Proceedings of the 7th WSEAS International Conference on Automation & Information, Cavtat, Croatia, June 13-15, 2006 (pp173-178)

Neural Based Adaptive Control of a Class of Dynamical Nonlinear Processes

EMIL PETRE, DAN SELIŞTEANU, DORIN ŞENDRESCU
Department of Automatic Control
University of Craiova
A.I. Cuza Str. No. 13, RO-200585 Craiova
ROMANIA
http://www.automation.ucv.ro

Abstract: - A nonlinear adaptive controller for a class of nonlinear plants with incompletely known and time varying dynamics is presented. It is based on a recurrent neural network used as a dynamical model of the plant. The adaptive controller design is realized by using an input-output feedback linearizing technique. The model parameters, that is the controller parameters are updated on-line such that the behaviour of closed loop system is closely to those of a linear system. A local convergence of the algorithm is provided for the case of constant reference output. Computer simulations are included to illustrate the performances of the proposed controller.

Key-Words: - Nonlinear systems, Nonlinear control, Adaptive control, Neural networks.

1 Introduction

In recent years, there has been considerable research activity in the applications of neural networks (NN) to identification and control of nonlinear systems [1], [2], [6], [7], [9]. It is well known that the NNs can be considered as general tools for modelling nonlinear functions [3].

It must be noted that in most of the above results, the role of neural networks is usually a model that can mimic a nonlinear input-output relation. Another main advantage of using a NN in a control application is that it can dynamically store complicated nonlinear control algorithms and recall them instantly where demanded. Furthermore, the learning capability of the NN enables the resulting controller to adapt itself to possible variations in the controlled plant dynamics while in operation.

In this paper, recurrent neural networks are used as models of the unknown plant, practically transforming the originally unknown system, to a dynamic neural network model. The structure of this model is known but it contains a number of unknown constant parameters namely the network weights. The ability of recurrent neural networks to learn static and dynamic highly nonlinear systems is a well-known property [4], [10]. Mainly, this is a reason why the recurrent neural networks have concentrated many research activities, especially in the area of identification and more recently in control. However, when one uses models to develop control algorithms, the presence of a modelling error term, which is unavoidable, could destroy the stability of the system. As a result of these considerations, in this paper will be presented some aspects of the stability properties of the closed loop system. Moreover, the convergence properties of the determined control algorithm are provided in the case when the output reference has a constant value.

The paper is organized as follows: in Section 2, it is stated the problem and the form of the recurrent neural network model, while in Section 3 the adaptive control algorithm and update laws are developed. An application to a real process is presented in Section 4. Section 5 concludes the paper.

2 Problem Statement

Consider the multi-input/multi-output *square* non-linear dynamical systems (that is systems with as many inputs as outputs) of the form:

$$\dot{x} = f(x) + \sum_{i=1}^{n} g_i(x)u_i = f(x) + G(x)u \qquad (1)$$
$$y = Cx$$

with the state $x \in \Re^n$, the input $u \in \Re^n$ and the output $y \in \Re^n$. $f : \Re^n \to \Re^n$ is an unknown smooth vectorfield called the drift term and $G = [g_1 \ g_2 \ldots g_n]$ is a matrix whose columns are the unknown smooth vectorfields $g_i : \Re^n \to \Re^n$, $i = 1,...,n$. The relative degree of (1) is equal to 1 and the states x are assumed available. C is a $n \times n$ constant matrix. Thus, particularly, in the second equation in (1) C can be equal to identity matrix.

Note that the functions f and g_i ($i = 1,...,n$) contain parametric uncertainties which are not necessarily linear parameterizable.

In this paper we deal with the control problem of the processes described by the model (1). The *control objective* is to make output y of the system (1) tracks a specified trajectory denoted $y_{ref} \in \Re^n$. However, the problem, as it is stated above for the process (1), is very difficult or even impossible to be solved since the vectorfields f and g_i ($i = 1,...,n$) are assumed to be completely unknown. Therefore, in order to provide a solution to our problem, it is necessary to have a more accurate model for the unknown plant. For that purpose, in order to model the nonlinear system (1) we use dynamical neural networks.

