networks (DRNNs) come into their own, for these can be described by sets of differential equations and can be analysed within a differential geometric framework [8, 9].

The control strategies that can be evolved using differential geometric techniques involve the plant linearisation

using what is known as linearising state feedback. In this approach, the key principle is the cancellation of nonlinear terms yielding a linear plant which can then be controlled using standard PID controllers; this particular strategy was proposed by Kravaris and Chung [10] and it is known as globally linearising control (GLC).

In this work a nonlinear plant is identified with a DRNN. Then the network is used in two ways

(i) to calculate the linearising state feedback that cancels the nonlinear terms of the plant

(ii) to observe the plant and generate the state vector needed in the linearising feedback.

2 Mathematical preliminaries

This Section presents the concepts of relative degree, input–output linearisation and the GLC structure for continuous time, single-input/single-output (SISO), nonlinear-control affine systems.

Consider the nonlinear-control affine system

$$\left. \begin{aligned} \dot{x}_1 &= f_1(x_1, \dots, x_n) + g_1 u \\ \dot{x}_n &= f_n(x_1, \dots, x_n) + g_n u \\ y &= h(x_1, \dots, x_n) \end{aligned} \right\} \quad (1)$$

These equations can be written in compact form

$$\left. \begin{aligned} \dot{x} &= f(x) + gu \\ y &= h(x) \end{aligned} \right\} \quad (2)$$

where $x \in \mathcal{R}^n$, $u \in \mathcal{R}$, $y \in \mathcal{R}$, $f(x)$ and g are vector fields and $h(x)$ is a scalar field, i.e. $f(x) \in \mathcal{R}^n$, $g \in \mathcal{R}^n$ and $h(x) \in \mathcal{R}$.

2.1 Relative degree

The relative degree r for any SISO control system is defined as the number of times that the output $y(t)$ must be differentiated with respect to time in order to have the input $u(t)$ appearing explicitly.

If the system described by eqns. 2 has relative degree $r = 1$, the first derivative of the output $y(t)$ is given by

$$\dot{y} = L_f h(x) + L_g h(x)u$$

A. Delgado thanks the Colombian Institute for the Development of Science and Technology (COLCIENCIAS) and the National University of Colombia for supporting his PhD studies at the Cybernetics Department of the University of Reading.

© IEE, 1995

Paper 1873D (C2, C8), received 21st December 1994

The authors are with the Cybernetics Department, University of Reading, PO Box 225, Reading, RG6 6AY, United Kingdom

IEE Proc.-Control Theory Appl., Vol. 142, No. 4, July 1995

307

where $L_f h(x)$ and $L_g h(x)$ are known as the Lie derivatives of the scalar field $h(x)$ in the direction of the vector fields $f(x)$ and g , respectively, i.e.

$$L_f h(x) = \sum_{i=1}^n \frac{\partial h(x)}{\partial x_i} f_i(x)$$

$$L_g h(x) = \sum_{i=1}^n \frac{\partial h(x)}{\partial x_i} g_i; \quad L_g h(x) \neq 0$$

If $L_g h(x) = 0$ and the relative degree is $r = 2$, the second derivative of the output $y(t)$ is given by

$$\ddot{y} = L_f^2 h(x) + L_g L_f h(x)u$$

where

$$L_f^2 h(x) = \sum_{i=1}^n \frac{\partial \{L_f h(x)\}}{\partial x_i} f_i(x)$$

$$L_g L_f h(x) = \sum_{i=1}^n \frac{\partial \{L_f h(x)\}}{\partial x_i} g_i; \quad L_g L_f h(x) \neq 0$$

In general, the nonlinear system of eqn. 2 is said to have relative degree r if

$$L_g L_f^i h(x) = 0 \quad i = 0, \dots, r-2$$

$$L_g L_f^{r-1} h(x) \neq 0$$

with $L_f^0 h(x) = h(x)$. An important case occurs when the relative degree r is equal to the number of states ($r = n$). The output and its derivatives can be written as

$$\left. \begin{aligned} y &= h(x) \\ \dot{y} &= L_f h(x) \\ \ddot{y} &= L_f^2 h(x) \\ &\dots \\ y^{(n-1)} &= L_f^{n-1} h(x) \\ y^{(n)} &= L_f^n h(x) + L_g L_f^{n-1} h(x)u \end{aligned} \right\} \quad (3)$$

2.2 Input-output linearisation

The goal of the input-output linearisation technique is to cancel the nonlinear terms of the plant, using a state feedback, in order to have a linear dynamics between a new input $v(t)$ and the actual plant output $y(t)$.

If the input u for the system described by eqns. 3 is defined as

$$u = \frac{\{v - L_f^n h(x) - \beta_1 L_f^{n-1} h(x) - \dots - \beta_{n-1} L_f h(x) - \beta_n h(x)\}}{L_g L_f^{n-1} h(x)} \quad (4)$$

then the last equation of eqns. 3 becomes,

$$y^{(n)} = v - \beta_1 L_f^{n-1} h(x) - \dots - \beta_{n-1} L_f h(x) - \beta_n h(x)$$

i.e.

$$y^{(n)} + \beta_1 y^{(n-1)} + \dots + \beta_{n-2} \ddot{y} + \beta_{n-1} \dot{y} + \beta_n y = v \quad (5)$$

The nonlinear terms $L_g L_f^{n-1} h(x)$, $L_f^n h(x)$ have been cancelled by the state feedback (eqn. 4) and the resulting system is linear between the new input v and the output y . The parameters β_1, \dots, β_n are based on the desired input-output characteristic; this means that the $v - y$ system can have arbitrarily placed poles.

