Comparison of Two Construction Algorithms for Takagi-Sugeno Fuzzy Models

Oliver Nelles, Alexander Fink, Robert Babuška*, and Magne Setnes*

Darmstadt University of Technology, Institute of Automatic Control,
Laboratory of Control Systems and Process Automation,
Landgraf-Georg-Str. 4, D-64283 Darmstadt,Germany,
E-mail: {ONelles, AFink}@iat.tu-darmstadt.de
* Delft University of Technology, Faculty of Information Technology and Systems,
Control Laboratory, PO Box 5031, 2600 GA Delft, The Netherlands
E-mail: {r.babuska, m.setnes}@its.tudelft.nl

ABSTRACT: This paper compares two different approaches for the construction of Takagi-Sugeno fuzzy models from data. The local linear model tree (LOLIMOT) algorithm incrementally generates the fuzzy model by an axis-orthogonal decomposition of the input space. The product space clustering approach uses the Gustafson-Kessel algorithm for input space decomposition. The fundamental advantages and drawbacks of both alternative strategies are pointed out. Their properties and real-world applicability are illustrated by building a dynamic model of a truck Diesel engine turbocharger.

KEYWORDS: modeling, identification, Takagi-Sugeno fuzzy models, local linear models, turbocharger

1 Introduction

Nonlinear static and dynamic models are necessary, for instance, in prediction, simulation, model-based control, and fault diagnosis. In most cases, the derivation of such models by first principles (physical, chemical, biological, etc. laws) is expensive, time-consuming and involves many unknown parameters and heuristics. Hence, methods for data-driven modelling and identification are of great interest. A wide class of nonlinear dynamic processes with p inputs u_i and one output y can be described in the discrete time domain by, see [10]

$$y(k) = f\left(\boldsymbol{\xi}(k)\right) \text{ with } \boldsymbol{\xi}(k) = [u_1(k-1)\ \dots\ u_1(k-m_1)\ \dots\ u_p(k-1)\ \dots\ u_p(k-m_p)\ y(k-1)\ \dots\ y(k-n)]^T \quad (1)$$

with the discrete time k and the dynamic orders m_i, n. The unknown function $f(\cdot)$ in (1) can be approximated from measurement data by Takagi-Sugeno fuzzy models. They are briefly introduced in the subsequent section. Sections 3 and 4 discuss the two alternative strategies for the identification of Takagi-Sugeno fuzzy models and point out their fundamental properties. Finally, as an application example the modeling and identification of a truck Diesel engine turbocharger is demonstrated with both methods.

2 Takagi-Sugeno Fuzzy Models

Throughout this contribution, the unknown function $f(\cdot)$ in (1) is approximated by Takagi-Sugeno type fuzzy models [13]. The rule base comprises M rules of the form

$$R_j: \text{ IF } z_1 \text{ is } A_{j,1} \text{ AND } \dots \text{ AND } z_{nz} \text{ is } A_{j,nz} \text{ THEN } y(k) = w_{j,0} + w_{j,1}x_1 + \dots + w_{j,nx}x_{nx}\,, \quad (2)$$

$j = 1 \dots M$, where $A_{j,i}$ is a fuzzy set defined on the universe of discourse of input i. Both the nz-dimensional vector $\mathbf{z}(k) = [z_1\ z_2\ \dots\ z_{nz}]^T$ in the rule premise and the nx-dimensional vector $\mathbf{x}(k) = [x_1\ x_2\ \dots\ x_{nx}]^T$ in the consequent contain subsets of the elements of $\boldsymbol{\xi}(k)$. The rule consequents represent linear difference equations which are linear in the parameters $w_{j,i}$. The additional constants $w_{j,0}$ define the operating points. This type of

