NEURAL PREDICTIVE CONTROL BASED ON KALMAN FILTERING ALGORITHMS

Atair Rios Neto[1], Jaime A. da Silva[2]
[1]Universidadade do Vale do Paraiba-UNIVAP
Instituto de Pesquisa e Desenvolvimento-IP&D
12245-720 São Jose dos Campos, SP
atair@univap.br
[2]Instituto Nacional de Pesquisas Espaciais – INPE
12201-970 São José dos Campos, SP
jaimes@directnet.com.br

ABSTRACT

A neural predictive control scheme is considered where Kalman filtering is used not only to train the associated feedforward neural network modeling the dynamics but to also estimate the control. An approach is proposed in which the optimization of the predictive quadratic performance functional used to determine the discrete control actions is viewed in a typical iteration as a stochastic optimal linear parameter estimation problem. Direct analogy with Kalman filtering algorithms already developed for feedforward neural networks training allows the derivation of full non parallel as well as approximated parallel processing versions of Kalman filtering control algorithms. These algorithms are shown to be the result of applying Newton's Method to appropriate control optimization functionals and to provide solutions which converge to smooth and reference tracking controls

1. INTRODUCTION

Many authors have explored optimal linear parameter estimation procedures to develop both off line and on line neural network supervised training recursive least squares (Chen and Billings, 1992) and Kalman filtering (Chandran, 1994, Chen and Ögmen, 1994, Iiguni and Sakai, 1992, Lange, 1995, Scalero and Tepedelenlioglu, 1992, Singhal and Wu, 1989, Watanabe, Fukuda and Tzafestas, 1991, Rios Neto, 1997) type of algorithms. Numerical testing has indicated that in most of the applications they have a better performance than the usual Backpropagation algorithm.

In internal model control schemes where neural networks are used to represent the plant model and its inverse (Hunt, Sbarbaro, Zbikowski and Gawthrop, 1992, Lightbody and Irwin, 1995) these algorithms can be used to train both neural networks and thus solve the control problem. However, when a predictive control scheme is considered, besides training the neural network to represent a predictive model of the plant, one needs to solve an optimization problem to get the control action (Mills, Zomaya, Tadé, 1994, Su, McAvoy, 1993).

In this paper, previous experience in using stochastic optimal parameter estimation to solve optimization problems (Rios Neto and Pinto, 1989, Rios Neto and Cruz, 1990, Pinto and Rios Neto, 1990, Prado and Rios Neto, 1994) is explored to consider an adaptive neural predictive control scheme completely based on Kalman filtering algorithms. The problems of the associated feedforward neural network training and of control determination are both viewed and treated in an integrated way as stochastic linear parameter estimation problems. This allows to view the problem in a more general stochastic framework and to derive full non parallel and approximated parallel processing versions of control algorithms which are formally equivalent to versions of Kalman filtering previously derived and used for the problem of feedforward neural network training (Rios Neto, 1997). Analysis of these control algorithms shows that they converge to the optimized solution of performance indexes formulated to guarantee smooth and reference trajectory tracking controls.

2. PROBLEM FORMULATION AND SOLUTION SCHEME

The problem at hand is that of controlling a dynamic system:

$$\dot{x} = f(x,u) \tag{1}$$

for which discrete time nonlinear input-output models can be taken to predict approximate responses:

$$y(t_j) \cong f(y(t_{j-1}),..,y(t_{j-n_y}); u(t_{j-1}),..,u(t_{j-n_u})) \tag{2}$$

where $t_j = t + j\Delta t$.

