FTSM — FAST TAKAGI-SUGENO FUZZY MODELING

Manfred Männle

Institute for Computer Design and Fault Tolerance
University of Karlsruhe, D-76128 Karlsruhe, Germany
Manfred.Maennle@informatik.uni-karlsruhe.de

Abstract: Takagi-Sugeno type fuzzy models are widely used for model-based control and model-based fault diagnosis. They provide high accuracy with relatively small and easy to interpret models. The problem that we address in this paper is that data driven identification of such fuzzy models is computationally costly. Whereas most identification algorithms for Takagi-Sugeno models restrict the model's generality in order to simplify the identification, a different approach is taken here: we apply resilient propagation (RPROP), an efficient non-linear optimization technique, for parameter identification in order to achieve a fast Takagi-Sugeno modeling (FTSM) that is suited to model high-dimensional data sets containing a large number of data.

Keywords: fuzzy modeling, identification algorithms, nonlinear optimization, fault detection

1. INTRODUCTION

Takagi-Sugeno type fuzzy models (Takagi and Sugeno, 1985; Sugeno and Kang, 1986; Sugeno and Kang, 1988; Sugeno and Tanaka, 1991; Sugeno and Yasukawa, 1993), also being referred to as TSK-models (after Takagi, Sugeno, and Kang), are widely used for *model-based control* and *model-based fault diagnosis*. This is due to the model's properties of, on one hand being a general nonlinear approximator that can approximate every continuous mapping, and on the other hand being a piecewise linear model that is relatively easy to interpret (Johansen and Foss, 1995) and whose linear submodels can be exploited for control and fault detection (Füssel *et al.*, 1997; Ballé *et al.*, 1997).

The generality of TSK-like models makes the *data driven identification* of such models very complex. A fuzzy model consists of multiple rules, each rule containing a premise part and a consequence part. The premise part specifies a certain input subspace by a conjunction of fuzzy clauses that contain the input variables. The consequence part is a linear regression model. The identification task includes two subtasks: *structure identification*, like determination of the number of rules and the determination of the variables involved in the rule premises, and *parameter identification*, the estimation of the membership function parameters and the estimation of the consequence regression coefficients.

There is a possibility to address the three identification tasks separately: When fixing the structure and the premise parameters, consequence parameter identification becomes a linear optimization problem and can therefore be solved by a linear least mean squares optimization like singular value decomposition (SVD) (Klema and Laub, 1980; Männle, 1999). Premise parameter optimization remains in any case a *nonlinear optimization* problem. Finally, structure identification, when solved exhaustively, is a combinatorial search problem.

Because of this complexity, most TSK-identification algorithms simplify the model structure or apply heuristics or so-called meta-optimization techniques like genetic algorithms for structure identification and (at least for the nonlinear part of) parameter optimization: The algorithms LOLIMOT and Product Space Clustering, as described in (Nelles, 1999; Nelles *et al.*, 1999), do not compute a premise parameter optimization but determine the structure and premise parameters by either heuristic search or fuzzy clustering of the input-output space. In ANFIS (Jang, 1993; Jang and Sun, 1995), the model structure is predefined as a grid partition. Parameter optimization is then performed by a hybrid learning rule, a combination of least squares estimate and a gradient method (backpropagation). In (Yen *et al.*, 1998), the number of rules created by the grid partition is finally reduced by a pruning algorithm. The NFIN algorithm (Lin *et al.*, 1999) in a first step performs a heuristic input space partition (similar to clustering) and then optimizes parameters using backpropagation.

Predefining the structure by a grid partition is not suited for high-dimensional input spaces, since for a N-dimensional input space partitioned into K parts at each dimension the initial partition contains K^N rules which cannot be handled any more already for "medium-size" problems with, e. g., $k = 3$ and $N = 10$. We therefore use a bottom up approach yielding a tree partition as described in (Sugeno and Kang, 1988; Jang and Sun, 1995; Nelles *et al.*, 1999; Männle, 1999). The main problem when applying the heuristic search is its computational cost because a lot of models must be evaluated. So, the main idea of this work is to *apply a sophisticated optimization technique for parameter identification* which enables the use of the heuristic search even when being applied to high-dimensional problems. For this purpose, we chose resilient propagation (RPROP) (Riedmiller and Braun, 1993; Braun and Riedmiller, 1993; Zell *et al.*, 1994) because it is easy to apply (only needs first derivatives) but has a performance like second order methods as for example the Levenberg-Marquardt algorithm (Hagan and Menhaj, 1994).

