NON-LINEAR PREDICTIVE CONTROL BASED ON A RECURRENT NEURAL NETWORK ¤

Paulo Gil†, Jorge Henriques†, António Dourado† and H. Duarte-Ramos‡
email: {pgil, jh, dourado}@ dei.uc.pt; hdr@mail.fct.unl.pt
†CISUC - Centro de Informática e Sistemas da Universidade de Coimbra
Informatics Engineering Department, UC
Pólo II, Pinhal de Marrocos 3030 Coimbra - Portugal
Phone: +351 39 790000, Fax: +351 39 701266
‡Electrical Engineering Department, UNL
2825 Monte da Caparica - Portugal
Phone: +351 1 2948545, Fax: +351 1 2948532

ASTRACT: Taking advantage of the recognised properties of universal approximation of neural networks, a description of a non-linear plant in the state space form is obtained by using a recurrent Elman network. Learning is implemented on-line with a truncated backpropagation through time algorithm. Based on the neural model, a predictive control strategy is implemented, being the sequence of control actions computed by solving an optimisation problem iteratively. This hybrid predictive technique is applied to a control experiment on a laboratory three-tank system and the performance compared with that of a first principle model based predictive controller. Experimental results for several operating conditions not merely demonstrate the viability of utilising neural networks with learning implemented on-line within a predictive control framework but also, given their adaptive features, they are a remarkable choice, particularly, for those systems having unmodelled dynamics.

Keywords: Predictive control; Elman networks; learning; MIMO systems; optimisation; moving horizon.

1. INTRODUCTION

Given their inherent ability to approximate any non-linear continuous function without requiring any prior knowledge, neural networks (NN), in the last few years, have increasingly deserving the attention of the control community, leading to the implementation of several neural based control strategies. One such approach incorporates NN into the predictive control scheme. This approach has the appealing advantage of handling one of the main limitations exhibited by standard model based predictive control (MPC), that is, the requirement of a suitable mathematical model embodying all the salient features of the plant. Methodologies based on feedforward neural networks (FNN) with external recurrence and off-line training have been proposed by Temeng, Schenelle and McAvoy (1995), Chen (1998), Tan and Cauwenberghe (1996). Using recurrent neural networks (RNN) Zamarreño and Vega (1996) proposed a neural non-linear adaptive control and Shaw and Doyle (1997) the Internal Model Control scheme.
Following an approach based as well on recurrent neural networks, the present work proposes a hybrid finite horizon predictive control strategy where the model of the plant is obtained with an Elman network. The learning stage is implemented on-line using a truncated backpropagation through time algorithm, while the sequence of control actions is computed by extending the predicted outputs to a multiple-step-ahead horizon.
For assessing the potentialities of this hybrid predictive approach its performance in controlling a laboratory three-tank system is compared with that of a first principle model predictive control scheme. In this methodology, at each sampling time, the model of the plant is first subjected to a local linearisation followed by a time discretisation. Next, a constrained optimisation problem is solved in order to compute the sequence of manipulated variables.
The paper is organised as follows: Section 2 presents the recurrent Elman neural architecture and the truncated backpropagation through time learning algorithm. In section 3 the neural predictive control and the first principle model predictive control strategies are described. Section 4 characterises briefly the laboratory three-tank system and discusses some experimental results obtained with both predictive schemes. Finally, the main conclusions are drawn in section 5.

¤ To be presented at ESIT'99, Orthodox Academy of Crete, Greece, June 3-4, 1999.

2. ELMAN NETWORK: MODEL AND TRAINING

For modelling, purposes the plant is assumed as described by a general non-linear discrete time state space model, as follows:

$$x(k+1) = f\,\{\,x(k), u(\mathrm{k})\,, k\} \tag{1}$$

$$y(k) = g\,\{\,x(k), u(k)\,, k\} \tag{2}$$

where $f : \Re^n \times \Re^m \times \Re \to \Re^n$ and $g : \Re^n \times \Re^m \times \Re \to \Re^q$ are non-linear functions; $u(k) \in \Re^{\mathrm{m}}$ and $y(t) \in \Re^q$ are the inputs and the outputs; $x(k) \in \Re^{\mathrm{n}}$ denotes the states, which are assumed to be directly observable.

