

Schnelle Takagi-Sugeno Fuzzy Modellierung

Dipl.-Inform. Manfred Männle

Institut für Rechnerentwurf und Fehlertoleranz

Universität Karlsruhe, D-76128 Karlsruhe

Tel. +49/721/608-6323, Fax +49/721/608-3962

Manfred.Maennle@informatik.uni-karlsruhe.de

Zusammenfassung — **Unscharfe (Fuzzy) Takagi-Sugeno Modelle sind bei der modellbasierten Regelung und Fehlerdiagnose weit verbreitet. Sie verbinden die Eigenschaften einer hohen Genauigkeit und einer relativ einfachen Interpretierbarkeit. Die Allgemeinheit der Takagi-Sugeno Modelle macht ihre datengestützte Identifikation jedoch sehr rechenaufwendig. Wir wenden deshalb RPROP (Resilient Propagation) an, ein leistungsfähiges Verfahren zur nichtlinearen Optimierung, das ursprünglich zum Lernen Neuronaler Netze entwickelt wurde. Damit erreichen wir eine schnelle und robuste Parameteroptimierung und sind nun in der Lage, auch ohne Einschränkungen (zum Beispiel bei der Wahl der Modellstruktur) Identifikationsprobleme hoher Dimension mit großen Datenmengen anzugehen.**

1 Einleitung

Takagi-Sugeno Modelle (TS-Modelle, auch TSK-Modelle, nach Takagi, Sugeno und Kang) [14, 10, 11, 12, 13] erfreuen sich hoher Beliebtheit, was die hohe Anzahl an

Veröffentlichungen und Anwendungen zeigt. Das liegt einerseits an deren Eigenschaft, als nichtlinearer Approximator jede kontinuierliche Abbildung beliebig genau annähern zu können [15, 16], und andererseits, als stückweise lineares Modell relativ einfach interpretierbar zu bleiben.

Ein Problem ist jedoch, daß die Allgemeinheit der TS-Modelle die *datengestützte Identifikation* sehr rechenaufwendig macht. Ein Unscharfes Modell besteht aus mehreren Regeln, wobei sich jede Regel aus einem Prämissen- und einem Konsequenzteil zusammensetzt. Die Prämisse enthält die Eingangsvariablen und definiert über eine Konjunktion von Unscharfen Klauseln den Gültigkeitsbereich der Regel im Ein-/Ausgaberaum. Die Konsequenz enthält ein lineares Regressionsmodell.

Die Identifikationsaufgabe läßt sich in mehrere Teile aufspalten: *Strukturidentifikation*, das heißt Bestimmung der Anzahl Regeln und der Eingangsvariablen in den Prämissen, und *Parameteroptimierung* der Prämissen- und Konsequenzparameter. Diese drei Aufgaben können getrennt gelöst werden. Fixiert man die Struktur und die Prämissenparameter, so ergibt sich für die Konsequenzparameter ein lineares Optimierungs-

problem, das mit Kleinste Quadrate Methoden, wie beispielsweise Singular Value Decomposition (SVD) [4, 6], gelöst werden kann. Die Optimierung der Prämissenparameter bleibt in jedem Fall ein nichtlineares Optimierungsproblem. Die Strukturbestimmung schließlich beinhaltet, falls sie ohne Einschränkung gelöst wird, eine kombinatorische Suche.

Um diese Komplexität zu bewältigen, vereinfachen die meisten TS-Identifikationsalgorithmen die Modellstruktur und wenden teilweise Heuristiken für die Bestimmung der Prämissenparameter an: In [7] werden die Prämissenparameter nicht direkt optimiert, sondern zusammen mit der Strukturidentifikation durch eine heuristische Suche (LOLIMOT) oder durch ein Clustering des Ein-/Ausgaberaums (Product Space Clustering) bestimmt. Beim ANFIS-Verfahren [2, 3] wird die Modellstruktur statisch vorgegeben (äquidistante Aufteilung der Eingaberäume) und die Modellparameter mittels Backpropagation und Kleinste Quadrate Optimierung bestimmt. Der NFIN Algorithmus führt in einem ersten Schritt ein Clustering des Eingaberaums durch und optimiert die Regelparameter anschließend mittels Backpropagation [5].