2.3 Globally linearising control

After linearising the system defined by eqns. 2 with the state feedback (eqn. 4), one can use an external PI loop

$$v = k_p \{r(t) - y(t)\} + k_i \int_0^{t_f} \{r(t) - y(t)\} dt \quad (6)$$

to force the output $y(t)$ to track a given desired trajectory $r(t)$. This control structure was proposed in Reference 10 and it is called the globally linearising control (GLC) structure; see Fig. 1.

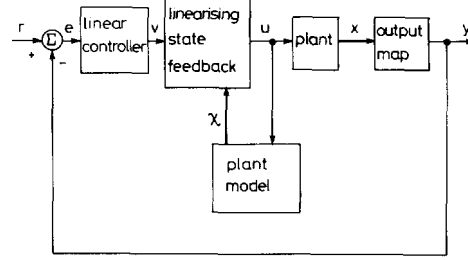


Fig. 1 Globally linearising control (GLC) structure

The design procedure using the GLC structure is simple: compute the linearising state feedback (eqn. 4) from the system model and then tune the PI controller (eqn. 6).

3 Dynamic recurrent neural network

ANNs consist of many interconnected simple nonlinear systems called neurons. Generally speaking, neuron models can be divided in two basic types, static and dynamic. A dynamic neuron is the one whose output is described by a differential equation (see Fig. 2). A DRNN

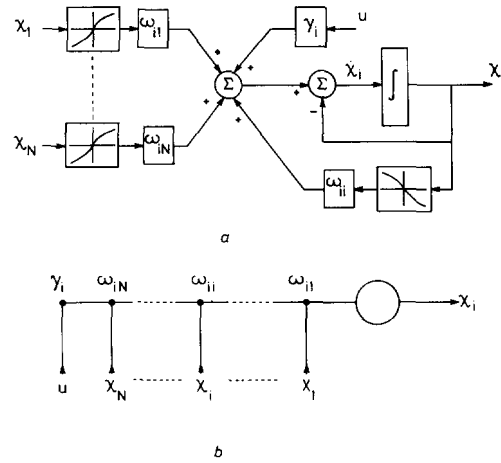


Fig. 2 Dynamic neuron model

a The neuron is depicted in detail; note the differential equation for the output x_i ,

$$\dot{x}_i = -x_i + \sum_{j=1}^N \omega_{ij} \sigma(x_j) + \gamma_i u$$

b This simplified picture is used as the building block for a dynamic network

is a network of dynamic neurons with forward and backward connections (see Fig. 3).

In this section it is demonstrated that a DRNN can approximate control affine systems of the class (2).

Theorem 1: The dynamic recurrent neural network

$$\dot{\chi} = -T\chi + W\sigma(\chi) + \Gamma u \quad (7)$$

where $\chi \in \mathfrak{R}^N$, $T \in \mathfrak{R}^{N \times N}$, $W \in \mathfrak{R}^{N \times N}$, $\Gamma \in \mathfrak{R}^{N \times 1}$ and $\sigma(\chi) = \{\sigma(\chi_1), \dots, \sigma(\chi_N)\}$ can approximate the nonlinear

system (eqn. 2)

$$\begin{aligned}\dot{x} &= f(x) + gu \\ y &= h(x)\end{aligned}$$

The number of units $N \geq n$.

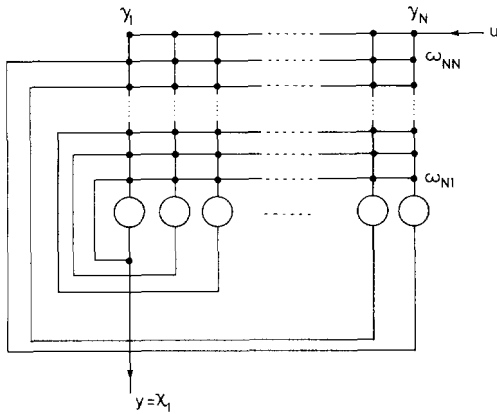


Fig. 3 Dynamic recurrent neural network (DRNN)

Proof: The nonlinear system (eqns. 2) can be written as

$$\begin{aligned}\dot{x} &= -Ax + f_0(x) + gu \\ y &= h(x)\end{aligned}\quad (8)$$

where

$$f_0(x) = Ax + f(x) \quad A \in \mathfrak{R}^{n \times n}$$

It is now well known that feedforward neural networks with one hidden layer can approximate any nonlinear function with arbitrary degree of accuracy [11–13]; in Reference 14 it is stated that a feedforward neural network can be used to approximate the mapping $f_0(x)$

$$f_0(x) = C\sigma(Dx + \theta) \quad (9)$$

where $C \in \mathfrak{R}^{n \times N_1}$, $D \in \mathfrak{R}^{N_1 \times n}$, $\theta \in \mathfrak{R}^{N_1}$ and $\sigma(x) = \{\sigma(x_1), \dots, \sigma(x_{N_1})\}$. N_1 is the number of hidden neurons in the feedforward network.

