DIRECT ADAPTIVE CONTROL USING FEEDFORWARD NEURAL NETWORKS

Daniel Oliveira Cajueiro*
danoc@pos.ucb.br

Elder Moreira Hemerly†
hemerly@ele.ita.cta.br

*Universidade Católica de Brasília, SGAN 916, Módulo B – Asa Norte. Brasília (DF) CEP: 70790-160

†Instituto Tecnológico de Aeronáutica ITA-IEE-IEES, Praça Marechal Eduardo Gomes 50 – Vila das Acácias São José dos Campos (SP) CEP: 12228-900

ABSTRACT

This paper proposes a new scheme for direct neural adaptive control that works efficiently employing only one neural network, used for simultaneously identifying and controlling the plant. The idea behind this structure of adaptive control is to compensate the control input obtained by a conventional feedback controller. The neural network training process is carried out by using two different techniques: backpropagation and extended Kalman filter algorithm. Additionally, the convergence of the identification error is investigated by Lyapunov's second method. The performance of the proposed scheme is evaluated via simulations and a real time application.

KEYWORDS: Adaptive control, backpropagation, convergence, extended Kalman filter, neural networks, stability.

RESUMO

Este artigo propõe uma nova estratégia de controle adaptativo direto em que uma única rede neural é usada para simultaneamente identificar e controlar uma planta. A motivação para essa estratégia de controle adaptativo é compensar a entrada de controle gerada por um controlador retroalimentado convencional. O processo de treinamento da rede neural é realizado através de duas técnicas: *backpropagation* e filtro de Kalman estendido. Adicionalmente, a convergência do erro de identificação é analisada através do segundo método de Lyapunov. O desempenho da estratégia proposta é avaliado através de simulações e uma aplicação em tempo real.

PALAVRAS-CHAVE: *Backpropagation*, controle adaptativo, convergência, estabilidade, filtro de Kalman estendido, redes neurais.

1 INTRODUCTION

Different neural networks topologies have been intensively trained aiming at control and identification of nonlinear plants (Agarwal, 1997; Hunt et al., 1992). The neural networks usage in this area is explained mainly by their following capability: flexible structure to model and learn nonlinear systems behavior (Cibenko, 1989). In general, two neural networks topologies are used: feedforward neural networks combined with tapped delays (Hemerly and Nascimento, 1999; Tsuji et al., 1998) and recurrent neural networks (Sivakumar et al., 1999; Ku and Lee, 1995). The neural network used here belongs to the former topology.

In this paper, one neural network is used for simultane-

Artigo submetido em 23/11/2000
1a. Revisão em 15/10/2001; 2a. Revisão 3/7/2003
Aceito sob recomendação dos Eds. Associados Profs. Fernando Gomide e Ricardo Tanscheit

ously identifying and controlling the plant and the uncertainty can be explicitly identified. The idea behind this approach of direct adaptive control is to compensate the control input obtained by a conventional feedback controller. A conventional controller is designed by employing a nominal model of the plant. Since the nominal model may not match the real plant, the performance of the nominal controller previously designed will not be adequate in general. Thus the neural network, arranged in parallel with the feedback conventional controller, identifies the uncertainty explicitly and provides the control signal correction so as to enforce adequate tracking.

For some other neural direct adaptive control schemes the neural network is placed in parallel with the feedback conventional controller as has been presented (Kraft and Campanha, 1990; Kawato *et al.*, 1987). The scheme proposed here differs from these results, in the sense that the neural network aim at improving and not replacing the nominal controller. Additionally, Lightbody and Irwin (1995) proposed a neural adaptive control scheme where the neural network is arranged in parallel with a fixed gain linear controller. The main difficult of this scheme is to update the neural network weights by using the error calculated between the real plant and the reference, in other words, the problem known as backpropagation through the plant (Zurada, 1992). In this case, the convergence of the neural network identification error to zero is often more difficult to be achieved (Cui and Shin, 1993). This problem does not appear here.

Although our approach is similar to that used by Tsuji et al. (1998), it has three main advantages (Cajueiro, 2000): (a) while their method demands modification in the weight updating equations which is computationally more complicated, our training procedure can be performed by a conventional feedforward algorithm; (b) while their paper presents a local asymptotic stability analysis of the parametric error for the neural network trained by backpropagation algorithm where the nominal model must be SPR, that analysis can also be applied to our scheme without this condition; (c) their scheme can be only applied for plants with stable nominal model, which is not the case here.

This paper is organized as follows. In section 2, the control scheme and the model of the plant are introduced. In section 3 the model of the neural network is described and in section 4 the convergence of the NN based adaptive control is investigated. In section 5 some simulations are conducted. Finally, in section 6 a real time application is presented. Section 7 deals with the conclusion of this work.

2 PLANT MODEL AND THE PROPOSED ADAPTIVE CONTROL SCHEME

2.1 Plant Model with Multiplicative Uncertainties

We firstly consider the case in which the controlled plant presents multiplicative uncertainty (Maciejowski, 1989). Consider the SISO plant

$$y(k) = G(z^{-1})u(k) \tag{1}$$

Let $G_n(z^{-1})$ be the nominal model and $\Delta G_m(z^{-1})$ the multiplicative uncertainty, i.e.,

$$G(z^{-1}) = G_n(z^{-1})[1 + \Delta G_m(z^{-1})] \tag{2}$$

where $G_n(z^{-1})$ and $\Delta G_m(z^{-1})$ are given as

$$G_n(z^{-1}) = \frac{B_n(z^{-1})}{A_n(z^{-1})} \tag{3}$$

being $B_n(z^{-1})$ is Hurwitz,

$$\Delta G_m(z^{-1}) = \frac{\Delta B_m(z^{-1})}{\Delta A_m(z^{-1})} \tag{4}$$

2.2 Proposed Adaptive Control Scheme

We start by designing the usual feedback control system. If there is no uncertainty, i.e., $\Delta G_m(z^{-1}) = 0$ in (2), then the nominal feedback controller $C_n(z^{-1})$ can be designed to produce the desirable performance.

