Supervised Hierarchical Clustering in Fuzzy Model Identification

Benjamin Hartmann, Oliver Bänfer, Oliver Nelles, Anton Sodja, Luka Teslić and Igor Škrjanc

***Abstract*—This paper presents a new, supervised, hierarchical clustering algorithm (SUHICLUST) for fuzzy model identification. The presented algorithm solves the problem of global model accuracy together with the interpretability of local models as valid linearizations of the modeled nonlinear system. The algorithm combines the merits of supervised, hierarchical algorithms, which are based on heuristic tree-construction algorithms together with the advantages of fuzzy product space clustering. The high flexibility of the validity functions obtained by fuzzy clustering combined with supervised learning results in an efficient partitioning algorithm, which is independent of initialization and results in a parsimonious fuzzy model. Furthermore, the usability of SUHICLUST is very undemanding, because it delivers in contrast to many other methods reproducible results. In order to get reasonable results, the user only has to set either a threshold for the maximum number of local models or a value for the maximum allowed global model error as a termination criterion. For fine-tuning, the interpolation smoothness controls the degree of regularization. The performance is illustrated on both analytical examples and benchmark problems from literature.**

***Index Terms*—Fuzzy model identification, fuzzy clustering, supervised learning, hierarchical tree-construction**

I. Introduction

A. Motivation

IN PRACTICE, data-driven models can be used for a broad range of application fields, e.g. one-step-ahead prediction, simulation, optimization, data mining, control and fault detection. In this sense, for the purpose of identifying nonlinear, static and dynamic processes the application of Takagi-Sugeno neuro-fuzzy systems that employ local models is very common. Hence, nonlinear relationships are emulated with a definite number of local sub-models. However, widely different strategies have been pursued for the partitioning of the input space which determines the validity regions of the local models. The model properties crucially depend on the chosen strategy.

Once, the validity regions of the fuzzy model are determined, it is easy to efficiently estimate the parameters of the local linear models by local or global least squares methods. The decisive difference between all proposed algorithms to construct local linear model structures is the strategy to partition the input space, i.e., to choose the validity regions and consequently the parameters of the validity functions. Therefore, in this contribution an efficient structure identification algorithm is proposed that automatically finds a very flexible input space partitioning such that the resulting fuzzy model is a valid representation of the underlying process data.

B. Hartmann, O. Bänfer and O. Nelles are with the Department of Mechanical Engineering, University of Siegen, Paul-Bonatz-Str. 9–11, D-57068 Siegen, Germany. e-mail: benjamin.hartmann@uni-siegen.de.

A. Sodja, L. Teslić and I. Škrjanc are with the Department of Electrical Engineering, University of Ljubljana, Tržaška 25, SI-1000 Ljubljana, Slovenia. e-mail: igor.skrjanc@fe.uni-lj.si.

B. Survey Over the Related Works

Many different approaches have appeared in the literature relating to the design of fuzzy models, see e.g. [6] and [7]. To date, the Gustafson-Kessel [33] or Gath-Geva [34] clustering algorithms are the most popular partitioning strategies for building local linear model networks. These algorithms are applied for searching hyper-ellipsoids of equal or different volumes in the input space or the product space, respectively. Due to the high flexibility of the validity functions in size and orientation the curse of dimensionality is a much lesser issue than for most competing strategies. However, this comes at the price of a reduced interpretability in terms of fuzzy logic, because the multi-dimensional validity functions cannot be projected to one-dimensional membership functions without loosing modeling accuracy.

In the following, an overview about the popular literature proposed for input space partitioning is given.

1) Partitioning based on fuzzy clustering: The use of fuzzy clustering algorithms, in general, involves two major problems: the algorithm is strongly dependent on the initialization and requires prior knowledge about the number of clusters or the number of local models, respectively. To overcome these problems, a lot of efforts have been made in systematic design approaches for fuzzy model identification [11], [12], [13] and [14]. In [10] an iterative scheme of regression-based fuzzy c-means clustering in the input-output space is proposed. The fuzzy rules are obtained by a projection of these clusters onto each of the problem dimensions. [9] uses fuzzy c-means clustering to find an appropriate number of clusters in the output space and then the projection is used to define the fuzzy partition. The major problem with this approach is that the relation between the input and the output space is not injective, and usually there is more than one cluster in the input space that corresponds to a cluster in the output space. Recently, in [54] fuzzy c-means clustering with Gaussian membership functions together with the application of local least squares estimation is applied on modeling the nitrate concentration in groundwater. [15] proposes a systematic methodology of fuzzy logic modeling which applies fuzzy c-means clustering. The algorithm is based on a validity index optimization which is problematic, because no reliable validity index exists to

solve the problem of optimal fuzzy clustering. The problem of dimensionality is partially solved by using decomposed fuzzy systems as proposed in [20] and by involving hierarchical fuzzy systems as shown in [19]. Another hierarchical fuzzy-clustering approach can be found in [51]. It is based on a weighted fuzzy c-means algorithm. A neuro-fuzzy algorithm which automatically defines the partitioning of the space and the model parameters using recursive singular-value decomposition is reported in [23] as an extension of the grid-based mountain clustering method. A complete fuzzy systems identification, known as subtractive clustering is proposed in [24].

2) Fuzzy systems based on global optimization methods: A straightforward idea is to simply optimize all parameters of the validity functions. One drawback of this approach is that the model complexity (number of validity functions = number of local models) must be fixed beforehand. Furthermore, the number of parameters can become huge, especially for high-dimensional input spaces. That is the reason why global optimization methods are popular for this partitioning approach. However, the main problem of this kind of model-development tools is the high computational cost which causes severe problems in terms of their applicability [5].

One of the most popular algorithm in this framework is the ANFIS algorithm [21]. The idea of the adaptive neural-fuzzy scheme ANFIS is to reduce the number of validity function parameters by constraining them, e.g. to a grid structure. It is used to find the global fuzzy model. The structure of the model and the model parameters are usually estimated independently, as proposed in [22], where the input domain is partitioned in fuzzy subspaces.

In [12], [16], [52], [53] and [17] genetic algorithms are used to find the appropriate structure and parameters of the fuzzy model. An evolutionary programming algorithm is proposed in [14] and a tabu search algorithm in [18] to define the fuzzy model structure and parameters. Furthermore, examples for some recent methods in this field can be found in [55], [41], [42] and [43].

3) Evolving fuzzy systems: Evolving neuro-fuzzy systems are able to online-adapt rule premises and rule consequents parameters of a Takagi-Sugeno fuzzy system simultaneously. DENFIS [44], eTS [46], FLEXFIS [47], SONFIN [48] and SOFMLS [49] are recently popular candidates representing this group of algorithms.

4) Heuristic tree-construction algorithms: Another possible way of partitioning the input-output space involves heuristic tree-construction algorithms which include a supervised learning paradigm. One idea to realize a tree construction is a simple, axis-orthogonal partitioning strategy which results in a fuzzy model. CART [25] and LOLIMOT [28], [29] are popular algorithms in this field. The biggest advantage of these methods is their very low computational effort, because the structure parameters can be found via heuristical methods. Therefore, it does not require time-consuming nonlinear optimization methods to obtain the partitioning.

An improvement of the axes-orthogonal approach is the application of an axes-*oblique* partitioning strategy, because it leads to parsimonious model structures that are well suited for mapping high-dimensional relationships. *Hinging hyperplanes*, firstly introduced in [26], can be used to realize this kind of partition. Hinging hyperplanes are functions that look like the cover side of a partly opened book. The direction of the hinge is then optimized to fit the underlying data optimally. [27] introduced hinging hyperplanes for piecewise local linear models that are smoothed by interpolation functions. Efficient construction algorithms that extend this idea are proposed in [30] and [50].

C. Our Approach

Goal behind our work was the development of a modeling approach that is well suited for the modeling of highly nonlinear processes owing to high flexible validity functions. The strengths of heuristic tree-construction algorithms like LOLIMOT, namely the supervised learning strategy and the incremental growing, are combined with the advantages of product space clustering. Figure 1 illustrates the connection of the new SUHICLUST algorithm to these approaches. Due to the incorporation of the model error in the partitioning procedure, the proposed algorithm is supervised. Additionally, SUHICLUST contains unsupervised learning, because of the local application of product space clustering. The model complexity is incrementally increased and the algorithm will stop, if the model error is small enough or the maximum number of local models is derived.

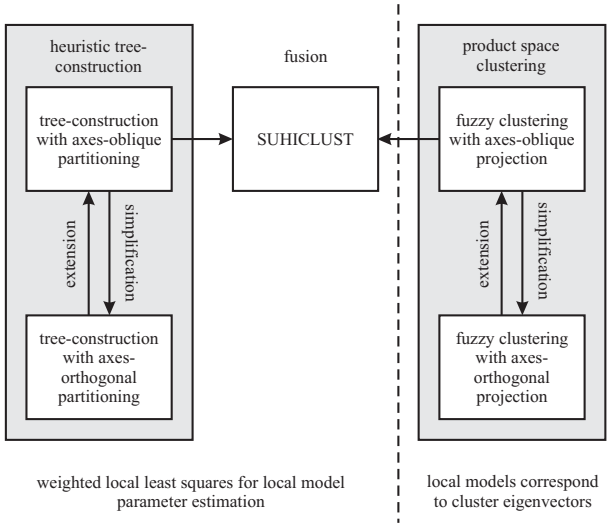

Fig. 1. SUHICLUST as a combination of the heuristic tree-construction algorithm and product-space clustering. The parameters of the local models are estimated with weighted least squares (WLS).

In contrast to many established methods, SUHICLUST models are generated with an axes-oblique partitioning strategy. The validity functions, normalized Gaussian basis functions, are more flexible than, e.g., sigmoidal basis functions and, therefore, especially good applicable for modeling nonlinear relationships with many variables. Unsupervised learning approaches mostly deliver non-reproducible results and are possibly very sensitive to local minima or the distribution of the training data. The model complexity has to be fixed a-priori and the modeling result depends crucially on the chosen number of clusters.

An important aspect in the development of SUHICLUST was its good usability. The algorithm must be easy to use, i.e. the algorithm has to have as less tuning parameters for the user to adjust as achievable. It is possible to generate feasible modeling results only by setting either a threshold for the maximum allowed global model error or by setting the maximum number of rules or local models, respectively. Additionally, the user can adjust the interpolation smoothness between the local models. Due to the application of local, weighted least squares estimation this is a powerful tool to control the regularization of the model [37]. The algorithm delivers *reproducible* modeling results which is a remarkable advantage compared to other modeling schemes that produce different results in each run of the algorithm.

The paper is organized as follows. Section II presents the structure of Takagi-Sugeno fuzzy models. Section III compares the attributes of clustering versus hierarchical tree-construction algorithms. In Sect. IV the supervised, hierarchical clustering algorithm SUHICLUST is proposed and described in detail. Section V describes a series of experiments where SUHICLUST is compared with other recent algorithms. We considered both analytical examples and the comparison with results from the literature. Concluding remarks are given Sect. VI.

