A Neural Network with Minimal Structure for Maglev System Modeling and Control

Mostafa Lairi, Gérard Bloch

▶ **To cite this version:**

Mostafa Lairi, Gérard Bloch. A Neural Network with Minimal Structure for Maglev System Modeling and Control. IEEE International Symposium on Intelligent Control / Intelligent Systems & Semiotics, ISIC/ISAS'99, Sep 1999, Cambridge, MA, United States. pp.40-45, 10.1109/ISIC.1999.796627 . hal-00405124

HAL Id: hal-00405124
https://hal.science/hal-00405124

Submitted on 18 Jul 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Proceedings of the 1999 IEEE
International Symposium on Intelligent Control/Intelligent Systems and Semiotics
Cambridge, MA • September 15-17, 1999

A neural network with minimal structure for MagLev system modeling and control

Mostafa Lairi and Gérard Bloch
Centre de Recherche en Automatique de Nancy (CRAN-CNRS ESA 7039)
ESSTIN, Rue Jean Lamour, 54500 Vandoeuvre, France
E-mail: bloch@cran.esstin.u-nancy.fr

Abstract

The paper is concerned with the determination of a minimal structure of a one hidden layer perceptron for system identification and control. Structural identification is a key issue in neural modeling. Decreasing the size of a neural network is a way to avoid overfitting and bad generalization and leads moreover to simpler models which are required for real time applications, particularly in control.

A learning algorithm and a pruning method both based on a criterion robust to outliers are presented. Their performances are illustrated on a real example, the inverse model identification of a MagLev system, which is nonlinear, dynamical and fast. This inverse model is used in a feedforward neural control scheme. Very satisfactory approximation performances are obtained for a network with very few parameters.

Keywords

one hidden layer perceptron, structural identification, pruning, robust criterion, magnetic levitation.

1. Introduction

Artificial neural networks have been the focus of a great deal of attention during the last decade, due to their capabilities to solve nonlinear problems by learning. Backpropagation or derived algorithms have been particularly applied for process modeling and identification [5, 16] and for control [1, 9, 14].

In order to diminish the influence of the outliers (i.e. to provide a more reliable and robust estimate of the unknown parameter vector), a M-estimator can be used. This approach, stemming from robust statistics, has been introduced for neural networks learning by several authors [2, 3, 6, 7, 13, 17]. It must be noted that neural networks are not at all intrinsically insensitive to outliers and ignoring the effect of outliers, particularly for industrial data which are frequently corrupted by spiky noise, can lead to biased estimators but also to overparametrized structures and poor generalization as a consequence of overfitting.

Even if in some cases perceptrons with several hidden layers carry out more precise mappings, most of the time perceptrons with one hidden layer and linear activation at the output are sufficient and the universal approximation ability of such a structure has been proved. The search for the appropriate structure of a perceptron for modeling a particular system is one of the key issues in neural modeling. Apart from the fact that decreasing the size of the network leads to simpler models which are required for real time control with short sampling period, constraining the model to be "simple" in some sense is a way to avoid overfitting which often causes bad generalization [10]. Several approaches can be used to constrain the number of the network parameters or the parameters themselves, including the use of regularized learning criteria (e.g. "weight decay") or the pruning of useless parameters after learning. This paper focuses on pruning methods, particularly the Optimal Brain Surgeon (OBS) method [8], insofar as, in practice, tuning the balance parameter for regularized criteria is not an easy task, which is most of the time done by hand.

In the second part, the form of a one hidden layer perceptron with linear activation function at the output and the general Levenberg-Marquardt estimation algorithm are briefly recalled. In the third part, a robust weighted learning criterion is described while the fourth part recalls the principle of the OBS procedure and the weak modifications necessary to incorporate the robust criterion for weight elimination.

The last part illustrates the proposed robust algorithms for parameter estimation and pruning on a real magnetic levitation system (MagLev), which is nonlinear and unstable. The goal is to build a neural inverse model of the MagLev system to be included in a feedforward control strategy [11, 12]. Results obtained with standard Levenberg-Marquardt parameter algorithm followed by OBS pruning and outlier-robust learning and pruning are compared, showing that a network with very few parameters and very satisfactory approximation performances can be obtained with the second approach.

