Mathematics and Computers in Simulation 51 (2000) 181–199
www.elsevier.nl/locate/matcom

Inverse model control using recurrent networks

C. Kambhampati *, R.J. Craddock, M. Tham, K. Warwick
Department of Cybernetics, The University of Reading, Whiteknights, Reading, UK

Abstract

This paper illustrates how internal model control of nonlinear processes can be achieved by recurrent neural networks, e.g. fully connected Hopfield networks. It is shown that using results developed by Kambhampati et al. (1995), that once a recurrent network model of a nonlinear system has been produced, a controller can be produced which consists of the network comprising the inverse of the model and a filter. Thus, the network providing control for the nonlinear system does not require any training after it has been trained to model the nonlinear system. Stability and other issues of importance for nonlinear control systems are also discussed. ©2000 IMACS/Elsevier Science B.V. All rights reserved.

Keywords: Relative order; Left-inverses; Neural networks; Inverse model control

1. Introduction

For the control of nonlinear systems, many different control strategies can be applied [1–6], of these, the Internal Model Control (IMC) strategy (Fig. 1) [4] is considered in this paper. The motivating factor for this is that if a neural network can model the input–output relationship of the system, it should also be able to model the inverse of this relationship.

Certain Artificial Neural Networks (ANNs), e.g. Multi-Layer Perceptrons (MLPs), Radial Basis Functions (RBFs) and Recurrent Neural Networks (RNNs), have been shown to have the ability to model any nonlinear mappings to a desired degree of accuracy [7–9]. Due to this property, such networks are suitable for the identification and control of nonlinear plants [10]. The main class of networks used for nonlinear control is that of feedforward neural networks, in particular the MLP network. Recently, Lightbody and Irwin [11] have shown how MLPs can be used to provide controllers for use in an Inverse Model Control strategy. Initially an MLP is trained to model the plant. The network obtained can then be linearised to form a linearised model of the plant, either by further training or by analytic differentiation of the neural model. Using this linearised internal model, an inverse controller can be produced, via Kalman's method.

* Corresponding author.
E-mail addresses: cybck@cyber.reading.ac.uk (C. Kambhampati), cybrjc@cyber.reading.ac.uk (R.J. Craddock)

PII: S0378-4754(99)00116-0

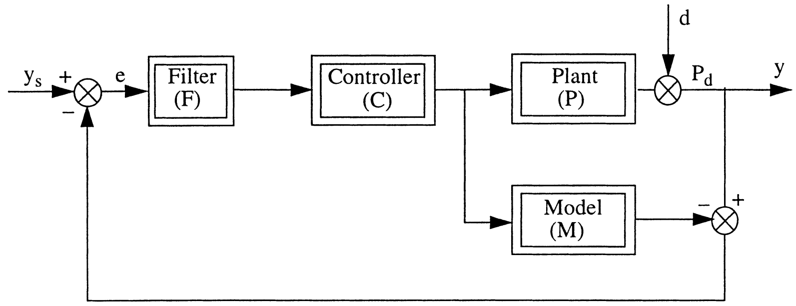

Fig. 1. Schematic diagram of internal model control.

The neural network utilised, acts as a medium for estimating the linear model parameters, but does not itself act as a controller.

An alternative to the standard feedforward neural network is the recurrent neural network which was first introduced by Hopfield [12]. Recurrent neural networks are feedforward ANNs with feedback connections. The introduction of feedback enables the description of temporal behaviour and hence the capacity to directly account for the dynamics of nonlinear systems. Temporal characteristics depend upon both the network inputs and the state of the neurons in the hidden layer of the networks. Narendra and Parthasarathy [13] (see also [38]) proposed a type of recurrent network topology for the purpose of identification and control of nonlinear systems. In their configuration, a feedforward ANN is combined with dynamical elements in the form of stable filters, to construct a recurrent network. The feasibility of applying this type of architecture to various control problems was demonstrated via simulation. Polycarpou and Ioannou [14] have further shown that this recurrent network configuration is capable of approximating a large class of dynamical systems, and Funahashi and Nakamura [15] argued that any finite time trajectory of a given autonomous dynamical system can be approximately realised by the internal state of the output units of a continuous time recurrent network.

The ability of recurrent neural networks to model dynamic nonlinear mappings/functions makes them attractive for use in nonlinear system control strategies. For use in the IMC strategy, certain recurrent networks, e.g. the Hopfield network, the RN1 network and the RN2 network have additional characteristics which make them suitable for use in IMC. In a recent paper, Kambhampati et al. [16,37] showed that a given recurrent network can have a finite range of values for its relative order, the exact value being dependent on the topology of the recurrent network. They also established, using the Hirschorn Inversion Theorem, that the left inverse of a recurrent network, is the same network, but requires a different input [17]. This is in marked contrast to other procedures for establishing the inverse of the plant [11,18,19], where a separate training scheme was used to establish the inverse. Although direct inversion schemes which often lead to minimal realisations exist, using such schemes it is difficult to obtain an inverse which is the original net itself [20]. This property of recurrent networks means that if such a network is used within the IMC strategy to model a plant, the inverse controller network is the same as the model network. Since the input–output dynamics of the model and the controller are the same, they share the same stability properties, thus it is sufficient to show stability of the model only, when determining stability of the model and of the controller.

This paper demonstrates the use of recurrent neural networks for the identification and control (through the IMC strategy) of two nonlinear systems. The systems considered are a chemical stirred tank reactor and a single link manipulator. In Section 2, the mathematical preliminaries required for analysis of the recurrent neural networks and production of the inverse network are discussed briefly; Section 3 discusses the IMC strategy; in Section 4, the use of Hopfield neural networks within the IMC strategy is discussed; Section 5 demonstrates the use of recurrent neural networks in IMC for the identification and control of nonlinear systems; and Section 6 concludes the paper by discussing the results.

2. Preliminaries

A brief description of the mathematical tools that will be utilised in subsequent analysis is now presented. Before proceeding any further, however, it is important that some definitions are made. (See Appendix A for more details of Lie derivatives.)

Consider the following single-input single-output (SISO) control affine system Σ of the type:

$$\Sigma = \begin{cases} \dot{x} = f(x) + g(x)u \\ y = h(x) \end{cases} \tag{1}$$

where $f(x)$ and $g(x)$ are vector fields and $h(x)$ is a scalar field.

The relative order of the non-linear system Σ is given by the following definition:

Definition 1. For the SISO system Σ, the relative order α is defined as the smallest integer α such that $L_g L_f^{\alpha-1} h(x) \neq 0$, where $L_g L_f^{\alpha-1} h(x)$ is the Lie derivative of *h(x)* in the directions of *f(x)* and *g(x)*.

The relative order of the Σ may be regarded as the number of times that the output must be differentiated with respect to time, before the input appears explicitly.

Hirschorn [17] provided necessary and sufficient conditions for the invertibility of nonlinear systems, and a methodology for constructing a left inverse of the nonlinear system. For a nonlinear system to be invertible, it must have finite relative order, i.e. $\alpha < \infty$. The left inverse of a nonlinear system is another nonlinear system which, when driven by the appropriate derivative of y, produces u as its output. The important consequence of this definition is that, given a nonlinear map from u to y, the inverse map can be obtained by differentiating this mapping until the αth derivative of y is linear with respect to u. The manner in which the left-inverse is obtained is stated by the Hirschorn Inversion Theorem:

Definition 2. The left inverse of the system Σ is given by the following:

$$\dot{Z} = f(Z) + g(Z)u' \tag{2}$$

where

$$u' = \frac{(\mathrm{d}^{\alpha} y/\mathrm{d}t^{\alpha}) - L_f^{\alpha} h(Z)}{L_g L_f^{\alpha-1} h(Z)} \tag{3}$$

is the input to the left inverse system, α is the relative order of the system, and Z is the state of the inverse. $f(\cdot)$, $g(\cdot)$ and $h(\cdot)$ are the same as in the original system and $L_f^{a} h(Z)$ and $L_g L_f^{a-1} h(z)$ are Lie derivatives [17].