II. Takagi-Sugeno fuzzy models

Fuzzy models in Takagi-Sugeno form are very important nonlinear approximators for nonlinear static functions and nonlinear dynamic model approximation [7], [31] and [32]. This is mainly due to the transparency of the local *linear* models and the transfer of the methods from the classical linear control theory to the nonlinear world.

The output $\hat{y}$ of the fuzzy model with nu inputs $\mathbf{u} = [u_1 \ u_2 \ \cdots \ u_{nu}]^T$ can be calculated as the interpolation of M local model outputs $\hat{y}_i(\cdot)$, $i = 1, \ldots, M$, see Fig. 2,

$$\hat{y} = \sum_{i=1}^{M} \hat{y}_i(\mathbf{u})\Phi_i(\mathbf{u}) \tag{1}$$

where the $\Phi_i(\cdot)$ are called the normalized membership or weighting functions. These normalized membership functions describe the regions and the contribution of the local model (LM) to the global fuzzy model output. The fuzzy model in Eq. 1 realizes a set of M fuzzy rules where the $\Phi_i(\cdot),\ i = 1, ..., M$ represent the rule premises and the $\hat{y}_i(\cdot),\ i = 1, ..., M$ are the associated rule consequents. Continuous behavior assumes a smooth transition between the local models, and this implies a smooth normalized membership in the interval $[0, 1]$. The normalized membership functions form a *partition of unity*:

$$\sum_{i=1}^{M} \Phi_i(\mathbf{u}) = 1 \tag{2}$$

Thus, everywhere in the input space the contributions of all the local models are equal to one.

In principle, the structure of the consequent part is of an arbitrary type. The most common structures of the consequent part are polynomials. Polynomials of degree 0 (constants) yield a fuzzy system with singletons. Polynomials of degree 1 (linear) provide local linear model structures, which is by far the most popular choice. As the degree of the polynomials increases, the number of local models required for a certain accuracy decreases. Besides the possibility of transferring the classical linear theory to the nonlinear world, the fuzzy models with a linear consequent part seem to represent a good trade-off between the required number of local models and the complexity of the local models themselves. As a result of all these facts, in the rest of this paper the consequent part is given in linear, strictly speaking, affine form, as follows in Eq. 3.

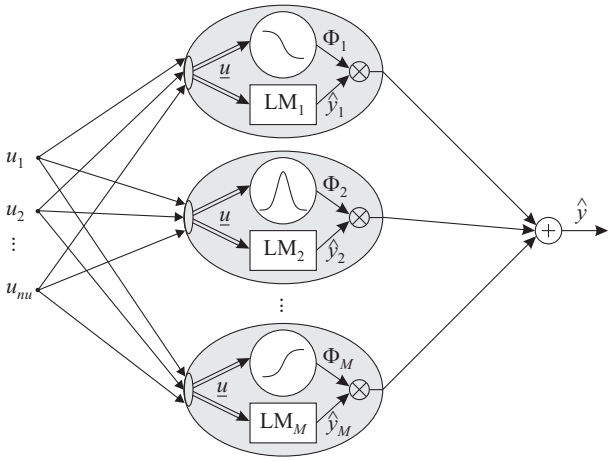

Fig. 2. Local model network: The outputs $\hat{y}_i(\cdot)$ of the local models (LM_i) are weighted with their normalized membership-function values $\Phi_i(\cdot)$ and summed up.

$$\hat{y}_i(\mathbf{u}) = \theta_{i,0} + \theta_{i,1}u_1 + \theta_{i,2}u_2 + \ldots + \theta_{i,nu}u_{nu} \tag{3}$$

One of the key features of TS fuzzy models is that the input spaces for the local models and for the normalized membership functions can be chosen independently. This means that Eq. 1 has to be extended to Eq. 4:

$$\hat{y} = \sum_{i=1}^{M} \hat{y}_i(\mathbf{x})\Phi_i(\mathbf{z}) \tag{4}$$

with $\mathbf{x} = [x_1 \ x_2 \ \cdots \ x_{nx}]^T$ spanning the consequent input space and $\mathbf{z} = [z_1 \ z_2 \ \cdots \ z_{nz}]^T$ spanning the premise input space. Graphically, this is presented in Fig. 3. This feature enables the user to incorporate prior knowledge about the strength of the nonlinearity from each input to the output into the model structure. Or conversely, the user can draw such conclusions from a black-box model that has been identified from the data.

Especially for dynamic models, where the model inputs include delayed versions of the physical inputs and outputs, the dimension nx becomes very large in order to cover all the dynamical effects. In the most general case (universal approximator) this is also true for nz. However, for many practical problems a lower-dimensional $\mathbf{z}$ can be chosen, sometimes even one or two scheduling variables can yield sufficiently accurate models. This feature can substantially weaken the curse of dimensionality.

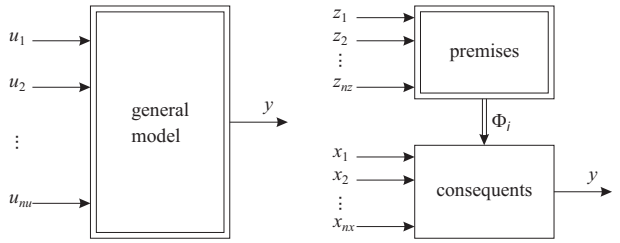

Fig. 3. For TS fuzzy models the inputs can be assigned to the premise and/or consequent input space according to their nonlinear or linear influence on the model output.

If the normalized membership functions are determined once, it is easy to efficiently estimate the parameters of the local linear models θ_{ij} by local or global least-squares methods. The decisive difference between all the proposed algorithms for constructing local linear model structures is the strategy of the input-space partitioning spanned by $\mathbf{z}$, i.e., to choose the validity regions and consequently the parameters of the validity functions. This strategy determines the key properties of both: the construction algorithm and the finally constructed model.

An example of a fuzzy model with three local models (LMs) and the corresponding normalized membership functions is shown in Fig. 4.

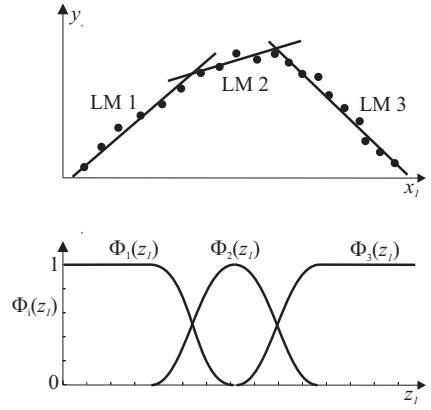

Fig. 4. Local model network. Local linear models (top) and normalized membership functions (bottom).

III. Clustering versus hierarchical tree-construction

In this section the basic ideas of two conceptually different space-partitioning paradigms are given: the clustering and the hierarchical tree-construction algorithms. Some of the advantages and drawbacks of each approach are discussed.

Many clustering algorithms focus on the product space that is jointly spanned by the inputs and the output. In the case of product-space clustering the consequent input space and the premise input space coincide. This means that the data set used in the clustering procedure consists of the inputs and the output. Unfortunately, clustering algorithms are mostly restricted to have only *one* single output dimension. However, in the literature solutions to this problem are already proposed. For example, in [8] Gustafson-Kessel clustering for MIMO systems is introduced.

The main clustering algorithms that are usable in designing fuzzy control models are presented in [6]. The most commonly used algorithms are Gustafson-Kessel (GK) [33], Gath-Geva (GG) [34] and the extended Gath-Geva algorithm proposed in [5]. The cluster is usually defined as its center and the fuzzy covariance matrix. The fuzzy center matrix $\mathbf{C} = [\mathbf{c}_1, ..., \mathbf{c}_i, ..., \mathbf{c}_M]$ and $\mathbf{c}_i,\ i = 1, ..., M$ stands for the center of the i-th fuzzy cluster $\mathbf{c}_i = [c_{i,u_1}, ..., c_{i,u_{nu}}, c_{i,y}]^T$ in the product space. The fuzzy covariance matrix of the i-th fuzzy cluster $\mathbf{\Sigma}_i \in \mathbb{R}^{(nu+1)\times(nu+1)}$ is defined as follows:

$$\mathbf{\Sigma}_i = \sum_{k=1}^{N} \Phi_i^2(\mathbf{z}(k)) \left(\mathbf{z}(k) - \mathbf{c}_i\right) \left(\mathbf{z}(k) - \mathbf{c}_i\right)^T \tag{5}$$

where $\Phi_i(\mathbf{z}(k))$ defines the normalized membership degree of the data vector $\mathbf{z}(k)$ to the i-th cluster. The data vector is defined as $\mathbf{z}(k) = \left[\mathbf{u}^T(k)\, y(k)\right]^T,\ k = 1, ..., N$. The dimension of the data vector $\mathbf{z}$ is $nz = nu + 1$. The fuzzy covariance matrix defines the directions and the variability of the data in the input-output space. With the application of singular-value decomposition the fuzzy covariance matrix is decomposed as will be stated next:

$$\mathbf{\Sigma}_i = \mathbf{P}_i \mathbf{\Lambda}_i \mathbf{P}_i^T \tag{6}$$

where $\mathbf{P}_i \in \mathbb{R}^{(nz)\times(nz)}$ stands for the matrix of eigenvectors $\mathbf{P}_i = [\mathbf{p}_{i,1}, \mathbf{p}_{i,2}, \ldots, \mathbf{p}_{i,nz}]$ and $\mathbf{\Lambda}_i \in \mathbb{R}^{(nz)\times(nz)}$ for the matrix of eigenvalues $\mathbf{\Lambda}_i = diag(\lambda_{i,j}),\ j = 1, ..., nz$. Trying to find the local linear models that will describe the nonlinear process given by the data set means that one has to find the appropriate number of clusters and their centers to have the smallest normalized eigenvalues of all the clusters smaller than a certain threshold Δ. This threshold is described with

$$\frac{\lambda_{i,nz}}{trace(\mathbf{\Lambda}_i)} < \Delta \quad i = 1, ..., M \tag{7}$$

Usually, the threshold Δ is chosen to be equal to 0.05. This means that 5 percent of the data variability is neglected. The vector in the direction of the neglected data $\mathbf{p}_{i,nz}$ is normal to the local hyperplane described in the vector equation form as:

$$\left(\mathbf{r}_i - \mathbf{c}_i\right)^T \cdot \mathbf{p}_{i,nz} = 0, \quad i = 1, ..., M \tag{8}$$

In particular, the GK algorithm is able to discover local hyperplanes in the input-output space by forming ellipsoids that can be calculated from the fuzzy cluster center and the fuzzy covariance matrix [7]. Due to the high flexibility of the normalized membership functions, in size and orientation, the problem of dimensionality is a much smaller issue than for most competing strategies.

