Adaptive control of discrete-time nonlinear systems using recurrent neural networks

L. Jin
P.N. Nikiforuk
M.M. Gupta

Abstract: A learning and adaptive control scheme for a general class of unknown MIMO discrete-time nonlinear systems using multilayered recurrent neural networks (MRNNs) is presented. A novel MRNN structure is proposed to approximate the unknown nonlinear input-output relationship, using a dynamic back propagation (DBP) learning algorithm. Based on the dynamic neural model, an extension of the concept of the input-output linearisation of discrete-time nonlinear systems is used to synthesise a control technique for model reference control purposes. A dynamic learning control architecture is developed with simultaneous online identification and control. The potentials of the proposed methods are demonstrated by simulation studies.

1 Introduction

The objective of neural networks-based adaptive control systems for unknown nonlinear plants is to develop algorithms for identification and control using neural networks through a learning process. To avoid modelling difficulties for complex physical systems, the neural method of learning and control provides a natural framework for the design of tracking controllers for unknown nonlinear systems, which can be viewed as the nonlinear dynamic mapping of control inputs into observation outputs. A number of multilayered feedforward neural network-based controllers have been recently proposed [5-21]. For such types of adaptive learning control systems, feedforward neural networks [1-5] are used to approximate the unknown nonlinear static functions that are contained in the nonlinear systems, so that the adaptive control laws can be then designed on the basis of the neural approximation model.

References 9 and 14 have proposed adaptive control approaches using neural networks for discrete-time nonlinear systems with all states observable. They divided the adaptive control problem for an unknown nonlinear system into an identification, or system modelling stage, and a nonlinear control stage. Conversely, References 15 and 16 have proposed and evaluated the Gaussian network-based adaptive controller for a class of continuous-time nonlinear dynamic systems for which an explicit linear parameterisation of the uncertainty in the dynamics is either unknown or impossible to evaluate, and References 20 and 21 have provided a novel algorithm for adaptive tracking for SISO nonlinear systems using layered feedforward neural networks. They have also analysed the convergence of the weight learning and the stability of the closed system. In fact, because of the complexities of nonlinear dynamic systems, currently, the main results of neural networks based nonlinear adaptive control approaches are focused on SISO nonlinear systems with all states observable. Furthermore, the important current research topics in this field are the problems of neural learning and control for general multi-input and multi-output (MIMO) nonlinear systems, which are described by MIMO nonlinear state equations.

More recently, several studies have noted that an appropriate dynamic mapping may be realised by a dynamic recurrent neural network which is trained through a series-parallel or a parallel learning model similar to the case of the feedforward networks, so that a desired response can be obtained [7, 8]. On the other hand, there are potential advantages in using dynamic feedback neural models, like its better prediction capabilities than the static feedforward neural model [17]. The recurrent neural network consists of both feedforward and feedback connections between the layers and neurons forming a complicated dynamic system. Obviously, the ability of a recurrent neural network to approximate a continuous/discrete nonlinear dynamic system by the neural dynamics defined by a system of nonlinear differential/difference equations has the potential for application to adaptive control systems. When dynamic recurrent neural networks are used to approximate and control a unknown nonlinear system through on-line learning processes, they may be treated as subsystems of such adaptive control systems, where the weights of the networks need to be updated using a dynamic learning algorithm during the control processes.

2 Multilayered recurrent neural networks (MRNNs)

An artificial neural network consists of many interconnected identical simple processing units called neurons or nodes. An individual neuron sums its weighted inputs and yields an output through a nonlinear activation function with a threshold. A novel multilayered recurrent neural network (MRNN) architecture is proposed in this section. The MRNN is a hybrid feedforward and feedback neural network, with the feedback represented by the recurrent connections and crosstalk, appropriate for
and \( N_M \) outputs. Let \( M \) be total number of hidden layers of the MRNN, the \( i \)th neuron in the \( s \)th hidden layer be denoted by \( (s, i) \), \( N_s \) be total number of neurons in the \( s \)th hidden layer, and \( u_i \) be the \( i \)th input of the MRNN. \( \alpha_{ij}^s \) be the intra-layer linkweight coefficient from the neuron \((s, j)\) to the neuron \((s, i)\), \( \omega_{ij}^s \) be the feedforward linkweight coefficient from the neuron \((s-1, j)\) to the neuron \((s, i)\), and \( \epsilon_{ij}^s \) be the threshold of the neuron \((s, i)\). Mathematically, the operation of the neuron \((s, i)\) is defined by following dynamic equations.

