Non-linear Multivariable Predictive Control Neural versus First Principle Modelling Approach.
IASTED Control and Applications, May 20-22, Cancun, Mexico, 2002.

Nonlinear Multivariable Predictive Control: Neural *versus* First Principle Modelling Approach

Jorge Henriques[†], Paulo Gil[†‡], António Dourado[†] and H. Duarte-Ramos[‡]
(jh, pgil, 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, 3030 Coimbra – Portugal

[‡]Electrical Engineering Department, UNL, 2825 Monte da Caparica - Portugal

Abstract
A neural network predictive control scheme is compared with a first principle model predictive control strategy when controlling a three tanks system. The neural network approach involves a recurrent Elman network for capturing the plant's dynamics being the learning stage implemented on-line using a modified version of the back-propagation through time algorithm. In the first principle model predictive control scheme a real-time open-loop linear constrained optimisation problem is solved with a standard quadratic programming algorithm.
Experimental results collected from the non-linear plant are presented.

Keywords: Model predictive control; Elman networks; receding horizon control; MIMO systems; optimisation.

1. INTRODUCTION

Model predictive control (MPC) has received a great deal of attention in the past years as valuable tool for controlling industrial processing plants. The fundamental concepts concerning model predictive control can be traced back to 1963 and the work of Propoi where the moving horizon approach was for the first time proposed [2]. Yet, it was not until the late 1970s and the contribution of Richalet and co-authors [3], that, for the first time, has been reported and succeeded a model predictive implementation in the process control field, particularly for linear plants.

Recently, there have been some attempts to extend MPC techniques to non-linear systems. One such approach is based on successive linear approximations of the plant [4]. Oliveira [5] in order to ensure the stability of the control system incorporates into the optimisation problem an extra contraction constraint, initially proposed by Polak and Yang [6], leading to the so-called contractive MPC. Other possible way for achieving stability in the regulation problem considers a finite horizon terminal constraint that forces all states to be zero at the end of the prediction horizon, [7]. Another alternative approach is proposed by Nevistić and Morari [8] where a combination of feedback linearisation and model predictive control is considered.

The success of the MPC technology is attributed to three relevant factors [1]. First and foremost is the incorporation of an explicit model into the control computation. This allows the controller to deal with the most significant features of the plant, depending on the accuracy of the mathematical model. Secondly, because it predicts future plant behaviour, the effects either of feedforward and feedback disturbances can be anticipated and hence adequately rejected. Finally, MPC methodologies have the potential of dealing with either input and output constraints in an explicit way during the design and implementation stages.

In many cases, as result of the complexity of the non-linear plant, a good enough model is not available either by economic issues or because some parameters are not accessible. In this circumstances one should turn to plant identification methodologies in order to come up with a feasible model of the plant.

Given the learning capabilities of the neural networks (NN) [9], these structures can be used as non-linear black-box models of the plant. In the context of neural predictive control methodologies, several works have been reported. Donat *et al.* [10] applied a multilayer feedforward neural network (FNN) trained with a back-propagation algorithm together with an optimisation problem that is solved with a sequential quadratic programming method. Hao *et al.* [11] combine a FNN with the one-step-ahead predictive control scheme, being the current and future inputs obtained by means of a standard Quasi-Newton non-linear optimisation algorithm. Temeng *et al.* [12] has proposed a hybrid multivariable non-linear predictive control based on a FNN model and on the Fletcher variable metric method for solving the optimisation problem. Draeger *et al.* [13] incorporated a FNN for non-linear prediction in an extended standard dynamic matrix control algorithm. More recently, Chen [14] has proposed a three-layer FNN with hyperbolic tangent functions using the Levenberg-Marquardt algorithm in minimising a cost functional. Using a FNN as well, Tan and Cauwenberghe [15] presented a non-linear one-step-ahead control strategy being the control action evaluated by a gradient descent algorithm.

In this paper a neural predictive control is compared with a standard MPC scheme. The neural approach is based on a recurrent Elman network for modelling purposes and the learning procedure is performed on-line by means of a modified back-propagation through time algorithm. Instead of one-step-ahead prediction, as proposed in [15], a sequence of control actions is computed by extending the prediction outputs to a multi-step-ahead horizon. In what the first principle model predictive control scheme concerns, the non-linear real plant dynamics, is linearised each discrete time. Next, a standard constrained quadratic programming problem is solved in order to obtain a sequence of current and future manipulated variables. In both approaches only the first control action is fed to the plant.

