NONLINEAR SYSTEM IDENTIFICATION USING ADDITIVE DYNAMIC NEURAL NETWORKS.

R. GRIÑÓ*

Institut de Cibernètica (UPC-CSIC), Diagonal 647, 08028-Barcelona, Spain. E-mail: grino@ic.upc.es

Abstract. In this work additive dynamic neural models are used for the identification of nonlinear plants in on-line operation. In order to accomplish this task a gradient parameter adaptation method based in sensitivity analysis is formulated taking into account that the parameters of the model are arranged in matrix form. This methodology is applied to several nonlinear systems in simulation and with a real dataset to verify its performance.

Key Words. Neural nets; system identification; sensitivity analysis; matrix algebra; gradient methods

1. INTRODUCTION

An important effort in the study of nonlinear systems in control theory has been carried out in the last years. This interest stems from the need to give new solutions to some historic necessities of the Automatic Control (Antsaklis, 1990): to work with more and more complex systems, to satisfy stricter criterions of design and to fulfill the previous points with less and less a priori knowledge of the plant.

In view of this tendency, a great effort is being carried out, within the area of system identification, in the development of nonlinear models of real processes. Within this line is framed all the work in the development of nonlinear models (Haber and Unbehauen, 1990) , that in the last years has been stated explicitly in methodologies like, for example, the NARMAX models (Korenberg and Paarmann, 1991; Mathews, 1991).

Next to this line of work with classical characteristics, it is added in the few last years a set of works that apply the artificial neural networks to the tasks of identification and control of systems. These works are supported by two of the most important capacities of the neural nets: their learning capacity (Narendra and Parthasarathy, 1990; Narendra and Parthasarathy, 1991)(based on the optimization of an adequate cost function) and their good behavior in the approximation of nonlinear functions (Polycarpou and Ioannou, 1991; Girosi and Poggio, 1989).

At present, most of the works on system identification performed with neural nets are based on multilayer feedforward neural networks with backpropagation learning or more efficient variations of this algorithm. These methods have been applied to real processes and they have demonstrated an adequate behavior (Bhat *et al.*, 1990; Weerasooriya and El-Sharkawi, 1991). It is important to note that the great majority of works until today use discrete time models that capture the dynamics of the real process through the use of tapped delay lines in the model inputs and outputs.

Nevertheless, at this moment internal dynamics and continuous time models are being analyzed, since this is a way to achieve simpler structures that represent the real process in a suitable form (Sanner and Slotine, 1991*a*; Sanner and Slotine, 1991*b*).

2. ARCHITECTURE OF THE CONNECTIONIST MODELS

In this work, continuous time additive dynamics neural networks (Kosko, 1992) have been used as the models to adjust to the real processes. This approach has several advantages with regard to the discrete time tapped delay line based models (Wu *et al.*, 1992; Chen, 1990):

- the number of configuration parameters of the model is considerably less. It is only necessary to specify the dimension of the state space since the number of inputs and outputs is determined by their counterparts in the real process.
- the models obtained with this approach are in state space form and work in continuous time, which is very interesting in order to apply

the differential geometric theory for nonlinear control (Isidori, 1989).

The architecture of the connexionist models is described by the structure of their basic elements and how they are interconnected.

2.1. *Basic Elements*

The basic processing element of a neural model is the node, also called neuron by analogy with the biological neurons. In general, the basic model of a node (see Fig. 1) is composed of a weighted adder, a linear dynamic SISO system and a nonlinear static function. In this work, these elements have the following form:

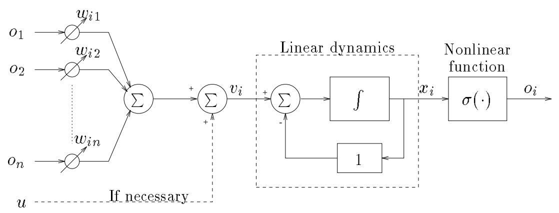

Fig. 1. Basic node in the additive neural model.