Now, let us consider the general case in which $\Delta G_m(z^{-1}) \neq 0$. Hence the controller must be modified accordingly, i. e.,

$$C(z^{-1}) \triangleq C_n(z^{-1})[1 + \Delta C_m(z^{-1})] \tag{5}$$

It is easy to prove that

$$\Delta C_m(z^{-1}) = -\frac{\Delta G_m(z^{-1})}{1 + \Delta G_m(z^{-1})} \tag{6}$$

is the controller correction necessary to enforce the desired performance (Tsuji et al., 1998).

However, $\Delta G_m(z^{-1})$ is unknown, hence $\Delta C_m(z^{-1})$ can not be calculated in (6), and then the controller (5) is not directly implementable. In order to circumvent this difficulty, we propose the NN based scheme for identifying the uncertainty $\Delta G_m(z^{-1})$ shown in Fig. 1.

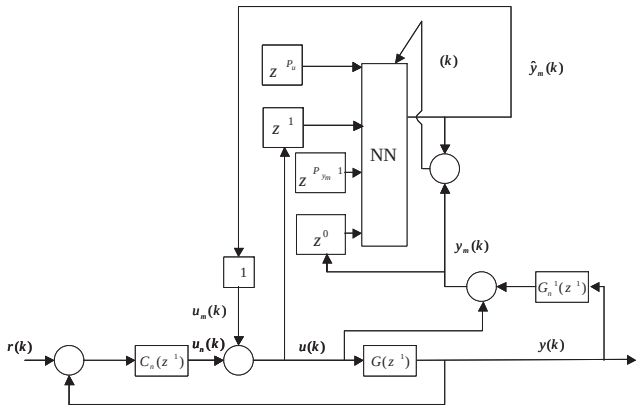

Figura 1: Block diagram of the proposed NN based adaptive control scheme.

Definition 1: $y_n(k)$, the output of the nominal model, is given by

$$y_n(k) = G_n(z^{-1})u(k) \tag{7}$$

Definition 2: $\Delta y_m(k)$, the filtered mismatch between the plant output and its nominal model output denoted by $y(k)$ and $y_n(k)$ respectively, is given by

$$\begin{aligned}\Delta y_m(k) \triangleq & \Delta G_m(z^{-1})u(k) \\ = & G_n^{-1}(z^{-1}) \cdot (y(k) - y_n(k)) \\ = & G_n^{-1}(z^{-1}) \cdot y(k) - u(k)\end{aligned} \tag{8}$$

Definition 3: $u(k)$, the control input, from (5) can be written as

$$u(k) = C_n(z^{-1})[1 + \Delta C_m(z^{-1})]e(k) \triangleq u_n(k) + \Delta u_m(k) \tag{9}$$

where

$$u_n(k) = C_n(z^{-1})e(k) \tag{10}$$

is the nominal control signal and

$$\Delta u_m(k) = \Delta C_m(z^{-1})u_n(k) \tag{11}$$

is the control signal modification.

Definition 4: $\varepsilon(k)$, the identification error, is given by

$$\varepsilon(k) = \Delta y_m(k) - \Delta \hat{y}_m(k) \tag{12}$$

From (1), (2), (7) and (8) the output of the plant is given by

$$\begin{aligned}y(k) = & G_n(z^{-1})[1 + \Delta G_m(z^{-1})]u(k) \\ = & y_n(k) + G_n(z^{-1})\Delta y_m(k)\end{aligned} \tag{13}$$

Now, from (9) and (13) we get

$$\begin{aligned}\Delta y_m(k) = & \Delta G_m(z^{-1})[u_n(k) + \Delta u_m(k)] \\ = & \Delta G_m(z^{-1})[1 + \Delta C_m(k)]u_n(k)\end{aligned} \tag{14}$$

and from (9) and (14), the control signal correction $\Delta u_m(k)$ can be rewritten as

$$\Delta u_m(k) = \frac{\Delta C_m(z^{-1})}{\Delta G_m(z^{-1})[1 + \Delta C_m(z^{-1})]}\Delta y_m(k) \tag{15}$$

By replacing (6) into (15), we obtain

$$\Delta u_m(k) = -\Delta y_m(k) \tag{16}$$

Moreover, if $\varepsilon \to 0$, from equations (12) and (16) then $\Delta \hat{y}_m(k) = \Delta y_m(k) = -\Delta u_m(k)$.

On the other hand, from equations (1), (2), (8) and (16), one should write

$$\begin{aligned}y(k) = & y_n(k) + G_n(z^{-1})\Delta y_m(k) \\ = & G_n(z^{-1})(u_n(k) + \Delta u_m(k)) + G_n(z^{-1})\Delta y_m(k) \\ = & G_n(z^{-1})(u_n(k) - \Delta y_m(k)) + G_n(z^{-1})\Delta y_m(k) \\ = & G_n(z^{-1})u_n(k)\end{aligned} \tag{17}$$

Remark 1: It is clear from (17) that if $\varepsilon \to 0$ then the control scheme will behave as desired. Therefore, from (10) it is obvious that the nominal controller signal converges to the nominal controller signal obtained by controlling the nominal plant.