The zero dynamics of nonlinear systems, like the zeros of a linear system, describe the internal behaviour of a nonlinear system, when the output is forced to be equal to zero [21]. For a nonlinear system with relative order α, the initial state and input can be set as follows:

- The first α states are set equal to zero and the last $n-\alpha$ states can have their values chosen arbitrarily:

$$\begin{aligned} x_1 &= 0 \\ &\cdots \\ x_\alpha &= 0 \\ x_{\alpha+1} &= a_1 \\ &\cdots \\ x_n &= a_{n-\alpha} \end{aligned} \tag{4}$$

  where $a_1,\ldots,a_{n-\alpha}$ are chosen arbitrarily;
- The input is the solution of the following equation:

$$0 = f(0,\ldots,0,x_{\alpha+1}(t),\ldots,x_n(t)) + g_\alpha u \tag{5}$$

By solving for u, the zero dynamics of the SISO system can be given by the set of differential equations:

$$\begin{aligned} \dot{x}_{\alpha+1} &= f_{\alpha+1}(0,\ldots,0,x_{\alpha+1},\ldots,x_n) - \tfrac{g_{\alpha+1}}{g_\alpha} f_\alpha(0,\ldots,0,x_{\alpha+1},\ldots,x_n) \\ &\cdots \\ \dot{x}_n &= f_n(0,\ldots,0,x_{\alpha+1},\ldots,x_n) - \tfrac{g_n}{g_\alpha} f_\alpha(0,\ldots,0,x_{\alpha+1},\ldots,x_n) \end{aligned} \tag{6}$$

From this set of equations, the zero dynamics can be investigated. If the zero dynamics of the nonlinear system are stable, then the system is said to be minimum phase.

To demonstrate the stability of a linear system Σ the system needs to be linearised. This system can be linearised around an equilibrium point (x_0,u_0) of the system, by using the jacobians produced by differentiating f and g at the equilibrium point:

$$\dot{z} = Az + Bu \tag{7a}$$

where z is the linearised state and

$$A = \left. \begin{bmatrix} \dfrac{\partial}{\partial x_1} F_1(x) & \cdots & \dfrac{\partial}{\partial x_N} F(x) \\ \cdots & \dfrac{\partial}{\partial x_i} F_i(x) & \cdots \\ \dfrac{\partial}{\partial x_1} F_N(x) & \cdots & \dfrac{\partial}{\partial x_N} F_N(x) \end{bmatrix} \right|_{\substack{x=x_0 \\ u=u_0}} \tag{7b}$$

$$B = \left. \begin{bmatrix} \dfrac{\partial}{\partial u} F_1(u) \\ \cdots \\ \dfrac{\partial}{\partial u} F_M(u) \end{bmatrix} \right|_{\substack{x=x_0 \\ u=u_0}} \tag{7c}$$

where $F_i(x) = f_i(x) + g_i(x)u$. By analysing the matrix produced for A, the nature of the system's stability can be determined using the theories outlined below.

Theorem 1. (Liapunov's first method of stability). *If all the eigen values of the matrix A have non-zero real parts, the stability of the equilibrium point x_0 of the nonlinear system is the same as that of the equilibrium point z = 0 of the linearised system*.

Theorem 2. Linearisation Theorem [22]. *Let x_0 be an isolated equilibrium point of a system described by:*

$$\dot{x} = f(x) \tag{8}$$

If all the eigenvalues of the jacobian A have negative real parts, the equilibrium point x_0 is asymptotically stable. If at least one eigenvalue has a positive real part, x_0 is unstable.

Theorem 3. Gerschgorin's Theorem [23]. *Let $C \in \Re^{n\times n}$ representing the jacobian of f about an equilibrium point x_0. If:*

$$C_{ij} < 0 \quad i = 1, \dots, n$$
$$\sum_{\substack{j=1 \\ j\neq i}}^{n} \left|C_{ij}\right| < |C_{ii}| \quad i = 1, \dots, n \tag{9}$$

then every eigenvalue of matrix C has negative real part.

If Gerschgorin's Theorem holds for a particular system's *A* matrix, then the system can be said to be asymptotically stable. However, since Gerschgorin's Theorem is only sufficient and not necessary for eigenvalues to have negative real parts, systems for which Gerschgorin's Theorem do not hold may still be asymptotically stable.

3. Internal model control (IMC)

The concept of IMC was initially introduced by Garcia and Morari [4], to compensate for the inefficiency in the control design methodologies which were available at the time. IMC provides a useful control strategy for a large class of nonlinear systems which are input–output stable. A number of control schemes have been developed which are very similar to IMC, e.g. the Model Algorithm Control [24], Dynamic Matrix Control [25], and many others. Despite differences in their formulation, the underlying principles of each of these schemes are very similar.

IMC can theoretically provide perfect control. This is achieved by obtaining a model of the plant and then using the inverse of this model as the controller. IMC consists of three parts: an internal model, a filter and a controller. Construction of the IMC system involves two stages: selection of a controller (usually the model inverse) to give perfect control, followed by the introduction of a filter to make the system robust to model-plant mismatch. Fig. 1 shows the nonlinear block diagram of IMC for a nonlinear system.

Economou et al. [3] summarize the advantages of the IMC structure for nonlinear systems in three properties:

1. *Property* 1. Dual stability: Assume that the controller *C* and the plant with the effect of disturbances P_d are input–output stable and that a perfect model of the plant is available, i.e. ($M = P_d$). Then the closed-loop system is input–output stable.

2. *Property* 2. Perfect control: Assume that the inverse of the model exists, that this inverse is used as the controller and that the closed-loop system is input–output stable with this controller. Then the control will be perfect, i.e. $y=y_s$, where y is the output and y_s is the set-point.
3. *Property* 3. Zero offset: Assume that the inverse of the steady-state model operator exists, that the controller is equal to this inverse and that the closed-loop system is input–output stable with this controller. Then offset free control is attained for asymptotically constant inputs.

Property l guarantees closed loop stability of the system while Property 2 prescribes the structure and parameters of the controller which will result in perfect control. To achieve the perfect control of Property 2, the feedback controller requires infinite gain. This will result in sensitivity problems under model uncertainty and if the model is not exact, the closed-loop system can be unstable. The inclusion of a filter F reduces the gain at high frequency, and hence improves the robustness of the system. The filter also projects the error signal in to the appropriate input space for the controller and smooths out noisy/ rapidly changing signals, reducing the transient response of the controller.

These properties of the IMC controller are the motivating factors for the use of neural networks within the IMC framework. When feedforward neural networks are used, the inverse is usually produced by training a network. Economou et al. [3] do though describe a variety of other methods for producing the model inverse for use as a controller. Such techniques include analytic construction techniques e.g. the Hirsehorn inversion theorem [17], and numerical techniques which involve techniques such as Newton's algorithm to compute the inverse.

