

A hyperstable neural network for the modelling and control of nonlinear systems

K WARWICK^{1*}, Q M ZHU² and Z MA³

¹ Department of Cybernetics, University of Reading, PO Box 225, Whiteknights, Reading RG6 6AY, UK

² Department of Mechanical and Electrical Engineering, Aston University, Aston Triangle, Birmingham B4 7ET, UK

³ Center for Engineering Research, Technikon Natal, PO Box 953, Durban 4000, South Africa

e-mail: kw@cyber.rdg.ac.uk

Abstract. A multivariable hyperstable robust adaptive decoupling control algorithm based on a neural network is presented for the control of nonlinear multivariable coupled systems with unknown parameters and structure. The Popov theorem is used in the design of the controller. The modelling errors, coupling action and other uncertainties of the system are identified on-line by a neural network. The identified results are taken as compensation signals such that the robust adaptive control of nonlinear systems is realised. Simulation results are given.

Keywords. Computer control; neural networks; nonlinear systems; adaptive control.

1. Introduction

In recent years, a number of controllers based on the Lyapunov and Popov hyperstability theorems have been presented, as described in Tomizuka *et al* (1992) and Guonam & Rui (1989). However, such controllers are only suitable for use on linear single input single output (SISO) systems with known parameters and structure. Further, no modelling error is considered in their design and so they are not applicable for many actual real-world systems in which unmodelled dynamics are present, due to robustness problems.

Since the robustness problem for adaptive control was looked at by Rohrs *et al* (1982), many robust adaptive control results have been obtained, e.g. Peterson & Narendra (1982), Ioannou & Kokotovic (1983), Ioannou & Sun (1996) and Praly (1983). Unfortunately, in most cases some unrealistic assumptions and a fair amount of *a priori* knowledge are required in the algorithms, and furthermore, in each case, the performance deteriorates as overall robustness is improved. In particular, with multivariable coupled systems, the coupling action in general cannot be ignored, such that satisfactory control results are often not

*For correspondence

obtainable. Therefore, a subject of considerable significance is to find an effective adaptive control algorithm, which is suitable for nonlinear, multivariable, coupled systems with unknown parameters and structure.

Neural networks have been employed successfully for the control of a wide range of nonlinear systems, e.g. by Garces *et al* (1998), Behera *et al* (1998) and Mazak *et al* (1996). In this paper, a neural network is introduced into the algorithm design, principally because of its numerous advantages. It is known that artificial neural networks can (1) sufficiently approximate a complicated nonlinear mapping; (2) learn and adapt to dynamics of uncertain systems; and (3) have strong robustness and fault-tolerant abilities due to the rich connection and nonlinear activation functions of the neurons. In light of the above advantages a neural network is used to identify the modelling error in a system.

In the design, the nonlinear multivariable coupled system under investigation is described by an equivalent first-order linear uncoupled system along with the unmodelled dynamics. The modelling error between the equivalent linear system and the actual system is then handled as unmodelled dynamics.

In the algorithm, the results of the modelling error identified by the neural network are taken as compensation signals in the feedforward compensator, the aim being to eliminate the effect of the modelling error, thereby improving the robustness of the overall system. Furthermore, coupling action, which is not directly involved in the equivalent model, is also dealt with. The most important result, however, is that the system performance is not appreciably degraded, as robustness is improved. The algorithm design is based on the Popov hyperstability theorem, so that the adaptive law is chosen rationally according to the system parameters and the Popov inequality. Since the controller is based on a first-order model, there are only two (unknown) parameters, which means that the algorithm is relatively simple. In addition, *a priori* knowledge about the system under control is not needed.

In the next section the overall controller structure is introduced. This is followed by a description of the neural network modelling procedure and subsequently the overall adaptive control algorithm. The algorithm which is described is now open to comparison with other neural network based adaptive controllers (for example, Narendra 1991, Chen & Liu 1994, Gupta & Sinha 1996, Zhu *et al* 1999).

