A HYBRID NEURAL-DECOUPLING POLE PLACEMENT CONTROLLER AND ITS APPLICATION

J. Henriques, A. Dourado

CISUC - Centre for Informatics and Systems of the University of Coimbra
Departamento de Engenharia Informática, Pólo II
3030 COIMBRA, PORTUGAL
Fax: +351 39 701266 email: jh,dourado@dei.uc.pt

Abstract

A hybrid control architecture is proposed integrating recurrent dynamic neural networks into the pole placement context. The neural network topology involves a modified recurrent Elman network to capture the dynamics of the plant to be controlled, being the learning phase implemented on-line using a truncated backpropagation through time algorithm. At each time step the neural model, modelling a general non-linear state space system, is linearized to produce a discrete linear time varying state space model. Once the neural model is linearised some well-established standard linear control strategies can be applied. In this work the design of a decoupling pole placement controller is considered at each instant, which combined with the on-line learning of the network results in a self-tuning adaptive control scheme. Experimental results collected from a laboratory three tank system confirm the viability and effectiveness of the proposed methodology.

Keywords: Hybrid methods, recurrent neural networks, pole placement, decoupling, multivariable adaptive control.

1 Introduction

The evolution of automatic control in the last ten years has been characterised by a certain antagonism between two schools: the one based on the analytical-algebraic approach and the other based on information processing tools issued from artificial intelligence. Both have worked to develop control systems for complex, non-linear, hardly modelled processes. The analytical-algebraic school, using rigorous methods of linear and non-linear systems, built a coherent body of knowledge but still fails to solve problems where it is not possible to obtain sufficiently precise processes and disturbances models. The other school, based on neural networks and fuzzy systems, developed a high number of methods and control architectures that solve efficiently some difficult problems, but the resulting body of knowledge lacks coherence, systematisation and generality.

It is now becoming clear that only the collaboration of the two schools may lead to a new stage in the automatic control science and technology. In recent years several works contributed to integrate the two approaches in an embracing hybrid philosophy. For example, Cao et al. [3] developed a way to analyse and design complex control systems using a combination of fuzzy logic and modern control theory, including qualitative and quantitative knowledge in an unique mathematical framework (importing concepts from robust control theory and from linear uncertain systems to analyse and design fuzzy control systems, using Lyapunov theory for stability analysis). Shaw and Doyle [14] used neural control for MIMO systems by input-output linearization in a IMC structure, also for non-linear predictive control. Wang and Wu [19] use neuro computation of feedback gain matrices for the pole assignment problem. Jagannathan and Lewis [8] faced the non-linear identification task by neural networks for mapping non-linear functions of the identification error equation. Fuh and Tung [7] studied robust stability of fuzzy control systems through the Popov-Lyapunov approach, transforming the fuzzy systems into a Lur's system with uncertainties and non-linearities. Lygeros [10] develops a framework for hybrid systems, extending techniques both from fuzzy systems and conventional adaptive control. Tanaka et al. [18] use Takagi-Sugeno fuzzy models with fuzzy state feedback for eigenvalues assignment, obtaining a fuzzy regulator and a fuzzy observer, systematically designed by linear matrix inequalities and Lyapunov theory. In the sequence of this work, Ma et al. [11] presented and proved a fuzzy separation principle for controller-observer synthesis. Chen and Chang [4] rethink the sliding mode control in a fuzzy approach, obtaining a hybrid controller inheriting the advantages of both.

The present work intends to be a contribution in this direction. It propoes a control architecture combining a ecurrent Elman neural network model with a self-tuning po in particular, Elman networks can be interpreted as a non-linear state space model, so the use of this network topology for modelling purposes seems to be quite natural in the control context. At every operating point the neural model is linearized and a standard linear discrete state space model is obtained. Then a state feedback controller is synthesised for pole placement and decoupling, resulting in an adaptive control scheme. To assess its potentialities this hybrid control scheme is applied to a non-linear multivariable three-tank system.