0-7803-5665-9/99/$10.00 © 1999 IEEE

Such a network can be then easily incorporated in a real time application, with short sampling time.

2. Neural model and general learning rule

The one hidden layer perceptron with linear activation function at the output is considered here. Its form is given, for single output, by:

$$\hat{y} = \sum_{i=1}^{n_1} w_i^2 g\left(\sum_{h=1}^{n_0} w_{ih}^1 x_h^0 + b_i^1\right) + b^2 \qquad (1)$$

where x_h^0, $h = 1,\cdots,n_0$, are the inputs of the network, w_{ih}^1 and b_i^1, $i = 1,\cdots,n_1$, $h = 1,\cdots,n_0$, are the weights and biases of the hidden layer, the activation function g is the hyperbolic tangent, w_i^2, $i = 1,\cdots,n_1$, and b^2 are the weights and bias of the output neuron. All the weights and biases of the network can be grouped in the parameter vector θ, and the inputs x_h^0 in the regression vector $\varphi(x) = [x_1^0 \cdots x_h^0 \cdots x_{n_0}^0]$. So for the case k, the output predicted by the network can be written: $\hat{y}(k,\theta) = NN(\varphi(x(k)),\theta)$.

To estimate from data the parameter vector θ, the prediction error:

$$\varepsilon(k,\theta) = y(k) - \hat{y}(k,\theta) \qquad (2)$$

with y(k) the desired output, is formed and incorporated in a criterion to be minimized. A general form for this criterion, which leads to a M-estimator, is given, for n samples, by:

$$V(\theta) = \frac{1}{n}\sum_{k=1}^{n} L(\varepsilon(k,\theta)) \qquad (3)$$

where L(.) is a scalar case cost function.

The minimization of the criterion (3) can therefore be carried out using the Gauss-Newton algorithm:

$$\hat{\theta}^{i+1} = \hat{\theta}^i - (H(\hat{\theta}^i))^{-1} V'(\hat{\theta}^i) \qquad (4)$$

In (4), the gradient $V'(\theta)$ of the criterion (3) with respect to θ is given by:

$$V'(\theta) = -\frac{1}{n}\sum_{k=1}^{n} \psi(k,\theta) L'(\varepsilon(k,\theta)) \qquad (5)$$

where $\psi(k,\theta)$ is the gradient of $\hat{y}(k,\theta)$ with respect to θ and $L'(\varepsilon(k,\theta))$, the score or influence function, is the first derivative of $L(\varepsilon(k,\theta))$ with respect to ε. The second derivative $H(\theta)$ of the criterion (3) with respect to θ, known as the Hessian matrix, is obtained by differentiating (5) and can be approximated with the Levenberg-Marquardt update rule by:

$$H(\theta) = \frac{1}{n}\sum_{k=1}^{n} \psi(k,\theta) L''(\varepsilon(k,\theta)) \psi^T(k,\theta) + \beta I \qquad (6)$$

where $L''(\varepsilon(k,\theta))$ is the second derivative of $L(\varepsilon(k,\theta))$ with respect to ε, I is the identity matrix and β a small non negative scalar, adjusted during learning.

The criterion most frequently used for parameter estimation is the classical ordinary least-squares criterion (L2 norm), with as case cost function a quadratic one:

$$L(\varepsilon(k,\theta)) = \frac{1}{2}\varepsilon^2(k,\theta) \qquad (7)$$

whose derivatives are simply:

$$L'(\varepsilon(k,\theta) = \varepsilon(k,\theta) \quad , \quad L''(\varepsilon(k,\theta) = 1 \qquad (8)$$

Such a criterion receives the largest contributions from the points which have the largest errors and the solution can be dominated by a very small number of points which can be gross errors or outliers.

