PARAMETER OPTIMIZATION FOR TAKAGI-SUGENO FUZZY MODELS — LESSONS LEARNT

Manfred Männle

Inst. for Computer Design and Fault Tolerance
Univ. of Karlsruhe, 76128 Karlsruhe, Germany
maennle@computer.org

Brokat Technologies AG
Industriestr. 3, 70565 Stuttgart, Germany
Manfred.Maennle@brokat.com

Abstract

This article describes an approach to automatically build a Takagi-Sugeno fuzzy model (TSK-model) based on a set of input-output data (system identification). Identifying rule-based fuzzy models consists of two parts: structure modeling, i. e. determining the number of rules and input variables involved respectively, and parameter optimization, i. e. optimizing the rules consequences and the location and steepness of the fuzzy sets. For structure modeling, we investigate several search heuristics, also proposed by Takagi, Sugeno, and Kang in [1, 2, 3] or by Nelles in [4]. In order to find a good model structure, such search heuristics make it necessary to optimize and then evaluate a large number of different candidate models. To be applicable to real world problems, the parameter optimization must be highly efficient. For this, we investigate the use of two gradient descent algorithms: standard gradient descent (backpropagation) and resilient propagation (RPROP) [5, 6]. The combination of a structure search with a fast parameter optimization yields a powerful modeling algorithm that is capable to identify large real world systems.

Several preceding publications showed the applicability of TSK fuzzy models to real world problems, e. g. [4, 7, 8]. In this paper we evaluate a number of varieties of TSK-like fuzzy models by performing a nonlinear regression benchmark of I. Frank [9]. We compare several types of fuzzy models that are constructed as combinations of different fuzzy sets, structuring algorithms, and parameter optimization techniques.

Keywords

Takagi-Sugeno fuzzy model, nonlinear system identification, parameter optimization, algorithm evaluation, benchmark.

0-7803-7087-2/01/$10.00 © 2001 IEEE

1 Introduction

The problem of finding a model that describes a technical system by using measured input-output data is known as black box modeling or identification [10, 11]. Identification of nonlinear systems, i. e. systems showing a nonlinear input-output relationship, is a difficult task. For this purpose, Takagi-Sugeno fuzzy models are widely investigated, since they are universal approximators and provide good interpolation and extrapolation characteristics [12, 13].

The proposed method of fuzzy modeling consists of two parts: structure modeling, i. e. determining the number of rules and input variables involved respectively, and parameter optimization, i. e. optimizing the location and steepness of the fuzzy sets. The generality of Takagi-Sugeno models makes it computationally costly to build them from data. In order to keep computational costs reasonably limited, previous approaches introduced simplifications, e. g. a predefined model structure [14, 15] or a limited parameter optimization [2, 4].

The key aspect of this work is to achieve a combination of both, a *flexible model structure* by using a bottom-up search method and a *full parameter optimization* by applying sophisticated parameter optimization techniques. Several choices of model structures and modeling techniques exist when designing a fuzzy model. In this paper we evaluate and compare 32 different TSK fuzzy modeling varieties.

2 The Fuzzy Model

For building TSK fuzzy models we use membership functions of either trapezoidal shape

$$F(u) := \max(1; \min(0; 0.5 + \sigma(u - \mu))) \tag{1}$$

or sigmoidal shape

$$F(u) := 1/\left(1 + e^{\sigma \cdot (u-\mu)}\right) \tag{2}$$

with the parameters μ and σ to be optimized. The parameter μ describes the location and σ describes the steepness of the membership function. Both types of membership functions are (at least piecewise) derivable, which is necessary to apply a gradient descent algorithm. Examples of trapezoidal and sigmoidal fuzzy sets are depicted in figures 1 and 2.

