Nonlinear System Identification by Gustafson–Kessel Fuzzy Clustering and Supervised Local Model Network Learning for the Drug Absorption Spectra Process

Luka Teslić, Benjamin Hartmann, Oliver Nelles, and Igor Škrjanc

***Abstract*— This paper deals with the problem of fuzzy nonlinear model identification in the framework of a local model network (LMN). A new iterative identification approach is proposed, where supervised and unsupervised learning are combined to optimize the structure of the LMN. For the purpose of fitting the cluster-centers to the process nonlinearity, the Gustafsson–Kessel (GK) fuzzy clustering, i.e., unsupervised learning, is applied. In combination with the LMN learning procedure, a new incremental method to define the number and the initial locations of the cluster centers for the GK clustering algorithm is proposed. Each data cluster corresponds to a local region of the process and is modeled with a local linear model. Since the validity functions are calculated from the fuzzy covariance matrices of the clusters, they are highly adaptable and thus the process can be described with a very sparse amount of local models, i.e., with a parsimonious LMN model. The proposed method for constructing the LMN is finally tested on a drug absorption spectral process and compared to two other methods, namely, Lolimot and Hilomot. The comparison between the experimental results when using each method shows the usefulness of the proposed identification algorithm.**

***Index Terms*— Gustafson–Kessel fuzzy clustering, local model networks, nonlinear system identification, (un)supervised learning.**

I. Introduction

THE local model network (LMN) approach for the purpose of identifying nonlinear static and dynamic processes and model-based control has generated a great deal of research interest. LMN models describe the process nonlinearity with a definite number of local submodels. Architectures based on the interpolation of local models have, in the past few decades, attracted more and more interest for the purpose of modeling nonlinear dynamic systems and also for approximating static functions. Local *linear* models allow the transfer of many parts of the mature linear theory to the nonlinear world. The boom in local linear model structures is also a consequence of recent advances in the area of convex optimization and the development of efficient algorithms for the solution of linear matrix inequalities.

According to [1], a LMN approach has the potential to allow conventional modeling techniques and established linear control methods to be applied within an inherently nonlinear framework, with the advantage of transparency of operation and the capability to readily comprise *a priori* knowledge. In [2]–[9], various domains where the LMNs can be utilized are provided. Product-space clustering algorithms, such as Gath–Geva [10], and heuristic tree-construction algorithms, such as Cart [11] and Lolimot [12], are two of the most popular partitioning strategies for defining the validity regions of local models.

Neural networks (NNs), the radial basis function network (RBFN), and the Gaussian process (GP) model are also very often applied for the purpose of system modeling. A major limitation of the NNs is the time-consuming training process [13]. The main drawbacks of the NNs [14], which are used to model nonlinear systems for control, are the troublesome "curse of dimensionality" and the lack of transparency. According to [15], the main advantage of the GP model in comparison with the NNs is an estimation of the model uncertainty. But the GP model is computationally demanding and nontransparent, and to cope with these two difficulties, a local linear GP model network, where the GP prior approach is combined with the LMN approach, has been proposed [15]. In [16], a hybrid linear/nonlinear training algorithm for feedforward NNs is presented, which is especially useful for the LMN architecture with large linear to nonlinear parameter ratio. A major advantage of the LMN over the RBFN [14] is a considerable reduction in the number of local models and the increased transparency of the model.

Network optimization, where the optimal number of local linear models, their parameters, and the validity functions must be found, is the core problem to be solved in the process of system modeling with the LMN. Often, the goal is to find a parsimonious model of the system. A parsimonious network structure in the framework of fuzzy local linearization modeling was achieved in [17] by applying a modified adaptive spline modeling algorithm for defining the membership

Manuscript received October 6, 2010; revised September 18, 2011; accepted September 20, 2011.

L. Teslić and I. Škrjanc are with the Department of Electrical Engineering, University of Ljubljana, Ljubljana SI-1000, Slovenia (e-mail: luka.teslic@fe.uni-lj.si; igor.skrjanc@fe.uni-lj.si).

B. Hartmann and O. Nelles are with the Department of Mechanical Engineering, University of Siegen, Siegen D-57068, Germany (e-mail: benjamin.hartmann@uni-siegen.de; oliver.nelles@uni-siegen.de).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNN.2011.2170093

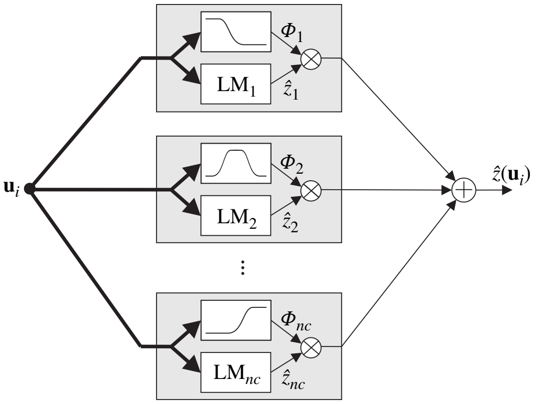


Fig. 1. Principle of LMNs.

functions. In [18], a parsimonious RBFN is produced by simultaneously determining the network structure and optimizing the parameters.

In this paper, fuzzy clustering forms the basis for optimizing the structure of the LMN. In most of the fuzzy clustering algorithms, an important problem is that the optimum number of clusters [19] and their initial values (e.g., cluster centers) must be estimated. In [20], each data point is considered as a potential cluster center at the beginning of the algorithm. The optimum number of clusters can be determined, e.g., by applying cluster validity index [21]. A wide variety of fuzzy clustering algorithms can be found in the literature. In [22], some popular fuzzy clustering methods are reviewed. One of the most often used method is fuzzy C-means [23], [24]. Fuzzy probabilistic C-means (PCM) clustering is derived from the PCM method [25] and two other approaches are fuzzy clustering applying t-distributions [26] and fuzzy compactness and separation [27]. In [28], the comparison between kernel-based fuzzy clustering and fuzzy clustering is given. In this paper, the purpose of the clustering is to find fuzzy subsets of data from a large set of process data, where each subset corresponds to a certain local linear region of the nonlinear process. The ability to find local hyperplanes in the product space makes the Gustafson–Kessel (GK) fuzzy clustering [29] very convenient for this purpose. Each cluster obtained with the GK algorithm therefore corresponds to a certain local linear model of the LMN. The basic GK fuzzy clustering solution is very sensitive to the choice of the number and the initial locations of cluster centers. This issue is addressed in this paper by increasing the number of clusters and simultaneously determining the initial locations of the cluster centers, which arise from splitting the worst modeled cluster. The number of clusters increases until a predefined level of the LMN model accuracy is achieved.

A fuzzy nonlinear model-identification technique in the framework of a LMN is proposed in this paper. Here, supervised and unsupervised learning are combined in order to optimize the structure of the LMN. The main contributions of this paper are the following.

1) A new iterative and incremental approach for defining the number and the initial locations of cluster centers for the GK fuzzy clustering method is proposed in connection to the LMN learning procedure.
2) Highly adaptable validity functions are obtained by applying the GK fuzzy clustering. The obtained validity functions adapt to the local linear regions of the nonlinear process. For this reason, the nonlinear process can be described with a very sparse amount of local models, which leads to a parsimonious LMN structure.
3) Experimental results of identifying the model of the 3-D drug absorption spectra process using the proposed method and two other methods for constructing the LMNs, i.e., Lolimot [12] and Hilomot [30], are presented and compared.

This paper is structured as follows. In Section II, a LMN model is defined and some strategies for partitioning the input space are discussed. Section III describes the initialization part, the iterative part, and the final part of the proposed fuzzy nonlinear identification technique. In Section IV, the experimental results of identifying the LMN model of the drug absorption spectra process using the proposed approach and two other methods are compared. This paper is concluded in Section V.

II. LMNs

The output $\hat{z}(\mathbf{u}_i)$ of a LMN can be calculated as the interpolation of nc local model outputs $\hat{z}_j(\mathbf{u}_i)$, $j = 1, \ldots, nc$, see Fig. 1

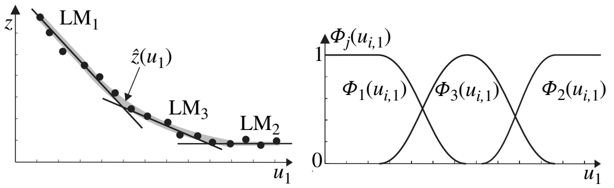
$$\hat{z}(\mathbf{u}_i) = \sum_{j=1}^{nc} \hat{z}_j(\mathbf{u}_i)\Phi_j(\mathbf{u}_i) \tag{1}$$

where $\mathbf{u}_i = [u_{i,1}, \ldots, u_{i,p}]$ denotes the ith sample or measurement of the input vector and p denotes the number of inputs to the system. $\Phi_j(\mathbf{u}_i)$ are the validity or weighting functions that describe the regions where the local models are valid. The validity functions define the contribution of each local model to the LMN output. The validity functions are smooth functions in the interval $(0, 1]$, since here a smooth transition (no switching) between the local models is desired. For a reasonable interpretation of LMNs, the contributions of all the local models must sum up to 100% everywhere in the input space. It is therefore necessary that the validity functions form a *partition of unity*

$$\sum_{j=1}^{nc} \Phi_j(\mathbf{u}_i) = 1, \quad \Phi_j(\mathbf{u}_i) > 0. \tag{2}$$

The local models can, in principle, be chosen as an arbitrary type. Since in this paper their parameters are estimated from the data, it is extremely beneficial to choose a linearly parameterized model class. Polynomials of degree 1 are used here for the local linear model structure, which is also by far the most popular choice. Fig. 2(a) shows an example of a LMN with three local linear models LM$_j$ $(j = 1, \ldots, 3)$ and the resulting superposition $\hat{z}(u_1)$. The corresponding validity functions $\Phi_j(u_{i,1})$ are shown in Fig. 2(b).