2. NEURAL PREDICTIVE CONTROL AND MPC

In the MPC technique, also known as receding horizon control (RHC), an explicit dynamic model of the plant is used to predict its outputs over some specified finite prediction horizon P, when the control actions are accordingly changed over some finite control horizon M. The predictive control approach is a discrete time technique where, at time step k, an optimisation procedure computes on-line and in real-time the open-loop sequence of present and future control moves $\{u(k|k), \ldots, u(k+M-1|k\}$, such that the predicted outputs follow a predefined trajectory and taking into account constraints on the outputs and on the inputs. Only the control action $u(k|k)$ is fed to the real plant over the time interval $[k, k+1]$. At the next sample time $k+1$, the prediction and control horizons are shifted ahead by one step and a new optimisation problem is solved. Thus, by repeatedly solve an open-loop optimisation problem with every initial conditions updated at each time step, the model predictive control strategy results in a closed-loop constrained optimal control technique.

The general open-loop optimisation problem can be formulated as

$$\min_{\mathrm{u(t)}} V = \min_{\mathrm{u(t)}} \int_{t_k}^{t_k+P} L\big(x(t), u(t), t\big)\, dt \tag{1}$$

subject to the system dynamics (2) and constraints (3).

$$\begin{aligned} \dot{x}(t) &= f^{\,p}\big(x(t), u(t), t\big) \\ y(t) &= g^{\,p}\big(x(t), u(t), t\big) \end{aligned} \tag{2}$$

$$\begin{aligned} l\big(x(t), u(t)\big) &= 0 \\ c\big(x(t), u(t)\big) &\geq 0 \end{aligned} \tag{3}$$

$f^p{:}\Re^n\times\Re^m\to\Re^n$ and $g^p{:}\Re^n\times\Re^m\to\Re^q$ are twice continuous differentiable; L is the Lagrangian; $u(t)\in\Re^m$, $y(t)\in\Re^q$ and $x(t)\in\Re^n$ are, respectively, the inputs, outputs and states; $c(\cdot)$ and $l(\cdot)$ are functions characterising equality and inequality constraints; t_k is the discrete current time. Depending on the cost function, the dynamic model and the constraints involved, several particular formulations can be stated in terms of the general formulation given above.

2.1 Linear MPC Formulation

Consider the following linear model of the plant (2):

$$x\big(k+1\big) = \Phi x\big(k\big) + \Gamma u\big(k\big) + \eta \tag{4}$$

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

Assuming all the states of the plant measurable and considering a 2-norm as a performance index and linear constraints on the input and states, the open-loop optimal receding horizon control problem can be stated as follows

$$\begin{aligned} \min_{u(k),\ldots,u(k+M-1)} J = \min_{u(k),\ldots,u(k+M-1)} & \left\{ \sum_{i=1}^{P} \left\| x\big(k+i\,|\,k\big) - r\big(k+i\big) \right\|_{Q_i}^2 \right\} \\ + \left\{ \sum_{i=0}^{M-1} \left[\left\| u\big(k+i\,|\,k\big) \right\|_{R_i}^2 + \left\| \Delta u\big(k+i\,|\,k\big) \right\|_{S_i}^2 \right] \right\} & \end{aligned} \tag{5}$$

subject to the system dynamics (4) and the following constraints:

$$\begin{aligned} x_{\min} \leq x\big(k+i\big) &\leq x_{\max},\ i = 1,\ldots,P \\ u_{\min} \leq u\big(k+i\big) &\leq u_{\max},\ i = 0,\ldots,M-1 \\ \left| \Delta u\big(k+i\big) \right| &\leq \Delta u_{\max},\ i = 0,\ldots,M-1 \\ \left| \Delta u\big(k+i\big) \right| &= 0,\ i = M,\ldots,P-1 \end{aligned} \tag{6}$$

where $\|\cdot\|_\Omega$ is the weighted Euclidean norm; $Q_i\in\Re^{n\times n}$, $R_i\in\Re^{m\times m}$ and $S_i\in\Re^{m\times m}$ denote positive definite, symmetric weighting matrices; $\Delta u\in\Re^m$ is the predicted control action increment; $r(k)\in\Re^n$ denotes the discrete reference trajectory; i is the current discrete time index. This optimisation problem can be formulated as quadratic programming problem (QP) where the functional cost to be minimised is obtained by means of a second-order Taylor series approximation to $J(\cdot)$. The new optimisation problem can then be stated as:

$$\textit{minimise} \qquad h^T \Delta\tilde{u} + \frac{1}{2} \Delta\tilde{u}^T H \Delta\tilde{u} \tag{7}$$

$$\textit{subject to} \quad A^T \Delta\tilde{u} \leq b \tag{8}$$

with $A\in\Re^{mM\times(4mM+2nP)}$, $b\in\Re^{(4mM+2nP)}$. The gradient vector and the Hessian matrix of functional $J(\cdot)$ are respectively $h\in J(\cdot)\in\Re^{mM}$ and $H\in J(\cdot)\in\Re^{mM\times mM}$. The extended incremental input vector $\Delta\tilde{u}\in\Re^{mM}$ is the vector of decision variables. If the Hessian matrix is positive definite then the QP format cost function is strictly convex and consequently $\Delta\tilde{u}^*$ is the unique global minimum for the optimisation problem.

2.2 Non-linear Neural Predictive Control

For modelling purposes the plant is described by the following non-linear discrete time state space equations.

$$x(k+1) = \phi_f\left\{ x(k), u(\mathrm{k}), k \right\} \tag{9}$$

$$y(k) = \phi_{\mathrm{g}}\left\{ x(k), u(k), k \right\} \tag{10}$$

where $\Phi_f{:}\Re^n\times\Re^m\times\Re\to\Re^n$ and $\Phi_g{:}\Re^n\times\Re^m\times\Re\to\Re^q$ are non-linear functions; $u(k)\in\Re^m$, $y(k)\in\Re^q$ and $x(k)\in\Re^n$ are, respectively, the inputs, outputs and states at a discrete time k. As in the MPC formulation, the states are assumed to be directly observable.

2.2.1 Elman Networks Topology

Recurrent neural networks comprising dynamic elements and feedback connections are suitable for approximating dynamical systems [16]. One of the useful characteristics of these topologies is that they can represent any order of delays implicitly rather than explicitly, as in feed-forward networks with external recurrence. For recurrent networks it was proved that they may be used to approximate to any arbitrary precision a discrete time state space description, Jin *et al.* [16]. One of the simplest RNN is the Elman network.

Elman [17] proposed a partially recurrent network where the feed-forward connections are modifiable and the recurrent connections are fixed. Additionally to the input and the output units, the Elman network has a hidden unit, $x^h(k)\in\Re^n$, and a context unit, $x^c(k)\in\Re^n$. The interconnection matrices $W^x\in\Re^{n\times n}$, $W^u\in\Re^{n\times m}$ and $W^y\in\Re^{q\times n}$ are the interconnection weights for the context-hidden layer, for the input-hidden

layer and for the hidden-output layer. In the original architecture the context layer only holds a copy of the activation of the hidden units from the previous time step, being the trace of the entire history accumulated in the context unit. Due to practical difficulties related to the identification of higher order systems, some modifications have been proposed. In [18] a self-connection or a feedback gain $\alpha \in \Re^+$ in the context units is incorporated, as depicted in Figure 1, improving the dynamic memorisation ability of the network.

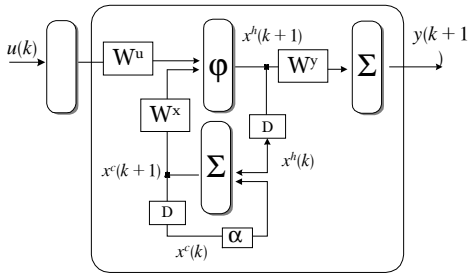


Figure 1 – Block diagram of the modified Elman network.