- The weighted adder is described by the equation:

$$v_i = \sum_{j=1}^{n} w_{ij} o_j + \delta_{ik} u; \quad i = 1 \ldots n \tag{1}$$

  where the weighted sum v_i is a linear combination of the outputs of the nodes o_i of the net and the external input u, if it is applicable because only one node of the network receives it. If we take all the equations (1) and arrange them in matrix form we obtain

$$\mathbf{v} = \mathbf{W}\mathbf{o} + \mathbf{b}u \tag{2}$$

  where $\mathbf{W} = [w_{ij}] \in M_n(\mathrm{I\!R})$ is the weights matrix of the network and, $\mathbf{b} = \left[0, \ldots, \overset{k)}{1}, \ldots, 0\right]^T \in \mathrm{I\!R}^n$ is the fixed parameter vector that applies the input to the model.
- The linear dynamical system has the weighted sum v_i as input and x_i as output. In the present work, we take a first order linear system with a fixed time constant ($\tau_i = 1$) for all the nodes in the network. That is to say, all the nodes in the net have the following differential equation in their structure:

$$\dot{x}_i + x_i = v_i \tag{3}$$

- The nonlinear static activation function selected in this work is the hyperbolic tangent and it maps the state of a node into its ouput ($\sigma : \mathrm{I\!R} \to (-1, 1), \sigma \in C^\infty$).

2.2. *Additive Models*

The composition of basic elements explained above constitutes a node of the connexionist additive model and the arrangement in matrix form of all of the nodes, together with a linear output equation with constant parameters has the following structure (see Fig. 2):

$$\dot{\mathbf{x}} = -\mathbf{x} + \mathbf{W}\mathbf{g}(\mathbf{x}) + \mathbf{b}u \tag{4}$$

$$\hat{y} = \mathbf{c}\mathbf{x} \tag{5}$$

where $\mathbf{x} \in \mathrm{I\!R}^n$ is the state vector, $u \in \mathrm{I\!R}$ the input to the model and the plant, $\hat{y} \in \mathrm{I\!R}$ the output of the model, $\mathbf{W} \in M_n(\mathrm{I\!R})$ the weight (parameter) matrix, $\mathbf{c} = [1, 1, \ldots, 1] \in \mathrm{I\!R}^{1\times n}$ the output vector, $\mathbf{b} \in \mathrm{I\!R}^n$ the input vector with the elements cited in the previous section and

$$\mathbf{g}(\mathbf{x}) = [\tanh(x_1), \ldots, \tanh(x_n)]^T \tag{6}$$

a C^∞ nonlinear vector field from $\mathrm{I\!R}^n$ to (-1,1).

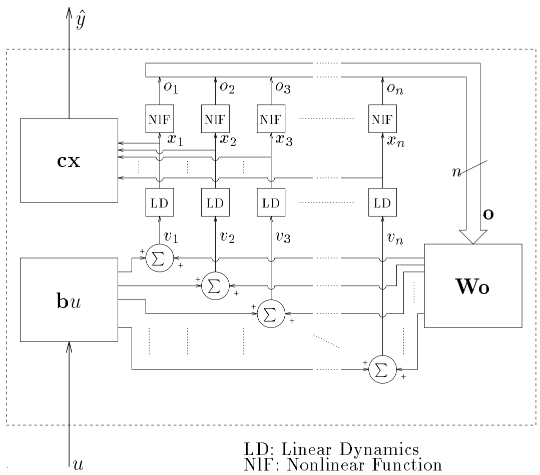

Fig. 2. Additive neural network model.