Substituting eqn. 9 in eqn. 8

$$\begin{aligned}\dot{x} &= -Ax + C\sigma(Dx + \theta) + gu \\ y &= h(x)\end{aligned}\quad (10)$$

Define the new variable ξ and its derivative

$$\xi = Dx + \theta \quad (11)$$

$$\dot{\xi} = D\dot{x} \quad (12)$$

Substituting eqn. 11 into eqn. 10 and eqn. 10 into eqn. 12, the following dynamic recurrent network is obtained

$$\begin{aligned}\dot{x} &= -Ax + C\sigma(\xi) + gu \\ \dot{\xi} &= -D Ax + DC\sigma(\xi) + Dgu \\ y &= h(x)\end{aligned}$$

In matrix form,

$$\begin{aligned}\begin{bmatrix} \dot{x} \\ \dot{\xi} \end{bmatrix} &= -\begin{bmatrix} A & 0 \\ DA & 0 \end{bmatrix} \begin{bmatrix} x \\ \xi \end{bmatrix} + \begin{bmatrix} 0 & C \\ 0 & DC \end{bmatrix} \begin{bmatrix} \sigma(x) \\ \sigma(\xi) \end{bmatrix} + \begin{bmatrix} g \\ Dg \end{bmatrix} u \\ y &= h(x)\end{aligned}$$

This can be written as

$$\begin{aligned}\dot{\chi} &= -T\chi + W\sigma(\chi) + \Gamma u \\ y &= h(x)\end{aligned}\quad (13)$$

where

$$\chi = \begin{bmatrix} x \\ \xi \end{bmatrix} \quad T = -\begin{bmatrix} A & 0 \\ DA & 0 \end{bmatrix} \quad W = \begin{bmatrix} 0 & C \\ 0 & DC \end{bmatrix} \quad \Gamma = \begin{bmatrix} g \\ Dg \end{bmatrix}\quad (14)$$

$\chi \in \mathfrak{R}^N$, $T \in \mathfrak{R}^{N \times N}$, $W \in \mathfrak{R}^{N \times N}$, $\Gamma \in \mathfrak{R}^{N \times 1}$ and $\sigma(\chi) = \{\sigma(x_1), \dots, \sigma(x_N)\}$. The total number of neurons is $N = N_1 + n$.

The network (eqn. 13) with $T = I$ and no restrictions on the weight matrix W is known as the generalised Hopfield network and was first proposed by Hopfield [15] in the context of associative memories. This network is described by

$$\begin{aligned}\dot{\chi} &= -\chi + W\sigma(\chi) + \Gamma u \\ y &= h(\chi)\end{aligned}\quad (15)$$

where $\chi \in \mathfrak{R}^N$, $T \in \mathfrak{R}^{N \times N}$, $W \in \mathfrak{R}^{N \times N}$, $\Gamma \in \mathfrak{R}^{N \times 1}$ and $\sigma(\chi) = \{\sigma(x_1), \dots, \sigma(x_N)\}$.

The computer simulations have shown that the model of eqn. 15 is more general than eqn. 13 and also the learning is faster. In this paper the identification and control of nonlinear systems is carried out using the model of eqn. 15.

4 Relative degree of the DRNN

In Section 2 it was shown that the relative degree is a key concept in order to linearise systems of the class (eqn. 2). In the following, it is demonstrated that a DRNN has the capacity to identify not only the dynamics of systems (eqn. 2) but also their relative degree.

There are several patterns to obtain any relative degree with the dynamic neural network (eqn. 15); in this paper one of them is shown [9] when $\gamma_i = 0, \forall i \neq r$.

Theorem 2: The recurrent neural network

$$\begin{aligned}\dot{\chi}_i &= \hat{f}_i(\chi) + \hat{g}_i u \\ y &= \hat{h}(\chi)\end{aligned}\quad (16)$$

where

$$\begin{aligned}\hat{f}_i(\chi) &= -\chi_i + \sum_{j=1}^N \omega_{ij} \sigma(\chi_j) \\ \hat{g}_i(\chi) &= \gamma_i, \quad i = 1, \dots, N \\ \hat{h}(\chi) &= \chi_1\end{aligned}\quad (17)$$

can have any desired relative degree $r \leq N$ as follows:

- (i) $r = 1$, if $\gamma_1 \neq 0$
- (ii) $r = 2$, if $\gamma_i = 0, \forall i \neq 2$ and $\omega_{12} \neq 0$
- (iii) $r > 2$, if $\gamma_i = 0, \forall i \neq r$ and $\omega_{ij} = 0, i = 1, \dots, r-2; j = i+2, \dots, N$

The case when the relative degree is infinite is not considered. This situation occurs when the input does not affect the output, i.e. the system is autonomous.

Proof: The proof of this theorem is illustrated by way of an example. Consider the following four-state ($i = 1, 2, 3, 4$) neural network

$$\begin{aligned}\dot{\chi}_i &= -\chi_i + \sum_{j=1}^4 \omega_{ij} \sigma(\chi_j) + \gamma_i u \\ y &= \hat{h}(\chi) = \chi_1\end{aligned}$$

In this model

$$\hat{f}_i(\chi) = -\chi_i + \sum_{j=1}^4 \omega_{ij} \sigma(\chi_j)$$

$$\hat{g}_i(\chi) = \gamma_i$$

Recall the mathematical definition of relative degree (Section 2.1):