Seit der ersten Veröffentlichung über TS-Modelle im Jahr 1985 gab es große Fortschritte bei den nichtlinearen Optimierungsverfahren. Die *Idee* des in diesem Beitrag vorgestellten Ansatzes besteht in der *Anwendung einer dieser neuen, leistungsfähigen Algorithmen auf das Problem der nichtlinearen Parameteroptimierung bei TS-Modellen*. Hierfür fiel die Wahl auf

RPROP [9, 1, 17]. Die Leistungsfähigkeit von RPROP ist mit der Effizienz von Methoden zweiter Ordnung vergleichbar. Außerdem ist das Verfahren robust, d. h. es liefert mit seiner Standardeinstellung durchweg gute Ergebnisse.

2 Identifikation von Takagi-Sugeno Modellen

TS-Modelle eignen sich zur Identifikation von statischen oder dynamischen MISO (multiple input single output) Systemen. Das Modell bildet einen N -dimensionalen Eingabevektor $\vec{u} \in \mathcal{U}_1 \times \dots \times \mathcal{U}_N \subset \mathbb{R}^N$ auf einen Ausgabewert $\hat{y} \in \mathcal{Y} \subset \mathbb{R}$ ab.

Das Unscharfe Modell trennt den Eingaberaum in Bereiche, die durch die Unscharfen Mengen abgegrenzt werden. Jede Regel deckt einen Bereich ab. Das Modell enthält $d = 1, \dots, D$ Unscharfe Mengen mit sigmoidförmiger Zugehörigkeitsfunktion

$$F_d(u) := \frac{1}{1 + e^{\sigma_d \cdot (u - \mu_d)}}. \quad (1)$$

Die Indexmenge $J_d \subset \{1, \dots, R\}$ enthalte die die Indizes aller Regeln, welche F_d in ihrer Prämisse enthalten.

Das Unscharfe Modell besteht aus $r = 1, \dots, R$ Regeln der Form

$$\text{if } u_{i_{1r}} \text{ is } F_{1r} \text{ and } \dots \text{ and } u_{i_{nr}} \text{ is } F_{nr} \text{ then} \\ f_r = c_{0r} + \underbrace{c_{1r} \cdot u_1 + \dots + c_{Nr} \cdot u_N}_{\text{optional}}. \quad (2)$$

Die Indexmenge $I_r \subset \{1, \dots, D\}$ enthalte die Indizes aller Mengen, die in der r . Regel auftauchen.

Die Ausgabe des Modells ist damit

$$\hat{y}(\vec{u}) = \frac{\sum_{k=1}^R \left(w_k(\vec{u}) \cdot f_k(\vec{u}) \right)}{\sum_{k=1}^R w_k(\vec{u})} \quad (3)$$

mit den Regelzugehörigkeiten

$$w_r(\vec{u}_m) = \prod_{i \in I_r} F_{ir}(u_{[i]m}), \quad (4)$$

wobei der Index $[d] \in \{1, \dots, N\}$ angibt, in welcher Dimension F_d wirkt.

2.1 Strukturidentifikation

Um Probleme hoher Dimension, d.h., mit vielen Eingangsvariablen lösen zu können, ist ein „bottom up“-Ansatz notwendig, in dem mit einem linearen Modell (Modell mit einer Regel) begonnen wird und dann schrittweise durch Hinzufügen weiterer Regeln und Aufnahme weiterer Variablen in die Prämissen ein mächtigeres Modell gebildet wird. Für diese Strukturidentifikation wird eine heuristische Suche eingesetzt. Dabei werden in jeder Epoche eine Reihe von Modellen gebildet, deren Parameter optimiert werden und das beste dieser Modelle den Ausgangspunkt für die nächste Epoche bestimmt. Für Einzelheiten der Heuristik sei der Leser auf [11, 3, 6] verwiesen.

2.2 Parameteridentifikation

Grundlage der Parameteroptimierung ist die Minimierung des quadratischen Fehlers der M Trainingsvektoren (\vec{u}_m, y_m) ,

$$E_2 := \frac{1}{2} \sum_{m=1}^M \varepsilon^2(\vec{u}_m), \quad \varepsilon(\vec{u}_m) = y_m - \hat{y}(\vec{u}_m). \quad (5)$$

