

A SCHEME FOR NONLINEAR MODELING

World Multiconference on Systemics, Cybernetics and Informatics (SCI '99), 5th International Conference on Information Systems Analysis and Synthesis (ISAS '99), July-August 1999, Orlando, USA, Volume 5, pp. 450 - 456.

JÜRGEN VAN GORP

Department ELEC, Vrije Universiteit Brussel
Pleinlaan 2, B-1050 Brussel, Belgium
Email: Jurgen.Van.Gorp@vub.ac.be

JOHAN SCHOUKENS

Department ELEC, Vrije Universiteit Brussel
Pleinlaan 2, B-1050 Brussel, Belgium
Email: Johan.Schoukens@vub.ac.be

ABSTRACT

Neural Networks (NN) and Fuzzy Logic (FL) systems are often considered as separate fields, apart from the classic linear domain. Both modeling methods are closely related to the Artificial Intelligence domain and as such are sometimes considered as special modeling schemes that come with inherited human behaviour. A correct placement of FL and NN within the System Identification (SI) theory however, can only be for the best of all three domains. This paper situates both NN and FL modeling schemes within a general framework for nonlinear modeling. Overlaps between the schemes are pointed out and some information is given how NN and FL can profit of the knowledge that is already present within nonlinear system identification, such as learning with noisy input measurements.

I. INTRODUCTION

Neural Network identification started as a method to imitate the human brain. Within computer science NN were considered as the base for the sixth generation computers that would be capable to human reasoning and intelligence. Fuzzy Logic systems were often looked at in the same way, as they were given the properties of human linguistics and human reasoning. Even to date both modeling techniques are believed to contain hidden intelligence that is able to solve a problem whenever the experimenter doesn't know what he is doing. This paper goes for a demystification of both modeling schemes. In practise both NN and FL are mere mathematical equations that map inputs to outputs, i.e. they form a multidimensional mapping from $\mathbf{u} \in \mathbb{R}^n$ to $\mathbf{y} \in \mathbb{R}^m$ via a nonlinear function $f_{NL}: \mathbb{R}^n \rightarrow \mathbb{R}^m$ with parameters θ , or

$$\mathbf{y} = f_{NL}(\mathbf{u}, \theta) \quad (1)$$

The functional f_{NL} can be a static NFIR, NARX or NARMAX mapping [28] [44] or a dynamic system with recurrent outputs [32]. The parameters θ can be time dependent. Regardless of the model, the goal of the *modeling* (also called *mapping* within NN and FL) is to find the proper model parameters θ . In general the identification is done in three major

steps, as shown in Fig. 1:

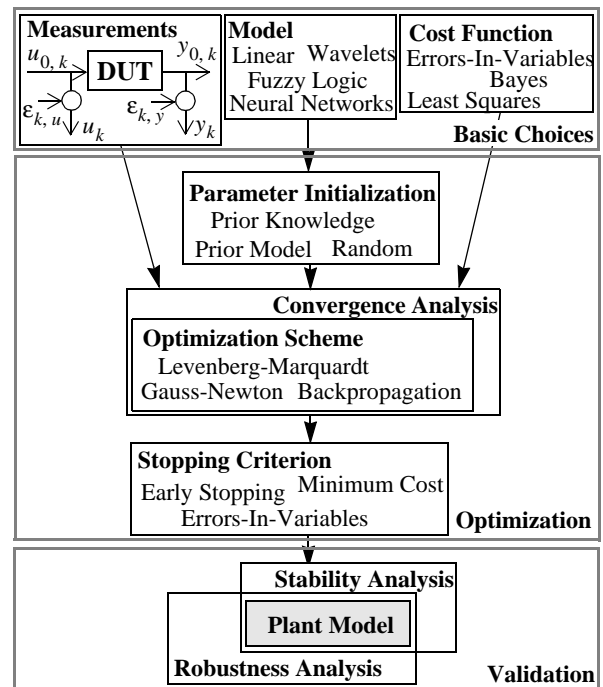


Fig. 1 Nonlinear modeling scheme

- 1) Take basic decisions prior to the modeling. Which measurements are needed for training? Which linear or nonlinear model is used, and which cost function is minimized?

- 2) Minimization of the cost function. This is the learning part (or adaptive part) of the model. The parameters are initialized and a learning (optimization) scheme is chosen. The experimenter must also decide upon when to stop optimization.
- 3) Validation of the model. Usually the model is tested on a separate test set, or on new data. Optionally robustness and stability analysis is performed on the model.