The adopted neural predictive control scheme uses a feedforward neural network which can uniformly and with the desired accuracy learn a mapping as that of Eq.(2) (Chen and Billings, 1992) to model the dynamic system of Eq. (1). This internal model neural network then provides the response model that can be used to determine smooth and reference trajectory tracking control actions by minimizing a predictive quadratic index of performance of the type usually adopted in predictive control schemes (see, e.g., Hunt, Sbarbaro, Zbikowski and Gawthrop, 1992, and Su and McAvoy, 1993):

$$J = [\sum_{j=1}^{n}[y_r(t_j) - \hat{y}(t_j)]^T R_y^{-1}(t_j)[y_r(t_j) - \hat{y}(t_j)] + \sum_{j=0}^{n-1}[u(t_j) - u(t_{j-1})]^T R_u^{-1}(t_j)[u(t_j) - u(t_{j-1})]]/2 \tag{3}$$

where as before $t_j = t + j\Delta t$; $y_r(t_j)$ is the reference response; n defines the horizon over which the tracking errors and control increments are considered; $R_y(t_j), R_u(t_j)$ are positive definite weight matrices; $\hat{y}(t_j)$ is the output of the feedforward neural network trained to approximately model the dynamic system of Eq. (1) and which can be formally represented by:

$$\hat{y}(t_j) = \hat{f}\,(\hat{y}(t_{j-1}),..,\hat{y}(t_{j-n_y}); u(t_{j-1}),..,u(t_{j-n_u}), \hat{w}) \qquad (4)$$

where $\hat{w}$ are the neural network parameters adjusted or estimated along training.

Thus, in summary, for the solution of the resulting neural predictive control problem it is needed:

(i) to choose a feedforward neural network with appropriate architecture and size, which in a process usually involving both off line and on line supervised training can learn from dynamic system input output data sets how to represent the mapping which is a nonlinear discrete model of this dynamic system;

(ii) to solve with respect to the control actions, on line and in a small fraction of Δt the nonlinear programming problem of minimizing an objective function constraining smooth and reference trajectory tracking control actions, as that in Eq. (3), subjected to the constraint of Eq. (4).

3. KALMAN FILTERING INTEGRATED SOLUTION

The problem of supervised training of the feedforward neural network used in the predictive control scheme can be treated using Kalman filtering algorithms. Versions of this kind of algorithms with different levels of approximation can be found in the literature. These versions may vary from full non parallel algorithms, mostly suitable for off line use , to simplified parallel processing algorithms (Rios Neto, 1997) for on line use.

Exploring previously developed and related results (Rios Neto and Pinto, 1989, Rios Neto and Cruz, 1990, Pinto and Rios Neto, 1990, Prado and Rios Neto, 1994), a method is proposed where the problem of detemining the predictive control actions is also treated as one of stochastic optimal linear parameter estimation, allowing the derivation and use in a given iteration of the same Kalman filtering type of algorithms, as in the neural network training.

The method starts by assuming that the problem of control determination of Eq. (3) can be viewed in a more general stochastic framework as the following stochastic parameter problem with the output of the neural network, $\hat{y}(t_j)$, represented as in Eq. (4):

$$y_r(t_j) = \hat{f}\left(\hat{y}(t_{j-1}), \cdots, \hat{y}(t_{j-n_y}), u(t_{j-1}), \cdots, u(t_{j-n_u}), \hat{w}\right) + \nu_y(t_j) \qquad (5)$$

$$0 = u(t_{j-1}) - u(t_{j-2}) + \nu_u(t_{j-1}) \qquad (6)$$

$$E[\nu_y(t_j)] = 0; \quad E[\nu_y(t_j)\nu_y^T(t_j)] = R_y(t_j) \qquad (7)$$

$$E[\nu_u(t_j)] = 0; \quad E[\nu_u(t_j)\nu_u^T(t_j)] = R_u(t_j) \qquad (8)$$

where j = 1, 2, ..., n; noticing that $\hat{y}(t_{j-1}), \cdots, \hat{y}(t_{j-n_y})$ and $u(t_{j-1}), \cdots, u(t_{j-n_u})$ are the already happened and known delayed system responses and actions; and the erros $\nu_y(t_j)$ and $\nu_u(t_j)$ are of uncorrelated components as well as uncorrelated for different values of t_j. A first consequence of this more general stochastic framework in the treatment of the problem is that the weight matrices in the objective function (Eq. (3)) have now the meaning of covariance matrices. This certainly facilitates their definition.