2. Controller design

It is assumed that the plant in question can be described by the following model

$$Y(t) = f_s(\cdot), \quad (1)$$

where f_s is a complicated nonlinear reflection vector, and is a smooth (i.e. infinitely differentiable) function of

$$[Y(t-1), \dots, Y(t-n_y); U(t-d), \dots, U(t-n_u)],$$

in which $Y(t)$ and $U(t)$ denote the output and the input signal vectors respectively. Due to f_s being infinitely differentiable, and assuming for simplicity that $d = 1$, the system (1) may be parametrised as

$$Y(t) = AY(t-1) + BU(t-1) + f_0(\cdot), \quad (2)$$

where an assumption is made that B is non-zero. A and B are unknown diagonal parameter matrices such that

$$A = \text{diag}(a^{ii}); \quad i = 1, \dots, n$$

$$B = \text{diag}(b^{ii})$$

and $f_0(\cdot)$ is the nonlinear part of the equivalent model, possibly it is a nonlinear vector function of $[Y(t-1), \dots, Y(t-n_y), U(t-1), \dots, U(t-n_u)]$. It denotes the modelling error between the equivalent linear parts and the actual nonlinear plant. In order to obtain a controller, a difference operator $\Delta = 1 - z^{-1}$ is introduced into the system (2), such that equation (2) may be described alternatively by

$$Y(t) = AY(t-1) + BU(t) - B\Delta U(t) + f_0(\cdot), \quad (3)$$

in which

$$\Delta U(t) = U(t) - U(t-1), \quad (4)$$

then by defining

$$B\Delta U(t) = B'\Delta U(t-1), \quad (5)$$

this means that a change between $\Delta U(t)$ and $\Delta U(t-1)$ is included in the coefficient B' , i.e. B is essentially a version of B' operated on by a unit delay. Equation (3) therefore becomes

$$Y(t) = AY(t-1) + BU(t) - B'\Delta U(t-1) + f_0(\cdot). \quad (6)$$

The tracking error can be defined by

$$E(t) = R(t) - Y(t), \quad (7)$$

where $R(t)$ is a reference input vector. The controller consists of a feedback controller component and a feedforward compensator component. The overall output vector of the controller is then given by

$$U(t) = U_e(t) + U_f(t), \quad (8)$$

where

$$U_e(t) = K_p E(t), \quad (9)$$

and

$$U_f(t) = K_1 R(t) + K_2 R(t-1) + K_3 \Delta U(t-1) + K_4 f_0(\cdot), \quad (10)$$

where K_1, K_2, K_3, K_4 and K_p are diagonal matrices such that

$$K_1 = \text{diag}(k_1^{ii}); \quad K_2 = \text{diag}(k_2^{ii}); \quad K_3 = \text{diag}(k_3^{ii});$$

$$K_4 = \text{diag}(k_4^{ii}); \quad K_p = \text{diag}(k_p^{ii}); \quad i = 1, \dots, n.$$

The reason for this particular controller selection will become apparent as the exposition progresses.

Since back propagation networks have good nonlinear modelling properties (Hunt *et al* 1992, 1995; Karny *et al* 1998), one such is used to detect the system modelling error on-line. The network output is taken as one of the feedforward signals of the control law. By

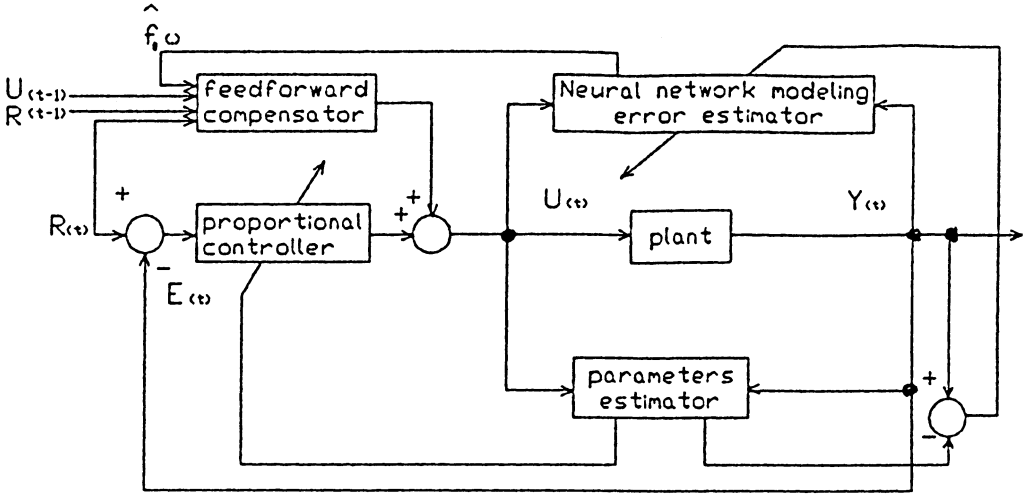


Figure 1. System structure.