3. PARAMETER ADAPTATION IN THE CONNECTIONIST MODELS

The basic idea of the identification process is shown in Fig. 3. The connexionist model is arranged in a parallel manner with the real plant, and the excitation signal is the same for both. Clearly, the objective is to have the same output signals from the plant y and the model $\hat{y}$ for each time t. However, in general it will not be possible in a strict sense because we do not have a knowledge of the structure of the real plant and we can only rely on the possibility that our model may be able to identify the class of systems to which our plant belongs.

Because of the special capabilities of the neural networks in the approximation tasks (Funahashi and Nakamura, 1993) the problem can be avoided

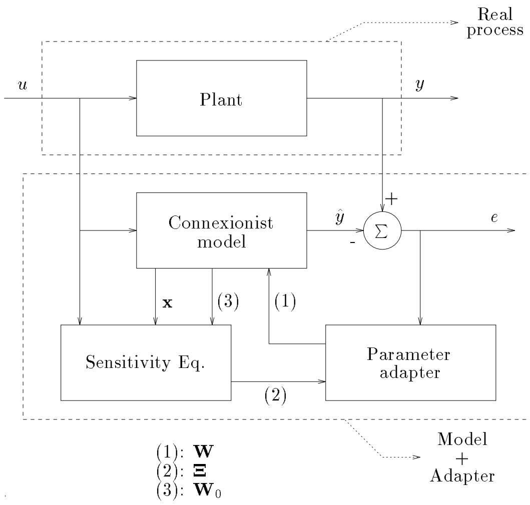

Fig. 3. Identification structure.

assuming that the proposed models are able to approximate the plant output except for a residual error due to the structural modelling mismatch between the plant and the model.

3.1. *Cost Function and Adaptation Law*

The identification error e can be defined as

$$e = y - \hat{y} \tag{7}$$

where y is the output of the real process and $\hat{y}$ the output of the connectionist model. This error is a function of time and the parameters of the model

$$e = e(t, \mathbf{W}) \tag{8}$$

and it takes zero value when the model fits perfectly the behaviour of the system to be identified. Accordingly, the problem is to find a set of parameters in a on-line manner in order to attain an identification error going to zero or to a small residual value due to missmodelling and noise in the process.

Then, it is necessary to define a cost functional over this error to perform the minimization process. Perhaps the simplest functional is

$$V = \frac{1}{2} e^2 \tag{9}$$

that attains its minimum when the error is equal to zero and is strictly convex with respect to the error. To minimize the cost functional V on-line the following gradient based update rule can be selected

$$\dot{\mathbf{W}} = -\epsilon \frac{\partial V}{\partial \mathbf{W}} \tag{10}$$

where the parameter ϵ is a small positive constant as it is explained in the next section.

The term $\frac{\partial V}{\partial \mathbf{W}}$ can be developed using the chain rule of matrix calculus and Kronecker products [1](Vetter, 1970; Brewer, 1978) obtaining

$$\frac{\partial V\left(e\left(\mathbf{W}\right)\right)}{\partial \mathbf{W}} = \left(\mathbf{I}_n \otimes \frac{\partial V}{\partial e}\right) \frac{\partial e}{\partial \mathbf{W}} \tag{11}$$

where the rightmost member is, by the chain rule,

$$\frac{\partial e\left(\hat{y}\left(\mathbf{W}\right)\right)}{\partial \mathbf{W}} = \left(\mathbf{I}_n \otimes \frac{\partial e}{\partial \hat{y}}\right) \frac{\partial \hat{y}}{\partial \mathbf{W}} \tag{12}$$