Hierzu setzen wir das Gradientenabstiegsverfahren RPROP [9, 1, 17] ein. Die dazu notwendigen partiellen Ableitungen des Fehlers nach den Parametern ergeben sich für $r = 1, \dots, R$ und $i = 0, \dots, C$ zu

$$\frac{\partial E_2}{\partial c_{ir}} = \sum_{m=1}^M \frac{-\varepsilon(\vec{u}_m)}{\sum_{k=1}^R w_k(\vec{u}_m)} \cdot w_r(\vec{u}_m) \cdot u_{im} \quad (6)$$

und für $d = 1, \dots, D$ zu

$$\begin{aligned} \frac{\partial E_2}{\partial \mu_d} = \sum_{m=1}^M \frac{\varepsilon(\vec{u}_m)}{\sum_{r=1}^R w_r(\vec{u}_m)} \cdot \left(\hat{y}(\vec{u}_m) \sum_{r \in J_d} w_r(\vec{u}_m) \right. \\ \left. - \sum_{r \in J_d} f_r(\vec{u}_m) w_r(\vec{u}_m) \right) \\ \cdot \left(1 - F_d(u_{[d]m}) \right) \cdot \sigma_d \end{aligned} \quad (7)$$

und

$$\begin{aligned} \frac{\partial E_2}{\partial \sigma_d} = \sum_{m=1}^M \frac{\varepsilon(\vec{u}_m)}{\sum_{r=1}^R w_r(\vec{u}_m)} \cdot \left(\hat{y}(\vec{u}_m) \sum_{r \in J_d} w_r(\vec{u}_m) \right. \\ \left. - \sum_{r \in J_d} f_r(\vec{u}_m) w_r(\vec{u}_m) \right) \\ \cdot \left(1 - F_d(u_{[d]m}) \right) \cdot (\mu_d - u_{[d]m}). \end{aligned} \quad (8)$$

2.3 Komplexität

Aus Gleichung (3) ist die Komplexität eines Durchlaufs ersichtlich: Bei einer Dimension des Eingaberaumes von N , M Wertevektoren (patterns) und einer Anzahl R von Unschärfe Regeln des Modells beträgt der Aufwand für einen einfachen Simulations- oder Prädiktionsschritt (feedforward step) aller Werte $O(RNM)$.

Mit Hilfe von Vorberechnungen und Speicherung von Zwischenergebnissen bei der Berechnung der Parameteranpassung, z. B. der

R	2	3	4	5	6	7
Prädiktionsschritt	280	440	650	780	960	1130
RPROP-Iteration	870	1520	2490	3550	4530	5610

Tabelle 1: Rechenzeit für eine Iteration in μs auf Sun Ultra 5.

f_r und w_r in den Gleichungen (7) und (8), ergibt sich für eine RPROP-Iteration dieselbe Komplexität von $O(RNM)$! Bei einer Anzahl von I notwendigen RPROP-Iterationen erhält man somit als Aufwand für die Parameteroptimierung eines einzelnen Modells $O(RNMI)$.

Tabelle 1 gibt gemessene Rechenzeiten (für einen Prädiktionsdurchlauf sowie eine RPROP-Iteration) für eine Beispielmodellierung mit $N = 10$ und $M = 148$ für verschiedene R wieder. Die gemessenen Zeiten bestätigen die theoretischen Komplexitätsbetrachtungen.

Bei der Strukturidentifikation durch die heuristische Suche nach Kang werden $\sum_{r=1}^R N(r-1) = N \frac{(R-1)R}{2}$ Modelle gebildet und evaluiert, so daß sich für den Modellierungsalgorithmus ein Gesamtaufwand von $O(R^3 N^2 MI)$ ergibt. Durch eine Beschränkung der Suche auf jeweils die Regel mit dem größten Approximationsfehler erhält man in der Praxis geringfügig schlechtere Ergebnisse, nun jedoch mit dem geringeren Aufwand von $O(R^2 N^2 MI)$.

Nach bisheriger Erfahrung ist die Anzahl benötigter RPROP-Iterationen unabhängig von der Anzahl der Parameter und Trainingsdaten. Auch blieb dabei die Anzahl R an notwendigen Regeln unter 10 und die Anzahl I der RPROP-Iterationen läßt sich auf zum Beispiel 500 begrenzen (Mittelwert ist ca. 100), so daß der Rechenaufwand im we-

sentlichen quadratisch von der Eingangsdimension und linear von der Anzahl der Meßwerte abhängt.

3 Anwendungsbeispiel: Ein Turbolader

Als Anwendungsbeispiel wählten wir das Beispiel eines Turboladers aus [8, 7]. Dieser dynamische Prozeß hat die zwei Eingänge m_b (Einspritzrate) und n_e (Motordrehzahl) und einen Ausgang p_2 (Ladedruck des Turboladers). Die Abtastrate beträgt 5 Hz.