For the first hidden layer:

\[
x_i(k + 1) = \sigma \left( \sum_{j=1}^{N_{s-1}} \omega_{ij}^{s-1} x_j(k) + \sum_{j=1}^{N_{s-1}} \alpha_{ij}^{s-1} y_j(k) + \sum_{j=1}^{N_{s-1}} \epsilon_{ij}^s + u_i(k) \right)
\]

For the \( s \)th hidden layer:

\[
x_i(k + 1) = \sigma \left( \sum_{j=1}^{N_{s-1}} \omega_{ij}^{s-1} x_j(k) + \sum_{j=1}^{N_{s-1}} \alpha_{ij}^{s-1} y_j(k) + \sum_{j=1}^{N_{s-1}} \epsilon_{ij}^s + u_i(k) \right)
\]

Note that there are no feedback actions from the output layer in the \( s \)th hidden layer; that is, \( \epsilon_{ij}^s = 0 \). If the activation function \( \sigma(\cdot) \) is a symmetric ramp function, the MRNN is then a special type of the brain-state-in-a-box (BSR) model with a nonsymmetric weight matrix. The terms on the right-hand side of above equation represent the feedback from the upper hidden layer, the intra-layer connections, and the feedforward from the lower layer, respectively. Indeed, the output equations of the MRNN are derived as

\[
x(k) = \sum_{i=1}^{N_M} \omega_{km}^{M-1} x_m(k)
\]
Dynamic learning algorithm will be derived in this section. The learning approach was first studied by Williams and Zipser [6], and Narendra and Parthasarathy [9]. A new dynamic learning algorithm is given as:

\[
\underline{v}(k+1) = \underline{v}(k) + \eta \frac{\partial E(k)}{\partial \underline{w}(k)}
\]

where \(\underline{w}(k)\) is an estimation of the weight vector at time \(k\), and \(\eta\) is a step size parameter, which affects the rate of convergence of the weights during learning. The output of the network at current instant \(k\) may be obtained only using the state and input of the network at past time \(k-M\). The error index \(E(k)\) should be then defined as:

\[
E(k) = \frac{1}{2} \sum_{i=1}^{n} [\underline{y}(k) - y(k)]^2 = \frac{1}{2} \sum_{i=1}^{n} e_i(k)^2
\]

where \(e_i(k) = y_i(k) - y_i(k)\) is a learning error between the desired and network outputs at time \(k\). The partial derivatives of the error index \(E(k)\), with respect to the weight of the network, are obtained using the dynamic neural model as follows:

\[
\frac{\partial E(k)}{\partial \underline{w}(k)} = -\sum_{i=1}^{n} e_i(k) \sum_{j=1}^{M} \frac{\partial y_i(k)}{\partial \underline{w}(k)}
\]

where \(\partial y_i(k)/\partial \underline{w}(k)\) is defined as:

\[
\frac{\partial y_i(k)}{\partial \underline{w}(k)} = \sum_{j=1}^{M} \frac{\partial y_i(k)}{\partial \underline{x}(M)} \frac{\partial \underline{x}(M)}{\partial \underline{w}(k)}
\]

and

\[
y_i(k) = h_i(x_i(k), \underline{u}(k), \underline{w})
\]

where \(x_i(k)\) is the state vector of the dynamic hidden neurons of the MRNN, and \(\underline{w}\) is the weight vector which consists of the input layer weights \(\underline{w}_1\), hidden layer weights \(\underline{w}_2\), and threshold \(\underline{w}_3\) of the MRNN. Next, a new input variable or the so-called equivalent control variable \(v \in \mathbb{R}^n\) is defined as:

\[
u(k) = h(x(k), \underline{u}(k), \underline{w})
\]

or

\[
y_i(k) = h_i(x_i(k), \underline{u}(k), \underline{w})
\]

where \(h_i(x_i(k), \underline{u}(k), \underline{w})\) is defined as:

\[
h_i(x_i(k), \underline{u}(k), \underline{w}) = \sum_{j=1}^{M} \frac{\partial y_i(k)}{\partial \underline{x}(M)} \frac{\partial \underline{x}(M)}{\partial \underline{w}(k)}
\]

and

\[
h(x(k), \underline{u}(k), \underline{w})
\]

where \(x\) is the state vector of the dynamic hidden neurons of the MRNN, and \(\underline{w}\) is the weight vector which consists of the input layer weights \(\underline{w}_1\), hidden layer weights \(\underline{w}_2\), and threshold \(\underline{w}_3\) of the MRNN. Next, a new input variable or the so-called equivalent control variable \(v \in \mathbb{R}^n\) is defined as:

\[
u(k) = h(x(k), \underline{u}(k), \underline{w})
\]

or

\[
y_i(k) = h_i(x_i(k), \underline{u}(k), \underline{w})
\]

where \(h_i(x_i(k), \underline{u}(k), \underline{w})\) is defined as:

\[
h_i(x_i(k), \underline{u}(k), \underline{w}) = \sum_{j=1}^{M} \frac{\partial y_i(k)}{\partial \underline{x}(M)} \frac{\partial \underline{x}(M)}{\partial \underline{w}(k)}
\]

and

\[
h(x(k), \underline{u}(k), \underline{w})
\]
to develop nonlinear adaptive control systems for unknown MIMO discrete-time nonlinear systems with on-line identification and control abilities.

Consider a general class of unknown multi-input and multi-output (MIMO) discrete-time nonlinear systems of the form

\[
\begin{align*}
\dot{x}_p(k+1) &= f_p(x_p(k), u(k)) \\
y_p(k) &= h_p(x_p(k))
\end{align*}
\]

where \(x_p \in \mathbb{R}^n\) is an \(n\)-dimensional state vector, \(u \in \mathbb{R}^m\) is an \(m\)-dimensional control vector, and \(y_p \in \mathbb{R}^m\) is a \(m\)-dimensional output vector. The mapping \(f_p\) and function \(h_p\) are assumed to be unknown and analytic. The problem of producing an output, irrespective of the initial state of the unknown nonlinear system, that converges asymptotically to a given reference output \(y_p(k)\) will now be investigated. The reference output is not just a fixed function of time, but is the output of a reference model, which in turn is subject to some input \(r(k)\), described by equations of the form

\[
\begin{align*}
\dot{x}_m(k+1) &= A x_m(k) + B r(k) \\
y_m(k) &= C x_m(k)
\end{align*}
\]

where \(x_m \in \mathbb{R}^n\), \(r \in \mathbb{R}^n\), and \(y_m \in \mathbb{R}^m\) are the state, input, and output of the reference model, respectively, \(A\) is a \(n \times n\) Hurwitz matrix, and \(B\) and \(C\) are \(n \times m\) vectors.

For the unknown nonlinear system (20), the design procedure of the learning control system is divided into the following two steps. Let the neural networks MRNN with the weight \(w\) be first used to approximate the nonlinear plant (20). Then, eqn. 20 may be governed by using the neural networks as follows:

\[
\begin{align*}
\dot{x}(k+1) &= [W_p(k)x(k) + W_k(k)u(k) + w_r(k)] \\
y(k) &= W_p(k)x(k)
\end{align*}
\]

where \(x\) is the state of the recurrent networks MRNN, \(u\) is the input of the neural network MRNN, and \(y\) is the output of the MRNN. The matrices \(W_p(k)\), \(W_k(k)\), and \(w_r(k)\) are, respectively, the estimations of the weight matrices of the hidden, input, and output layers, and \(w_r(k)\) is the estimation of the threshold vector. The expressions of the \(W_p(k)\), \(W_k(k)\), and \(w_r(k)\) are easily implied from earlier discussions in eqns. 1-3.