Recurrent neural networks comprising dynamic elements and feedback connections are able to approximate to any arbitrary precision a discrete time state space description, Jin, Nikiforuk and Gupta (1995), that is the case of Elman networks, Elman (1990). In this special type of RNN the feedforward connections are modifiable, while the recurrent connections are fixed. Additionally to the input and the output units, the Elman network has a hidden unit, $x^h(k) \in \Re^{\mathrm{n}}$, and a context unit, $x^c(k) \in \Re^{\mathrm{n}}$. The interconnection matrices $W^x \in \Re^{\mathrm{n \times n}}$, $W^u \in \Re^{\mathrm{n \times m}}$ and $W^y \in \Re^{\mathrm{q \times n}}$ define, respectively, the interconnection for the context-hidden layer, for the input-hidden layer and for the hidden-output layer.

Due to practical difficulties concerning the identification of higher order systems, some modifications have been proposed. In Pham and Xing (1995) a self-connection or a feedback gain $\alpha \in \Re^+$ is incorporated in the context units, improving this way the dynamic memorisation, Figure 1.

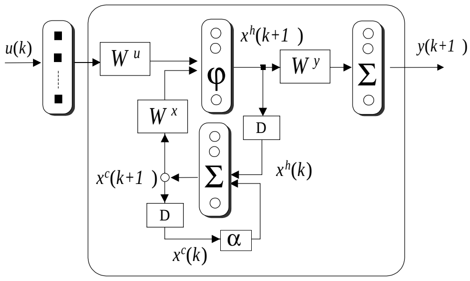


Figure 1: Block diagram of a modified Elman network.

The dynamics of the modified Elman neural network is described by the difference equations (3)-(5).

$$x^h(k+1) = \varphi\left\{W^x\, x^c(k+1) + W^u\, u(k)\right\} \tag{3}$$

$$x^c(k+1) = x^h(k) + \alpha\, x^c(k) \tag{4}$$

$$y(k+1) = W^y x^h(k+1) \tag{5}$$

where $\varphi(\cdot)$ denotes a hyperbolic tangent function. The goal of the learning stage is to find W^x and W^u such that the squared error between the output neurons and the desired states in the horizon $[k-N, \ldots, k]$ given by (6) is minimised.

$$E(k) = \sum_{\mathrm{i=k-N}}^{\mathrm{k}} e_m(i)^2 \tag{6}$$

where $e_m(k) \in \Re^{\mathrm{n}}$ is the modelling error at time step k, given by $e_m(k) = x(k) - x^h(k)$. For adjusting the weighting values, the present work follows the backpropagation through time algorithm (BTT), Werbos (1990), but with a particular simplification. The infinity backpropagation is truncated to a finite number N of prior time steps, Henriques and Dourado (1998), instead of $BTT(\infty)$.

3. NEURAL NETWORK AND FIRST PRINCIPLE BASED PREDICTIVE CONTROL

Both neural and first principle model predictive schemes are discrete time techniques where an explicit dynamic model of the plant is used to predict its outputs over the finite prediction horizon P, when the manipulated variables are appropriately changed over the finite control horizon M. For both techniques, at time step k, an optimiser computes on-line and in real-time the open-loop sequence of present and future control actions, such that, the predicted outputs follow a predefined reference, Figure 2, Garcia, Prette and Morari (1989). The first control action $u(k\,|\,k)$ in the optimal control sequence is then picked for implementation on the real plant over the interval $[k, k+1]$, being discarded the remaining sequence.

