Technical Report
FLLL–TR–0217

FUZZY.LOGIC.LAB.LINZ

Online Adaptation of Takagi-Sugeno Fuzzy Inference Systems

Edwin Lughofer
Fuzzy Logic Laboratorium Linz-Hagenberg
e-mail edwin.lughofer@jku.at

Abstract — Adaptive algorithms for data-based models are often of fundamental importance in order to identify real-time processes which possess a time-variant behaviour that would make a time-invariant model too inaccurate. Beyond that, an insufficiency of amount, distribution and/or quality of actual recorded measurement data can occur, such that the model cannot meet the expectations at a particular time. In this case, the incorporation of new recorded data into previously generated models can improve the model's accuracy and reduce the bias or model error captured due to original noisy data. In this paper algorithms and strategies for adapting a special kind of data-based models, namely so-called fuzzy inference systems, are demonstrated.

Key words — *adaptive algorithms, time-invariant model, accuracy of a model, bias error, model error, fuzzy inference system*

Johannes Kepler Universität Linz
Fuzzy Logic Laboratorium Linz-Hagenberg

Institut für Algebra, Stochastik und
wissensbasierte mathematische Systeme
Johannes Kepler Universität Linz
A-4040 Linz
Austria

FLLL
Softwarepark Hagenberg
Hauptstrasse 99
A-4232 Hagenberg
Austria

1 Motivation and Requirements

A detailed description of general requirements to adaptive algorithms for data-based models can be found in the first chapter in [11] and can be summarized by the following items:

- Fast identification algorithms for dynamic systems
- Refinement of data-based models at an arbitrary point of time
- More precise rendering of vague linguistic expert knowledge about physical relationships with online data (grey box models)
- Auto-adaptation of empirical models to an adjacent test object

Due to the fact that expert knowledge mostly consists of linguistic rules which can be simply brought into a premise-conclusion form (if ... then) and hence coded in a fuzzy expert system (for practical examples see [12] or [2]), due to the simplicity of re-adjusting a fuzzy expert system manually after being generated from data and due to the drawbacks of adaptive correlation and regression models mentioned in [11], chapter 7, namely:

- Desiderative localization when describing physical relationships
- Deficiencies due to preserving the shape of models when adapting them to a new adjacent test object
- Deficiencies due to the enlargement of models to a new range of influence in the case of describing a new operating condition

adaptive algorithms for fuzzy inference systems were researched, developed and verified due to real-life measurements coming from a large diesel engine. The results including the avoidance of the first two drawbacks mentioned above are stated in chapter 5. The enlargement of fuzzy inference systems was not implemented and tested till now, hence only some theoretical ideas and concepts are given in chapter 4.

2 Problem Statement

2.1 Definition of a Takagi-Sugeno FIS

The starting point is a Takagi-Sugeno fuzzy inference system, whose output is defined by:

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

where

$$\Phi_i(u) = \frac{\mu_i(u)}{\sum_{j=1}^{M} \mu_j(u)} \tag{2}$$

are called *basis functions*, which normalize the degrees of rule fulfillment by using the product t-norm, i.e.

$$\mu_i(u) \quad = \quad \prod_{j=1}^{p} \mu_{ij}(u_j)$$

where u_j is the j-th component in current data vector, hence reflecting the value of the j-th channel and μ_{ij} the membership degree of u_j to the fuzzy set describing the j-th premise part of the i-th rule. The f_i's are the so-called consequent functions of the M rules and are defined by:

$$f_i = w_{i0} + w_{i1}u_1 + w_{i2}u_2 + ... + w_{ip}u_p \tag{3}$$

For a Sugeno controller (as a special case of a Takagi-Sugeno controller) only one constant output value per rule, i.e. w_{i0} is applied. Arbitrary fuzzy sets can be chosen depending on the special task and behaviour of the FIS, most common are B-splines of several orders (e.g. 2nd order B-splines reflect triangular fuzzy sets), Gaussian functions or sigmoid functions.

2.2 Methods for the Generation of Takagi-Sugeno FIS

In the literature a large amount of different methods exist for generating FIS from data, namely for instance with

- ANFIS: Adaptive Network-based Fuzzy Inference Systems
- genfis2: cluster-based generation of FIS (=Fuzzy Inference System)
- RENO: Regularized Numerical Optimization

The *ANFIS* approach defines a Takagi-Sugeno fuzzy inference system through a Neural Network approach by defining 5 layers:

- Layer 1: fuzzyfication of the input values due to MSFs -> membership degrees
- Layer 2: aggregation of membership degrees due to an appropriate t-norm applied in the premise parts
- Layer 3: evaluation of the *basis functions* by normalization of aggregated membership degrees
- Layer 4: weighting of basis functions with linear (=> Takagi-Sugeno system) or constant (=> Sugeno system) consequent functions
- Layer 5: evaluation of output values by applying equation (1).

The basic approach for generating the FIS is denoted by a hybrid method, i.e. a combination of gradient descent method with LS. While firstly the nonlinear parameters in the fuzzy sets are fixed and the linear parameters in the consequent (or constant) functions are estimated with conventional global LS method (see [10]), afterwards the nonlinear parameters are calculated by a

special gradient descent approach as described in [5] (for further details about ANFIS see also [5] and [6]).

Genfis2 performs cluster analysis in order to generate fuzzy sets due to projections of clusters in the multidimensional space onto the corresponding axis. For cluster analysis itself different methods can be applied, which results in different outlooks and properties of the fuzzy sets. The rule extraction method of *genfis2* accomplishes this by extracting a set of rules that models the data behaviour. It first uses the subtractive clustering algorithm (see [15]) to determine the number of rules and antecedent membership functions and then uses global linear least squares estimation to determine each rule's consequent equations. It returns a Takagi-Sugeno-type fuzzy inference system structure that contains a set of fuzzy rules to cover the feature space. For a detailed inspection of genfis2 algorithm refer to [17]. As a matter of fact, cluster can strongly overlap each other when they are projected to the axis in order to obtain fuzzy sets, hence it is obvious, that linguistic interpretability of the fuzzy-sets cannot be ensured. As opposed to that, the advantage of this method is that it generates Gaussian membership functions as fuzzy sets, defined by

$$\mu(u) = e^{-\frac{1}{2}\frac{(u-c)^2}{\sigma^2}} \tag{4}$$

where c is the center and σ the width of the Gaussian function, which have, by nature, infinite support, therefore for every special input vector a membership degree to each fuzzy set greater than 0 is computed, and hence every rule in the rule-base fires. This leads to the possibility of generating only a couple of rules, describing the relationship between input and output channels accurate enough.