The nature of the GK algorithm, which initializes the partitioning matrix randomly, makes the direct use of this algorithm sometimes difficult. The randomized initialization does not enable a unique solution in all cases and when a very small threshold Δ is used, this can cause the problem of termination. Another disadvantage is the unknown number of fuzzy clusters. The algorithm does not converge or gives very

bad results because of the singularity of the fuzzy covariance matrices, if the data set has some sparse regions.

The second partitioning paradigm that is presented here is the heuristic tree search algorithm, which is given in the literature [25]. Based on this idea, many similar partitioning strategies are proposed in the literature [28], [35], [36]. The key idea is to incrementally subdivide the input space using axes-orthogonal cuts. Besides their simplicity, the strict separation between the rule premises and the rule consequents and their low computational demand, one major advantage is their easy interpretability in the sense of fuzzy logic. The axes-orthogonal partitioning always allows a projection of the normalized membership regions to the one-dimensional input variables. Their main drawback inherently lies in the partitioning technique, which does not give a parsimonious fuzzy model in the sense of the number of local models. This becomes more and more important and problematic by increasing the dimensionality of the premise input space.

Both of the strategies yield flat models. Even if the *algorithm* is hierarchically organized, the constructed *fuzzy model* itself is flat in the sense that all the normalized membership functions $\Phi_i(\cdot)$ can be calculated in parallel. This is an important feature, if the fuzzy model really should be realized in hardware or using some parallel computers.

The heuristic tree-construction algorithms offer a number of attractive features. Their only major shortcoming is their sensitivity to the problem of dimensionality as a consequence of the restriction of axes-orthogonal splits. A fusion with fuzzy clustering algorithms gives an algorithm that can overcome this drawback. The usage of highly flexible normalized membership functions obtained by fuzzy clustering enables the algorithm to overcome the problem of dimensionality. This flexibility is a consequence of the fuzzy covariance matrix that allows an arbitrary orientation and size of the clusters in the input-output space.

IV. Supervised hierarchical clustering

The SUHICLUST (SUpervised HIerarchical CLUSTering) is a fusion of the unsupervised fuzzy clustering algorithm and the supervised hierarchical tree-construction algorithm. It combines the advantages of both algorithms. The main features of this fuzzy model construction algorithm are:

- *Incremental:* In each iteration an additional local model is generated.
- *Splitting:* In each iteration the local model with the worst local error measure is split into two submodels. The procedure of splitting is shown in Fig. 5.
- *Local least squares:* The parameters of the local models are locally estimated using a weighted least-squares method. This is computationally extremely cheap and introduces a regularization effect, which increases the robustness [37].
- *Adaptive resolution:* The smoothness of the local model interpolation depends on the fuzzy covariance matrix obtained by fuzzy clustering and therefore on the size of the normalized membership regions. The smaller the normalized membership regions are, the less smooth the interpolation will be.
- *Split optimization:* The application of the Gustafson-Kessel [33] fuzzy clustering in the product space determines the new split in the input space.
- *Nested optimization:* After an evaluation of the new split by fuzzy clustering, the parameters of the two involved local models are newly estimated by a local, weighted least-squares method.

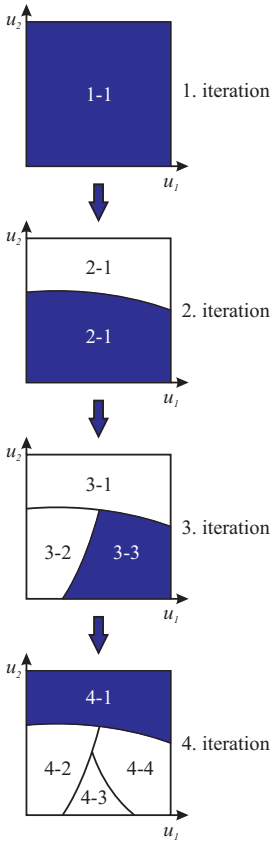

Fig. 5. Operation of the SUHICLUST structure search algorithm in the first four iterations for a two-dimensional input space.

To optimize the splitting parameters (fuzzy covariance matrix), Gustafson-Kessel fuzzy clustering is used as a nonlinear optimization technique. The data points that correspond to the local model that is going to be split are taken as the clustering data. The normalized membership values classify the data. Each time after the generation of two new local models, two local, weighted least-squares estimations are carried out in order to optimize the local model parameters of the two newly generated local models.

A. SUHICLUST algorithm

A detailed description of the SUHICLUST algorithm will be given next. Let the $(N \times nu + 1)$-dimensional data set of measurements be denoted as follows:

$$\tilde{\mathbf{Z}} = \begin{bmatrix} \tilde{u}_1(1) & \cdots & \tilde{u}_{nu}(1) & \tilde{y}(1) \\ \tilde{u}_1(2) & \cdots & \tilde{u}_{nu}(2) & \tilde{y}(2) \\ \vdots & \ddots & \vdots & \vdots \\ \tilde{u}_1(N) & \cdots & \tilde{u}_{nu}(N) & \tilde{y}(N) \end{bmatrix} \tag{9}$$

The measured data set can also be expressed as $\tilde{\mathbf{z}}(k) = [\tilde{\mathbf{u}}(k)\ \tilde{y}(k)]^T,\ k = 1, ..., N$ and the vector of input measurements at the time instant k is described by $\tilde{\mathbf{u}}(k) = [\tilde{u}_1(k)\ \tilde{u}_2(k)\ \cdots\ \tilde{u}_{nu}(k)]^T,\ k = 1, ..., N$.

The data matrix is centered and scaled and the normalized data matrix $\mathbf{Z}$ is then described as:

$$\mathbf{Z} = \begin{bmatrix} u_1(1) & \cdots & u_{nu}(1) & y(1) \\ u_2(2) & \cdots & u_{nu}(2) & y(2) \\ \vdots & \ddots & \vdots & \vdots \\ u_1(N) & \cdots & u_{nu}(N) & y(N) \end{bmatrix} \tag{10}$$

where the elements of the data matrix are defined with $u_j(k) = \frac{\tilde{u}_j - m_{uj}}{\tilde{u}_{jMAX}},\ j = 1, ..., nu,\ k = 1, ..., N$, $y(k) = \frac{\tilde{y} - m_y}{\tilde{y}_{MAX}},\ k = 1, ..., N$, where $m_{u_j} = \frac{1}{N}\sum_{i=1}^{N} \tilde{u}_j(i),\ j = 1, ..., nu$ and $m_y = \frac{1}{N}\sum_{i=1}^{N} \tilde{y}(i)$ and where $\tilde{u}_{j_{MAX}} = \max_k \left(|\tilde{u}_j(k) - m_{uj}|\right),\ j = 1, ..., nu$ and $\tilde{y}_{MAX} = \max_k \left(|\tilde{y}(k) - m_y|\right)$. The normalized data set is also presented with the data vectors $\mathbf{z}(k) = [\mathbf{u}(k)\ y(k)]^T,\ k = 1, ..., N$ and the vector of normalized input measurements at the time instant k, described by $\mathbf{u}(k) = [u_1(k)\ u_2(k)\ \cdots\ u_{nu}(k)]^T$, $k = 1, ..., N$.

In the first step of the SUHICLUST algorithm, the covariance matrix $\mathbf{\Sigma}_0$ of the data $\mathbf{Z}$ is computed. The covariance matrix is described with:

$$\mathbf{\Sigma}_0 = \frac{1}{N-1}\mathbf{Z}^T\mathbf{Z} \tag{11}$$

The unit eigenvectors and the corresponding eigenvalues of the data covariance matrix $\mathbf{\Sigma}_0$ are calculated using singular-value decomposition. The eigenvalues of the covariance matrix $\mathbf{\Sigma}_0$ represent the variances of the data matrix $\mathbf{Z}$ in the direction of the corresponding eigenvectors. With $\mathbf{c}_0$ the center of the measured data set is denoted, where $\mathbf{c}_0 = [m_{u_1} \ldots m_{u_{nu}}\ m_y]^T$.

By using the singular-value decomposition from Eq. 6 a matrix of eigenvectors and eigenvalues is obtained. The main direction of the data expansion is the direction with the largest eigenvalue λ_{01}, and this is denoted by the main eigenvector $\mathbf{p}_{01}$. This means that the variance σ_0^2 around the center of the data (mean of the data) in the direction of the main eigenvector equals λ_{01}. The initial centers of the clusters $\mathbf{v}_{11}$ and $\mathbf{v}_{12}$ for the centered data set $\mathbf{Z}$ are defined as follows:

$$\begin{aligned} \mathbf{v_{11}} &= -\boldsymbol{\delta}_1 \\ \mathbf{v_{12}} &= \boldsymbol{\delta}_1 \end{aligned} \tag{12}$$

where $\boldsymbol{\delta}_1$ defines the main eigenvector scaled with the corresponding standard deviation to capture the majority of the data:

$$\boldsymbol{\delta}_1 = \sigma_0 \mathbf{p}_0$$

With the deterministic procedure of calculating the unit eigenvector and the corresponding standard deviation by using singular-value decomposition, the initial position of the cluster centers is directed. The initial centers are placed away from the mean value of the measured variables $\mathbf{c}_0$ to embrace the majority of the data matrix $\mathbf{Z}$ in the direction of the data expansion.

The data matrix $\mathbf{Z}$ and the initial cluster centers $\mathbf{v_{11}}$, $\mathbf{v_{12}}$ are the inputs to the Gustafson-Kessel algorithm, which results in two cluster centers $\mathbf{c}_1$, $\mathbf{c}_2$ ($M = 2$) and in the fuzzy covariance matrices $\mathbf{\Sigma}_{11}$, $\mathbf{\Sigma}_{12}$.

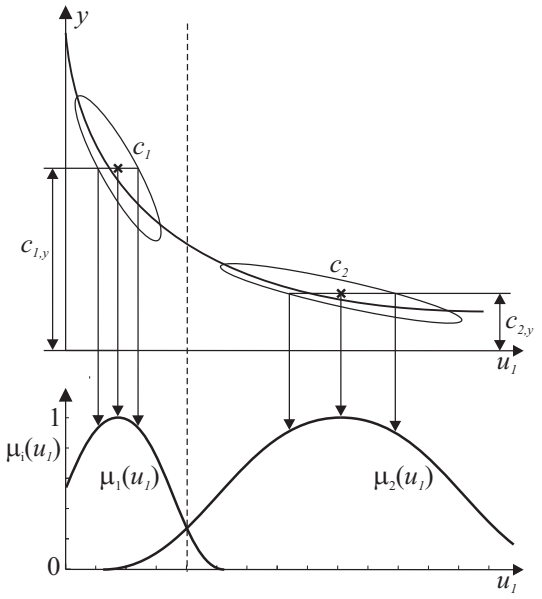

Fig. 6. Projection of clusters (top) from the product space to the input space leads to membership functions (bottom).