Remark 2: The neural network here is a direct model of the uncertainty $\Delta G_m(z^{-1})$ and is aiming at approximating the uncertainty output $\Delta y_m(k)$.

Remark 3: In spite of the nominal model minimum phase restriction this scheme of adaptive control can also be applied to non-minimum phase systems. It depends on the neural network ability for identifying the uncertainty arising from a non-minimum phase plant modeled by a minimum phase nominal model.

Remark 4: Although this neural adaptive control scheme is developed to compensate the nominal controller signal of plants with multiplicative uncertainty, it also can compensate other type of non structured uncertainty. Consider, for instance, the additive case

$$G(z^{-1}) = G_n(z^{-1}) + \Delta G_a(z^{-1}) \tag{18}$$

Since the nominal model is non minimum phase, from (2) and (18), we conclude that there is always a correspondent multiplicative uncertainty for the additive uncertainty given by (18).

On the other hand, although equations (16) and (17) can only be applied for linear plants, this scheme of neural adaptive control also presents good performance when applied for nonlinear plants (Cajueiro and Hemerly, 2000).

3 NEURAL NETWORK TOPOLOGY

The neural network topology used here is a feedforward one combined with tapped delays. The training process is carried out by using two different techniques: backpropagation and extended Kalman filter. The two approaches used to train the neural network are justified due to the slow nature of the training with the standard backpropagation algorithm (Sima, 1996; Jacobs, 1988). Thus the neural network trained via extended Kalman filter can be useful in more difficult problems.

The input of the neural network is defined as follows

$$\begin{aligned}\Phi^I(k) = [\ & u(k-1) \quad \cdots \quad u(k-P_u\ ) \\ & \Delta y_m(k-1) \quad \cdots \quad \Delta y_m(k-P_{\Delta y_m})\ ]^T \end{aligned} \tag{19}$$

where $\Delta y_m(k-1) \quad \ldots \quad \Delta y_m(k-P_{\Delta y_m})$ is calculated by using (8) and the cost function is defined, from (12) as

$$J(k) = \frac{1}{2}\varepsilon(k)^2 = \frac{1}{2}(\Delta y_m(k) - \Delta \hat{y}_m(k))^2 \tag{20}$$

3.1 Learning Algorithm Based on the Extended Kalman Filter

The Kalman filter approach for training a multilayer perceptron considers the optimal weights of the neural network as the state of the system to be estimated and the output of the neural network as the associated measurement. The optimal weights are those that minimize the cost $J(k)$. The weights are in general supposed to be constant, and the problem boils down to a static estimation problem. However, it is often advantageous to add some noise, which prevents the gain from decreasing to zero and then forces the filter to continuously adjusting the weight estimates. Therefore, we model the weights as

$$\begin{aligned} w(k+1) &= w(k) + v_w(k), \\ E[v_w(i)v_w(j)^T] &= P_{v_w}\delta(i-j) \end{aligned} \tag{21}$$

The measurements of the system are assumed to be some nonlinear function of the states corrupted by zero-mean white Gaussian noise. Thus, the observations of the system are modeled as

$$\begin{aligned}\Phi^O(k) &= h(w(k), \Phi^I(k)) + v_{\Phi^O}(k) E[v_{\Phi^O}(i) v_{\Phi^O}(j)^T] \\ &= P_{\Phi^O}\delta(i-j), \end{aligned} \tag{22}$$

The extended Kalman filter equations associated to problem (19) and (20) are (Singhal and Wu, 1991)

$$\hat{w}(k+1) = \hat{w}(k) + K(k)\varepsilon(k) \tag{23}$$

$$P(k+1) = P(k) - K(k)H(k)P(k) + P_{v_w} \tag{24}$$

where $K(k)$ is the Kalman filter gain given by

$$K(k) = P(k)H(k)^T[H(k)P(k)H(k)^T + P_{v_{\Phi^O}}]^{-1} \tag{25}$$

with

$$H(k) = \left[\frac{\partial \Phi^O(k)}{\partial w(k)}\right]^T \Bigg| \ w(k) = \hat{w}(k) \tag{26}$$

There are some local approaches (Iiguni et al., 1992; Shah et al., 1992) for implementing the extended Kalman filter to train neural networks in order to reduce the computational cost of training. Since we are using small neural networks, this is not an issue here. Then, we consider only the global extended Kalman algorithm, more precisely, the approach known by GEKA - Global Extended Kalman Algorithm (Shah et al., 1992).

3.2 The Standard Backpropagation Algorithm

The standard backpropagation algorithm is a gradient descent method for training weights in a multilayer preceptron that was popularized by Rumelhart et al. (1986). The basic equation for updating the weights is

$$\Delta \hat{w}(k) = -\eta\varepsilon(k)\frac{\partial J(k)}{\partial \hat{w}(k)} = \eta\varepsilon(k)\frac{\partial \hat{\Phi}^O(k)}{\partial \hat{w}(k)} \tag{27}$$

From (23) – (26), it can be seen that the extended Kalman filter algorithm reduces to the backpropagation algorithm when

$$[H(k)P(k)H(k)^T + P_v]^{-1} = aI \tag{28}$$

$$P(k) = pI \tag{29}$$

and $\eta$is given by

$$\eta = p \cdot a \tag{30}$$

See Ruck et al. (1992) for more details.