A fuzzy model contains D *fuzzy sets* F_d; $d = 1,\ldots,D$. The index $[d] \in \{1,\ldots,N\}$ denotes the input space dimension in which the fuzzy set F_d is valid. The *index set* I_r contains the indices of all fuzzy sets that appear in rule $\mathcal{R}_r$. A fuzzy set is valid in exactly one input space dimension and may occur in several rule premises. The *index set* J_d of the fuzzy set F_d; $d = 1,\ldots,D$

$$J_d := \{j : d \in I_j; j = 1,\ldots,R\}, \tag{3}$$

contains the indices of all rules that have F_d in their premise.

The fuzzy sets are acting as borders that divide the input space into subspaces (each of which is covered by a fuzzy rule). Fuzzy sets are always introduced pairwise, with one set initially covering the first half and the other set covering the second half of a region (see section 2.2). When optimizing the parameter μ without restrictions, the initially overlapping fuzzy sets con diverge, yielding an invalid fuzzy model. This problem may be avoided by linking the μ parameters of each fuzzy set pair. In this paper we investigate both kinds of models, having either *free* or *linked* μ parameters.

The fuzzy model may either contain rules with *Sugeno-type consequences* ($C = N$), i. e. a linear equation of the input variables, or rules with *Yasukawa-type consequences* ($C = 0$), i. e. only a constant. The consequence parameter vector is either $\vec{c} = (c_0, c_1,\ldots,c_N)$ or $\vec{c} = (c_0)$.

The *fuzzy rule* $\mathcal{R}_r$ has for the empty premise $I_r = \emptyset$ the general form

$$\texttt{if TRUE then } f_r = c_{0r} \underbrace{+c_{1r}\cdot u_1 + \ldots + c_{Nr}\cdot u_N}_{\text{optional}} \tag{4}$$

and for $I_r \neq \emptyset$ the form

$$\begin{aligned}&\texttt{if } u_{i_{1r}} \texttt{ is } F_{1r} \texttt{ and} \ldots \texttt{and } u_{i_{n_r r}} \texttt{ is } F_{n_r r} \texttt{ then}\\ &\qquad f_r = c_{0r} \underbrace{+c_{1r}\cdot u_1 + \ldots + c_{Nr}\cdot u_N}_{\text{optional}},\end{aligned} \tag{5}$$

where f_r denotes the consequence of rule $\mathcal{R}_r$.

Using this, the *fuzzy model* $\mathcal{M}$ consists of a set of R fuzzy rules $\mathcal{R}_r$; $r = 1,\ldots,R$, i. e.

$$\mathcal{M} := \{\mathcal{R}_1,\ldots,\mathcal{R}_R\}. \tag{6}$$

The *membership* w_r of $\vec{u}_m$ to the rule $\mathcal{R}_r$ is given by

$$w_r(\vec{u}_m) := \bigwedge_{i\in I_r} F_{ir}(u_{[i]m}) \tag{7}$$

and by choosing the *product* as t-norm we obtain

$$w_r(\vec{u}_m) = \prod_{i\in I_r} F_{ir}(u_{[i]m}). \tag{8}$$

The *normalized membership* $v_r(\vec{u})$ be

$$v_r(\vec{u}) := \frac{w_r(\vec{u})}{\sum\limits_{k=1}^{R} w_k(\vec{u})}. \tag{9}$$

Finally, the *model output* $\hat{y}(\vec{u})$ is calculated via *product inference* (Larsen) and *weighed average* by

$$\hat{y}(\vec{u}) = \sum_{k=1}^{R} v_k(\vec{u})\cdot f_k(\vec{u}) = \frac{\sum\limits_{k=1}^{R}\Big(w_k(\vec{u})\cdot f_k(\vec{u})\Big)}{\sum\limits_{k=1}^{R} w_k(\vec{u})}. \tag{10}$$

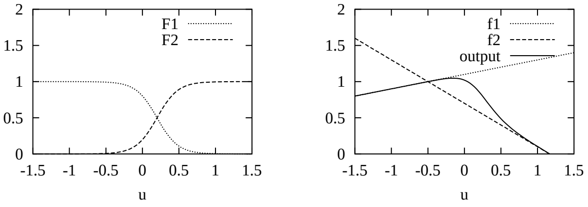

Figure 1: Sigmoidal fuzzy sets (left) and model output (right).