The dynamics of the modified Elman neural network is described by the difference equations (11)-(14).

$$s(k+1) = W^x\, x^c(k+1) + W^u\, u(k) \tag{11}$$

$$x^h(k+1) = \varphi\{s(k+1)\} \tag{12}$$

$$x^c(t+1) = x^h(t) + \alpha\, x^c(t) \tag{13}$$

$$y(k+1) = W^y x^h(k+1) \tag{14}$$

The function $f(x) = x$ is used for the output layer, $s(k) \in \Re^n$ is an intermediate variable and $\varphi(\cdot)$ is the hyperbolic tangent function. If an augmented state, $\bar{x}(k) \in \Re^{2n}$, is defined by

$$\bar{x}(k) = \begin{bmatrix} x^h(k) \\ x^c(k) \end{bmatrix} \tag{15}$$

equations (11)-(14) can be rewritten as

$$\bar{x}(k+1) = \varphi\left\{\bar{x}(k), u(k), \mathrm{W}^{\mathrm{x}}, \mathrm{W}^{\mathrm{u}}\right\} \tag{16}$$

$$y(k+1) = g\left\{\bar{x}(k+1), W^{\mathrm{y}}\right\} \tag{17}$$

These equations can be interpreted as a state space model, analogous to the non-linear system dynamics defined in (9) and (10).

2.2.2 Learning Methodology

Assuming that $x_d(k) \in \Re^n$ denotes the plant state at time step k, the goal is to find W^x and W^u (W^y is known and fixed) such that the squared error between the output neurons and the desired states in the horizon $[k-N, \ldots, k]$, defined is minimised.

$$E(k) = \sum_{\mathrm{i=k-N}}^{\mathrm{k}} e_m(i)^2 \tag{18}$$

where $e_m(k) \in \Re^n$ is the modelling error at time k given by (19).

$$e_m(k) = x_d(k) - x^h(k) \tag{19}$$

Several training algorithms have been proposed to adjust the weighting values in recurrent networks. Examples of these methods are the Narendra's dynamic back-propagation [19], the real time recurrent algorithm from Williams and Zipser [20] and the back-propagation through time algorithm (BTT) from Werbos [21], which is being considered in the present work. All these methods use a gradient based learning algorithm and involve the computation of partial derivatives or sensitivity functions. To updated W^x and W^u equation (20) is used.

$$\Delta W^i(k+1) = \rho_{\mathrm{m}} \Delta W^i(k) - \mu_m(1-\rho_m) \frac{\partial E(k)}{\partial W^i(k)}, \quad i = x,u \tag{20}$$

$\mu_m \in \Re^+$ is a learning rate; $\rho_m \in \Re^+$ is a momentum term; the increment ΔW is evaluated according to

$$\Delta W^i(k+1) = W^i(k+1) - W^i(k), \quad i = x,u \tag{21}$$

For evaluating W^x and W^u a simplification of the $BTT(\infty)$ algorithm is considered in this work by truncating the infinity back-propagation of information to a finite number (N) of prior time steps [22]. A value of α near to 1 enables the context unit to remember more past information and a value near to 0 let the context unit to forget rapidly past data, similarly to a forgetting factor. The BTT algorithm is based on an extension of the standard back-propagation for feed-forward networks. A recurrent network is expanded into a multilayer FNN, being a new layer added for each time step. The computation of the gradient in equation (20), is accomplished with (22) and (23)

$$\frac{\partial E(k)}{\partial W^x} = \sum_{\mathrm{i=k-N}}^{\mathrm{k}} \delta^h(i)\, x^c(i)^{\mathrm{T}} \tag{22}$$

$$\frac{\partial E(k)}{\partial W^u} = \sum_{\mathrm{i=k-N}}^{\mathrm{k}} \delta^h(i)\, u(i-1)^{\mathrm{T}} \tag{23}$$

where the sensitivity to the units of the net inputs $\delta^h(k) \in \Re^+$ is computed recursively for $i = k$ to $i = k - N$ according to:

$$\varepsilon^{\mathrm{h}}(i) = e_{\mathrm{m}}(i) + \delta^{\mathrm{c}}(i+1) \tag{24}$$

$$\delta^{\mathrm{h}}(i) = \varepsilon^{\mathrm{h}}(i) \otimes \varphi'\{s(i)\} \tag{25}$$

$$\varepsilon^c(k) = \alpha\, \delta^{\mathrm{c}}(k+1) + W^{\mathrm{x}\,\mathrm{T}} \delta^{\mathrm{h}}(k) \tag{26}$$

$$\delta^{\mathrm{c}}(i) = \varepsilon^c(i) \tag{27}$$

The process starts at time k with

$$\varepsilon^{\mathrm{h}}(k) = e_{\mathrm{m}}(k) = x_d(k) - \mathrm{x}^{\mathrm{h}}(k) \tag{28}$$

$$\delta^{\mathrm{h}}(k) = \varepsilon^{\mathrm{h}}(k) \otimes \varphi'\{s(k)\} \tag{29}$$

$$\varepsilon^c(k_N) = W^{\mathrm{x}\,\mathrm{T}} \delta^{\mathrm{h}}(k) \tag{30}$$

The symbol $\otimes$ denotes an element to element multiplication and $\varphi'(\cdot)$ is the derivative of the hyperbolic tangent function.