When the validity functions are determined, the parameters of the local linear models can be easily and efficiently estimated by local or global least-squares (LS) methods. The



(a) (b)

Fig. 2. LMN with three models. (a) Local models and the resulting superposition. (b) Validity functions.

weighted LS method for parameter estimation was applied in [31]. In [32], the differences and benefits of using particle swarm optimization and expectation-maximization algorithm for LMN learning are highlighted. In [33], a genetic learning approach to the optimization of LMN structure is presented.

A. Partitioning Algorithm

The main difference between all the proposed algorithms for constructing a LMN is the strategy of partitioning the input space, where the validity regions and consequently the parameters of the validity functions must be chosen. This strategy determines the basic properties of both the LMN learning procedure and the final structure of the LMN model. The aim of optimizing the structure of the LMN [14] is to find the optimal number of local linear models, their placement, and the shape and the size of the validity functions. In [1], a nonlinear optimization of the centers and widths of the validity functions was performed by minimizing the LS cost function.

Based on the heuristic tree-search algorithm of Cart [11], many similar partitioning strategies, such as Lolimot [12], have been proposed for LMNs, see also [34] and [35]. Their main idea is to incrementally subdivide the input space by axes-orthogonal cuts. By using these techniques, the undesirable normalization side effects and the extrapolation behavior can be improved in comparison with the clustering or data-based strategies [36]. Their main drawback inherently lies in the axes-orthogonal partitioning strategy: the performance of these algorithms is likely to degrade more and more with increasing dimensionality of the input space. They are therefore relatively sensitive to the curse of dimensionality.

Product-space clustering strategies focus on the product space that is jointly spanned by the inputs and the output, which means that the dataset used in the clustering procedure consists of the inputs and the output. In this paper, the properties of the GK fuzzy clustering are utilized to achieve an axes-flexible partitioning of the input space. The GK algorithm [29] is very often applied to search for hyper ellipsoids of equal or different volumes. In Fig. 3, an example with one input dimension and one output dimension is shown. The process is modeled with three fuzzy clusters ($nc = 3$), where each cluster ($j = 1, \ldots, 3$) is defined by its center $\mathbf{c}_j = [c_{j,u_1}\ c_{j,z}]$ and the fuzzy covariance matrix. The fuzzy covariance matrix of the jth fuzzy cluster $\mathbf{P}_j \in \mathbb{R}^{(p+1)\times(p+1)}$ is defined as follows:

$$\mathbf{P}_j = \sum_{i=1}^{n} \mathbf{F}_j^2(\mathbf{d}_i)\left(\mathbf{d}_i - \mathbf{c}_j\right)^T\left(\mathbf{d}_i - \mathbf{c}_j\right) \tag{3}$$

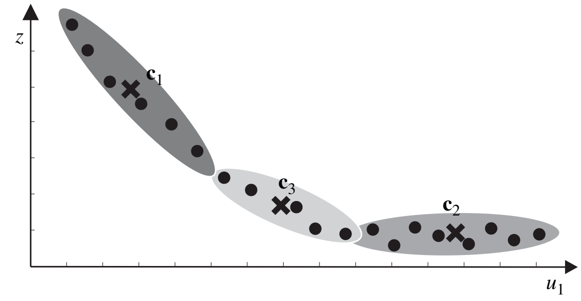


Fig. 3. Example for GK fuzzy clustering.

where $\mathbf{F}_j(\mathbf{d}_i)$ defines the normalized membership degree of the data vector $\mathbf{d}_i$ ($i = 1, \ldots, n$) to the jth cluster. Each fuzzy covariance matrix defines the directions and the variability of the corresponding cluster's data. By observing the fuzzy covariance matrix [37], it is clear that the GK algorithm is able to discover local hyperplanes in the product space by forming ellipsoids with a very small extension in one direction, see Fig. 3. The randomized initialization of the GK algorithm makes the direct use of this algorithm sometimes difficult and may prevent a unique solution in all cases. Another disadvantage is that the optimal number of clusters nc and the placement of the initial cluster centers for the GK algorithm is not known *a priori*.

Each cluster found by the GK algorithm corresponds to the local hyperplane in the product space and therefore also to the local linear model in the framework of a LMN. The size and direction of the cluster's data, i.e., the size and direction of a local linear region, is described with the corresponding fuzzy covariance matrix. The information obtained from the fuzzy covariance matrix is therefore very useful for defining the validity region of a certain local linear model.

Using the GK fuzzy clustering, the process of LMN learning is not supervised. In this paper, supervised and unsupervised learning are combined to construct the LMN for the purpose of nonlinear model identification.

III. Fuzzy Nonlinear Identification

Here, a new method for defining the number and the initial placement of the cluster centers for the GK fuzzy clustering algorithm is proposed in connection with the LMN construction. To optimize the structure of the LMN, a forward regression approach [38] is adopted. In forward regression, only two local models are assumed at the beginning of the algorithm and the number of local models then increases with respect to the complexity of the system.

In this paper, the proposed fuzzy nonlinear identification approach is illustrated in the 3-D problem space. The approach can analogously be applied in the 2-D case or generalized to more than 3-D. The original dataset representing the measurements from the static process is denoted as

$$\acute{\mathbf{D}} = \left[\acute{\mathbf{x}},\ \acute{\mathbf{y}},\ \acute{\mathbf{z}}\right] = \begin{bmatrix} \acute{x}_1 & \acute{y}_1 & \acute{z}_1 \\ \cdots & \cdots & \cdots \\ \acute{x}_n & \acute{y}_n & \acute{z}_n \end{bmatrix} \tag{4}$$

where $\acute{x}_i$ and $\acute{y}_i$ $(i = 1, \ldots, n)$ represent the input-space data and $\acute{z}_i$ represent the output data. n denotes the number of process measurements. A LMN is then defined as

$$\hat{z}(\acute{\mathbf{u}}_i) = \sum_{j=1}^{nc} \hat{z}_{i,j}(\acute{\mathbf{u}}_i)\acute{\Phi}_j(\acute{\mathbf{u}}_i), \qquad i = 1, \ldots, n$$

$$\hat{z}_{i,j}(\acute{\mathbf{u}}_i) = \hat{a}_j \acute{x}_i + \hat{b}_j \acute{y}_i + \hat{c}_j, \qquad \acute{\mathbf{u}}_i = (\acute{x}_i,\ \acute{y}_i) \tag{5}$$

where $\acute{\mathbf{u}}_i = (\acute{x}_i,\ \acute{y}_i)$, $i = 1, \ldots, n$ are the input-space data. $\hat{a}_j$, $\hat{b}_j$, and $\hat{c}_j$ are the optimal parameters of the jth local linear model (plane), and $\hat{z}_{i,j}(\acute{\mathbf{u}}_i)$ is the output of the jth local linear model. $\hat{z}(\acute{\mathbf{u}}_i)$ is the overall output of the LMN.

A. Normalizing the Data

To deal with various sets of process data $\acute{\mathbf{D}}$ (4), which correspond to the different orders of magnitude of the output-space data $\acute{\mathbf{z}}$ and the input-space data $\acute{\mathbf{x}}$ and $\acute{\mathbf{y}}$, the data $\acute{\mathbf{D}}$ are first centered and normalized. The fuzzy nonlinear identification is then performed on the basis of the newly defined dataset $\mathbf{D}$ (6). The result of the identification is recalculated to correspond to the original data set $\acute{\mathbf{D}}$ and, finally, local linear models describing the nonlinear static process are found.

By subtracting the mean m_x (or m_y, m_z) (7) from the elements of the vector $\acute{\mathbf{x}}$ (or $\acute{\mathbf{y}}$ and $\acute{\mathbf{z}}$) and by dividing the obtained differences by the maximal absolute difference ρ_x (or ρ_y and ρ_z) (8), the original data $\acute{\mathbf{D}}$ are centered and normalized to obtain the data set $\mathbf{D}$

$$\mathbf{D} = \left[\mathbf{x},\ \mathbf{y},\ \mathbf{z}\right] = [\mathbf{U},\ \mathbf{z}] = \begin{bmatrix} x_1 & y_1 & z_1 \\ \cdots & \cdots & \cdots \\ x_n & y_n & z_n \end{bmatrix} = \begin{bmatrix} \frac{\acute{x}_1 - m_x}{\rho_x} & \frac{\acute{y}_1 - m_y}{\rho_y} & \frac{\acute{z}_1 - m_z}{\rho_z} \\ \cdots & \cdots & \cdots \\ \frac{\acute{x}_n - m_x}{\rho_x} & \frac{\acute{y}_n - m_y}{\rho_y} & \frac{\acute{z}_n - m_z}{\rho_z} \end{bmatrix} \tag{6}$$

$$m_x = \frac{1}{n}\sum_{i=1}^{n} \acute{x}_i, \quad m_y = \frac{1}{n}\sum_{i=1}^{n} \acute{y}_i, \quad m_z = \frac{1}{n}\sum_{i=1}^{n} \acute{z}_i \tag{7}$$

$$\begin{aligned} \rho_x &= \max(|\acute{x}_1 - m_x|, \ldots, |\acute{x}_n - m_x|) \\ \rho_y &= \max(|\acute{y}_1 - m_y|, \ldots, |\acute{y}_n - m_y|) \\ \rho_z &= \max(|\acute{z}_1 - m_z|, \ldots, |\acute{z}_n - m_z|). \end{aligned} \tag{8}$$

The operator $|\cdot|$ above denotes the absolute value, and the operator $\max(\cdot)$ denotes the maximum value of a vector.