4. Hopfield neural networks for IMC

Hopfield networks are single layer networks, in which the outputs of the nodes are fed back both to themselves and to all the other nodes. Such networks can be described by a set of differential equations:

$$\dot{x}_i = -\beta_i x_i + \sum_{j=1}^{n} w_{i,j}\sigma(x_j) + \gamma_i u \quad i = 1, 2, \dots, n \qquad y = x_1 \tag{10}$$

where n is the number of nodes, x_i represents the state of the ith neuron, $w_{i,j}$ is the weight between the ith and jth nodes, β_i incorporates the time constant of the ith node, $\gamma_i u$ is the weighted external input, y is the output of the network and $\sigma(x)$ is an analytic and positive nonlinearity. $\sigma(x)$ is commonly defined to be either the hyperbolic tangent function, tan$h$ x, or the sigmoidal function, $1/(1+\mathrm{e}^{-x})$. The Hopfield network is an example of the control affine SISO system Σ. This can be verified by setting:

$$f_i(x) = -\beta_i x_i = \sum_{j=1}^{n} w_{i,j}\sigma(x_j), \qquad g_i = \gamma_i, \qquad h(x) = x_1 \tag{11}$$

An alternative to Eqs. (10) and (11) is given by the following:

$$\dot{x}_i = -\beta_i x_i + \sum_{j=1}^{n} \sigma(w_{i,j}, x_j) + \sigma(w_{i,n+1}, u), \quad i = 1, 2, \dots, n; \qquad y = x_i \tag{12}$$

where $w_{i,n+1}$ is γ_i [26].

Such a network can be used to produce a model of the nonlinear plant. Using the Hirschorn Inversion Theorem (Definition 2) it is possible to produce an inverse of a trained Hopfield neural network, without the need for further training [16]. However, in order to do so, the relative order of the network needs to be established.

4.1. Inversion of recurrent neural networks

In order to invert recurrent neural networks, it is required that such networks exhibit finite relative order. The relative order is bounded by the number of internal states of the network. Since each state of the network corresponds to a node, and the total number of states is the same as the number of nodes, for recurrent networks, the upper bound on α is the number of nodes. The following two theorems illustrate this particular property and provide us with canonical forms which if a priori information is available enables us to improve the efficiency of training these networks [27,28]. For plants where the relative order is not known, these theorems can be used to confirm the relative order.

The relative order of a Hopfield neural network can be established by using conditions generated from the definition of relative order Definition 1 [16,29].

Theorem 4. *Assume that $\gamma_i = 0$, $i = 1,2,\ldots, n-1$; $\gamma_n \neq 0$. Assume further that the output y is not constant, then given an analytic and positive nonlinear function $\sigma(x)$, the relative order of the Hopfield network is α, $\alpha \in [2, n]$, if the elements of the network weights satisfy the following set of conditions:*

$$\begin{aligned}
&w_{1,n} = 0\\
&w_{1,3}, \ldots, w_{1,n-1}, w_{2,n} = 0\\
&w_{2,4}, \ldots, w_{2,n-1}, w_{3,n} = 0\\
&w_{3,5}, \ldots, w_{3,n-1}, w_{4,n} = 0\\
&\ldots\\
&w_{\alpha-4,\alpha-2}, \ldots, w_{\alpha-4,n-1}, w_{\alpha-3,n} = 0\\
&w_{\alpha-3,\alpha-1}, \ldots, w_{\alpha-3,n-1}, w_{\alpha-2,n} = 0\\
&w_{\alpha-2,\alpha}, \ldots, w_{\alpha-2,n-1}, w_{\alpha-1,n} \neq 0
\end{aligned}$$

Theorem 5. *Assume that $\gamma_i = 0$, $i = 1,2,\ldots,m$; $m < n$, and $\gamma_j \neq 0$; $j = m + 2,\ldots,i$, where n is the number of states. Assume further that the output ii is not constant. Then given a sigmoidal function $\sigma(x)$, the relative order of the Hopfield network is α, $\alpha \in [2, m + 1]$, if the elements of the network weights satisfy the following set of conditions:*

$$\begin{aligned}
&w_{\alpha-2,\alpha}, \ldots, w_{\alpha-2,n-1}, w_{\alpha-1,m+1}, \ldots, w_{\alpha-1,n} \neq 0\\
&w_{\alpha-3,\alpha-1}, \ldots, w_{\alpha-3,n-1}, w_{\alpha-2,m+1}, \ldots, w_{\alpha-2,n} = 0\\
&w_{\alpha-4,\alpha-2}, \ldots, w_{\alpha-4,n-1}, w_{\alpha-3,m+1}, \ldots, w_{\alpha-3,n} = 0\\
&\ldots\\
&w_{3,5}, \ldots, w_{3,n-1}, w_{4,m+1}, \ldots, w_{4,n} = 0\\
&w_{2,4}, \ldots, w_{2,n-1}, w_{3,m+1}, \ldots, w_{3,n} = 0\\
&w_{1,3}, \ldots, w_{1,n-1}, w_{2,m+1}, \ldots, w_{2,n} = 0\\
&w_{1,m+1}, \ldots, w_{1,n} = 0
\end{aligned}$$

If, however, $\gamma_1 \neq 0$ then the relative order is $\alpha = 1$.

For completeness, details of these can be seen in Appendix B.

As an illustration, consider the example of a Hopfield network with three states, the state space equations are given by:

$$\begin{aligned}
\dot{x}_1 &= -\beta_1 x_1 + w_{1,1}\sigma(x_1) + w_{1,2}\sigma(x_2) + w_{1,3}\sigma(x_3) + \gamma_1 u \\
\dot{x}_2 &= -\beta_2 x_2 + w_{2,1}\sigma(x_1) + w_{2,2}\sigma(x_2) + w_{2,3}\sigma(x_3) + \gamma_2 u \\
\dot{x}_1 &= -\beta_3 x_3 + w_{3,1}\sigma(x_1) + w_{3,2}\sigma(x_2) + w_{3,3}\sigma(x_3) + \gamma_3 u
\end{aligned} \tag{13}$$

If the components of γ are given such that $\gamma_1 = 0$, $\gamma_2 \neq 0$ and $y_3 \neq 0$, therefore, $m = 1$, and if the weights of the matrix are such that:

$$\left.\begin{matrix} w_{1,2} \\ w_{1,3} \end{matrix}\right\} \neq 0 \quad \text{and} \quad \left.\begin{matrix} w_{2,2} \\ w_{2,3} \\ w_{3,2} \\ w_{3,3} \end{matrix}\right\} = 0$$

then the network has a relative order α of 2.

To illustrate the construction of an inverse of a Hopfield neural network, consider the three state Hopfield network used above Eq. (13). Using the Hirsehorn inversion theorem (Definition 2), the inverse of this network is given by the following equations:

$$\begin{aligned}
\dot{z}_1 &= -\beta_1 z_1 + w_{1,1}\sigma(z_1) + w_{1,2}\sigma(z_2) + w_{1,3}\sigma(z_3) + \gamma_1 u' \\
\dot{z}_2 &= -\beta_2 z_2 + w_{2,1}\sigma(z_1) + w_{2,2}\sigma(z_2) + w_{2,3}\sigma(z_3) + \gamma_2 u' \\
\dot{z}_3 &= -\beta_3 z_3 + w_{3,1}\sigma(z_1) + w_{3,2}\sigma(z_2) + w_{3,3}\sigma(z_3) + \gamma_3 u'
\end{aligned} \tag{14}$$

where z_1 and z_2 are the states of the inverse and

$$u' = \frac{\begin{array}{c}[(\mathrm{d}^2 y/\mathrm{d}t)\{(-\beta_1 + w_{1,1}\sigma'(x_1))(-\beta_1 x_1 + \sum_{j=1}^{3} w_{1,j}\sigma(x_j)) + w_{1,2}\sigma'(x_2)(-\beta_2 x_2 \\ +\sum_{j=1}^{3} w_{2,j}\sigma(x_j)) + w_{1,3}\sigma'(x_3)(-\beta_3 x_3 + \sum_{j=1}^{3} w_{3,j}\sigma(x_j))\}]\end{array}}{\gamma_1(-\beta_1 x_1 + \sum_{j=1}^{3} w_{1,j}\sigma(x))}$$

It is clear from this example, that the inverse of a given Hopfield network is another Hopfield network with the same weights, and a different input.