The following sections provide a description of the fuzzy model, the identification procedure (including the derivatives necessary to apply RPROP), the identification of a nonlinear dynamical process, and a fault detection experiment.

2. FUZZY IDENTIFICATION

Fuzzy identification is done for MISO systems (multiple input single output) system, i. e., the model performs a mapping $\hat{y}$ from an N-dimensional input vector $u = (u_1, \ldots, u_N) \in \mathcal{U}_1 \times \cdots \times \mathcal{U}_N \subset \mathbb{R}^N$ to an output value $\hat{y} \in \mathcal{Y} \subset \mathbb{R}$.

2.1 *The Fuzzy Model*

Membership functions of TSK-models as used here have a trapezoidal shape

$$F(u) := \max(1; \min(0; 0.5 + \sigma(u - \mu))), \qquad (1)$$

with the parameters μ and σ to be optimized. The parameter μ describes the location and σ describes the steepness of the membership function. This type of membership function is piecewise derivable, which is necessary for applying RPROP. It is also possible to use sigmoidal shape membership functions (Männle, 1996), yielding comparable results.

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; $d = 1, \ldots, D$

$$J_d := \{ j : d \in I_j; j = 1, \ldots, R \}, \qquad (2)$$

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

Fuzzy rules may contain *"full" consequences* ($C = N$), i. e., a linear equation of the input variables (Takagi-Sugeno type) or *"simple" consequences* ($C = 0$), i. e., only a constant (Sugeno-Yasukawa type). The consequence parameter vector is either $c = (c_0, c_1, \ldots, c_N)$ or $c = (c_0)$.

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

$$\texttt{if TRUE then } f_r = c_{0r} \underbrace{+ c_{1r} \cdot u_1 + \ldots + c_{Nr} \cdot u_N}_{\text{optional}} \qquad (3)$$

and for $I_r \neq \emptyset$ the form

$$\begin{aligned} &\texttt{if } u_{i_{1r}} \texttt{ is } F_{1r} \texttt{ and} \ldots \texttt{and } u_{i_{n_r r}} \texttt{ is } F_{n_r r} \texttt{ then} \\ &\qquad f_r = c_{0r} \underbrace{+ c_{1r} \cdot u_1 + \ldots + c_{Nr} \cdot u_N}_{\text{optional}}, \end{aligned} \qquad (4)$$

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

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

$$\mathcal{M} := \{ \mathcal{R}_1, \ldots, \mathcal{R}_R \}. \qquad (5)$$

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

$$w_r(u_m) := \bigwedge_{i \in I_r} F_{ir}(u_{[i]m}) \qquad (6)$$

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

$$w_r(u_m) = \prod_{i \in I_r} F_{ir}(u_{[i]m}). \qquad (7)$$

The *normalized membership* $v_r(u)$ be

$$v_r(u) := \frac{w_r(u)}{\sum\limits_{k=1}^{R} w_k(u)}. \qquad (8)$$

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

$$\hat{y}(u) = \sum_{k=1}^{R} v_k(u) \cdot f_k(u) = \frac{\sum\limits_{k=1}^{R} \Big(w_k(u) \cdot f_k(u) \Big)}{\sum\limits_{k=1}^{R} w_k(u)}. \qquad (9)$$

2.2 *Structure Identification*

Problems of dimension $N \geq 10$ make a bottom up approach necessary. In this paper, a bottom up tree partition algorithm is applied. The optimal structure is determined by a heuristic search. The structure modeling starts with a one rule model that is further refined at each epoch by adding one rule, i. e., partitioning one of the models subspaces. At each epoch, the best partitioning (i. e., the rule to split and the dimension where to split) is determined by evaluation of all possibilities. The best performing model is then used as starting point for the next epoch. For further details on the heuristic search the reader may refer to (Sugeno and Kang, 1988; Jang and Sun, 1995; Nelles, 1999; Männle, 1999).