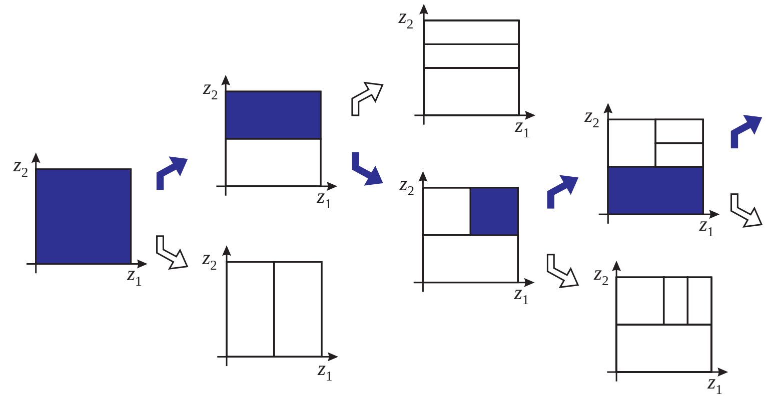

Figure 1: Tree construction and input space decomposition by LOLIMOT.

fuzzy model is a universal approximator of the function $f(\cdot)$ under the condition that the premise input space includes all inputs $\mathbf{z}(k) = \boldsymbol{\xi}(k)$.
The output of the Takagi-Sugeno fuzzy system with M rules is aggregated as

$$y(k) = \sum_{j=1}^{M} (w_{j,0} + w_{j,1}x_1 + \ldots + w_{j,nx}x_{nx})\, \Phi_j(\mathbf{z}) \tag{3}$$

with the validity functions Φ_j. These validity functions are normalized such that $\sum_{j=1}^{M} \Phi_j(\mathbf{z}) = 1$ for all premise inputs $\mathbf{z}$. This normalization is achieved by $\Phi_j(\mathbf{z}) = \mu_j(\mathbf{z}) / \sum_{i=1}^{M} \mu_i(\mathbf{z})$ where $\mu_j(\mathbf{z})$ represent the multi-dimensional membership functions of the fuzzy model. [11] refers to this type of model as local model network which interpolates local linear models by overlapping local basis functions.
The task of fuzzy model construction is to determine both the nonlinear parameters of the membership functions and the linear parameters of the local models [7]. In general, there are two ways to obtain this information. Possibly, human experts are able to formulate their process knowledge in fuzzy rules. Unfortunately, this usually delivers only a rough idea of the plant behavior, because the human being cannot sense all the details and might not be able to quantitatively express the observations. Therefore, numerous approaches have been proposed [2] which compute nonlinear dynamic fuzzy models from input/output measurement data. This paper compares a tree construction and a product space clustering strategy. In contrast, to the brute force nonlinear optimization approach proposed in [6] these strategies are much less computationally demanding.

3 Local Linear Model Tree (LOLIMOT)

The local linear model tree (LOLIMOT) algorithm proposed in [12] utilizes Gaussian membership functions

$$\mu_j(\mathbf{z}) = \exp\left(-\frac{1}{2}\frac{(z_1 - c_{j,1})^2}{\sigma_{j,1}^2}\right) \cdot \exp\left(-\frac{1}{2}\frac{(z_2 - c_{j,2})^2}{\sigma_{j,2}^2}\right) \cdot \cdots \cdot \exp\left(-\frac{1}{2}\frac{(z_{nz} - c_{j,nz})^2}{\sigma_{j,nz}^2}\right) \tag{4}$$

where $c_{j,l}$ denote the centers and $\sigma_{j,l}$ denote the standard deviations in dimension l for the membership function associated with rule j.
LOLIMOT splits up the identification procedure into two parts. In an outer loop of the algorithm, the premise structure and the corresponding membership functions, respectively, are determined by a tree construction algorithm. The input space is decomposed into hyper-rectangles by axis-orthogonal splits. Each local linear model belongs to one hyper-rectangle in whose center the basis function is placed. The standard deviations are chosen proportional to the size of the hyper-rectangle. This makes the size of the validity region of each local linear model proportional to its hyper-rectangle extension. At each iteration, the local linear model with the worst local error measure is subdivided by splitting it into two halves. Splits in all dimensions are tested and the one with the highest performance improvement is chosen. For an illustration of the first four iterations of LOLIMOT refer to Fig. 1.
In an inner loop, the linear regressors and parameters of the local models are selected and estimated by an orthogonal weighted least-squares (OLS) algorithm. The LOLIMOT algorithm gains its high performance

from local estimation of the linear parameters in the rule consequents. When the premise structure has been determined, M linear optimization problems are solved separately. The non-weighted output of the j-th rule can be written as

$$y_j = \psi_j^T \mathbf{w}_j \tag{5}$$

with parameter and regression vectors

$$\mathbf{w}_j = [w_{j,0}\ w_{j,1}\ \dots\ w_{j,nx}]^T, \quad \psi_j = [\psi_1\ \psi_2\ \dots\ \psi_{nx+1}]^T = [1\ x_1\ \dots\ x_{nx}]^T. \tag{6}$$