These steps and are treated more in detail in the following sections. The section headings correspond to the different sub parts of Fig. 1.

II. MEASUREMENTS

It is impossible to identify a model without measurements that fully cover the wanted behaviour. Interpolation or generalization is very well possible with nonlinear models, even under conditions of sparse data [1] [41] [43] [49]. Extrapolation of a nonlinear model however usually leads to unpredictable results [52]. In general the modeled plant should be persistent excited, regardless of the model that is used (i.e. [15]). The basic decision is the choice of proper excitation signals that fully cover the wanted model behaviour.

Measuring the system is the only way of getting valuable information needed for modeling. In this scope one can consider the use of expert knowledge as a way of extracting information from the plant [36]. It is a severe misunderstanding to think that expanding the measurement set by adding random noise will add information. This technique is however often used in the field of NN [4] [13] to overcome overtraining or to increase generalization [1] [29]. The only effect of adding noise to the measurements lies in its influence on the cost function, leading to bias on the model parameters [54] [49]. A better technique to increase the number of measurements, is the use of interpolation [54].

III. MODEL

Regardless of the fact that one uses linear modeling [28] [38], a Neural Network mapping [4] [41] [44], a Fuzzy Logic System [62] [64] or any other nonlinear model, a model is just a mapping from a given input to a given output. This mapping is a mathematical representation dependent on static or time-dependent model parameters. Usually a White Box representation of a plant is favoured whenever precise knowledge on the plant is available. Sometimes it is impossible to extract

detailed information and Black Box [22] [55] or Grey Box modeling [12] [41] is used.

The basic choice here is the selection of the model. There have been many fierce discussions in favour or against different models. Yet, for any given continuous function that operates on \mathbb{R} it is possible to prove universal approximation properties [6] [19] [23] [35] [57] [59], such that from a theoretical point of view any model would fit the task. From a practical point of view however the best fit is the one that needs the least number of parameters. A sine wave is best fitted using Fourier analysis, while locally linear systems are best fitted using e.g. splines or a FL system. Choosing the model is fully based upon typical transfer functions, and convergence properties whenever the parameters are optimized. Apart from these discussions, one can make the following comments, purely based on the transfer functions f_{NL} of the system.

- 1) Linear systems

$$\mathbf{y} = \mathbf{A}\mathbf{u} \quad \theta = \mathbf{A} \quad (2)$$

The transfer function is globally defined and perfectly smooth (the second derivative equals zero). The properties of linear systems are well known, and identification methods and stability criteria are treated in detail in the literature. The optimization of the parameters is a convex problem with a single global minimum. Within the scope of linear systems we also consider the Fourier transformation.

- 2) Polynomial mappings

$$y = a_0 + a_1 u + a_2 u^2 + \dots \quad (3)$$

$$\theta = [a_0 \ a_1 \ a_2 \dots]^T \quad (4)$$

Polynomial mappings are excellent for modeling systems with known soft nonlinearities of a low order. Once the order of nonlinearity increases, the condition number of the Jacobian matrix tends to increase rapidly and learning becomes difficult. The optimization is a convex problem.

- 3) Splines are mainly an extension to polynomial mappings in order to overcome discontinuities in an otherwise soft nonlinear transfer function. Splines are polynomials that are activated once the input reaches a threshold.

- 4) Neural Networks have the relationship

$$\mathbf{y} = \sigma_1 \mathbf{A}_1 (\dots \sigma_n (\mathbf{A}_n \mathbf{u} + \beta_n)) + \beta_1 \quad (5)$$

with σ_i a nonlinear function that is applied element wise (e.g. $\sigma(x) = \tanh(x)$). The parameter space is chosen as

$$\theta = [\mathbf{A}_1 \beta_1 \mathbf{A}_2 \beta_2 \dots \mathbf{A}_n \beta_n] \quad (6)$$

The input/output transfer function is typically a soft nonlinear relationship. Sometimes hard-bounded functions are used, such as $\sigma(x) = \text{sat}(x)$ or $\sigma(x) = \text{sign}(x)$, mainly for pattern recognition [4] and classification. The optimization of the parameters suffers from local minima and is often badly conditioned.