Dynamical neural networks are recurrent, fully interconnected nets, containing dynamical elements in their neurons. They can be described by the following system of coupled first-order differential equations:

$$\dot{\hat{x}}_i = a_i \hat{x}_i + b_i \sum_{j=1}^{n} w_{ij} \phi(x_i) + b_i w_{i,n+1} \psi(x_i) u_i ,$$
$$i = 1,...,n \qquad (2)$$

or in a compact form

$$\begin{aligned} \dot{\hat{x}} &= A\hat{x} + BW\Phi(x) + BW_{n+1}\Psi(x)u \\ y_N &= C\hat{x} \end{aligned} \qquad (3)$$

where the state $\hat{x} \in \Re^n$, the input $u \in \Re^n$, W is a $n \times n$ matrix of adjustable synaptic weights, A in a $n \times n$ diagonal matrix with negative eigenvalues, B is a $n \times n$ diagonal matrix with elements the scalars b_i, $i = 1,...,n$ and W_{n+1} is a $n \times n$ diagonal matrix of adjustable synaptic weights, of the form $W_{n+1} = diag\{w_{1,n+1} \cdots w_{n,n=1}\}$. $\Phi(x)$ is a n-dimensional vector and $\Psi(x)$ is a $n \times n$ diagonal matrix, with elements the activation functions $\phi(x_i)$ and $\psi(x_i)$ respectively, both smooth, monotone increasing functions, usually represented by sigmoids of the form:

$$\phi(x_i) = \frac{m}{1 + e^{-\delta_1 x_i}}, \quad \psi(x_i) = \frac{m}{1 + e^{-\delta_2 x_i}} + \theta, \; i = 1,...,n$$

where m and $\delta_k, k = 1,2$ are constants representing the bound and the slope of sigmoid's curvature respectively, and $\theta > 0$ is a constant that shifts the sigmoid, such that $\psi(x_i) > 0$ for all $i = 1,...,n$.

A bloc diagram of this dynamical neural network is shown in Fig.1.

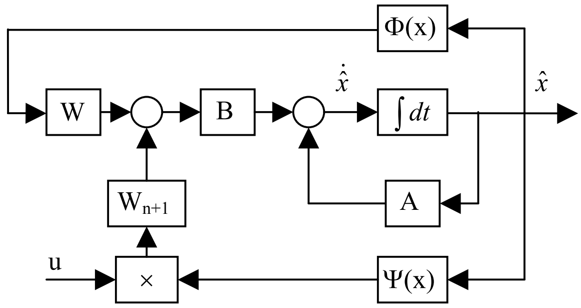

Fig.1. The bloc diagram of dynamical neural network

3 Control Strategies

In this section, by using the feedback linearization techniques, we present two nonlinear controllers for the system (1): a nonlinear inverse dynamic controller and an adaptive controller using recurrent neural networks.

3.1 Nonlinear inverse dynamic controller

Firstly, we consider an idealized case where maximum prior knowledge concerning the process is available. In particular, we suppose that the functions $f(\cdot)$ and $G(\cdot)$ in process model (1) are completely known and all state variables are available for on-line measurements.

Assume now that we wish to have the following first order linear stable closed loop (process + controller) dynamical behaviour:

$$\frac{d}{dt}(y_{ref} - y) + \Lambda(y_{ref} - y) = 0 \qquad (4)$$

with $\Lambda = diag\{\lambda_i\}, \lambda_i > 0, i = 1,...,n$

Then, by combining the equations (1) and (4) we obtain the following multivariable decoupling linearizing feedback control law

$$u = \left(CG(x)\right)^{-1}\left(-Cf(x) + \nu\right) \qquad (5)$$

with $\left(CG(x)\right)^{-1}$ assumed invertible, which applied to the process (1) result in

$$\dot{y} = \nu \qquad (6)$$

where ν is the new input vector designed as

$$\nu = \dot{y}_{ref} + \Lambda(y_{ref} - y) \qquad (7)$$

The control law (5) leads to the following linear error model:

$$\dot{e}_t = -\Lambda e_t \qquad (8)$$

where $e_t = y_{ref} - y$ represent the tracking error. It is clear that for $\lambda_i > 0, i = 1,...,n$, the error model (8) has an exponential stable point at $e = 0$.