The paper is organised as follows. In section two, the modified Elman type RNN used to model the plant is presented. Next, in the third section, the synthesis of the pole placement controller and decoupler is faced. In section four the laboratory three-tank system is briefly characterised and some experimental results are presented showing the effectiveness of the proposed methodology. Finally, in section five, some conclusions are stated.

2 System Identification Using Elman Networks

For modelling purposes it is assumed that the plant to be controlled is described by a multivariable discrete time non-linear state space, equations (1) and (2).

$$x(k+1) = f\{x(k); u(k)\} \qquad (1)$$

$$y(k) = g\{x(k)\} \qquad (2)$$

where $f \in \Re^p \times \Re^n \rightarrow \Re^n$ and $g \in \Re^n \rightarrow \Re^q$ are non-linear functions; $u(k) \in \Re^p$ and $y(k) \in \Re^q$ are, respectively, the input and the output vector, at a discrete time k; $x(k) \in \Re^n$ denotes the state vector, assumed to be directly observable.

2.1 Modified Elman Network

Due to some of its features, like the ability to approximate a discrete time non-linear system, and its interpretation as a state space model, a modified Elman network is considered here. Elman [5] has proposed a partially recurrent network, where the feedforward connections are modifiable and the recurrent connections are fixed. Theoretically, an Elman network with n hidden units is able to represent a n^{th} order dynamic system. However, due to pratical difficulties with the identification of high order systems, some modifications have been proposed. In Pham and Xing [13] a self-connection $\alpha \in \Re^+$ in the context unit is introduced, improving the network memorisation ability. In figure 1 the block diagram of this modified Elman network is depicted. Additionally to the input and the output, an 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 are $W^x \in \Re^{n\times n}$, $W^u \in \Re^{n\times p}$ and $W^y \in \Re^{q\times n}$, respectively for the context-hidden layer, input-hidden layer and hidden-output layer.

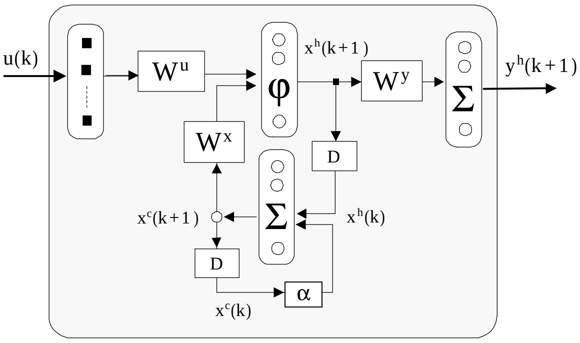

Figure 1: Block diagram of the modified Elman network.

The dynamics of the neural network is described by the following difference equations.

$$o(k+1) = W^x x^c(k+1) + W^u u(k) \qquad (3)$$

$$x^h(k+1) = \varphi\{o(k+1)\} \qquad (4)$$

$$x^c(k+1) = \alpha x^c(k) + x^h(k) \qquad (5)$$

$$y^h(k) = W^y x^h(k) \qquad (6)$$

where $o(k) \in \Re^n$ is an intermediate variable and $\varphi\{\cdot\}$ is an hyperbolic tangent function, given by (7).

$$\varphi(x) = \frac{e^{ax} - e^{-ax}}{e^{ax} + e^{-ax}}; \qquad a \in \Re^+ \qquad (7)$$

If an augmented state $\overline{x}(k) \in \Re^{2n}$, is defined by (8)

$$\overline{x}(k) = \begin{bmatrix} x^h(k) \\ x^c(k) \end{bmatrix} \qquad (8)$$

the equations (3), (4) and (5) can be rewritten by (9) and (10),

$$\overline{x}(k+1) = \varphi\{\overline{x}(k); u(k); W^x; W^u\} \qquad (9)$$

$$y^h(k) = g\{\overline{x}(k); W^y\} \qquad (10)$$

which can be seen as a non-linear state space model, similar to the system defined by (1) and (2). The first n elements of $\overline{x}(k)$ corresponding to the hidden states and the last elements to the context states, resulting in a non-minimal state dimension. Additionaly, since a measurable state problem is considered, the matrix W^y is assumed known and fixed. Therefore the goal of the learning stage consists in finding the unknown matrices W^x and W^u.