4.2. *Stability of recurrent neural networks*

When using the Hopfield network in IMC, it is important that the network is stable, since IMC closed loop is internally stable if all its component parts are stable, i.e. if the plant, the model, the controller and the filter are all input–output stable. Different theories exist for determining the stability of a recurrent network, but many of these require specific conditions on the weight matrix of the network, e.g. the weight matrix W must be symmetric ($w_{ij} = w_{ji}$) and that all the weights must be greater than, or equal to 0, ($w_{ij} \geq 0$) [30], or that the weight matrix must be symmetric and that it must have a zero diagonal ($w_{ii} = 0$) [12,31–33]. These conditions are restrictive and it is possible to demonstrate the stability of a Hopfield neural network without the need for these conditions [29].

Using matrix measures (see Appendix C), sufficient conditions for the asymptotic stability can be presented [29], using the first method of Liapunov. Using this method, the network is linearised and the stability of each equilibrium state is investigated separately. The Hopfield network given by Eq. (10), when linearised can be given by the following equations:

$$\dot{z} = Az \tag{15}$$

where z is the linearised state and A is shown in Eq. (16):

$$A = \begin{bmatrix} h_{11} - \beta_1 & \dots & h_{1N} \\ \dots & h_{ii} - \beta_i & \dots \\ h_{N1} & \dots & h_{NN} - \beta_N \end{bmatrix} \quad h_{ij} = w_{ij}\sigma'(x_0), \quad i = 1, \dots, N; \quad j = 1, \dots, N \tag{16}$$

where x_0 is the equilibrium state and

$$\sigma'(x_0) = \left. \frac{d}{dx}\sigma(x) \right|_{x=x_0}$$

For asymptotic stability, the matrix measure μ^1 gives the following condition:

$$-(h_{jj} - \beta_j) > \sum_{i \neq j} |h_{ij}| \quad \forall j \tag{17}$$

and the matrix measure μ^∞ gives the following condition for asymptotic stability:

$$-(h_{ii} - \beta_i) > \sum_{i \neq j} |h_{ij}| \quad \forall i \tag{18}$$

4.3. *Hopfield neural networks in IMC*

Using recurrent neural networks within the IMC closed loop system has two advantages over the use of techniques such as feedforward neural networks. These are:

1. The model inverse (for use as a controller) can easily be obtained from the model;
2. Stability of the recurrent neural network and of the inverse controller can easily be observed.

Hopfield network are particularly suited for use as models within the IMC strategy, since they have the ability to model nonlinear mappings/functions. When such a network is been used to model the plant, the inverse of the model is easily obtainable, since the inverse is the model itself, provided with a different input. This is due to the fact that the input–output dynamics of the Hopfield network and the inverse network are the same. Therefore, an inverse model can be obtained from the forwards model without requiring extra training or calculation.

Stability of Hopfield networks is easily demonstrated, using the theory presented above. Thus, stability of the plant model can easily be observed. Since the inverse controller has the same dynamics as the plant model, it also has the same stability. Therefore, it is sufficient to show that the plant model is stable, to establish stability of the inverse controller, when such a model is a Hopfield network.

5. Simulation examples

To illustrate the use of Hopfield networks for IMC, two systems were chosen, a single link manipulator and a chemical reactor. In each case a Hopfield network was trained to model the system, using a modified

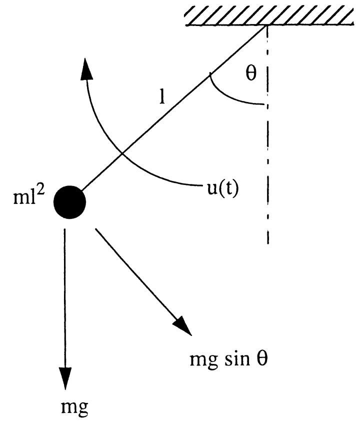

Fig. 2. Single link manipulator.

chemotaxis algorithm [28,34]. From this network, a controller was produced using the Hirschorn inversion theorem.

5.1. *The single link manipulator*

The single link manipulator (Fig. 2) is described by the following second order nonlinear differential equation:

$$ml^2\ddot{\theta}(t) + v\ddot{\theta}(t) + mgl\sin\theta(t) = u(t) \tag{19}$$

where $m = 2.0$ kg, $l = 1$ m, and $v = 1.0$ kg m^2/s. The corresponding state space representation is:

$$\begin{aligned} \dot{x}_2 &= x_2 \\ \dot{x}_2 &= -\frac{v}{ml^2}x_2 - \frac{mgl}{ml^2}\sin x_1 + \frac{u}{ml^2} \\ y &= x_1 \end{aligned} \tag{20}$$

where $\dot{x}_1 = \dot{\theta}$ and $\dot{x}_2 = \dot{\theta}$. It can be verified that the relative order of the system is 2. The eigenvalues of the linearised system are:

$$\xi_{ps} = \begin{bmatrix} -0.25 + 3.1205i \\ -0.25 - 3.1205i \end{bmatrix}$$

The real parts ot these eigenvalues are not equal to 0, thus from Liapunov's first theorem, stability of the nonlinear plant is equivalent to stability of its linearised counterpart. Further since the real parts of these eigenvalues are less than 0, it is implied that the system is stable.

5.2. *The Van der Vusse chemical stirred tank reactor*

The Van der Vusse chemical stirred tank reactor [35,36] can be represented by the following set of equations:

$$\dot{x}_1 = -50x_1 - 10x_1^2 + q_c(10 - x_1)$$

Table 1
Weight matrix W for the Hopfield network modelling the single link manipulator

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

Table 2
Weight vectors Γ and β for the Hopfield network modelling the single link manipulator

Γ	β
−0.0050	1.0
−0.2111	1.0
0.1689	1.0
0.0645	1.0
−0.0413	1.0

Table 3
Weight matrix W for the Hopfield network modelling the Van der Vusse reactor

−36.111633	2.907848
54.807533	−4.787264

$$\dot{x}_2 = 50x_1 - 10x_2 + q_c x_2 \tag{21}$$

$$y = x_2$$

where q_c is the input to the system and represents the coolant flow rate, x_1 is the concentration of the input chemical and x_2 is the concentration of the output chemical. The relative order of this system is 1, and the eigenvalues of the linearised system are:

$$\xi_{pv} = \begin{bmatrix} -100 \\ -50 \end{bmatrix}$$

As was the case with the single link manipulator, the real parts of these eigenvalues are not equal to 0, thus from Liapunov's first theorem, stability of the nonlinear plant is equivalent to stability of its linearised counterpart and since the real parts of these eigenvalues are less than 0, it is implied that the system is stable.

5.3. Analysis of the neural network controllers

The Hopfield network used to model the single link manipulator consisted of five hidden nodes and the network used to model the Van der Vusse reactor consisted of two hidden nodes. The weight matrices for these networks are presented in Tables 1–4. The input–output approximations are shown in Fig. 3.