B. Initialization of the Fuzzy Nonlinear Identification Procedure

In this section, the initialization of the proposed fuzzy nonlinear identification technique will be described. In the first step of the identification procedure, the covariance matrix $\mathbf{C}$ of the data $\mathbf{D}$ (6) is computed as

$$\mathbf{C} = \frac{1}{n-1}\mathbf{D}^T\mathbf{D}. \tag{9}$$

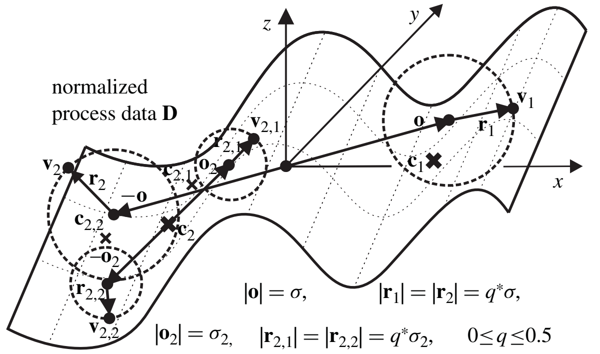


Fig. 4. Defining the initial prototypes for the GK fuzzy clustering.

The unit eigenvectors and the corresponding eigenvalues of the data covariance matrix $\mathbf{C}$ are computed with singular value decomposition (SVD). The eigenvalues represent the variances of the data $\mathbf{D}$ in the direction of the eigenvectors. The unit eigenvector $\mathbf{g}$ with the largest variance σ^2 is used as the measure for the direction of the main data expansion. This eigenvector is scaled with the corresponding standard deviation $\mathbf{o} = \sigma * \mathbf{g}$ to capture the majority of the data with the two vectors $\pm\mathbf{o}$ (Fig. 4). These two points could already be used for the initial prototypes in the GK fuzzy clustering process. The initial prototypes are then globally defined with the described deterministic procedure. To extend the search space when choosing initial prototypes, they can be placed somewhere closely and randomly around the points $\pm\mathbf{o}$. Local randomness is achieved by adding up the random vectors $\mathbf{r}_1$ and $\mathbf{r}_2$ to the already defined vectors $\pm\mathbf{o}$

$$\mathbf{v}_1 = \mathbf{o} + \mathbf{r}_1, \qquad \mathbf{v}_2 = -\mathbf{o} + \mathbf{r}_2. \tag{10}$$

The vectors $\mathbf{r}_1$ and $\mathbf{r}_2$ (10) lie randomly somewhere on the 3-D sphere surfaces (Fig. 4)

$$\begin{aligned} \mathbf{r}_1 &= f(\sigma,\ \Upsilon) = q\sigma * (\sin\theta_1\cos\varphi_1,\ \sin\theta_1\sin\varphi_1,\ \cos\theta_1) \\ \mathbf{r}_2 &= f(\sigma,\ \Upsilon) = q\sigma * (\sin\theta_2\cos\varphi_2,\ \sin\theta_2\sin\varphi_2,\ \cos\theta_2) \end{aligned} \tag{11}$$

where φ_1, φ_2 and θ_1, θ_2 are random angles drawn from the standard uniform distribution Υ on the open interval $(0, 2\pi)$ and $(0, \pi)$, respectively. The spheres have the constant radius $q * \sigma$, which is proportional to the data standard deviation σ and a predefined constant q. The range of possible values for the constant is $0 \leq q \leq 0.5$ and the typically chosen values are $q = 0.125$ or $q = 0.25$. The factor q (11) is found experimentally. In order to keep the good repeatability of the proposed nonlinear model identification technique, the radii of both spheres or the factor q should not be too large.

The data $\mathbf{D}$ (6) and their initial prototypes $\mathbf{v}_1$, $\mathbf{v}_2$ are the input into the GK algorithm, which results in $nc = 2$ cluster centers $\mathbf{c}_1$ and $\mathbf{c}_2$ (Fig. 4), the clusters' membership functions $\mathbf{f}_1$ and $\mathbf{f}_2$, and the clusters' fuzzy covariance matrices $\mathbf{P}_1$ and $\mathbf{P}_2$.

With the deterministic procedure of calculating the unit eigenvector and the corresponding standard deviation by using the SVD, the initial prototypes placement is globally directed. The prototypes are placed away from the data mean (coordinate origin) to embrace the majority of the data $\mathbf{D}$ (6) in the direction of the main data expansion. In order to extend the search space in defining the initial prototypes

(10), local random vectors lying on the sphere surfaces (11) are added to the deterministically chosen eigenvectors. This paradigm, where two initial prototypes are defined by the globally dictated deterministic initialization followed by the locally random initialization, is also kept in the iterative part of the identification procedure (Fig. 4). There more local linear models are defined to better describe the process nonlinearity.

If the dataset $\acute{\mathbf{D}}$ (4) is not normalized, then due to possibly different orders of the data magnitude according to the $\acute{x}$, $\acute{y}$, and $\acute{z}$ coordinates, an ellipsoid surface must be considered in (11). It is therefore more convenient to use the normalized dataset $\acute{\mathbf{D}}$ (4) for the nonlinear identification, since then the sphere instead of the ellipsoid surface can be used.

C. Iterative Fuzzy Nonlinear Identification Procedure

This section describes the iterative part of the proposed fuzzy nonlinear identification procedure. Here, the number of clusters is increased until all the identified local models corresponding to each cluster fit the modeled 3-D surface (Fig. 4) well with respect to defined criteria or until a predefined maximum number of loop iterations is reached. The iterative part of the identification procedure is therefore executed in a loop $\mathcal{L}$.

First, the validity functions $\Phi_j(\mathbf{u}_i)$ ($j = 1, \ldots, nc$ and $i = 1, \ldots, n$) are calculated from the Gaussian functions $\mu_j(\mathbf{u}_i)$ as follows:

$$\begin{aligned}\mu_j(\mathbf{u}_i) &= e^{-\gamma * (\mathbf{u}_i - \mathbf{c}_{u_j})\mathbf{P}_{uj}^{-1}(\mathbf{u}_i - \mathbf{c}_{u_j})^T} && (12)\\ \mathbf{u}_i &= (x_i,\ y_i), \quad i = 1, \ldots, n\\ \mathbf{c}_j &= (c_{x_j},\ c_{y_j},\ c_{z_j}) \Rightarrow \mathbf{c}_{u_j} = (c_{x_j},\ c_{y_j})\\ \mathbf{P}_j &= \begin{bmatrix} \sigma_{x_j}^2 & \sigma_{xy_j} & \sigma_{xz_j}\\ \sigma_{yx_j} & \sigma_{y_j}^2 & \sigma_{yz_j}\\ \sigma_{zx_j} & \sigma_{zy_j} & \sigma_{z_j}^2 \end{bmatrix} \Rightarrow\\ \mathbf{P}_{u_j} &= \begin{bmatrix} \sigma_{x_j}^2 & \sigma_{xy_j}\\ \sigma_{yx_j} & \sigma_{y_j}^2 \end{bmatrix}, \quad j = 1, \ldots, nc && (13)\end{aligned}$$

where $\mathbf{u}_i$ ($i = 1, \ldots, n$) are the input space data, $\mathbf{c}_{u_j}$ ($j = 1, \ldots, nc$) are the input space coordinates of the cluster centers obtained in the last execution of the GK fuzzy-clustering algorithm, and $\mathbf{P}_{u_j}$ are the fuzzy covariance matrices of the input space data. These matrices are the submatrices of the fuzzy covariance matrices $\mathbf{P}_j$ from the last execution of the GK algorithm. γ is a factor that defines the crispness of the Gaussian functions $\mu_j(\mathbf{u}_i)$ and consequently the sharpness of the validity functions $\Phi_j(\mathbf{u}_i)$, i.e., normalized Gaussian functions

$$\Phi_j(\mathbf{u}_i) = \frac{\mu_j(\mathbf{u}_i)}{\Sigma_{k=1}^{nc}\mu_k(\mathbf{u}_i)}, \quad i = 1, \ldots, n, \quad j = 1, \ldots, nc. \quad (14)$$

The normalization in the equation above is necessary to build the partition of unity (2). The large factor γ (12) causes crisp validity functions $\Phi_j(\mathbf{u}_i)$ (14). The range of possible values for the constant is $0 < \gamma$ and the typically chosen values are 0.5, 1, or 2. How to choose the parameter γ will be discussed in what follows.