3. An outlier-robust learning rule

The batch learning method presented here is detailed in [4, 17]. This Iteratively Reweighted Least Squares (IRLS) method starts, following Huber, from a distribution of the noise e contaminated by outliers expressed by a mixture of two probability density functions. The first one corresponds to the basic distribution of a measurement noise, for example Gaussian, with variance σ_1^2 and the second one, corresponding to outliers, is arbitrary symmetric long-tailed, for example also Gaussian, but with variance σ_2^2 such as $\sigma_1^2 << \sigma_2^2$:

$$e(k) \sim (1-\mu)N(0,\sigma_1^2) + \mu N(0,\sigma_2^2)$$

where μ is the probability of occurring a large error. In practice, neither the probability μ nor the two variances σ_1^2 and σ_2^2 are known and the preceding model is replaced by:

$$\varepsilon(k) \sim (1-\delta(k))N(0,\sigma_1^2) + \delta(k)N(0,\sigma_2^2) \qquad (9)$$

where $\varepsilon(k)$ is the prediction error, given by (2), $\delta(k) = 0$ for $|\varepsilon(k)| \le M$ and $\delta(k) = 1$ for $|\varepsilon(k)| > M$, and M is a bound which can be taken as $3\sigma_1$.

The unknown variances σ_1^2 and σ_2^2 are estimated as follows. At each iteration i of the algorithm (4), the prediction error sequence ε is calculated by (2), and the variances recursively estimated by:

for $|\varepsilon(k)| \le 3\sigma_1(k-1)$

$$\sigma_1^2(k) = \sigma_1^2(k-1) + \frac{1}{k-\tau(k)}(\varepsilon^2(k) - \sigma_1^2(k-1)) \quad (10a)$$

otherwise

$$\sigma_1^2(k) = \sigma_1^2(k-1)$$

and

for $|\varepsilon(k)| > 3\sigma_1(k-1)$

$$\sigma_2^2(k) = \sigma_2^2(k-1) + \frac{1}{\tau(k)}(\varepsilon^2(k) - \sigma_2^2(k-1)) \quad (10b)$$

otherwise

$$\sigma_2^2(k) = \sigma_2^2(k-1)$$

with $\tau(0) = 0$ at each iteration, and $\tau(k+1) = \tau(k) + 1$ whenever $|\varepsilon(k)| > 3\sigma_1(k-1)$. $\sigma_2^2(0)$ can be chosen equal to $\sigma_1^2(0)$ and $\sigma_1^2(0)$ equal to the classically calculated variance of the prediction errors at the first iteration. Note that $\tau(n)$ is the estimate of the number of outliers.

The variance $\sigma_n^2(k)$ of $\varepsilon(k)$ is finally given by:

$$\sigma_n^2(k) = (1-\delta(k))\sigma_1^2(n) + \delta(k)\sigma_2^2(n) \quad (11)$$

leading to the weighted robust norm:

$$L(\varepsilon(k)) = \frac{1}{2}\frac{\varepsilon^2(k)}{\sigma_n^2(k)} \quad (12)$$

Algorithm (4) can be then employed with (12) as case cost function in criterion (3), with its first derivative:

$$L'(\varepsilon(k)) = \frac{\varepsilon(k)}{\sigma_n^2(k)} \quad (13)$$

in the criterion gradient (5) and with its second derivative:

$$L''(\varepsilon(k)) = \frac{1}{\sigma_n^2(k)} \quad (14)$$

in the approximate Hessian (6).

4. Robust Pruning

After estimation of the parameter vector, it can be useful to determine the minimal structure of the network. Removing the useless parameters leads obviously to a simpler model, but moreover diminishes the overfitting of the model to data, i.e. the learning of the noise and of the unknown underlying model of the system at the same time, and can thus improve the generalization abilities of the model. The pruning algorithm which is probably the most classical is the Optimal Brain Surgeon, proposed by Hassibi and Stork [8]. This algorithm minimizes the sensitivity of the error criterion subject to the constraint of nullity of a weight, which expresses the deletion of this weight. The sensitivity $\delta V(\theta)$ of the criterion $V(\theta)$ is approximated by a Taylor expansion around θ to order two:

$$\delta V(\theta) \approx \delta\theta^T V'(\theta) + \frac{1}{2}\delta\theta^T H \delta\theta \quad (15)$$