2.2.3 Neural Network-based Predictive Control

The general principle underlying neural predictive control is the same as that of any standard MPC technique, that is, the existence of an explicit model of the plant and the computation of a manipulated variables sequence by solving an optimisation problem.

The control action sequence $\{u(k|k), \ldots, u(k+M-1|k\}$ is evaluated such that the predicted time response has certain desirable features according to a pre-defined design criterion, as equation (5), with $R_i = 0$, $Q_i = I_m$ and S_i a constant weighting matrix. Since $J(\cdot)$ is non linearly dependent on the control action sequence, the minimisation of the criterion proceeds iteratively, as in the neural learning stage. For the i^{th} iteration, the gradient descent algorithm is evaluated at each instant k, according to

$$u^{i+1}(k\,|\,k) = u^{i}(k\,|\,k) - \mu_p \frac{\partial J(\cdot)}{\partial u^{i}(k\,|\,k)} \tag{31}$$

where $\mu_p \in \Re^{+}$ is the optimisation step. The derivative of the cost function (31) with respect to the present and future inputs $\{u(k|k), \ldots, u(k+M-1|k\}$ is given by

$$\frac{\partial J(\cdot)}{\partial u(k+j\,|\,k)} = 2\left\{\sum_{i=1}^{P} e_p(k+i\,|\,k)\frac{\partial x(k+i\,|\,k)}{\partial u(k+j\,|\,k)}\right\} - \left\{S\left[u(k+j-1\,|\,k) - 2u(k+j\,|\,k) + u(k+j+1\,|\,k)\right]\right\} \tag{32}$$

where $j = 0, \ldots, M-1$; $i = 1, \ldots, P$; $e_p(k+1|k) \in \Re^{n}$ is the prediction error at instant $k+i$, given by:

$$e_p(k+i\,|\,k) = x(k+i\,|\,k) - \mathrm{r}(k+i\,|\,k) \tag{33}$$

Based on the sequence of manipulated variables provided by the optimiser, the model evaluates the trajectory of the predicted outputs over the prediction horizon. These predictions are fed into the optimiser again, where the objective function is evaluated and a new sequence of manipulated variables is calculated. This iterative task continues until the convergence is reached.

3. The Laboratory Three-Tanks System

The three-tanks system used in the experiments comprises three plexiglas cylinders, being one of the tanks (T3) connected to the other two tanks by means of circular cross section pipes equipped with manually adjustable ball valves, Figure 2. At tank T2 is located the main outlet of the plant, which is connected to the collecting reservoir in a similar way by a circular cross section pipe and a outflow ball valve. Additionally, each tank is provided with a straight connection to the reservoir in order enabling the simulation of leaks. Two of the tanks, namely, T1 and T2, are fed with liquid, usually distilled water, pumped from the reservoir by two diaphragm pumps with fixed piston stroke and driven by a DC motor. In order to measure the current liquid level in each of the tanks, the plant is equipped with piezoresistive differential pressure transducers.

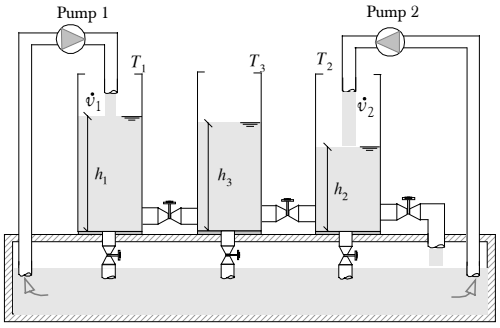


Figure 2 - Structure of the laboratory Three-Tank system.