2.3 *Parameter Identification*

Parameter optimization minimizes the error E_2 which is defined by the Euclidean norm L_2 for the model output (9) as

$$E_2 := \frac{1}{2}\,\|\varepsilon\|_2^2 = \frac{1}{2}\sum_{m=1}^{M}\left(y_m - \sum_{q=1}^{R} v_q(u_m)\cdot f_q(u_m)\right)^2. \tag{10}$$

We also investigated the use of the L_1 norm. This usually results in slightly worse models, but the modeling is more robust in presence of outliers in the training data.

Identification of premise *and* consequence parameters is achieved through RPROP, a 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 (Riedmiller and Braun, 1993; Braun and Riedmiller, 1993; Zell *et al.*, 1994; Männle, 1996).

In order to apply RPROP, 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 in appendix A.The resulting formulae (A.3), (A.12), and (A.13) show many equal terms which allows an efficient implementation: The complexity of one RPROP iteration for input dimension N, R rules and M patterns is $O(RNM)$, the same as for a feedforward step of all M patterns!

3. FAULT DETECTION

Fault detection is performed in two steps: symptom generation and symptom evaluation.

3.1 *Symptom Generation*

There are different ways to gererate symptoms based on TSK-like models, see for example (Füssel *et al.*, 1997; Ballé *et al.*, 1997). In this paper, the output errors between model and process (residuals) during (closed loop) operation are used as fault symptoms.

In order to isolate single faults in a set of multiple faults it may be necessary to design symptoms. For this purpose exist several methods, as for example the parity space approach (Füssel *et al.*, 1997; Ballé *et al.*, 1997).

3.2 *Symptom Evaluation*

The easiest way to detect faults is to define borders for the residuals which can be tolerated and to fire an alarm if such a border is exceeded. In order to make the detection more sensitive while still keeping a low false alarm rate, methods for on-line detection of jumps in means as a moving average filter or a Page/Hinkley detector (Basseville, 1986) can be applied. Then, the tolerable deviations can be choosen considerably smaller.

To apply a Page/Hinkley detector, one must first define the two parameters μ_{inc} and μ_{dec} of tolerable deviations (of residual increase and decrease). Deviations of and greater than μ will be detected. The sensitivity of the detector is adjusted by choosing the parameter λ. A bigger λ makes the detection more robust, but also yields a bigger detection delay. See (Basseville, 1986) for further details.

4. EXAMPLE

4.1 *Tank System Identification*

In this section a simple nonlinear process is used as an example to show the modeling capabilities of the algorithm and how the identified model can be used for fault detection.

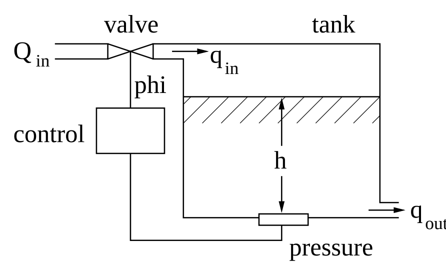

Figure 1. A simple tank system.

Figure 1 depicts a simple tank system. The fluid pours out a the rate q_{out} depending on the height h and the (constant) outlet plan a_{out}. The amount q_{in} of fluid flowing in depends on the angle φ of the valve, which is controlled depending on the fluid height, and the (constant) maximal fluid Q_{in}. The fluid height depends on the in/out streams and the tank plane A. The process can be described by the following equations:

$$q_{in}(t) = Q_{in} \cdot \sin(\varphi(t)), \quad \varphi \in [0, \pi/2] \tag{11}$$

$$q_{out}(t) = a_{out}\sqrt{2gh(t)}, \quad g = 9.81\text{ms}^{-2} \tag{12}$$

$$h(t) = h(0) + \frac{1}{A}\int_0^t (q_{in}(\tau) - q_{out}(\tau))\mathrm{d}\tau. \tag{13}$$

For identification and test, discrete time data series were generated with a sampling time of one second, $h(0) = 2\text{m}$, $A = 1\text{m}^2$, $a_{out} = 0.01\text{m}^2$, $Q_{in} = 0.12\text{m}^3\text{s}^{-1}$, and an activation φ shown in figure 2.