The centers of the validity functions correspond to the centers of the clusters. The size and the shape of each validity function is defined with the fuzzy covariance matrix of the corresponding cluster and with the Gaussian functions of the neighboring clusters. With the fuzzy covariance matrix $\mathbf{P}_j$ (13), an arbitrary orientation and size of the clusters can be described. The obtained validity functions $\Phi_j(\mathbf{u}_i)$ (14) are therefore highly flexible in terms of adapting the validity regions of the local linear models to the local linear regions of the process. In this way an axes-flexible partitioning of the input space is achieved.

The results of the last execution of the GK algorithm are also the fuzzy membership functions $\mathbf{f}_j(\mathbf{u}_i)$ for all the clusters ($j = 1, \ldots, nc$). From all the data $\mathbf{D}$ (6), the data $\tilde{\mathbf{d}}_{k,j} = [\tilde{x}_{k,j},\ \tilde{y}_{k,j},\ \tilde{z}_{k,j}] = [\tilde{\mathbf{u}}_{k,j},\ \tilde{z}_{k,j}]$ ($k = 1, \ldots, m_j$) and their validity function values $\Phi_j(\tilde{\mathbf{u}}_{k,j})$ satisfying the criteria

$$\mathbf{f}_j(\mathbf{u}_i) > \delta \quad (15)$$

where $\delta = 0.5$, are found for each cluster ($j = 1, \ldots, nc$). The data $\tilde{\mathbf{d}}_{k,j}$ together with their validity function weights $\Phi_j(\tilde{\mathbf{u}}_{k,j})$ correspond to the jth cluster, with respect to the upper defined inequality. Fuzzy membership functions $\mathbf{f}_j(\mathbf{u}_i)$, which form a partition of unity, are functions in the interval (0, 1]. Therefore, at the intersection of the two neighboring membership functions their value is approximately $\mathbf{f}_j(\mathbf{u}_i) = 0.5$ [see Fig. 5(b)]. If the membership functions are soft, a threshold value δ much lower than 0.5 causes that many data points are captured in both neighboring sets $\tilde{\mathbf{d}}_{k,j}$ and are therefore considered twice in the modeling procedure. And conversely, a threshold value δ much higher than 0.5 causes that many data points between the two neighboring clusters are discarded. However, if the membership functions $\mathbf{f}_j(\mathbf{u}_i)$ are crisp, this effect is not so evident and values higher than 0.5 can be chosen for the threshold. To consider the worst case when soft membership functions occur, a threshold value of $\delta = 0.5$ is chosen here, since then almost no data between the two neighboring clusters are considered twice or ignored in the modeling procedure.

The 3-D local linear model (plane) weighted with the validity function $\Phi_j(\tilde{\mathbf{u}}_{k,j})$

$$\begin{aligned}\tilde{z}_{k,j} &= (a_j\tilde{x}_{k,j} + b_j\tilde{y}_{k,j} + c_j) * \Phi_j(\tilde{\mathbf{u}}_{k,j})\\ k &= 1, \ldots, m_j, \quad j = 1, \ldots, nc && (16)\end{aligned}$$

where a_j, b_j, and c_j are the parameters of the explicit plane equation, is fitted to the jth cluster's data $\tilde{\mathbf{d}}_{k,j}$ ($k = 1, \ldots, m_j$) by using the LS method. The above-defined system of the m_j linear equations with the three variables a_j, b_j, and c_j solved with the LS method results in the jth local linear model parameters $\hat{a}_j$, $\hat{b}_j$, and $\hat{c}_j$.

The quality of each local linear model, which corresponds to the cluster $\mathbf{c}_j$ ($j = 1, \ldots, nc$), is estimated with the standard deviation of the error between the linear model and the cluster's data $\tilde{\mathbf{d}}_{k,j} = [\tilde{x}_{k,j},\ \tilde{y}_{k,j},\ \tilde{z}_{k,j}]$ as follows:

$$\begin{aligned}\sigma_{e_j} &= \sqrt{\frac{1}{m_j - 1}\sum_{k=1}^{m_j}\delta_{k,j}^2}\\ \delta_{k,j} &= \tilde{x}_{k,j}\cos\hat{\alpha}_j + \tilde{y}_{k,j}\cos\hat{\beta}_j + \tilde{z}_{k,j}\cos\hat{\gamma}_j - \hat{p}_j. && (17)\end{aligned}$$

$\hat{\alpha}_j$, $\hat{\beta}_j$, $\hat{\gamma}_j$, and $\hat{p}_j$ above are the parameters of the normal plane equation converted from the explicit plane equation parameters $\hat{a}_j$, $\hat{b}_j$ and $\hat{c}_j$. $\delta_{k,j}$ (17) is the normal distance (error) between the data point $\tilde{\mathbf{d}}_{k,j}$ and the jth linear model (plane).

If the standard deviation of the error σ_{e_j} (17) is below the threshold T (18) for all the clusters ($j = 1, \ldots, nc$), then all the local linear models fit the corresponding subregion of the process well. Then the iterative part of the fuzzy nonlinear identification is completed, which is shown in the next subsection. The termination threshold T (18) is defined empirically according to the desired level of the final LMN model accuracy. By setting a small threshold T, the final number of all the local linear models nc is increased, which means that the overall LMN model (5) becomes more complex and more accurate. However, the value of the threshold T should not be lower than the noise level of the normalized process data.

If the standard deviation of the error σ_{e_j} (17) is, for some models, above the threshold T

$$\sigma_{e_j} > T \tag{18}$$

then these local linear models fit badly to the corresponding cluster's data. The cluster with the largest standard deviation σ_{e_j} has the worst local linear model. The data of this cluster ($j = w$) is denoted as $\tilde{\mathbf{d}}_{k,w}$. To better describe the nonlinearity of this data, the wth cluster is further split into two clusters, each modeled with a new local linear model. Choosing the worst modeled data cluster that has to be split is a supervised operation in the LMN learning procedure. The cluster is split with a procedure that is analogous to the already described initialization part of the fuzzy nonlinear identification (Fig. 4). There, the globally directed deterministic initialization for the GK fuzzy clustering is followed by the locally random initialization. Fig. 4 also shows the first step of the iterative identification procedure, where the second ($w = 2$) cluster is split into two new clusters. Two new initial prototypes are placed away from the splitting cluster's center $\mathbf{c}_w$ to embrace the majority of the splitting cluster's data in the direction of the cluster's main data expansion

$$\begin{aligned} \mathbf{v}_{w,1} &= \mathbf{c}_w + \mathbf{o}_w + \mathbf{r}_{w,1}, \qquad \mathbf{o}_w = \sigma_w * \mathbf{g}_w \\ \mathbf{v}_{w,2} &= \mathbf{c}_w - \mathbf{o}_w + \mathbf{r}_{w,2}. \end{aligned} \tag{19}$$

$\mathbf{g}_w$ in the equation above denotes the unit eigenvector of the fuzzy covariance matrix of the splitting cluster $\mathbf{P}_w$ ($j = w$) (13) which corresponds to the largest variance σ_w^2. The unit eigenvector $\mathbf{g}_w$ therefore indicates the direction of the main data expansion of the splitting cluster. $\mathbf{r}_{w,1} = f(\sigma_w, \Upsilon)$ and $\mathbf{r}_{w,2} = f(\sigma_w, \Upsilon)$ (11) and (19) are random vectors, which lie randomly somewhere on the 3-D sphere surfaces (Fig. 4). The spheres have a constant radius $q * \sigma_w$, which is proportional to the standard deviation of the splitting cluster's data σ_w and a constant q. The factor q has already been defined in the initialization part of the identification procedure.

The centers of the clusters obtained with the iterative GK algorithm are not accurate, but they are close to the optimal cluster centers with respect to some tolerance region. Deterministically defined initial prototypes for the GK clustering $\mathbf{c}_w \pm \mathbf{o}_w$ (19) are defined on the basis of the splitting cluster's center $\mathbf{c}_w$, which is not accurate. The error of the inaccurate center of the splitting cluster is, in each iteration of the identification algorithm, therefore also reflected on the deterministically defined initial prototypes $\mathbf{c}_w \pm \mathbf{o}_w$. The error of each splitting cluster's center $\mathbf{c}_w$ therefore has an influence on the evolution of the iterative identification. To take this error into account, the search space in choosing the two initial prototypes for the GK clustering is, in the iterative part of the identification algorithm, also extended by introducing a local randomness with the two vectors $\mathbf{r}_{w,1} = f(\sigma_w, \Upsilon)$ and $\mathbf{r}_{w,2} = f(\sigma_w, \Upsilon)$ (19).