Opposed to *genfis2*, *RENO* or also called *Regularized Numerical Optimization* is based on the minimization of the loss function defined by the summed deviation of process output values to the estimated output values obtained through equation (1). In order to guarantee linguistic interpretability on the input side (Note: for Takagi-Sugeno, respectively Sugeno fuzzy inference systems linguistic interpretability can be achieved only for the input side anyway, while for Mamdani controllers also the output side fulfills this property), constraints for the knots of B-Spline MSFs are added to the optimization problem. With so-called regularization techniques a well-posed problem can be achieved which can be solved by a generalized Gauss-Newton method (see [4] and [14]). The great drawback of using B-Splines instead of Gaussian-shaped membership functions with infinite support arises when decreasing the number of fuzzy sets and hence improving the linguistic interpretability of the FIS, for example with genetic algorithms. In this case holes in the input space can appear when discarding some rules or fuzzy sets. Besides, to guarantee well-defined interpolation over the whole input space, every combination of fuzzy sets has to be taken into account in the rulebase, which leads to an enormous number of rules if applied to a high-dimensional measurement system (as it is the case for engine test benches, for instance).

A comparison of the 2 methods, *genfis2* and *RENO*, applied onto real-life measurement data from an engine test bench can be seen in figure 1.

In table 1 a comparison of the methods with respect to maximal error, average squared error and computation time is stated for the N_MI-PE-LAMBDA relationship, estimated from 1810 data points. Obviously, *genfis2* is superior to *RENO* and *ANFIS* with respect to *bias error*.

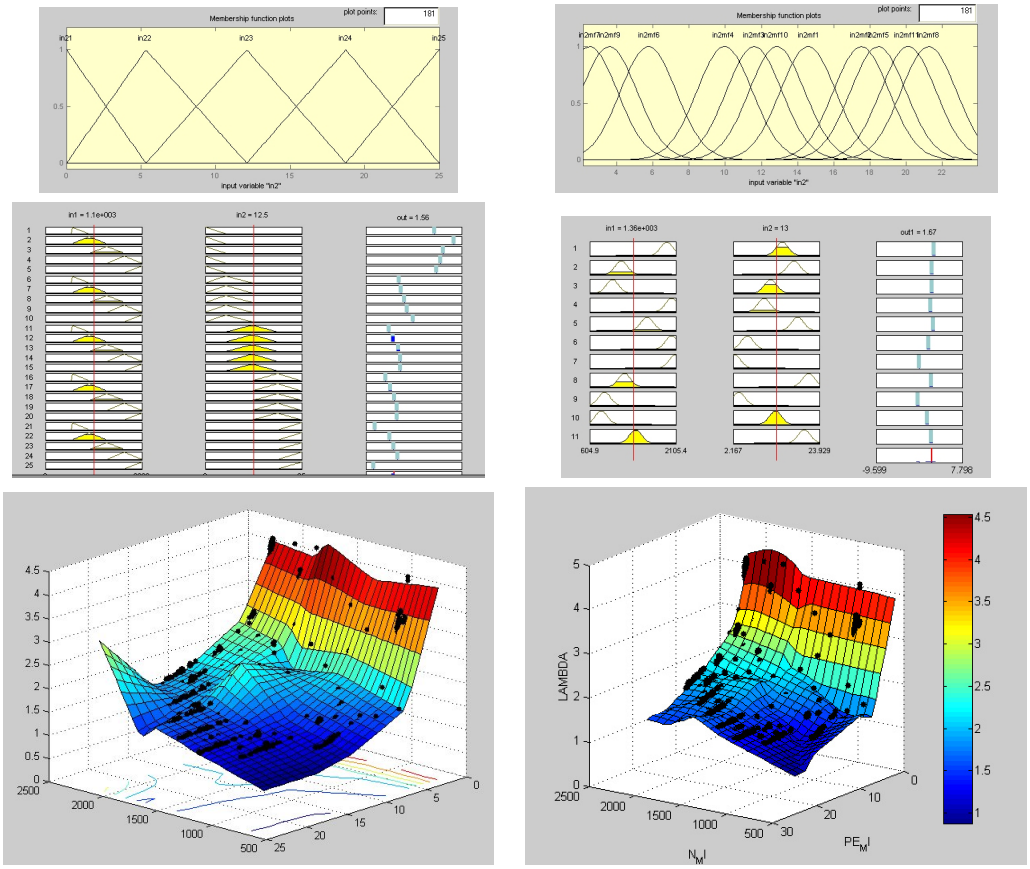

Figure 1: Fuzzy inference systems generated out of engine test bed data, by using 3 channels N_MI, PE and LAMBDA: the left side shows the inference system with the usage of RENO together with Tichonov regularization terms, the right side demonstrates the inference system built up by using genfis2-method, the 2 images below show us the corresponding control surface, dark dots represent the train data points

Method	Max_error	Av_squared_error	Comp. time
RENO	1.012	0.033	12.57
ANFIS	1.38	0.69	3.52
genfis2	0.29	0.029	7.74

Table 1: Comparison of FIS generation methods with respect to maximal error, average squared error and computation time on a PC with 800 Mhz and 256 RAM

3 Basic Theoretical Aspects for Adaptation

3.1 What to Adapt?

First of all we have to consider which parts of a fuzzy inference system can be and should be adapted in order to guarantee an acceptable performance regarding to approximation error and correctness of identification. When inspecting the different layers of a FIS as described in the previous chapter and used by the ANFIS approach, we have to realize that principally 3 different kinds of components can be adapted:

- Premise Parameters: they appear in the input membership functions as centers and widths
- Consequent Parameters: they appear in the rules consequents as output weights
- Rule structure

Consequent parameters are linear parameters, premise parameters are nonlinear ones, hence 2 different approaches for adaptation are described in the following 2 sub-chapters. Optimizing the rule structure can be an important approach, because it allows one to determine the optimal complexity of the fuzzy inference system in order to achieve a better linguistic interpretability, and by that it weakens the curse of dimensionality. Methods such as modified linear subset selection schemes (e.g. *OLS*) or nonlinear global search techniques (e.g. *genetic algorithms*) exist for rule structure optimization (for details see [13]). A fast online adaptation strategy for rules structures is not dealt in this paper, but can be found in [16]. By the way, if applying *genfis2* (see previous chapter) for generating the initial model, mostly the resulting model possess a rule structure with only a handful of rules, such that rule structure optimization is not the crucial point for adaptation.