(i) Condition: $\gamma_1 \neq 0, \gamma_2 = 0, \gamma_3 = 0, \gamma_4 = 0$

$$L_{\hat{g}} \hat{h}(\chi) = \gamma_1$$

$$L_{\hat{g}} \hat{h}(\chi) \neq 0. \text{ Then } r = 1.$$

(ii) Condition: $\gamma_1 = 0, \gamma_2 \neq 0, \gamma_3 = 0, \gamma_4 = 0$

$$L_{\hat{g}} \hat{h}(\chi) = 0$$

$$L_f \hat{h}(\chi) = \hat{f}_1(\chi)$$

$$L_{\hat{g}} L_f \hat{h}(\chi) = \frac{\partial \hat{f}_1(\chi)}{\partial \chi_2} \gamma_2$$

$$L_{\hat{g}} L_f \hat{h}(\chi) = \omega_{12} \sigma'(\chi_2) \gamma_2$$

where

$$\sigma'(\chi_i) = \frac{d\sigma(\chi_i)}{d\chi_i}$$

$$L_{\hat{g}} L_f \hat{h}(\chi) \neq 0 \text{ if } \omega_{12} \neq 0. \text{ Then } r = 2.$$

(iii) Condition: $\gamma_1 = 0, \gamma_2 = 0, \gamma_3 \neq 0, \gamma_4 = 0$:

$$L_{\hat{g}} L_f \hat{h}(\chi) = \omega_{13} \sigma'(\chi_3) \gamma_3$$

$$L_{\hat{g}} L_f \hat{h}(\chi) = 0 \text{ if } \omega_{13} = 0$$

$$L_{\hat{g}}^2 \hat{h}(\chi) = \{-1 + \omega_{11} \sigma'(\chi_1)\} \hat{f}_1(\chi) + \omega_{12} \sigma'(\chi_2) \hat{f}_2(\chi) + \omega_{14} \sigma'(\chi_4) \hat{f}_4(\chi)$$

$$L_{\hat{g}} L_{\hat{g}}^2 \hat{h}(\chi) = \omega_{12} \sigma'(\chi_2) \omega_{23} \sigma'(\chi_3) \gamma_3 + \omega_{14} \sigma'(\chi_4) \omega_{43} \sigma'(\chi_3) \gamma_3$$

$$L_{\hat{g}} L_{\hat{g}}^2 \hat{h}(\chi) \neq 0. \text{ Then } r = 3.$$

(iv) Condition: $\gamma_1 = 0, \gamma_2 = 0, \gamma_3 = 0, \gamma_4 \neq 0, \omega_{13} = 0$

$$L_{\hat{g}} L_{\hat{g}}^2 \hat{h}(\chi) = \{-1 + \omega_{11} \sigma'(\chi_1)\} \omega_{14} \sigma'(\chi_4) \gamma_4 + \omega_{12} \sigma'(\chi_2) \omega_{24} \sigma'(\chi_4) \gamma_4 + \omega_{14} \sigma'(\chi_4) \omega_{44} \sigma'(\chi_4) \gamma_4$$

$$L_{\hat{g}} L_{\hat{g}}^2 \hat{h}(\chi) = 0 \text{ if } \omega_{14} = 0 \quad \omega_{24} = 0$$

$$L_{\hat{g}}^3 \hat{h}(\chi) = \omega_{11} \sigma''(\chi_1) \hat{f}_1^2(\chi) + \{-1 + \omega_{11} \sigma'(\chi_1)\}^2 \hat{f}_1(\chi) + \{-1 + \omega_{11} \sigma'(\chi_1)\} \omega_{12} \sigma'(\chi_2) \hat{f}_2(\chi) + \omega_{12} \sigma'(\chi_2) \omega_{21} \sigma'(\chi_1) \hat{f}_1(\chi) + \omega_{12} \sigma'(\chi_2) \omega_{22} \sigma'(\chi_2) \hat{f}_2(\chi) + \omega_{12} \sigma'(\chi_2) \omega_{23} \sigma'(\chi_3) \hat{f}_3(\chi)$$

$$L_{\hat{g}} L_{\hat{g}}^3 \hat{h}(\chi) = \omega_{12} \sigma'(\chi_2) \omega_{23} \sigma'(\chi_3) \omega_{34} \sigma'(\chi_4)$$

$$L_{\hat{g}} L_{\hat{g}}^3 \hat{h}(\chi) \neq 0. \text{ Then } r = 4.$$

Finally, it is concluded that:

$$r = 1 \text{ for } \gamma_1 \neq 0, \gamma_2 = 0, \gamma_3 = 0, \gamma_4 = 0$$

$$r = 2 \text{ for } \gamma_1 = 0, \gamma_2 \neq 0, \gamma_3 = 0, \gamma_4 = 0 \text{ and } \omega_{12} \neq 0$$

$$r = 3 \text{ for } \gamma_1 = 0, \gamma_2 = 0, \gamma_3 \neq 0, \gamma_4 = 0 \text{ and } \omega_{13} = 0$$

$$r = 4 \text{ for } \gamma_1 = 0, \gamma_2 = 0, \gamma_3 = 0, \gamma_4 \neq 0$$

and $\omega_{13} = 0, \omega_{14} = 0, \omega_{24} = 0$.

5 Stability of the DRNN

After the dynamic network has been trained, the model is used in two ways: (a) to calculate the linearising state

feedback (eqn. 4) that cancels the nonlinear terms of the plant; and (b) to observe in open loop the plant and generate the state vector χ needed in the linearising feedback. Then it is important to determine if the network as an open loop observer is stable, so the linearising feedback will work properly.

The stability of the DNRR can be examined with the first method of Liapunov; in this method each equilibrium state is investigated separately.