In the next step, the parameters of the local linear models are calculated. In order to project the clusters obtained by the Gustafson-Kessel algorithm from the product space to the input space, the cluster dimension of the output y is kept constant at the corresponding cluster-center value $c_{i,y}$. Therefore, the cluster rotation in the output dimension is neglected. Fig. 6 illustrates this using an example with one input z_1 and one output y. The two cluster projections μ_i are generated by slicing the clusters at their output center coordinate $c_{i,y}$.

The distance $d_i(k)$ from a data point k to each center $\mathbf{c}_i = [c_{i,1}\ c_{i,2}\ \cdots\ c_{i,nu}\ c_{i,y}]^{\mathrm{T}}$ is calculated by using the fuzzy covariance matrix $\mathbf{\Sigma_i}$, which scales and rotates the axes:

$$d_i^2(k) = \left(\mathbf{z}(k) - \mathbf{c}_i\right)^{\mathrm{T}} \mathbf{\Sigma}_i^{-1} \left(\mathbf{z}(k) - \mathbf{c}_i\right) \tag{13}$$

The fuzzy covariance matrix $\mathbf{\Sigma_i}$ has a symmetric shape and is of size $(nu + 1 \times nu + 1)$, if one output is applied:

$$\mathbf{\Sigma_i} = \begin{bmatrix} \sigma_{1,1}^2 & \sigma_{1,2}^2 & \cdots & \sigma_{1,nu+1}^2 \\ \sigma_{2,1}^2 & \sigma_{2,2}^2 & \cdots & \sigma_{2,nu+1}^2 \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{nu+1,1}^2 & \sigma_{nu+1,2}^2 & \cdots & \sigma_{nu+1,nu+1}^2 \end{bmatrix}_{\mathbf{i}} \tag{14}$$

with $\sigma_{k,l}^2 = \sigma_{l,k}^2 \quad \forall \quad k, l = 1, \ldots, nu + 1$. The inverse is calculated using the Moore-Penrose pseudo-inverse that is well suited, if the fuzzy covariance matrices are ill conditioned.

The membership functions $\mu_i(\cdot)$ of a Gaussian basis function network are given by:

$$\mu_i\left(\mathbf{z}(k)\right) = e^{-\alpha d_i^2(k)} \tag{15}$$

where $\alpha \geq 1$ is a factor that defines the smoothness of the Gaussian functions $\mu_i(\mathbf{z}(k)) > 0$ and consequently the

sharpness of the normalized membership functions $\Phi_i(\mathbf{z}(k))$, i.e., the normalized Gaussian functions.

To achieve a *partition of unity* the membership functions have to be normalized to obtain the normalized membership functions:

$$\Phi_i(\mathbf{z}(k)) = \frac{\mu_i(\mathbf{z}(k))}{\sum_{j=1}^{M} \mu_j(\mathbf{z}(k))} \tag{16}$$

Once the fuzzy covariance matrices are determined, i.e., the partitioning of the input space for the rule premises is accomplished, the parameters of the local linear models, i.e., the parameters of the rule consequents, can be estimated with the weighted least-squares (WLS) method.

A new set $\mathbf{Z}_i,\ i = 1, ..., M$ is formed for each cluster from the data set $\mathbf{Z}$, such that it satisfies the criteria:

$$\Phi_i(\mathbf{z}(k)) > 0 \tag{17}$$

The set $\mathbf{Z}_i$ is defined as $\mathbf{Z}_i = \{\mathbf{z}_i(k)\}$ where $\Phi_i(\mathbf{z}_i(k)) > 0,\ k = 1, ..., N_i$ and where N_i stands for the number of data rows from $\mathbf{Z}$ that fulfill the criteria in Eq. 17. The data set $\mathbf{Z}_i$ together with the normalized membership function weights $\Phi_i(\mathbf{z}_i(k))$ correspond to the i-th cluster.

The parameters of the i-th local linear model are obtained to optimally approximate the output variable $y_i(k) = y(k)\Phi_i(\mathbf{z}_i(k))$, i.e., minimize the loss function J_i defined in Eq. 18 using the WLS method.

$$J_i = \sum_{k=1}^{N_i} \left(y_i(k) - \left(\theta_i^T \mathbf{u}_i(k) + \theta_{i0}\right) \Phi_i\left(\mathbf{z}_i(k)\right)\right)^2 \tag{18}$$

for $i = 1, ..., M$ and where θ_i and θ_{i0} are the parameters of the local linear model that approximate the i-th cluster data set $\mathbf{Z}_i$. The application of the WLS method delivers the $nx+1$ unknown parameters of the i-th local linear model $\hat{\theta}_i$ and $\hat{\theta}_{i0}$. This means that the approximation of the output variable $\hat{y}_i(k)$ of the i-th cluster equals:

$$\hat{y}_i(k) = \hat{\theta}_i^T \mathbf{u}_i(k) + \hat{\theta}_{i0},\ \ i = 1, ..., M,\ k = 1, ..., N_i \tag{19}$$

The quality of the local linear model that approximates the data of the i-th cluster is estimated as the relative standard deviation σ_{qi} of the approximation error. This local error measure is given as follows:

$$\sigma_{qi}^2 = \frac{1}{N_i - 1} \sum_{k=1}^{N_i} \frac{\epsilon_i^2(k)}{\sigma_{y_i}^2} \tag{20}$$

where

$$\epsilon_i(k) = y_i(k) - \left(\hat{\theta}_i^T \mathbf{u}_i(k) + \hat{\theta}_{i0}\right) \Phi_i\left(\mathbf{z}_i(k)\right)$$

with $\sigma_{y_i}^2 = \frac{1}{N_i - 1} \sum_{k=1}^{N_i} (y_i(k) - m_{y_i})^2$ and $m_{y_i} = \frac{1}{N_i - 1} \sum_{k=1}^{N_i} y_i(k)$.

In further iterations the cluster with the largest relative standard-deviation σ_{qi}, has to be splitted into two clusters and each is modelled by two new, local linear models. The procedure is repeated until a suitable approximation is reached.

The initialization of the new cluster centers is made in the following way:

$$\begin{aligned} \mathbf{v}_{i1} &= \mathbf{c}_{i-1} + \boldsymbol{\delta}_i \\ \mathbf{v}_{i2} &= \mathbf{c}_{i-1} - \boldsymbol{\delta}_i \end{aligned} \tag{21}$$

where $\boldsymbol{\delta}_i = \sigma_i \mathbf{p}_{i,1}$. A graphical representation of the i-th iteration of the splitting procedure is given in Fig. 7.

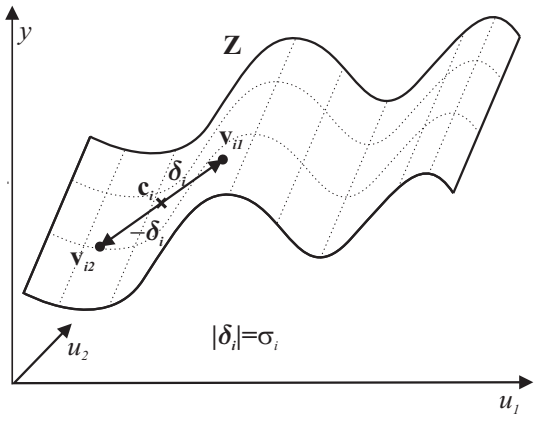

Fig. 7. A graphical representation of the i-th iteration of the splitting procedure.

The centers of the clusters $\mathbf{c}_i$ and the fuzzy covariance matrices $\mathbf{\Sigma}_i,\ i = 1, ..., M$ are re-transformed to the measured data set and are denoted as $\tilde{\mathbf{c}}_i$ and $\tilde{\mathbf{\Sigma}}_i$. This means that the centers of the fuzzy clusters for the measured data set are now defined as:

$$\tilde{\mathbf{c}}_i = \mathbf{c}_i + \mathbf{c}_0,\ i = 1, ..., M \tag{22}$$

and the fuzzy covariance matrices are rescaled with:

$$\tilde{\mathbf{\Sigma}}_i = \mathbf{\Sigma}_i \cdot \mathbf{T} \tag{23}$$

where $\mathbf{T}$ stands for

$$\mathbf{T} = \begin{bmatrix} m_{u_1}^2 & m_{u_1} m_{u_2} & \cdots & m_{u_1} m_{u_{nu}} \\ m_{u_1} m_{u_2} & m_{u_2}^2 & \cdots & m_{u_2} m_{u_{nu}} \\ \vdots & \vdots & \ddots & \vdots \\ m_{u_1} m_{u_{nu}} & m_{u_2} m_{u_{nu}} & \cdots & m_{u_{nu}}^2 \end{bmatrix} \tag{24}$$

The re-transformed centers of the clusters and their fuzzy covariance matrices represent, in a way, the model of the process that results in the measured data set $\tilde{\mathbf{Z}}$. A more transparent and usable form of the model is the parametric fuzzy model defined by $\hat{\theta}_i$ and $\hat{\theta}_{i0},\ \ i = 1, ..., M$. This type of model is used to calculate the model output variable. The procedure is described next. First, the distances of the measured samples to the centers of the clusters are calculated as:

$$D_i^2(k) = \left(\left[\tilde{\mathbf{u}}^T(k)\ \tilde{c}_{iy}\right]^T - \tilde{\mathbf{c}}_i\right)^{\mathrm{T}} \tilde{\mathbf{\Sigma}}_i^{-1} \left(\left[\tilde{\mathbf{u}}^T(k)\ \tilde{c}_{iy}\right]^T - \tilde{\mathbf{c}}_i\right)$$

where $\tilde{\mathbf{\Sigma}}_i,\ i = 1, ..., M$ stands for the de-normalized fuzzy covariance matrix. Then, the normalized membership values are defined with:

$$\Phi_i(\tilde{\mathbf{u}}(k)) = \frac{\mu_i(\tilde{\mathbf{u}}(k))}{\sum_{j=1}^{M} \mu_j(\tilde{\mathbf{u}}(k))},\ \ k = 1, ..., N \tag{25}$$

where $\mu_i\left(\tilde{\mathbf{u}}(k)\right) = e^{-\alpha D_i^2(k)}$.

The membership values for the whole data set $\tilde{\mathbf{Z}}$ and the data set itself are used to estimate the parameters of the model $\hat{\theta}_i$ and $\hat{\theta}_{i0},\ i = 1, ..., M$ using the local WLS algorithm.

Finally, the whole model output is computed as follows:

$$\hat{y}(k) = \sum_{i=1}^{M} \left(\hat{\theta}_i^T \tilde{\mathbf{u}}(k) + \hat{\theta}_{i0}\right) \Phi_i(\tilde{\mathbf{u}}(k)), \quad k = 1, ..., N \quad (26)$$

In the next step, the quantitative validation of the obtained model is calculated using the normalized root-mean-squared error between the measured output and the approximated model output (*NRMSE*):

$$NRMSE(\tilde{\mathbf{y}}, \hat{\tilde{\mathbf{y}}}) = \sqrt{\frac{\sum_{k=1}^{N}(\tilde{y}(k) - \hat{\tilde{y}}(k))^2}{\sum_{i=1}^{N}(\tilde{y}(i) - \bar{y})^2}}$$

$$\bar{y} = \frac{1}{N}\sum_{k=1}^{N} \tilde{y}(k) \quad (27)$$

where $\tilde{\mathbf{y}}$ and $\hat{\tilde{\mathbf{y}}}$ stand for $\tilde{\mathbf{y}} = [\tilde{y}(1), \ldots, \tilde{y}(N)]^T$ and $\hat{\tilde{\mathbf{y}}} = \left[\hat{\tilde{y}}(1), \ldots, \hat{\tilde{y}}(N)\right]^T$, respectively.