The gradient $V'(\theta)$ being zero after convergence, the first term in (15) vanishes, leading to:

$$\delta V(\theta) = \frac{1}{2}\delta\theta^T H \delta\theta \quad (16)$$

which involves only the Hessian H. Noting e_q a canonical vector selecting the q^{th} component of θ ($e_q^T = [0 \cdots 0\ 1\ 0 \cdots 0]$), the deletion of the weight θ_q (i.e. $e_q^T(\delta\theta + \theta) = 0$) must lead to a minimal increase of the criterion. The following Lagrangian can thus be written:

$$\mathcal{L}(\delta\theta) = \frac{1}{2}\delta\theta^T H \delta\theta + \lambda(e_q^T(\delta\theta + \theta)) \quad (17)$$

and minimized, leading to:

$$\delta\theta = -\frac{\theta_q}{H_{qq}^{-1}} H^{-1} e_q \quad (18)$$

where H_{qq}^{-1} is the q^{th} diagonal term of H^{-1}. The weight to be deleted is the one which minimizes (15). Equation (18) allows to force the q^{th} weight to zero and to update the remaining weights, without retraining. It can be nevertheless useful to retrain the network after each pruning of a weight, in order to compensate the approximation introduced by the Taylor expansion (15).

If the true quadratic criterion is used, the corresponding approximate Hessian is given by:

$$H(\theta) = \frac{1}{n}\sum_{k=1}^{n} \psi(k,\theta)\psi^T(k,\theta) \quad (19)$$

For the robust weighted criterion, the Hessian becomes:

$$H(\theta) = \frac{1}{n}\sum_{k=1}^{n} \psi(k,\theta)\frac{1}{\sigma_n^2(k)}\psi^T(k,\theta) \quad (20)$$

where $\sigma_n^2(k)$ is given by (11).

As presented in [18] on simulation examples, the use of weighted robust criterion for initial learning as well as for pruning greatly improves the estimation of the parameters, particularly the associated generalization capabilities, and the model structure selection.

5. Magnetic levitation example

Outlier robust learning and structure determination presented above are applied to model the inverse behavior of a magnetic levitation system (MagLev). Precise description of the MagLev system which allows to balance a metallic ball without any support, by the use of an electromagnet, and to steer the object tracking a desired vertical trajectory, can be found in [11, 12]. The system input is the voltage U applied to the coil and the output is the voltage V_x corresponding to the position X of the ball (cf. figure 1).

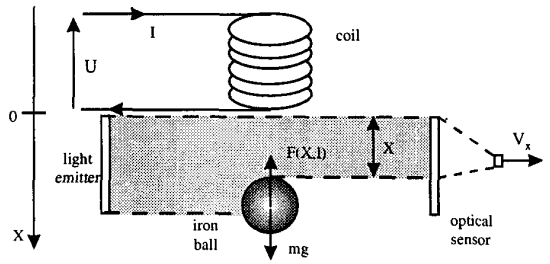

Figure 1: Principle diagram of Magnetic Levitation

A neural inverse model of the system is used with a PID in a feedforward control strategy (cf. figure 2).

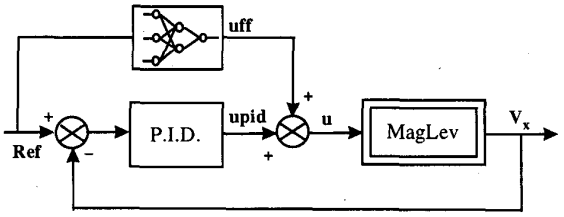

Figure 2: Principle of neural feedforward control

Based on direct linear model of the system, the expression of the output V_x at time k+1 can be written:

$$V_x(k+1) = f(V_x(k), V_x(k-1), U(k), U(k-1)) \qquad (21)$$

where f is a non linear function modeling the system behavior. The inverse model can be thus expressed as:

$$U(k) = f^{-1}(V_x(k+1), V_x(k), V_x(k-1), U(k-1)) \qquad (22)$$

So, taking $\phi(k) = [V_x(k+1)\ V_x(k)\ V_x(k-1)\ U(k-1)]$ as regression vector, the neural network estimates the inverse model: $\hat{U}(k) = NN(\phi(k), \theta)$.

To collect data for estimation and validation of the neural model, experiment is carried out in closed loop, the system being unstable in open loop. A white noise is added to the control signal U, given by the PID controller, to enrich the excitation signal, as presented in figure 3. In this figure, "ML system" is the block which represents the real system (real time interfaced).