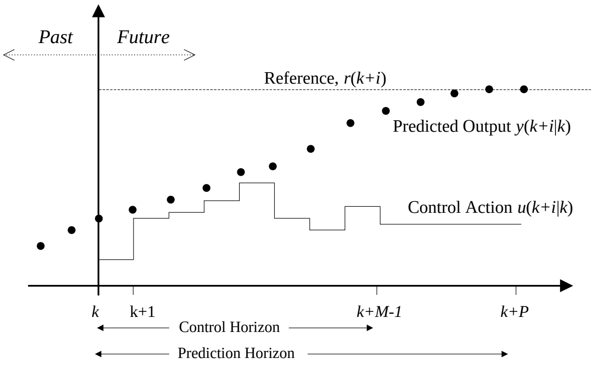


Figure 2: The moving horizon implementation of MPC.

At the next sampling time, $k+1$, the prediction and control horizons are shifted ahead by one step and a new optimisation problem is solved using the latest outputs measurements $y(k+1)$ together with the control action $u(k \mid k)$.

3.1 FIRST PRINCIPLE MODEL BASED PREDICTIVE CONTROL

Consider a time-invariant discrete-time system to be controlled of the form:

$$x(k+1) = \Phi x(k) + \Gamma u(k) + \eta \tag{7}$$

where $\Phi \in \Re^{n \times n}$ and $\Gamma \in \Re^{n \times m}$ denote the state and input matrices; $x(k) \in \Re^{n}$ and $u(k) \in \Re^{m}$ are the discrete state and control vectors; $\eta \in \Re^{n}$ is the independent term vector; k denotes an integer sampling time index.

Assuming the plant as having all states measurable and considering a 2-norm for the cost functional, as well linear constraints on the input and states, the constrained open-loop optimal control problem can be stated as follows:

$$\min_{u(k),\ldots,u(k+M-1)} J = \min_{u(k),\ldots,u(k+M-1)} \left\{ \sum_{i=1}^{P} \left\| x(k+i \mid k) - r(k+i) \right\|_{Q_i}^{2} + \sum_{i=0}^{M-1} \left[\left\| u(k+i \mid k) \right\|_{R_i}^{2} + \left\| \Delta u(k+i \mid k) \right\|_{S_i}^{2} \right] \right\} \tag{8}$$

subject to the system dynamics (7) and the following constraints:

$$\begin{aligned} x_{\min} \le x(k+i) \le x_{\max},&\ \ i = 1,\ldots,P \\ u_{\min} \le u(k+i) \le u_{\max},&\ \ i = 0,\ldots,M-1 \\ \left| \Delta u(k+i) \right| \le \Delta u_{\max},&\ \ i = 0,\ldots,M-1 \\ \left| \Delta u(k+i) \right| = 0,&\ \ i = M,\ldots,P-1 \end{aligned} \tag{9}$$

where $Q_i \in \Re^{n \times n}$, $R_i \in \Re^{m \times m}$, $S_i \in \Re^{m \times m}$, $\Delta u \in \Re^{m}$, $r(k) \in \Re^{n}$ and i is the current discrete time index.

Since the finite horizon optimal control formulation stated above is based on a quadratic cost function and assumes that all constraints are linear, then it may be rewritten as a quadratic programming problem (10) and (11).

$$\text{minimise} \qquad h^{T} \Delta\tilde{u} + \frac{1}{2} \Delta\tilde{u}^{T} H \Delta\tilde{u} \tag{10}$$

$$\text{subject to} \qquad A^{T} \Delta\tilde{u} \le b \tag{11}$$

with $A \in \Re^{mM \times (4mM + 2nP)}$, $b \in \Re^{(4mM + 2nP)}$, $\Delta\tilde{u} \in \Re^{mM}$ and $h \in \Re^{mM}$ and $H \in \Re^{mM \times mM}$ given by:

$$\begin{aligned} h_l^{\mathrm{T}} = 2 \Bigg\{ & x_o^{\mathrm{T}} \left[\sum_{i=l-1}^{P-1} \left(\Phi^{i+1}\right)^{\mathrm{T}} Q_{i+1} \sum_{q=0}^{i-l+1} \Phi^{q} \right] \Gamma + \left[\Gamma u_o + \eta \right]^{\mathrm{T}} \sum_{i=l-1}^{P-1} \left[\sum_{q=0}^{i} \left(\Phi^{q}\right)^{\mathrm{T}} Q_{i+1} \sum_{s=0}^{i-l+1} \Phi^{s} \right] \Gamma \\ & - \left[\sum_{i=l-1}^{P-1} \left(r(i+1)\right)^{\mathrm{T}} Q_{i+1} \sum_{q=0}^{i-l+1} \Phi^{q} \right] \Gamma + u_o^{\mathrm{T}} \sum_{i=l-1}^{M-1} R_i \Bigg\}, \ \ l = 1,\ldots, M \end{aligned} \tag{12}$$

$$\begin{aligned} H_{ll} &= 2 \left\{ \Gamma^{\mathrm{T}} \sum_{i=0}^{P-1} \left[\sum_{q=0}^{i} \left(\Phi^{q}\right)^{\mathrm{T}} Q_{i+1} \sum_{q=0}^{i} \Phi^{s} \right] \Gamma + \sum_{i=l-1}^{M-1} R_i + S_{l-1} \right\}, \ \ l = 1,\ldots, M \\ H_{lp} &= 2 \left\{ \Gamma^{\mathrm{T}} \sum_{i=l-1}^{P-1} \left[\sum_{q=0}^{i-l+1} \left(\Phi^{q}\right)^{\mathrm{T}} Q_{i+1} \sum_{q=0}^{i-p+1} \Phi^{q} \right] \Gamma + \sum_{i=l-1}^{M-1} R_i \right\}, \ \ l, p = 1,\ldots, M;\ \ p \ne l \end{aligned} \tag{13}$$

3.2 NON-LINEAR NEURAL PREDICTIVE CONTROL

In the neural predictive controller case, the manipulate variables are computed iteratively by solving (8), using a gradient descent algorithm, and assuming $R_i = 0$, $Q_i = I$ and S_i a constant weighting matrix:

$$u^{i+1}(k \mid k) = u^{i}(k \mid k) - \mu_p \frac{\partial J(\cdot)}{\partial u^{i}(k \mid k)} \tag{14}$$

where $\mu_p \in \Re^+$ is the optimisation step. The derivative of the cost function (14) is given by:

$$\frac{\partial J(\cdot)}{\partial u(k+j \mid k)} = 2\left\{\sum_{i=1}^{P} e_p(k+i \mid k)\frac{\partial x(k+i \mid k)}{\partial u(k+j \mid k)} - S\left[u(k+j-1 \mid k) - 2u(k+j \mid k) + u(k+j+1 \mid k)\right]\right\} \tag{15}$$

where $j = 0,\ldots,M-1$; $i = 1,\ldots,P$; $e_p(k+i \mid k) \in \Re^n$ is the prediction error. The derivative in (15) is given by:

$$\frac{\partial x(k+i \mid k)}{\partial u(k+j \mid k)} = \begin{cases} 0 & \text{if } i \le j \\ \displaystyle\sum_{l=1}^{i-1} \frac{\partial x(k+i \mid k)}{\partial x(k+l \mid k)} \frac{\partial x(k+l \mid k)}{\partial u(k+j \mid k)} + \frac{\partial x(k+i \mid k)}{\partial u(k+j \mid k)} & \text{if } i > j \end{cases} \tag{16}$$

4. EXPERIMENTS

4.1 THE LABORATORY THREE-TANK SYSTEM

The three-tank system used for experiments comprises three plexiglas cylinders, being one of the tanks (T_3) connected with the other two tanks by the means of circular cross section pipes equipped with manually adjustable ball valves, Figure 3. At T_2 is located the main outlet of the plant connecting it to the collecting reservoir in a similar way. Additionally, each tank is provided with a direct connection to the reservoir. Two of the tanks, T_1 and T_2, are fed with liquid, which is pumped from the reservoir by two diaphragm pumps.