Die Identifikation des Prozesses wurde durch die Verwendung des folgenden Regressionsvektors erreicht (NOE-Modell):

$$\varphi(m_b^t, m_b^{t-1}, m_b^{t-2}, e_n^t, e_n^{t-1}, e_n^{t-2}, \hat{p}_2^{t-1}, \hat{p}_2^{t-2}).$$

Die Simulation des identifizierten Modells der Trainingsdaten ist in Abbildung 1 gegeben. Abbildung 2 zeigt die Simulation einer Stadtfahrt. Für ein Modell mit 5 Regeln beträgt der RMS Ausgabefehler für die Trainingsdaten 0,025 und für die Stadtfahrt 0,038. Das Modell mit 9 Regeln erzielte RMS-Werte von 0,023 und 0,030. Die Genauigkeit ist damit vergleichbar mit den Ergebnissen in [7]. Durch eine weitere Parameteroptimierung des NOE-Modells konnte der Fehler für die Trainingsdaten bis auf 0,013 gesenkt werden. Gleichzeitig stieg der Fehler aber bei den Validierungsdaten an, d. h. es trat ein Overlearning-Effekt ein.

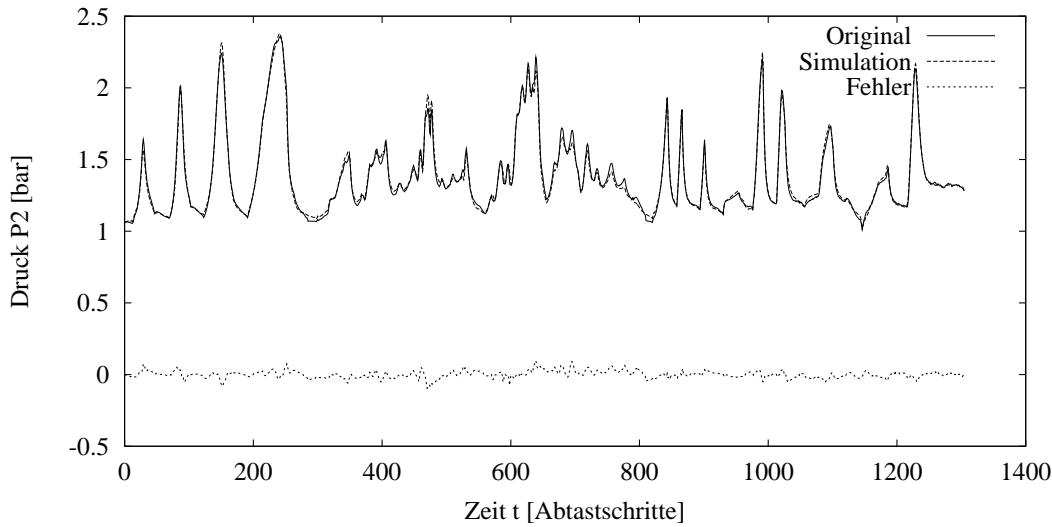


Abbildung 1: Simulation des Turboladers (Lerndaten), $R = 5$.