The last execution of the GK algorithm results in nc cluster centers $\mathbf{c}_j$. The splitting cluster's data $\tilde{\mathbf{d}}_{k,w}$ and their initial prototypes $\mathbf{v}_{w,1}$ and $\mathbf{v}_{w,2}$ (19) are the input into the GK algorithm, which results in two cluster centers $\mathbf{c}_{w,1}$ and $\mathbf{c}_{w,2}$ (Fig. 4). The original center of the splitting cluster $\mathbf{c}_w$ is replaced with the two new cluster centers $\mathbf{c}_{w,1}$ and $\mathbf{c}_{w,2}$

$$\begin{aligned} (\mathbf{c}_1, \ldots, \mathbf{c}_w, \ldots, \mathbf{c}_{nc}) &\rightarrow (\mathbf{v}_1, \ldots, \mathbf{v}_w, \mathbf{v}_{w+1}, \ldots, \mathbf{v}_{nc_{new}}) \\ &= (\mathbf{c}_1, \ldots, \mathbf{c}_{w,1}, \mathbf{c}_{w,2}, \ldots, \mathbf{c}_{nc}), \quad nc_{new} = nc + 1. \end{aligned} \tag{20}$$

$\mathbf{c}_{w,1}$ and $\mathbf{c}_{w,2}$ together with the remaining cluster centers $\mathbf{c}_j$ ($j = 1, \ldots, w-1, w+1, \ldots, nc$), that are not split, form new initial prototypes $\mathbf{v}_j$ ($j = 1, \ldots, nc_{new}$) for the GK clustering and therefore the overall number of initial prototypes is increased by one $nc_{new} = nc + 1$. The data $\mathbf{D}$ and their new initial prototypes $\mathbf{v}_j$ are the input into the GK algorithm, which results in the new cluster centers $\mathbf{c}_j$, the cluster membership functions $\mathbf{f}_j$, and the clusters' fuzzy covariance matrices $\mathbf{P}_j$ ($j = 1, \ldots, nc_{new}$). Using the GK fuzzy clustering the LMN learning is not supervised. Fuzzy nonlinear identification is then continued at the beginning of the loop $\mathcal{L}$, and the described procedure of finding the local linear models that better fit to the process nonlinearity is repeated.

If the unnormalized dataset $\acute{\mathbf{D}}$ (4) is used for the identification, then the absolute distance $\delta_{k,j}$ (17) can, intuitively, be normalized by the cluster's output data $\tilde{z}_{k,j}$ to deal with various static processes that meet different orders of the output data magnitude. The cluster's standard deviation of the error σ_{e_j} (17) then becomes very large if the cluster's data are close to zero. This tends to split the clusters of data that are close to zero, even though they can be better modeled than some other clusters of data that are far away from zero. For this reason, the distances $\delta_{k,j}$ ($k = 1, \ldots, m_j$) (17) of all the clusters ($j = 1, \ldots, nc$) should be normalized with the same value, e.g., with the standard deviation (sparsity) of the output data $\acute{\mathbf{z}}$ (4). Normalizing the original dataset $\acute{\mathbf{D}}$ (4) that represents a static nonlinear process has a similar effect as normalizing the error distance $\delta_{k,j}$ (17) with the standard deviation of the output data.

D. Terminating the Fuzzy Nonlinear Identification

If all the local linear models fit to the corresponding local linear regions of the process well, the iterative part of the fuzzy nonlinear identification is completed. This happens when the standard deviation of the error between the local linear model and the corresponding cluster's data σ_{e_j} (17) is below the threshold T (18) for all the clusters ($j = 1, \ldots, nc$). If the maximum number of loop ($\mathcal{L}$) executions is reached, the identification procedure is also completed. By determining when the number of local linear models stops increasing, the LMN learning is supervised. Finally, the whole output model, considering the original data set $\acute{\mathbf{D}}$ (4), is computed as follows. The result of the last execution of the GK fuzzy-clustering algorithm needed for calculating the whole output model $\hat{z}(\acute{\mathbf{u}}_i)$ (5) is the input space coordinates of the cluster centers $\mathbf{c}_{u_j}$ ($j = 1, \ldots, nc$) (13) and the corresponding fuzzy covariance matrices of the input space data $\mathbf{P}_{u_j}$ (13). To approximate the original dataset $\acute{\mathbf{D}}$ (4) with the output model $\hat{z}(\acute{\mathbf{u}}_i)$ (5), first the cluster centers $\mathbf{c}_{u_j} = (c_{x_j},\ c_{y_j})$ and the fuzzy covariance matrices $\mathbf{P}_{u_j}$ must be converted to

$$\acute{\mathbf{c}}_{u_j} = (\rho_x * c_{x_j} + m_x,\ \rho_y * c_{y_j} + m_y)$$
$$\acute{\mathbf{P}}_{u_j} = \begin{bmatrix} \rho_x^2 * \sigma_{x_j}^2 & \rho_x \rho_y * \sigma_{xy_j} \\ \rho_y \rho_x * \sigma_{yx_j} & \rho_y^2 * \sigma_{y_j}^2 \end{bmatrix}, \quad j = 1, \ldots, nc \qquad (21)$$

which is the inverse of the transformation (6), where the original data $\acute{\mathbf{D}}$ (4) are centered and normalized. m_x, m_y, ρ_x, and ρ_y are defined in (7) and (8). The Gaussian functions $\acute{\mu}_j(\acute{\mathbf{u}}_i)$ for each cluster ($j = 1, \ldots, nc$) and all the data ($i = 1, \ldots, n$) are calculated with (12), where $\mathbf{u}_i$, $\mathbf{c}_{u_j}$, and $\mathbf{P}_{u_j}$ are replaced with $\acute{\mathbf{u}}_i = (\acute{x}_i,\ \acute{y}_i)$ and the above-defined $\acute{\mathbf{c}}_{u_j}$ and $\acute{\mathbf{P}}_{u_j}$, respectively. The validity functions $\acute{\Phi}_j(\acute{\mathbf{u}}_i)$ are calculated with (14), where $\mu_j(\mathbf{u}_i)$ is replaced with $\acute{\mu}_j(\acute{\mathbf{u}}_i)$.

Now the parameters of the local linear models for approximating the original process data $\acute{\mathbf{D}}$ (4) can be computed. In the final part of the identification procedure, these parameters are found with a global learning approach [39]. The parameters of all the local linear models are estimated with a single regression operation, where all the data of the process are considered.

The small factor γ (12) causes the fuzzy validity functions $\acute{\Phi}_j(\acute{\mathbf{u}}_i)$. If the validity functions $\acute{\Phi}_j(\acute{\mathbf{u}}_i)$ are fuzzy, there is some overlapping between the neighboring validity functions, especially in the transit regions between the local models [Fig. 5(b)]. In this case, the estimated parameters of the local models depend on their neighboring local models. This can cause the resulting local linear models not describing the corresponding local regions of the nonlinear process very accurately [Fig. 5(a)], even though these subregions are very accurately described with the whole LMN model (1). If the validity functions $\acute{\Phi}_j(\acute{\mathbf{u}}_i)$ are crisp (large parameter γ), the resulting local linear models [Fig. 5(a)] accurately describe the corresponding subregions of the nonlinear process. But the overall LMN model is then less accurate compared to the case when fuzzy validity functions $\acute{\Phi}_j(\acute{\mathbf{u}}_i)$ are used. The parameter γ is found empirically so that a compromise between the accuracy of the local linear models and the accuracy of the overall LMN model is reached.

The sum of all ($j = 1, \ldots, nc$) local linear models weighted with their validity functions $\acute{\Phi}_j(\acute{\mathbf{u}}_i)$, taking into account all the input space data $\acute{\mathbf{u}}_i$ ($i = 1, \ldots, n$), tends to approximate the outputs $\acute{z}_i$

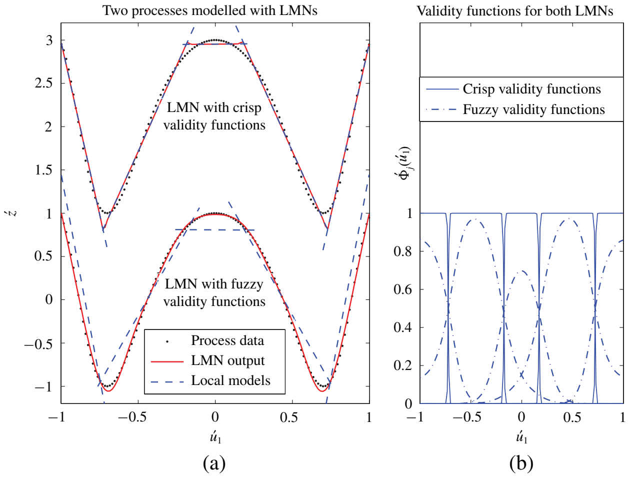


Fig. 5. Two LMNs. (a) Outputs and local models. (b) Crisp and fuzzy validity functions.

$$\acute{z}_i = \sum_{j=1}^{nc} (a_j \acute{x}_i + b_j \acute{y}_i + c_j)\acute{\Phi}_j(\acute{\mathbf{u}}_i), \quad i = 1, \ldots, n \qquad (22)$$

which forms a system of n linear equations with $3 * nc$ variables a_j, b_j, and c_j ($j = 1, \ldots, nc$). The system is solved by using the LS method and results in the parameters $\hat{a}_j$, $\hat{b}_j$, and $\hat{c}_j$ ($j = 1, \ldots, nc$) of all local linear models. The LMN model $\hat{z}(\acute{\mathbf{u}}_i)$ considering all the input space data $\acute{\mathbf{u}}_i$ ($i = 1, \ldots, n$) is finally calculated with (5). The quality of the whole LMN model is estimated as the standard deviation σ_z (23) of the error between the centered and normalized output generated from the model $\check{z}_i$ and the centered and normalized output data z_i (6)

$$\sigma_z = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(\check{z}_i - z_i)^2}, \qquad \check{z}_i = \frac{\hat{z}_i - m_z}{\rho_z} \qquad (23)$$

where m_z and ρ_z are defined in (7) and (8), respectively. Since the normalized output data $\check{z}_i$ and z_i ($i = 1, \ldots, n$) are considered in the equation above, σ_z represents the standard deviation of the relative error between the original output from the process $\acute{z}_i$ (4) and the output generated from the model $\hat{z}(\acute{\mathbf{u}}_i)$ (5).