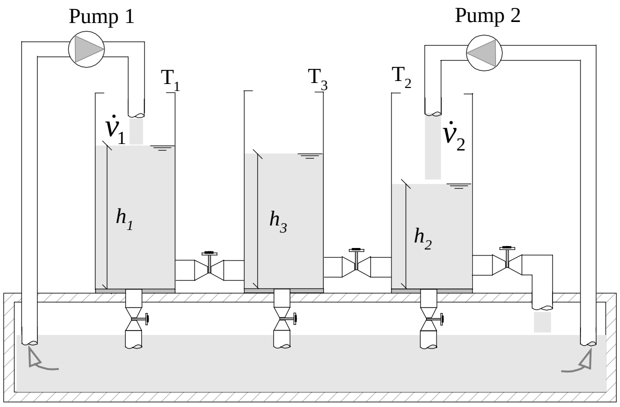


Figure 3: Structure of the laboratory Three-Tank system.

The mathematical model can be described by the following three non-linear equations:

$$\begin{aligned} \frac{dh_1}{dt} &= \frac{1}{A_T}\left[\dot{v}_1 - \zeta_{13} S_t \,\mathrm{sgn}(h_1 - h_3)\sqrt{2g\left|h_1 - h_3\right|}\right] \\ \frac{dh_2}{dt} &= \frac{1}{A_T}\left[\dot{v}_2 + \zeta_{32} S_t \,\mathrm{sgn}(h_3 - h_2)\sqrt{2g\left|h_3 - h_2\right|} - \zeta_{20} S_t \sqrt{2gh_2}\right] \\ \frac{dh_3}{dt} &= \frac{1}{A_T}\left[\zeta_{13} S_t \,\mathrm{sgn}(h_1 - h_3)\sqrt{2g\left|h_1 - h_3\right|} - \zeta_{32} S_t \,\mathrm{sgn}(h_3 - h_2)\sqrt{2g\left|h_3 - h_2\right|}\right] \end{aligned} \tag{17}$$

where $h_i, i = 1,2,3$, is the tank level; $\dot{v}_i, i = 1,2$, is the flow rate; A_T and S_T are the cross sections of each tank and the interconnecting pipes; $\zeta_{ij} \in [0, 1]$, with $j = 0$ referring the collecting reservoir; g is the gravity.

4.2 EXPERIMENTAL RESULTS

To assess comparatively the performance of both controllers a set of experiments was carried out. In the neural network predictive control scheme a modified Elman network was implemented with two inputs and three hidden units. The number of hidden units and context units were chosen as 3 and previous knowledge in the form of a linear state space equation was considered for the initialisation. Regarding the controller parameters, they were chosen as: prediction horizon $P = 3$, control horizon $M = 1$, weighting matrix $S = 0.01 I_2$ and optimisation step $\mu_p = 0.1$.