- 5) Fuzzy Logic Systems vary a lot with respect to the transfer functions. Modeling is done with a fuzzification part, inference mechanism, and a defuzzification part. Different models exist for each sub part. Two models rule within Fuzzy Logic: the Takagi-Sugeno systems and the Mamdani system. The Takagi-Sugeno model is defined in [33] as:

$$\mathbf{y} = \frac{\sum_{i=1}^R b_i \mu_i(\mathbf{u})}{\sum_{i=1}^R \mu_i(\mathbf{u})} \quad (7)$$

with $\mu_i(\mathbf{u})$ the membership degree of the fuzzified input and b_i a functional, e.g. the relationship

$$b_i = a_{i,0} + a_{i,1}u + a_{i,2}u^2 + \dots \quad (8)$$

R is the number of rules that make up the inference mechanism. The Mamdani model defined as a Generalized Fuzzy System in [23] uses the special case that $b_i = a_{i,0}$. Usually FL systems are configured as locally linear systems with a smooth interpolation from one operating point to the other. The optimization depends on the chosen parameters, but is difficult due to local minima and redundancy of the parameters. The latter is usually overcome by learning each sub part separately, or by fixing a sub part, e.g. the inference mechanism. A popular method is the feedforward calculation of the fuzzy sets and learn the parameters $a_{i,j}$ in (8).

- 6) Radial Basis Functions (RBF), defined in [6] [40] as

$$f_{RBF,i}(\mathbf{u}) = e^{-\frac{\|\mathbf{u} - \mathbf{c}_i\|^2}{2\sigma_i^2}} \quad (9)$$

with \mathbf{c}_i the centre vector and σ_i^2 the spread of the basis function. RBFs are much used for the mapping of local hard nonlinear features and are many times used in a NN configuration [15] such that $\sigma_i = f_{RBF,i}$. The hard nonlinear properties makes them also useable for classification [25]. An advantage of using RBFs is the ability to avoid local minima [3]

To fix the ideas, Fig. 3 shows a few typical transfer functions, based on a 2-input, single-output plant, such as given in Fig. 2.

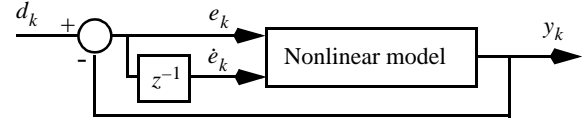


Fig. 2 2-inputs, single output controller

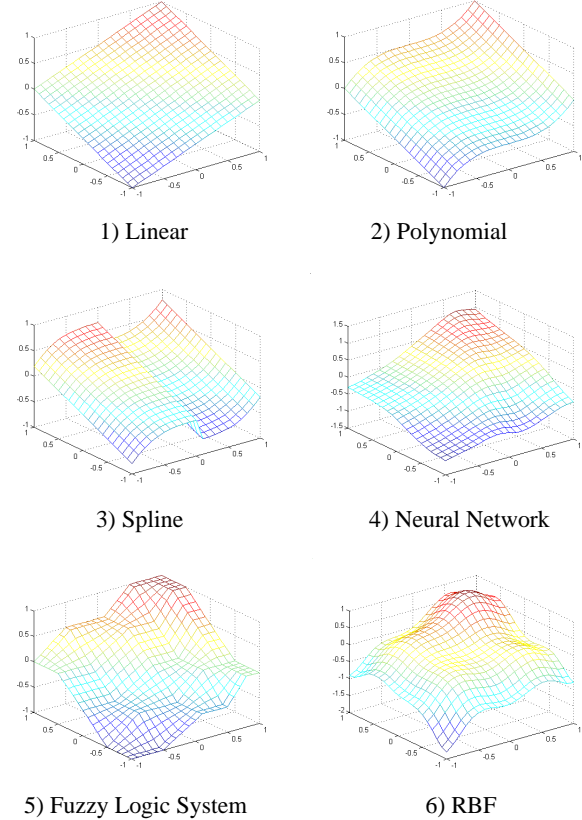


Fig. 3 Typical output planes for different models

The above list is not exhaustive. Sometimes very dedicated models are used, such as Feature Space Mapping [8], triangular nonlinear structures [37] or saturating linear systems [18]. The very general definitions for NN and FL makes them appropriate for mixed models, such as Neurofuzzy systems [20] [26] [58] or mixed with PI and PD controllers [31] [61] [63]. Overviews of nonlinear systems are given in [16] [22] [28] [42] [45] and [48]. All of these models are based on the same principle: how to provide a mathematical relationship between a given input and a given output. The remaining task is the choice of the model parameters that should be optimized or learned.