Substituting the equation (12) in (11) results

$$\begin{aligned}
\frac{\partial V}{\partial \mathbf{W}} &= \left(\mathbf{I}_n \otimes e\right)\left(\mathbf{I}_n \otimes (-\mathbf{I}_1)\right) \frac{\partial \hat{y}}{\partial \mathbf{W}} \\
&= -\left[\left(\mathbf{I}_n \mathbf{I}_n\right) \otimes \left(e \mathbf{I}_1\right)\right] \frac{\partial \hat{y}}{\partial \mathbf{W}} \\
&= -\left(\mathbf{I}_n \otimes e\right) \frac{\partial \hat{y}}{\partial \mathbf{W}} \qquad (13)
\end{aligned}$$

at which the update equation of the weight matrix $\mathbf{W}$ is

$$\dot{\mathbf{W}} = \epsilon \left(\mathbf{I}_n \otimes e\right) \frac{\partial \hat{y}}{\partial \mathbf{W}}\ , \quad \mathbf{W}(0) = \mathbf{W}^0 \tag{14}$$

where $\mathbf{W}^0$ is the arbitrary starting value of the weight matrix.

In the last equation appears the term $\frac{\partial \hat{y}}{\partial \mathbf{W}}$ that is known as the sensitivity matrix of the output trajectories versus the parameters of the model, in this case arranged in matrix form.

3.2. *Sensitivity Equations of the Model Trajectories*

In this section the sensitivity equations (Frank, 1978) of the connexionist model that are necessary to the equation (14) are developed. Rewriting the equations (4) and (5) as

$$\begin{aligned}
\dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}, u, \mathbf{W}) \\
&= -\mathbf{x} + \mathbf{W}\mathbf{g}(\mathbf{x}) + \mathbf{b}u \qquad (15) \\
\hat{y} &= \mathbf{c}\mathbf{x} \qquad (16)
\end{aligned}$$

in order to obtain more concise developments, and applying the partial derivative with respect to the weight matrix to both sides of the equations it results for the state equation (15),

$$\frac{\partial \dot{\mathbf{x}}}{\partial \mathbf{W}} = \frac{\partial \mathbf{f}}{\partial \mathbf{W}} + \left(\mathbf{I}_n \otimes \frac{\partial \mathbf{f}}{\partial \mathbf{x}^T}\right)\left(\frac{\partial \mathbf{x}}{\partial \mathbf{W}} \otimes \mathbf{I}_1\right) \tag{17}$$

and for the output equation (16),

$$\frac{\partial \hat{y}}{\partial \mathbf{W}} = \left(\mathbf{I}_n \otimes \mathbf{c}\right) \frac{\partial \mathbf{x}}{\partial \mathbf{W}} \tag{18}$$

---

[1] Also called tensor products.

Now, let $\mathbf{\Lambda} = \frac{\partial \mathbf{x}}{\partial \mathbf{W}} \in M_{n^2,n}(\mathrm{I\!R})$ and $\mathbf{\Xi} = \frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{W}} \in M_{n,n}(\mathrm{I\!R})$ the state trajectory sensitivity matrix and the output trajectory sensitivity matrix of the model, respectively. If the parameters of the model were constant, or quasi constant, the derivative operators in the equation (17) (derivative w.r.t. the weight matrix and w.r.t. time) could be swapped. It is necessary to point out that if the parameters of the model were constant the sensitivity equations give the true gradient of the output trajectories with respect to the parameters. Obviously, in the development above the parameters are not constant because they are updated with equation (9), but if in this equation the constant ϵ were sufficiently small, it could be assumed that the parameters are constant, swapping the operators and have a pseudo gradient with a low error. Performing these operations in the equation (17) gives

$$\dot{\mathbf{\Lambda}} = \left(\mathbf{I}_n \otimes \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}^T} \right|_{\mathbf{W}_0} \right) \mathbf{\Lambda} + \left. \frac{\partial \mathbf{f}}{\partial \mathbf{W}} \right|_{\mathbf{W}_0} \tag{19}$$

$$\mathbf{\Xi} = \left(\mathbf{I}_n \otimes \mathbf{c} \right) \mathbf{\Lambda} \tag{20}$$