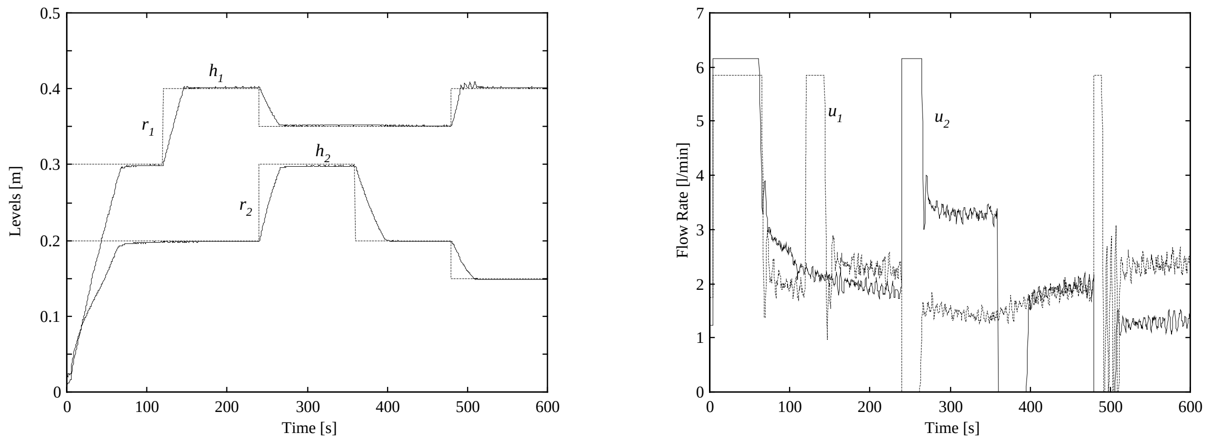
For implementing the first principle model-based predictive control scheme the non-linear continuous-time plant dynamics (17) is first subjected to a local linearisation, followed by a discrete-time approximation, assuming a sampling time of 0.5 second, the same for both strategies. The weighting matrices in the objective functional (8) were chosen as, $R_i = 0.5 I_2$, $S_i = 0.1 I_2$ and Q_i given by:

$$Q_i = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 0 \end{bmatrix}, i < P; \quad Q_i = \begin{bmatrix} 100 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 0 \end{bmatrix}, i = P$$

The open-loop constrained optimal control problem, (10) and (11), is solved with The MATLAB function, `qp`, choosing for prediction horizon $P = 5$ and control horizon $M = 1$. Since each pump flow rate and the liquid levels are bounded, the following inequality constraints should be imposed in solving the optimisation problem:

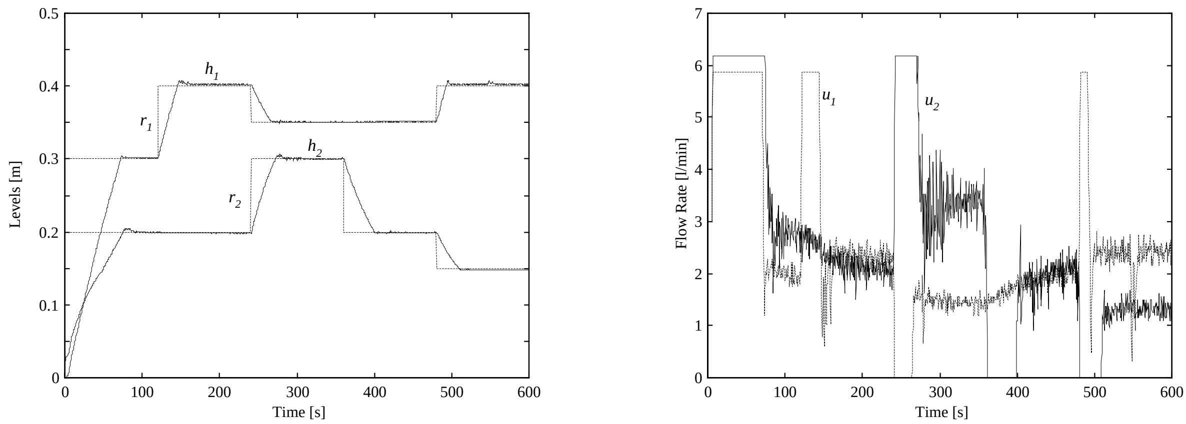
$$[0 \quad 0 \quad 0]^T \le x \le [0.6 \quad 0.6 \quad 0.6]^T \ \ m\,; \ [0 \quad 0]^T \le u \le [97.50 \quad 102.67]^T \times 10^{-6} \ \ m^3 s^{-1}; \ |\Delta u| \le [16 \quad 16]^T \times 10^{-6} \ \ m^3 s^{-1}$$

In Figure 4 is depicted the set-point trajectory, the plant outputs and the control actions for the neural network predictive controller run, while Figure 5 shows analogous results for the first principle model predictive controller.