In order to reiteratively treat the problem of Eqs. (5) and (6) as one of linear parameter estimation, one takes in an ith iteration the linearized approximation of Eq. (5):

$$\alpha(i)[y_r(t_j)-\bar{y}(t_j,i)]=\sum_{k=k_0}^{j-1}[\partial y(t_j)/\partial u(t_k)]_{\{\bar{u}(t_k;i)\}}[u(t_k,i)-\bar{u}(t_k,i)]+v_y(t_j) \tag{9}$$

where $k_0=\max[0,(j-n_y-n_u)]$; $0<\alpha(i)\le 1$, to be adjusted to guarantee the linear perturbation approximation hypothesis; and the partial derivatives are calculated using the backpropagation rule in the feedforward neural network that approximates the dynamic system response model (see, e.g., Chandran, 1994). This observation type of conditions are then processed taking as a priori information, based on conditions of Eq. (6), and consistently with the linearized approximation in Eq. (9), the following:

$$\boldsymbol{\alpha}(\mathrm{i})[\hat{\mathrm{u}}(\mathrm{t}_{-1})-\bar{\mathrm{u}}(\mathrm{t}_1,\mathrm{i})]=[\mathrm{u}(\mathrm{t}_1,\mathrm{i})-\bar{\mathrm{u}}(\mathrm{t}_1,\mathrm{i})]\;+\;\sum_{\mathrm{k}=0}^{1}\mathrm{v}_\mathrm{u}(\mathrm{t}_\mathrm{k}) \tag{10}$$

where l=0,1,..,n-1; i=1,2,..,I; $\hat{u}(t_{-1})$ is the estimated solution from last control step; $\alpha(i)\leftarrow\alpha(i+1)$; $\bar{u}(t_l,i+1)=\hat{u}(t_l,i)$, the approximated estimated value of $u(t_l)$ in the ith iteration; and for i=1 estimates or extrapolations of estimates of last control step are used.
For *j=1,2,...,n* and *l=0,1,...,n-1*,the problem of Eqs. (9) and (10) is one of stochastic linear parameter estimation, and in a more compact notation where:

$$U(t,i)\,\hat{=}\,[u^T(t_0,i):u^T(t_1,i):\ldots:u^T(t_{n-1},i)]^T\;;\;\hat{U}_l(t_{-1})\,\hat{=}\,\hat{u}(t_{-1})$$

it can equivalently be expressed as :

$$\alpha(i)[\hat{U}(t_{-1})-\overline{U}(t,i)]=U(t,i)-\overline{U}(t,i)+V_u(t) \tag{11}$$

$$\alpha(i)\overline{Z}^u(t,i)=\;H^u(t,i)[U(t,i)-\overline{U}(t,i)]\;+V_y(t) \tag{12}$$

where the meanings of the compact notation variables are obvious by identification of Eqs. (11) and (12) with Eqs.(10) and (9), respectively. Using a Kalman filtering estimator there results in a typical iteration:

$$\hat{U}(t,i)=\overline{U}(t,i)+\alpha(i)[\hat{U}(t_{-1})-\overline{U}(t,i)]+K(t,i)\alpha(i)[\overline{Z}^u(t,i)-H^u(t,i)[\hat{U}(t_{-1})-\overline{U}(t,i)]] \tag{13}$$

$$K(t,i)=R_u(t)H^{u^T}(t,i)[H^u(t,i)R_u(t)H^{u^T}(t,i)+R_y(t)]^{-1}\equiv$$
$$[R_u^{-1}(t)+H^{u^T}(t,i)R_y^{-1}(t)H^u(t,i)]^{-1}H^{u^T}(t,i)R_y^{-1}(t) \tag{14}$$

$$\hat{U}(t)\;=\hat{U}(t,I),\;\;\hat{R}_u(t,I)=[I_u-K(t,I)H^u(t,I)]R_u(t) \tag{15}$$

where $R_u(t),R_y(t),\hat{R}_u(t,I)$ are the error covariance matrices of $V_u(t),V_y(t),(\hat{U}(t,I)-U(t))$, respectively; and I_u an identity matrix. The control calculated with this algorithm is the minimum of the functional :