2.2 Learning Methodology

The main difficulty related to the training of recurrent networks arises from the fact that the output of the network and its partial derivatives with respect to the weights depend on the inputs (since the beginning of the training process) and on the initial state of the network. Therefore, a rigorous computation of the gradient, which implies taking into account all the past history, is not practical. In this work, however, the gradient is approximated considering a finite number of previous sampling periods. The training is defined on a sliding window mode, where at each time step k the identification criterion in the horizon $[k - N, \ldots, k]$ is defined as (11):

$$E^{total}(k) = \frac{1}{2} \sum_{l=k-N} e(l)^2 \qquad (11)$$

The modelling error, $e(k) \in \Re^n$, is given by (12),

$$e(k) = d(k) - x^h(k) \qquad (12)$$

where $d(k) \in \Re^n$ denotes the actual plant states at time step k.

Several training algorithms have been proposed to adjust the weight values in recurrent networks. Examples of these methods are the Narendra's dynamic backpropagation [12], the real time recurrent algorithm of Williams and Ziepser [21] and the Werbos' backpropagation trough time [20], among others. The backpropagation through time is considered in the present work. Being a gradient type algorithm the updated of weight values $W^s; s = x; u$ (W^y is known and fixed) are given by (13).

$$\Delta w^s_{ij} = \eta \Delta w^s_{ij} - \mu (1 - \eta) \frac{\partial E^{total}(k)}{\partial w^s_{ij}} \qquad (13)$$

where $w_{ij} \in \Re$ is the weigth connecting from j^{th} unit at time k to i^{th} unit at time $k + 1$, $\mu \in \Re^+$ is a learning rate and $\eta \in \Re^+$ is an additional momentum term. The recurrent network is expanded into a multilayer feedforward network, where a new layer is added at each time step. The computation of the derivatives is then performed as in a standard feedforward backpropagation network case, [15], according to (14) and (15).

$$\frac{\partial E^{total}(k)}{\partial w^x_{ij}} = \sum_{l=k-N} \delta^h_i(l)\, x^c_j(l) \qquad (14)$$

$$\frac{\partial E^{total}(k)}{\partial w^u_{ij}} = \sum_{l=k-N} \delta^h_i(l)\, u_j(l - 1) \qquad (15)$$

The values of $\delta^h(k) \in \Re^n$ are computed recursively for $l \in [k - N; \ldots; k]$ and for $i = 1; \ldots; n$ according to (16)-(19).

$$\varepsilon^h_i(l) = e_i(l) + \delta^c_i(l + 1) \qquad (16)$$

$$\delta^h_i(l) = \varepsilon^h_i(l) \cdot \varphi' \{o_i(l)\} \qquad (17)$$

$$\varepsilon^c_i(l) = \alpha \delta^c_i(l + 1) + \sum_{s=1} w^x_{si} \delta^h_s(l) \qquad (18)$$

$$\delta^c_i(l) = \varepsilon^c_i(l) \qquad (19)$$

The process starts at time k with

$$\varepsilon^h_i(k) = e_i(k) = d_i(k) - x^h_i(k) \qquad (20)$$

$$\delta^h_i(k) = \varepsilon^h_i(k) \cdot \varphi' \{o_i(k)\} \qquad (21)$$

$$\varepsilon^c_i(k) = \delta^c_i(k) = \sum_{s=1} w^x_{si} \delta^h_s(k) \qquad (22)$$

3 Control Strategy

3.1 Elman Network Linearisation

A well known technique to deal with non-linear control systems is based on a linearization of the non-linear plant model. At a given operating point, a nominal linear model of the plant is obtained and the controller is design using some well-established standard linear control strategy.

In the context of neural networks linearisation there has been some works. Ahmed and Tasaddup [1] proposed a control scheme based on the linearisation of a feedforward neural network model of the plant. The training is performed off-line and a time-varying linear controller (gain schedule) is designed based on the linearised plant model at each operating point. Sorenson [16] has shown the possibility to on-line extract, from a neural model, the actual linearized parameters by taking the derivatives of the outputs with respect to the inputs. Using the features of this strategy for parameter estimation, a conventional pole placement adaptive controller can be adopted for non-linear control. Suykens et al. [17] proposed a linear fractional transformation representation, making possible to interpret a non-linear neural network as a nominal linear model. This linear model is then used in the design of a standard robust control scheme.