3.2 Global versus Local Adaptation of the Rules Consequents

While in the global estimation approach all linear consequent parameters are estimated simultaneously, the local estimation approach estimates the consequent parameters for each rule and hence for each local linear model separately. From the definition of a Takagi-Sugeno fuzzy inference system (see chapter 2.1), in the case of global estimation approach the regression matrix X for N measured data samples becomes

$$X = [X_1 X_2 ... X_M] \tag{5}$$

where M are the number of rules in the system and

$$X_i = \begin{bmatrix} \Phi_i(u(1)) & u_1(1)\Phi_i(u(1)) & u_2(1)\Phi_i(u(1)) & ... & u_p(1)\Phi_i(u(1)) \\ \Phi_i(u(2)) & u_1(2)\Phi_i(u(2)) & u_2(2)\Phi_i(u(2)) & ... & u_p(2)\Phi_i(u(2)) \\ \vdots & \vdots & \vdots & \vdots & \\ \Phi_i(u(N)) & u_1(N)\Phi_i(u(N)) & u_2(N)\Phi_i(u(N)) & ... & u_p(N)\Phi_i(u(N)) \end{bmatrix} \tag{6}$$

where the Φ_i's are the *basis functions* defined in formula (2), $u_i(j)$ denotes the value of the ith variable (channel) in the jth data point. With this notation the model output as defined in equation (1) becomes

$$\hat{y} = Xw \tag{7}$$

where w is the parameter vector from equation (3) containing only the linear weights w_{ij} obtained from the membership degrees through the multidimensional membership functions. Obviously, these linear parameters can be estimated with LS method as it is described in [10] and adapted through RLS formulas:

$$\hat{w}(k+1) = \hat{w}(k) + \gamma(k)(y(k+1) - x^T(k+1)\hat{w}(k)) \tag{8}$$

with the correction vector

$$\gamma(k) = P(k+1)x(k+1) = \frac{P(k)x(k+1)}{1 + x^T(k+1)P(k)x(k+1)} \tag{9}$$

$P(k+1)$ can be computed recursively by:

$$P(k+1) = (I - \gamma(k)x^T(k+1))P(k) \tag{10}$$

where $P(k)$ denotes the inverse Hesse matrix $(X^T X)^{-1}$ at time step k and $x(k)$ is the transferred input data vector, transferred due to the regressors containing *basis function* weighted with data values. In order to forget older data points (for instance in the case of auto-adaptation to a new adjacent test object) a forgetting factor λ with $0 < \lambda < 1$ can be incorporated into above formula (see [11]).

Opposed to the global estimation, for the local approach M separate local estimations and consequently adaptations are carried out for the $p+1$ parameters of each local linear model, hence for each rule. The model parameter vector for each of these $i = 1, ..., M$ estimations is

$$w_i \quad = \quad [w_{i0} w_{i1} ... w_{ip}]^T$$

The corresponding regression matrices are given by

$$X_i = \begin{bmatrix} 1 & u_1(1) & u_2(1) & ... & u_p(1) \\ 1 & u_1(2) & u_2(2) & ... & u_p(2) \\ \vdots & \vdots & \vdots & \vdots & \\ 1 & u_1(N) & u_2(N) & ... & u_p(N) \end{bmatrix} \tag{11}$$

Obviously, the regression matrices of all local linear models $i = 1, ..., M$ are identical, since the entries of X_i do not depend on i. A local linear model with the output $\hat{y_i} = [y_i\hat{(}1)y_i\hat{(}2)...y_i\hat{(}N)]^T$

$$\hat{y_i} \quad = \quad X_i w_i \tag{12}$$

is valid only in the region where the associated *basis function* $\Phi_i(.)$ is close to 1. This will be the case close to the center of $\Phi_i(.)$. Data in this region is highly relevant for the estimation of w_i. As the *basis functions* decreases the data becomes less relevant for the estimation and adaptation of w_i and more relevant for the estimation and adaptation of the neighboring models. Consequently it is straightforward to apply a weighted least squares optimization where the weighting factors are denoted by the *basis function* values, i.e.

$$\sum_{j=1}^{N} \Phi_i(u(j))e^2(j) \longrightarrow \min_{w_i} \tag{13}$$

where $e(j) = y(j) - \hat{y(j)}$ represent the model errors. With the weighting matrix

$$Q = \begin{bmatrix} \Phi_i(u(1)) & 0 & ... & 0 \\ 0 & \Phi_i(u(2)) & ... & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & ... & \Phi_i(u(N)) \end{bmatrix}$$

a weighted LS method (*WLS*) can be performed in order to estimate the linear consequent parameters $\hat{w}_i$ for the ith rule. Consequently, the adaption for the local approach can be carried out by *RWLS = recursive weighted least squares* as described in [11], chapter 4.2. Forgetting older data points with local adaptation approach can be achieved by a combination of *RWLS* and incorporation of forgetting factor λ, hence by the following formulas:

$$\hat{\beta}(k+1) = \hat{\beta}(k) + \gamma(k)(y(k+1) - x^T(k+1)\hat{\beta}(k)) \tag{14}$$

$$\gamma(k) = P(k+1)x(k+1) = \frac{P(k)x(k+1)}{\lambda/q(k) + x^T(k+1)P(k)x(k+1)} \tag{15}$$

$$P(k+1) = \frac{1}{\lambda}(I - \gamma(k)x^T(k+1))P(k) \tag{16}$$

Note: As *RLS* and all related methods such as *recursive weighted least squares* or *recursive (weighted) least squares with data forgetting* are exact methods, the adaptation can be performed pointwise. In practice, in order to meet online identification requirements, a special strategy for FIS extensions and enlargements need to be carried out, which demand data blocks with at least a few dozen data points (see chapter 4).