$$J(\alpha,i) = [[\alpha(i)\bar{Z}^u(t,i) - H^u(t,i)[U(t,i) - \bar{U}(t,i)]]^T R_y^{-1}(t)[\alpha(i)\bar{Z}^u(t,i) - H^u(t,i)[U(t,i) - \bar{U}(t,i)]] + [U(t,i) - \bar{U}(t,i) - \alpha(i)[\hat{U}(t_{-1}) - \bar{U}(t,i)]]^T R_u^{-1}(t)[U(t,i) - \bar{U}(t,i) - \alpha(i)[\hat{U}(t_{-1}) - \bar{U}(t,i)]]]/2 \quad (16)$$

Thus, convergence to a smooth $\hat{U}(t)$ control which will track the reference trajectory $y_r(t)$ is guaranteed since the feedforward network has the capacity of representing the dynamic system of Eq. (1) and of allowing a linearized approximation (Chen and Billings, 1992) in a ith iteration, provided a sufficiently small $\alpha(i)$ is considered. Another way of showing that convergence is guaranteed is by considering the equivalent form of algorithm of Eqs. (13) and (14):

$$\hat{U}(t,i) = \bar{U}(t,i) + [R_u^{-1}(t) + H^{u^T}(t,i)R_y^{-1}(t)H^u(t,i)]^{-1}\alpha(i)[R_u^{-1}(t)[\hat{U}(t_{-1}) - \bar{U}(t,i)] + H^{u^T}(t,i)R_y^{-1}(t)\bar{Z}^u(t,i)] \quad (13^a)$$

and noticing that this is the result of applying Newton's Method to the functional (see, e.g., Luenberger, 1984):

$$J_P = [Z^{u^T}(t)R_y^{-1}(t)Z^u(t) + [U(t) - \hat{U}(t_{-1})]^T R_u^{-1}(t)[U(t) - \hat{U}(t_{-1})]]/2 \quad (17)$$

Following a way completely analogous to that adopted for the problem of neural network training (Rios Neto, 1997), one can generate an approximated version of problem of Eqs. (11) and (12) which can be paralleled processed for each value of l=0,1,..,n-1. To get this simplified version of the problem one approximates the values of $U_k(t,i), k \neq l,$ in Eq. (12) by $\bar{U}_k(t,i)$. From these approximations results a problem which can be locally processed, and which is of the form:

$$\alpha(i)[\hat{u}(t_{-1}) - \bar{u}(t_l,i)] = u(t_l,i) - \bar{u}(t_l,i) + V_{ul}(t) \quad (17)$$

$$\alpha(i)\bar{Z}^u(t,i) = H_l^u(t,i)[u(t_l,i) - \bar{u}(t_l,i)] + V_y(t) \quad (18)$$

The use of Kalman filtering to solve this problem leads to:

$$\hat{u}(t_l,i) = \bar{u}(t_l,i) + \alpha(i)[\hat{u}(t_{-1}) - \bar{u}(t_l,i)] + K(t,l,i)\alpha(i)[\bar{Z}^u(t,i) - H_l^u(t,i)[\hat{u}(t_{-1}) - \bar{u}(t_l,i)]] \quad (19)$$

$$K(t,l,i) = [R_{ul}^{-1}(t) + H_l^{u^T}(t,i)R_y^{-1}(t)H_l^u(t,i)]^{-1}H_l^{u^T}(t,i)R_y^{-1}(t) \quad (20)$$

$$\hat{u}(t_l) = \hat{u}(t_l,I), \quad \hat{R}_{ul}(t,I) = [I_u - K(t,l,I)H_l^u(t,I)]R_{ul}(t) \quad (21)$$