Next, assume that an nonlinear control law \(u(k)\) is designed based on the equivalent control concept and the dynamic neural system (eqn. 22) such that the output \(y(k)\) of the dynamic neural system (eqn. 22) will track asymptotically the output \(y_p(k)\) of the reference model; that is,

\[
\lim_{k \to \infty} [y_p(k) - y(k)] = 0
\]

On the other hand, for learning control scheme shown in Fig. 3, the error used to train the neural network MRNN is defined as

\[
e^*(k) = y(k) - y_p(k)
\]

where \(y(k)\) and \(y_p(k)\) are the outputs of the neural network and the plant, respectively.

As shown in Fig. 3, note that the error between the outputs of the model and the plant satisfies

\[
|e(k)| = |y_p(k) - y^*(k)| \\
\leq |y(k) - y^*(k)| + |y(k) - y_p(k)|
\]

Indeed, the output \(y(k)\) of the MRNN tracks asymptotically the output \(y^*(k)\) of the model by means of the learning control law \(u(k)\). Hence, if the output of neural network MRNN is trained to approximate the output of plant with \(\lim_{k \to \infty} e(k) = 0\), the output of the plant is then adaptively controlled to track asymptotically the output \(y^*(k)\) of the model; that is, \(\lim_{k \to \infty} e(k) = 0\). In fact, the recurrent network MRNN is used to identify the nonlinear plant on-line, while the control law is constructed based on the identification results of the neural network.

4.2 Design of the equivalent control law

The purpose of the designing equivalent control is to find a new feedback control \(v(k)\) such that the output \(y(k)\) of the MRNN will asymptotically converge to the corres-
Let the equivalent input in eqn. 8 be designed as
\[ y'(k) = c_j A^j x(k) + c_i A^i B(k) \]
\[ + \cdots + c_j A B(k + i - 2) + c_i B(k + i - 1) \]
\[ i = 1, 2, \ldots, r_j - 1 \] (26)
where \( C' = \{ c_1', c_2', \ldots, c_p' \} \). Suppose that the reference model (eqn. 21) has a relative degree \( \{ r_1, \ldots, r_m \} \), and every \( r_j \) is equal to or possibly larger than the corresponding relative degree \( M \) of the neural system (eqn. 22). Then,
\[ c_j B = c_i A B = \cdots = c_j A^{i-1} B = 0 \] (27)
and the relationship between the input and output of the reference model (eqn. 21) may then be represented as
\[ y'(k + i) = c_j A^j x(k) \]
\[ 0 < i < r_j - 1 \quad 1 < j < m \] (28)
and
\[ y'(k + j) = c_j A^j x(k) + c_i A^{i-1} B(k) \]
\[ j = 1, 2, \ldots, m \] (29)
Let the equivalent input in eqn. 8 be designed as
\[ s(k) = \sum_{k=1}^{M-1} \sum_{j=1}^{m} \beta_{j} A^j \left( \sum_{h=0}^{k} y(k+h) - y(k+h) \right) \]
\[ + \sum_{k=0}^{N-1} \sum_{j=1}^{m} \beta_{j} A^j \left( \sum_{h=0}^{k} y(k+h) - y(k+h) \right) \] (30)
The output tracking error equation is then derived by substituting eqn. 30 into eqn. 9 as follows:
\[ s(k + j) + \sum_{h=0}^{M-1} \beta_{j} A^j \left( y(k+h) - y(k+h) \right) = 0 \]
\[ j = 1, 2, \ldots, m \] (31)
where \( s(k) = y'(k) - y(k) \) is the tracking error of the \( j \)th output. If the coefficients \( \beta_{j}, \beta_{j+1}, \ldots, \beta_{j+r_j-1} \) are chosen such that the \( z \)-polynomial
\[ z^j + \beta_{j+1} z^{j+1} + \cdots + \beta_{j+r_j-1} z^{j+r_j-1} = 0 \]
\[ j = 1, 2, \ldots, m \] (32)
has all its zeros inside the unit circle in the complex \( z \)-plane, the output \( y(k) \) of the system will track asymptotically the desired output \( y'(k) \); that is,
\[ \lim_{k \to \infty} y(k) - y'(k) = 0 \] (33)
Since the equivalent input \( r(k) \) depends explicitly, at each time \( k \), on the state \( x(k) \) of the system, on the input \( u(k) \) of the model, and on the state \( x^*(k) \) of the model, which in turn obeys the reference model (eqn. 21), \( u(k) \) can be regarded as the 'output' of a dynamic system of the form
\[ x^*(k + 1) = A x^*(k) + B r(k) \]
\[ u(k) = c_i A x^*(k), x(k), r(k) \] (34)
where the internal state \( x^*(k) \) is driven by the 'input' \( r(k) \) and \( x(k) \). In the following section, we will provide two simulation examples.