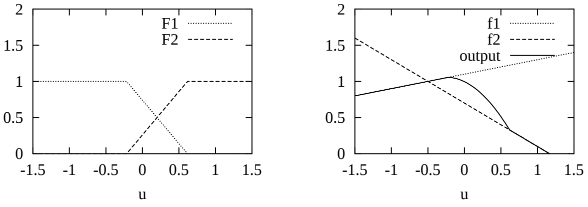

Figure 2: Trapezoidal fuzzy sets (left) and model output (right).

Figure 1 illustrates how the model output is constructed out of the rules consequences f_1 and f_2 for the fuzzy sets F_1 and F_2. Output is calculated for the model

$$\begin{aligned}&\mathrm{R}_1: \texttt{ if } u_1 \texttt{ is } F_1 \texttt{ then } f_1 = 1,1 + 0,2\cdot u_1\\ &\mathrm{R}_2: \texttt{ if } u_1 \texttt{ is } F_2 \texttt{ then } f_2 = 0,7 - 0,6\cdot u_1\end{aligned}$$

with parameters $\mu_{1,2} = -0,2$, $\sigma_1 = 7,0$ and $\sigma_2 = -7,0$.

Figure 2 shows the output for the same model but for trapezoidal fuzzy sets with parameters $\mu_{1,2} = -0,2$, $\sigma_1 = -1,2$ and $\sigma_2 = 1,2$.

2.1 Parameter Identification

Parameter optimization is designed to minimize the approximation error E_2 which is defined by the Euclidean norm L_2 for the model output (10) as

$$E_2 := \frac{1}{2}\|\vec{e}\|_2^2 = \frac{1}{2}\sum_{m=1}^{M}\Big(y_m - \sum_{q=1}^{R} v_q(\vec{u}_m)\cdot f_q(\vec{u}_m)\Big)^2. \qquad (11)$$

Identification of all, premise *and* consequence parameters is achieved through a gradient descent algorithm. In this work, we investigate two algorithms: standard gradient descent and resilient propagation.

Standard gradient descent, also known as backpropagation in a neural network context, performs parameter updates iteratively depending on the gradient of the target function ($\frac{\partial E_2}{\partial p_i}$). Details can be found for example in [16, 17, 18].

Resilient propagation (RPROP) is an enhanced gradient descent algorithm that was initially developed for neural network training. It has a resilient parameter update step which is based on a local adaption to the topology of the target function (E_2). Further details on RPROP can be found in [6, 5, 18, 19].

In order to apply a gradient descent algorithm, one needs to compute the derivatives of all parameters to be optimized, namely the consequence parameters $\frac{\partial E_2}{\partial c_{ir}}$ for all rules R_r and the premise parameters $\frac{\partial E_2}{\partial \mu_d}$ and $\frac{\partial E_2}{\partial \sigma_d}$ for all fuzzy sets F_d. The derivatives are given [8] and in more detail in [19].

2.2 Structure Identification

The optimal model structure can be found through a full combinatorial search which is inefficient for high dimensional input spaces. We want to identify processes with a large number N of inputs. This requirement imposes us to use some heuristics in order to find a good model structure in reasonable time. The optimal one-step search is a bottom-up search approach that solves this problem. (In the context of the defined fuzzy model structure, full combinatorial search would correspond to an R-step look-ahead search.)

The optimal one-step search algorithm, also described in e. g. [2, 4, 19], alternately determines a new model structure and then optimizes this structure. Within each epoch, the best structure of all investigated structures is saved and becomes the starting point for the next epoch. In every epoch r, each rule is refined in each input variable, yielding $r \cdot N$ candidate models to examine. Thus, at every epoch, the input space is once more divided by adding a new rule.