4 STABILITY AND CONVERGENCE ANALYSIS

The stability analysis is divided into two parts: (a) the study of the influence of the neural network in the control system stability when the nominal controller employed to control the real plant is analyzed; (b) the investigation of the conditions under which the identification error of the neural network asymptotically converges to zero are investigated.

4.1 Control System Stability

The closed loop equation that represents the control system shown in Fig.1, without considering the dynamics uncertainty and the output of the neural network, is

$$y(k) = \frac{G_n(z^{-1})C_n(z^{-1})}{1 + G_n(z^{-1})C_n(z^{-1})} r(k) \tag{31}$$

It represents a stable system, since $C_n(z^{-1})$ is properly designed.

By considering now the introduction of the uncertainties and the control signal correction via NN, from Fig. 1 we get

$$y(k) = \frac{(1 + \Delta G_m(z^{-1}))G_n(z^{-1})}{1 + (1 + \Delta G_m(z^{-1}))G_n(z^{-1})C_n(z^{-1})} \hat{\Phi}^O(k) + \frac{(1 + \Delta G_m(z^{-1}))G_n(z^{-1})C_n(z^{-1})}{1 + (1 + \Delta G_m(z^{-1}))G_n(z^{-1})C_n(z^{-1})} r(k) \tag{32}$$

Since $\hat{\Phi}^O(k) = \tanh(\bullet)$, where $\tanh(\bullet)$ is the hyperbolic tangent function, and $r(k)$ are bounded, in order to guarantee the stability we have to analyze under which conditions the denominator of (32) is a Hurwitz polynomial, when the denominator of (31) also is. These conditions can be found in a very general result, known as the small gain theorem, which states that a feedback loop composed of stable operators will certainly remain stable if the product of all the operator gains is smaller than unity. Therefore, if a multiplicative perturbation satisfies the conditions imposed by the small gain theorem, then one should assert that $\varepsilon(k)$, given by equation (12), is bounded. Since $\hat{\Phi}^H(k)$ and $\hat{\Phi}^O(k)$, the output of the neural network layers, depend on the weights whose boundedness depends on the boundedness of $\varepsilon(k)$, it is clear that the boundedness of $\varepsilon(k)$ is the first condition for the output of the neural network layers, $\hat{\Phi}^H(k)$ and $\hat{\Phi}^O(k)$, not saturating. If saturation happens, then in (26) $H(k) = 0$ and we can not conclude, as in section 4.2, that the identification error converges asymptotically to zero.

4.2 Identification Error Convergence

We start by analyzing the conditions under which the neural network trained by Kalman filter algorithm guarantees the asymptotic convergence of the identification error to zero. Next, we show that a similar result is valid for the neural network trained by backpropagation algorithm.

Theorem 1: Consider that the weights $w(k)$ of a multilayer perceptron are adjusted by the extended Kalman algorithm. If $w(k) \in L_\infty$, then the identification error $\varepsilon(k)$ converges locally asymptotically to zero.

Proof:

Let $V(k)$ be the Lyapunov function candidate, which is positive definite, given by

$$V(k) = \frac{1}{2}\varepsilon^2(k) \tag{33}$$

Then, the change of $V(k)$ due to training process is obtained by

$$\Delta V(k) = \frac{1}{2}[\varepsilon^2(k+1) - \varepsilon^2(k)] \tag{34}$$

On the other hand, the identification error difference due to the learning process can be represented locally by (Yabuta and Yamada, 1991)

$$\Delta\varepsilon(k) = \left[\frac{\partial\varepsilon(k)}{\partial\hat{w}(k)}\right]^T \Delta\hat{w}(k) = \left[\frac{\partial\hat{\Phi}^O(k)}{\partial\hat{w}(k)}\right]^T \Delta\hat{w}(k) \tag{35}$$

From equations (21) and (33),

$$\Delta\varepsilon(k) = -\left[\frac{\partial\hat{\Phi}^O(k)}{\partial\hat{w}(k)}\right]^T \Delta\hat{w}(k) = -H(k)K(k)\varepsilon(k) = -Q_{EKF}\varepsilon(k) \tag{36}$$

where

$$Q_{EKF}(k) \triangleq H(k)K(k) \tag{37}$$

and it can be easily seen that $0 \leqslant Q_{EKF}(k) < 1$ and then, from (32) and (34),

$$\Delta V(k) = 2\varepsilon(k)\Delta\varepsilon(k) + \Delta\varepsilon^2(k) = -\varepsilon^2(k)(2Q_{EKF}(k) - Q^2_{EKF}(k)) \tag{38}$$

From (38) follows that for asymptotic convergence of $\varepsilon(k)$ to zero we need only $Q_{EKF}(k) \neq 0$. Now, from

(37) a sufficient condition for this is $H(k) \neq 0$. On the other hand, (26) implies this only occurs when the weights are bounded. However, this can not be proved to happen, since the Lyapunov candidate function does not explicitly include the weight error. This depends on the proper choice of the neural network size and initial parameters. Hence, we have to assume that the neural network size and initial parameters have been properly selected. This difficulty is also present, although disguised, in Ku and Lee (1995) and Liang (1997).

Corollary 1: Consider that the weights $w(k)$ of a multilayer perceptron are adjusted by the backpropagation algorithm. If $w(k) \in L_\infty$ and $0 < \eta < \frac{2}{\left\| \frac{\partial \Phi^O(k)}{\partial w(k)} \right\|^2}$, where $\Phi^O(k)$ is the output of the neural network, then the identification error $\varepsilon(k)$ converges locally asymptotically to zero.