The optimal parameters $\mathbf{w}_j$ of each local linear model j are estimated from N data samples by local weighted least squares

$$\mathbf{w}_j = \left(\mathbf{\Psi}_j^T \mathbf{Q}_j \mathbf{\Psi}_j\right)^{-1} \mathbf{\Psi}_j^T \mathbf{Q}_j \mathbf{y}_\mathrm{d} \tag{7}$$

where $\mathbf{y}_\mathrm{d} = [y_\mathrm{d}(1)\ y_\mathrm{d}(2)\ \dots\ y_\mathrm{d}(N)]^T$ is the vector of desired model outputs and $\mathbf{\Psi}_j = [\psi_j(1)\ \psi_j(2)\ \dots\ \psi_j(N)]^T$ is the regression matrix. The weighting matrix $\mathbf{Q}_j = \mathrm{diag}\{\Phi_j(z(1)),\ \Phi_j(z(2)),\ \dots,\ \Phi_j(z(N))\}$ contains the values of the validity function which yields the local property of the estimation. Data samples located close to the center of the respective validity function have a higher influence on the parameter estimates than data points which are far away in the input space. Hence, in contrast to global parameter estimation, it can be guaranteed that the local models represent the local behavior of the nonlinear system and that they do not suffer from compensation effects. One should notice, that the overlap of the local models is neglected by the estimation scheme which might deteriorate the model's accuracy if the standard deviation of the Gaussian membership functions is chosen too large.
The main features of LOLIMOT can be summarized as follows:

1. *Gaussian membership functions* are used.
2. The *parameter estimation* is performed *locally*. This yields a regularization effect that reduces the variance error at the price of an increased bias error. The computational complexity grows only linearly with the number of fuzzy rules.
3. The *input space decomposition* is performed in an *axis-orthogonal* manner. This approach is very simple and allows a computationally effective implementation. However, the restriction to axis-orthogonal splits with the ratio 1:1 leads to suboptimal results and thus LOLIMOT constructs models with relatively many rules.
4. Different *input spaces* for the rule premises and consequents can be used.
5. The computational complexity grows only cubic with the consequent space dimensionality $\dim(\mathbf{x})$ and linear with the premise input space dimensionality $\dim(\mathbf{z})$. This overcomes the *curse of dimensionality.*
6. The axis-orthogonal input space partitioning allows a direct *interpretation* in terms of *fuzzy logic.*
7. It is a *growing* algorithm, i.e., it incrementally increases the number of rules in each iteration.
8. It is not restricted to local *linear* models. Rather any type of local model can be incorporated and the local model structures can be different in one Takagi-Sugeno fuzzy system.

4 Product Space Clustering

Fuzzy clustering is applied to discover fuzzy regions in the data space in which the system can be approximated locally by linear submodels [1]. The regression matrix $\mathbf{\Phi}$ and the output vector $\mathbf{y}$ are first constructed from the available data:

$$\mathbf{\Phi}^T = [\varphi_1, \dots, \varphi_N], \quad \mathbf{y}^T = [y_1, \dots, y_N]. \tag{8}$$

Here, N is the number of samples used for identification. Then, the pattern matrix $\mathbf{P}$ to be clustered is composed by appending $\mathbf{y}$ to $\mathbf{\Phi}$:

$$\mathbf{P}^T = [\mathbf{\Phi}, \mathbf{y}]. \tag{9}$$

Given $\mathbf{P}$ and an *estimated* number of clusters M, a fuzzy clustering algorithm [5] is applied to compute the fuzzy partition matrix $\mathbf{U}$.
The number of clusters M determines the number of rules in the fuzzy model obtained. Two main approaches to finding an appropriate number of clusters can be distinguished:

- Cluster the data for different values of M and then use *validity measures* to assess the goodness of the obtained partitions [4].