Figure 3 shows the possibilities of model refinements to examine. The algorithm starts with one rule containing a linear model. At the second epoch, assuming the refinement in the u_1 variable was best, we have a model with two rules. Then, the model is further refined by comparing all $2 \cdot 2$ possibilities yielding a three-rule model, where the refinement in dimension u_2 of $\mathcal{R}_2$ was best. Finally, further refinement of rule $\mathcal{R}_3$ yields a four-rule model at the fourth epoch.

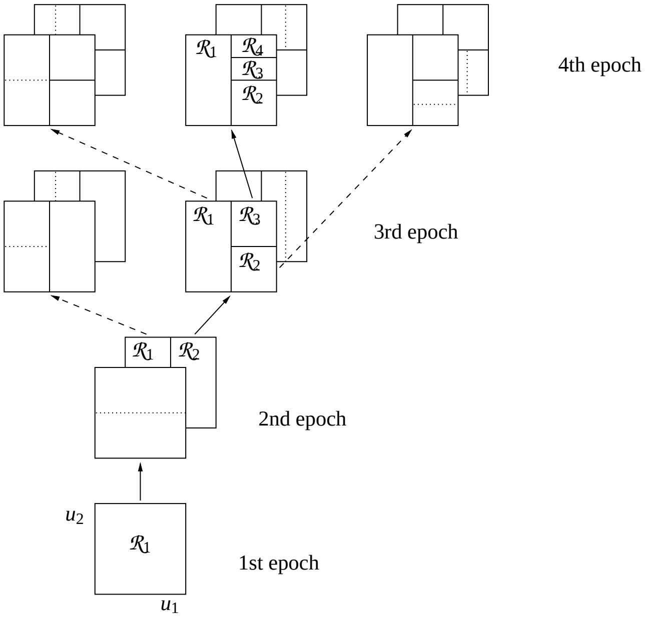

Figure 3: Structure search in a twodimensional input space.

The *full one-step search* algorithm finds the best structure that can be found using a one-step look-ahead and is in this sense optimal. For identification of a R-rule model with N inputs, altogether

$$1 + \sum_{r=1}^{R} N(r-1) = 1 + N\frac{(R-1)R}{2} \qquad (12)$$

models must be created, optimized, and compared.

The *shortcut one-step search* further reduces the number of models to examine by applying a simple idea: in each epoch, not all rules are further refined, but only the rule having the biggest approximation error, as this is the possibility most likely to achieve the best refinement. With this, only $1 + (R-1) \cdot N$ models need to be examined, yielding a model structure which is expected to be only slightly worse than the one found by a full one-step search.

3 Evaluation Method

So far, we defined two different fuzzy sets (trapezes and sigmoids), two rule premise types (with free or linked μ parameters), two different rule consequence types (Sugeno and Yasukawa), two parameter optimization techniques (standard gradient descent and RPROP), and two structure identification algorithms (full and shortcut one-step search). In this section we investigate the achievable model accuracies and compare all 32 model varieties by performing a nonlinear regression benchmark of I. Frank [9].

The benchmark is a Monte Carlo simulation of 40 different linear and nonlinear identification tasks. The model quality is assessed using the R^2 criterion given by

$$\mathrm{R}^2 = 1 - \frac{s_e^2}{s_y^2} = 1 - \frac{\sum_{m=1}^{M} (e_m - \overline{e})^2}{\sum_{m=1}^{M} (y_m - \overline{y})^2} \qquad (13)$$

with the modeling error e defined as $e = y - \hat{y}$.

The benchmark contains four four-dimensional static functions (A–D) and an example based on measured data from food industry (E).