Using Definition 1 and the parameters from Tables 1 and 2 that the Hopfield network modelling the single link manipulator has a relative order of 2, since $\gamma_1 = 0$ and $\gamma_2 \neq 0$. Similarly from Tables 3 and 4, it

Table 4
Weight vectors Γ and β for the Hopfield network modelling the Van der Vusse reactor

Γ	β
58.145828	14.000782
50.849155	95.250351

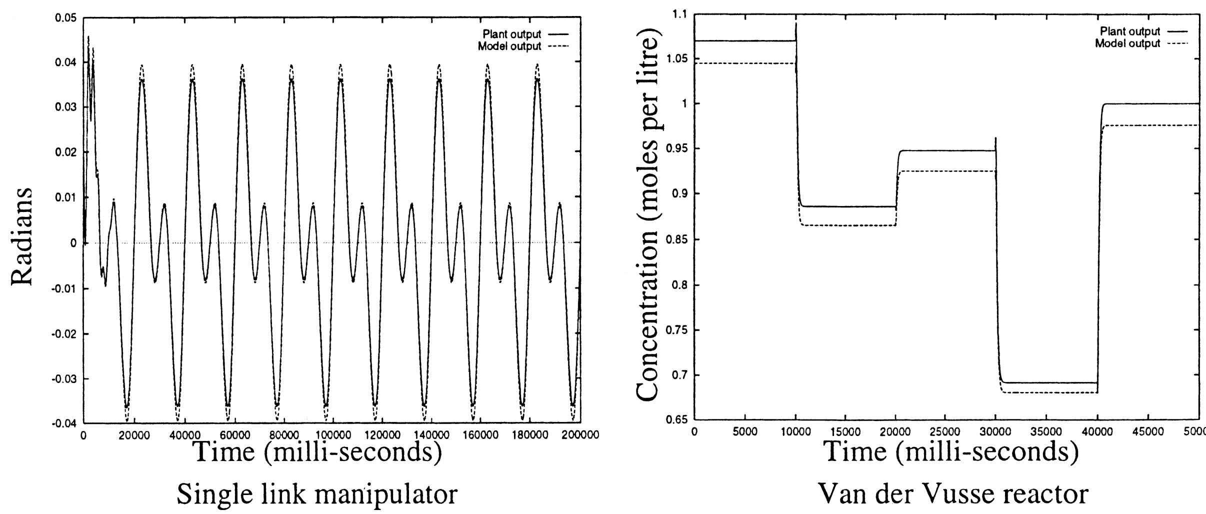

Fig. 3. Comparison between plant and model outputs.

can be seen that the Hopfield network modelling the Van der Vusse reactor, has a relative order of 1 since $\gamma_1 \neq 0$. Both networks have finite relative order, thercfore, they are invertible and the inverse networks can be produced using Definition 2.

Before constructing the inverse, it is imperative that it is verified that the networks produced arc (a) stable, and (b) have eigenvalues close to those of the plants. It has been discussed in Section 4.2 that the stability of neural networks and their inverscs can be established by linearising the networks and using conditions produced from matrix measures. Hopfield networks trained for the two systems can be linearised using the forms developed in Eqs. (15) and (16).

The Hopfield network modelling the single link manipulator when linearised has the following eigenvalues:

$$\xi_{\mathrm{ms}} = \begin{bmatrix} -0.2260 + 3.1483i \\ -0.2260 - 3.1483i \\ -2.0288 + 0.1798i \\ -2.0288 - 0.1798i \\ -0.4082 \end{bmatrix}$$

Since the real part of these eigenvalues are not equal to 0, stability of the linearised and nonlinear forms is equivalent. It is also important to note that the dominant eigenvalues (indicated by arrows) of the model are close to those of the single link manipulator.

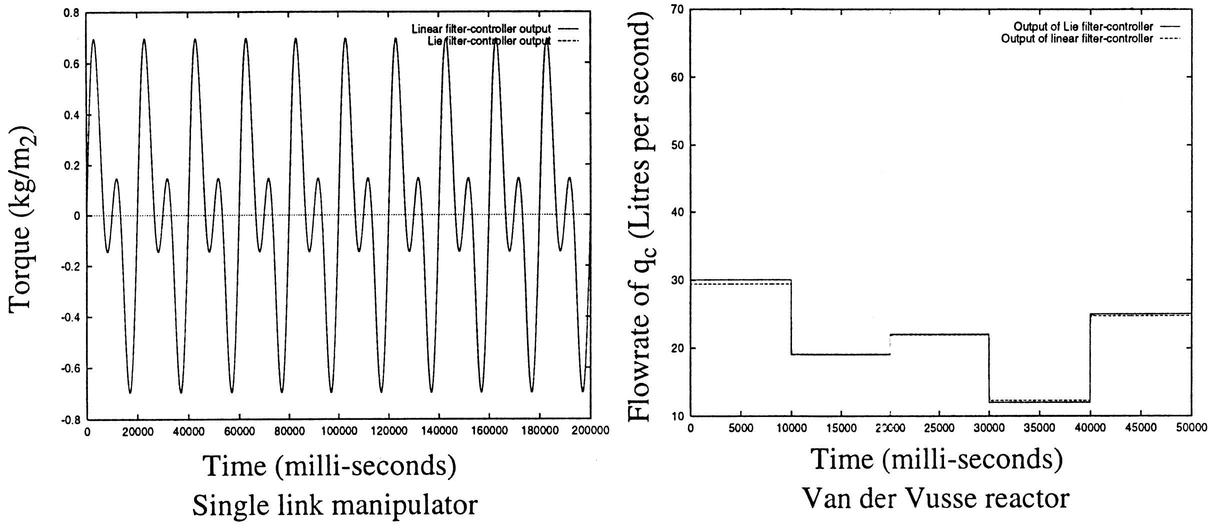

Fig. 4. Comparison of controller outputs produced using Lie derivative filter and linear filter.

The Hopfield network modelling the Van der Vusse reactor, when linearised has the following eigenvalues:

$$\xi_{\mathrm{mv}} = \begin{bmatrix} -47.1018 \\ -103.0483 \end{bmatrix}$$

As with the network modelling the single link manipulator, for the network modelling the Van der Vusse reactor, the real parts of the eigenvalues of the linearised form are not equal to 0, thus stability of the linearised and nonlinear forms is equivalent. Again notice that the eigenvalues of the network are close to those of the linearised plant.

Using the condition for asymptotic stability provided by the matrix measure m^{∞}, the Hopfield networks modelling both the single link manipulator and the Van der Vusse reactor are asymptotically stable.

5.4. Control results

The following diagrams show the results of producing a controller for both the single link manipulator and for the Van der Vusse reactor. Each diagram shows two graphs, the first for the manipulator and the second for the reactor.

Using the models described above, the inverse controllers were produced. Fig. 4 shows the outputs of the controller network using both the Lie derivatives and the linear filter to produce the input to the controller. In comparison to the Lie derivatives, the linear filter produces an approximate input for the controller. This is clearly visible in the graph produced by the Van der Vusse reactor. However, the resultant controller outputs are close enough to allow the linear filter to be used instead of the Lie derivatives. The linear filter has the advantage of being less computationally intensive than the Lie derivatives. In Fig. 5, it is shown that when supplied with the output of the plant, the controller produces an output equal to the plant input, thus demonstrating that the controller is the inverse of the plant. Figs. 6 and 7 show the plant outputs and reference signal from the closed loop control, using the linear filter and inverse network to