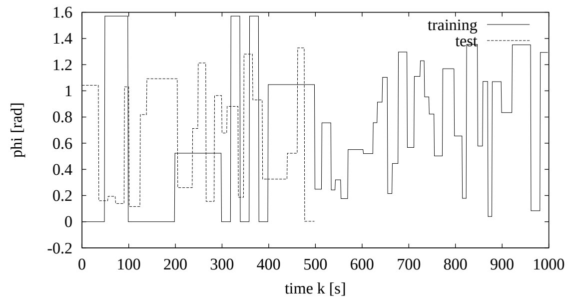

Figure 2. Activation φ (training and test data).

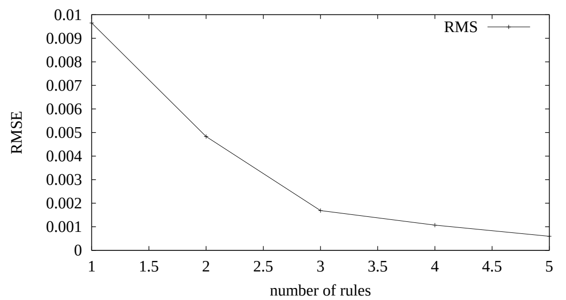

Figure 3. Modeling error during identification.

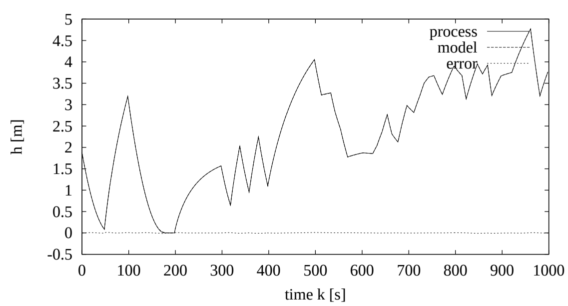

Figure 4. Simulation of tank system (training data).

The fluid level h at discrete time k is to be identified by the model, i. e. $y(k) = h(k)$. Identification is done using a series-parallel model:

$$\hat{y}(k) = f(\varphi(k-1), y(k-1)). \tag{14}$$

Figure 3 depicts the root mean square error of the first five models, starting with a one-rule linear model. The accuracy of the five-rule model is sufficient and identification can be stopped there. The simulation is performed with the parallel model:

$$\hat{y}(k) = f(\varphi(k-1), \hat{y}(k-1)). \tag{15}$$

The simulation of the training data is given in figure 4.

4.2 *Fault Detection Experiment*

In order to investigate the fault detection capabilities, a fault injection experiment with two faults is presented:

(1) sudden hardware fault: valve leakage of 0.3 % of Q_{in} from $k = 250$, and
(2) drift sensor fault: drifting from $k = 250$ to $k = 350$ the pressure sensor measures 0.75 % less than the real h

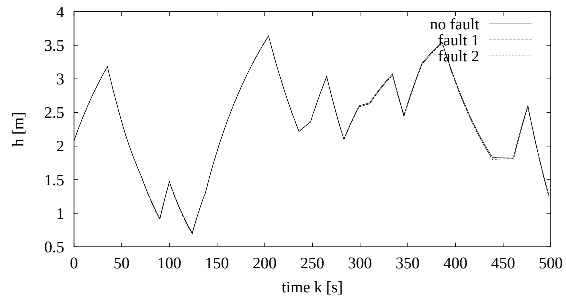

Figure 5. Simulation in presence of fault 1 and 2 (test data).

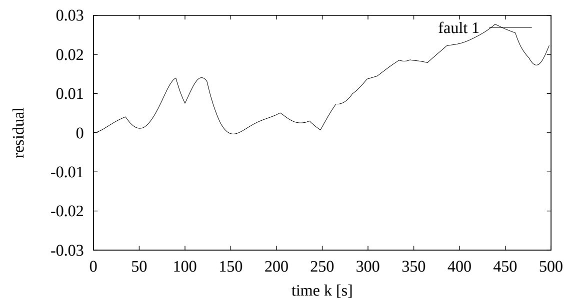

Figure 6. Residual of fault 1 (test data).