Proof:

The convergence of the identification error of a neural network trained by backpropagation algorithm is a special case of the above result. More precisely, by considering equations (25), (28) - (30), one can arrive at an equation similar to (37), with

$$Q_{BP}(k) \triangleq \eta \left\| \frac{\partial \Phi^O(k)}{\partial w(k)} \right\|^2 \tag{39}$$

and the correspondent variation in $V(k)$ is

$$\Delta V(k) = -\varepsilon^2(k)(2Q_{BP}(k) - Q_{BP}^2(k)) \tag{40}$$

Equation (40) states that the convergence of the identification error for the neural network trained by backpropagation method is guaranteed as long as $\left\| \frac{\partial \Phi^O(k)}{\partial w(k)} \right\| \neq 0$, and

$$0 < \eta < \frac{2}{\left\| \frac{\partial \Phi^O(k)}{\partial w(k)} \right\|^2} \tag{41}$$

so as to enforce $\Delta V(k) < 0$ in (40) when $\varepsilon(k) \neq 0$.

Remark 5: Although (38) and (40) have the same form, the conditions for the identification error convergence are more restrictive when the backpropagation algorithm is used, since an upper bound in the learning rate is required, given by (41), as we should have expected.

Remark 6: Since the candidate Lyapunov function given by equation (33) does not include the parametric error $\tilde{w}(k) = w^*(k) - \hat{w}(k)$, even if there were an optimal set of parameters, the convergence of $\tilde{w}(k)$ to zero would depend on the signal persistence.

Remark 7: Since (40) is a quadratic equation, a bigger learning rate η does not imply that there is a faster learning.

5 SIMULATIONS

In this section, simulations of two different plants are presented to test the proposed control scheme. We start by considering a linear plant to which can be applied equations (16) and (17). Next, a non-BIBO nonlinear plant is used as a test. Moreover, the stability of this control system can not be assured, since the nominal controller designed by using the nominal model results in an unstable control scheme when it is applied to the real plant.

5.1 Simulation with Linear Plant

The plant used here has the nominal model given by

$$G_n(z^{-1}) = \frac{0.01752z^{-1} + 0.01534z^{-2}}{1 - 1.637z^{-1} + 0.6703z^{-2}} \tag{42}$$

and it is considered the following multiplicative uncertainty

$$\Delta G_m(z^{-1}) = \frac{0.1848z^{-1}}{1 - 0.8521z^{-1}} \tag{43}$$

Thus, from equations (2), (42) and (43), one should write the model of the real plant

$$G(z^{-1}) = \frac{0.01752z^{-1} + 0.003642z^{-2} - 0.01023z^{-3}}{1 - 2.49z^{-1} + 2.066z^{-2} - 0.5712z^{-3}} \tag{44}$$

The nominal controller here is given by

$$u_n(k) = u_n(k-1) + K_p(e(k) - e(k-1)) + K_I e(k) \tag{45}$$

with $K_p = 2.0$ and $K_I = 0.26$.

As it can be seen in Fig. 2, although the control system presents good performance when nominal controller is applied to the nominal model, the control system is too oscillatory when the nominal controller is applied to the real plant.

The input of the neural network is

$$\Phi^I(k) = \begin{bmatrix} u(k-1) & \Delta y_m(k-1) \end{bmatrix} \tag{46}$$

The initial weights were initialized within the range $\begin{bmatrix} -0.1 & 0.1 \end{bmatrix}$. The simulation was performed using the

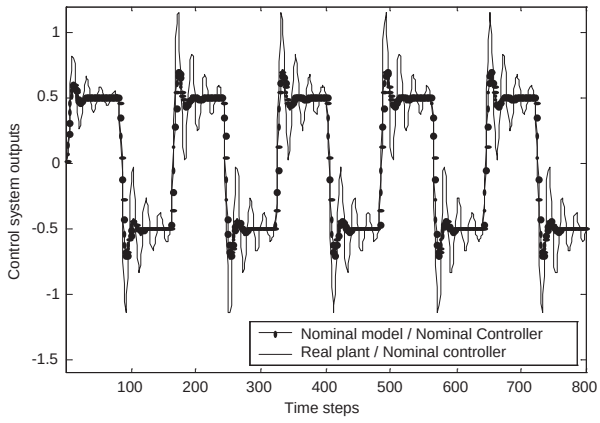

Figura 2: Control systems performance using the nominal controller

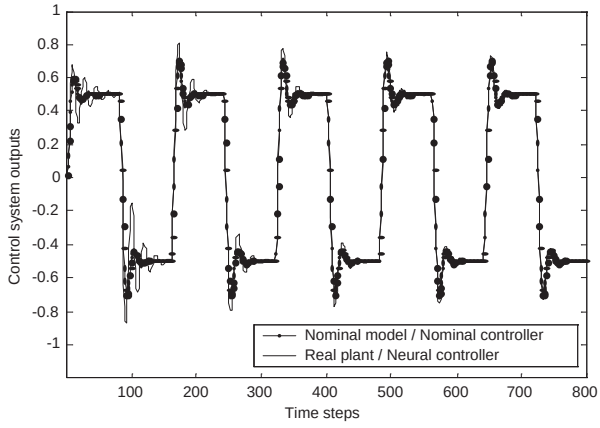

Figura 3: Performance of the real plant controlled by the proposed scheme using the backpropagation algorithm.

usual backpropagation. The usage of the algorithm based on the extended Kalman filter is not justified since the neural network task here is simple. Fig. 3 shows the output of the nominal model and the output of the real plant controlled by the proposed scheme using the backpropagation algorithm, after few time steps an adequate performance is achieved.