From eqn. 15, the autonomous dynamic network is,

$$\dot{\chi} = -\chi + W\sigma(\chi) \quad (18)$$

where $\chi = [\chi_1, \dots, \chi_N]^T$, $W \in \mathfrak{R}^{N \times N}$, $\sigma(\chi) = [\sigma(\chi_1), \dots, \sigma(\chi_N)]^T$. One equilibrium point χ^* is the solution of

$$\chi^* = W\sigma(\chi^*)$$

Expanding the right-hand side of eqn. 18 in Taylor series about the equilibrium state χ^* and deleting the higher-order derivatives, we obtain

$$\dot{z} = [-I + H]z \quad (19)$$

with

$$z = \chi - \chi^*$$

$$I = \begin{bmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{bmatrix}$$

$$H = WG$$

$$G = \text{diag} [g_1, \dots, g_N]$$

$$G = \begin{bmatrix} g_1 & & 0 \\ & \ddots & \\ 0 & & g_N \end{bmatrix}$$

$$g_i = \frac{d\sigma(\chi_i)}{d\chi_i} |_{\chi_i = \chi_i^*}, \text{ then } g_i \geq 0, \forall i$$

We can write eqn. 19 as

$$\dot{z} = Az \quad (20)$$

where

$$A = -I + H$$

$$A = \begin{bmatrix} h_{11} - 1 & \dots & h_{1N} \\ \vdots & h_{ii} - 1 & \vdots \\ h_{N1} & \dots & h_{NN} - 1 \end{bmatrix} \quad (21)$$

$$h_{ij} = \omega_{ij} g_j$$

$$i = 1, \dots, N \quad j = 1, \dots, N$$

Lyapunov showed that, if all the eigenvalues of the matrix A in eqn. 20 have nonzero real parts, then the stability of the equilibrium point χ^* of the original system (eqn. 18), is the same as that of the equilibrium point $z = 0$ of the linearised equation (eqn. 20). It is possible to establish bounds for the real part of the eigenvalues of the matrix (eqn. 21) using the matrix measure [16].

5.1 Matrix measure

A normed linear space is an ordered pair $(X, \|\cdot\|)$ where X is a linear vector space and $\|\cdot\|: X \rightarrow \mathfrak{R}$ is a real valued function called the norm. The concept of norm can be extended to include matrices

$$\|A\|_i = \sup_{x \neq 0} \frac{\|Ax\|}{\|x\|}$$

this is called the induced matrix norm of A corresponding to the vector norm $\|\cdot\|$.

The matrix measure $\mu(A)$ of a matrix A is the directional derivative of the induced norm $\|\cdot\|_1$ at the point I (identity matrix), in the direction of A . That is

$$\mu(A) = \lim_{\varepsilon \rightarrow 0^+} \frac{\|I + \varepsilon A\|_1 - 1}{\varepsilon}$$

There is one interesting property of the matrix measure [16]; if λ_i is an eigenvalue of A , then

$$-\mu(-A) \leq \operatorname{Re} \lambda_i \leq \mu(A)$$

for asymptotic stability $\mu(A) < 0$ and $\mu(-A) > 0$.

In this section two matrix measures corresponding to the l^1, l^∞ norms in \mathbb{R}^n are used to find the bounds for the real part of the eigenvalues for the linearised neural network,

$$\mu_1(A) = \max_j \left[a_{jj} + \sum_{i \neq j} |a_{ij}| \right]$$

$$\mu_\infty(A) = \max_i \left[a_{ii} + \sum_{j \neq i} |a_{ij}| \right]$$

5.1.1 Stability from μ_1 : Calculating μ_1 for eqn. 21

$$\mu_1(A) = \max_j \left[h_{jj} - 1 + \sum_{i \neq j} |h_{ij}| \right] \quad (22)$$

$$\mu_1(-A) = \max_j \left[1 - h_{jj} + \sum_{i \neq j} |h_{ij}| \right] \quad (23)$$

For asymptotic stability eqn. 22 must be negative

$$\max_j \left[h_{jj} - 1 + \sum_{i \neq j} |h_{ij}| \right] < 0$$

This is equivalent to

$$h_{jj} - 1 + \sum_{i \neq j} |h_{ij}| < 0 \quad \forall j$$

and finally

$$-(h_{jj} - 1) > \sum_{i \neq j} |h_{ij}| \quad \forall j \quad (24)$$

note that if eqn. 24 is true then eqn. 23 is automatically satisfied.

5.1.2 Stability from μ_∞ : The same procedure is repeated for the matrix measure μ_∞ . The condition for asymptotic stability is

$$-(h_{ii} - 1) > \sum_{j \neq i} |h_{ij}| \quad \forall i \quad (25)$$

The conditions defined by eqn. 24 and eqn. 25 are sufficient and agree with those derived in Reference 17 using an eigenvalue localisation theorem known as the Gerschgorin's circle theorem. It is important to mention that the matrix measure states bounds and in particular the upper bound for the real part of the eigenvalues can be positive; this does not imply that the actual real part of the eigenvalues is positive. In other words the conditions of eqns. 24 and 25 are sufficient but not necessary.

6 Identification

It was shown in theorem 1 that the DRNN (eqn. 16) can approximate control affine systems of the class (eqn. 2). To illustrate this result, a single-link manipulator is identified with a DRNN using random noise as the input. After

training, the network is tested for stability and for generalisation with other inputs.