In this work a linear model is obtained by a Taylor expansion of the non-linear neural model around an operating point. Combining equations (3)-(5) a non-linear equation that describes the neural behaviour can de obtained as (23):

$$x^h(k + 1) = \varphi \left\{ W^x x^h(k); \alpha W^x x^c(k); W^u u(k) \right\} \qquad (23)$$

Following the Taylor expansion and neglecting higher order terms the linear model is given by (24).

$$x^h(k + 1) \approx \frac{\partial \varphi\{\cdot\}}{\partial x^h(k)} x^h(k) + \frac{\partial \varphi\{\cdot\}}{\partial x^c(k)} x^c(k) + \frac{\partial \varphi\{\cdot\}}{\partial u(k)} u(k) \qquad (24)$$

where

$$\frac{\partial \varphi\{\cdot\}}{\partial x^h(k)} = \mathrm{diag}[\varphi'\{\cdot\}]\, W^x = \overline{A} \qquad (25)$$

$$\frac{\partial \varphi\{\cdot\}}{\partial x^c(k)} = \alpha\, \mathrm{diag}[\varphi'\{\cdot\}]\, W^x = \alpha\, \overline{A} \qquad (26)$$

$$\frac{\partial \varphi\{\cdot\}}{\partial u(k)} = \mathrm{diag}[\varphi'\{\cdot\}]\, W^u = \overline{B} \qquad (27)$$

Using the augment state, defined by (8), a linear neural state space model can be rewritten by (28) and (29),

$$\overline{x}(k+1) = W^A\, \overline{x}(k) + W^B\, u(k) \qquad (28)$$

$$y^h(k) = W^C\, \overline{x}(k) \qquad (29)$$

The matrices $W^A \in \Re^{2n \times 2n}$, $W^B \in \Re^{2n \times p}$ and $W^C \in \Re^{q \times 2n}$ are defined respectively by equations (30), (31) and (32).

$$W^A = \begin{bmatrix} \overline{A} & \alpha\overline{A} \\ I_{n;n} & \alpha I_{n;n} \end{bmatrix} \qquad (30)$$

$$W^B = \begin{bmatrix} \overline{B} \\ \bot_{n;p} \end{bmatrix} \qquad (31)$$

$$W^C = \begin{bmatrix} \overline{C} & \bot_{q;n} \end{bmatrix} = \begin{bmatrix} W^\dagger & \bot_{q;n} \end{bmatrix} \qquad (32)$$

where $\bot$ and I represent, respectively, a zero and an identity matrix of appropriate dimensions.

Once obtained a linear model, several standard control strategies can be employed. In the present work the control parametres are evaluated from a decoupling pole placement algorithm. At each time step the neural model parameters are updated using the truncated backpropagation through time algorithm, being this neural model linearised to produce a discrete time linear state space model suitable for linear pole placement control. In figure 2 is depicted the resulting adaptive self-tuning control scheme.

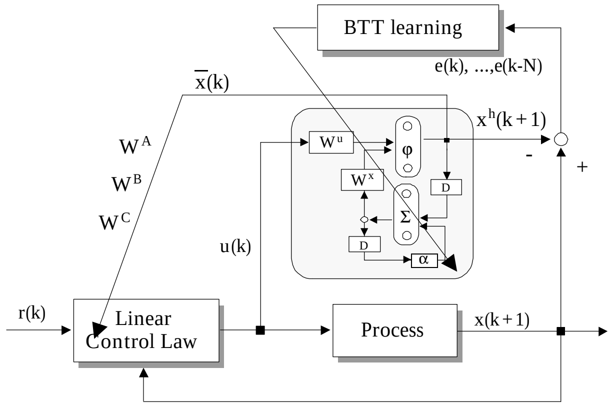

Figure 2: Adaptive control structure using a RNN.

