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 nonlinear 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 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.
with the parameters \( \mu \) 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 \( I \) contains the indices of all fuzzy sets that appear in rule \( \mathcal{R} \).

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 \( F_d; d = 1, \ldots, D \)

\[
J_d := \{ j : d \in I_j; j = 1, \ldots, R \},
\]

contains the indices of all rules that have \( F_d \) 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 \( \mathcal{R} \) has for the empty premise \( I = \emptyset \) the general form

\[
\text{if TRUE then } f_r = c_{0_r} + c_{1_r} \cdot u_1 + \ldots + c_{N_r} \cdot u_N
\]

and for \( I \neq \emptyset \) the form

\[
\text{if } u_{i_1} \text{ and } \ldots \text{ and } u_{i_r} \text{ is } F_{i_r} \text{ then } f_r = c_{0_r} + c_{1_r} \cdot u_1 + \ldots + c_{N_r} \cdot u_N
\]

where \( f_r \) denotes the consequence of rule \( \mathcal{R} \).

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

\[
\mathcal{M} := \{ \mathcal{R}_1, \ldots, \mathcal{R}_R \}.
\]

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

\[
w_r(u_m) := \bigwedge_{i \in I_r} F_{i_r}(u_{i_m})
\]

and by choosing the product as t-norm we obtain

\[
w_r(u_m) = \prod_{i \in I_r} F_{i_r}(u_{i_m}).
\]

The normalized membership \( v_r(u) \) be

\[
v_r(u) := \frac{w_r(u)}{\sum_{k=1}^{R} w_k(u)}.
\]

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

\[
\hat{y}(u) = \sum_{k=1}^{R} v_k(u) \cdot f_k(u) = \frac{\sum_{k=1}^{R} \left( w_k(u) \cdot f_k(u) \right)}{\sum_{k=1}^{R} w_k(u)}.
\]

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 \( u = (u_1, \ldots, u_N) \in U_1 \times \cdots \times U_N \subset \mathbb{R}^N \) to an output value \( \hat{y} \in \mathbb{Y} \subset \mathbb{R} \).

2.1 The Fuzzy Model

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

\[
F(u) := \max(1; \min(0; 0.5 + \sigma(u - \mu)))
\]

with the parameters \( \mu \) and \( \sigma \) to be optimized. The parameter \( \mu \) describes the location and \( \sigma \) 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

\[
y(u) = \sum_{k=1}^{R} v_k(u) \cdot f_k(u) = \frac{\sum_{k=1}^{R} \left( w_k(u) \cdot f_k(u) \right)}{\sum_{k=1}^{R} w_k(u)}.
\]
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

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

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 $\frac{\partial E}{\partial c_{u \ell}}$ for all rules $R$, and the premise parameters $\frac{\partial E}{\partial q_{\ell j}}$ and $\frac{\partial E}{\partial d_{\ell j}}$ 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.

There are different ways to generate 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 chosen considerably smaller.

To apply a Page/Hinkley detector, one must first define the two parameters $\mu_{inc}$ and $\mu_{dec}$ of tolerable deviations (of residual increase and decrease). Deviations of and greater than $\mu$ will be detected. The sensitivity of the detector is adjusted by choosing the parameter $\lambda$. A bigger $\lambda$ 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.](image-url)
The process can be described by the following equations:

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

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

\[
h(t) = h(0) + \frac{1}{A} \int_{0}^{t} (q_{\text{in}}(\tau) - q_{\text{out}}(\tau)) d\tau.
\]

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

### 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 \(Q_{\text{in}}\) 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 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
high dimensional data sets automatically build TSK-models based on large and We developed the identification procedure FTSM to exhaustive search strategies show only marginally better number of rules (bottom up approach). Even more ex-

haustive search strategies show only marginally better 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.

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.

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.

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

\[ \frac{\partial E_2}{\partial \mu_d} = - \sum_{m=1}^{M} \left( \varepsilon(m) \cdot \sum_{r=1}^{R} f_r(u_m) \cdot \frac{\partial \gamma_r(u_m)}{\partial \mu_d} \right) \] (A.4)

Derivating (8) for all examples u_m, m = 1, . . . , M, all rules R_r, r = 1, . . . , R, and all fuzzy sets F_d, d = 1, . . . , D yields for r \notin J_d

\[ \frac{\partial \gamma_r(u_m)}{\partial \mu_d} = - \frac{w_r(u_m)}{\left( \sum_{q=1}^{R} w_q(u_m) \right)^2} \sum_{j \in J_d} \frac{\partial \gamma_j(u_m)}{\partial \mu_d} \] (A.5)

and for r \in J_d

\[ \frac{\partial \gamma_r(u_m)}{\partial \mu_d} = \left( \sum_{q=1}^{R} w_q(u_m) \right) \frac{\partial w_r(u_m)}{\partial \mu_d} - \frac{w_r(u_m)}{\left( \sum_{q=1}^{R} w_q(u_m) \right)^2} \sum_{j \in J_d} \frac{\partial \gamma_j(u_m)}{\partial \mu_d} \] (A.6)

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

\[ \sum_{r=1}^{R} f_r(u_m) \frac{\partial \gamma_r(u_m)}{\partial \mu_d} = \frac{1}{\sum_{r=1}^{R} w_r(u_m)} \left( \sum_{r \in J_d} \frac{\partial \gamma_r(u_m)}{\partial \mu_d} \right) \] (A.7)

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

\[ \frac{\partial \gamma_r(u_m)}{\partial \mu_d} = \prod_{r \neq d} F_i(u_{d|m}) \cdot \frac{\partial F_r(u_{d|m})}{\partial \mu_d} \] (A.8)

With

\[ \frac{\partial F_r(u_{d|m})}{\partial \mu_d} = \begin{cases} -\sigma_d & u_l < u_{d|m} < u_r \\ 0 & \text{else} \end{cases} \] (A.9)
with the limits

\[ u_l = \mu - 0.5 \left\lvert \frac{d}{\sigma d} \right\rvert \quad \text{and} \quad u_r = \mu + 0.5 \left\lvert \frac{d}{\sigma d} \right\rvert \quad (A.11) \]

we finally obtain for all \( \mu_d, d = 1, \ldots, D \):

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

and correspondingly for all \( \sigma_d, d = 1, \ldots, D \):

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

Appendix B. REFERENCES