6.1 The plant

The nonlinear plant is a single-link manipulator, shown schematically in Fig. 4. This plant is described by the

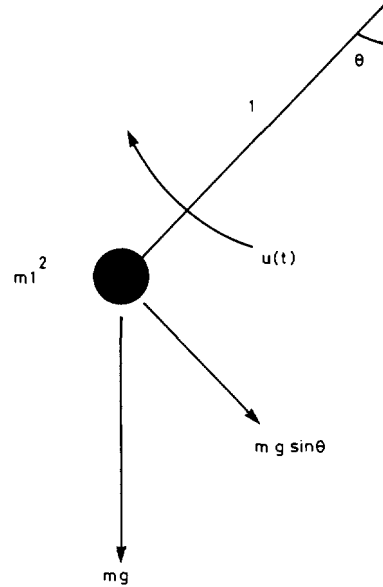


Fig. 4 Single-link manipulator

second-order nonlinear differential equation

$$ml^2 \ddot{\theta}(t) + v \dot{\theta}(t) + mgl \sin \theta(t) = u(t)$$

where the length, mass and friction coefficients are $l = 1$ m, $m = 2.0$ kg and $v = 1.0$ kg m²/s, respectively [18].

The corresponding state representation is

$$\left. \begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -9.8 \sin x_1 - 0.5x_2 + 0.5u \\ y &= x_1 \end{aligned} \right\} \quad (26)$$

with $x_1(0) = x_2(0) = 0$ and $y = \theta(t)$.

The plant (eqn. 26) was identified with the neural network (eqn. 16 and 17); the identification scheme assumes that the plant is a black box and the only available information is the input-output data. Fig. 5 shows

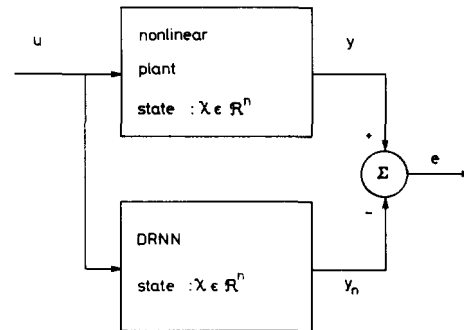


Fig. 5 Plant identification using the DRNN

the nonlinear plant and the dynamic neural network during the identification process.

6.2 Training

The DRNN (eqns. 16 and 17) was trained with $N = 5$ neurons and $\sigma(x) = \tanh(x)$ to identify the nonlinear plant (eqn. 26); the training was carried out repetitively over the fixed time interval $[0, t_f]$ with the chemotaxis algorithm [19] to minimise the performance index (eqn. 27). The training input was random noise and the initial conditions for the neuron states were selected at random, $t_f = 20$ s.

$$J = \left\{ \frac{1}{t_f} \int_0^{t_f} e^{2(t)} dt \right\}^{1/2} = \left[\frac{1}{t_f} \int_0^{t_f} \{y(t) - y_n(t)\}^2 dt \right]^{1/2} \quad (27)$$

Fig. 6 shows the output of the plant and the output of the neural network for the training input; the performance index after the training was $J = 0.0064$.

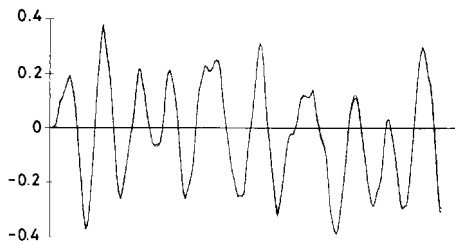


Fig. 6 Plant output and network output for the training input

The parameters of the network after the training were as listed in Table 1.

Table 1: Network parameters after training

Weight matrix W				
0.4684	-2.4995	0.4211	-0.2848	0.1995
1.3615	0.0642	0.0413	-1.8925	-1.6608
-0.8185	-0.9241	-0.0743	-0.1264	0.1484
-0.3257	1.2319	-1.0997	0.2192	-0.8547
-1.2444	0.4396	-0.5466	1.7342	-0.5953
Weight vector Γ		Initial conditions		
-0.0050	-0.0097			
-0.2111	-0.0065			
0.1689	0.0171			
0.0645	-0.0097			
-0.0413	0.0025			

The first component of the weight vector Γ is $\gamma_1 \cong 0$; then according to the relative-degree theorem, the trained network has $r = 2$. It can be seen that this is the same relative degree as that of the plant, so the network is capable to identify the dynamics of the plant and its relative degree.

To verify the generalisation of the neural-network model, different inputs were used. One such input, is a sine wave

$$u(t) = \frac{\pi}{2} \sin\left(\frac{2\pi t}{2.5}\right) + \frac{\pi}{2} \sin\left(\frac{2\pi t}{5.0}\right) \quad (28)$$

Fig. 7 shows the output of the plant and the output of the network for the input (eqn. 28); the performance index was $J = 0.0069$.