If the obtained *NRMSE* does not fulfill the criteria given by

$$NRMSE(\tilde{\mathbf{y}}, \hat{\tilde{\mathbf{y}}}) < NRMSE_{max}$$

where $NRMSE_{max}$ stands for the maximum allowed value, the whole procedure is repeated until the criteria is reached or until the maximal number of local linear models is reached.

The pseudo-code of SUHICLUST is presented in Table I where the whole algorithm is divided into 15 steps, from the initialization to the step where the approximation of the output variable is calculated. The algorithm has one repeating sequence. The repeat-until loop from step 5 to step 15 is terminated when either the number of clusters becomes equal to the maximum number of possible clusters or the algorithm satisfies the *NRMSE* criteria.

V. Benchmark Study

The advantages of the supervised hierarchical clustering are shown in the following illustration examples. Several different comparisons are made to show the potential of this algorithm. For the experiments, SUHICLUST is compared to the following toolboxes: LOLIMOT [29], Gustafson-Kessel (GK) product space clustering [33], LOLIMOT GP / GP-V [41], FMID [6], ANFIS [21], DENFIS [44] and the modified Gath-Geva (GG) clustering algorithm proposed in [5]. For the comparisons with LOLIMOT GP / GP-V and the modified GG algorithm existing results from the literature are used. For all investigations with SUHICLUST, GK clustering, LOLIMOT, FMID, ANFIS and DENFIS the default toolbox parameter values are applied. DENFIS is used in offline mode with the high-order Takagi-Sugeno-type fuzzy rule set.

A comprehensive benchmark study is made with five different examples:

1) *Analytical example I*: The aim of the first example is to illustrate the very flexible partitioning capabilities of SUHICLUST. This is done on two analytical functions with a two-dimensional input space.
2) *Analytical example II*: In this example the advantages of SUHICLUST are shown compared to the axes-orthogonal partitioning strategy LOLIMOT and product space clustering for highly non-linear data sets with more than two inputs.
3) *Pharmaceutical data set*: This is a highly non-linear example with two inputs and the presence of noise.
4) *Modeling an engine characteristic map*: As an example taken from the literature, SUHICLUST is compared to a state-of-the-art-method, LOLIMOT GP / GP-V, that can be seen as an extension of the classical LOLIMOT algorithm. In this contribution, the partitioning is done with the application of genetic programming. Although the data set consists only of two inputs, the modeling exercise is challenging, because the data has sparse regions that can cause overfitting.
5) *Comparison on Automobile MPG benchmark*: As a last example the well known Automobile MPG data set is used to show the generalization performance of SUHICLUST. We compared the results with the findings in [5].

The first two examples are investigated without the influence of noise in order to compare the key characteristics of the different algorithms, namely the flexibility of the partitioning, the computation time and the applicability for a higher-dimensional problem with a strong non-linearity. Next, the remaining examples show the good applicability for real data sets with the presence of noise.

A. Analytical example I

As a first example, the benchmark problem Mars1, which is also used in [38] and [39], is presented. The function that is modeled is the following:

$$y = \frac{2e^{\left(8\left[(u_1-0.5)^2+(u_2-0.5)^2\right]\right)}}{e^{(8[(u_1-0.2)^2+(u_2-0.7)^2])} + e^{(8[(u_1-0.7)^2+(u_2-0.2)^2])}} \quad (28)$$

For the approximation, 900 equally distributed, noise-free data samples are generated.

Goal of the modeling was to achieve an NRMS error less than 0.05. With LOLIMOT, 21 local linear models are needed, while the SUHICLUST algorithm could achieve the same accuracy with 10 local linear models, and using the Anfis algorithm the same result is obtained with 16 local models, see Fig. 8. In this case the function *genfis1* is used, which actually does the grid partition on the data without clustering. The second function that can be used to generate a fuzzy inference system inside the Anfis toolbox in Matlab is *genfis2*, which uses fuzzy subtractive clustering applied to the data. The rule-extraction method first uses the *subclust* function to determine the number of rules and antecedent membership functions and then uses a linear least-squares estimation to determine each rule's consequent equations. It does not provide the algorithm that will give the best possible clustering for a certain number of clusters. For this reason a comparison of all the algorithms is rather difficult.

Next, the comparison between SUHICLUST and LOLIMOT is made on a 3D set of data where the process nonlinearity

Step 1. Initialization of the algorithm.
- Transformation of the row data set $\tilde{\mathbf{Z}} \longrightarrow \mathbf{Z}$ or $\tilde{\mathbf{z}} \longrightarrow \mathbf{z}$.
- Definition of SUHICLUST algorithm parameters:
  - The smoothness parameter α, $(\alpha = 1)$.
  - The maximum number of clusters M_{max}.
  - The tolerance for the approximation error $NRMSE_{max}$, $(NRMSE_{max} = 0.05)$.

Step 2. Computation of the covariance matrix $\mathbf{\Sigma_0}$.

Step 3. Definition of the initial centers of clusters: $\mathbf{v}_{11} = \delta_1$, $\mathbf{v}_{12} = -\delta_1$.

Step 4. GK clustering algorithm on whole data set $\tilde{\mathbf{Z}} \longrightarrow$ cluster centers $\mathbf{c}_1$ and $\mathbf{c}_2$, $M = 2$.

repeat

Step 5. Computation $\Phi_i(\tilde{\mathbf{z}})$, $k = 1, ..., N$, $i = 1, ..., M$.

Step 6. Computation $\hat{\theta}_i$ and $\hat{\theta}_{i0}$, $i = 1, ..., M$ using local LS.

Step 7. Computation σ_{qi}, $i = 1, ..., M$.

Step 8. For the cluster with the largest local error measure σ_{qi} (i=p) defines the initial cluster centers:

$$\mathbf{v}_{p1} = \mathbf{c}_{p-1} + \delta_p \,,$$
$$\mathbf{v}_{p2} = \mathbf{c}_{p-1} - \delta_p \,.$$

Step 9. GK clustering using only the splitting-cluster's data results in cluster-centers $\mathbf{c}_{p1}$ and $\mathbf{c}_{p2}$.

Step 10. Definition of new initial cluster centers:

$$(\mathbf{v}_1, ..., \mathbf{v}_p, \mathbf{v}_{p+1}, ..., \mathbf{v}_{M_{new}}) = (\mathbf{c}_1, ..., \mathbf{c}_{p,1}, \mathbf{c}_{p,2}, ..., \mathbf{c}_M), \quad M_{new} = M + 1 \,.$$

Step 11. GK clustering using all the data results in the cluster centers $\mathbf{c}_i$, $i = 1, ..., M$, $M = M_{new}$.

Step 12. Re-transformation of centers and fuzzy covariance matrices to the measured data set:

$$\mathbf{c_i} \longrightarrow \tilde{\mathbf{c}}\text{: } \tilde{\mathbf{c}}_i = \mathbf{c}_i + \mathbf{c}_0, \quad i = 1, ..., M \,,$$

$$\mathbf{\Sigma_i} \longrightarrow \tilde{\mathbf{\Sigma}}_i\text{: } \tilde{\mathbf{\Sigma}}_i = \mathbf{\Sigma}_i \mathbf{T}, \; i = 1, ..., M \,.$$

Step 13. Calculation of $\Phi_i(\tilde{\mathbf{z}}(k))$, $k = 1, ..., N$, $i = 1, ..., M$.

Step 14. Local WLS estimation to calculate $\hat{\theta}_i$ and $\hat{\theta}_{i0}$, $i = 1, ..., M$.

Step 15. $\hat{y}(k) = \sum_{i=1}^{M} (\hat{\theta}_i^T \tilde{\mathbf{x}}(k) + \hat{\theta}_{i0}) \Phi_i(\tilde{\mathbf{z}}(k))$, $k = 1, ..., N$.

until $NRMSE(\tilde{\mathbf{y}}, \hat{\tilde{\mathbf{y}}}) < NRMSE_{max}$ or $M = M_{max}$

TABLE I
PSEUDOCODE OF SUHICLUST ALGORITHM

stretches along the diagonal of the input space. Consider the Hyperbola function

$$y = \frac{1}{0.1 + \frac{1}{2}(1 - u_1) + \frac{1}{2}(1 - u_2)} \tag{29}$$

This function shall be approximated with a normalized root-mean-squared error of less than 5 percent. With LOLIMOT, 11 local linear models were needed, while the SUHICLUST algorithm (axis-oblique partitioning strategy) could achieve the same accuracy with 5 local linear models, see Fig. 9.

Table II summarizes the results with SUHICLUST compared to LOLIMOT, FMID, ANFIS and DENFIS. It comes out that SUHICLUST shows the best modeling results. In order to achieve a NRMSE below 0.05 SUHICLUST needs for the Mars-benchmark only 10 and in the case of the Hyperbola-benchmark with two inputs 5 local linear models whereas ANFIS needs 16 and 9 local models in both examples. The second best performing method is FMID with 13 and 9 local models, respectively. With the default configuration, DENFIS produced 90 local models in both cases which is not adequate in this comparison.

B. Analytical example II

In this example, the data set is enlarged up to a five-dimensional input space for the Hyperbola benchmark. The generalization of this function

$$y = \frac{1}{0.1 + \frac{1}{p}\sum_{i=1}^{p}(1 - u_i)} \tag{30}$$

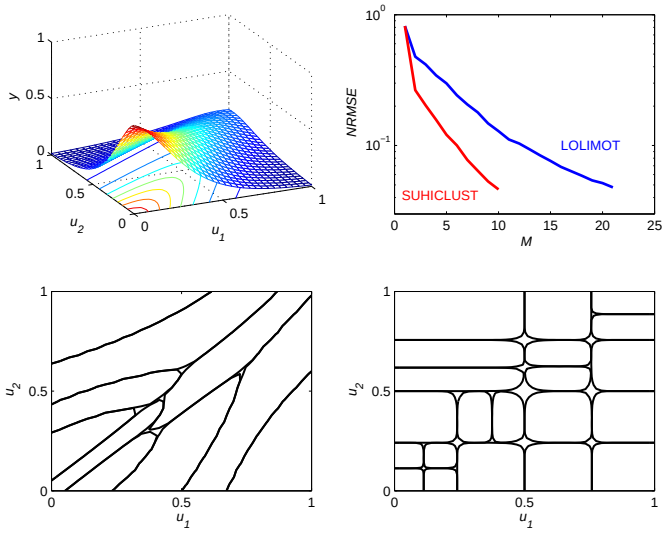

Fig. 8. Top left: Process data. Top right: Convergence behavior of LOLIMOT, SUHICLUST and ANFIS. Bottom left: The contours of SUHICLUST partitions drawn at a normalized membership degree of 0.5. Bottom right: The partitions generated with LOLIMOT drawn at a normalized membership degree of 0.5.