5 Simulation studies

Example 1 (SISO nonlinear system)

In this example, the plant is a single-input and single-output system with unknown dynamics described by the nonlinear state difference equation
\[ \begin{align*}
    x_1(k+1) &= x_2(k) \\
    x_2(k+1) &= x_3(k) \\
    x_3(k+1) &= f(x_1(k), x_2(k), x_3(k), u(k)) \\
    x_4(k+1) &= u(k) \\
    y(k) &= x_3(k)
\end{align*} \] (35)
where \( y(k) \) is the uniformly bounded reference input. Since \( 1 + x_1^2 + x_2^2 + x_3^2 \neq 0 \), for any \( x_1 \in \mathbb{R}, x_2 \in \mathbb{R}, \) and \( x_3 \in \mathbb{R} \), the model reference control problem may be solved based on the approaches obtained in the previous section. A three-layered recurrent neural network (MRNN) with a single hidden layer, a single input \( u(k) \), and a single output \( y'(k) \) was used to approximate the unknown nonlinear system (eqn. 35) by using the on-line dynamic learning algorithm. For this purpose, the nonlinear activation function was chosen as \( \alpha(x) = \tanh(x) \), and the number of neurons in the hidden layer of the network was set as \( n_2 = 10 \). The nonlinear algebraic equation (eqn. 8) was solved using the Newton method at each instant \( k \).

Let the reference input \( r(k) \) be a square wave with a circle of 100 steps, and an amplitude of 0.5 and \(-0.5\). The initial values of the weights were chosen randomly in the interval \([-1, 1]\), and the learning rate \( \eta \) was selected to be 0.005. Fig. 4 shows the histories of the outputs of the reference model and the controlled plant, the output tracking error \( e(k) \), and the control input \( u(k) \), respectively. The simulation results in Fig. 4 show that the model reference control of the unknown system (eqn. 35) was performed using the DBP learning algorithm. Although, the initial few steps of the controlled plant oscillated around the reference output \( y'(k) \), the tracking error converged asymptotically to zero after the learning period. In fact, the suitable choice of the iterative initial values of the MRNN is important, even if which can be determined randomly in a fixed range. As mentioned above, the unit interval was used in this simulation study.