means of the signals used, the effect of uncertainties may be effectively overcome. A diagram of the controller is given in figure 1.

By substituting (7)–(10) into (6), we have

$$\begin{aligned} (I + BK_p(t))E(t) &= AE(t - 1) + (I - BK_1(t))R(t) - (A + BK_2(t))R(t - 1) \\ &= (B' - B'K_3(t))\Delta U(t - 1) - (I + BK_4(t))f_0(\cdot). \end{aligned} \tag{11}$$

Let us then define

$$\begin{aligned} S_1(t) &= (I - BK_1(t))R(t) - (A + BK_2(t))R(t - 1) \\ &\quad + (B' - BK_3(t))\Delta U(t - 1) - (I + BK_4(t))f_0(\cdot), \end{aligned} \tag{12}$$

such that

$$(I + BK_p(t))E(t) = AE(t - 1) + S_1(t). \tag{13}$$

Equation (13) can then be represented by the equivalent feedback form shown in figure 2.

According to the Popov hyperstability criterion, if both the following conditions are satisfied, the system in figure 2 will be asymptotically hyperstable.

- (1) The transfer function of the linear block in figure 2 is strictly positive real (SPR).
- (2) The following Popov inequality for the nonlinear block in figure 2 denoted by NL, is satisfied.

$$\eta(0, t_1) = \sum_{t=0}^{t_1} S(t)E(t) \geq \gamma^2, \tag{14}$$

in which

$$S(t) = -S_1(t). \tag{15}$$

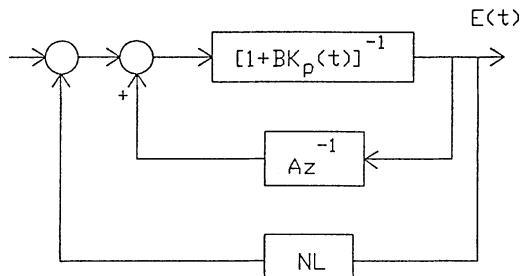


Figure 2. Equivalent construction diagram.

From figure 2, one has

$$H(z^{-1}) = ((I + BK_p(t)) - Az^{-1})^{-1}, \tag{16}$$

and by using (16), this transfer function is SPR as long as

$$k_p^{ii} > \frac{a^{ii} - 1}{b^{ii}}; \quad i, \dots, n. \tag{17}$$

In the system, the parameter matrices A and B are considered to be unknown, although they may be identified on-line by a recursive algorithm.

A suitable adaptive control law to obtain the parameters for (8) can be written as:

$$k_1^{ii} = \sum_{j=0}^t l^{ii} \lambda_{1i} r_i(j) e_i(j); \quad i = 1, \dots, n, \tag{18}$$

$$k_2^{ii} = \sum_{j=0}^t l^{ii} \lambda_{2i} r_i(j-1) e_i(j); \quad i = 1, \dots, n, \tag{19}$$

$$k_3^{ii} = \sum_{j=0}^t l^{ii} \lambda_{3i} \Delta u_i(j-1) e_i(j); \quad i = 1, \dots, n, \tag{20}$$

$$k_4^{ii} = \sum_{j=0}^t l^{ii} \lambda_{4i} \hat{f}_{0i}^j(\cdot, W) e_i(j); \quad i = 1, \dots, n, \tag{21}$$

where λ_{1i} , λ_{2i} , λ_{3i} and λ_{4i} are arbitrary real constants. Also l can be chosen to be

$$l^{ii} = 1, \quad \text{if } \hat{b}^{ii} > 0;$$

and

$$l^{ii} = -1, \quad \text{if } \hat{b}^{ii} < 0, \tag{22}$$

in which \hat{b}^{ii} is the estimate of b^{ii} . Also $r_i(j)$ are the constituent parameters of $R(j)$, and $e_i(j)$ of $E(j)$; both $R(j)$ and $E(j)$ containing n such parameters.