Accordingly, the equations involved in the identification process are

$$\dot{\mathbf{x}} = -\mathbf{x} + \mathbf{W}\mathbf{g}(\mathbf{x}) + \mathbf{b}u \tag{21}$$

$$\hat{y} = \mathbf{c}\mathbf{x} \tag{22}$$

$$e = y - \hat{y} \tag{23}$$

$$\dot{\mathbf{\Lambda}} = \left(\mathbf{I}_n \otimes \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}^T} \right|_{\mathbf{W}_0} \right) \mathbf{\Lambda} + \left. \frac{\partial \mathbf{f}}{\partial \mathbf{W}} \right|_{\mathbf{W}_0} \tag{24}$$

$$\mathbf{\Xi} = \left(\mathbf{I}_n \otimes \mathbf{c} \right) \mathbf{\Lambda} \tag{25}$$

$$\dot{\mathbf{W}} = \epsilon \left(\mathbf{I}_n \otimes e \right) \mathbf{\Xi} \tag{26}$$

where (21) and (22) are the connectionist model, (23) is the identification error equation, (24) and (25) are the sensitivity equations of the model and (26) is the on-line parameter update equation. It is important to note that the initial conditions for the equation (24) are zero.

Clearly, all the equations involved in this structure go to positive time which enables the on-line operation of the whole set. But a serious shortcoming of the proposed approach is the high complexity in the number of operations. This problem has been partially avoided through the simplification of the matrix operations involved in the equations using a symbolic manipulator in order to get a sequential code with only the nonzero operations.

4. SIMULATION RESULTS

The models and the associated equations developed in the previous sections have been implemented together with the equations of the real processes in the continuous time simulation language ACSL. The plants used in this work are two kinds of second order nonlinear system and a real data set of the electrical power desired and generated by a real hydroelectric power plant.

4.1. *Nonlinear Second Order System*

The two systems used can be seen in Fig. 4. This kind of systems is very usual in industrial processes, specially in motion control systems. In all the experiments a maximun length binary sequence has been used as input to the plant and the model. This kind of input has been chosen because its industrial use is more extended than the pure white noise. Also, its necessary to point out that has not been carried out any tuning of the initial conditions of the models, and it is envisaged that this fact could improve the adjustment of the models.

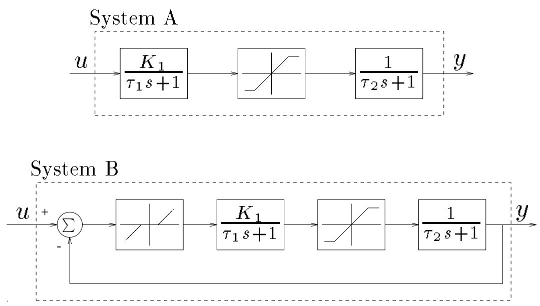

Fig. 4. Second order nonlinear systems used in the identifications.

System A. This system is composed of two cascaded first order systems with a saturation between them. In Fig. 5 can be seen the identification error and the real and model output for an experiment with a five nodes connectionist model and $\epsilon = 0.1$. At time $t = 100$ seconds the adaptation mechanism is stopped and the model performs its prediction task quite well.

The five nodes model is the minimum size model that acquires the dynamical behaviour of the real process, and this fact is recognized because the weights remain in quasi constant values some time after the performance of the model is good. That is to say, the model and not the adapter is the dynamical subsystem that learns the process behaviour.

System B. This system is like the System A but with a dead zone before the first dynamic block and works in closed loop. In Fig. 6 and 7 can be seen the temporal and the frequencial responses, respectively, of the real process and a model with five nodes and $\epsilon = 0.5$. It is specially important to note the good adjust of the gain and phase at