Due to neglecting the interactions between the local models (because of overlapping fuzzy sets), the *bias error* indeed increases when applying local estimation approach, but a lot of improvements for FIS generation and adaptation can be achieved, such as:

- Higher flexibility for adjoining fuzzy sets when new operating conditions for the measurement process occur (see also chapter 4). When global estimation method is used, the global inverse Hesse matrix in RLS formulation has to be set back to αI, because it completely changes if new fuzzy sets and hence rules and consequent parameters are added. This setting back, which can be avoided by the local estimation approach, results in a worse approximation (see chapter 5.1 and also [11], chapter 6.1). Moreover, this property of FIS extension entails an additional advantage, namely that new relationships can be learned in one operating regime while the old information is conserved in all others.

- Higher flexibility when adjusting premise parameters in a preliminary step with nonlinear optimization methods: a change in the premise parameter of the membership function causes a change in the *basis functions* and hence in the inverse Hesse matrix again, when applying global estimation approach. As for the local estimation the local Hesse matrices have the same appearance for all local linear functions and do not depend on *basis function* values, the adjusting of fuzzy sets do not influence local adaptation.

- Improvement of numerical stability of LS and RLS method, as dealing with small and hence better-conditioned Hesse matrices.

- Acceleration of LS and RLS, above all for systems with a high dimensionality and a large amount of rules. Global estimation approach takes all the consequent parameters joined

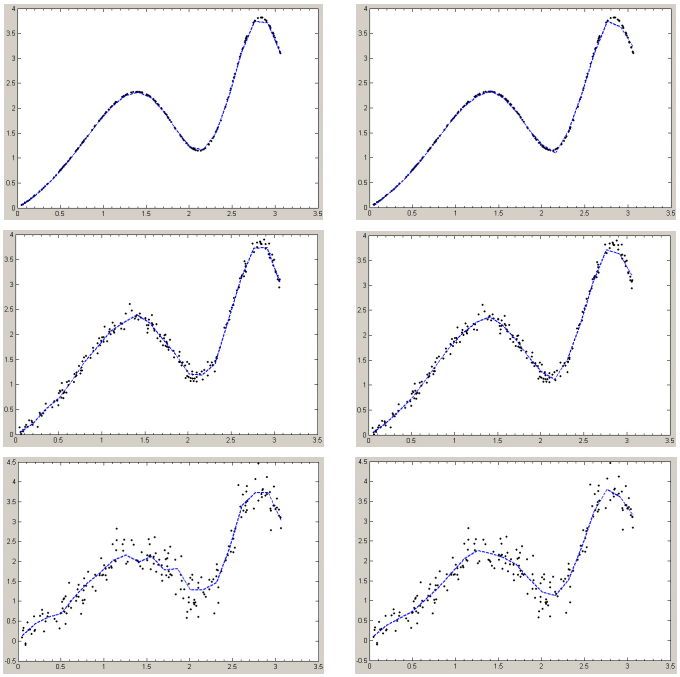

Figure 2: The upper 2 images represents sin-curve approximation when applying it to noise-free data, middle row: a white noise with standard deviation of 0.1 is added, lower row: a white noise with standard deviation of 0.3 is added; for all rows on the left side global estimation, on the right side local estimation is performed; in the case of a low or a high noise level the local estimation is superior to global estimation

together in a matrix as input, hence the number of parameters is $M(p+1)$, where M is the number of rules and p the number of inputs. Therefore, the computational complexity is $O((M(p+1))^3)$ for LS and $O((M(p+1))^2)$ for RLS. Opposed to that, local estimation approach performs M LS estimations or RLS adaptations with matrices of dimension size $(p+1)$. Therefore the computational complexity is just $O(M(p+1)^3)$ for LS and $O(M(p+1)^2)$ for RLS.

- Local estimation is even superior to global estimation with respect to the *bias error*, if a high noise level is present in the data. The reason lies in an inherent regularization effect which can be achieved with local estimation. In figure 2 an examination of this property is demonstrated, the corresponding *bias errors* are stated in table 3.2, as measure value for the *bias error APE = Average Percent Error* as defined in [11], chapter 6.1, was used. The approximation is carried out with a fuzzy inference system generated by *genfis2*. While for noise-free data the approximation of the formula $f(x) = \sin(x^2) + x$, with global estimation approach leads better results with respect to *bias error*, for a low and above all a high noise level the opposite is the case. Besides, the curvature generated by global estimation approach gets non-differentiable as edges appear when adding high noise level (distributed noise with 0.3 standard deviation) to the original function.

Method	Noise Level	APE
global est.	no noise	0.065
local est.	no noise	1.285
global est.	$\sigma = 0.1$	2.489
local est.	$\sigma = 0.1$	1.325
global est.	$\sigma = 0.3$	12.46
local est.	$\sigma = 0.3$	5.297

Table 2: *Bias errors* represented by *APE* for the 2 methods when approximating $\sin(x^2) + x$ with different noise levels