E. Key Features of the Proposed Algorithm

Many attributes of the proposed algorithm can be derived from theoretical investigations. In the following, the key attributes of the proposed algorithm are summarized.

1) *Noise sensitivity:* Generally, LMNs are robust against noise [40]–[43]. All local models are locally estimated with the weighted LS method. The local estimation has an inherent regularization effect because of the overlaps between the local models. The partitioning itself is robust against noise because of two main reasons. Fuzzy clustering itself is more robust against noise than hard clustering [44], [45] and, furthermore, with the heuristic

Algorithm 1 Pseudocode of the Iterative Identification Algorithm

```
Transform all the process data.
Define initial prototypes: $\mathbf{v}_1 = \mathbf{o} + \mathbf{r}_1$,   $\mathbf{v}_2 = -\mathbf{o} + \mathbf{r}_2$.
GK clustering using all the data results in cluster centers $\mathbf{c}_1$ and $\mathbf{c}_2$, $nc = 2$.
While (End criteria is not met)
    Compute $\Phi_j(\mathbf{u}_i)$,  $i = 1, \ldots, n$,  $j = 1, \ldots, nc$.
    Compute $\hat{a}_j$, $\hat{b}_j$, and $\hat{c}_j$, $j = 1, \ldots, nc$ using local LS.
    Compute $\sigma_{e_j}$,  $j = 1, \ldots, nc$.
    For the cluster with the largest $\sigma_{e_j}$ $(j = w)$ define initial prototypes: $\mathbf{v}_{w,1} = \mathbf{c}_w + \mathbf{o}_w + \mathbf{r}_{w,1}$,   $\mathbf{v}_{w,2} = \mathbf{c}_w - \mathbf{o}_w + \mathbf{r}_{w,2}$.
    GK clustering using only the splitting-cluster's data results in cluster centers $\mathbf{c}_{w,1}$ and $\mathbf{c}_{w,2}$.
    Define initial prototypes: $(\mathbf{v}_1, \ldots, \mathbf{v}_w, \mathbf{v}_{w+1}, \ldots, \mathbf{v}_{nc_{new}}) = (\mathbf{c}_1, \ldots, \mathbf{c}_{w,1}, \mathbf{c}_{w,2}, \ldots, \mathbf{c}_{nc})$,   $nc_{new} = nc + 1$.
    GK clustering using all the data results in cluster centers $\mathbf{c}_j$,  $j = 1, \ldots, nc$,  $nc = nc_{new}$.
End While
Re-transform $\mathbf{c}_{u_j}$ and $\mathbf{P}_{u_j}$,  $j = 1, \ldots, nc$ and use original process data in what follows.
Compute $\acute{\Phi}_j(\acute{\mathbf{u}}_i)$,  $i = 1, \ldots, n$,  $j = 1, \ldots, nc$.
Compute $\hat{a}_j$, $\hat{b}_j$, and $\hat{c}_j$, $j = 1, \ldots, nc$ using global LS.
$\hat{z}(\acute{\mathbf{u}}_i) = \sum_{j=1}^{nc} (\hat{a}_j \acute{x}_i + \hat{b}_j \acute{y}_i + \hat{c}_j) \acute{\Phi}_j(\acute{\mathbf{u}}_i)$,  $i = 1, \ldots, n$.
```

construction algorithm, well-suited initial values for the covariance matrix optimization are provided.

2) *Computational effort:* In each iteration, the proposed algorithm performs GK fuzzy clustering with two additional clusters. Therefore, the number of parameters of each iteration increases by the size of the additional fuzzy clustering covariance matrix. For each single split, one covariance matrix is required. The covariance matrices are of symmetric structure, i.e., for each split $(p + 1)(p + 2)/2$, parameters (p = # inputs) have to be optimized. According to [46], the computational complexity of the GK clustering algorithm is considered as $O(n^2)$, where n is the number of data. The proposed identification algorithm is incrementally growing, so this leads to fast training times. Although the training is slow compared to greedy algorithms such as, e.g., Lolimot [40], the training times are more or less equal compared to product space clustering or even faster.

3) *Data fitting:* The partitioning with the new algorithm is very flexible. This leads to an algorithm that is able to cover with highly nonlinear process. Furthermore, due to the flexibility the algorithm is well suited to model processes with high dimensionality. Process nonlinearities are covered efficiently as a result of the flexible unsupervised product space clustering on the one hand and the supervised tree construction algorithm on the other [47], [48]. Besides the LS regression used in this paper, the orthogonal LS method is also often applied for model identification [49], [50].

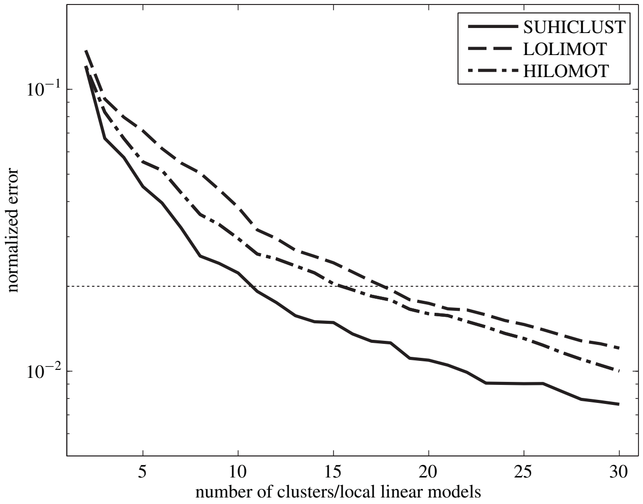


Fig. 6. Convergence behavior of Suhiclust, Lolimot, and Hilomot. The training error threshold of $\sigma_z = 0.02$ is marked with the dotted line.

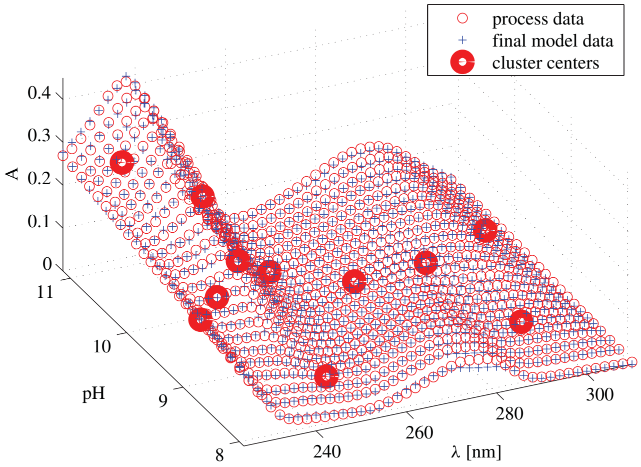


Fig. 7. Resulting final model data generated with Suhiclust compared to the process data. The LMN consists of 11 local linear models.

In addition to the theoretically derived attributes the next section shows the performance of the proposed algorithm with respect to nonlinear process data.

IV. Results of the Fuzzy Nonlinear Model Identification of the Drug Absorption Spectra Process

The fuzzy nonlinear model identification technique proposed in this paper is tested on a 3-D absorbance–response surface [51], which represents the multiwavelength absorption spectra of the protonation equilibria of phenylephrine in terms of the dependence on the pH. This defines a static process with two inputs (pH and wavelength λ[nm]) and one output (absorbance A), which is shown in Figs. 7 and 8(a). The data matrix $\acute{\mathbf{D}}$ (4) is of size $n \times 3$, where $n = 1056$ is the number of data.

The training of the LMN was performed with three different partitioning algorithms. The algorithm that is proposed in this paper is called *Suhiclust*, which stands for *Su*pervised *Hi*erarchical *Clust*ering. The second one is Lolimot [12].

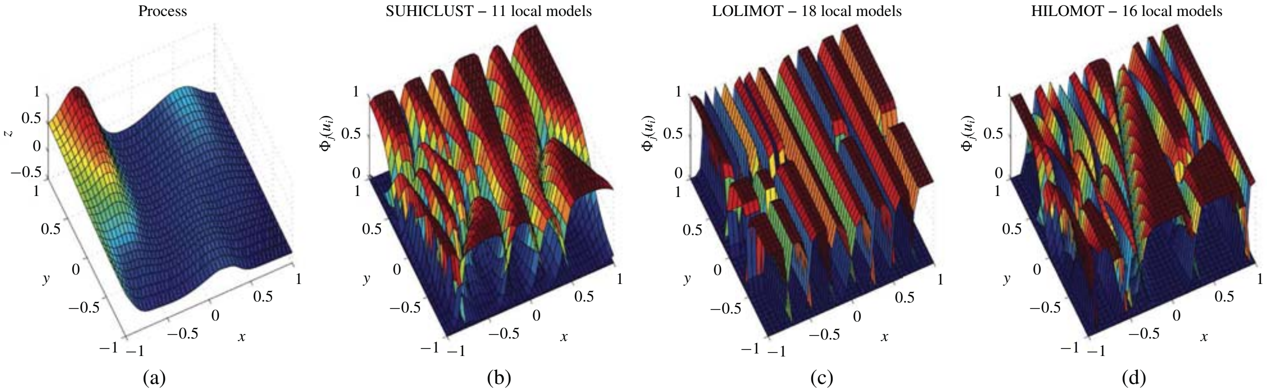


Fig. 8. (a) Normalized process data, (b) partitioning results of Suhiclust, (c) Lolimot, and (d) Hilomot. The complexity of the shown partitions is chosen such that a training error of $\sigma_z = 0.02$ is achieved. Suhiclust is the most flexible approach in this comparison.