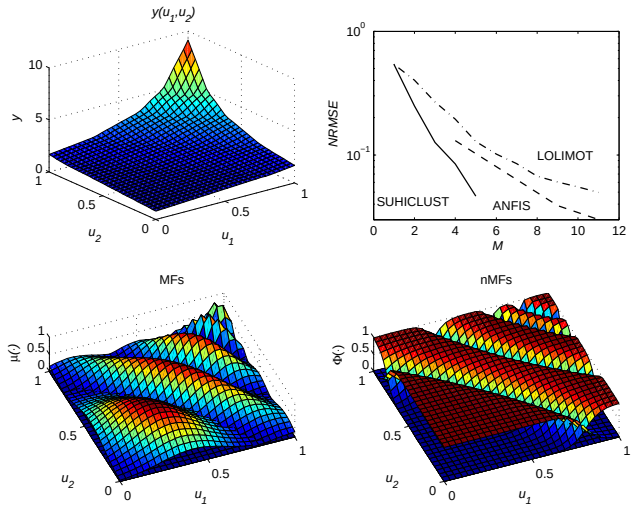

Fig. 9. Top left: Process (light) and model (solid) output with SUHICLUST. Top right: The convergence behavior for LOLIMOT and SUHICLUST. The membership functions (bottom left) and normalized membership functions (bottom right) of the models constructed by the proposed axes-oblique algorithm (SUHICLUST).

TABLE II
ERRORS FOR MODELING THE MARS- (28) AND HYPERBOLA-FUNCTION (29)

Method	Mars (NRMSE / # LMs)	Hyperbola (NRMSE / # LMs)
SUHICLUST	0.0462 / 10	0.0412 / 5
LOLIMOT	0.0479 / 21	0.0498 / 11
FMID	0.0473 / 13	0.0499 / 9
DENFIS	0.0679 / 90	0.0930 / 90
ANFIS	0.0411 / 16	0.0390 / 9

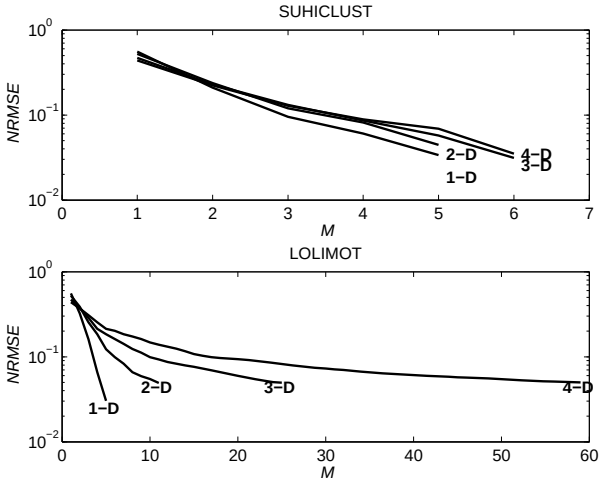

Fig. 10. The convergence behavior for 1-, 2-, 3-, and 4-dimensional approximation problem with the SUHICLUST (upper) and the LOLIMOT (lower) partitioning strategy.

shows that the hierarchical axes-orthogonal strategy roughly increases the needed number of local models exponentially with the input-space dimensionality. In the meantime, the SUHICLUST strategy is independent of the input-space dimension and requires only 5 to 6 local linear models to reach an error measure of 5 percent; the axes-orthogonal strategy requires 5, 11, 25, and 59 local linear models for the 1-, 2-, 3-, and 4-dimensional cases, see Fig. 10. The dimension of the input space equals to the parameter p in Eq. 30. Table III shows the number of models and the computation time up to the 6-dimensional (Eq. 30; $p = 2, ..., 6$) case, for the application of LOLIMOT, SUHICLUST and Gustafson-Kessel product-space clustering. The results of LOLIMOT and SUHICLUST are generated by running each algorithm only a single time. For a comparison, the Gustafson-Kessel clustering procedure is called ten times and the best result is selected. In each run the number of clusters is iteratively increased until the error measure is achieved. The cluster centers are randomly initialized. The ANFIS algorithm breaks down already in the case with the three-dimensional input space.

C. Pharmaceutical data set

In this section the absorption spectra of the protonation equilibria of Silychristin with a dependence on different wavelengths $u_1[nm]$ and pH ($u_2[pH]$) at a temperature of 25^oC is discussed. The problem of absorption spectra is presented in detail in [40] where the data set is simulated without noise. In this investigation, noise is added to the data ($\sigma_n = 0.01$). The results of modeling with SUHICLUST are shown in Fig. 11 with the data, the contour diagram of the normalized membership functions for $\Phi_i(\cdot) > 0.75$ with centers of the clusters (+), the 3D error surface and the 3D presentation of the normalized membership functions.

The comparison of the investigated methods is given by the variance accounted for (*VAF*) criterion which is defined as

TABLE III
RESULTS OF THE DEMONSTRATION EXAMPLE*

input dimension	1D	2D	3D	4D	5D	6D
# data points	15k	14.9k	15.6k	14.6k	16.8k	15.6k
LOLIMOT	0.3s / 5 LM	1.4s / 11 LM	11.6s / 25 LM	85.6s / 59 LM	494.2s / 114 LM	1084.7s / 159 LM
SUHICLUST	35.5s / 5 LM	33.2s / 5 LM	51.1s / 6 LM	48.0s / 6 LM	65.1s / 6 LM	60.5s / 6 LM
Product-space Clustering	6.8s / 5 LM	37.7s / 7 LM	56.7s / 9 LM	43.1s / 11 LM	75.0s / 14 LM	102.5s / 14 LM

* Computed with Intel Core 2 processor, 1.83 GHz, 2 GB RAM

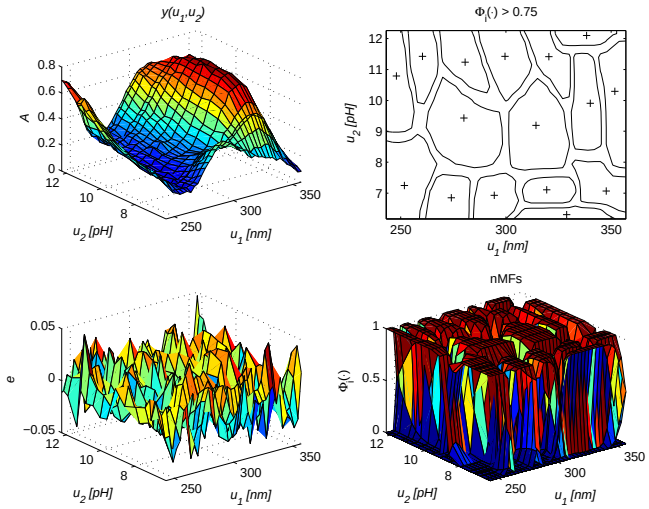

Fig. 11. Data set of absorbtion spectra of Silychristin approximated by SUHICLUST. Top left: 3D data set. Top right: Contour diagram of the normalized membership functions for $\Phi_i(\cdot) > 0.75$ with centers of the clusters (+). Bottom left: Approximation error. Bottom right: Normalized membership functions $\Phi_i(\cdot)$.

follows:

$$VAF(\tilde{\mathbf{y}},\hat{\tilde{\mathbf{y}}}) = 100\left(1 - \frac{\sum_{k=1}^{N}(\tilde{y}(k)-\hat{\tilde{y}}(k))^2}{\sum_{i=1}^{N}(\tilde{y}(i)-\bar{y})^2}\right)$$
$$\bar{y} = \frac{1}{N}\sum_{k=1}^{N}\tilde{y}(k) \qquad (31)$$

The relation between the *NRMSE* and *VAF* criteria is the following $VAF = 100 \cdot (1 - NRMSE^2)$. The ideal case of modeling is given by $NRMSE = 0$ or $VAF = 100$.

With SUHICLUST 16 rules were generated in order to get the VAF value $VAF(\mathbf{y},\hat{\mathbf{y}}) = 99.07$. The VAF measures for each local model show that the modeling with SUHICLUST gives a very good local approximation. In order to calculate the local VAF measure, the data is locally weighted with the corresponding validity function

$$VAF_i(\mathbf{y}_i,\hat{\mathbf{y}}_i) = $$
$$= 100\left(1 - \frac{\sum_{k=1}^{N}(y(k)\Phi_i(\mathbf{u}(k)) - \hat{y}_i(k)\Phi_i(\mathbf{u}(k)))^2}{\sum_{k=1}^{N}(y(k)\Phi_i(\mathbf{u}(k)) - \bar{y}_i)^2}\right) \qquad (32)$$
$$\bar{y}_i = \frac{1}{N}\sum_{k=1}^{N} y(k)\Phi_i(\mathbf{u}(k))$$

The corresponding local VAF measures for all 16 local models are calculated as 99.2813, 99.6805, 99.6395, 99.2821, 99.1864, 99.6094, 99.6318, 99.8836, 99.4744, 99.8028, 99.9438, 99.8680, 99.9205, 99.8939, 99.7971 and 99.8626.

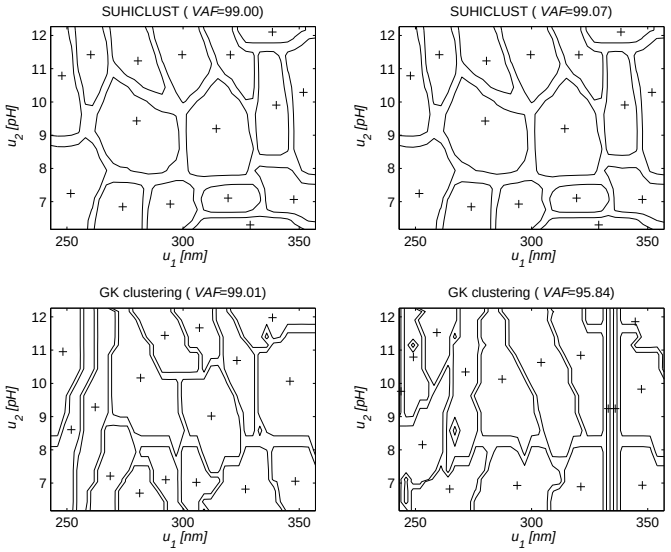

Fig. 12. Partitioning for the absorption spectra data set of silychristin approximated by SUHICLUST (top row) and GK clustering (bottom row), with $\Phi_i(\cdot) > 0.75$. With both algorithms two runs were performed. In contrast to GK clustering, SUHICLUST is deterministic and delivers almost identical results in both runs.