3.2 Multivariable Decoupling Pole Placement Linear Control

Assuming the system described by a linear state space model, a standard state feedback control law can be given by (33)

$$u(k) = G\, r(k) - F\, \overline{x}(k) \qquad (33)$$

where $r(k) \in \Re^q$ is the set-point vector (q is the number of outputs). The matrices $F \in \Re^{q \times 2n}$ and $G \in \Re^{q \times q}$ are computed by a pole placement law and such that the i^{th} input $r_i(k)$ affects only the output $y_i(k)$. Falb and Wollovith [6] have established this decoupling pole placement control law. Let $d_1, \ldots, d_q$ be defined by (34) $-$ (35).

$$d_i = \min\left\{ j : C_i A^j B \neq 0 \right\} \qquad (34)$$

or

$$d_i = n - 1 \text{ if } C_i A^j B = 0 \text{ for all } j \qquad (35)$$

The F and G matrices are computed respectively by (36) and (37)

$$F = B^{*-1} \left[\sum_{i=0} \{ M_i C A_i + A^* \} \right] \qquad (36)$$

$$G = B^{*-1} \qquad (37)$$

where

$$A^* = \begin{bmatrix} C_1 A^{d_1+1} \\ \ldots \\ C_q A^{d_q+1} \end{bmatrix} \qquad B^* = \begin{bmatrix} C_1 A^{d_1} B \\ \ldots \\ C_q A^{d_q} B \end{bmatrix} \qquad (38)$$

The constant $\lambda = \max\{d_i\}$ and the matrices $M_i \in \Re^{q \times q}$ are suitably chosen to specify the closed-loop poles locations.

4 Control of a Three-Tank System

4.1 Process Description

The DTS200 three-tank system [2] is a non-linear system composed by three plexiglas tanks interconnected in series by two connecting pipes (figure 3). The liquid leaving T_2 is collected in a reservoir from which pumps 1 and 2 supply the tanks T_1 and T_2. The three tanks are equipped with piezo-resistive pressure transducer for measuring the level of the liquid (usually distilled water) $h_1(k)$, $h_2(k)$ and $h_3(k)$.

The connecting pipes and the tanks are additionally equipped with manually adjustable valves for the purpose of simulating clogs as well as leaks. A digital controller manipulates the flow rates $u_1(k)$ and $u_2(k)$, respectively, for pump 1 and pump 2. The goal of the control system is to control the levels in tanks T_1 and T_2, respectively $h_1(k)$

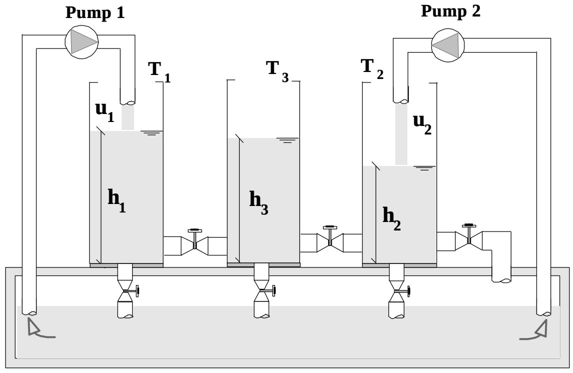

Figure 3: Schematic diagram of the three-tank system.

and $h_2(k)$, by adjusting the flow rates $u_1(k)$ and $u_2(k)$. For this particular plant the relationship between the state variables (level values) and outputs is given by $y_1(k) = x_1(k) = h_1(k)$ and $y_2(k) = x_2(k) = h_2(k)$: Therefore the assumed known output matrix $\overline{C} = W^y$, can be defined by (39).

$$W^y = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \qquad (39)$$

4.2 Modelling and Control Specifications

For modelling purposes the laboratory three-tank system is assumed to be described by a third order non-linear state space discrete time model ($n = 3$), equation (1). It has two inputs ($p = 2$) and two outputs ($q = 2$). For the identification task the following parameters were used: learning rate ¹ = 0:01; momentum ´ = 0:6; self-connection ® = 0:6; window size $N = 4$; $a = 1{:}4$.