Lolimot produces axes-orthogonal splits. Therefore, the partitioning is less flexible than the partitioning with Suhiclust. Furthermore, the process was modeled with an extension of the Lolimot algorithm. Hilomot is a tree-construction algorithm like Lolimot. In contrast to Lolimot, Hilomot generates a LMN with axes-oblique partitioning which is realized with sigmoidal splitting functions. As with Lolimot and Suhiclust, Hilomot adds in each iteration one local linear model to the overall model. The detailed functioning of Hilomot is explained in [30].

The dataset $\acute{\mathbf{D}}$ (4) representing the original process is first normalized, as shown in (6), to obtain the dataset $\mathbf{D}$, which is shown in Fig. 8(a). Then the nonlinear identification procedure is performed on the basis of the normalized process data. The Suhiclust parameters are set to $q = 0.125$ (11) and $\gamma = 0.5$ (12). The convergence behavior of the three partitioning algorithms is shown in Fig. 6. The normalized training error σ_z (23) is monotonically decreasing with all investigated training algorithms. At each number of local models, Suhiclust algorithm results in lower error σ_z than Lolimot and Hilomot. The model complexity is chosen such that an error less than $\sigma_z = 0.02$ is achieved (dotted line in Fig. 6), which corresponds to the termination threshold of the Suhiclust algorithm $T = 0.05$ (18). In this case, Suhiclust requires 11 local models, Lolimot 18 local models, and Hilomot 16 local models.

Fig. 8(b)–(d) shows the resulting partitions of the structure identification procedure for Suhiclust, Lolimot, and Hilomot, respectively. They illustrate the higher flexibility of the partitioning with Suhiclust compared to Lolimot and Hilomot. Due to the restriction of axes-orthogonal splitting, Lolimot needs the largest number of local linear models in order to achieve the given training error threshold of $\sigma_z = 0.02$. Although Hilomot is able to produce a very flexible partition due to the application of axes-oblique splits that are generated with sigmoidal splitting functions, Hilomot requires only two local models less than Lolimot to achieve the goal. The normalized Gaussian membership functions $\Phi_j(\mathbf{u}_i)$ (14), generated with Suhiclust, are even more flexible than the validity functions of Hilomot. This leads to the most effective LMN structure in this comparison.

The generalization behavior of the model can be monitored visually in a 3-D plot, because the process has no more than two input dimensions. Although the model complexity with 11 local models is quite low, the model generated with Suhiclust shows good modeling results as illustrated in Figs. 6 and 7.

V. Conclusion

In this paper, a new fuzzy nonlinear identification technique in the framework of a LMN was presented. In order to construct the LMN, a new incremental approach for defining the number and the initial locations of the cluster centers for the GK fuzzy clustering was proposed. In each iteration of the identification algorithm, the worst modeled cluster was split into two new clusters. Two initial prototypes for the GK clustering were placed away from the splitting cluster's center to embrace the majority of the splitting cluster's data in the direction of the main data expansion. This direction was defined with the largest eigenvector of the splitting cluster's fuzzy covariance matrix.

The validity functions were defined with the fuzzy covariance matrices of the clusters obtained with the GK algorithm and for this reason they were highly adaptable to the local linear regions of the nonlinear process. In this way, an axes-flexible partitioning of the input space was achieved and therefore the process can be described with a very sparse amount of local models, which leads to a parsimonious LMN structure. The proposed fuzzy nonlinear identification technique and two other methods for constructing the LMNs, i.e., Lolimot [12] and Hilomot [30], were tested on the drug absorption spectra process. The experimental comparison illustrates the higher adaptability of partitioning with the proposed algorithm compared to Lolimot and Hilomot.

References

[1] M. D. Brown, D. Flynn, and G. W. Irwin, "Multiple model nonlinear control of synchronous generators," *Trans. Inst. Meas. Control*, vol. 24, no. 3, pp. 215–230, Aug. 2002.

[2] K. Azman and J. Kocijan, "Non-linear model predictive control for models with local information and uncertainties," *Trans. Inst. Meas. Control*, vol. 30, no. 5, pp. 371–396, Dec. 2008.

[3] J. Novak and V. Bobal, "Predictive control of the heat exchanger using local model network," in *Proc. 17th Medit. Conf. Control Autom.*, Thessaloniki, Greece, Jun. 2009, pp. 657–662.

[4] S. Garcia-Nieto, M. Martinez, X. Blasco, and J. Sanchis, "Nonlinear predictive control based on local model networks for air management in diesel engines," *Control Eng. Practice*, vol. 16, no. 12, pp. 1399–1413, Dec. 2008.

[5] R. Gao, A. O'Dywer, and E. Coyle, "A nonlinear PID controller for CSTR using local model networks," in *Proc. IEEE 4th World Congr. Intell. Control Autom.*, vol. 4. Shanghai, China, Jun. 2002, pp. 3278–3282.

[6] E. N. Skoundrianos and S. G. Tzafestas, "Finding fault - fault diagnosis on the wheels of a mobile robot using local model neural networks," *IEEE Robot. Autom. Mag.*, vol. 11, no. 3, pp. 83–90, Sep. 2004.

[7] J. Moll, R. T. Schulte, B. Hartmann, C.-P. Fritzen, and O. Nelles, "Multi-site damage localization in anisotropic plate-like structures using an active guided wave structural health monitoring system," *Smart Mater. Struct.*, vol. 19, no. 4, pp. 45022–45037, 2010.

[8] S. Seher-Weiss, "Identification of nonlinear aerodynamic derivatives using classical and extended local model networks," *Aerosp. Sci. Technol.*, vol. 15, no. 1, pp. 33–44, Jan.–Feb. 2011.

[9] L. Monzon, A. Ferreira, and I. Pedreira, "Incremental local model networks for time series prediction," in *Proc. Int. Joint Conf. Neural Netw.*, Washington D.C., Jul. 2001, pp. 1652–1656.

[10] I. Gath and A. B. Geva, "Unsupervised optimal fuzzy clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 7, pp. 773–780, Jul. 1989.

[11] L. Breiman, J. H. Friedman, R. Olshen, and C. J. Stone, *Classification and Regression Trees*. New York: Chapman & Hall, 1984.

[12] O. Nelles, S. Sinsel, and R. Isermann, "Local basis function networks for identification of a turbocharger," in *Proc. UKACC Int. Conf. Control*, vol. 1. Exeter, U.K., Sep. 1996, pp. 7–12.

[13] A. Brandstetter and A. Artusi, "Radial basis function networks GPU-based implementation," *IEEE Trans. Neural Netw.*, vol. 19, no. 12, pp. 2150–2154, Dec. 2008.

[14] G. Gregorčič and G. Lightbody, "Nonlinear system identification: From multiple-model networks to Gaussian processes," *Eng. Appl. Artif. Intell.*, vol. 21, no. 7, pp. 1035–1055, Oct. 2008.

[15] G. Gregorčič and G. Lightbody, "Local model network identification with Gaussian processes," *IEEE Trans. Neural Netw.*, vol. 18, no. 5, pp. 1404–1423, Sep. 2007.

[16] S. McLoone, M. Brown, G. Irwin, and G. Lightbody, "A hybrid linear/nonlinear training algorithm for feedforward neural networks," *IEEE Trans. Neural Netw.*, vol. 9, no. 4, pp. 669–684, Jul. 1998.

[17] Q. Gan and C. J. Harris, "A hybrid learning scheme combining EM and MASMOD algorithms for fuzzy local linearization modeling," *IEEE Trans. Neural Netw.*, vol. 12, no. 1, pp. 43–53, Jan. 2001.

[18] J.-X. Peng, K. Li, and D.-S. Huang, "A hybrid forward algorithm for RBF neural network construction," *IEEE Trans. Neural Netw.*, vol. 17, no. 6, pp. 1439–1451, Nov. 2006.

[19] M. Falasconi, A. Gutierrez, M. Pardo, G. Sberveglieri, and S. Marco, "A stability based validity method for fuzzy clustering," *Pattern Recognit.*, vol. 43, no. 4, pp. 1292–1305, Apr. 2010.

[20] S. L. Chiu, "Fuzzy model identification based on cluster estimation," *J. Intell. Fuzzy Syst.*, vol. 2, no. 3, pp. 267–278, 1994.

[21] K. R. Zalik, "Cluster validity index for estimation of fuzzy clusters of different sizes and densities," *Pattern Recognit.*, vol. 43, no. 10, pp. 3374–3390, 2010.

[22] C.-H. Li, B.-C. Kuo, and C.-T. Lin, "LDA-based clustering algorithm and its application to an unsupervised feature extraction," *IEEE Trans. Fuzzy Syst.*, vol. 19, no. 1, pp. 152–163, Feb. 2011.