If this algorithm is considered in the equivalent and usual a priori form of a stochastic linear estimator, there results:

$$\hat{u}(t_l,i) = \bar{u}(t_l,i) + [R_{ul}^{-1}(t) + H_l^{u^T}(t,i)R_y^{-1}(t)H_l^u(t,i)]^{-1}\alpha(i)[R_{ul}^{-1}(t)[\hat{u}(t_{-1}) - \bar{u}(t_l,i)] + H_l^{u^T}(t,i)R_y^{-1}(t)\bar{Z}^u(t,i)] \quad (22)$$

which is the result of applying Newton's Method to the functional of Eq. (17) but having

$$\tilde{R}_u^{-1}(t) = diag.[[R_{ul}^{-1}(t) + H_l^{u^T}(t,i) R_y^{-1}(t) H_l^u(t,i)]^{-1} : l = 0,1,...,n-1] \tag{23}$$

in place of $R_u^{-1}(t)$. Convergence to a smooth control which tracks the reference trajectory is thus also guaranteed for the parallel processing version of the control determination, as given by Eqs. (19) and (20) or Eq.(22).

3. SIMULATION AND RESULTS

As an example of application of the previously described methodology, consider the following system proposed by Chen and Khalil, 1995, and also used by Liu, Kadirkamanathan and Billings, 1998, as an example for testing a predictive control scheme:

$$y_t = \frac{2.5\ y_{t-1}\ y_{t-2}}{1 + y_{t-1}^2 + y_{t-2}^2} + 0.3\ \cos\left(0.5\left(y_{t-1} + y_{t-2}\right)\right) + 1.2\ u_{t-1} \tag{24}$$

Let the reference input signal be given by r(t) = sin(πt/500). The initial condition of the plant is given by (y_{-1}, y_{-2}) = (0, 0). Then, it is proposed to control a plant represented by Eq. (24) to track, as close as possible, the reference input signal r(t) using the proposed predictive control strategy.

To accomplish such task, a multilayer perceptron neural network with 5 neurons at input layer, 20 neurons at the hidden layer and 1 neuron at output layer was trained. Patterns for training were obtained from Eq. (24) by using random values of u_{t-1} uniformly distributed between –6.0 and 6.0. Then, the neural network was chosen with configuration:

$$\hat{y}_t = f\left(y_{t-1}, y_{t-2}, y_{t-3}, u_{t-1}\right) \tag{25}$$

Results showed that for this plant, the prediction of only one step ahead was enough for obtaining good tracking of the reference signal. Figure 1 shows examples of patterns used for neural network training. Figure 2 shows the tracking signal r(t) and output of the plant y_t and in Figure 3 it is shown the tracking error r(t) – y_t. Finally, Figure 4 shows the input control u_{t-1} used for plant control. As can be observed from Figures 2 and 3, the control action obtained by application of the proposed methodology, results in a smooth and relatively accurate tracking of the reference input by the plant. These results compare with the ones obtained by Liu, Kadirkamanathan and Billing, 1998.

4. CONCLUSIONS

The use of Kalman filtering as a tool to derive adaptive neural predictive control algorithms was explored. Viewing the solution of the optimization problem of control action determination as one of stochastic parameter estimation reduced this problem to one formally equivalent to that of estimating the weights in feedforward neural network supervised training . This allowed an integrated treatment of both problems, using Kalman filtering algorithms.