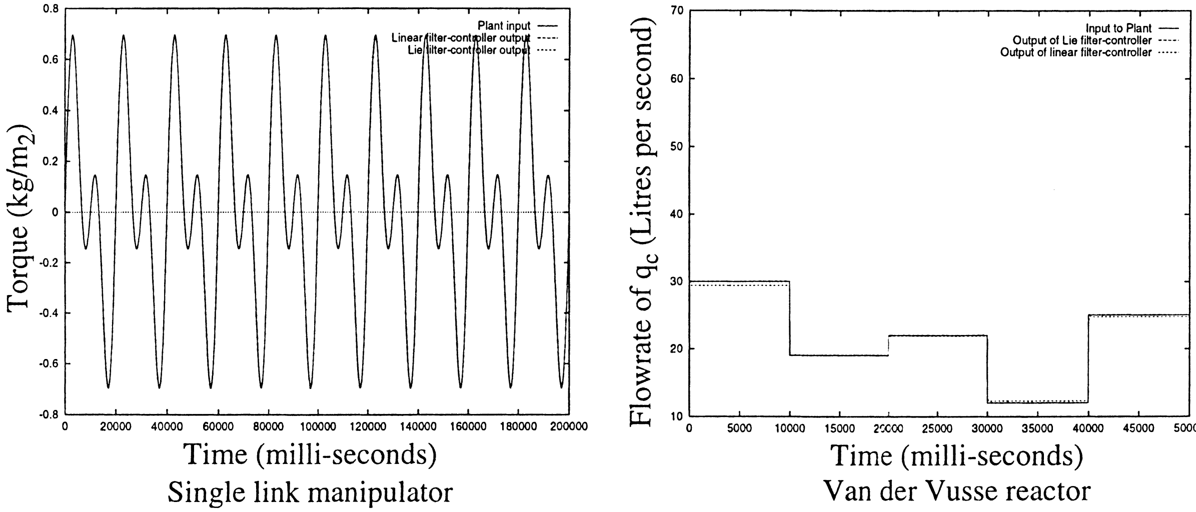

Fig. 5. Comparison between controller outputs and plant input.

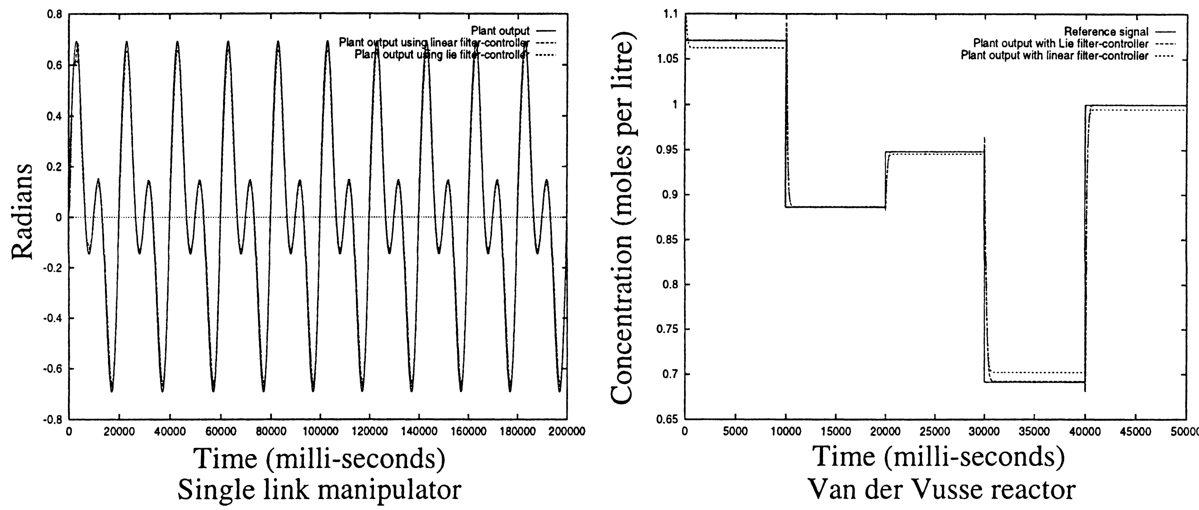

Fig. 6. Output of plant in closed loop when supplied with a reference signal.

control the plant. The plant outputs in Fig. 7 have a disturbance of ±0.1 added to them. It can be seen that the control is good, both in the absence and presence of a disturbance.

6. Discussion

It has been shown that Hopfield networks can control nonlinear systems, when incorporated into the IMC strategy. Using such network in this strategy has the main advantage that the strucutre and parameters of the inverse controller are the same as those of the network used to model the nonlinear system. Most other techniques for production of the network inverse involve training an addtional network for use as

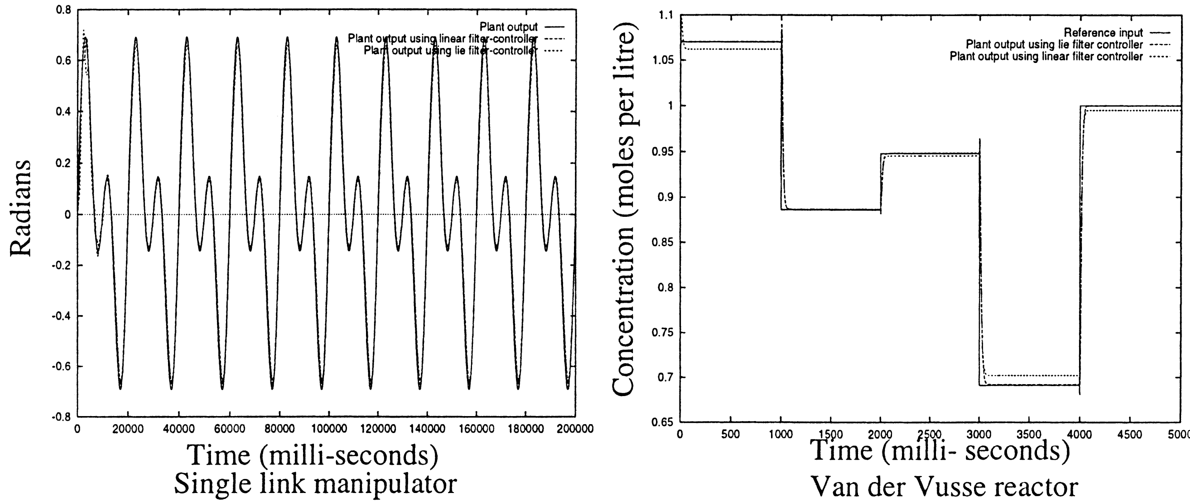

Fig. 7. Plant output for closed loop control with disturbance added to the output.

the inverse controller. Stablity of the complete control system can be established, since the IMC closed loop is stable if all its components are stable. Stablity of the plant and network models can be easily demonstrated by linearisation and examination of the eigenvalues of the linearised autonomous system. Stablity of the inverse controller networks can be established from the stablity of the network models, since the two components have the same structure and parameters. The linear filters have their parameters chosen so as to ensure their stablity, through placement of their poles.

The network models can also be shown have modelled the plants well, by having the same realtive order as the plants, and by having equivalent dynamics to those of the plants, shown in the eigenvalues of plant and network being equal to or approximatly equal. The inverse network can be thought of as cancelling the nonlinearities of the plant, thus, a block continuing the inverse network and the plant has been effectively linearised, and its nonlinear elements cancelled.

Although the recurrent networks examined in this paper are all of the Hopfield architecture, the theories can be applied to other types of recurrent network e.g. the RN1 network and the RN2 network.