3.2 Neural network adaptive controller

Because the prior knowledge concerning the process assumed in the previous subsection is not realistic, in this subsection we analyze a more realistic case, where the dynamical process (1) is unknown, that is the functions $f(\cdot)$ and $G(\cdot)$ are completely unknown and time varying. To solve the control problem, a recurrent neural network (3) is used as a dynamic model of the process based on which control law is synthesized.

Assume that the unknown process (1) can be completely described by a dynamical neural network. In other words, there exist weight values W^* and W^*_{n+1} such that the process (1) can be written as:

$$\begin{aligned} \dot{x} &= Ax + BW^*\Phi(x) + BW^*_{n+1}\Psi(x)u \\ y &= Cx \end{aligned} \tag{9}$$

where all matrices are as defined previously.

Now the tracking problem is analyzed for the system (9) instead of (1). Since W^* and W^*_{n+1} are unknown, our solution consists in designing a control law $u(W, W_{n+1}, x)$ and appropriate update laws for W and W_{n+1} such that the network model output y track a reference trajectory y_{ref}. The evolution of the network model output (9) can be expressed as:

$$\dot{y} = C\dot{x} = CAx + CBW^*\Phi(x) + CBW^*_{n+1}\Psi(x)u \tag{10}$$

Assume that $CBW^*_{n+1}\Psi(x)$ is invertible which implies relative degree equal one for input-output relation (10). Then, the following linearizing control u is given by

$$u = \left(CBW^*_{n+1}\Psi(x)\right)^{-1}\left(-CAx - CBW^*\Phi(\xi) + \nu\right) \tag{11}$$

where the new input vector ν is defined as

$$\nu = \dot{y}_{ref} + \Lambda(y_{ref} - y) \tag{12}$$

which applied to the network (9) results in it being decoupled and linear with respect to this new input.

$$\dot{y} = \nu \tag{13}$$

Defining the error between the network output and the reference trajectories as

$$e_t = y - y_{ref} \tag{14}$$

then the control law (12) leads to the following error model:

$$\dot{e}_t = -\Lambda e_t \tag{15}$$

It is clear that for $\lambda_i > 0, i = 1,...,n$, the error e_t converges to the origin exponentially.

Note that the control-input (11) is applied to both plant and neural model.

Now, we can define the modelling error between the neural network output and real system output as

$$e_m = y_N - y \tag{16}$$

Then, from equations (9) and (3) we obtain the following error equation:

$$\dot{e}_m = CAe_m + CB\widetilde{W}\Phi(x) + CB\widetilde{W}_{n+1}\Psi(x)u \tag{17}$$

where

$$\widetilde{W} = W - W^*,\ \widetilde{W}_{n+1} = W_{n+1} - W^*_{n+1} \tag{18}$$

Because the control law (11) contains the unknown weight matrices W and W_{n+1}, this becomes an adaptive control law if the matrices W and W_{n+1} are substituting by their on-line estimates calculating by appropriate updating laws. Since we are interested to obtain stable adaptive control laws the Lyapunov synthesis method is used. Consider the following Lyapunov function candidate:

$$V = \frac{1}{2}\left(e^T_m P e^T_m + e^T_t \Lambda^{-1} e^T_t + tr\{\widetilde{W}^T\widetilde{W}\} + tr\{\widetilde{W}^T_{n+1}\widetilde{W}_{n+1}\}\right) \tag{19}$$

where $P > 0$ is chosen to satisfy the Lyapunov equation

$$PA + A^T P = -I \tag{20}$$

Differentiating (19) along the solution of (17) and (15) where C is considered to be equal to identity matrix and using (18), finally we obtain:

$$\begin{aligned} \dot{V} = &-\frac{1}{2}e^T_m e_m - e^T_t e_t + \Phi^T(x)\widetilde{W}^T BPe_m \\ &+ u^T\Psi^T(x)\widetilde{W}_{n+1}BPe_m + tr\{\dot{W}^T\widetilde{W}\} + tr\{\dot{W}^T_{n+1}\widetilde{W}_{n+1}\} \end{aligned} \tag{21}$$

If we chose

$$tr\{\dot{W}^T\widetilde{W}\} = -\Phi^T(x)\widetilde{W}^T BPe_m \tag{22}$$

$$tr\{\dot{W}^T_{n+1}\widetilde{W}_{n+1}\} - u^T\Psi^T(x)\widetilde{W}_{n+1}BPe_m \tag{23}$$

then (21) becomes:

$$\dot{V} = -\frac{1}{2}e^T_m e_m - e^T_t e_t = -\frac{1}{2}\| e_m \|^2 - \| e_t \|^2 \le 0 \tag{24}$$