IV. THE COST FUNCTION

The proper choice of the *cost function* (*penalty function* in the NN and FL domain) is often

neglected and a simple Least Squares (LS) cost function

$$K_{LS} = \frac{1}{K} \sum_{k=1}^K \frac{1}{N} \sum_{i=1}^N (f_{NL,i}(\mathbf{u}^{[k]}, \theta) - y_i^{[k]})^2 \quad (10)$$

is taken with N the length of the output vector and K the number of measurement pairs $(\mathbf{u}^{[k]}, \mathbf{y}^{[k]})$. If the inputs and outputs are noisy, this LS cost function is not sufficient. In the case that only the measured outputs $\mathbf{y}^{[k]}$ are noisy with a known variance $\sigma_{y,k}^2$, a Weighted Least Squares cost function (or Bayesian cost function in the theory of NN)

$$K_{WLS} = \frac{1}{K} \sum_{k=1}^K \frac{1}{N} \sum_{i=1}^N \frac{(f_{NL,i}(\mathbf{u}^{[k]}, \theta) - y_i^{[k]})^2}{\sigma_{y,k}^2} \quad (11)$$

can be used [4] [29] [53]. If also the inputs $\mathbf{u}^{[k]}$ are noisy with a known (calculated or estimated) variance $\sigma_{u,k}^2$, then the Errors-In-Variables cost function [51] [53]

$$K_{EIV} = \frac{1}{K} \sum_{k=1}^K \left[\frac{1}{N} \sum_{i=1}^N \frac{(f_{NL,i}(\hat{\mathbf{u}}^{[k]}, \theta) - y_i^{[k]})^2}{\sigma_{y,k}^2} + \frac{1}{M} \sum_{i=1}^M \frac{(\hat{u}_i^{[k]} - u_i^{[k]})^2}{\sigma_{u,k}^2} \right] \quad (12)$$

should be used, with $\hat{u}_i^{[k]}$ the estimated values for the true inputs and M the length of the input vector. It should be stressed that the only way to implement stochastic properties in the model, is by choosing the proper cost function. Neither the chosen model, nor the optimization method deal with the problem of noisy measurements. Choosing the wrong cost function leads to biasing and thus to faulty model parameters.

Except for the stochastic properties, other properties can be enforced by adding extra terms in the cost function. A way to limiting the number of model parameters is regularization [11] [41] which results in a cost function

$$W_{NL} = K(\mathbf{u}^{[k]}, \mathbf{y}^{[k]}, \theta) + \delta |\theta - \hat{\theta}|^2 \quad (13)$$

with θ some nominal guess of the true model parameters, $\hat{\theta}$ is the estimated parameter space, δ is a regularization parameter defined in [41], and K is any of the above cost functions. Other possibilities are the addition of stability [44] and robustness [31] constraints by adding boundary terms in the cost function [10].

V. PARAMETER INITIALIZATION

The easiest way to find initial model parameters is by starting from random parameters. In the field of FL the parameters are usually defined by linguistic rules [36], Fuzzy Patches [23] [24] or clustering [56] and many times further optimization of the fuzzy sets is omitted. When using the Errors-In-Variables cost function, it is observed that the optimization suffers from many local minima [39] [51] [53] such that the initial parameters are best chosen as the results of an Output Error minimization, using the LS cost function. In the field of Neurofuzzy, the advantages of FL (easy parameter initialization) and NN (well known optimization techniques) are combined.

VI. THE OPTIMIZATION SCHEME

What is called *optimization* within the field of System Identification, is called *learning* in NN and *adaptation* within the FL field. Yet the idea is just the same: change the model parameters such that the model performs better (“such that a cost function is minimal”). The optimization is done by recursively updating the model parameters

$$\theta_{t+1} = \theta_t + \Delta\theta_t. \quad (14)$$

The parameter update vector $\Delta\theta$ is called the *learning rule* in the NN field, and *optimization step* within system identification. Various methods exist for the calculation of the update vector. The choice is based on robustness of the optimization method, the memory needs, speed of optimization and the ability to avoid local minima. The most common methods are:

Gradient Descent step [2] [34], based on an update vector

$$\Delta\theta_t = -\eta J_t^T \quad (15)$$

with J_t the jacobian matrix at the discrete time t with elements $J_{i,j}$, which are calculated as $J_{i,j} = \partial e_n / \partial \theta_j$. The error e_n depends on the chosen cost function. In the case of the LS cost function, the error would be

$$e_n = f_{NL,i}(\mathbf{u}^{[k]}, \theta) - y_i^{[k]} \quad (16)$$

with $n = 1, \dots, K \times N$. The index j equals $j = 1, \dots, P$ with P the number of model parameters that must be optimized. η is the optimization rate (called *learning rate* within NN). Gradient methods can be rather slow, but have minimal memory needs. In order to improve the optimization speed, the learning rate is made

adaptive. Avoiding local minima is improved by implementing a momentum term [30] in the learning rate.