3.3 Adaptation of the Premise Parameters

The premise parameters appear more or less in the MSFs, i.e. mostly as widths and centers of the fuzzy sets. As they are generally non-linear parameters (for instance in the case of Gaussian MSFs - see equation (4) - which are widely used and also generated by *genfis2*), the estimation and adaptation is a little bit more sophisticated as for linear consequent parameters. The starting point is an optimization problem of the general loss function for a Takagi-Sugeno fuzzy inference system, namely

$$e = \sum_{k=1}^{N} \frac{1}{2}(\hat{y_k} - y_k)^2 \longrightarrow \min \tag{17}$$

where $\hat{y_k}$ is the output of the FIS for data point k as defined in equation (1) and contains the non-linear parameters Ψ_{ij} which should be adapted. Due to the consideration, that for ensuring interpretable extrapolation behaviour the widths of all fuzzy sets for one input dimension should be the same (see chapter 4), the parameters describing the widths of the fuzzy sets are not taken into account for adaptation, so Ψ_{ij} denotes the center of the jth fuzzy set in the ith dimension. The minimization problem defined above is a so-called *non-linear least squares problem* which can be efficiently solved with an appropriate gradient-based optimization method such as steepest descent, Newton's method or the Quasi-Newton approach. As the Hesse matrix can be approximated through (see also [13])

$$H \quad = \quad J^T J \tag{18}$$

where J is the Jacobian matrix defined by

$$J = \begin{bmatrix} \partial e(1)/\partial \Psi_{11} & ... & \partial e(1)/\partial \Psi_{nn} \\ \vdots & \vdots & \vdots \\ \partial e(N)/\partial \Psi_{11} & ... & \partial e(N)/\partial \Psi_{nn} \end{bmatrix} \tag{19}$$

a computation with low computational cost is possible also for Newton's method which achieve a faster convergence rate among others. In order to ensure that the centers Ψ_{ij} of the membership functions do not fall outside the input range of the corresponding input dimensions during estimation and adaptation, restriction in the form of $\min_i \leq \Psi_{ij} \leq \max_i$ (where min_i and max_i denotes the lower and upper bound of the input variable) have to be incorporated into the minimization problem (17). This leads us to a constrained nonlinear least squares optimization problem

which can be reformulated through Kuhn-Tucker equations by using Lagrange multipliers and numerically solved by *sequential quadratic programming (SQP)* (see [1]).

Note: If using Gaussian membership functions together with product t-norm, as it is done by *genfis2* method, the loss function defined in equation (17) is continuously differentiable and hence the gradients in the Jacobi matrix well defined over the whole input space.

Estimation of nonlinear parameters can only be solved with iterative methods and hence never exactly, therefore adaptation of nonlinear parameters cannot be deduced in a way as it is possible for linear parameters, so it is based on the same algorithms, more or less. The difficulty in adaptation, opposed to estimation, lies in the fact of catching the right point of time for the termination of the iterative estimation algorithm for the new data block, otherwise the model will be adapted to the new data points too strong while forgetting the older ones completely, even if choosing the premise parameters of the old FIS as starting point for the iteration. Therefore, with the termination criteria for the iterative method the strongness of the forgetting of data points can be controlled. Besides, as difficulties with local optima, extremely slow convergence etc. appear, mostly online adaptation of nonlinear parameters can be not carried out in a reliable manner (see also chapters 5 and 6).

4 A Generic Strategy for Online Adaptation in an Identification System

Due to all these reasons mentioned in the chapter 2.2 *genfis2* is used as the starting point for generating the initial data-based fuzzy model. During an online test procedure, the amount of data which should be collected for initial generation depends strongly on the distribution and the quality of data, i.e. the amount of faulty-free data points in the data set. Theoretically, *genfis2* works, if only $2 * m$ data points are available, where m denotes the dimensionality of the FIS, but for practical usage it turned out that approximately $20 * m$ well-distributed and faulty-free data points are needed to be able to guarantee an appropriate valid initial fuzzy inference system. Above the reasons mentioned above, *genfis2*, as it applies *subtractive clustering* for obtaining fuzzy sets, ensures interpretable extrapolation behaviour, which can be an important fact if the some new incoming points appear outside previously defined input ranges. Interpretable extrapolation behaviour can be only guaranteed if one of the following 2 cases holds:

- The standard deviations of all membership functions are equivalent in each dimension
- The standard deviations of the outermost membership functions are equivalent or larger than the standard deviations of all inner membership functions in each dimension

In general, subtractive clustering produces ellipsoid clusters, but for each dimension the axis of all ellipses (clusters) possess the same length, hence the first property of the enumeration above is fulfilled.

As soon as a sufficient amount of train data points is available and the initial fuzzy inference system is generated with *genfis2* (see [17]), the adaptation of the rules consequents and the premise parameters as it is described in chapter 3 starts. As in real-life often high-dimensional measurement data should be handled, a dimension reduction technique or variable selection method (see

[9] and [3]) is carried out for each measurement channel (as target channel) in order to receive most significant channels for approximating the target channel. The complete strategy for adaptation of FIS is stated in flowchart 3. The first process block in the flowchart represents decision making if new incoming data points are accepted for adaptation or not. Decision making can be carried out by either using a pre-filtering algorithm on historic data for the offline case or by applying physical-based or linguistic-based check methods to the new incoming data points for the online case. Both strategies will be dealt in a more detailed way in a further technical report when talking about "Fault Detection with data-based Models". After this first process block, the complete workflow consists of more or less 3 parts arisen from 3 different operating situations representing 3 different requirements, namely:

- Auto-adaptation of models to a new adjacent test object
- Refinement of models at an arbitrary point of time (concerns white box and black box models)
- Adaptation of models from the same test object during an online cycle

The process block "Check ranges and input channels" appears for all operating situations and calculates the ranges of the input channels for the actual model due to current data block. In the case of online adaptation and refinement of previously generated models, if one of these new ranges lies outside the previously calculated one, probably a new operating condition takes place and the FIS is extended by placing new fuzzy sets into this newly "occupied" region. After that, a conventional adaptation of the FIS (without forgetting) as described in 3 is performed. A criteria for placing new fuzzy sets for the affected input variable can be defined by

$$max_{new} \quad > \quad max_{old} + (max_{old} - min_{old}) rad_{inf}$$

for the upper region or for the lower region by

$$min_{new} \quad < \quad min_{old} - (max_{old} - min_{old}) rad_{inf}$$

where max_{new} denotes the maximum value over the new data block with respect to the specified input variable, min_{old} and max_{old} minimum and maximum values over previous data blocks with respect to the specified input variable. rad_{inf} denotes the radius of influence of one cluster in the affected input dimension and lies always between 0 and 1. A high value reflects a rough partition of the input space in separate clusters and hence in separate fuzzy sets, a small value a granular partition. This parameter is used by the initial FIS generation as input parameter, therefore the same value is used for adaptation in order to keep interpretable extrapolation behaviour as the new added fuzzy sets have the same widths as the old ones. In the case of a pointwise adaptation of the models (when data blocks contain just one point), which is possible if updating only the rules consequents (see chapter 3.2), some data points need to be collected first in order to specify new input ranges and to extend the FIS.