A previous knowledge about the process was assumed, in form of a linear state space equation which was used to initialise the weights of the RNN. Regarding the controller parameters, the F and G matrices are computed such the desired pole locations were the same for both sub-system and located at z=0.8.

4.3 Experimental Results

In order to assess the performance of the proposed hybrid methodology a set of experiments was carried out with the laboratory plant. These experiments have been conducted using a PC being the algorithms implemented in C code. For sampling time was chosen 1.5 seconds and to prevent possible long training times, the maximum number of iterations in the learning task, in each sampling time, was limited to 20.

In figures 4, 5 and 6 can be observed the performance of the proposed strategy with respect to a set-point tracking problem. Figure 4 shows the desired set-point trajectory and the corresponding output levels. For this particular experiment it can be concluded that a very acceptable control performance is obtained by combining on-line a non-linear neural model estimation with a standard linear pole placement controller. In figure 5 is depicted the corresponding control actions.

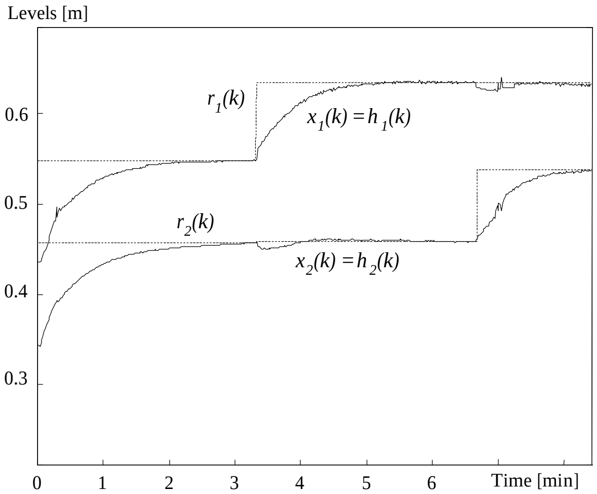

Figure 4: Set-point trajectory and outputs.

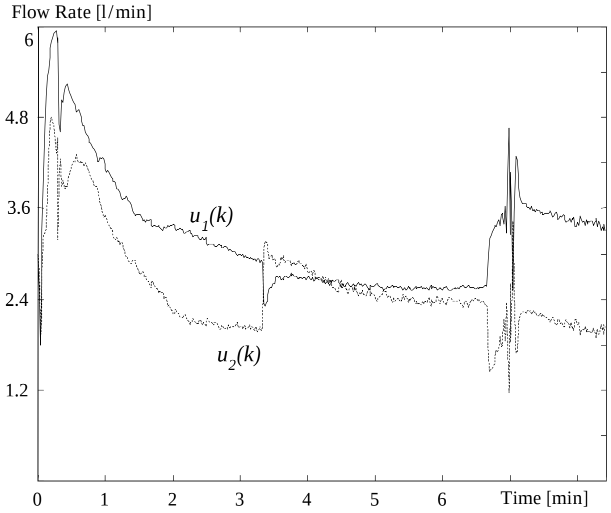

Figure 5: Control actions.

In figure 6 the on-line identification of the plant states (liquid levels) with the modified Elman neural model is illustrated. As can be seen the neural model performs considerably well in tracking the actual states.

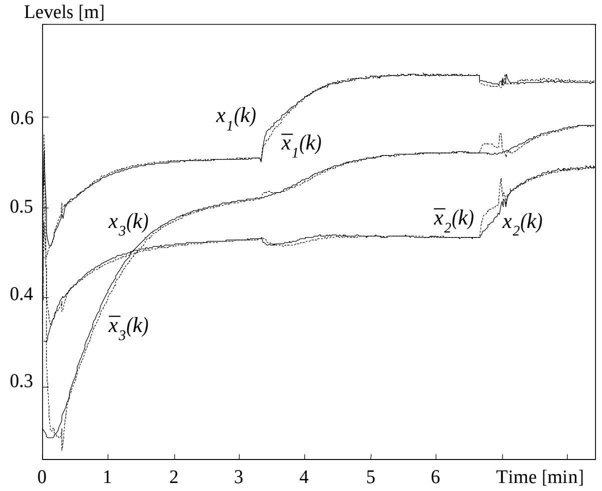

Figure 6: States and estimated states.