The most important advantage of the proposed SUHICLUST algorithm, besides the accuracy of the global and local models, is the fact that the algorithm always converges practically to the same results. This is shown in Fig. 12, where two different runs for the same problem are made. It is shown that the centers, the contours and the VAF measures are almost the same. The same experiment is conducted using the GK clustering algorithm by fixing the number of clusters to 16. It is shown that due to the random initialization the results of

TABLE IV
RESULTS FOR MODELING THE PHARMACEUTICAL PROCESS

Method	# LMs	Error (NRMSE)
SUHICLUST	10	0.0822
LOLIMOT	24	0.0984
FMID	15	0.0985
DENFIS	29	0.1536
ANFIS	16	0.0899

TABLE V
RESULTS FOR MODELING THE ENGINE CHARACTERISTIC MAP

Method	# LMs	Error (NRMSE)
SUHICLUST	8	0.0602
LOLIMOT	15	0.0694
FMID	15	0.0648
LOLIMOT GP	11	0.0691
LOLIMOT GP-V	14	0.0685
DENFIS	22	0.0734
ANFIS	25	0.0452

the algorithm become very different. In particular, the resulting centers are quite different and also the VAF measures show an initialization-dependent algorithm. The absorption data set is also modeled with ANFIS. In this example *genfis1* with Bell functions is used. Each input is divided into 4 subspaces which delivers 16 local models in the two-dimensional product-space. The results of the modeling are given as follows: The overall VAF measure shows a very good result ($VAF = 99.01$), but the corresponding local VAF measures show that the corresponding local models do not estimate the local behavior adequately. The corresponding local VAF measures in this case are, for all 16 local models, calculated as 95.9995, 94.5331, 92.9437, 96.8007, 94.1893, 96.9742, 87.3911, 85.0560, 91.6943, 81.7055, 93.8893, 96.9630, 90.9209, 81.0264, 63.8170 and 79.0808. These values show that the overall approximation applies very well, but the fit of the local models is not appropriate. This means that the approximation is distributed between overlapping membership functions. The procedure is repeated with the *genfis2* function in Matlab, which involves subtractive fuzzy clustering. In this case the algorithm results in 149 rules, which is again a number that is not comparable with other algorithms.

For the sake of completeness, Table IV shows the modeling results of the five investigated algorithms. The goal was to achieve an error less than 0.1 with as less local models as possible. Again, SUHICLUST shows the best results.

D. Modeling an engine characteristic map

Another application example is the modeling of an engine characteristic map. The investigated highly non-linear process consists of 433 samples where the engine torque in $[Nm]$ is measured as a function of the engine speed in $[rpm]$ and the injection mass in $[mg]$. In order to compare the results the same data set is used like proposed in [41] and [42]. In these citations an extension of LOLIMOT with the application of genetic programming, LOLIMOT GP, is proposed. Furthermore, the algorithm was implemented with a variable split ratio, LOLIMOT GP-V, i.e. the partitioning is more flexible than with the classical LOLIMOT approach.

In this experiment SUHICLUST is compared to LOLIMOT, LOLIMOT GP and LOLIMOT GP-V. Additionally, the algorithms ANFIS [21] and DENFIS [44] are considered for the comparison. Goal of the modeling with SUHICLUST, ANFIS and DENFIS was to reach at least the same error (NRMSE) as LOLIMOT GP-V which was the best performing algorithm in [41]. The results in Table V show clearly the advantage of SUHICLUST. All results were visually monitored with respect to overfitting.

The best result with LOLIMOT GP-V was a NRMSE of 0.0685 with the usage of 14 local linear models (LMs). SUHICLUST needs only 8 local linear models to achieve the same modeling accuracy. ANFIS reaches with 5 rules per input axis a slightly better error value, but then the model consists of 25 local linear models. Obviously, this leads to 75 consequent parameters whereas SUHICLUST has only 24 local model parameters. The results with DENFIS are comparable to the results with ANFIS. Unfortunately, it is not possible to preset the number of local models with DENFIS. Thats the reason why the error is slightly worser than with LOLIMOT GP-V.

E. Comparison on Automobile MPG benchmark

In [5] the proposed, modified Gath-Geva (GG) Clustering algorithm is tested on the well known Automobile MPG benchmark data set that is available from the UCI Repository of Machine Learning Databases (*http://archive.ics.uci.edu/ml/*). Furthermore, they used the FMID toolbox [6] and ANFIS for the comparison.

As proposed in [5], also the reduced data set with 392 samples is used in this comparison. The following input variables were chosen for the prediction problem: u_1: displacement; u_2: horsepower; u_3: weight; u_4: acceleration; and u_5: model year. Goal is to predict the fuel consumption of an automobile.

In order to compare the results of SUHICLUST with the results in [5] the data set is randomly split in 50% for training and 50% for testing as it was done in the cited paper. Then the FMID model is trained with different train and test sets a several times until we got a similar solution like shown in [5]. The chosen data sets for training and testing lead to an RMS error of 2.76 for training and 3.02 for testing. In [5] the results were 2.67 and 2.95, respectively. As shown in Table VI, SUHICLUST offers, like the FMID toolbox and the modified GG algorithm, very good generalization properties. The RMSE values of SUHICLUST with 4 rules are almost the same like the results with the modified GG culstering.

Next, the extrapolation behavior is tested on the same problem as stated in [45], as it is performed in [5]. The best SUHICLUST result is obtained with 5 local models. For training an RMS error of 2.82 and 2.83 for testing is observed. The best solution of the modified GG algorithm in [5] is 2.77 and 2.95 which is almost the same performance.

In [5] it is pointed out that the ANFIS model in [45] has severe problems with sparse data regions and, therefore, the

TABLE VI
PERFORMANCE COMPARISON ON THE AUTOMOBILE MPG BENCHMARK

Method	Training (RMSE)	Testing (RMSE)
SUHICLUST	2.75	2.83
Mod. GG [5]	2.72	2.85
FMID [5]	2.67	2.95
ANFIS [5]	1.96	91.35

ANFIS model spuriously estimates higher MPG for heavy cars. Figure 13 illustrates the prediction surface of the underlying SUHICLUST model. It shows that the extrapolation behavior of the SUHICLUST model suits to the data. In consideration of this figure, it should be mentioned that an interpretation of the resulting rules is only possible, if the partitioning is analyzed for a certain cut in the input space. But this is the price to be paid for the higher amount of model quality.

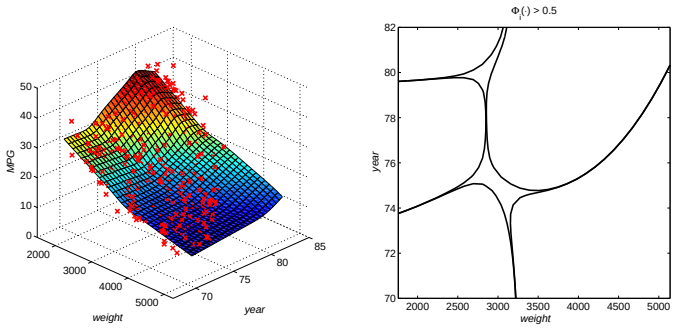

Fig. 13. Prediction surface (left) compared to all data samples and partitioning (right) of Automobile MPG model.

VI. CONCLUSIONS

This contribution introduces the algorithm SUHICLUST which stands for SUpervised HIerarchical CLUSTering and provides a partitioning algorithm that combines a supervised, heuristical tree-construction algorithm with unsupervised, highly flexible product-space clustering. Key feature of SUHICLUST is its easy usability. The algorithm delivers reliable, deterministic and well generalizing results with a parsimonious partitioning. Easy-to-use means that the algorithm produces reasonable results while the user only has to set either a threshold for the maximum number of local models or a value for the maximum allowed global model error. The performance of this new approach is shown in comparison with state-of-the-art algorithms and on both analytical illustration examples and benchmark data sets. The investigated benchmark results with SUHICLUST outperform existing results from the literature.

ACKNOWLEDGMENT

The authors would like to thank the German Research Foundation (*Deutsche Forschungsgemeinschaft* (DFG), project code NE 656/3-2) and the *Alexander von Humboldt Stiftung* (Grant SLO-1133479 STP).

REFERENCES

[1] M. Sugeno and G.T. Kang. "Structure identification of fuzzy model". In *Fuzzy Sets and Systems*, 28(1), pp. 15–33, 1988.
[2] J. Abonyi and R. Babuška. "Local and global identification and interpretation of parameters in Takagi-Sugeno fuzzy models," In *Proceedings of IEEE Conference on Fuzzy Systems*, San Antonio, TX, pp. 835–840, 2000.
[3] T.A. Johansen, R. Babuška "Multiobjective Identification of Takagi-Sugeno Fuzzy Models". In *IEEE Transaction on Fuzzy Systems*, 11(6), pp. 847–860, 2003.
[4] J. Yen, L. Wang, and C. W. Gillespie, "Improving the interpretability of TSK fuzzy models by combining global learning and local learning," In *IEEE Transaction on Fuzzy Systems*, 6, pp. 530–537, Aug. 1998.
[5] J. Abonyi, R. Babuška and F. Szeifert. "Modified Gath-Geva fuzzy clustering for identification of Takagi-Sugeno fuzzy models". In *IEEE Transactions on Systems, Man, and Cybernetics, Part B* 32, 5, pp. 612–621, 2002.
[6] R. Babuška. "Fuzzy Modeling for Control". *Kluwer Academic Publishers*, Boston, 1998.
[7] R. Babuška and H.B. Verbruggen. "An overview of fuzzy modeling for control". In *Control Engineering Practice*, 4(11), pp. 1593–1606, 1996.
[8] R. Babuška and J.A. Roubos and H.B. Verbruggen. "Identification of MIMO systems by input-output TS fuzzy models". In *Proceedings of IEEE World Congress on Computational Intelligence*, pp. 657–662, 1998.
[9] Sugeno M, Yasukawa T. "A fuzzy-logic-based approach to qualitative modeling". In *IEEE Transaction on Fuzzy Systems*, 1, pp. 7–31, 1993.
[10] Kim E, Park M, Ji S, Park M. "A transformed input-domain approach to fuzzy modeling". In *IEEE Transaction on Fuzzy Systems*, 6, pp. 596–604, 1998.
[11] Q. Gan and C. J. Harris. "Fuzzy local linearization and logic basis function expension in nonlinear system modeling," In *IEEE Transaction on Systems, Man, and Cybernetics - Part B*, 29, 4, pp. 559–565, 1999.
[12] Y. Shi, R. Eberhart, and Y. Chen. "Implementation of evolutionary fuzzy systems," In *IEEE Transaction on Fuzzy Systems*, 7, 2, pp. 109–119, 1999.
[13] C. K. Lin and S.D. Wang. "Fuzzy system identification using an adaptive learning rule with terminal attractors," In *Journal on Fuzzy Sets and Systems*, pp. 343–352, 1999.
[14] S. J. Kang, C. H. Woo, H. S. Hwang, and K. B. Woo. "Evolutionary design of fuzzy rule base for nonlinear system modeling and control," In *IEEE Transaction on Fuzzy Systems*, 8, 1, pp. 37–45, 2000.
[15] Emani MR, Turksen IB, Goldenberg AA. "Development of a systematic methodology of fuzzy logic modeling". In *IEEE Transaction on Fuzzy Systems*, 6(3), pp. 46–61, 1998.
[16] B. Wu and X. Yu, "Fuzzy modelling and identification with genetic algorithm based learning," In *Fuzzy Sets and Systems*, 113, pp. 352–365, 2000.
[17] Y.P. Huang and S. -F. Wang, "Designing a fuzzy model by adaptive macroevolution genetic algorithms," In *Fuzzy Sets and Systems*, 113, pp. 367–379, 2000.
[18] M. Denna, G. Mauri, and A. M. Zanaboni, "Learning fuzzy rules with tabu search-an application to control," In *IEEE Transaction on Fuzzy Systems*, 7, 2, pp. 295–318, 1999.
[19] L. X.Wang, "Analysis and design of hierarchical fuzzy systems," In *IEEE Transaction on Fuzzy Systems*, 7, 5, pp. 617–624, 1999.
[20] O. Huwendiek and W. Brockmann. "Function approximation with decomposed fuzzy systems," *Fuzzy Sets and Systems*, 101, pp. 273–286, 1999.
[21] JSR. Jang "ANFIS: Adaptive-network-based fuzzy inference systems". In *IEEE Transaction Systems, Man and Cybernetics*, 23, pp. 665–685, 1993.
[22] Sun CT. "Rule-based structure identification in an adaptive network based fuzzy inference system". In *IEEE Transaction on Fuzzy Systems*, 3, pp. 64-73, 1994.
[23] Lee SJ, Ouyang CS. "A neuro-fuzzy system modeling with selfconstructing rule generation and hybrid SVD-based learning. In *IEEE Transaction on Fuzzy Systems*, 11, pp. 341–53, 2003.
[24] Chiu S. "Fuzzy model identification based on cluster estimation". In *Journal of Intelligent and Fuzzy Systems*, 2, pp. 267–278, 1994.
[25] L. Breiman, C.J. Stone J.H. Friedman R. and R. Olshen. *Classification and Regression Trees*". Chapman & Hall, New York, 1984.
[26] L. Breiman. "Hinging hyperplanes for regression, classification, and function approximation". In *IEEE Transactions on Information Theory*, 39(3), pp. 999–1013, 1993.