Remark: Of course, data points lying outside of previously estimated input ranges can also be caused by damages in the measurement system and hence do not reflect a new operating condition. This separation is more or less performed by the first data block, such that we can assume that all points are faulty-free afterwards.

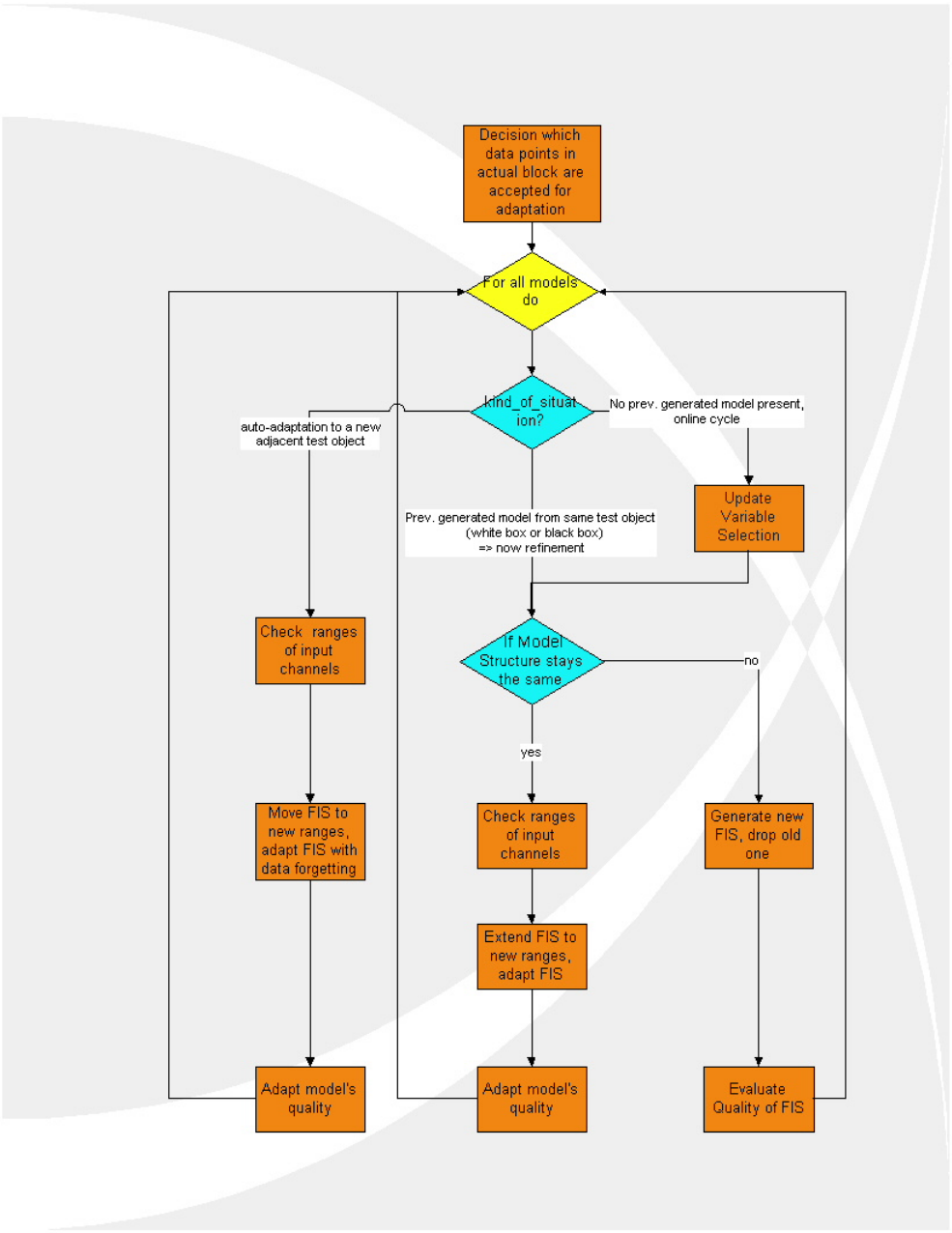

Figure 3: Core Strategy for updating fuzzy expert systems

In the case of the auto-adaptation of models to a new adjacent test object (left part of flowchart), usually the new incoming data points recorded for the new test object possess completely different ranges, while the shape of the model should be still the same (otherwise an auto-adaptation is not valid - note: the operator with physical knowledge and experience has to know if a model of an adjacent test object is valid to be taken as initial model for the new test object). So, in that case the complete model is shifted to the new range of influence denoted by recently incoming data points. The shifting can be simply performed by transferring the centers of the Gaussian MSFs to the new range for each input variable while letting the range of influence, i.e. the sigmas, fixed, which does not change the shape and outlook of the model, obviously. Pointwise adaptation does not play any role here, because first a certain amount of data points need to be gathered before being able to perform an auto-adaptation at all. After that, adaptation of the FIS with forgetting of older data points (hence data points which were recorded for the old test object) as described in 3 is carried out. With this forgetting a shift along the direction of the target channel can be achieved, too.