The system dynamics can be described by the following non-linear equations:

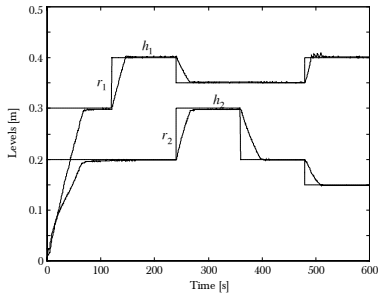
$$\begin{aligned}
\frac{dh_1}{dt} &= \frac{1}{A_T}\left[\dot{v}_1 - \zeta_{13} S_t\, \mathrm{sgn}(h_1 - h_3)\sqrt{2g|h_1 - h_3|}\,\right] \\
\frac{dh_2}{dt} &= \frac{1}{A_T}\left[\dot{v}_2 + \zeta_{32} S_t\, \mathrm{sgn}(h_3 - h_2)\sqrt{2g|h_3 - h_2|} - \zeta_{20} S_t \sqrt{2gh_2}\,\right] \\
\frac{dh_3}{dt} &= \frac{1}{A_T}\left[\zeta_{13} S_t\, \mathrm{sgn}(h_1 - h_3)\sqrt{2g|h_1 - h_3|} - \zeta_{32} S_t\, \mathrm{sgn}(h_3 - h_2)\sqrt{2g|h_3 - h_2|}\,\right]
\end{aligned} \tag{34}$$

where h_i, $i = 1,2,3$, is the tank level; $\dot{v}_i, i = 1,2$, denotes the flow rate; A_T and S_T are the cross section of each tank and the interconnecting pipes cross section, respectively; $\zeta_{ij} \in [0,1]$ is a dimensionless flow coefficient, with index $j = 0$ identifying the collecting reservoir; g is the gravity; *sgn* stands for the sign of a given argument.

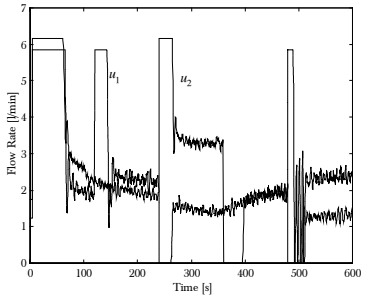
4. Experimental Results

In order to compare the performance of both predictive controllers a set of experiments was carried out on the laboratory plant. With respect to the neural strategy, an Elman network with two inputs ($m = 2$) and three outputs ($n = 2$) was implemented. The number of hidden units and context units is the same as the number of the plant states, $n = 3$. For the identification task, the following parameters are chosen as: learning rate μ_m=0.02, momentum ρ_m=0.4, self-connection α=0.6, window size N=4. Concerning the neural predictive scheme, the prediction horizon was P=3, the control horizon M=1 and the optimisation step was μ_p=0.1. The weighting matrices, Q_i and S_i are chosen as: $Q_i = I_3$, $S_i = 0.02 \times I_2$. In the first principle model-based predictive controller the weighting matrices Q_i, R_i and S_i in the objective functional (5) were chosen as:

$$Q_i = \begin{cases} \begin{bmatrix} 10 & 0 & 0 \\ 0 & 50 & 0 \\ 0 & 0 & 0 \end{bmatrix} \Leftarrow i < P \\ \begin{bmatrix} 100 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 0 \end{bmatrix} \Leftarrow i = P \end{cases} \qquad R = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix} \qquad S = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix} \tag{35}$$

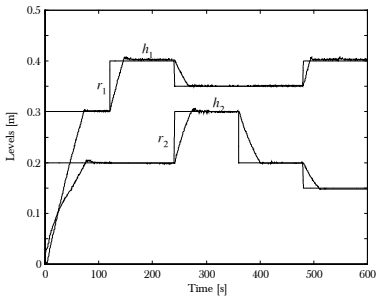


a)

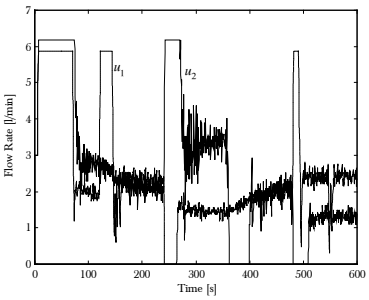


b)

Figure 3 – Neural predictive control; (a) set-point trajectory and outputs; (b) control actions.



a)



b)

Figure 4 – First principle model predictive control; (a) set-point trajectory and outputs; (b) control actions.