Substituting (12) and (14) into (15), and using (18)–(21) and the following relationship (Guonam & Rui 1989),

$$\sum_{k=0}^{t1} \left[\sum_{i=0}^t x(i) + c \right] x(k) = \frac{1}{2} \left[\sum_{k=0}^{t1} x(k) + c \right]^2 + \frac{1}{2} \sum_{k=0}^{t1} x^2(k) - \frac{c^2}{2} \geq -\frac{c^2}{2}, \tag{23}$$

this becomes

$$\begin{aligned}
 \eta^i(0, t_1) &= \sum_{k=0}^{t_1} S_i(k) e_i(k) \\
 &= - \sum_{k=0}^{t_1} [1 - b^{ii} k_1^{ii}(k)] r_i(k) e_i(k) + \sum_{k=0}^{t_1} [a^{ii} - b^{ii} k_2^{ii}(k)] r_i(k-1) e_i(k) \\
 &\quad - \sum_{k=0}^{t_1} [b^{i'ii} - b^{ii} k_3^{ii}(k)] \Delta u_i(k-1) e_i(k) + \sum_{k=0}^{t_1} [1 + b^{ii} k_4^{ii}(k)] f_{0i}^k(\cdot, W) e_i(k) \\
 &= \lambda_{1i} l^{ii} b^{ii} \sum_{k=0}^{t_1} \left[\sum_{j=0}^t r_i(j) e_i(j) - \frac{l^{ii}}{\lambda_{1i} b^{ii}} \right] r_i(k) e_i(k) \\
 &\quad + \lambda_{2i} l^{ii} b^{ii} \sum_{k=0}^{t_1} \left[\sum_{j=0}^t r_i(j-1) e_i(j) + \frac{a^{ii} l^{ii}}{\lambda_{2i} b^{ii}} \right] r_i(k-1) e_i(k) \\
 &\quad + \lambda_{3i} l^{ii} b^{ii} \sum_{k=0}^{t_1} \left[\sum_{j=0}^t \Delta u_i(j-1) e_i(j) - \frac{b^{i'ii} l^{ii}}{\lambda_{3i} b^{ii}} \right] \Delta u_i(k-1) e_i(k) \\
 &\quad + \lambda_{4i} l^{ii} b^{ii} \sum_{k=0}^{t_1} \left[\sum_{j=0}^t \hat{f}_i^j(\cdot, W) e_i(j) + \frac{l^{ii}}{\lambda_{4i} b^{ii}} \right] \hat{f}_{0i}^k(\cdot, W) e_i(k) \geq -\gamma_{i0}^2,
 \end{aligned} \tag{24}$$

where

$$\gamma_{i0}^2 = \frac{l^{ii}}{2b^{ii}} \left(\frac{1}{\lambda_{1i}} + \frac{a^{ii}}{\lambda_{2i}} + \frac{b^{i'ii}}{\lambda_{3i}} + \frac{1}{\lambda_{4i}} \right). \tag{25}$$

If the proportional gain k_p^{ii} and adaptive control parameters k_1^{ii} , k_2^{ii} , k_3^{ii} and k_4^{ii} are chosen by means of the equation set (17)–(22), then the system in figure 2 will be asymptotically hyperstable.

It is worth commenting here that \hat{b}^{ii} , the estimate of b^{ii} , must be biased away from zero to ensure that the hyperstability results follow. In order that (17) is satisfied in a realistic way, with the gain not tending to infinity, the estimate of b^{ii} must be kept well away from zero in practice. A simple dead-zone algorithm is, in fact, enough to ensure that this occurs.

Equation (17) is quite telling insofar as it indicates that when b^{ii} tends to zero on an actual system, the control structure may well result in feedback with relatively high gain. In a practical sense such a situation could affect the robustness of the overall algorithm, particularly in the presence of unmodelled dynamics such as time delays. It is felt, therefore, that the algorithm will operate, in practice, much more effectively on systems in which b^{ii} is relatively large in magnitude.