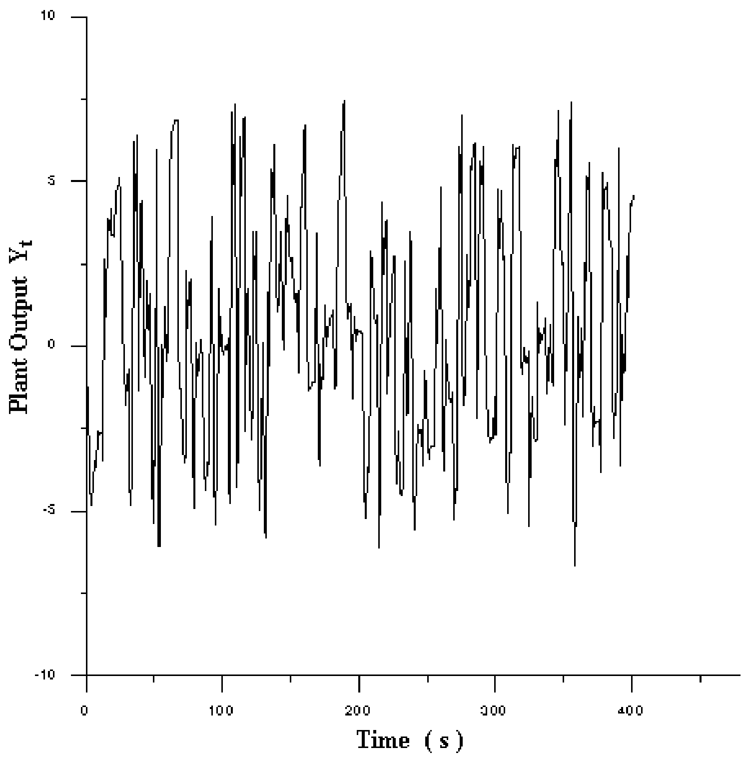

Figure 1 – Pattern Examples Used for Neural Network Training.

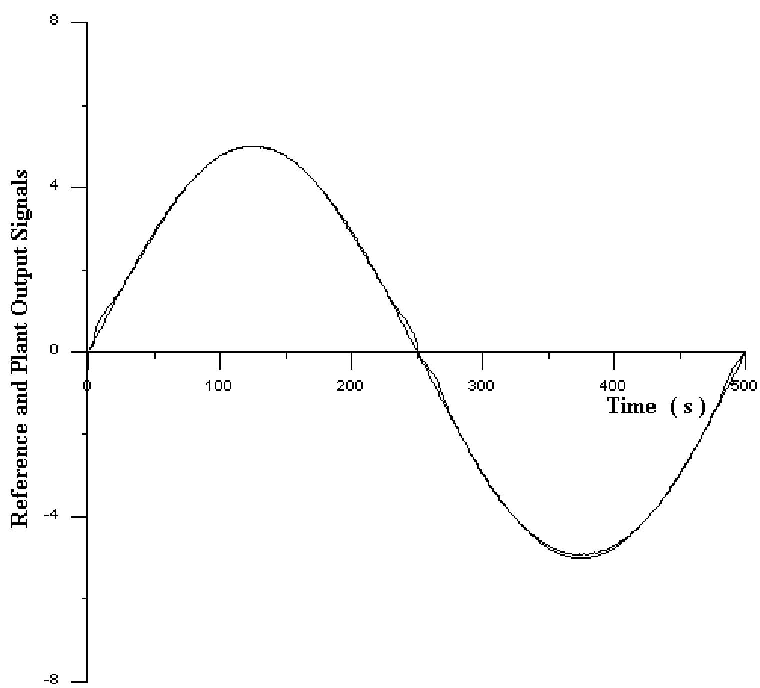

Figure 2 – Tracking and Reference Signals.