The open-loop constrained optimal control problem, is solved by choosing for prediction horizon P=5 and for control horizon M=1. Since each pump flow rate and liquid level for each of the three tanks must be not only positive but also upper bounded, inequality constraints should be imposed in solving the optimal control problem.

$$\begin{bmatrix}0 & 0 & 0\end{bmatrix}^T \le x \le \begin{bmatrix}0.6 & 0.6 & 0.6\end{bmatrix}^T \quad m$$

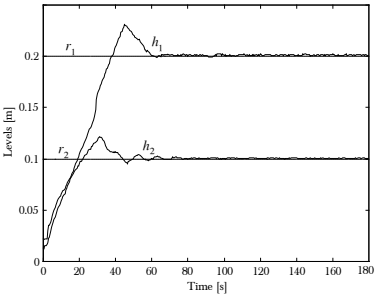
$$\begin{bmatrix}0 & 0\end{bmatrix}^T \le u \le \begin{bmatrix}97.50 & 102.67\end{bmatrix}^T \times 10^{-6} \quad m^3 s^{-1}$$

In order to prevent from excessive input changes, increments on the control actions are not merely penalised by the weighting matrix S in (5) but also constrained to

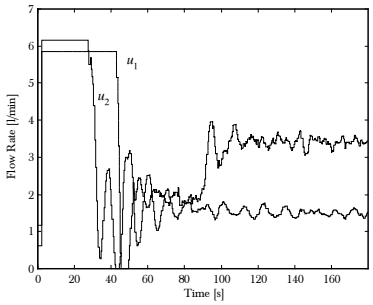
$$\left|\Delta u\right| \le \begin{bmatrix}16 & 16\end{bmatrix}^T \times 10^{-6} \quad m^3 s^{-1} .$$

In Figure 3 it is depicted the set-point trajectory, the plant outputs and the control actions for the neural network predictive controller while in Figure 4 it is shown the results for the first principle model predictive. As can be seen from these figures both controllers are able to follow adequately the set-point trajectory, regardless the set-points. However, in the first principle predictive controller case the control error is closer to zero. In addition, the transient response exhibited by this control system is slightly faster than the neural predictive control system. These features are reflected in the corresponding control actions, where it is observed a higher magnitude for the input changes. Nevertheless, this somehow nervous behaviour could be prevented by an adequate modification of the weighting matrices, particularly S.

In order to assess the robustness of each controller, a disturbance at instant 90 second was implemented in tank T1 by opening partially the valve connecting this tank to the reservoir. Additionally, no previous knowledge was incorporated into the neural predictive controller, being the initialisation implemented randomly (weighting matrices $\in [0,1]$). The control system response for the disturbed plant is shown in Figure 5 and Figure 6, respectively, for the neural and first principle controllers.

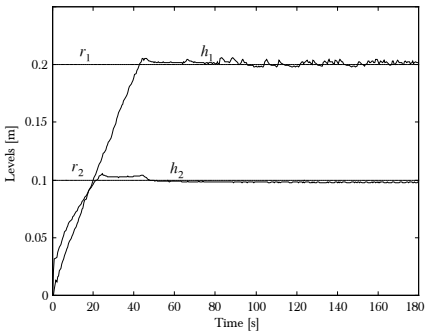


a)

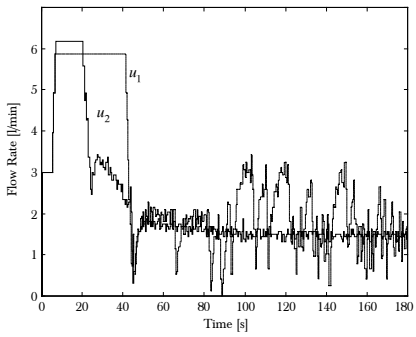


b)

Figure 5 – Neural predictive control; (a) set-point trajectory and outputs; (b) control actions.



a)



b)

Figure 6 – First principle model predictive control; (a) set-point trajectory and outputs; (b) control actions.

With respect to the neural predictive controller, since no previous knowledge is provided there exits an initial time (approximately 60 second) for which the network is gathering updated data from the plant and gradually improving the its dynamics. After this initial period, the deviation from the set-point is tiny even after the application of the disturbance. For the first principle model predictive controller the disturbance is also quite well rejected despite an higher magnitude of the input increments in the corresponding actuator (pump1). This somehow oscillating behaviour in pump 1 is not so explicitly observed for the neural controller due to the adaptive nature of this strategy.

5. CONCLUSIONS

In this work two finite horizon predictive techniques were applied to the control of a laboratory plant. One is based on an Elman network model for capturing the system dynamics while the other is based on a mathematical physical model and takes into account constraints on the inputs and the outputs.