3. Neural network estimator

A back propagation network, as shown in figure 3, is used to estimate the modelling error.

The network has three layers. The input layer has l neurons, the hidden layer m sigmoidal neurons and the output layer has n sigmoidal neurons. The input of the back propagation

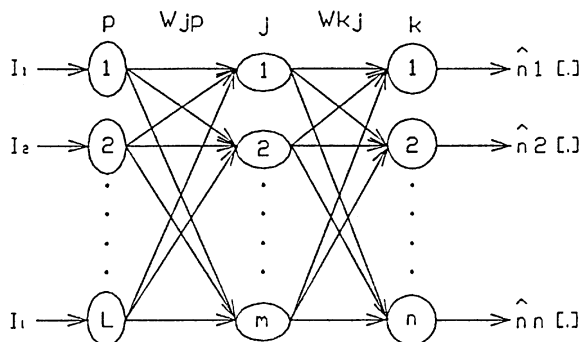


Figure 3. Back propagation network.

network is given by

$$I_p = [Y(t)^T, Y(t-1)^T, U(t-1)^T, \dots, U(t-2)^T, \dots; \hat{Y}(t), \hat{Y}(t-1), \dots;], \quad (26)$$

where

$$\hat{Y}(t) = \hat{A}Y(t-1) + \hat{B}U(t-1), \quad (27)$$

and the signals used for teaching the network are the modelling errors,

$$f_{kd} = Y(t) - \hat{Y}(t). \quad (28)$$

Also the total input of the hidden layer is given by

$$\text{Net}_j = \sum_{p=1}^L W_{jp} I_p. \quad (29)$$

Further, a sigmoidal nonlinearity is applied to each unit to obtain the hidden layer outputs; this takes the form:

$$0_j = f(\text{Net}_j) = \frac{1}{1 + \exp(-\text{Net}_j - \beta_j)}. \quad (30)$$

The input of the output layer is given by

$$\text{Net}_k = \sum_{j=1}^m W_{kj} 0_j, \quad (31)$$

whereas the overall network output can be obtained from

$$\hat{f}_0(\cdot, W) = f(\text{Net}_k) = \frac{1}{1 + \exp(-\text{Net}_k - \beta_k)}, \quad (32)$$

in which $j = 1, \dots, m$; $k = 1, \dots, n$; $p = 1, \dots, L$. W_{jp} and β_j are the weights and the thresholds respectively between the input and the hidden layer, whereas W_{kj} and β_k are the weights and the thresholds between the hidden and output layer. Each weight and threshold

is updated by means of the set of equations described as follows:

$$W_{jp}(t+1) = W_{jp}(t) + \eta_{jp}\delta_k O_p(t), \quad (33)$$

$$W_{kj}(t+1) = W_{kj}(t) + \eta_{kj}\delta_j I_k(t), \quad (34)$$

$$\beta_j(t+1) = \beta_j(t) + \rho_j\delta_j, \quad (35)$$

$$\beta_k(t+1) = \beta_k(t) + \rho_k\delta_k, \quad (36)$$

where η_{jp} , η_{kj} , ρ_j and ρ_k are the learning rates of the corresponding weights and thresholds, which can be selected independently. The training error of each unit is given by

$$\text{output layer: } \delta_k = \hat{f}_0(\cdot, W)(1 - \hat{f}_0(\cdot, W))(f_{kd} - \hat{f}_0(\cdot, W)); \quad (37)$$

$$\text{hidden layer: } \delta_j = O_j(1 - O_j) \sum_{k=1}^n \delta_k W_{kj}. \quad (38)$$

4. Adaptive control algorithm

Directly from (2), the plant model can be rewritten in the form of n separate equations:

$$y_i(t) = x_i(t-1)^T \theta_i + f_{0i}(\cdot), \quad (39)$$

where

$$x_i(t-1) = [y_i(t-1) u_i(t-1)], \quad (40)$$

and

$$\theta_i = [a^{ii} \cdot b^{ii}], \quad (41)$$

for

$$i = 1, \dots, n.$$

The robust adaptive control algorithm can then be expressed in terms of the following steps:

- (1) measure the system outputs $y_i(t)$, establish the data vector $x_i(t-1)$ and the input vector to the back propagation network, $I_p(t)$;
- (2) the following recursive algorithm is used to estimate the parameters a^{ii} and b^{ii} .

$$\hat{\theta}_i(t+1) = \hat{\theta}_i(t) + \frac{P_i(t)x_i(t)}{1 + x_i(t)^T P_i(t)x_i(t)} \nu_i^0(t+1), \quad (42)$$

$$P_i(t) = \frac{P_i(t)}{\beta_{i1}(t)} - \frac{P_i(t)x_i(t)^T x_i(t) P_i(t) \beta_{i2}(t)}{\beta_{i1}(t) + \beta_{i2}(t)x_i(t)^T P_i(t)x_i(t)}, \quad (43)$$

in which $0 < \beta_{i1}(t) \leq 1$; $0 \leq \beta_{i2}(t) \leq 2$. Common practice though is $\beta_{i2}(t) = 1$ for all time and typically $\beta_{i1}(t) \approx 0.97$ for all time, although varying factors are possible (Fortescue *et al* 1981),

$$\nu_i^0(t+1) = H_s(z^{-1})[y_i(t+1) - x_i(t)^T \hat{\theta}_i(t) - \hat{f}_{0i}(\cdot, W)], \quad (44)$$

where $H_s(z^{-1})$ is a convergent filter, $H_s(z^{-1}) = [H_{s1}(z^{-1})]/[H_{s2}(z^{-1})]$, and both $H_{s1}(z^{-1})$ and $H_{s2}(z^{-1})$ are stable polynomials with $H_{s1}(1) = 1$; $H_{s2}(0) = 1$. Also $\hat{f}_{0i}(\cdot, W)$ is the

output of the back propagation network. The next teaching signal for the back propagation network is then generated from

$$f_{id} = y(t+1) - x(t+1)^T \hat{\theta}_i(t+1), \quad (45)$$

- (3) using the back propagation network, the next step modelling error $\hat{f}_0(\cdot)$ is generated. The network is trained S times at each sampling period;
- (4) the tracking errors are calculated. These are dependent on the system reference input and the system output;
- (5) the gain $K_p(t)$ is calculated and is dependent on the parameter estimates \hat{A} and \hat{B} ;
- (6) from (18)–(22) $K_1(t)$, $K_2(t)$, $K_3(t)$ and $K_4(t)$ are obtained;
- (7) the control input $U(t)$ is calculated from (8)–(10);
- (8) the above steps are repeated at each sampling interval, along with an update of the information vectors.

Remark. Here, the recursive algorithm based on (42) and (43) is used to provide convergence of the identified parameters in the case of the nonlinear modelling error being introduced into the algorithm. A full mathematical proof of convergence is felt to be beyond the scope of this paper.

5. Simulation results

In order to show how the algorithm described in § 4 can be applied, the following nonlinear two input–two output multivariable system is considered.

$$y_1(t) = -\frac{-0.5y_1(t-1)^2}{1 + y_1(t-1)y_2(t-1) + y_1(t-2)y_2(t-2)} + 0.2y_2(t-2) + 0.5u_1(t-1),$$

$$y_2(t) = -\frac{-0.27y_2(t-1)^2}{1 + y_1(t-1)y_2(t-1) + y_1(t-2)y_2(t-2)} + 0.1y_1(t-2) + 0.38u_2(t-1).$$

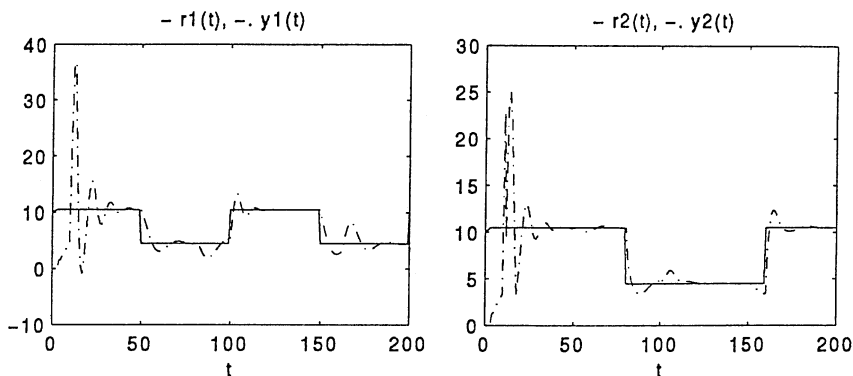


Figure 4. System tracking trajectories.