5 Conclusions

Hybrid control systems may contribute to the establishment of an unifying control theory merging traditional analytic-algebraic methods with artificial intelligent tools. This may allow to extended to general non-linear systems concepts and architectures of the vast linear system theory. In the present work a dynamic recurrent neural network is the process identifier allowing to use a conventional decoupling pole-placement controller. This approach was tested in a non-linear 2x2 real process and good results were obtained. Further work is however need in order to study the general properties of stability and robustness of such architecture.

Acknowledges

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

References

[1] Ahmed M., Tasadduq I., "Neural net controller for non-linear plants: design approach trough linearisation", IEE Proc. Control Theory Applications, 141, 5, 315-322, (1994).

[2] Amira, "Three Tank System DTS200: Laboratory Setup", Amira GmbH, (1996).

[3] Cao S., Rees N., Feng G., "Analysis and design for a class of complex control systems, Part I and II.", Automatica, 33, 6, 1017-1039, (1996).

[4] Chen C.-Li, Chang M.-Hui, "Optimal design of fuzzy sliding-mode control: a comparative study", Fuzzy Sets and Systems, 93, 37-48, (1998).

[5] Elman J., "Finding Structure in time", University of California, San Diego, (1988).

[6] Falb P., Wollovith W., "Decoupling in the Design and Synthesis of Multivariable Control Systems", IEEE Trans. Automatic Control, 12, 6, 651-659, (1967).

[7] Fuh C., Tung P., "Robust stability analysis of fuzzy control systems", Fuzzy Sets and Systems, (1997).

[8] Jagannathan S., Lewis F., "Identification of non-linear dynamical systems using multilayer neural networks", Automatica, 32, 12, 1707-1712, (1996).

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

[10] Lygeros J., "A formal approach to fuzzy modelling", IEEE Trans. on Fuzzy Systems, 5, 3, 317-324, (1997).

[11] Ma. X-J. , Sun Z., He Y., "Analysis and design of fuzzy controller and fuzzy observer", IEEE Trans. On Fuzzy Systems, 6, 1, 41-51, (1998).

[12] Narendra K., Parthasarathy K., "Gradient methods for the optimization of dynamical systems containing neural networks", IEEE Trans. on Neural Networks, 2, 2, 252-262, (1991).

[13] Pham D., Xing L., "Dynamic system identification using Elman and Jordan networks", Neural Networks for Chemical Engineers, A. Bulsari Editor, Chap. 23, 572-591, (1995).

[14] Shaw A., Doyle III F., "Multivariable non-linear control applications for a high purity distillation column using a recurrent dynamic neuron model", J. Process Control, 7, 4, 255-268, (1997).

[15] Rumelhart D., Hinton G., Williams R., "Learning internal representations by error propagation", Explorations in the microstructure of cognition, 1. Foundations, Cambridge, MIT Press, (1986).

[16] Sorensen O., "Non-linear pole placement control with a neural network", European Journal of Control, 2, 36-43, (1996).

[17] Suykens J., Vandewallw J., Moor B., "Artificial neural networks for modeling and control of non-linear systems", Kluwer Academic Press, (1996).

[18] Tanaka K., Ikeda T., Wang H., "Fuzzy regulators and fuzzy observers: relaxed stability conditions and LMI-Based Designs", IEEE Trans. On Fuzzy Systems, 6, 2, 250-265, (1998).

[19] Wang J., Wu G., "A Multilayer recurrent neural network for on-line synthesis of minimum-norm linear feedback control systems via pole assignment ", Automatica, 32, 3, 435-442, (1996).

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

[21] Williams R., Zipser D., "Gradient-based learning algorithms for recurrent networks and their computational complexity", Backpropagation: Theory, architectures and applications, Yves Chauvin and D. Rumelhart Editors, Chap.13, 433-486, (1995).