From (22) and (23), for network weights, we obtain the following *updating laws*:

$$\dot{w}_{ij} = -b_i p_i \phi(x_j) e_{mi},\quad i, j = 1,...,n \tag{25}$$

$$\dot{w}_{i,n+1} = -b_i p_i \psi(x_i) u_i e_{mi},\quad i = 1,...,n \tag{26}$$

Now, we can prove the following result:

Theorem 1. Consider the control scheme (11), (15), (17). The updating laws (25) and (26) guarantees the following properties:

i) $\lim_{t\to\infty} e_m(t) = 0$, $\lim_{t\to\infty} e_t(t) = 0$

ii) $\lim_{t\to\infty} \dot{\tilde{W}}(t) = 0$, $\lim_{t\to\infty} \dot{\tilde{W}}_{n+1}(t) = 0$

Proof. Since $\dot{V}$ is negative semidefinite (see (24)), we have that $V \in L_\infty$, which implies that $e_t, e_m, \tilde{W}, \tilde{W}_{n+1} \in L_\infty$ [10]. Since V is a non-increasing function of time and bounded from below, then there exists $\lim_{t\to\infty} V(t) = V(\infty)$. Therefore, by integrating $\dot{V}$ from 0 to ∞ we have

$$\int_0^\infty \left(\frac{1}{2} \| e_m \|^2 + \| e_t \|^2 \right) dt = V(0) - V(\infty) < \infty$$

which implies that $e_t, e_m \in L_2$. By definition, the sigmoidal functions $\phi(x_i)$ and $\psi(x_i)$, $i = 1,...,n$ are bounded for all x and by assumption all inputs to the neural network, inclusive y_{ref} and its time derivative, are also bounded. Then, from (17) we conclude that $\dot{e}_m \in L_\infty$. Since $e \in L_2 \cap L_\infty$ and $\dot{e}_m \in L_\infty$, using Barbalat's Lemma [11], one obtains that $\lim_{t\to\infty} e_m(t) = 0$. Using now the boundedness of u, W, W_{n+1} and the convergence of $e_m(t)$ to zero, we have that $\dot{W}$ and $\dot{W}_{n+1}$ also converge to zero.

4 A Working Example

The simulations were conducted in the context of a nonlinear continuous biotechnological process for which dynamical kinetics and yield coefficients are not exactly known. The model used in this work corresponds to microalgae fermentation process and is described by the following differential equation system [8]:

$$\dot{X} = \mu X - X F_{in} / V \tag{27a}$$

$$\dot{S} = -1/Y_s \mu X + (S_{in} - S) F_{in} / V \tag{27b}$$

$$\dot{C}_t = q X^{1/3} - C_t F_{in} / V \tag{27c}$$

with X, S, C_t and S_{in}, the biomass, the substrate, the toxin and the influent substrate concentrations, V the volume of the liquid phase, F_{in} the total volumetric feed rate, μ the specific growth rate, Y_s the yield coefficient, and q is the toxin production constant. The parameters appearing in this description are complex functions of the variables of interest. The yield coefficient Y_s is given by

$$Y_s = \frac{k\,\mu(\cdot)}{M\,k + \mu(\cdot)} \tag{28}$$

where M and k are real positive constants, and the specific growth rate $\mu(\cdot)$ is described by the following nonlinear inhibited modified Monod model:

$$\mu = \mu_m \left(\frac{S}{K_s + S + S^3 / K_i} \right) \left(\frac{K_t}{K_t + C_t^2} \right) \tag{29}$$

where μ_m is the maximum specific growth rate, K_s is the Monod constant, K_i is the substrate inhibition constant, and K_t is toxin inhibition constant.