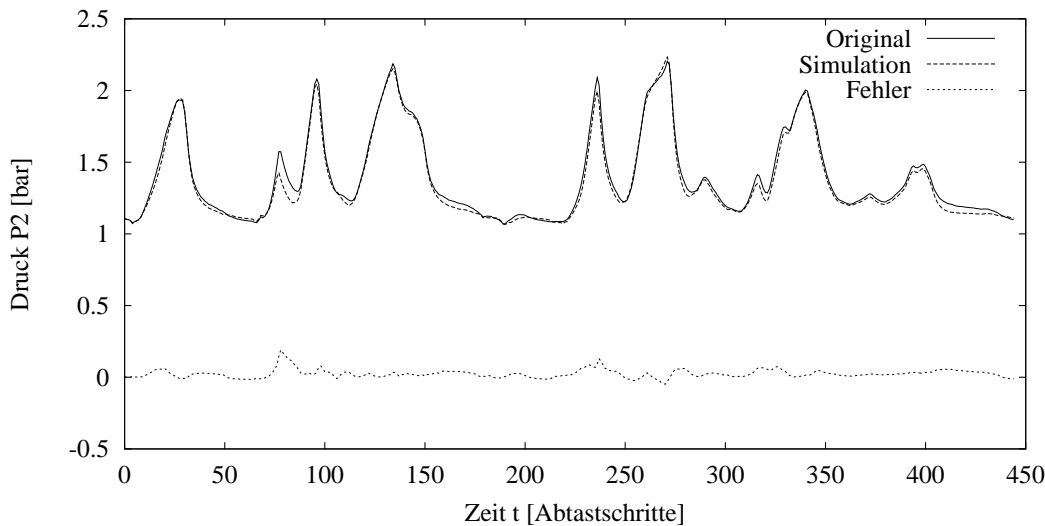


Abbildung 2: Simulation des Turboladers (Stadtfahrt), $R = 5$.

4 Schluß

Die Identifikation von Problemen hoher Dimension ist hier durch einen „bottom up“-Ansatz gelöst, der mit einem einfachen Modell beginnt und dann schrittweise dessen Mächtigkeit erhöht (Heuristik von Kang). Diese heuristische Suche macht die Optimierung einer großen Zahl von Modellen notwendig. Damit die Suche praktikabel

wird, benötigt man eine schnelle und effiziente Parameteroptimierung. In dieser Arbeit wird dies durch die Anwendung von RPROP erreicht.

Die Komplexität einer RPROP-Iteration ist gleich der eines Modelldurchlaufs zur Prädiktion, das heißt, der Rechenaufwand der Parameteroptimierung wächst auch bei großen Problemen nicht überproportional an.

Dank

Unser Dank gilt Herrn Alexander Fink und Frau Susanne Ernst vom IAT, Techn. Univ. Darmstadt, für die Beispieldaten des Turboladers und für interessante Diskussionen.

Literatur

- [1] H. Braun. *Neuronale Netze – Optimierung durch Lernen und Evolution*. Springer, 1997.
- [2] J.-S. R. Jang. ANFIS: Adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(3):665–684, 1993.
- [3] J.-S. R. Jang and C. Sun. Neuro-fuzzy modeling and control. In *Proceedings of the IEEE*, volume 83(3), pages 378–405, 1995.
- [4] V. Klema and A. Laub. The singular value decomposition: Its computation and some applications. *IEEE Transactions on Automatic Control*, AC-25(2):164–176, 1980.
- [5] C.-T. Lin, C.-F. Juang, and C.-P. Li. Temperature control with a neural fuzzy inference network. *IEEE Transactions on Systems, Man, and Cybernetics*, 29(3):440–451, 1999.
- [6] M. Männle. Identifying Rule-Based TSK Fuzzy Models. In *Proceedings of Seventh European Congress on Intelligent Techniques and Soft Computing (EUFIT'99)*, 14 pages, 1999.
- [7] O. Nelles, A. Fink, R. Babuška, and M. Setnes. Comparison of two construction algorithms for Takagi-Sugeno fuzzy models. In *Proceedings of Seventh European Congress on Intelligent Techniques and Soft Computing (EUFIT'99)*, 1999.
- [8] Oliver Nelles. *Nonlinear System Identification with Local Linear Fuzzy Models*. Dissertation, Techn. Univ. Darmstadt, Shaker Verlag, ISBN 3-9265-4880-9, 1999.
- [9] 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)*, pages 586–591, 1993.
- [10] M. Sugeno and G. Kang. Fuzzy modeling and control of multilayer incinerator. *Fuzzy Sets and Systems*, 18:329–346, 1986.
- [11] M. Sugeno and G. Kang. Structure identification of fuzzy model. *Fuzzy Sets and Systems*, 26(1):15–33, 1988.
- [12] M. Sugeno and K. Tanaka. Successive identification of a fuzzy model and its application to prediction of a complex system. *Fuzzy Sets and Systems*, 42:315–334, 1991.
- [13] M. Sugeno and T. Yasukawa. A fuzzy-logic-based approach to qualitative modeling. *IEEE Transactions on Fuzzy Systems*, 1(1):7–31, 1993.
- [14] T. Takagi and M. Sugeno. Fuzzy identification of systems and its application to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics*, 15(1):116–132, 1985.
- [15] 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*, 28(4):515–520, 1998.
- [16] 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*, 29(5):508–514, 1999.
- [17] A. Zell, N. Mache, T. Sommer, et al. *Stuttgart Neural Network Simulator, Users Manual, Version 3.2*. University of Stuttgart, June 1994.