The results show that both approaches are quite viable even in the presence of an additional disturbance. However, for the operating conditions and the tuning parameters chosen, the neural approach leads to a smother functioning of the actuators, but with a slightly larger deviation, particularly for free disturbances. When the plant is likely to change its dynamics the adaptive neural network within a predictive control technique demonstrates to be preferable

Acknowledgements

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

REFERENCES

[1] Qin, S. and Badgwell, T. - An Overview of industrial model predictive control applications – *Presented at Nonlinear MPC workshop*, Ascona, Switzerland, June 2-6, 1998.

[2] Garcia, C., Prett, D. and Morari, M. - Model Predictive Control: Theory and Practice – a Survey. *Automatica*, vol 25, nº3, 335-348, 1989.

[3] Richalet, J., Rault, A., Testud, J. and Papon, J. - Model predictive heuristic control: applications to an industrial process - *Automatica*, 14, 413-428, 1978.

[4] Garcia, C. - Quadratic dynamic matrix control of nonlinear process: an application to a batch reactor process - *AIChE Annual meeting,* San Francisco, 1984.

[5] Oliveira, S. - Model predictive control (MPC) for constrained nonlinear systems – *PhD Thesis*, CIT, Pasadena, CA, 1996.

[6] Polak, E. and Yang, T. - Moving horizon control of linear systems with input saturation and plant uncertainty - Part 1/2 - *Int. J. Control,* 58 (3), 613-638, 1993.

[7] Kwon, W. and Pearson, A. - A modified quadratic cost problem and feedback stabilisation of a linear system –*IEEE Trans. Automatic Control,* 22 (5), 838-842, 1977.

[8] Nevistić, V., and Morari, M. – Constrained control of feedback-linearizable systems – *Proceedings of ECC95*, Rome, 1726-1731,1995.

[9] Hornik, K., Stinchombe, M. and White, H. - Multilayer feedforward networks are universal approximators - *Neural Networks,* 2, 359-366, 1989.

[10] Donat, S., Bhat, N. and McAvoy, T. - Neural net based model predictive control - *Int. J. Control*, 54, 1453-1468, 1991.

[11] Hao, J., Tan, S. and J. Vandewalle - One step ahead predictive control of nonlinear systems by neural networks - *Proceedings of International Joint Conference on Neural Networks* - 2761-2764, 1993.

[12] Temeng, H., Schenelle, P. and McAvoy, T. - Model predictive control of an industrial packed bed reactor using neural networks - *J. Proc. Cont*, vol 5, nº1, 19-28, 1995.

[13] Draeger, A., Engell, S. and Ranke H. - Model Predictive control using neural networks - *IEEE Control Systems,* 61-66, October 1995.

[14] Chen, J. - Systematic derivations of model predictive control based on artificial neural networks - *Chemical Eng. Communications*, vol 164, 35-39, 1998.

[15] Tan, Y. and Cauwenberghe, A. - Nonlinear one step ahead control using neural networks: control strategy and stability design - *Automatica*, vol 32, nº 12, 1701-1706, 1996.

[16] Jin, L., Nikiforuk, P. and Gupta, M. - Approximation of discrete time state space trajectories using dynamic recurrent networks - *IEEE Trans. Automatic Control*, 40, nº 7, 1266-1270, 1995.

[17] Elman, J. - Finding Structure in time - *Cognitive Science*, 14, 1789-211, 1990.

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

[19] Narendra, K. and Parthasarathy, K. - Gradient methods for the optimization of dynamical systems containing neural networks - *IEEE Trans. on Neural Networks*, 2, nº 2, 252-262, 1991.

[20] Williams, R. and Zipser, D. - Gradient-based learning algorithms for recurrent networks and their computational complexity - *Backpropagation: Theory, architectures and applications*, Edit by Yves Chauvin and D. Rumelhart, Chap.13, 433-486, 1995.

[21] Werbos, P. - Backpropagation trough time, what it does and how do it - *Proc. IEEE*, 78, 1550-1560, 1990.

[22] Henriques J., and Dourado, A. - A Multivariable adaptive control using a recurrent neural network *Proceedings of Eann98 - Engineering Applications of Neural Networks*, Gibraltar, 9-12 June, 118-121, 1998.