- Start with a sufficiently large number of clusters and successively reduce this number by merging clusters that are compatible with respect to some predefined criteria [8, 9].

The fuzzy sets in the premise of the rules are obtained from the partition matrix $\mathbf{U}$. The ikth element $\mu_{ik} \in [0,1]$ of this matrix is the membership degree of the data object $\mathbf{p}_k$ in cluster i. One-dimensional fuzzy sets A_{ij} are obtained from the multidimensional fuzzy sets defined point-wise in the ith row of the partition matrix by projections onto the premise variables. The obtained point-wise fuzzy sets are approximated by suitable parametric functions.
The consequent parameters for each rule can be obtained either by solving the weighted least-square problem (7) for each rule separately or by solving a single "global" least-squares problem.
The transparency of the model obtained in the above way may be hampered by redundancy present in the form of many overlapping (compatible) membership functions (due to the projection). Similarity measures are used in order to asses the compatibility (pair-wise similarity) of fuzzy sets in the rule base, in order to identify sets that can be merged. Fuzzy sets estimated from data can also be similar to the universal set, thus adding no information to the model. Such sets can be removed from the antecedent of the rules. These operations reduce the number of fuzzy sets in the model. Reduction of the rule base follows when the antecedents of some rules become equal. Such rules are combined into one rule. The compatibility between the fuzzy sets A_{lj} and A_{mj} in the rules R_l and R_m, respectively, is assessed by the fuzzy analog of the Jaccard index [3]:

$$c_{jlm} = \frac{|\, A_{lj} \cap A_{mj} \,|}{|\, A_{lj} \cup A_{mj} \,|}, \tag{10}$$

where $l, m = 1, 2, \ldots, L$, and $c_{jlm} \in [0,1]$. The $\cap$ and $\cup$ operators are the intersection and the union, respectively, and $|\cdot|$ denotes the cardinality of a fuzzy set. The measure c_{jlm} is computed for discretized domains.
The main features of the clustering-based approach can be summarized as follows:

1. The decomposition of the antecedent space is done in the product space of the regressors. This is a more general and powerful way than the axis-orthogonal decomposition of LOLIMOT and related methods. The latter techniques give suboptimal results, as a greedy strategy is used for the selection of variables to split and also the position of the split is sought in an suboptimal way.

2. Initially, the same variables must be used for both the antecedents and the consequents. In the second step, however, subsets of these variables may be obtained as a result of the simplification.

3. The computational complexity grows a cubic manner with the sum of the premise and consequent space dimensions.

4. Similarly to LOLIMOT, a regularization effect is achieved that reduces the variance error at the cost of an increased bias error.

5. As clusters obtained in the product space may be of different shapes and orientations, the interpretation of the antecedent may be less accurate than with LOLIMOT.

5 Identification of a Turbocharger

Figure 2a schematically represents the charging process of a Diesel engine by an exhaust turbocharger. The charging process has a nonlinear input/output behavior as well as a strong dependency of the dynamic parameters on the operating point. This is known by physical insights.
In general, the static behavior of the turbocharger may be described sufficiently well by characteristic maps (look-up tables) of compressor and turbine. However, if the dynamics of the turbocharger need to be considered, basic mechanical and thermodynamical modeling is required, see [14]. The quality of theoretical models, however, essentially depends on the accurate knowledge of several process parameters, which have to be laboriously derived or estimated in most cases by analogy considerations. Another disadvantage is the considerable computational effort due to the complexity of those methods.
For these reasons, such methods are considered to be inconsistent with the requirement of typical control engineering applications such as controller design, fault diagnosis and hardware-in-the-loop simulations. Therefore, two alternative strategies for building Takagi-Sugeno fuzzy models are realized in the following which comply