A: $y = u_1 + u_2 + u_3 + u_4$

B: $y = 2(u_1 - 1)^2 + 2(u_2 - 1)^2 + \sin(2\pi u_3) + \ln(u_4 + 0.1)$

C: $y = \sin(u_1 + u_2 + u_3 + u_4) + \ln(u_1 + u_2 + u_3 + u_4)$

D: $y = 10 \sin(u_1 u_2) + 20(u_3 - 0.5)^2 + 5u_4$

Data of E can be found in [9, 19]. Each model is examined with four levels of white noise which is added with signal-to-noise ratios SNR = 1, 3, 7, and ∞, respectively. This corresponds to maximally reachable R^2 values of 0.5, 0.9, 0.98, and 1.0. Besides that, each model is also created with four additional input variables that only contain white noise. Altogether, there are 5x4x2 = 40 models to be identified. For all these 40 models, training sets (100 points) and test sets (1000 points) are generated by choosing the u-variables uniformly in $[0,1]$. Data generation, modeling, and test is performed fifty times and a final average R^2 is calculated.

4 Results

4.1 Runtime Complexity

Regarding the resulting derivations and parameter update formulae we find that the complexity of one gradient descent iteration for input dimension N, R rules and M patterns is $O(RNM)$ calculation steps[1] [8, 19]. This result is remarkable, since this is the same complexity as of a whole simulation run (i. e. feedforward step for all M patterns). This means that parameter optimization using a gradient descent algorithm scales the same way as using the model for simulation.

Additional computing resources are needed for structure identification. Full one-step search needs $O(NR^2)$ models to be examined (12), yielding an overall runtime complexity of $O(N^2MR^3)$.

4.2 Memory Complexity

Memory complexity is $O(NM)$ [19], i. e. the number of needed memory cells is mainly determined by the number and dimension of the data patterns used for model identification.

4.3 Accuracy

Accuracy is measured using the R^2 criterion (13). A higher R^2-value means a better model accuracy.

Sugeno-type models using RPROP, trapeze fuzzy sets with linked μ, and full one-step search achieved the best results within the many investigated varieties of TSK models. Table 1 shows the comparison of model accuracy of this TSK model with the accuracy of four modeling algorithms from [9]: MARS (multivariate adaptive regression splines), ACE (alternating conditional expectations), SMART (smart smooth multiple additive regression), and CART (classification and regression trees). The table contains mean R^2 values over all forty models. A value of $(1.0 + 0.98 + 0.9 + 0.5)/4 = 0.845$ means an optimal identification.

algorithm:	TSK	MARS	ACE	SMART	CART
∅ R^2:	0,722	0,695	0,605	0,573	0,408

Table 1: Comparison of five modeling algorithms.

The benchmark and other simulated and real world identification examples (turbo charger [7], heat exchanger [19], tank system [8]) show that the model accuracy of the proposed TSK model is comparable with and sometimes better than the accuracy of other modern modeling techniques.

4.4 Comparison of Different Model Types

In section 2 we defined two different fuzzy sets (trapezes and sigmoids), two rule premise types (with free or linked μ parameters), two different rule consequence types (Sugeno and Yasukawa), two parameter optimization techniques (standard gradient descent and RPROP), and two structure identification algorithms (full and shortcut one-step search).

Table 2 shows the mean R^2-values over all 40 modeling tasks of all 32 varietes (maximally achievable R^2 is 0.845). The most interesting results are:

- The choice of shape of the fuzzy set (trapeze or sigmoid) has almost no influence on the achievable model accuracy.

[1] For the O-calculus see [20].

		trapeze				sigmoid			
		μ free		μ linked		μ free		μ linked	
		Sug.	Yas.	Sug.	Yas.	Sug.	Yas.	Sug.	Yas.
full	RP.	0.72	0.68	0.72	0.67	0.71	0.70	0.72	0.70
	g.d.	0.72	0.68	0.72	0.64	0.72	0.69	0.71	0.63
short	RP.	0.70	0.60	0.70	0.58	0.70	0.65	0.71	0.65
	g.d.	0.69	0.60	0.70	0.54	0.71	0.64	0.70	0.54

Table 2: Mean quality values R^2 of different model types.

- Also linking the μ parameters does not reduce model accuracy (although models with linked μ have less free parameters). Therefore, the gain in readability using linked μ can be achieved without loss of accuracy.