[23] S. Krinidis and V. Chatzis, "A robust fuzzy local information C-means clustering algorithm," *IEEE Trans. Image Process.*, vol. 19, no. 5, pp. 1328–1337, May 2010.

[24] R. J. Hathaway and Y. Hu, "Density-weighted fuzzy C-means clustering," *IEEE Trans. Fuzzy Syst.*, vol. 17, no. 1, pp. 243–252, Feb. 2009.

[25] M.-S. Yang and C.-Y. Lai, "A robust automatic merging possibilistic clustering method," *IEEE Trans. Fuzzy Syst.*, vol. 19, no. 11, pp. 26–41, Feb. 2011.

[26] S. Chatzis and T. Varvarigou, "Factor analysis latent subspace modeling and robust fuzzy clustering using t-distributions," *IEEE Trans. Fuzzy Syst.*, vol. 17, no. 3, pp. 505–517, Jun. 2009.

[27] K.-L. Wu, J. Yu, and M.-S. Yang, "A novel fuzzy clustering algorithm based on a fuzzy scatter matrix with optimality tests," *Pattern Recognit. Lett.*, vol. 26, no. 5, pp. 639–652, Apr. 2005.

[28] D. Graves and W. Pedrycz, "Kernel-based fuzzy clustering and fuzzy clustering: A comparative experimental study," *Fuzzy Sets Syst.*, vol. 161, no. 4, pp. 522–543, Feb. 2010.

[29] D. E. Gustafson and W. C. Kessel, "Fuzzy clustering with a fuzzy covariance matrix," in *Proc. 17th Symp. IEEE Conf. Decsion Control*, San Diego, CA, Jan. 1978, pp. 761–766.

[30] O. Nelles, "Axes-oblique partitioning strategies for local model networks," in *Proc. Int. Symp. Intell. Control*, Munich, Germany, Oct. 2006, pp. 2378–2383.

[31] G. Karer, G. Mušič, I. Škrjanc, and B. Zupančič, "Hybrid fuzzy model-based predictive control of temperature in a batch reactor," *Comput. Chem. Eng.*, vol. 31, no. 12, pp. 1552–1564, Dec. 2007.

[32] C. Hametner and S. Jakubek, "Comparison of EM algorithm and particle swarm optimisation for local model network training," in *Proc. IEEE Conf. Cybern. Intell. Syst.*, Singapore, Jun. 2010, pp. 267–272.

[33] S. K. Sharma, S. McLoone, and G. W. Irwin, "Genetic algorithms for local model and local controller network design," in *Proc. Amer. Control Conf.*, vol. 2. Anchorage, AK, May 2002, pp. 1693–1698.

[34] M. Sugeno and G. T. Kang, "Structure identification of fuzzy model," *Fuzzy Sets Syst.*, vol. 28, no. 1, pp. 15–33, Oct. 1988.

[35] T. A. Johansen, "Identification of non-linear system structure and parameters using regime decomposition," *Automatica*, vol. 31, no. 2, pp. 321–326, Feb. 1995.

[36] R. Shorten and R. Murray-Smith, "Side-effects of normalising basis functions in local model networks," in *Multiple Model Approaches to Modelling and Control*, R. Murray-Smith and T. A. Johansen, Eds. London, U.K.: Taylor & Francis, 1997.

[37] R. Babuška and H. B. Verbruggen, "An overview of fuzzy modeling for control," *Control Eng. Practice*, vol. 4, no. 11, pp. 1593–1606, Nov. 1996.

[38] R. Murray-Smith, "A fractal radial basis function neural net for modeling," in *Proc. Int. Conf. Autom., Robot. Comput. Vis.*, vol. 1. 1992, pp. 1–6.

[39] T. A. Johansen and R. Babuška, "On multiobjective identification of Takagi-Sugeno fuzzy model parameters," in *Proc. 15th IFAC World Congr.*, Barcelona, Spain, 2002, pp. 847–860.

[40] O. Nelles, *Nonlinear System Identification*. Berlin, Germany: Springer-Verlag, 2001.

[41] R. Murray-Smith and T. A. Johansen, "Local learning in local model networks," in *Proc. 4th IEEE Int. Conf. Artif. Neural Netw.*, Cambridge, U.K., Jun. 1995, pp. 40–46.

[42] R. Murray-Smith and T. Johansen, "Local learning in local model networks," in *Multiple Model Approaches to Modelling and Control*, R. Murray-Smith and T. A. Johansen, Eds. London, U.K.: Taylor & Francis, 1997.

[43] R. Murray-Smith, "A local model network approach to nonlinear modeling," Ph.D. dissertation, Dept. Comput. Sci., Univ. Strathclyde, Glasgow, U.K., 1994.

[44] M. E. Futschik and B. Carlisle, "Noise-robust soft clustering of gene expression time-course data," *J. Bioinf. Comput. Biol.*, vol. 3, no. 4, pp. 965–988, Aug. 2005.

[45] M. Kowal and J. Korbicz, "Segmentation of breast cancer fine needle biopsy cytological images using fuzzy clustering," in *Advances in Machine Learning I, Studies in Computational Intelligence*, J. Koronacki, Z. W. Ras, S. T. Wierzchon, and J. Kacprzyk, Eds. Berlin, Germany: Springer-Verlag, 2010.

[46] M. Moshtaghi, S. Rajasegarar, C. Leckie, and S. Karunasekera, "An efficient hyperellipsoidal clustering algorithm for resource-constrained environments," *Pattern Recognit.*, vol. 44, no. 9, pp. 2197–2209, Mar. 2011.

[47] B. Hartmann, O. Nelles, I. Škrjanc, and A. Sodja, "Supervised hierarchical clustering (SUHICLUST) for nonlinear system identification," in *Proc. IEEE Symp. Comput. Intell. Control Autom.*, Nashville, TN, Mar.–Apr. 2009, pp. 41–48.

[48] B. Hartmann, O. Nelles, I. Škrjanc, and A. Sodja, "Global supervised and local unsupervised learning in local model networks," in *Proc. IFAC Symp. Syst. Ident.*, Saint-Malo, France, Jul. 2009, pp. 1517–1522.

[49] S. Jakubek and C. Hametner, "Identification of neurofuzzy models using GTLS parameter estimation," *IEEE Trans. Syst., Man, Cybern., Part B: Cybern.*, vol. 39, no. 5, pp. 1121–1133, Oct. 2009.

[50] S. A. Billings and H.-L. Wei, "Sparse model identification using a forward orthogonal regression algorithm aided by mutual information," *IEEE Trans. Neural Netw.*, vol. 18, no. 1, pp. 306–310, Jan. 2007.

[51] M. Meloun, T. Syrovy, and A. Vrana, "The thermodynamic dissociation constants of losartan, paracetamol, phenylephrine and quinine by the regression analysis of spectrophotometric data," *Anal. Chim. Acta*, vol. 533, no. 1, pp. 97–110, Mar. 2005.

Luka Teslić received the B.Sc. degree in electrical engineering from the University of Ljubljana, Ljubljana, Slovenia, in 2006. He is currently pursuing the Ph.D. degree in mobile robotics with the same university.

His current research interests include mobile robot localization, map building, and fuzzy system identification.

Benjamin Hartmann received the M.S. degree in mechanical engineering from the University of Siegen, Siegen, Germany, in 2007. He is currently pursuing the Ph.D. degree in automatic control engineering with the Department of Mechanical Engineering, Institute of Automatic Control and Mechatronics, University of Siegen.

He is currently a Research Assistant with the Department of Mechanical Engineering, Institute of Automatic Control and Mechatronics, where he works with Prof. Nelles. His current research interests include experimental nonlinear statics and dynamic modeling and design of experiments.

Oliver Nelles received the M.Sc. degree in electrical engineering in 1993 and the Ph.D. degree from the Institute of Automatic Control in 2000, both from the Technical University of Darmstadt, Darmstadt, Germany.

He was with Prof. Isermann at the Institute of Automatic Control from 1994 to 2000. With a DAAD scholarship, he worked as a Post-Doctoral Researcher with Prof. Tomizuka at the Mechanical Engineering Department, University of California, Berkeley. From 2000 to 2004, he was a Group Leader in the development of transmissions with SiemensVDO Automotive, Regensburg, Germany. He is currently a Professor of automatic control and mechatronics with the Department of Mechanical Engineering, University of Siegen, Siegen, Germany. He is the author of the book *Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models* (New York: Springer, 2001). His current research interests include experimental nonlinear statics and dynamic modeling, data mining, and design of experiments.

Igor Škrjanc received the B.Sc., M.Sc., and Ph.D. degrees in electrical engineering from the Faculty of Electrical and Computer Engineering, University of Ljubljana, Ljubljana, Slovenia, in 1988, 1991, and 1996, respectively.

He is currently a Professor of automatic control with the Faculty of Electrical Engineering and Computer, and the Head of the research program modeling, simulation, and control. His current research interests include intelligent, predictive control systems, and autonomous mobile systems.

Dr. Škrjanc received the Highest Research Award from the Faculty of Electrical Engineering and Computer in 2007, the highest award from the Republic of Slovenia for Scientific and Research Achievements in 2008, and the Zois Award for outstanding research results in the field of intelligent control. He received the Humboldt Research Fellowship for Experienced Researchers from 2009 to 2011.