Gauss-Newton methods (also called *Backpropagation* in the NN field). $\Delta\theta$ is calculated as

$$\Delta\theta_t = -\eta J_t^T E_t \quad (17)$$

The error vector E is a column vector and equals $E_t = [e_n(t)]$. The learning rate can be adaptive and can be implemented with momentum term. The low memory needs make Gauss-Newton an ideal method for minimization of the Errors-In-Variables cost function.

The **Levenberg-Marquardt** (LM) optimization step is calculated as

$$\Delta\theta_t = -(J_t^T J_t + \lambda I)^{-1} J_t^T E_t \quad (18)$$

with λ the Marquardt factor. The need for inversion of the Hessian $H = J_t^T J_t$ make that LM needs a lot of calculation power and memory. Yet the method is very robust against local minima, which makes it a good choice for nonlinear minimization of the LS and WLS cost functions.

The **Monte Carlo** scheme bases the optimization step on a random walk, where $\Delta\theta$ is a random vector with a small variance. The update is only performed if the cost function decreases [4]. In the **Metropolis** variant of the scheme, the optimization step is accepted also when the cost function increases. The acceptance is based on a probability function p , e.g.

$$p = e^{(K_t - K_{t+1})}. \quad (19)$$

Both methods are known to be trapped easily in local minima. For that reason the Metropolis variant is often changed into **Simulated Annealing** [17]. A temperature T is defined with a high initial value and the acceptance of a higher cost function is based on a probability

$$p = e^{-\frac{(K_t - K_{t+1})}{T_t}}. \quad (20)$$

The algorithm easily breaks out of local minima whenever the temperature is high. During optimization the temperature is typically decreased exponential

$$T_{t+1} = \alpha T_t \quad (21)$$

with $\alpha < 1$ (typical between 0.8 and 0.99).

Other optimization schemes worth mentioning are Linear Quadratic Programming [4], Hebbian

Learning [9] and Genetic Algorithms [14]. Remark that none of the above algorithms have properties that are related to the quality of the measurements (noise), the cost function properties, or the model properties (apart from convex optimization). For that reason it has no sense to speak about “Backpropagation Neural Networks”, since backpropagation is a learning scheme, while the NN is a model.

VII. STOPPING CRITERION

The decision when to stop the optimization can be taken apart from the optimization routine itself. Stopping is not necessarily done when the cost function reaches its minimum. In the case of noisy measurements or when a Black Box model is chosen, one can resort to early stopping [41] or stopped training [50]. For early stopping a second measurement set (the validation set) is used, apart from the training set used for learning. Optimization is stopped whenever the cost function, applied on the validation set, starts increasing. This technique is well known in the field of NN, but as it is shown in Fig. 1, it is also applicable for other models.

VIII. STABILITY AND ROBUSTNESS ANALYSIS

Most stability theorems for nonlinear systems make use of a Lyapunov function [17] and expect the nonlinear function to be sector bounded [44]. The resulting stability theorem is usually too conservative for practical use and better theorems are yet to be found. A method to partially overcome conservatism is given by NLq theory [44], applied on NN. The stability of NN is also studied in [46]. The stability of FS is studied in [5] [21] [24] and [60]. The stability of systems with saturation is studied in [18] and [27] and the stability using Linear Matrix Inequalities (LMIs) in [27] and [47]. Robustness analysis of the resulting model is studied in [7] [21] and [44]. While the stability analysis is well known for linear systems, it is still a very complex matter for nonlinear systems. Simple methods (such as NLq) exist, but are many times too conservative.

IX. CONCLUSIONS

This paper introduced a general identification scheme for system modeling, clearly separating the model, cost function and optimization scheme. Although Neural Networks and Fuzzy Logic originate from the Artificial Intelligence domain, it is shown that they fit well in the identification

scheme as a nonlinear model. This paper contributes to the demystification of both modeling techniques. The gain for both NN and FL is that they can profit from present knowledge within nonlinear system identification, such as cost function properties and optimization schemes.

The given references show that the current bibliography can be situated within each stage of the identification scheme, thus justifying its use.