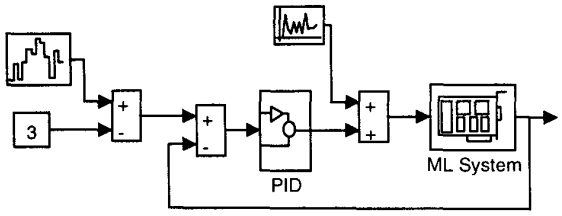

Figure 3: Data collection in closed loop

The reference voltage is chosen with two goals: to control the system around the operating point (-3 volts) and to explore an operating range as large as possible. From 1500 couples ($V_X(k)$, U(k)), presented figure 4, network parameters estimation is achieved by two ways.

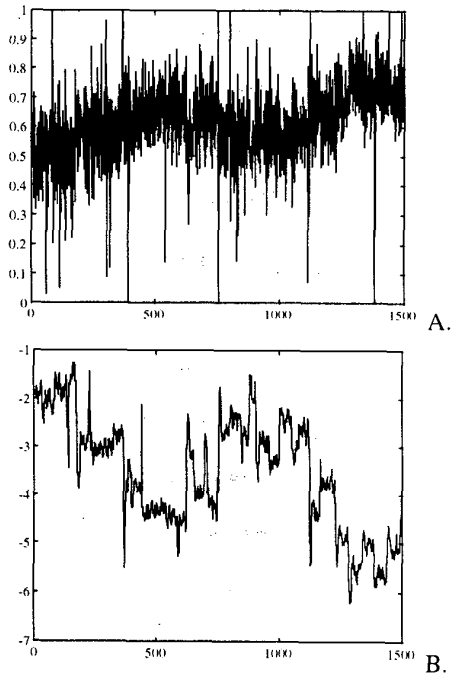

Figure 4: Learning set with 1500 data
(A) U in volts (B) V_X in volts

The first one is the standard Levenberg-Marquardt (LM) algorithm for initial learning and OBS algorithm for pruning as implemented in [15]. The initial fully connected model comprises 10 hidden hyperbolic tangent units, i.e. 61 parameters.

Figure 5A plots the evolution, during pruning, of the training error "X", of the test error "O" and of the Akaike's FPE estimate "+", with respect to parameter's number. Starting from the initial learning (with 61 parameters), the plot should be read from right to left. Figure 5B gives the final architecture of the pruned

network. At each parameter deletion, retraining for 50 iterations maximum is performed. According to the FPE (Final Prediction Error), the optimal network architecture was the one comprising 9 hidden neurons and 38 parameters.

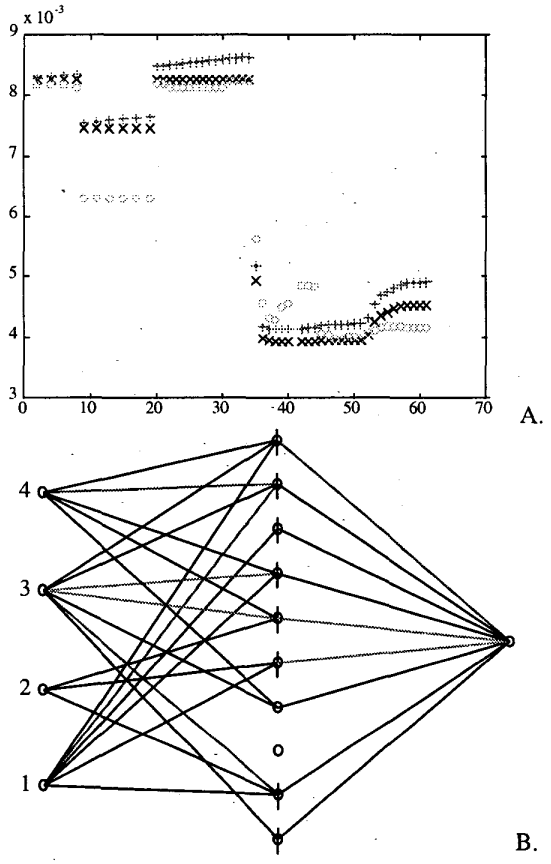

Figure 5: (A) Training error, test error and FPE with respect to parameter number
(B) Architecture of the pruned network

In [18], simulation examples show the importance of the quality of the model obtained after initial learning. Nevertheless, in order to focus on pruning process, the second way uses modified pruning presented in section 4 from the initial structure of 10 hidden neurons, with 61 parameters estimated as before with standard LM algorithm.