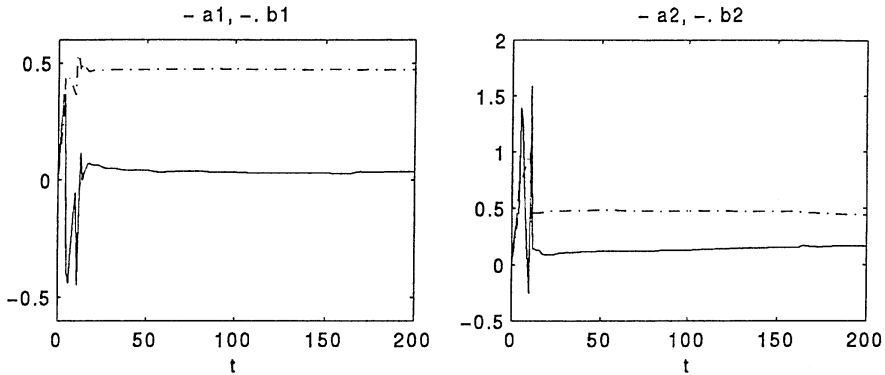


Figure 5. Identified parameter curves.

In the experiment, the following choices are made: $L = 8$, $m = 9$, $I = 2$. Also the learning rate $\eta = [0.0025 \ 0.0025]$; along with $\lambda_1 = [0.0083 \ 0.0083]$; $\lambda_2 = [0.00011 \ 0.00011]$; $\lambda_3 = [0.0001 \ 0.0001]$; $\lambda_4 = [0.0006666 \ 0.0006666]$. The back propagation network is trained twice during each sampling interval.

The reference input $W(t)$ and system output $Y(t)$ are shown in figure 4 and it can be seen that the control results of the proposed algorithm are satisfactory. In spite of applying the control algorithm based on a first order linear uncoupled model to a nonlinear high order coupled plant, stable control results are still obtained. The estimated values of \hat{A} and \hat{B} are shown in figure 5, and control input curves are shown in figure 6.

6. Conclusions

The hyperstable adaptive neural network, decoupling control algorithm introduced in this paper exhibits the following characteristics:

- (1) it is applicable to complicated nonlinear systems;
- (2) it implements robust adaptive decoupling control in the case of existing unmodelled dynamics;
- (3) since the algorithm is based on a first order linear model, it is simple enough such that real-time control can be realised;

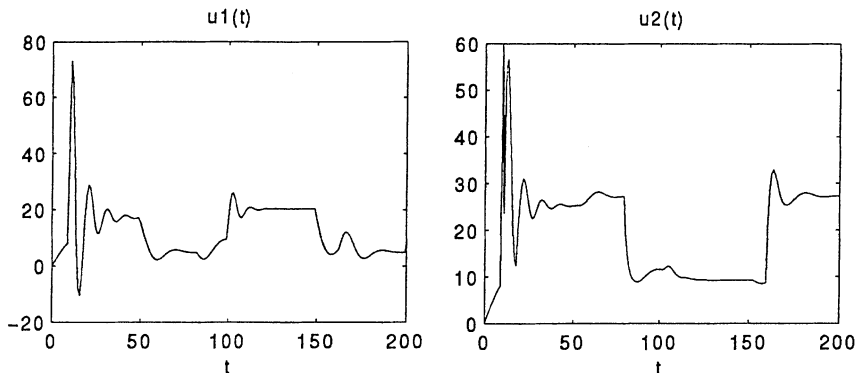


Figure 6. Controller inputs.

- (4) since the Popov hyperstability principle is used in the design of the adaptive control law, the final algorithm can overcome the problem of general self-tuning control algorithms, which is that solving ill-conditioned equations can cause the controller parameters to force the system out of control.