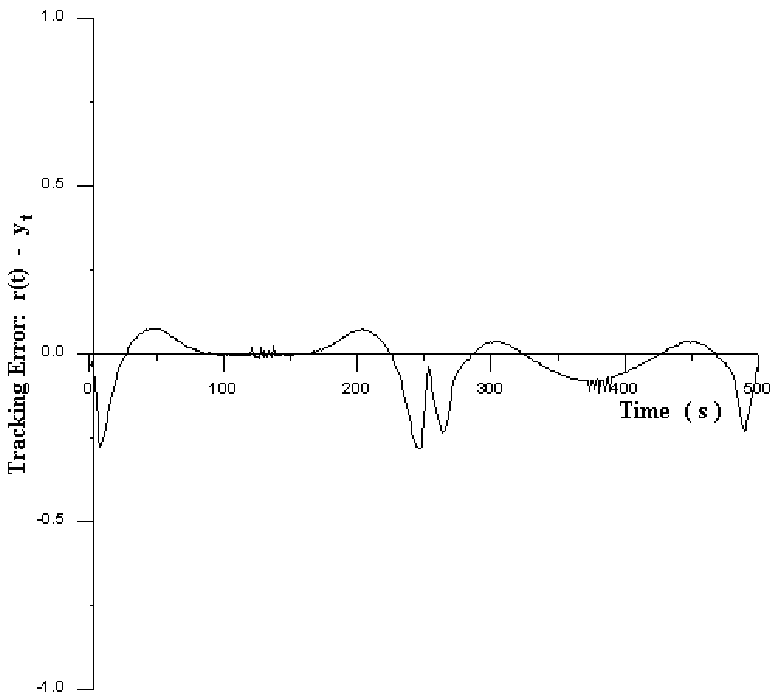

Figure 3 – Tracking Error: r(t) - y_t.

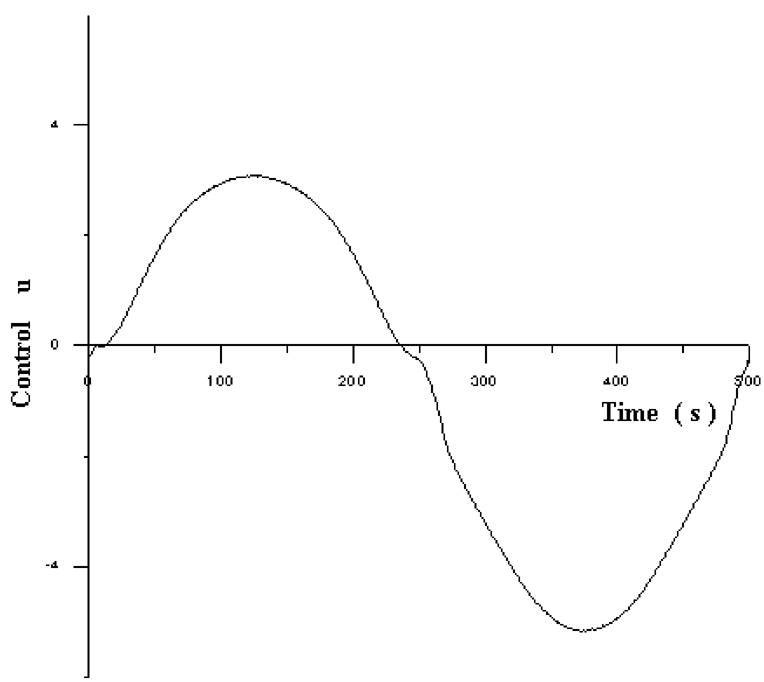

Figure 4 – Input Control Signal: u_{t-1}.

In analogy with results previously developed for feedforward neural network training [9], two versions of algorithms were developed for the control determination. The first was one where the approximation is the iterative approach due to linearization of equations , and where local parallel processing is not attained ; this version can be used with serial processing in situations where high speed of processing is available and plant time constants are not so small. The second one was an approximated version , but one which attains local parallel processing and intended for real time , adaptive control schemes.

Both versions of algorithms were shown to converge to the solution of applying Newton's Method to the minimization of functionals which constraint smooth and reference trajectory tracking controls.

The derived control Kalman filtering algorithms are expected to have a performance equivalent to that of the correspondent neural network training Kalman filtering algorithms, due to the fact that they are completely similar algorithms used to solve numerically equivalent parameter estimation problems.

ACKNOWLEDGEMENT

This research has been supported by CNPq and FAPESP grants.

REFERENCES

[1] Chandran, P. S., " Comments on Comparative Analisys of Backpropagation and the Extended Kalman Filter for Training Multilayer Perceptron", *IEEE Transactions on Pattern Analysis and Machine Inteligence*, 16(8), pp. 862-863, 1994.

[2] Chen, S. and Billings, S. A., " Neural Networks for Nonlinear Dynamic System Modelling an Identification", *Int. J. Control,*56(2), PP. 319-346, 1992.

[3] Chen, F. C. and Khalil, H. K., "Adaptive control of a class of non-linear discrete-time systems using neural networks, IEEE Transactions on Automatic Control, vol. 40, pp. 791-801, 1995.