Figure 6A plots, during robust pruning, the evolution of the training error "X", of the test error "O" and of the Akaike's FPE estimate "+" with respect to the parameter's number. According to the FPE, the optimal network architecture, presented figure 6B, is the one comprising 6 hidden neurons with 19 parameters.

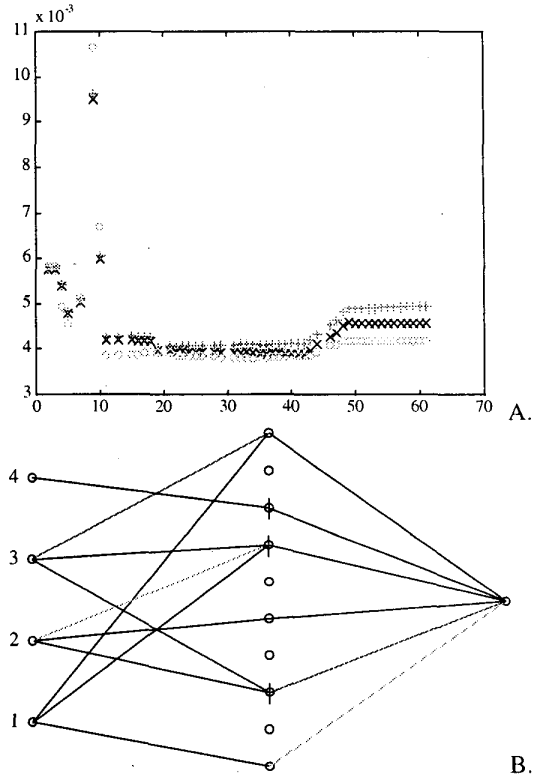

Figure 6: (A) Training error, test error and FPE with respect to parameter number
(B) Architecture of the pruned network

Comparisons are given in table 1. The "robust OBS" algorithm (ROBS) gives performances slightly better than the standard least squares OBS in terms of mean square error on the learning set (MSLE) and on the test set (MSTE), but with a number of parameters divided by 2. Moreover, the robust pruning process is performed quicker, the iteration number necessary for retraining at each weight deletion being clearly lower, with a stopping criterion on the gradient norm. Note, from figure 6A, that the network with only 11 parameters can be finally selected without significant worsening of the approximation performances.

	Before OBS	After OBS	After ROBS
# parameter	61	38	19
MSLE	$3.02\ 10^{-3}$	$3.95\ 10^{-3}$	$3.9\ 10^{-3}$
MSTE	$3.9\ 10^{-3}$	$4.3\ 10^{-3}$	$4.0\ 10^{-3}$

Table 1: Compared pruning results

Finally, figure 7 illustrates an example of the good results using the neural inverse model in feedforward control strategy, given figure 2. The reference signal and the

corresponding output V_X are plotted. The proposed strategy has been carried out in real time using Matlab-Simulink and RTW (Real Time Workshop) software environment, with sampling period of 3ms.

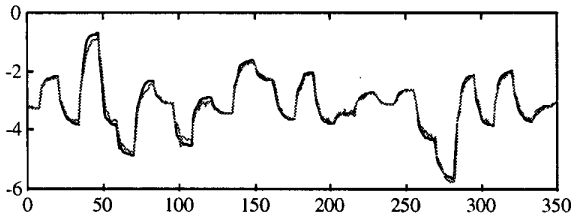

Figure 7: Reference trajectory and the output of the MagLev system

6. Conclusion

The use of artificial neural networks for control is motivated by their universal approximation capabilities. The feedforward one hidden layer perceptron with linear activation at the output gives a simple, but sufficient general structure for nonlinear modeling. In this paper, the usefulness of deleting spurious parameters and outlier-robust criterion for pruning are shown. The model size reduction is applied to the identification of the inverse model of a MagLev system for feedforward control. Obtaining a model with very few parameters and good approximation capabilities allows to envisage adaptive parameter estimation algorithms, which cannot be implemented in real time with standard environment, for systems requiring short sampling period, without minimal structure neural models.