(a) (b)

Figure 4: Neural predictive control (a) set-point trajectory and outputs; (b) control actions.



(a) (b)

Figure 5: First principle model predictive control (a) set-point trajectory and outputs; (b) control actions.

As can be seen from Figure 4 and Figure 5, both controllers are able to follow adequately well the set-point trajectory, regardless the set-points values and the difference in level between each reference. However, for the first principle predictive technique the control error is closer to zero and the transient response is slightly faster than the neural predictive control system. These features are reflected in the corresponding control actions, where is detected a higher magnitude in the input increments. Nevertheless, selecting carefully the weighting matrices could, in principle, prevent this somehow jumpy behaviour.

5. CONCLUDING REMARKS

A practical application of a neural model predictive control where the plant dynamics is captured by a modified Elman network is presented in this work. The training is performed on-line using a truncated backpropagation through time algorithm, while the control action is obtained solving iteratively, at each time step, an optimisation problem. The performance of this technique is compared with that of a first principle predictive controller.
Results obtained from a control experiment on a laboratory three-tank system show that both approaches perform satisfactory. Yet, at least for the operating conditions and the tuning parameters selected, the neural controller leads to a smother functioning of the actuators, but with the drawback of a little larger deviation.
Given the adaptive features of the neural predictive control scheme, its application for controlling plants with unmodelled dynamics or subjected to structural changes seems to be a viable or even preferable choice.

ACKNOWLEDGEMENTS

This work was partially supported by the Portuguese Ministry of Science and Technology (MCT), under the program PRAXIS XXI.

REFERENCES

Chen, J. (1998). Systematic derivations of model predictive control based on artificial neural networks, *Chemical Eng. Communications* **164**: 35-39.

Elman, J. (1990). Finding Structure in time, *Cognitive Science* **14**: 1789-211.

Garcia, C., Prett, D. and Morari, M. (1989). Model Predictive Control: Theory and Practice – a Survey, *Automatica* **25**(3): 335-348.

Henriques J., and Dourado, A. (1998). A Multivariable adaptive control using a recurrent neural network, *Proceedings of Eann98 - Engineering Applications of Neural Networks*, Gibraltar, pp 118-121.

Jin, L., Nikiforuk, P. and Gupta, M. (1995). Approximation of discrete time state space trajectories using dynamic recurrent networks, *IEEE Trans. Automatic Control* **40**(7): 1266-1270.

Pham D. and Xing, L. (1995). Dynamic System Identification Using Elman and Jordan Networks, *Neural Networks for Chemical Engineers*, Editor A. Bulsari, Chap. 23.

Shaw, A. and Doyle, F. (1997). Multivariable non-linear control applications for a high purity distillation column using a recurrent dynamic neuron model, *Journal of Process Control* **7**(4): 255-268.

Tan, Y. and Cauwenberghe, A. (1996). Non-linear one step ahead control using neural networks: control strategy and stability design, *Automatica* **32**(12): 1701-1706.

Temeng, H., Schenelle, P. and McAvoy, T. (1995). Model predictive control of an industrial packed bed reactor using neural networks, *J. Proc. Control* **5**(1): 19-28.

Werbos, P. (1990). Backpropagation trough time, what it does and how do it, *Proc. IEEE* **78**: 1550-1560.

Williams, R. and Zipser, D. (1996). Gradient-based learning algorithms for recurrent networks and their computational complexity, *Backpropagation: Theory, architectures and application*s, Edit. by Yves Chauvin and D. Rumelhart, Chap.13: 433-486.

Zamarreño, J. and Vega, P. (1996). Identification and predictive control of a melter unit from the sugar industry using recurrent neural nets, *Proceedings of Eann96 – Engineering Applications of Neural Networks*, pp. 119-126.