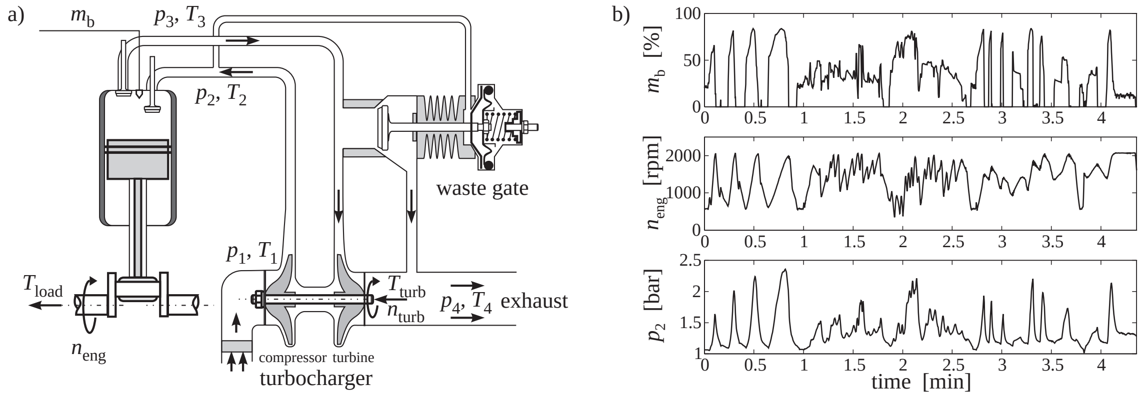

Figure 2: a) Scheme of the combustion engine turbocharger. b) Training data.

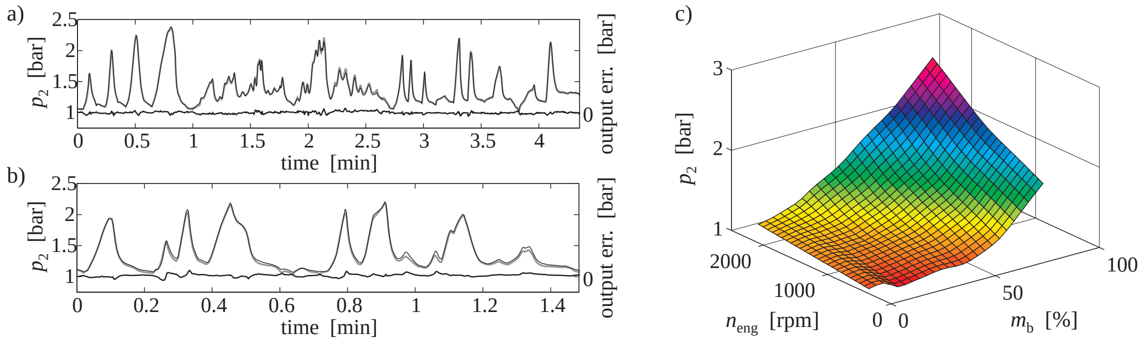

Figure 3: LOLIMOT-trained model performance on a) the training data, b) the urban traffic validation data, and c) the static model behavior.

with the demanded properties to a high degree. Only the recorded data of input/output measurements of the real process is required. Thus, no deeper theoretical knowledge of the process is necessary. For a hardware-in-the-loop simulation, the rate of injection $m_{\rm b}$ and the engine speed $n_{\rm eng}$ are chosen as inputs while the charging pressure p_2 is the output. The sampling time is $T_0 = 0.2s$.

The training data was generated by a special driving cycle to excite the system with all amplitudes and frequencies of interest, see Fig. 2b. The measurements were recorded on a flat test track. Also, in order to operate the engine in high load ranges the truck was driven with the biggest possible load. For validation, special driving cycles were recorded, which reproduce realistic conditions in urban traffic.

It was found by a trial-and-error approach (starting with first order and increasing the order in each trial) that the turbocharger can be described sufficiently well by assuming a second order process. Therefore, the charging pressure $p_2(k)$ at time instant k is modeled by the following relationship

$$p_2(k) \;=\; f\,(m_{\rm b}(k),\, m_{\rm b}(k-1),\, m_{\rm b}(k-2),\; n_{\rm eng}(k),\, n_{\rm eng}(k-1),\, n_{\rm eng}(k-2),\; p_2(k-1),\, p_2(k-2))\;. \quad (11)$$

The convergence behavior of the LOLIMOT algorithm indicates that a fuzzy model with 10 rules is sufficiently accurate and more complex models do not yield further significant improvements. In comparison, the product space clustering approach only requires 3 rules due to its higher flexibility. The computational demand for both approaches is comparable. The RMS output error on the training and both validation data sets is also similar with 0.021, 0.023, 0.032 for the LOLIMOT-trained fuzzy model and 0.024, 0.023, 0.036 with the clustering approach, compare Fig. 3 and 4.