7. References

[1] M. Agarwal, "A systematic classification of neural network based control", *IEEE Control Systems Mag.*, Vol. 17, 1997, pp. 78-84.

[2] G. Bloch, P. Thomas, M. Ouladsine and M. Lairi, "On several outlier-robust training rules of neural networks for identification of nonlinear systems", *8th Int. Conf. on Neural Networks and their Applications NEURAP'96,* Marseille, France, March 1996, pp. 13-19.

[3] G. Bloch, P. Thomas, and D. Theilliol, "Accommodation to outliers in identification of non linear SISO systems with neural networks", *Neurocomputing*, Vol. 14, n° 1, 1997, pp. 85-99.

[4] G. Bloch, F. Sirou, V. Eustache and P. Fatrez, "Neural intelligent control of a steel plant", *IEEE Trans. on Neural Networks*, Vol. 8, n° 4, July 1997, pp. 910-918.

[5] S. Chen, S.A. Billings, and P.M. Grant, "Non-linear system identification using neural networks," *Int. J. Control*, Vol. 51, n° 6, 1990, pp. 1191-1214.

[6] D.S. Chen and R.C. Jain, "A robust back propagation learning algorithm for function approximation", *Third Int. Workshop on Artificial Intelligence and Statistics*, Fort Lauderdale, FL, January 1991, pp. 218-239,

[7] S.J. Hanson and D.J. Burr, "Minkowski-r back-propagation: learning in connectionist models with non-Euclidean error signals", in *Neural Information Processing Systems*, D.Z. Anderson (Ed.), American Institute of Physics, New-York, 1988, pp. 348-357.

[8] B. Hassibi and D.G. Stork, "Second order derivatives for network pruning: optimal brain surgeon", in *Advances in Neural Information Processing Systems*, S.H. Hanson, J.D. Cowan and C.L. Gilles (Eds.), Morgan Kaufmann, San Mateo, CA, Vol. 5, 1993, pp. 164-171.

[9] K. Hunt, D. Sbarbaro, R. Sbikowski, and P.J. Gawthtop, "Neural networks for control systems - a survey", *Automatica*, Vol. 28, 1992, pp. 1083-1112.

[10] C. Jutten and O. Fambon, "Pruning methods: A review", *Proc. European Symp. on Artificial Neural Networks ESANN'95*, Brussels, April 1995, pp. 129-140.

[11] M. Lairi, "Identification et commande neuronales de systèmes non linéaires : application à un système de sustentation magnétique", Thèse de Doctorat de l'Université Henri Poincaré-Nancy 1, spécialité Automatique, 1998.

[12] M. Lairi, G. Bloch, and G. Millerioux, "Real time feedforward neural control of a MagLev system", *Neural Processing Letters*, Kluwer Academic Publishers, 1999, submitted.

[13] K. Liano, "Robust error for supervised neural network learning with outliers", *IEEE Trans. on Neural Networks*, Vol. 7, n° 1, 1996, pp. 246-250.

[14] K.S. Narendra, and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Trans. on Neural Networks*, 1990, Vol. 1, pp. 4-27.

[15] M. Norgaard, "Neural Network Based System Identification Toolbox", Tech. Report 95-E-773, Institute of Automation, Technical. University of Denmark, 1995.

[16] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P.Y. Glorennec, H. Hjalmarsson, and A. Juditsky, "Nonlinear black-box modeling in system identification: a unified overview", *Automatica*, Vol. 31, n° 12, 1995, pp. 1691-1724.

[17] P. Thomas and G. Bloch, "From batch to recursive outlier-robust identification of non linear dynamic systems with neural networks", *Proc. IEEE Int. Conf. on Neural Networks ICNN'96*, Vol. 1, Washington, DC, June 1996, pp. 178-183.

[18] P. Thomas and G. Bloch, "Robust pruning for multilayer perceptrons", *Proc. IMACS/IEEE Multiconference on Computational Engineering in Systems Applications CESA'98,* P. Borne, M. Ksouri and A. El Kamel (Eds.), Vol. 4, Nabeul-Hammamet, Tunisia, April 1998, pp. 17-22.