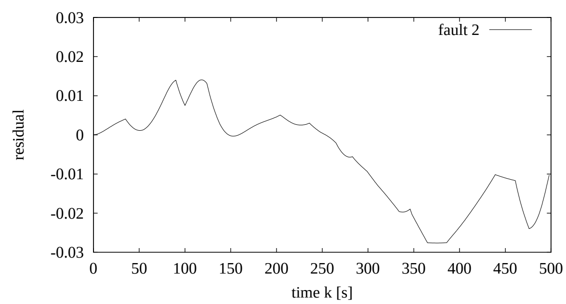

Figure 7. Residual of fault 2 (test data).

Figure 5 depicts the simulation of the test data under presence of fault 1 and fault 2. The residuals shown in the figures 6 and 7, i. e. the deviations from the model prediction and the real process, are small but still big enough to be reliably detectable through a Page/Hinkley detector.

During identification and tests it is found that the modeling error remains smaller than 0.015m under absence of faults. Therefore, a robust Page/Hinkley

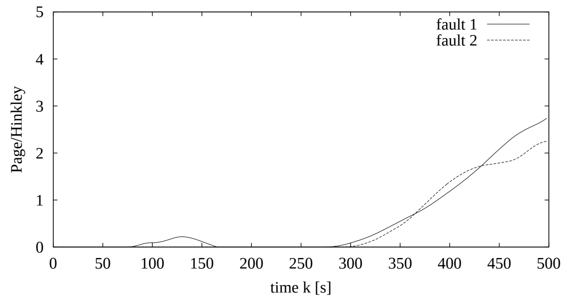

Figure 8. Detection of fault 1 and 2 (test data).

detector can be built by choosing the deviations to detect as for example $\mu_{inc} = \mu_{dec} = 0.015$m. Figure 8 shows the the result of the Page/Hinkley detection. The choice of a border $\lambda = 1$ would reliably detect the faults with a delay of about 120s.

5. CONCLUSIONS

TSK-like models combine the advantages of being general approximators that can reach high accuracy and being easy to interpret, since they are piecewise linear models that are represented in a quite natural way.

Owing to the generality of such models, the computational complexity of data-driven identification is very high. Therefore, the idea was to *apply one of the recent powerful nonlinear optimization techniques* for parameter optimization.

In the presented approach, RPROP, a sophisticated nonlinear optimization technique, is successfully applied to the problems of premise and consequence parameter optimization. The high efficiency of this parameter optimization allows to apply the *original heuristic search* for structure identification, first proposed in (Sugeno and Kang, 1988). This heuristic is relatively costly with respect to the number of models to be optimized and evaluated, but is well suited for high-dimensional and large problems, since it automatically determines the most important input variables and yields well-performing models with a low number of rules (bottom up approach). Even more exhaustive search strategies show only marginally better results (Johansen and Foss, 1995) and do not justify the additional computational costs.

We developed the identification procedure FTSM to *automatically build TSK-models based on large and high dimensional data sets*. The application to fault detection by the use of residuals is briefly described. The usage of a Page/Hinkley detector in order to achieve a more sensitive detection is presented and shown by a fault injection experiment with a simple tank system.

In our current work we further evaluate the capability of FTSM to model nonlinear processes and investigate methods for fault detection and isolation of multivariable processes.

Appendix A. DERIVATIVES FOR PARAMETER OPTIMIZATION

From (10) we get with $u_{0i} := 1$ for all consequence parameters c_{ir}, $r = 1,\dots,R$ and $i = 0,\dots,C$:

$$\frac{\partial E_2}{\partial c_{ir}} = \sum_{m=1}^{M} (y_m - \hat{y}_m)(-1) \sum_{q=1}^{R} v_q(u_m) \frac{\partial f_q(u_m)}{\partial c_{ir}} \qquad \text{(A.1)}$$

with

$$\frac{\partial f_q(u_m)}{\partial c_{ir}} = u_{im} \qquad \text{(A.2)}$$

yielding

$$\frac{\partial E_2}{\partial c_{ir}} = \sum_{m=1}^{M} \frac{-\varepsilon(u_m)}{\sum_{k=1}^{R} w_k(u_m)} \cdot w_r(u_m) \cdot u_{im}. \qquad \text{(A.3)}$$