Although hyperstability of the adaptive control law has been indicated, no formal proof of stability is included, which is a feature in common with most self-tuning control algorithms (e.g. Harris & Billings 1985). It is well known (e.g. Ioannou & Sun 1996) that, in a practical sense, robustness modifications to an adaptive control algorithm of this kind may be required in order to ensure satisfactory operation on a specific system. Whilst this point must be borne in mind, fine tuning of the algorithm to improve its performance would appear to be a sensible step to allow.

One final point is that the system selected for the simulation test was not particularly one on which the algorithm was expected to operate well. Although the performance does appear to improve with time, as one would expect of an adaptive algorithm, some large transients are apparent in the early stages whilst the controller is “bedding in”. It is felt that showing the algorithm operating successfully on such a (non-favoured) system, is a respectable indication of its potential. Further trials are now being made both to test its range of successful operation and to compare its performance with other related algorithms.

References

- Behera L, Chaudhury S, Gopal M 1998 Application of self-organizing neural networks in robot tracking control. *IEE Proc. Control Theor. Appl.* 145: 135–140
- Chen F C, Liu C C 1994 Adaptively controlling nonlinear continuous time systems using multilayer neural networks. *IEEE Trans. Autom. Control* 39: 1306–1310
- Fortescue T, Kershenbaum L S, Ydstie B E 1981 Implementation of self-tuning regulators with variable forgetting factors. *Automatica* 17: 831–835
- Garces F, Warwick K, Craddock C 1998 Multiple PID mapping using neural networks in a MIMO generator system. *Proc. Int. Conf. Control* 98, Swansea, pp 503–508
- Guonam F, Rui Y 1989 A microcomputer controller optimal adaptive flight simulator servo system. *Acta Autom. Sin.* 15: 193–199
- Gupta M M, Sinha N K (eds) 1996 *Intelligent control systems* (New York: IEEE Press)
- Harris C J, Billings S A 1985 *Self-tuning control: Theory and applications* rev. 2nd edn (London: Peter Peregrinus)
- Hunt K J, Sbarbaro D, Zbikowski R, Gawthrop P J 1992 Neural networks for control systems – A survey. *Automatica* 28: 1083–1112
- Hunt K J, Irwin G W, Warwick K (eds) 1995 *Neural network engineering in dynamic control systems* (Berlin: Springer-Verlag).
- Ioannou P A, Kokotovic P V 1983 *Adaptive systems with reduced models* (Berlin: Springer-Verlag)
- Ioannou P A, Sun J 1996 *Robust adaptive control* (Englewood Cliffs, NJ: Prentice Hall)
- Karny M, Warwick K, Kurkova V (eds) 1998 *Dealing with complexity: a neural network approach* (New York: Springer-Verlag)
- Mazak J, Damen A A H, Backx A C P M, Weiland S 1996 Nonlinear transition controller using neural networks tested on a polymerization reactor. *Proc. 35th IEEE Conference on Decision and Control*, Kobe, Japan, pp 1720–1721
- Narendra K S 1991 Adaptive control using neural networks. In *Neural networks for control* (eds) W T Miller, R S Sutton, P W Werbos (Cambridge, MA: MIT Press) pp 115–142
- Peterson B B, Narendra K S 1982 Bounded adaptive control. *IEEE Trans. Autom. Control* AC-27: 1161–1168

- Praly L 1983 Robustness of model reference adaptive control. *Proc. 3rd Yale Workshop on the Applications of Adaptive Systems Theory*, Yale University, New Haven, CT
- Rohrs C E, Valavani L, Athans M, Stein G 1982 Robustness of adaptive control algorithms in the presence of unmodelled dynamics. *Proc. 21st IEEE Conf. Decision & Control*, Orlando, Florida, pp 3–11
- Tomizuka M, Hu J-S, Chin T-C 1992 Synchronization of two motion control axes under adaptive feedforward control. *Trans. ASME* 114: 196–203
- Zhu Q M, Ma Z, Warwick K 1999 A neural network enhanced generalized minimum variance self-tuning controller for nonlinear discrete-time systems. *IEE Proc. Control Theor. Appl.* 146: 319–326