Appendix A. Lie derivatives

The Lie derivative of scalar field $h(x)$ in the direction of the vector field $f(x)$ is defined as:

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

and in the direction of the vector field $g(x)$ is defined as:

$$L_g h(x) = \sum_{i=1}^{n} \frac{\partial h(x)}{\partial x_i} g_i(x) \tag{A.2}$$

It is possible to differentiate the scalar field $h(x)$ first in the direction of the vector field $f(x)$ and then in the direction of the vector field $g(x)$:

$$L_g L_f h(x) = \sum_{i=1}^{n} \frac{\partial L_f h(x)}{\partial x_i} g_i(x) \tag{A.3}$$

or in the same direction $f(x)$ twice:

$$L_f^2 h(x) = \sum_{i=1}^{n} \frac{\partial L_f h(x)}{\partial x_i} f_i(x) \tag{A.4}$$

Appendix B. Proof of Theorems 4 and 5

Proof of Theorem 4. This proof is constructive by nature. It can be seen that $L_g h(X) = \gamma_1 \sigma'(x_1)$ which implies, by the assumptions that $L_g h(X) = 0$.

$$L_f h(x) = (\sum_{j=1}^{n} w_{1,j} \sigma(x_j) + \beta_1 x_1) \sigma'(x_1))$$

$$L_g L_f h(x) = \sum_{j=1}^{n} \gamma_i \frac{\partial}{\partial x_i} \{[\sum_{j=1}^{n} w_{1,j} \sigma(x_j) + \beta_1 x_1)] \sigma'(x_1)) \tag{B.1}$$

Thus,

$$L_g L_f h(x) \neq 0 \quad \text{if } w_{1,j} \neq 0 \quad \forall j < n \quad \text{and} \quad \text{then } \alpha = 2$$

We have

$$L_g L_f h(x) = 0 \quad \text{iff } w_{1,n} = 0 \quad \forall j < n \quad \text{and} \quad \text{then } \alpha > 2. \qquad \square \tag{B.2}$$

Then assuming that $\alpha > 2$ and also noting that $w_{1,n} = 0$, the following is obtained:

$$L_f^2 h(x) = \sigma''(x_1) f_1^2 + \sigma'(x_1)[\beta_1 + w_{1,1} \sigma'(x_1)] f_1 + w_{1,2} \sigma'(x_2) \sigma'(x_2) \sigma'(x_1) f_2 + \cdots$$
$$+ w_{1,n} \sigma'(x_n) \sigma'(x_1) f_n$$

where f_i is as in Eq. (1).

The conditions for which the Hopfield network has a relative order greater than 3 can now be verified. The Lie derivative of $L_f^2 h(x)$ with respect to $g(x)$ is:

$$L_g L_f^2 h(x) = (w_{1,2} \sigma'(x_2) \sigma'(x_1) w_{2,n} \sigma'(x_n) + \cdots + w_{1,n-1} \sigma'(x_{n-1}) \sigma'(x_1) w_{n-1,n} \sigma'(x_n)) \gamma_n \tag{B.3}$$

Thus, $L_g L_f^2 h(x) = 0$ if (i) $w_{2,n}, \ldots, w_{n-1,n} = 0$ or (ii) $w_{1,3}, \ldots, = 0$ and $w_{2,n} = 0$, and then $\alpha > 3$.

The first condition is not feasible, since this would mean that the input never affects the output. Thus, for relative order greater than 3, it is sufficient to have:

$$w_{1,3}, \ldots, w_{1,n} = 0 \quad \text{and} \quad w_{2,n} = 0 \tag{B.4}$$

Now assuming that Eq. (B.4) is satisfied, conditions for the relative order to be greater than 4 can be investigated. Thus, from Eqs. (B.2) and (B.4) the following are obtained:

$$L_f^2 h(x) = A + B \tag{B.5}$$

where

$$\begin{aligned} A = {} & \sigma''(x_1)[w_{1,1}\sigma(x_1) + w_{1,2}\sigma(x_2) + \beta_1 x_1]^2 + \sigma'(x_1)[\beta_1 + w_{1,1}\sigma'(x_1)] \\ & [w_{1,1}\sigma(x_1) + w_{1,2}\sigma(x_2) + \beta_1 x_1] \end{aligned} \tag{B.6}$$

$$B = w_{1,2}\sigma'(x_2)\sigma'(x_1) f_2 \tag{B.7}$$

and

$$\begin{aligned} L_f^3 h(x) = {} & \left[\frac{\partial A}{\partial x_1} + \frac{\partial B}{\partial x_1}\right][w_{1,1}\sigma(x_1) + w_{1,2}\sigma(x_2) + \beta_1 x_1] + \left[\frac{\partial A}{\partial x_2} + \frac{\partial B}{\partial x_1}\right][w_{2,1}\sigma(x_1) \\ & + w_{2,2}\sigma(x_2) + w_{2,3}\sigma(x_3) + \beta_2 x_2] + w_{1,2}\sigma'(x_2)\sigma'(x_1) w_{2,3}\sigma'(x_3) f_3 \\ & + \cdots + w_{1,n}\sigma'(x_n)\sigma'(x_1) w_{2,n}\sigma'(x_n) f_n \end{aligned} \tag{B.8}$$

$$\begin{aligned} L_g L_f^3 h(x) = {} & (w_{1,2}\sigma'(x_2)\sigma'(x_1) w_{2,3}\sigma'(x_3) w_{3,n}\sigma'(x_n) + w_{1,2}\sigma'(x_2)\sigma'(x_1) w_{2,4}\sigma'(x_4) w_{4,n}\sigma'(x_n) \\ & + \cdots + w_{1,2}\sigma'(x_2)\sigma'(x_1) w_{2,n-1}\sigma'(x_{n-1}) w_{n-1,n}\sigma'(x_n))\gamma_n \end{aligned} \tag{B.9}$$

Using similar arguments as above it can be seen that for $\alpha > 4$, $L_g L_f^3 h(x) = 0$. This is possible if Eq. (B.4) is satisfied and if $w_{2,4},\ldots,w_{2,n-1} = 0$ and $w_{3,n} = 0$. Recursively, the following can be obtained for $L_g L_f^{a-1} h(x)$:

$$\begin{aligned} & w_{1,2}\sigma'(x_2)\sigma'(x_1) w_{2,3}\sigma'(x_4) w_{3,4}\sigma'(x_4) \cdots w_{\alpha-1,n}\sigma'(x_n) w_{1,2}\sigma'(x_2)\sigma'(x_1) w_{2,4}\sigma'(x_4) w_{4,5}\sigma'(x_5) \\ & \cdots w_{\alpha,n}\sigma'(x_n) w_{1,2}\sigma'(x_2)\sigma'(x_1) w_{2,4}\sigma'(x_4) w_{4,5}\sigma'(x_5) \cdots w_{\alpha,n}\sigma'(x_n) \end{aligned} \tag{B.10}$$

For relative order α

$$L_g L_f^i h(x) = 0 \quad \forall i = 0, 2, \ldots, \alpha - 2 \qquad L_g L_f^{\alpha-1} h(x) \neq 0 \tag{B.11}$$

In terms of the network weights, these requirements Eq. (B.11) translate to the following set of conditions:

$$\begin{aligned} & w_{1,n} = 0 \\ & w_{1,3}, \ldots, w_{1,n-1}, w_{2,n} = 0 \\ & w_{2,4}, \ldots, w_{2,n-1}, w_{3,n} = 0 \\ & w_{3,5}, \ldots, w_{3,n-1}, w_{4,n} = 0 \\ & \ldots \\ & w_{\alpha-4,\alpha-2}, \ldots, w_{\alpha-4,n-1}, w_{\alpha-3,n} = 0 \\ & w_{\alpha-3,\alpha-1}, \ldots, w_{\alpha-3,n-1}, w_{\alpha-2,n} = 0 \\ & w_{\alpha-2,\alpha}, \ldots, w_{\alpha-42n-1}, w_{\alpha-1,n} \neq 0 \end{aligned} \tag{B.12}$$

and is the required weight matrix structure of the Hopfield network for relative order α. This result can be interpreted by regarding each neuron as an integrator, with α representing the number of intergrations that have to be performed before the input affects the system output.