With (10) we obtain the partial derivations of the fuzzy set parameters μ and σ for all fuzzy sets F_d, $d = 1,\dots,D$ as:

$$\frac{\partial E_2}{\partial \mu_d} = -\sum_{m=1}^{M} \left(\varepsilon(u_m) \cdot \sum_{r=1}^{R} f_r(u_m) \cdot \frac{\partial v_r(u_m)}{\partial \mu_d} \right). \qquad \text{(A.4)}$$

Derivating (8) for all examples $u_m, m = 1,\dots,M$, all rules $\mathcal{R}_r, r = 1,\dots,R$, and all fuzzy sets F_d, $d = 1,\dots,D$ yields for $r \notin J_d$

$$\frac{\partial v_r(u_m)}{\partial \mu_d} = \frac{-w_r(u_m)}{\left(\sum_{q=1}^{R} w_q(u_m) \right)^2} \cdot \sum_{j \in J_d} \frac{\partial w_j(u_m)}{\partial \mu_d} \qquad \text{(A.5)}$$

and for $r \in J_d$

$$\frac{\partial v_r(u_m)}{\partial \mu_d} = \frac{\left(\sum_{q=1}^{R} w_q(u_m) \right) \frac{\partial w_r(u_m)}{\partial \mu_d} - w_r(u_m) \sum_{j \in J_d} \frac{\partial w_j(u_m)}{\partial \mu_d}}{\left(\sum_{q=1}^{R} w_q(u_m) \right)^2}. \qquad \text{(A.6)}$$

Combining (A.5) and (A.6) we get

$$\sum_{r=1}^{R} f_r(u_m) \frac{\partial v_r(u_m)}{\partial \mu_d} = \frac{1}{\sum_{r=1}^{R} w_r(u_m)} \cdot \left(\sum_{r \in J_d} f_r(u_m) \frac{\partial w_r(u_m)}{\partial \mu_d} - \hat{y}(u_m) \sum_{r \in J_d} \frac{\partial w_r(u_m)}{\partial \mu_d} \right). \qquad \text{(A.7)}$$

Furthermore, from (7) we get for all $r = 1,\dots,R$ and all $d = 1,\dots,D$:

$$\frac{\partial w_r(u_m)}{\partial \mu_d} = \prod_{\substack{i \in I_r \\ i \neq d}} F_i(u_{[i]m}) \cdot \frac{\partial F_d(u_{[d]m})}{\partial \mu_d} = \frac{w_r(u_m)}{F_d(u_{[d]m})} \cdot \frac{\partial F_d(u_{[d]m})}{\partial \mu_d} \qquad \text{(A.8)}$$

With

$$\frac{\partial F_d(u_{[d]m})}{\partial \mu_d} = \begin{cases} -\sigma_d & u_l < u_{[d]m} < u_r \\ 0 & \text{else} \end{cases} \qquad \text{(A.9)}$$

and

$$\frac{\partial F_d(u_{[d]m})}{\partial \sigma_d} = \begin{cases} u_{[d]m} - \mu_d & u_l < u_{[d]m} < u_r \\ 0 & \text{else} \end{cases} \tag{A.10}$$

with the limits

$$u_l = \mu - \frac{0,5}{|\sigma|} \qquad \text{and} \qquad u_r = \mu + \frac{0,5}{|\sigma|} \tag{A.11}$$

we finally obtain for all μ_d, $d = 1, \ldots, D$:

$$\frac{\partial E_2}{\partial \mu_d} = \sum_{m=1}^{M} \frac{\varepsilon(u_m)}{\sum_{r=1}^{R} w_r(u_m)} \cdot \left(\hat{y}(u_m) \sum_{r \in J_d} w_r(u_m) - \sum_{r \in J_d} f_r(u_m) w_r(u_m) \right) \cdot \frac{-\sigma_d}{F_d(u_{[d]m})}, \tag{A.12}$$

and correspondingly for all σ_d, $d = 1, \ldots, D$:

$$\frac{\partial E_2}{\partial \sigma_d} = \sum_{m=1}^{M} \frac{\varepsilon(u_m)}{\sum_{r=1}^{R} w_r(u_m)} \cdot \left(\hat{y}(u_m) \sum_{r \in J_d} w_r(u_m) - \sum_{r \in J_d} f_r(u_m) w_r(u_m) \right) \cdot \frac{u_{[d]m} - \mu_d}{F_d(u_{[d]m})}. \tag{A.13}$$

Appendix B. REFERENCES

Ballé, P., O. Nelles and D. Füssel (1997). Fault detection for nonlinear processes based on local linear fuzzy models in parallel and series-parallel mode. In: *Proceedings of 4th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS'97)*.