If we denote by $x = [X\ S\ C_t]^T$, the state vector, and $u_1 = F_{in}/V$ and $u_2 = F_{in} S_{in} / V$, the control inputs variables, the bioreactor model equations can be rewritten in the form of (1) as follows:

$$\frac{d}{dt}\begin{bmatrix} X \\ S \\ C_t \end{bmatrix} = \begin{bmatrix} \mu X \\ -(M + \mu / y) X \\ q X^{1/3} \end{bmatrix} + \begin{bmatrix} -X & 0 \\ -S & 1 \\ -C_t & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{30}$$

and the drift vector field f and the input vector fields g_1, g_2 are therefore:

$$f = \begin{bmatrix} f_1(x) \\ f_2(x) \\ f_3(x) \end{bmatrix} = \begin{bmatrix} \mu X \\ -(M + \mu / y) X \\ q X^{1/3} \end{bmatrix},$$

$$g_1 = \begin{bmatrix} -X \\ -S \\ -C_t \end{bmatrix},\ g_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \tag{31}$$

The actual outputs of the system are identical with two states of the system $y_1 = X$, $y_2 = S$. It can be seen that the first and the second equations in (30) have the relative degree one. With the previous definitions, the decouplation matrix in (5) is particularized as follows:

$$CG(x) = \begin{bmatrix} -X & 0 \\ -S & 1 \end{bmatrix} \tag{32}$$

In a normal operation of bioreactor the biomass concentration can not be identically zero. This results from the fact that if $X \to 0$ would imply total washout of the species population. Then we state $X \neq 0$ which is sufficient for $rank(CG(x)) = 2$.

4.1 Nonlinear inverse dynamic controller

If we consider that the kinetics and yield coefficients in the fermentation model are known, then the exactly linearization feedback control

$$u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = -(CG(x))^{-1} \begin{bmatrix} f_1(x) \\ f_2(x) \end{bmatrix} + (CG(x))^{-1} \begin{bmatrix} \nu_1 \\ \nu_2 \end{bmatrix}$$

$$= -\begin{bmatrix} -X & 0 \\ -S & 1 \end{bmatrix}^{-1} \left(\begin{bmatrix} \mu X \\ -(M + \mu / y) X \end{bmatrix} + \begin{bmatrix} \lambda_1 (X_{ref} - X) \\ \lambda_2 (S_{ref} - S) \end{bmatrix} \right) \tag{33}$$

where the references X_{ref} and S_{ref} are piecewise constants, lead to the following error models:

$$\dot{e}_1 = -\lambda_1 e_1 ; \quad \dot{e}_2 = -\lambda_2 e_2 \tag{34}$$

with

$$e_1 = X_{ref} - X ; \quad e_2 = S_{ref} - S \tag{35}$$

4.2 Neural network adaptive controller

In the equations (30) describing the fermentation process model we assume now that only the terms Xu_1 and $u_2 - Su_1$ (which depend upon the inputs) are known. The first and the second equations in (30) can now be written in the form

$$\begin{aligned} \dot{y}_1 &= \dot{X} = \eta_1(x) - Xu_1 \\ \dot{y}_2 &= \dot{S} = \eta_2(x) + u_2 - Su_1 \end{aligned} \tag{36}$$

where $\eta_1(x)$ and $\eta_2(x)$ are considered as unknown functions. As in the previous case our objective is to determine u_1 and u_2 so that X and S follow the desired outputs X_{ref} and S_{ref}. If u_1 and u_2 can be chosen as

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = -\begin{bmatrix} -X & 0 \\ -S & 1 \end{bmatrix}^{-1} \left(\begin{bmatrix} \eta_1(x) \\ \eta_2(x) \end{bmatrix} + \begin{bmatrix} \lambda_1 (X_{ref} - X) \\ \lambda_2 (S_{ref} - S) \end{bmatrix} \right) \tag{37}$$

the resulting error equations have the form of (34). Because of the functions $\eta_1(x)$ and $\eta_2(x)$ are not known our objective is to estimate these unknown functions on-line using a neural network of the form

$$\dot{\hat{\eta}}_i = a_i \hat{\eta}_i + b_i \sum_{j=1}^{n} w_{ij} \phi(\eta_j) + b_i w_{i,n+1} \psi(\eta_i) u_i ,$$
$$i, j = 1, 2 \tag{38}$$

Hence, u_1 and u_2 in (37) are modified so that $\hat{\eta}_1(\cdot)$ and $\hat{\eta}_2(\cdot)$ are used in place of $\eta_1(x)$ and $\eta_2(x)$, where $\hat{\eta}_1$ and $\hat{\eta}_2$ are on-line estimations of η_1 and η_2. Then, the resulting error equations have the form

$$\begin{aligned} \dot{e}_1 &= -\lambda_1 e_1 + \eta_1(x) - \hat{\eta}_1 \\ \dot{e}_2 &= -\lambda_2 e_2 + \eta_2(x) - \hat{\eta}_2 \end{aligned} \tag{39}$$