Proof of Theorem 5. The proof of this theorem follows that of Theorem 4, noting that in this case:

$$\dot{x}_i = \sum_{j=1}^{m} w_{ij}\sigma(x_j) + \sum_{j=m+1}^{n} w_{ij}\sigma(x_j) + \beta_i x_i + \gamma_i u \quad \forall i = 1, 2, \ldots, n \qquad \square$$

Appendix C. Matrix measures

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

$$\mu(A) = \lim_{\varepsilon \to 0^+} \frac{||I + \varepsilon A||_i - 1}{\varepsilon} \tag{C.1}$$

Different norms can be used to produce different matrix measures. Eq. (C.2) shows the matrix measure corresponding to the i^1 norm in $\Re^2$ and Eq. (C.3) shows the matrix measure corresponding to the l^∞ norm in $\Re^2$.

$$\mu_1(A) = \max_j \left[a_{jj} + \sum_{i \neq j} |a_{ij}| \right] \tag{C.2}$$

$$\mu_\infty(A) = \max_i \left[a_{ii} + \sum_{j \neq 1} |a_{ij}| \right] \tag{C.3}$$

References

[1] J.D. Morningred, B.E. Paden, D.E. Seborg, D.A. Mellichamp, An adaptive nonlinear predictive controller, Proc. ACC, San Diego, USA, 1990. pp. 1614–1619.
[2] S. Manchanda, M.T. Tham, An Evaluation of Nonlinear Control Strategies, Control 91, 1991.
[3] C.G. Economou, M. Morari, B.O. Palsson, Internal model control. 5. Extension to nonlinear systems, Ind. Eng. Chem. Process Des. Dev. 25 (1986) 403–411.
[4] C.E. Garcia, M. Morai, Internal model control. 1. A unifying review and some new results, Ind. Eng. Chem. Process Des. Dev. 21 (1982) 308–323.
[5] M.J. Willis, C. Di Massimo, G.A. Montague, M.T. Tham, A.J. Morris, On artifical neural networks in process engineering, Proc. IEE Part D 138 (1991) 256–266.
[6] M.A. Henson, D.E. Sebong, A unified differential geometric approach to non-linear process control, AlChE Annual Meeting, 1989, San Francisco, USA
[7] G. Cybenko, Approximation by superpositions of a sigmoidal function, Math. Control Syst. 2 (1989) 303–314.
[8] K. Hornik, M. Stincheombe, H. White, Multi-layer feedforward networks are universal approximators, Neural Networks 2 (1989) 359–366.
[9] K. Funahashi, On the approximate realization of continuous mappings by neural networks, Neural Networks 2 (1989) 183–192.
[10] K.J. Hunt, D. Sbarbaro, R. Zbikowski, P.J. Gawthrop, Neural networks for control systems — a survey, Automatica 28(6) (1992) 1083–1112.
[11] G. Lightbody, G.W. Irwin, Nonlinear control structures based on embedded neural system models, IEEE Trans. Neural Networks 8(3) (1997) 553–567.

[12] J.J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, Proc. Natl. Acad. Sci. 79 (1982) 2554–2558.
[13] K.S. Narendra, K. Parthasarathy, Adaptive identification and control of dynamical systems using neural networks, IEEE Trans. Neural Networks 1 (1990) 4–27.
[14] M.M. Polycarpou, P.A. Ioannou, Identification and control of nonlinear systems using neural network models: design and stability analysis, Technical Report, USC, Los Angeles, Report Number 91-09-01, 1991.
[15] K. Funahashi, Y. Nakamura, Approximation of dynamical systems by continuous time recurrent neural networks, Neural Networks 6 (1993) 801–806.
[16] C. Kambhampati, S. Manchanda, A. Delgado, G. Green, K. Warwick, M. Tham, Relative order and inverses of a class of recurrent networks, Automatica 32(1) (1995) 117–123.
[17] R.M. Hirschorn, Invertibility of nonlinear control systems, SIAM J. Cont. Opt. 17 (1979) 289–297.
[18] C. Kambhampti, K. Warwick, Use of stochastic neural networks for process control, in: Proc. IFAC DYCORD 92, Maryland, USA, 1992, pp. 93–98.
[19] K. Hunt, D. Sbarbaro, Neural networks for nonlinear internal model control, IEE Proc. D 138(5) (1991) 431–438.
[20] C. Kravaris, J.C. Kantor, Geometric methods for nonlinear process control. 1. Background, Ind. Eng. Chem. Res. 29 (1990) 2295–2310.
[21] A. Isidori, Nonlinear Control Systems. An Introduction, 2nd edn., Springer, Berlin, 1989.
[22] J. Guckenheimer, P. Holmes, Nonlinear Oscillations, Dynamical Systems and Bifurcations of Vector Fields, Springer, Berlin, 1983.
[23] A. Ralston, A First Course in Numerical Analysis, McGraw-Hill, New York, 1965.
[24] J. Richalet, A. Rault, J. Testud, J. Papon, Automatica 14 (1978) 413.
[25] C.R. Cutler, B.L. Ramaker, Dynamic matrix control — a computer control algorithm, in: Proc. Conf. on Joint Automatic Control, San Francisco, 1980.
[26] F. Albertini, B. Sontag, For neural networks, function determines form, Neural Networks 6 (1993) 975–990.
[27] K.P. Dimopoulos, C. Kambhampati, Apriori information in network design, in: M. Karny, K. Warwick, V. Kurkova (Eds.), Dealing with Complexity: A Neural Network Approach, 1988.
[28] R.J. Craddock, C. Kambhampati, Use of apriori system knowledge to train recurrent neural networks to be stable, Internal Report, Department of Cybernetics, University of Reading, 1998.
[29] A. Delgado, C. Kambhampati, K. Warwick, Dynamic recurrent neural networks for system identification and control, IEE Proc. Control Theory Appl. 142(4) (1995) 307–314.
[30] M.A. Cohen, S. Grossberg, Absolute stability of global pattern formation and parallel memory storage by competitive neural networks., IEEE Trans. on System Man and Cybernetics 13 (1983) 815–826.
[31] J.J. Hopfield, D.W. Tank, Neural computation of decisions in optimization problems biology, Cybernetics 52 (1985) 141–152.
[32] J.J. Hopfield, D.W. Tank, Computing with neural circuits: a model, Science 23 (1986) 625–633.
[33] D.W. Tank, J.J. Hopfield, Simple ‘neural’ optimization networks: an A/D converter, signal decision circuit, and a linear programming circuit, IEEE Trans. Syst. Man Cybern. 33(5) (1986) 533–541.
[34] H.J. Bremmermann, R.W. Anderson, An alternative to back propagation: a simple rule of synaptic modification for neural network training and memory, Report PAM-483. Center for Pure and Applied Mathematics, University of California Berkley, 1990.
[35] J. Van de Vusse, Plug flow type verses tank reactor, Chem. Eng. Sci. 19 (1964) 994–997.
[36] J. Kantor, Stability of state feedback transformations for nonlinear systems — some practical considerations, in: Proc. American Control Conf., 1986, pp. 1014–1016.
[37] S. Manchanda, C. Kambhampati, M.T. Tham, G.G.R. Green, The relative order of a class of recurrent networks, in: Proc. IEE Conf. 1994, University of Warwick, UK, 1993, pp. 973–977.
[38] S. Manchanda, Nonlinear process control, Ph.D. Thesis, Department of Chemical and Process Engineering, University of Newcastle-upon-Tyne, UK, 1993.