Fig. 4 shows that the nominal controller signal converges to the nominal controller signal that would exist if there was no uncertainty.

Remark 8: Since the plant given by (44) is linear, one should note that the uncertainty could be identified by using a linear neural network or a least squares and an ARX model. This will not occur in section 5.2.

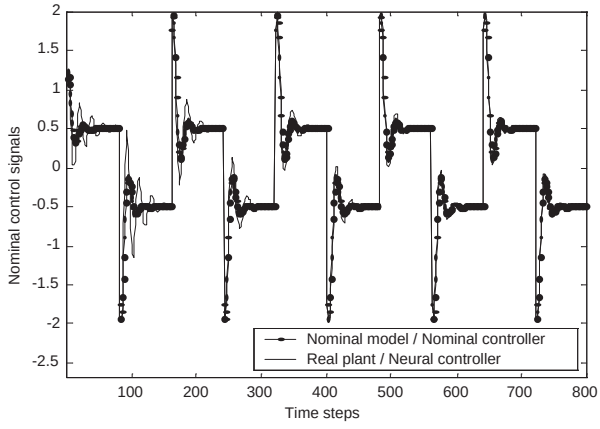

Figura 4: Nominal control signals.

5.2 Simulations with a Non-BIBO Nonlinear Plant

The nonlinear plant used to test the proposed method is the one given by Ku and Lee (1995), that is, a non-BIBO nonlinear plant, linear in the input.

The reference signal is given by

$$r(k+1) = 0.6r(k) + \beta \sin(2\pi k/100) \qquad (47)$$

where $\beta = 0.2$, and the real plant is given by

$$y(k+1) = 0.2y^2(k) + 0.2y(k-1) + 0.4\sin[0.5(y(k)+ y(k-1))] \cdot \cos[0.5(y(k)+y(k-1))] + 1.2u(k) \qquad (48)$$

This plant is unstable in the sense that given a sequence of uniformly bounded controls $\{u(k)\}$, the plant output may diverge. The plant output diverges when the step input $u(k) \geqslant 0.83,\ \forall k$ is applied.

Ku and Lee (1995) employed two DRNNs (Diagonal Recurrent Neural Networks), one as an identifier and the other as the controller, and an approach based on adaptive learning rates to control this system. Their neural network used as identifier employed 25 neurons and the one used as controller had 42. Here, we use the scheme proposed in section 2: the nominal model is that identified by employing an ARX model, and the neural network for simultaneous control and identification, has 21 neurons. Although our approach is much simpler and less computationally expensive than that in Ku and Lee (1995) and it produces better results.

The nominal model was identified by using Least Squa-

res and ARX model, thereby resulting

$$G_n(z^{-1}) = \frac{1.2134z^{-1} - 0.6124z^{-2}}{1 - 0.9061z^{-1} + 0.1356z^{-2}} \qquad (49)$$

In this simulation, the nominal controller used is a proportional-integral controller, given by (45) with $K_p = K_I = 0.5$.

The control system employing the nominal controller for controlling the real plant (48) results unstable, although the nominal controller provides good tracking of the reference signal (47) when applied to the nominal model (49).

The input of the neural network here is also given by (46).

The initial weights were randomly initialized in the range $[-0.1, 0.1]$. The simulations were performed by usual backpropagation and extended Kalman filter using the same conditions and the same seed for generating the initial weights. The learning rate used in the first case was $\eta = 1$ and the initial data for the second case was $P(0) = 100I$, $P_{\Phi o} = P_w = 10^{-5}$.

In Fig. 5, the output of the nominal model controlled by the nominal controller and the output of the real plant controlled by the proposed scheme using the backpropagation algorithm are presented. After a hundred time steps, the control system exhibits adequate tracking.

Fig. 6 shows the control system outputs when the extended Kalman filter algorithm is used. Since the extended Kalman filter uses the information contained in the data more efficiently, the convergence is much faster than that in Fig. 5.

It should be highlighted here that the speed of convergence in Figs. 5 and 6 is more than 10 times faster than that reported in (Ku and Lee, 1995).

Remark 9: A comparison between the algorithm based on the extended Kalman filter and the backpropagation algorithms can be done as follows: (a) The algorithm based on the extended Kalman Filter presents better transient performance, but it is computationally more expensive; (b) the algorithm based on the extended Kalman filter has presented more sensibility to the choice of its parameters $P(0)$, $P_{\Phi o}$ and P_w than the backpropagation algorithm for the choice of its only one parameter η.

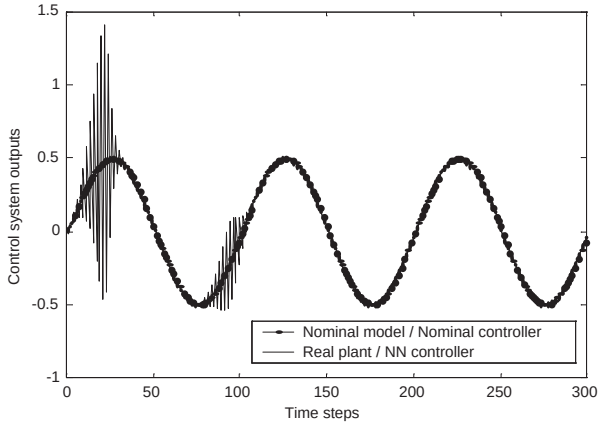

Figura 5: Simulation of a nonlinear non-BIBO plant using backpropagation algorithm.