6 Conclusions

Two different methods for the identification of Takagi–Sugeno fuzzy models from data have been presented and compared. The results obtained for the turbocharger indicate that approximately the same predictive accuracy is obtained with both methods and also the computational complexity of the identification methods

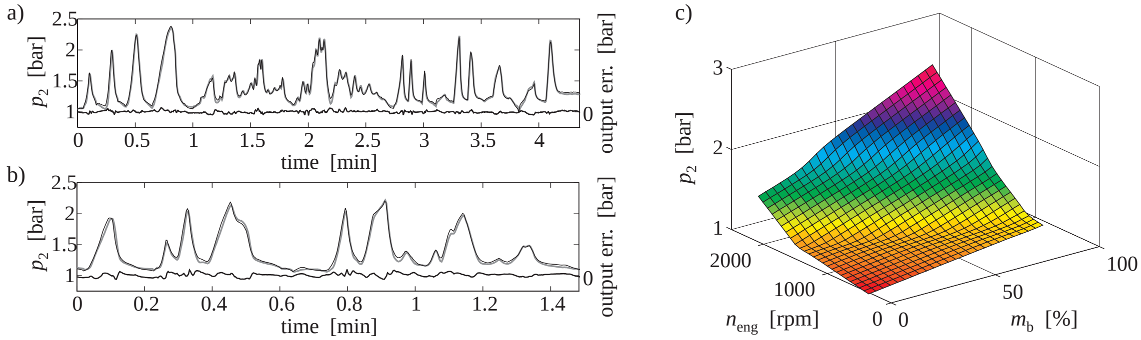

Figure 4: Product space clustering-trained model performance on a) the training data, b) the urban traffic validation data, and c) the static model behavior.

is comparable. For real-time simulations (such as in predictive control), however, the model obtained through clustering may be preferred, as it consists of fewer rules and is thus less expensive to simulate. An interesting topic for future research is the interpretation in terms of local linear submodels for the two techniques .

References

[1] R. Babuška. *Fuzzy Modeling for Control*. Kluwer Academic Publishers, Boston, USA, 1998.

[2] R. Babuška and H.B. Verbruggen. An overview of fuzzy modeling for control. *Control Engineering Practice*, 4(11):1593–1606, 1996.

[3] D. Dubois and H. Prade. *Fuzzy sets and systems: theory and applications*, volume 144 of *Mathematics in science and engineering*. Academic Press, 1980.

[4] I. Gath and A.B. Geva. Unsupervised optimal fuzzy clustering. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 7:773–781, 1989.

[5] D.E. Gustafson and W.C. Kessel. Fuzzy clustering with a fuzzy covariance matrix. In *Proc. IEEE CDC*, pages 761–766, San Diego, CA, USA, 1979.

[6] J.S.R. Jang. ANFIS: Adaptive-network-based fuzzy inference systems. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(3):665–685, 1993.

[7] T.A. Johansen and B.A. Foss. Constructing NARMAX models using ARMAX models. *International Journal of Control*, 58(5):1125–1153, 1993.

[8] U. Kaymak and R. Babuška. Compatible cluster merging for fuzzy modeling. In *Proceedings FUZZ-IEEE/IFES'95*, pages 897–904, Yokohama, Japan, March 1995.

[9] R. Krishnapuram and Chin-Pin Freg. Fitting an unknown number of lines and planes to image data through compatible cluster merging. *Pattern Recognition*, 25(4):385–400, 1992.

[10] I.J. Leontaritis and S.A. Billings. Input-output parametric models for nonlinear systems, part 1: Deterministic nonlinear systems. *International Journal of Control*, 41, 1985.

[11] R. Murray-Smith. *A Local Model Network Approach to Nonlinear Modeling*. Ph.D. Thesis Univ. Strathclyde, 1994.

[12] O. Nelles and R. Isermann. Basis function networks for interpolation of local linear models. In *IEEE Conference on Decision and Control (CDC)*, pages 470–475, Kobe, Japan, December 1996.

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

[14] K.A. Zinner. *Aufladung von Verbrennungsmotoren*. Springer, Berlin, 1985.