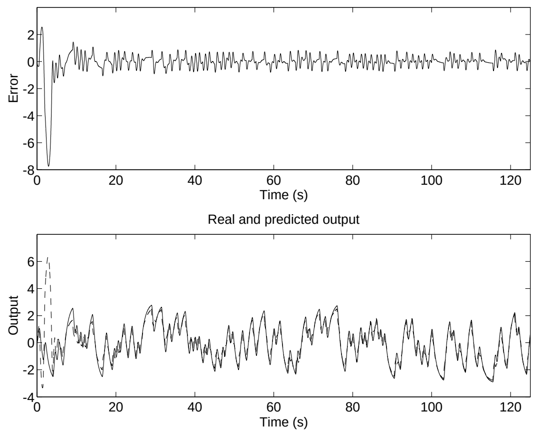

Fig. 5. System A: $\tau_1 = 0.2, K_1 = 2, \tau_2 = 0.2$, saturation limits = 3,-3.

low and medium frequency.

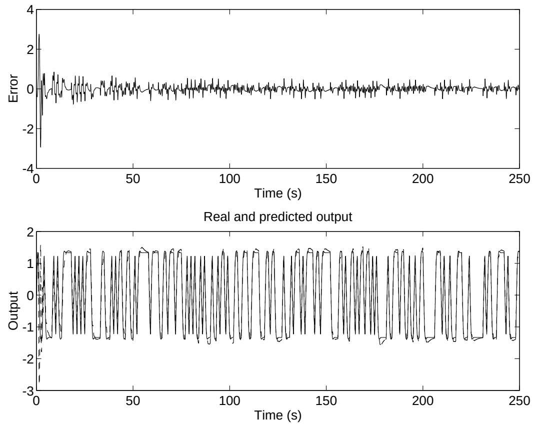

Fig. 6. System B: same values as system A and a dead zone bounds of +1,-1.

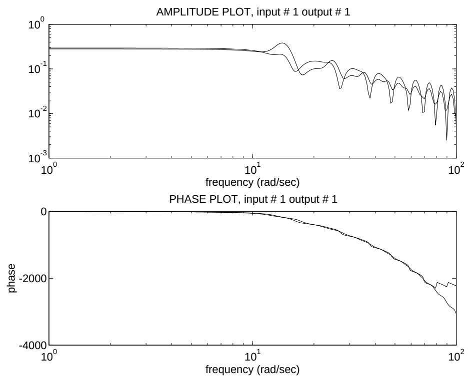

Fig. 7. System B: frequency response.

Among the experiments performed in this work, a lack of excitation in the input to the system has been tested and the results show a high degree of robustness since no drift is observed in the model parameters.

4.2. *Hydroelectric Power Plant*

In this case, the model works with a real data set obtained from an hydroelectric power plant of the utilities company ENHER. The dataset is composed of two time series corresponding to the desired electrical power of the plant and the real electrical power obtained. The sampling period of the dataset is 3 seconds.

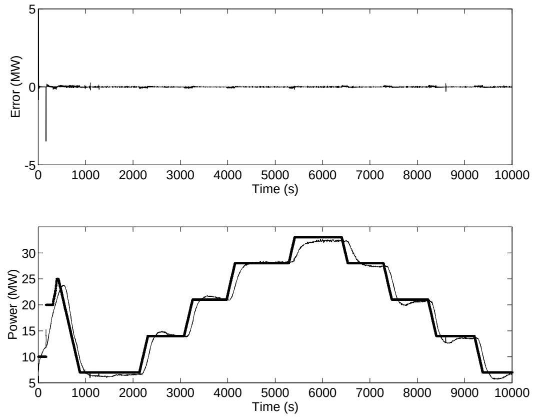

Fig. 8. Identification error and reference, output and model output power.

As it can be seen in Fig. 8 the real output power is indistinguishable from the model output power, except in the first moments of the on-line adjust.

5. CONCLUSION

In this work additive dynamic neural networks have been applied to nonlinear system identification tasks in on-line operation. These models are posed in state space form with the advantages that this kind of representation suppose for their later use in controller design. To accomplish the on-line requirement, it has been necessary to formulate a gradient update rule based on the sensitivity analysis of the model trajectories using matrix calculus to compact the development. Also, it is important to point out that the application of this methodology to several representative cases of nonlinear systems has given good results.