The used neural network has two inputs (u_1 and u_2) and two outputs ($\hat{\eta}_1$ and $\hat{\eta}_2$). The parameters w_{ij} and $w_{i,n+1}$ $(i, j = 1, 2)$ are adjusted using the errors e_1 and e_2 respectively, according the rules (25) and (26). The initial values of the weights and the design parameters were chosen as:

$$W(0) = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}; \; W_{n+1}(0) = \begin{bmatrix} 0.05 \\ 0.05 \end{bmatrix};$$

$m_1 = 10$, $m_2 = 20$, $\delta_1 = \delta_2 = 1$, $\theta_1 = \theta_2 = 0$, $a_1 = -5$, $a_2 = -15$, $b_1 = 0.2$, $b_2 = 0.1$, $p_1 = 0.55$, $p_2 = 0.15$.

For a proper comparison of the two control strategies, the simulations were carried out under identical conditions and the results were judged using the same set of criteria. The initial conditions and the kinetics and yield parameters are as follows:

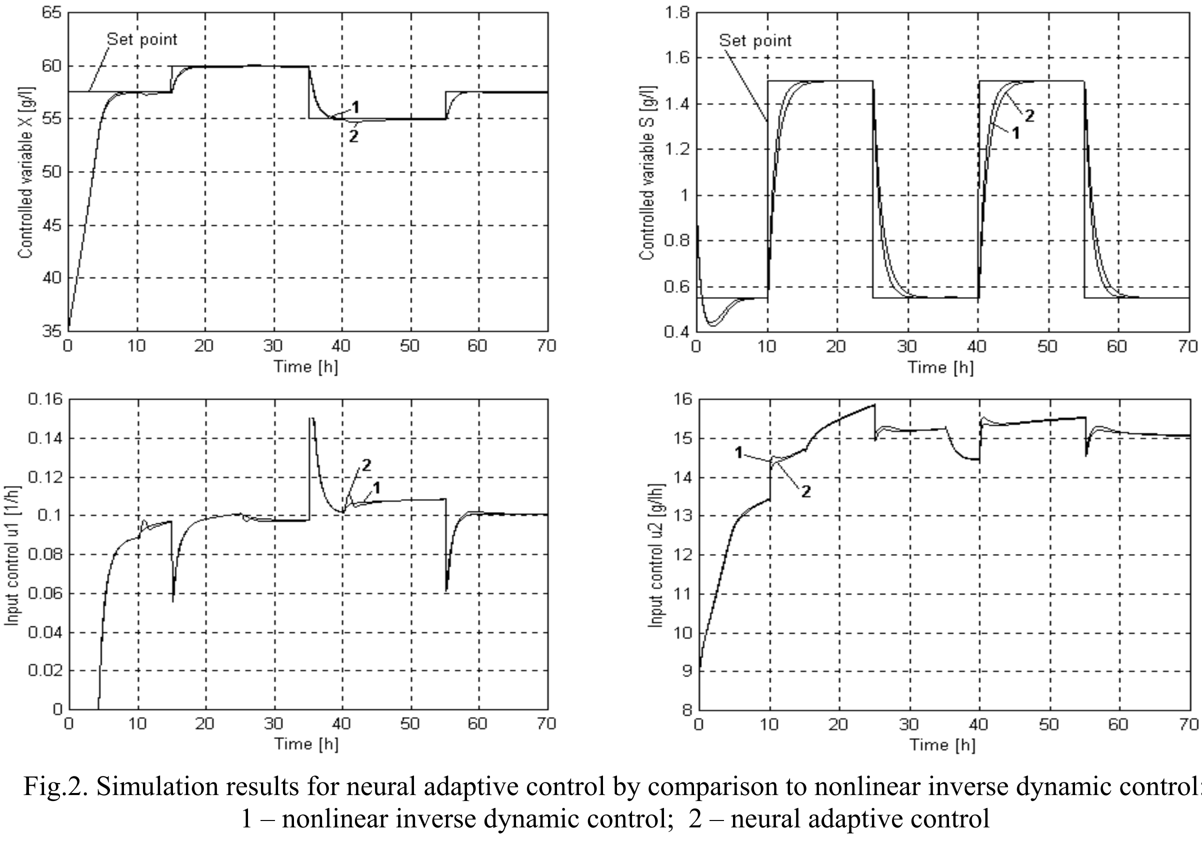

Fig.2. Simulation results for neural adaptive control by comparison to nonlinear inverse dynamic control: 1 – nonlinear inverse dynamic control; 2 – neural adaptive control