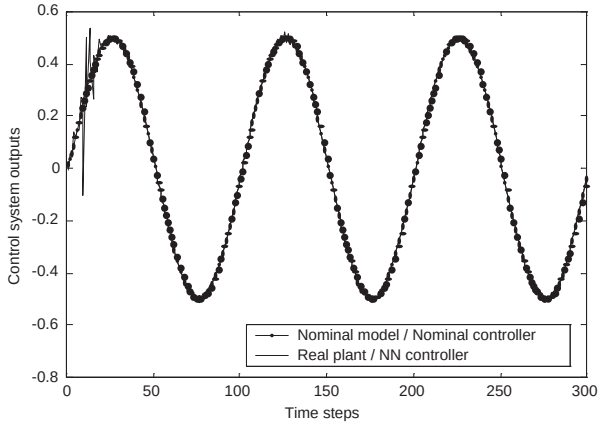

Figura 6: Simulation of a nonlinear non-BIBO plant using extended Kalman filter algorithm.

6 REAL TIME APPLICATION

The real time application employs the Feedback Process Trainer PT326, shown in Fig. 7, and the scheme proposed in section 2. This process is a thermal process in which air is fanned through a polystyrene tube and heated at the inlet, by a mesh of resistor wires. The air temperature is measured by a thermocouple at the outlet.

The nominal model used was identified using Least Squares with ARX model, as in Hemerly (1991), and is given by

$$G_n(z^{-1}) = \frac{0.0523}{z - 0.8064} \qquad (50)$$

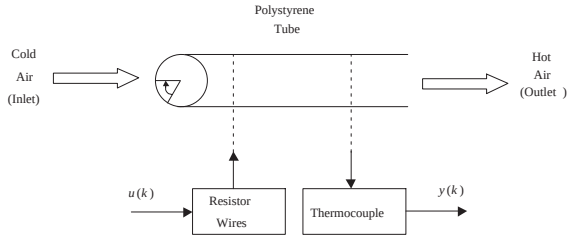

Figura 7: Set up for real time control of the thermal process PT326.

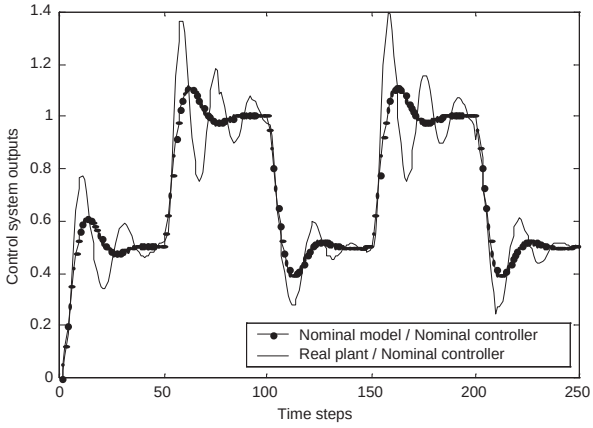

Figura 8: Control system performance using the nominal controller.

The nominal controller is a proportional-integral controller with $K_P = 1$ and $K_I = 1$, given by the equation (45). The input of the neural network is the same as in (46). The neural network used here is trained using the backpropagation method. The remaining design parameters are sampling time $0.3s$, learning rate $\eta = 0.05$ and the initial weights randomly in the range $[-0.1, 0.1]$.

As can be seen in Fig. 8, the nominal controller is such that the control system presents a too oscillatory behavior. Hence, the introduction of the neural network is well justified.

In Fig. 9 we can see that the NN controller compensates for the uncertainty, and adequate performance is achieved after few time steps.

7 CONCLUSIONS

This paper proposed a neural adaptive control scheme useful for controlling linear and nonlinear plants (Cajueiro, 2000; Cajueiro and Hemerly, 2000), using only one

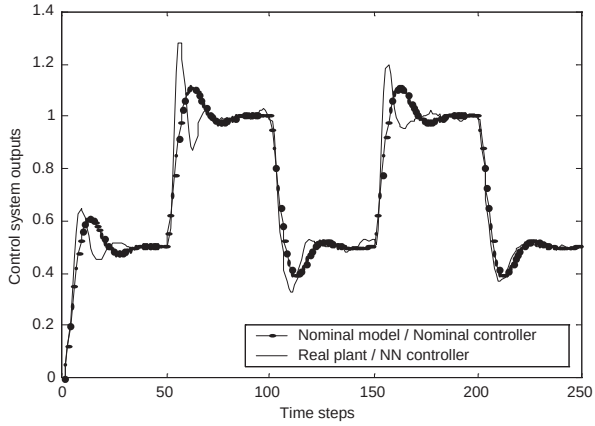

Figura 9: Control system performance using the NN based proposed scheme.

neural network which is simultaneously applied for control and identification. The nominal model can either be available or identified at low cost, for instance by using the Least Squares algorithm. The identification performed by the neural network is necessary only to deal with the dynamics not encapsulated in the nominal model. If the proposed scheme is compared to another schemes, one should consider the following: (a) The approaches which try to identify the whole plant dynamics can have poor transient performance. For instance, in Ku and Lee (1995) the plant described by (48) is identified and more than three thousand time steps are required. (b) The neural network control based methods that need to identify the inverse plant have additional problems. (c) If more than one neural network has to be used, in general the convergence of the control scheme is slow and the neural network tuning is usually difficult. (d) The approaches that have to backpropagate the neural network identification error through the plant are likely to have problems to update the neural network weights.