X. ACKNOWLEDGMENTS

This paper presents research results of the Belgian Programme on Interuniversity Poles of Attraction (IUAP 4/2), initiated by the Belgian State, Prime Minister's Office for Science, Technology and Culture; and the Flemish Government (GOA, IMMI2). The scientific responsibility rests with its authors.

XI. REFERENCES

- [1] An G., "The Effects of Adding Noise During Backpropagation Training on a Generalization Performance", *Neural Computation*, Vol. 8, 1996, pp. 643 - 674.
- [2] Baldi P., "Gradient Descent Learning Algorithm Overview: a General Dynamical Systems Perspective", *IEEE Transactions on Neural Networks*, Vol. 5, No. 1, January 1995, pp. 182 - 195.
- [3] Bianchini M., Frasconi P., Gori M., "Learning without Local Minima in Radial Basis Function Networks", *IEEE Transactions on Neural Networks*, Vol. 6, No. 3, May 1995, pp. 749 - 756.
- [4] Bishop C. M., "Neural Networks for Pattern Recognition", Clarendon Press Oxford, 1994.
- [5] Cao S. G., Rees N. W., Feng G., "Stability Analysis of Fuzzy Control Systems", *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, Vol. 26, No. 1, February 1996, pp. 201 - 204.
- [6] Chen T., Chen H., "Approximation Capability to Functions of Several Variables, Nonlinear Functionals, and Operators by Radial Basis Function Neural Networks", *IEEE Transactions on Neural Networks*, Vol. 6, No. 4, July 1995, pp. 904 - 910.
- [7] Dorato P., Tempo R., Muscato G., "Bibliography on Robust Control", *Automatica*, Vol. 29, No. 1, 1993, pp. 201 - 213.
- [8] Duch W., Diercksen G. H. F., "Feature Space Mapping as a Universal Adaptive System", *Computer Physics Communications*, North-Holland, 1994, pp. 1 - 30.
- [9] Fausett L., "Fundamentals of Neural Networks, Architectures, Algorithms and Applications", Prentice Hall, 1994.
- [10] Fletcher R., "Practical Methods of Optimization", John Wiley & Sons Ltd., 1991.
- [11] Foresee F. D., Hagan M. T., "Gauss-Newton Approximation to Bayesian Learning", *ICNN '97*, IEEE International Conference on Neural Networks, Alabama, USA, August 1997, pp. 1930 - 1935.
- [12] Forssell U., Lindskog P., "Combining Semi-physical and Neural Network Modeling: an Example of its Usefulness", *SYSID*, July 1997, Kitakyushu, Japan, pp. 795 - 798.
- [13] Geman S., Bienenstock E., Doursat R., "Neural Networks and the Bias/Variance Dilemma", *Neural Computation*, Vol. 4, 1992, pp. 1 - 58.
- [14] Goldberg D. E., "Genetic Algorithms in Search, Optimization & Machine Learning", Addison Wesley Longman Inc., 1998.
- [15] Gorinevsky D., "On the Persistency of Excitation in Radial Basis Function Network Identification of Nonlinear Systems", *IEEE Transactions on Neural Networks*, Vol. 6, No. 5, September 1995, pp. 1237 - 1244.
- [16] Grossberg S., "Nonlinear Neural Networks: Principles, Mechanisms, and Architectures", *Neural Networks*, Vol. 1, 1988, pp. 17 - 61.
- [17] Haykin S., "Neural Networks, a Comprehensive Foundation", Prentice Hall, 1999.
- [18] Hindi H., Boyd S., "Analysis of Linear Systems with Saturation Using Convex Optimization", *Proceedings of the 37th IEEE Conference on Decision & Control*, Tampa, Florida, USA, December 1998, pp. 903 - 908.
- [19] Hornik K., "Approximation Capabilities of Multilayer Feedforward Networks", *Neural Networks*, Vol. 4, 1991, pp. 251 - 257.
- [20] Jang J.-S. R., "ANFIS: Adaptive-Network-Based Fuzzy Inference System", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 23, No. 3, May/June 1993, pp. 665 - 685.
- [21] Johansen T. A., "Fuzzy Model Based Control: Stability, Robustness, and Performance Issues", *IEEE Transactions on Fuzzy Systems*, Vol. 2, No. 3, August 1994, pp. 221 - 234.
- [22] Juditsky A., Hjalmarsson H., Benveniste A., Delyon B., Ljung L., Sjöberg J., Zhang Q., "Nonlinear Black-box Models in System Identification: Mathematical Foundations", *Automatica*, Vol. 31, No. 12, 1995, pp. 1725 - 1750.
- [23] Kosko B., "Fuzzy Systems as Universal Approximators", *IEEE Transactions on Computers*, Vol. 43, No. 11, November 1994, pp. 1329 - 1333.
- [24] Kosko B., "Global Stability of Generalized Additive Fuzzy Systems", *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews*, Vol. 28, No. 3, August 1998, pp. 441 - 452.
- [25] Krzyzak A., Linder T., Lugosi G., "Nonparametric Estimation and Classification Using Radial Basis Function Nets and Empirical Risk Minimization", *IEEE Transactions on Neural Networks*, Vol. 7, No. 2, March 1996, pp. 475 - 487.
- [26] Lin C.-T., Juang C.-F., "An Adaptive Neural Fuzzy Filter and its Applications", *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, Vol. 27, No. 4, August 1997, pp. 635 - 656.
- [27] Liu D., Michel A. N., "Asymptotic Stability of Discrete-Time Systems with Saturation Nonlinearities with Applications to Digital Filters", *IEEE Transactions on Circuits and Systems: Fundamental Theory and Applications*, Vol. 39, No. 10, October 1992, pp. 798 - 807.
- [28] L. Ljung, "System Identification, Theory For The User", Prentice-Hall information and system sciences series, 1994, second edition: 1999.
- [29] MacKay D. J. C., "Bayesian Methods for Adaptive Models", PhD thesis, California Institute of Technology, Pasadena, California, 1992.
- [30] Masters T., "Practical Neural Network Recipes in C++", Academic Press, Inc., 1993.
- [31] Mudi R. K., Pal N. R., "A Robust Self-Tuning Scheme for PI- and PD-Type Fuzzy Controllers", *IEEE Transactions on Fuzzy Systems*, Vol. 7, No. 1, February 1999, pp. 2 - 16.