$X(0) = 35$, $S(0) = 1$, $C_t(0) = 1$, $S_{in}(0) = 150$, $F_{in}(0) = 20$; $\mu_m = 0.135$, $K_s = 0.05$, $K_i = 2150$, $K_t = 5.5$, $M = 0.0196$, $k = 0.147$, $q = 0.0296$, $V = 200$.

The bounds on control inputs are

$u_1 : 0 \leq F_{in} / V \leq 0.15$; $u_2 : 0 \leq S_{in} F_{in} / V \leq 45$

which are imposed by physical constraints and feed rate and input substrate concentration, respectively.

The simulation experiments were designed so that several set point changes on the controlled variables, biomass concentrations X and limiting substrate concentrations S occurred. The responses of the overall system for $\lambda_1 = \lambda_2 = 1.15$ are given in Fig.2.

While the nonlinear inverse dynamic controller (34) requires maximum prior information, it also yields the best response and can be used as a benchmark to evaluate the performances of other controllers that require less prior information. It can be seen that the responses of the overall system with neural adaptive controller, even if this used much less priority information, are comparable to those obtained using the exact linearizing controller.

5 Conclusions

This paper is concerned with the using of dynamical neural networks in an adaptive linearizing control problem, when the plant under consideration is described by a square multivariable model with relative degree one for every equation. The technique employed for arriving at the convergence result is also presented. Since, in most situations, the process nonlinearities are not known sufficiently accurately, it can be concluded that adaptive controllers and neural network controllers are two viable alternatives.

References:

[1] F.C. Chen, C.C. Liu, Adaptively controling nonlinear continuous-time systems using multi-layer neural networks, *IEEE Trans. Automat. Contr.*, Vol. 39, No. 6, 1994, pp. 1306-1310.

[2] B. Fidan, Y. Zhang, P.A., Ioannou, Adaptive Control of a Class of Slowly Time Varying Systems with Modelling Uncertainties, *IEEE Trans. Automat. Contr.*, Vol. 50, No. 6, 2005, pp. 915-920.

[3] K. Funahashi, On the approximate realization of continuous mappings by neural networks, *Neural Networks,* Vol. 2, 1989, pp. 183-192.

[4] S. Haykin, *Neural Networks. A Comprehensive Foundation*, Macmillan Publishing Company, Englewood Cliffs, 1994.

[5] A. Isidori, *Nonlinear Control Systems - The Third Edition,* Springer-Verlag, Berlin, 1995.

[6] R.B. McLain, M.A. Henson, M. Pottmann, Direct adaptive control of partially known non-linear systems. *IEEE Trans. Neural Networks*, Vol. 10, No. 3, 1999, pp. 714-721.

[7] K.S. Narendra, K. Parthasarathy, Identification and control of dynamical systems using neural networks, *IEEE Trans. Neural Networks*, Vol. 1, 1990, pp. 4-27.

[8] E. Petre, *Nonlinear Systems. Applications in Biotechnology* (in Romanian), Universitaria, Craiova, 2002.

[9] E. Petre, A Direct Adaptive Controller for a Class of Nonlinear Dynamic Systems, *Proc. of the Int. Symp. on Systems Theory - SINTES 11*. Craiova, Romania, 2003, pp. 96-101.

[10] G.A. Rovithakis, M.A. Christodoulou, Direct Adaptive Regulation of Unknown Nonlinear Dynamical Systems via Dynamic Neural Networks, *IEEE Trans. Syst. Man Cyber.*, Vol. 25, No. 12, 1995, pp. 1578-1594.

[11] S. Sastry, M. Bodson, *Adaptive control - Stability, Convergence and Robustness*. Prentice-Hall International, Inc., 1990.