Basseville, Michèle (1986). On-Line Detection of Jumps in Means. In: *Detection of Changes in Signals and Dynamical Systems*. pp. 11–26. Number 77 In: *Lecture Notes in Control and Information Science*. Springer Verlag.

Braun, H. and M. Riedmiller (1993). Rprop: A fast and robust backpropagation learning strategy. In: *Proceedings of the ACNN*.

Füssel, D., P. Ballé and R. Isermann (1997). Closed loop fault diagnosis based on a nonlinear process model and automatic fuzzy rule generation. In: *Proceedings of 4th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS'97)*.

Hagan, M. and M. Menhaj (1994). Training feedforward networks with the Marquardt algorithm. *IEEE Transactions on Neural Networks* **5**(6), 989–993.

Jang, J.-S. R. (1993). ANFIS: Adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man, and Cybernetics* **23**(3), 665–684.

Jang, J.-S. R. and C. Sun (1995). Neuro-fuzzy modeling and control. In: *Proceedings of the IEEE*. Vol. 83(3). pp. 378–405.

Johansen, T. and B. Foss (1995). Identification of non-linear system structure and parameters using regime decomposition. *Automatica* **31**(2), 321–326.

Klema, V. and A. Laub (1980). The singular value decomposition: Its computation and some applications. *IEEE Transactions on Automatic Control* **AC-25**(2), 164–176.

Lin, C.-T., C.-F. Juang and C.-P. Li (1999). Temperature control with a neural fuzzy inference network. *IEEE Transactions on Systems, Man, and Cybernetics* **29**(3), 440–451.

Männle, M. (1999). Identifying Rule-Based TSK Fuzzy Models. In: *Proceedings of Seventh European Congress on Intelligent Techniques and Soft Computing (EUFIT'99)*.

Männle, M.; Richard, A.; Dörsam T. (1996). Identification of Rule-Based Fuzzy Models Using the RPROP Optimization Technique. In: *Proceedings of Fourth European Congress on Intelligent Techniques and Soft Computing (EUFIT'96)*. pp. 587–591.

Nelles, O., A. Fink, R. Babuška and M. Setnes (1999). Comparison of two construction algorithms for Takagi-Sugeno fuzzy models. In: *Proceedings of Seventh European Congress on Intelligent Techniques and Soft Computing (EUFIT'99)*.

Nelles, Oliver (1999). Nonlinear System Identification with Local Linear Fuzzy Models. Dissertation. Techn. Univ. Darmstadt, Shaker Verlag, ISBN 3-9265-4880-9.

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

Sugeno, M. and G. Kang (1986). Fuzzy modeling and control of multilayer incinerator. *Fuzzy Sets and Systems* **18**, 329–346.

Sugeno, M. and G. Kang (1988). Structure identification of fuzzy model. *Fuzzy Sets and Systems* **26**(1), 15–33.

Sugeno, M. and K. Tanaka (1991). Successive identification of a fuzzy model and its application to prediction of a complex system. *Fuzzy Sets and Systems* **42**, 315–334.

Sugeno, M. and T. Yasukawa (1993). A fuzzy-logic-based approach to qualitative modeling. *IEEE Transactions on Fuzzy Systems* **1**(1), 7–31.

Takagi, T. and M. Sugeno (1985). Fuzzy identification of systems and its application to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics* **15**(1), 116–132.

Yen, J., L. Wang and Gillespie C. W. (1998). Improving the interpretability of TSK fuzzy models by combining global learning and local learning. *IEEE Transactions on Fuzzy Systems* **6**(4), 530–537.

Zell, A., N. Mache, T. Sommer et al. (1994). *Stuttgart Neural Network Simulator, Users Manual, Version 3.2*. University of Stuttgart.