When compared to Tsuji et al. (1998), the proposed scheme requires less stringent assumptions. In their local stability analysis the SPR condition is required for the nominal model. Moreover, we can employ an usual feedforward neural network, which is computationally less expensive than the one used there. Besides the scheme proposed here can also be applied to unstable plants. Additionally, here the asymptotic convergence of the identification error is analyzed for two different training algorithms.

The simulations and the real time application highlighted the practical importance of the proposed scheme.

ACKNOWLEDGEMENTS

This work was sponsored by Fundação CAPES - Comissão de Aperfeiçoamento de Pessoal de Nível Superior and CNPQ-Conselho Nacional de Desenvolvimento Científico e Tecnológico, under grant 300158/95-5(RN).

REFERÊNCIAS

Agarwal, M. (1997) A systematic classification of neural network based control, Control Systems Magazine, April, Vol. 17, No. 2, pp. 75-93.

Cajueiro, D. O. (2000) Controle adaptativo paralelo usando redes neurais com identificação explícita da dinâmica não modelada. Master Thesis, Instituto Tecnológico de Aeronáutica, ITA-IEEE-IEES.

Cajueiro, D. O. and Hemerly, E. M. (2000) A chemical reactor benchmark for parallel adaptive control using feedforward neural networks. Brazilian Symposium of Neural Networks, Rio de Janeiro.

Cibenko, G. (1989). Approximation by superpositions of a sigmoidal function, Mathematics of Control, Signals and Systems, Vol. 2, pp. 303-314.

Cui, X. and Shin, K. (1997) Direct control and coordination using neural networks. IEEE Transactions on Systems, man and Cybernetics, Vol. 23, No. 3, pp. 686-697.

Hemerly, E. M. (1991). PC-based packages for identification, optimization and adaptive control, IEEE Control Systems Magazine, Vol. 11, No. 2, pp. 37-43.

Hemerly, E. M. and Nascimento Jr., C. L. (1999). A NN-based approach for tuning servocontrollers, Neural Networks, Vol. 12, No. 3, pp. 113-118.

Hunt et al. (1992) Neural networks for control systems – a survey, Automatica, Vol. 28, No. 6, pp. 1083-1112.

Iiguni, Y., Sakai, H. and Tokumaru, H. (1992) A real time algorithm for a multilayered neural network based on the extended Kalman algorithm, IEEE Transactions on Signal Processing, Vol. 40, No. 4, pp. 959-966.

Jacobs, R. (1988). Increased rates of convergence through learning rate adaptation, Neural Networks, Vol. 1, pp. 295-307.

Kawato, M. Furukawa, K. and Suzuki, R. (1987) A hierarchical neural network model for control and learning of voluntary movement, Biological Cybernetics, Vol. 57, No. 6, pp. 169-185.

Kraft, L. G. and Campagna, D. S. A. (1990) A summary comparison of CMAC neural network and two traditional adaptive control systems, In: T. W. Miller III, R. S. Sutton, P. J. Werbos, Neural Networks for control. Cambridge: The MIT Press, pp. 143-169.

Ku, C. C. and Lee, K. Y. (1995). Diagonal recurrent neural networks for dynamic systems control, IEEE Transactions on Neural Networks, Vol. 6, No. 1, pp. 144-156.

Liang, X. (1997). Comments on "Diagonal recurrent neural networks for dynamic systems control" – Reproof of theorems 2 and 4, IEEE Transactions on Neural Networks, Vol. 8, No. 3, pp. 811-812.

Lightbody, G. and Irwin, G. (1995) Direct neural model reference adaptive control. IEE Proceedings in Control Theory and Applications, Vol. 142, No. 1, pp. 31-43.

Maciejowski, J. M. (1989) Multivariable feedback design. Addison-Wesley Publishing Company.

Ruck, D. W. et al. (1992). Comparative analysis of backpropagation and the extended Kalman filter for training multilayer perceptrons, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 14, No. 6, pp. 686-691.

Rumelhart, D. E., Mcclelland, J. L. and Williams, R. J. (1986). Learning representations by error propagation. In D. E. Rumelhart, G.E. Hilton, J. L. Mcclelland (Eds.), Parallel Distributed Processing, Vol. 1, pp. 318-362. Cambridge,MA: MIT Press.

Shah, S., Palmieri, F. and Datun, M. (1992). Optimal filtering algorithms for fast learning in feedforward neural networks, Neural Networks, Vol. 5, pp. 779-787.

Sima, J. (1996) Back-propagation is not efficient, Neural Networks, Vol. 9, No. 6, pp. 1017-1023.

Singhal, S. and Wu, L. (1991) Training feed-forward networks with the extended Kalman algorithm, Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Peocessing, pp. 1187-1190.

Sivakumar, S. C., Robertson, W. and Phillips, W. J. (1999). On line stabilization of block diagonal neural networks, IEEE Transactions on Neural Networks, Vol. 10, No. 1, pp. 167-175.

Tsuji, T. , Xu, B. H. and Kaneko, M. (1998). Adaptive control and identification using one neural network for a class of plants with uncertainties, IEEE Transactions on Systems, Man and Cybernetics – Part A: Systems and Humans, Vol. 28, No. 4, pp. 496-505.

Yabuta, T. and Yamada, T. (1991). Learning control using neural networks, Proceedings of the 1991 IEEE International Conference on Robotics and Automation, California, pp. 740-745.

Zurada, J. (1992) Introduction to artificial neural systems. West Publishing Company, 1992.