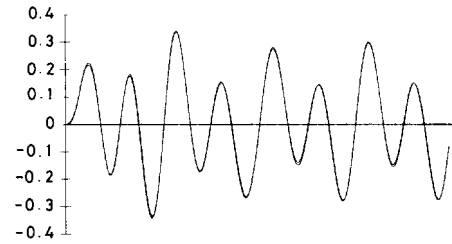


Fig. 7 Plant output and network output for the input of eqn. 28

The stability of the trained network was tested using the first method of Liapunov and the matrix measure μ_∞ . The real part of the eigenvalues of the linearised network lies in the interval

$$-5.8919 \leq \text{Re } \lambda_i \leq 4.0203$$

and the exact eigenvalues are

$$-0.2260 + j3.1483$$

$$-0.2260 - j3.1483$$

$$-2.0288 + j0.1798$$

$$-2.0288 - j0.1798$$

$$-0.4082$$

It is concluded that the trained network has good generalisation capabilities and it is stable, so it can be used as an open-loop observer.

7 Control

In this section the DRNN is used to cancel the nonlinear terms of the plant and to produce a desired linear dynamics between the new input v and the plant output y . Then the linearised plant is immersed in the GLC structure and a PI controller is adjusted to match a closed-loop desired response.

The desired dynamics for the linearised plant was chosen as

$$\frac{y}{v} = \frac{1}{(s^2 + 5s + 6)}$$

This is equivalent to the differential equation

$$\frac{d^2 y}{dt^2} + 5 \frac{dy}{dt} + 6y = v \quad (29)$$

The trained neural network was used to calculate the linearising state feedback

$$u = \frac{v - L_f^2 \tilde{h}(x) - \beta_1 L_f \tilde{h}(x) - \beta_2 \tilde{h}(x)}{L_g L_f \tilde{h}(x)} \quad (30)$$

where $\beta_1 = 5$ and $\beta_2 = 6$. Fig. 8 illustrates the linearisation of the plant (eqn. 26) with the input (eqn. 30), and Fig. 9 compares the desired open-loop dynamics (eqn. 29) and the linearised plant dynamics for a random noise input v ; the performance index was $J = 0.0187$.

Now the linearised plant is introduced into the GLC structure; the aim here is to ensure that the overall

closed-loop system matches some desired global dynamics. In this instance, a well damped second-order dynamics of the form

$$\frac{y_d}{r} = \frac{1}{(s+1)^2} \quad (31)$$

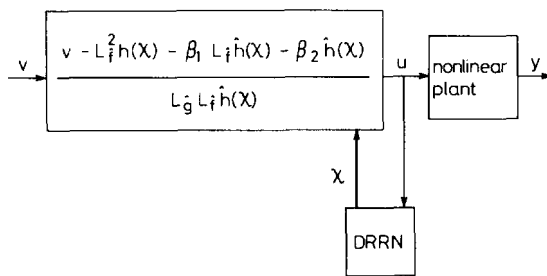


Fig. 8 Input-output linearisation using the DRNN

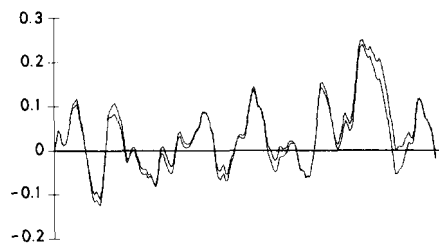


Fig. 9 Desired open-loop dynamics and actual input-output plant dynamics for a random input v

was selected. The PI controller is

$$v = \left[k_p + \frac{k_i}{s} \right] e$$

where $k_p = 1.0$ and k_i is adjusted.

Fig. 10 shows the desired closed-loop dynamics (eqn. 31) and the actual dynamics of the GLC structure for the

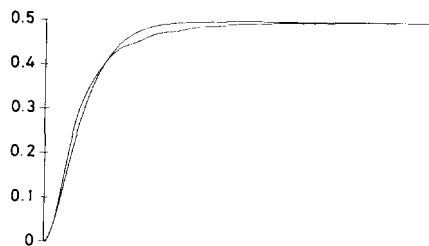


Fig. 10 Desired closed-loop response and actual GLC response for a step reference $r = 0.5$

reference input $r = 0.5$, the performance index (eqn. 27) [replacing $y_a(t)$ by $y_d(t)$] was $J = 0.0121$ with $k_i = 3.8$. Fig. 11 shows the desired output and the plant output for the sinusoidal input (eqn. 32); the performance index was $J = 0.0216$.

$$u(t) = \frac{\pi}{2} \sin\left(\frac{2\pi t}{2.5}\right) \quad (32)$$

Fig. 12 shows the response for the step $r = 0.5$, when the mass of the single link manipulator was changed from 2.0 kg to 2.5 kg; the performance index was $J = 0.0326$. Fig. 13 presents the response for the step $r = 0.5$, when a

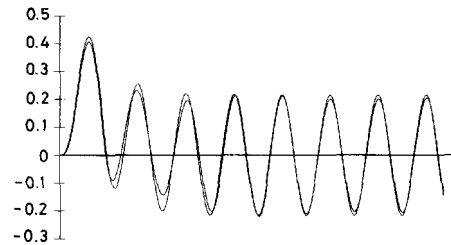


Fig. 11 Desired closed-loop response and actual GLC response for the input of eqn. 32

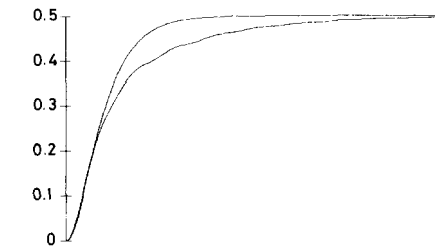


Fig. 12 Desired closed-loop response and GLC response for a step reference $r = 0.5$ when the mass of the plant is changed from 2.0 kg to 2.5 kg

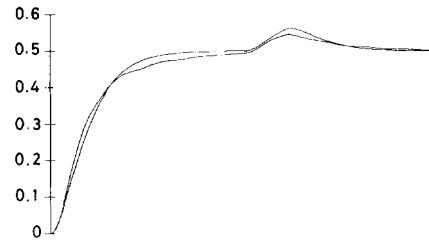


Fig. 13 Desired closed-loop response and GLC response for a step reference $r = 0.5$ when a disturbance of magnitude 0.1 is applied during the interval $[10.0, 12.0]$ s

disturbance of magnitude 0.1 was applied to the reference during the interval $\Delta t = [10.0, 12.0]$ s; the performance index was $J = 0.0132$.