[27] P. Pucar and M. Millnert. "Smooth hinging hyperplanes: A alternative to neural nets". In *European Control Conference (ECC)*, pp. 1173–1178, Rome, Italy, 1995.

[28] O. Nelles, S. Sinsel, and R. Isermann. "Local basis function networks for identification of a turbocharger". In *IEEE UKACC International Conference on Control*, pp. 7–12, Exeter, UK, September 1996.

[29] O. Nelles. *"Nonlinear System Identification"*. Springer, Berlin, Germany, 2001.

[30] O. Nelles. "Axes-oblique partitioning strategies for local model networks". In *International Symposium on Intelligent Control (ISIC)*, Munich, Germany, October 2006.

[31] T.A. Johansen, R. Shorten, and R. Murray-Smith, "On the interpretation and identification of Takagi-Sugeno fuzzy models," In *IEEE Transection on Fuzzy Systems*, 8, pp. 297-313, 2000.

[32] T. Takagi and M. Sugeno. "Fuzzy identification of systems and its application to modeling and control," In *IEEE Transaction on Systems, Man, and Cybernetics*, 15, 1, pp. 116–132, 1985.

[33] D.E. Gustafson and W.C. Kessel. "Fuzzy clustering with a fuzzy covariance matrix". In *IEEE Conference and Decesion and Control*, pp. 761–766, 1979.

[34] I. Gath and A.B. Geva. "Unsupervised optimal fuzzy clustering". In *IEEE Transaction Pattern Analysis and Machine Intelligence*, 11(7), pp. 773–781, 1989.

[35] T.A. Johansen. "Identification of non-linear system structure and parameters using regime decomposition". In *Automatica*, 31(2), pp. 321–326, 1995.

[36] S. Ernst. "Hinging hyperplane trees for approximation and identification". In *IEEE Conference on Decision and Control (CDC)*, pp. 1261–1277, 1998.

[37] R. Murray-Smith and T. A. Johansen, "Local learning in local model networks," In *Multiple Model Approaches to Modeling and Control*, Taylor and Francis, 1997.

[38] R. Murray-Smith. *"A Local Model Network Approach to Nonlinear Modeling"*. *PhD thesis, University of Strathclyde*, Strathclyde, UK, 1994.

[39] J.H. Friedman. "Multivariate adaptive regression splines (with discussion)". In *The Annals of Statistics*, 19(1), pp. 1–141, 1991.

[40] M. Melouna, D. Burkonova, T. Syrovy, A. Vrana. "Thermodynamic dissociation constants of silychristin, silybin, silydianin and mycophenolate by the regression analysis of spectrophotometric data". In *Analytica Chimica Acta*, 486, pp. 125–141, 2003.

[41] F. Hoffmann and O. Nelles. "Genetic programming for model selection of TSK-fuzzy systems". In *Information Sciences*, 136 (1–4), pp. 7–28, 2001.

[42] F. Hoffmann and O. Nelles. "Structure identification of TSK-fuzzy systems using genetic programming". In *Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, 2000.

[43] F. Hoffmann and D. Schauten and S. H"olemann "Incremental Evolutionary Design of TSK Fuzzy Controllers". In *IEEE Transactions on Fuzzy Systems*, 15(4), pp. 563–577. August 2007

[44] N.K. Kasabov and Q. Song. "DENFIS: dynamic evolving neural-fuzzy inference system and its application for time-series prediction". In *IEEE Transactions on Fuzzy Systems*, 10(2), pp. 144–154, 2002.

[45] J.S.R. Jang. "Input selection for ANFIS learning". In *Procedings IEEE ICFS*, pp. 1493–1499, New Orleans, LA, 1996.

[46] , P.P. Angelov and D.P. Filev. "An approach to online identification of Takagi-Sugeno fuzzy models". In *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 34(1), pp. 484–498, 2004.

[47] E.D. Lughofer. "Flexfis: A robust incremental learning approach for evolving Takagi–Sugeno fuzzy models". In *IEEE Transactions on Fuzzy Systems*, 16(6), pp. 1393–1410, 2008.

[48] C.F. Juang and C.T. Lin. "An online self-constructing neural fuzzy inference network and its applications". In *IEEE Transactions on Fuzzy Systems*, 6(1), pp. 12–32, 1998.

[49] J. de Jesús Rubio. "SOFMLS: online self-organizing fuzzy modified least-squares network". In *IEEE Transactions on Fuzzy Systems*, 17(6), pp. 1296–1309, 2009.

[50] S. Töpfer. "Approximation of Nonlinear Processes with Hinging Hyperplane Trees".
newblock In *at-Automatisierungstechnik, Oldenburg*, 50(4), 2002.

[51] G. Tsekouras and H. Sarimveis and E. Kavakli and G. Bafas. "A hierarchical fuzzy-clustering approach to fuzzy modeling". In *Fuzzy Sets and Systems*, 150(2), pp. 245–266, 2005.

[52] Y. Jin. "Fuzzy modeling of high-dimensional systems: complexity reduction and interpretability improvement". In *IEEE Transactions on Fuzzy Systems*, 8(2), pp. 212–221, 2000.

[53] J. A. Roubos and M. Setnes "Compact fuzzy models through complexity reduction and evolutionary optimization". In *Procedings IEEE ICFS*, pp. 762?767, 2000.

[54] B. Tutmez and Z. Hatipoglu. "Comparing two data driven interpolation methods for modeling nitrate distribution in aquifer". In *Ecological Informatics*, 5(4), pp. 311–315, 2010.

[55] J. Madar and J. Abonyi and F. Szeifert. "Genetic Programming for the Identification of Nonlinear Input?Output Models". In *Ind. Eng. Chem. Res.*, 44(9), pp. 3178?-3186, 2005.

Benjamin Hartmann received the M.S. degree in Mechanical Engineering in 2007 from the University of Siegen, Germany. He is presently research assistant and works with Prof. Nelles at the Institute of Automatic Control and Mechatronics in the Department of Mechanical Engineering at the University of Siegen, pursuing his Ph.D. degree in automatic control engineering. His current research interests are experimental nonlinear static and dynamic modeling and design of experiments.

Oliver Bänfer received the M.Sc. degree in Electrical Engineering in 2005 from the University of Siegen, Germany. He is currently concluding his Ph.D. degree in learning procedure for neural networks using local polynomial models and subset selection techniques at the University of Siegen, Germany. From August 2011 he will work at Mercedes-Benz Technology Center in Sindelfingen, Germany.

Oliver Nelles received the M.Sc. degree in Electrical Engineering in 1993 from the Technical University in Darmstadt, Germany. Between 1994 and 2000 he worked with Prof. Isermann at the Institute of Automatic Control and obtained his Ph.D. in 2000. With a DAAD scholarship he did research as a postdoc with Prof. Tomizuka in the Mechanical Engineering Department at UC Berkeley. 2000-2004 he was Group Leader in the development of transmissions with SiemensVDO Automotive, Regensburg, Germany. Currently, he is a Professor for Automatic Control and Mechatronics in the Department of Mechanical Engineering at the University of Siegen, Germany. Dr. Nelles is author of the book "Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models", Springer, 2001. His research focus is on experimental nonlinear static and dynamic modeling, data mining, and design of experiments.

Anton Sodja received his B.S. degree in electrical engineering from the University of Ljubljana in 2007. He is currently pursuing the Ph.D. degree in object-oriented modeling methodologies using modeling language Modelica at the Laboratory of Modelling, Simulation and Control, University of Ljubljana. His current research interests include model reduction techniques and object-oriented models debugging.

Luka Teslić received the B.Sc. degree in electrical engineering from the University of Ljubljana, Slovenia, in 2006. He is currently pursuing a Ph.D. degree in mobile robotics at the University of Ljubljana. His current research interests include mobile robot localization and map building and fuzzy system identification.

Igor Škrjanc received the B.Sc., the M.Sc. and the Ph.D. degrees in electrical engineering, from the Faculty of Electrical and Computer Engineering, University of Ljubljana, Slovenia, in 1988, 1991 and 1996, respectively. His main research interests are intelligent, predictive control systems and autonomous mobile systems. In 2007 he received the highest research award of the University of Ljubljana, Faculty of Electrical Engineering, and in 2008, the highest award of the Republic of Slovenia for Scientific and Research Achievements, Zois award for outstanding research results in the field of intelligent control. He also received the Humboldt Research Fellowship for Experienced Researchers for the period between 2009-2011. Currently, he is a Professor for Automatic Control at the Faculty of Electrical Engineering and the head of the research program Modelling, Simulation and Control.