- [32] Narendra K. S., Parthasarathy K., "Identification and Control of Dynamical Systems Using Neural Networks", IEEE Transactions on Neural Networks, Vol. 1, No. 1, March 1990, pp. 4 - 27.
- [33] Passino K. M., Yurkovich S., "Fuzzy Control", Addison-Wesley, 1997.
- [34] Pearlmutter B. A., "Gradient Calculations for Dynamic Recurrent Neural Networks: a Survey", IEEE Transactions on Neural Networks, Vol. 6, No. 5, September 1995, pp. 1212 - 1226.
- [35] Poggio T., Girosi F., "Networks for Approximation and Learning", Proceedings of the IEEE, Vol. 78, No. 9, September 1990, pp. 1481 - 1497.
- [36] Ralescu A., Narazaki H., "Integrating Artificial Intelligence Techniques in Linguistic Modeling From Numerical Data", IFES '91 Fuzzy Engineering Toward Human Friendly Systems, Part II: Fuzzy Technology, pp. 328 - 337.
- [37] Seto D., Annaswamy A. M., Baillieul J., "Adaptive Control of Nonlinear Systems with a Triangular Structure", IEEE Transactions on Automatic Control, Vol. 39, No. 7, July 1994, pp. 1411 - 1428.
- [38] Schoukens J., Pintelon R., "Identification of Linear Systems", Pergamon Press, London, 1991.
- [39] Schoukens J., Pintelon R., Rolain Y., "Maximum Likelihood Estimation of Errors-In-Variables Models Using a Sample Covariance Matrix Obtained from Small Data Sets", Recent Advances in Total Least Squares Techniques and Errors-In-Variables Modeling, 1997, pp. 59 - 68.
- [40] Sherstinsky A., Picard R. W., "On the Efficiency of the Orthogonal Least Squares Training Method for Radial Basis Function Networks", IEEE Transactions on Neural Networks, Vol. 7, No. 1, January 1996, pp. 195 - 200.
- [41] Sjöberg J., "Non-Linear System Identification with Neural Networks", Linköping Studies in Science and Technology, Dissertation No. 381, Linköping University, 1995.
- [42] Sjöberg J., Zhang Qinghua, Ljung Lennart, Benveniste A., Delyon B., Glorennec P.-Y., Hjalmarsson H., Juditsky A., "Nonlinear Black-box Modeling in System Identification: a Unified Overview", Automatica, Vol. 31, No. 12, 1995, pp. 1691 - 1724.
- [43] Srinivasa N., "Learning and Generalization of Noisy Mappings Using a Modified PROBART Neural Network", IEEE Transactions on Signal Processing, Vol. 45, No. 10, October 1997, pp. 2533 - 2555.
- [44] Suykens J. A. K. et al., "Artificial Neural Networks for Modelling and Control of Non-Linear Systems", Kluwer Academic Publishers, 1996.
- [45] Suykens J. A. K., Bersini H., "Neural Control Theory: an Overview", A-Journal, Vol. 37, No. 3, 1996, pp. 4 - 10.
- [46] Tanaka K., "An Approach to Stability Criteria of Neural-Network Control Systems", IEEE Transactions on Neural Networks, Vol. 7, No. 3, May 1996, pp. 629 - 642.
- [47] Tanaka K., Ikeda T., Wang H. O., "A Unified Approach to Controlling Chaos via an LMI-Based Fuzzy Control System Design", IEEE Transactions on Circuits and Systems - I: Fundamental Theory and Applications, Vol. 45, No. 10, October 1998, pp. 1021 - 1040.
- [48] Tomlinson G. R., "Linear or Nonlinear - That is the Question", ISMA 19 - Tools for Noise and Vibration Analysis, 1990, pp. 11 - 31.