The process block "Update Variable Selection" is only needed, if there is no previous model present, neither a black box model due to a previous test run nor a white box model due to a-priori physical or linguistic knowledge. In all other cases the model structures should be optimal or at least suboptimal (see [3] for the estimation of the target channels appearing in the measurement scheme. By using the T-values approach as described in [9], the model structure for a specific target channel y, can be updated by dividing the T-values, defined by:

$$T_{yj} \quad = \quad \frac{\hat{\beta}_{yj}}{s\sqrt{d_i}} \tag{20}$$

into 3 different parts, namely β_{yj}, s and d_i, which can be all updated either with RLS method as described in (8) or by recursive variance calculation approach (for details see [11]). If a model structure changes for the approximation of a specific target channel, a new FIS is generated on the basis of the recently obtained model structure, while the older model is dropped and not taken into account any longer. Note: usually, if collecting enough data points before the generation of the initial fuzzy expert system, model structures will change rarely over time, such that also an oscillating behaviour, which would entail a high computational complexity, can be excluded.

For updating the qualities of fuzzy inference systems, the so-called *r-squared* defined by:

$$R^2 = \frac{ssr}{sst} \tag{21}$$

where $ssr = \sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2$ and $sst = \sum_{i=1}^{n}(y_i - \bar{y})^2$ and $\bar{y}$ the mean value of output target channel y, or the measure value *r-squared-adjusted*, which extends *r-squared* by taking into account the degrees of freedom and can be evaluated through:

$$R^2_{adjusted} = 1 - \frac{(n-1)(1-R^2)}{n-m} \tag{22}$$

where m denotes the degrees of freedom, the recursive variance formulation can be applied again, as both quality values can be easily divided into parts, which have the same appearance as the traditional variance formula.

5 Validation and Results

In this chapter an overview about adaptation results concerning *Takagi-Sugeno fuzzy inference systems* with respect to model behaviour, quality of models and comparison of computation time between complete generation with *genfis2* and adaptation with the algorithms described in chapter 3 is demonstrated. The results were achieved during a large amount of test runs with real-life measurement data coming from engine test benches. In order to verify and validate the model's behaviour a MATLAB-application with a GUI was written, which performs *genfis2* and adaptation algorithms and visualizes both, models appearance and corresponding data points from which the models were generated or adapted. Screenshots of the results are demonstrated in the chapter below; in all figures dark data points represent data from the last loaded data block, while light data points represent data from former loaded data blocks, the block size for each data block can be parameterized, its value is mentioned separately for each example. Moreover, the data points which were recorded during a real-life test procedure for a large diesel engine are temporally ordered, such that a simulation of online adaptation as it appears in practice is performed throughout all test runs.

5.1 Adaptation Results without Forgetting

In figure 4 on the upper picture 3 fuzzy models with respect to the measured physical channels N_MI (rotation speed) and POEL (pressure of the oil) for a large diesel engine at an engine test bench are shown, each with a dashed line: starting with a model obtained through *genfis2* (initial model) from a data block with 100 data points and updating this model 2 times with adaptation algorithms for rules consequent parameters, each time after having loaded another 100 points. Besides, a dotted line is shown, which reflects the full train model for the complete data set of 300 points. On the middle picture, a comparison between the full train models achieved with LS (continuous line), *genfis2* with global estimation (dashed line) and *genfis2* with local estimation (dotted line) of the rules consequents is demonstrated. As stated in table 3, global estimation approach performs the best approximation with respect to *APE = average percent error*, while a conventional LS method for achieving a *global correlation model* is not able to adjust its shape to different local regions and hence results in the worst approximation error. The lower picture represents resulting fuzzy models when applying global estimation approach together with a setting back of the inverse Hesse matrix to αI with $\alpha = 1000$ after having loaded 100 data points and generated the initial model. More or less, this procedure can be seen as a simulation for the refinement of a white box model based on expert knowledge with data, because in that case only the parameters and not the inverse Hesse matrix of the rules consequents are a-priori known.

In all 3 pictures it can be comprehended that global estimation approach cause a stronger curvature of the fuzzy model than local estimation. This circumstance is no surprise, as global estimation takes into account the interactions between the local linear models, i.e. the overlapping of fuzzy sets. This interaction is more or less neglected in the local estimation approach, hence the fuzzy model gets more flat, but also more robust against noise levels. As global estimation approach causes a lower *average percent error* (see table 3) than local estimation, we can assume (due to considerations and tests with artificially generated noisy data as described in chapter 3.2) that the noise level in both channels, N_MI and POEL is not very high in this data set.

Table 3 shows us also the computation time needed for each method. Surprisingly, although