- As expected, Yasukawa-type models are less accurate than Sugeno-type models. Still good (with accuracies from 0.67 to 0.70) are Yasukawa models combined with full one-step search and RPROP.

- Shortcut one-step search yields, except for Yasukawa-type models, almost the same accuracy than full search. The gain in computing speed can therefore be used under little loss of accuracy.

- Both, standard gradient descent and RPROP achieve about the same model accuracies. Since RPROP is faster and much easier to use (no algorithm parameters such as learning rate or momentum to choose), RPROP is to be preferred.

		trapeze				sigmoid			
		μ free		μ linked		μ free		μ linked	
		Sug.	Yas.	Sug.	Yas.	Sug.	Yas.	Sug.	Yas.
full	RP.	0.32	0.30	0.33	0.32	0.31	0.28	0.32	0.33
	g.d.	0.34	0.31	0.34	0.31	0.33	0.32	0.32	0.29
short	RP.	0.33	0.26	0.34	0.28	0.32	0.25	0.33	0.30
	g.d.	0.34	0.27	0.34	0.25	0.33	0.28	0.33	0.24

Table 3: Mean R^2 of different model types, $SNR = 1$.

		trapeze				sigmoid			
		μ free		μ linked		μ free		μ linked	
		Sug.	Yas.	Sug.	Yas.	Sug.	Yas.	Sug.	Yas.
full	RP.	0.92	0.88	0.92	0.84	0.92	0.90	0.92	0.86
	g.d.	0.89	0.84	0.91	0.79	0.92	0.87	0.90	0.80
short	RP.	0.88	0.77	0.87	0.72	0.89	0.84	0.89	0.82
	g.d.	0.85	0.76	0.88	0.68	0.89	0.82	0.88	0.70

Table 4: Mean R^2 of different model types, $SNR = \infty$.

So far, we investigated the model quality in terms of mean R^2-values over all noise levels. It is also interesting to investigate the accuracies achieved for single noise levels, in particular for $SNR = 1$ and $SNR = \infty$. The tables 3 and 4 show the mean R^2 with maximally achievable values of 0.5 for $SNR = 1$ and 1.0 for $SNR = \infty$. Interesting results are:

- Under absence of noise, accuracies using RPROP are better than those using standard gradient descent. RPROP seems to perform a deeper optimization, which turns out as an advantage for data poor of noise and as a disadvantage for data containing much noise.

- Especially in the case of noiseless data, Sugeno-type models can provide significantly higher accuracies than Yasukawa-type models.

5 Conclusions

Structure identification using a (full or shortcut) one-step search combined with a gradient descent algorithm for parameter optimization (preferably RPROP) yields an automatic Takagi-Sugeno modeling approach that is capable to identify real world problems [4, 7, 19].

Several construction approaches exist for TSK models. The investigated method iteratively splits the input space into smaller parts and globally optimizes all model parameters without restrictions. This yields small models with a low number of parameters, but the optimization algorithm must be “guarded” in order to avoid degenerated models (e. g. by linking the μ parameters).

Using a global parameter optimization may produce a model where the rules do not reflect local system behavior. This is because rules may span large regions of the input space (i. e. are too fuzzy). If, e. g. for better interpretability, you want rules reflecting local system behavior, it is furthermore necessary to enforce a minimum steepness σ of the fuzzy sets during the parameter optimization procedure.

Another approach to solve this problem is to restrict numerical optimization to the rules consequences or to explicitly optimize locally as in [4] or [21]. The resulting models may need some more rules (i. e. more parameters) to achieve the same accuracy, but are easier to handle.

The benchmark has shown that for measured data containing little noise ($SNR > 10$), *Sugeno-type models optimized with RPROP* provide the best accuracy within all modeling varieties investigated in this work. Calculation can then be sped up using a shortcut search without major loss of model quality. If you prefer Yasukawa-type models, e. g. for better interpretability, best choice is using RPROP combined with a full one-step structure search.