- [49] Twomey J. M., Smith A. E., "Bias and Variance of Validation Methods for Function Approximation Neural Networks Under Conditions of Sparse Data", IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews, Vol. 28, No. 3, August 1998, pp. 417 - 430.
- [50] Utans J., "Stopped Training via Algebraic On-line Estimation of the Expected Test-Set Error", ICNN '97, IEEE International Conference on Neural Networks, Alabama, USA, August 1997, pp. 1088 - 1092.
- [51] Vandersteen G., "Identification of Linear and Nonlinear Systems in an Errors-In-Variables Least Squares and Total Least Squares Framework", PhD thesis, Vrije Universiteit Brussel, Belgium, April 1997.
- [52] Van Gorp J., "Control of a Static Nonlinear Plant Using a Neural Network Linearization", IJCNN 1998, Anchorage, Alaska, pp.2136 - 2141.
- [53] Van Gorp J., "The Errors-In-Variables Cost Function For Learning Neural Networks With Noisy Inputs", Intelligent Engineering through Artificial Neural Networks, Volume 8, Smart Engineering Systems, 1998, pp. 141 - 146.
- [54] Van Gorp J., Schoukens J., Pintelon R., "Adding Input Noise to Increase the Generalization of Neural Networks is a Bad Idea", ANNIE 1998, Intelligent Engineering Systems Through Artificial Neural Networks, Volume 8, pp. 127 - 132.
- [55] Van Gorp J., "Steel Plant Modeling and Analysis Using a Neural Network", submitted for ANNIE 1999, Intelligent Engineering Systems Through Artificial Neural Networks, Volume 9, St. Louis, USA.
- [56] Wang L., Langari R., "Complex Systems Modeling via Fuzzy Logic", IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics, Vol. 26, No. 1, February 1996, pp. 100 - 106.
- [57] Wang L-X, Mendel J. M., "Fuzzy Basis Functions, Universal Approximation, and Orthogonal Least-Squares Learning", IEEE Transactions On Neural Networks, Vol. 3, No. 5, September 1992, pp. 807 - 814.
- [58] Werbos P. J., "Neurocontrol and Elastic Fuzzy Logic: Capabilities, Concepts, and Applications", IEEE Transactions on Industrial Electronics, Vol. 40, No. 2, April 1993, pp. 170 - 180.
- [59] Williamson R. C., Helmke U., "Existence and Uniqueness Results for Neural Network Approximations", IEEE Transactions on Neural Networks, Vol. 6, No. 1, January 1995, pp. 2 - 13.
- [60] Yang T., Yang L.-B., "The Global Stability of Fuzzy Cellular Neural Network", IEEE Transactions on Circuits and Systems - I: Fundamental Theory and Applications, Vol. 43, No. 10, October 1996, pp. 880 - 883.
- [61] Yoshioka M., Omatu S., "Self-tuning Neuro-PID Control Applied to an Inverted Pendulum System", A-Journal, Vol. 37, No. 3, 1996, pp. 17 - 21.
- [62] Zadeh L. A., "Fuzzy Sets and Applications", John Wiley & Sons, 1987.
- [63] Zhao Z.-Y., "Fuzzy Gain Scheduling of PID Controllers", IEEE transactions on System, Man, and Cybernetics, Vol. 23, No. 5, September/October 1993, pp. 1392 - 1398.
- [64] Zimmermann H.-J., "Fuzzy Set Theory, and its Applications", Kluwer Academic Publishers, 1990.