As future work, it is envisaged to extend the kind of parameters in the neural models using adaptive time constants for each node of the net and to work on the theoretical aspects of the stability and convergence of parameters of the set composed by the real process, the model and the parameter adapter.

6. ACKNOWLEDGEMENTS

The author wishes to thank the support and the valuable advices of Dr. Carme Torras and Dr.

Gabriela Cembrano in the realization of the current work. This work has been partially supported by the Comisión Interministerial de Ciencia y Tecnologia (CICYT) under project TIC91-0423, and by a grant from the *Asociación / Colegio de Ingenieros Industriales de Cataluña.*

7. REFERENCES

Antsaklis, P.J. (1990). Neural networks in control systems. *IEEE Control Syst. Mag.* (2), 3–5.

Bhat, N.V., P.A. Minderman, T.J. McAvoy and N.S. Wang (1990). Modeling chemical process systems via neural computation. *IEEE Control Syst. Mag.* (2), 24–29.

Brewer, J.W. (1978). Kronecker products and matrix calculus in system theory. *IEEE Trans. Circuits and Syst.* **25**(9), 772–781.

Chen, F. (1990). Back-propagation neural networks for nonlinear self-tuning adaptive control. *IEEE Control Syst. Mag.* (2), 44–48.

Frank, P.M. (1978). *Introduction to System Sensitivity Theory.* Academic Press.

Funahashi, K. and Y. Nakamura (1993). Approximation of dynamical systems by continuous time recurrent neural networks. *Neural Networks* **6**, 801–806.

Girosi, F. and T. Poggio (1989). Representation properties of networks: Kolmogorov's theorem is irrelevant. *Neural Comp.* **1**, 465–469.

Haber, R. and H. Unbehauen (1990). Structure identification of nonlinear dynamic systems – a survey on input/output approaches. *Automatica* **26**(4), 651–677.

Isidori, A. (1989). *Nonlinear Control Systems.* Springer-Verlag.

Korenberg, M.J. and L.D. Paarmann (1991). Orthogonal approaches to time-series analysis and system identification. *IEEE Signal Proc. Mag.* (7), 29–43.

Kosko, B. (1992). *Neural Networks and Fuzzy Systems.* Prentice Hall Ed.

Mathews, V.J. (1991). Adaptive polynomial filters. *IEEE Signal Proc. Mag.* (7), 10–26.

Narendra, K.S. and K. Parthasarathy (1990). Identification and control of dynamical systems using neural networks. *IEEE Trans. Neural Net.* **1**(1), 4–26.

Narendra, K.S. and K. Parthasarathy (1991). Gradient methods for the optimization of dynamical systems containing neural networks. *IEEE Trans. Neural Net.* **2**(2), 252–262.

Polycarpou, M.M. and P.A. Ioannou (1991). Identificaction and control of nonlinear systems using neural network models: Design and stability analysis. Tech. Rep. 91-09-01. Univ. of Southern California.

Sanner, R.M. and J-J.E. Slotine (1991*a*). Direct adaptive control using gaussian networks. Tech. Rep. SL-910303. Nonlinear Systems Lab., MIT.

Sanner, R.M. and J-J.E. Slotine (1991*b*). Gaussian networks for direct adaptive control. Tech. Rep. SL-910503. Nonlinear Systems Lab., MIT.

Vetter, W.J. (1970). Derivative operations on matrices. *IEEE Trans. Autom. Control* **15**(4), 241–244.

Weerasooriya, S. and M.A. El-Sharkawi (1991). Identification and control of a dc motor using back-propagation neural networks. *IEEE Trans. Energy Con.* **6**(4), 663–669.

Wu, Q.H., B.W. Hogg and G.W. Irwin (1992). A neural network regulator for turbogenerators. *IEEE Trans. Neural Net.* **3**(1), 95–100.