References

[1] Tomohiro Takagi and M. Sugeno, "Fuzzy identification of systems and its application to modeling and control", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 15, no. 1, pp. 116–132, 1985.

[2] M. Sugeno and G. Kang, "Structure identification of fuzzy model", *Fuzzy Sets and Systems*, vol. 26, no. 1, pp. 15–33, 1988.

[3] M. Sugeno and T. Yasukawa, "A fuzzy-logic-based approach to qualitative modeling", *IEEE Transactions on Fuzzy Systems*, vol. 1, no. 1, pp. 7–31, 1993.

[4] Oliver Nelles, *Nonlinear System Identification with Local Linear Fuzzy Models*, Shaker Verlag, Dissertation, Techn. Universität Darmstadt, 1999.

[5] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation: the RPROP algorithm", in *Proceedings of the IEEE Int. Conf. on Neural Networks (ICNN)*, 1993, pp. 586–591.

[6] M. Riedmiller, "Advanced supervised learning in multi-layer perceptrons — from backpropagation to adaptive learning algorithms", *Computer Standards and Interfaces*, vol. 16, no. 3, pp. 265–278, 1994.

[7] M. Männle, "Schnelle Takagi-Sugeno Fuzzy Modellierung", in *VDI Berichte: GMA+GI Fachtagung Computational Intelligence*, Baden-Baden, 11./12. Mai 2000, pp. 323–328.

[8] M. Männle, "FTSM — Fast Takagi-Sugeno Fuzzy Modeling", in *Proceedings of SAFEPROCESS 2000*, Budapest, 14-16 june 2000, pp. 663–668.

[9] I. E. Frank, "Modern nonlinear regression methods", *Chemometrics and Intelligent Laboratory Systems*, vol. 27, pp. 1–19, 1995.

[10] Jonas 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, no. 12, pp. 1691–1724, 1995.

[11] Lotfi A. Zadeh, "The role of fuzzy logic in modeling, identification and control", *Modeling, Identification, and Control*, vol. 15, no. 3, pp. 191–203, 1994.

[12] H. Ying, "Sufficient conditions on uniform approximation of multivariate functions by general Takagi-Sugeno fuzzy systems with linear rule consequent", *IEEE Transactions on Systems, Man, and Cybernetics–Part A*, vol. 28, no. 4, pp. 515–520, 1998.

[13] H. Ying, Y. Ding, S. Li, and S. Shao, "Comparison of necessary conditions for typical Takagi-Sugeno and Mamdani fuzzy systems as universal approximators", *IEEE Transactions on Systems, Man, and Cybernetics–Part A*, vol. 29, no. 5, pp. 508–514, 1999.

[14] J.-S. R. Jang, "ANFIS: Adaptive-network-based fuzzy inference system", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, no. 3, pp. 665–684, 1993.

[15] J.-S. R. Jang and C. Sun, "Neuro-fuzzy modeling and control", in *Proceedings of the IEEE*, 1995, vol. 83(3), pp. 378–405.

[16] D. E. Rumelhart and J. McClelland, *Parallel Distributed Processing*, vol. 1, MIT Press, 1986.

[17] J. Hertz, A Krogh, and Palmer R., *Introduction to the theory of neural networks*, Addison-Wesley, 1991.

[18] A. Zell, N. Mache, T. Sommer, et al., *Stuttgart Neural Network Simulator, Users Manual, Version 3.2*, University of Stuttgart, june 1994.

[19] M. Männle, *Unscharfe Modelle zur Systemidentifikation und Fehlererkennung*, Dissertation, University of Karlsruhe, Germany, january 2001.

[20] B. Cormen, C. Leiserson, and R. Rivest, *Introduction to Algorithms*, MIT Press, 1992.

[21] J. Yen, L. Wang, and Gillespie C. W., "Improving the interpretability of TSK fuzzy models by combining global learning and local learning", *IEEE Transactions on Fuzzy Systems*, vol. 6, no. 4, pp. 530–537, 1998.