8 Conclusions

It has been shown that the dynamic neural network (eqn. 15) can identify the relative degree and the dynamics of nonlinear systems of the class (eqn. 2).

After training, the neural network can be used to synthesise a linearising state feedback that cancels the nonlinear terms of the plant and yields any desired linear dynamics $v - y$. The identified network is used in two ways:

- to calculate the Lie derivatives of the linearising feedback
- to generate, on-line, the state vector x needed in the Lie derivatives.

To cancel the nonlinear terms properly, the trained network must be stable. It was shown that, owing to the

particular structure of the proposed DRNN, the stability can be tested easily using the first method of Liapunov and a matrix measure.

Finally, the simulation results showed that the performance of the GLC structure is satisfactory and robust. For example, the network was trained with a payload of 2.0 kg and the system was tested with a payload of 2.5 kg; it can be seen that the resulting response does not vary significantly. At the same time, it can be seen that inputs in the form of unmeasurable disturbances do not effect the performance. In fact with disturbance the index value is $J = 0.0132$ while without the disturbance it is $J = 0.0121$.

In general, if the DRNN is trained properly and it is stable with good generalisation capabilities, it can be used as an open-loop observer for systems of the class defined by eqn. 2.

9 References

- 1 MILLER III, W.T., SUTTON, R.S., and WERBOS, P.J.: 'Neural networks for control' (MIT Press, 1992)
- 2 HUNT, K.J., SBARBARO, D., ZBIKOWSKI, R., and GAWTHROP, P.J.: 'Neural networks for control systems — a survey', *Automatica*, 1992, **28**, pp. 1083–1112
- 3 CHEN, S., and BILLINGS, S.A.: 'Neural networks for nonlinear dynamic system modelling and identification', *Int. J. Control*, 1992, **56**, pp. 319–346
- 4 NARENDRA, K.S., and PARTHASARATHY, K.: 'Identification and control of dynamical systems using neural networks', *IEEE Trans.*, 1990, NN-1, pp. 4–26
- 5 PAO, Y., PHILLIPS, S.M., and SOBAJIC, D.J.: 'Neural net computing and the intelligent control of systems', *Int. J. Control*, 1992, **56**, pp. 263–289
- 6 ISIDORI, A.: 'Nonlinear control systems' (Springer-Verlag, 1989)
- 7 KRAVARIS, C., and KANTOR, J.C.: 'Geometric methods for nonlinear process control: 1. Background', *Ind. Eng. Chem. Res.*, 1990, **29**, pp. 2295–2310
- 8 KAMBHAMPATI, C., MANCHANDA, S., DELGADO, A., GREEN, G.G.R., WARWICK, K., and THAM, M.T.: 'On the relative order inverses and the architecture of a class of recurrent networks', *Automatica* (to be published)
- 9 DELGADO, A.: 'Differential geometric control of nonlinear systems with recurrent neural networks'. Internal research report of the Cybernetics Department, Reading University, 1994
- 10 KRAVARIS, C., and CHUNG, C.B.: 'Nonlinear state feedback synthesis by global input/output linearization', *AIChE J.*, 1987, **33**, pp. 592–603
- 11 CYBENKO, G.: 'Approximation by superpositions of a sigmoidal function'. Technical report, University of Illinois Urbana-Champaign Department of Electrical and Computer Engineering, 1988
- 12 HORNIK, K.: 'Multilayer feedforward networks are universal approximators', *Neural Networks*, 1989, **2**, pp. 359–366
- 13 FUNAHASHI, K.I.: 'On the approximate realization of continuous mappings by neural networks', *Neural Networks*, 1989, **2**, pp. 183–192
- 14 FUNAHASHI, K.I.: 'Approximation of dynamical systems by continuous time recurrent neural networks', *Neural Networks*, 1993, **6**, pp. 801–806
- 15 HOPFIELD, J.J.: 'Neurons with graded response have collective computational properties like those of two state neurons', *Proc. Natl. Acad. Sci. USA*, 1984, **81**, pp. 3088–3092
- 16 VIDYASAGAR, M.: 'Nonlinear systems analysis' (Prentice-Hall, New Jersey, 1993)
- 17 PERFETTI, R.: 'Asymptotic stability of equilibrium points in dynamical neural networks', *IEE Proc. D*, 1993, **140**, pp. 401–405
- 18 JIN, L., NIKIFORUK, P.N., and GUPTA, M.: 'Direct adaptive output tracking control using multilayered neural networks', *IEE Proc. D*, 1993, **140**, pp. 393–398
- 19 BREMERMAN, H.J., and ANDERSON, R.W.: 'An alternative to back propagation: a simple rule of synaptic modification for neural net training and memory'. Report PAM-483, Center for Pure and Applied Mathematics, University of California, Berkeley, 1990