[4] Chen, G. and Ögmen, H., "Modified Extended Kalman Filtering for Supervised Learning", *Int. J. Systems Sci.* 24(6), pp. 1207-1214, 1993.

[5] Hunt, K. J., Sbarbaro, D., Zbikowski, R. and Gawthrop, P. J., "Neural Networks for Control Systems- A Survey", *Automatica,* 28(6), pp.1083-1112, 1992.

[6] Iiguni, Y. and Sakai, H., "Real Time Learning Algorithm for a Multilayered Neural Network Based on the Extended Kalman Filter", *IEEE Transactions on Signal Processing,*40(4), PP. 959-966, 1992.

[7] Lange, F., "Fast and Accurate Training of Multilayer Perceptrons Using an Extended Kalman Filter (EKFNet)", internal paper, *DLR(German Institute Aerospace Research Establishment), Institute for Robotics and Systems Dynamics,* 1995.

[8] Lightbody, G. and Irwin, G. W., "Direct Neural Model Reference Adaptive Control", *IEE Proc. Control Theory Appl.,* 142(1), PP. 31-43, 1995.

[9] Liu, G. P., Kadirkamanathan, V. and Billings, S. A., "Predictive control for non-linear systems using neural networks", Int. Journal of Control, vol. 71, no. 6, pp. 1119-1132, 1998.

[10] Luenberger, D. G., *Linear and Nonlinear Programming,* Addison-Wesley, Reading, MA, pp. 261-262, 1984.

[11] Mills, P. M., Zomaya A. Y. and Tadé, M. O., “Adaptive Model-Based Control Using Neural Networks”, *Int. J. Control*, 60(6), pp. 1163-1192, 1994.
[12] Pinto R. L. U. F., and Rios Neto, A., “An Optimal Linear Estimation Approach to Solve Systems of Linear Algebraic Equations”, *J. Computational an Applied Mathematics,* 33, pp. 261-268, 1990.
[13] Prado, A. F. B. A. and Rios Neto, A., “Suboptimal and Hybrid Numerical Solution Schemes for Orbit Transfer Maneuvers”, *SBA: Controle e Automação,* 4(2), pp. 82-88, 1994.
[14] Rios Neto, A., “Stochastic Optimal Linear Parameter Estimation and Neural Nets Training in Systems Modeling”, *RBCM- J. of the Braz. Soc. Mechanical Sciences*, XIX(2), 138-146, 1997.
[15] Rios Neto, A. and Cruz, J. J., “Design of a Stochastic Linear Discrete Regulator Based on an Optimal State Estimation Approach”, *SBA: Controle e Automação*, 2(3), pp. 151-157, 1990.
[16] Rios Neto, A. and Pinto, R. L. U. F., “A Stochastic Approach to Generate a Projection of the Gradient Type Method”, in *Proc. VIII Congresso Latino Americano e Ibérico sobre Métodos Computacionais para Engenharia*, PUC/RJ, Rio de Janeiro, Brasil, pp. 331-345, 1989.
[17] Scalero, R. S. and Tepedelenlioglu, N., “A Fast New Algorithm for Training Feedforward Neural Networks”, *IEEE Transactions of Signal Processing,* 40(1), pp.202-210, 1992.
[18] Singhal, S. and Wu, L., “Training Multilayer Perceptrons with the Extended Kalman Algorithm”, in *Advances in Neural Information Processing Systems, VI,* Morgan Kaufman Pub. Inc., pp 136-140, 1989)
[19] Su, H. T. and McAvoy, T. J., “Neural Model Predictive Control of Nonlinear Chemical Procesesses”, in *Proc. of the 1993 International Symposium on Inteligent Control,*Chicago, U.S.A., PP. 358-363, 1993.
[20] Watanabe, K., Fukuda, T. and Tzafestas, S. G., “Learning Algorithms of Layered Neural Networks Via Extended Kalman Filters”, *Int. J. System Science,* 22(4), pp. 753-768, 1991.