Example 2 (MIMO nonlinear system)
The unknown nonlinear system in this case was represented by a nonlinear multivariable plant with two
inputs and two outputs of the form
\[
\begin{align*}
x_i(k+1) &= x_i(k) \\
x_j(k+1) &= g_i[x_i(k), u_i(k)] + \Delta g_i[x_i(k)] \\
x_k(k+1) &= x_k(k) \\
y_i(k) &= g_i[x_i(k), u_i(k)] \\
y_j(k) &= y_i[k] + x_j(k) \\
y_z(k) &= y_i[k] + x_z(k)
\end{align*}
\] (38)
where \( x_i(k) = [x_1(k), x_2(k), x_3(k)] \) and \( y_i(k) = [u_i(k), u_j(k)] \) are the state and control vectors of the system, respectively. The relative degree of the system is \([1, 1]\) if \( \frac{\partial g_i}{\partial u} \neq 0 \) and \( \frac{\partial y_i}{\partial u} \neq 0 \) are satisfied. The specific plant used in the simulation study was
\[
\begin{align*}
g_i(\cdot) &= \frac{x_1 x_4 + u_1 u_2}{1 + x_z^2 + x_2^2} \\
g_j(\cdot) &= \frac{x_4 + u_0}{1 + x_z^2 + x_2^2}
\end{align*}
\] (39)
and \( \Delta g_i[x_i(k)] \) in eqn. 38 was the perturbation term of the plant, the purpose of which was to verify the robustness of the neural network based control system in the presence of structural or parametric variations of the plant. The varying term was then set to
\[
\Delta g_i(\cdot) = \begin{cases} 
0 & \text{if } k \leq 100 \\
0.02 x_1 x_2 (1 + x_2) & \text{if } k > 100
\end{cases}
\] (40)
The stable reference model was given by following MIMO linear system
\[
\begin{align*}
x^{\text{r}}(k+1) &= A x^{\text{r}}(k) + B r(k) \\
y^{\text{r}}(k) &= C x^{\text{r}}(k)
\end{align*}
\] (42)
where \( x^{\text{r}}(k) = [x_1^{\text{r}}(k), x_2^{\text{r}}(k)] \), \( r(k) = [r_1(k), r_2(k)] \), and \( y^{\text{r}}(k) = [y_1^{\text{r}}(k), y_2^{\text{r}}(k)] \) are the state, input and output vectors of the reference model, and
\[
A = \begin{bmatrix} 0.5 & 0.3 \\ 0.2 & -0.7 \end{bmatrix} \quad B = C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}
\] (43)
Note that \( \frac{\partial g_i}{\partial u} \neq 0 \) and \( \frac{\partial y_i}{\partial u} \neq 0 \) since \( 1 + x_z^2 + x_2^2 \neq 0 \) for any \( x_z \in \mathbb{R} \) and \( x_2 \in \mathbb{R} \). The learning controller for the unknown system (eqn. 38) may be therefore designed using the techniques proposed in the previous section. A three-layered network with the two inputs and the two outputs was used to identify the unknown nonlinear system (eqn. 38) on-line by using the DBP learning algorithm. The parameters of the network, the activation function, and the initial values of the learning control process were chosen to be similar to those in Example 1.

Fig. 5 shows the outputs of both the reference model and plant, and the track error under the learning control law for the reference inputs \( r_1(k) = \sin (2\pi k/50) \) and \( r_2(k) = \cos (2\pi k/50) \). For this condition, the simulation results indicate that the outputs of the unknown MIMO plant tracked perfectly the outputs of the reference model by the learning control scheme proposed even though the structure and parameters of the plant varied discontinuously during control process. The oscillation around \( k = 100 \) was due to the varying of the plant structure described by eqn. 41. These simulation studies demonstrate that the MRNNs based learning control system has good robustness for time-varying unknown plants.

6 Conclusions
A multilayered recurrent neural networks (MRNNs) based adaptive control scheme for unknown MIMO discrete-time nonlinear systems has been proposed in this paper. This approach uses the input–output linearisation concept of nonlinear systems and the dynamic weight learning process. As in all adaptive control techniques, the MRNNs-based learning control scheme combines identification and control performed by an on-line adaptively weighted updating process. The ability of the MRNNs with the dynamic learning algorithm to model arbitrary dynamic nonlinear systems was used to approximate the unknown input–output relationship of a non-
linear system, and the control strategy was constructed based on the approximation model. As the unknown nonlinear systems were modelled on-line and controlled by dynamic neural networks, the control mechanisms were less sensitive to the varying of the system parameters and structures, and this phenomenon was demonstrated by simulation results.

A comparison of the dynamic neural network based controllers proposed in this paper and the feedforward network based controllers [9-16] shows that the former needs less a priori knowledge about the unknown plant. The structure of the former is also simpler. Another advantage of the proposed control algorithm is that only the output signal of the plant is fed back to the controller through the neural networks at each instant, because the parallel learning architectures of the MRNNs are utilised during the modeling process. In other words, the dynamic neural networks-based control system is a type of output-feedback adaptive control scheme. Hence, the difficulties of the implementation of the conventional fully-state feedback systems are avoided in such neural control systems.

7 References