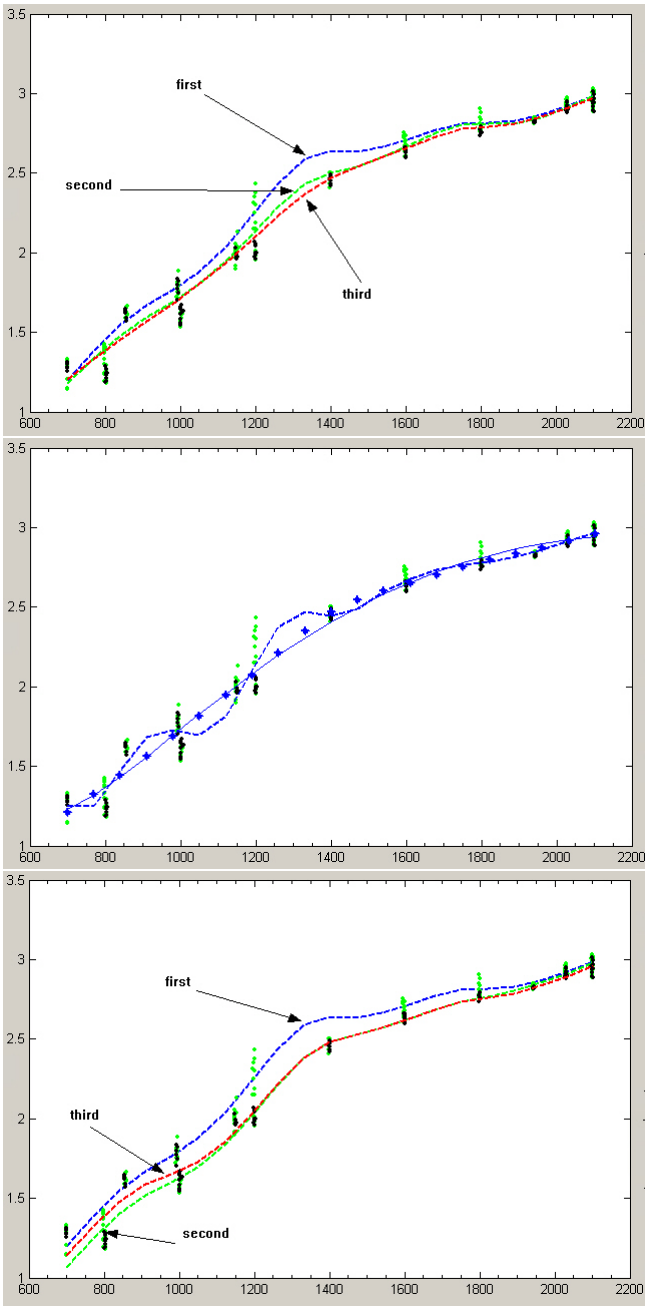

Figure 4: Upper image: 3 fuzzy models between measure N_MI and POEL of a diesel engine, the second and the third obtained through updating with global adaptation approach, Middle image: comparison of full train models with LS (continuous line), *genfis2* with global estimation (dashed line) and *genfis2* with local estimation (dotted line) of the rules consequents, Lower Image: Grey box adaptation with global adaptation approach

Method	**Model**	**APE**	**Comp. Time**
full train with LS (300 points)	$POEL = -80.58 - 0.004 * N\_MI + 5712.62 * 1/N\_MI + 11.68 * log(N\_MI)$	3.69%	0.25 sec.
full train with *genfis2*, global est. (300 points)	linguistic interpretable FIS	3.27%	0.46 sec.
full train with *genfis2*, local est. (300 points)	linguistic interpretable FIS	3.54%	0.7 sec.
global adaptation of rules consequents (100+100+100)	linguistic interpretable FIS	3.54%	0.42 sec.
local adaptation of rules consequents (100+100+100)	linguistic interpretable FIS	3.58%	0.57 sec.
grey box adaptation with global approach	linguistic interpretable FIS	3.72%	0.42 sec.
hybrid method for grey box adaptation = adaptation of centers of fuzzy sets + global adaptation of rules consequents	linguistic interpretable FIS	3.59%	7.92 sec.
hybrid method for black box adaptation = adaptation of the centers of fuzzy sets + local adaptation of rules consequents	linguistic interpretable FIS	5.7%	20.93 sec.

Table 3: Summary about model parameters, model quality and computation time obtained through full train with LS, *genfis2* with global and local estimation, global and local adaptation, grey box adaptation with global approach and the hybrid method for grey box adaptation (=adaptation of premise parameters first and afterwards adaptation of consequent parameters), for global as well as for local estimation

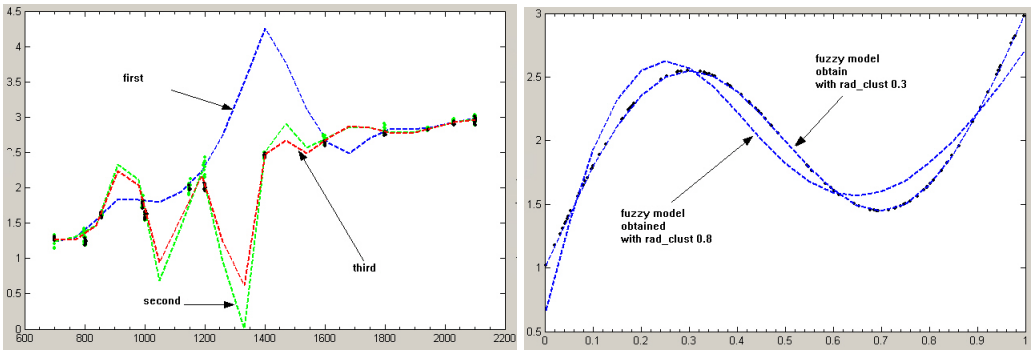

Figure 5: Left image: a too small chosen value for parameter *radius_cluster* (here 0.2) for the initial FIS leads to a so-called "overlocalization" of the model (again for channels N_MI and POEL as it is done in figure 4), such that data holes lead to an unintentional behaviour of the curvature, right image: a too big chosen value for parameter *radius_cluster* (here 0.8) for the initial FIS leads to a bad approximation of $\sin(2\pi u) + 2u$ as APE increases from 0.23% to 7.34% when shifting *radius_cluster* from 0.3 to 0.8

the theoretical deduced computational complexity for global estimation and adaptation of the rules consequent parameters is much higher than for local estimation and adaptation (see chapter 3.2), the tests in MATLAB achieved better results for the global estimation and adaptation. Since now, we cannot completely explain this occurrence, we just assume that in the case of fuzzy models possessing a small amount of rules the factor M plays a less crucial role than the incorporation of a weighting matrix as it has to be performed for the local estimation approach. Additionally, table 3 shows us another surprising result: in the case of a grey box adaptation, the hybrid method, so first adjusting centers of fuzzy sets with a nonlinear optimization technique as described in chapter 3.3 and then performing global adaptation of the rules consequents, achieves a better *APE* than with applying the global adaptation alone. Unfortunately, computation time explodes by a factor of more than 10 if applying nonlinear optimization techniques. By the way, local adaptation together with adjustment of fuzzy sets was not target-oriented. Here it should be also mentioned that the adaptation of the widths of the fuzzy sets was not taken into account because of extrapolation reasons stated in chapter 4.

In all the examples for generating and updating fuzzy models as shown in figure 4 the parameter *radius_cluster* plays a key role for the generation of the initial model. *radius_cluster* denotes the range of influence of one cluster and hence the range of influence of one fuzzy set for each dimension. If *radius_cluster* is a scalar, this value is applied to all input dimensions. Besides, an appropriate value of *radius_cluster* depends strongly on the given amount and quality of data and on the dimensionality of the approximation problem. A suitable formula for calculating *radius_cluster* in dependency of the input dimension is described in [3], chapter 5. Figure 5 shows us the impact of an inappropriately chosen *radius_cluster* onto the shape, outlook and approximation behaviour of fuzzy models. In the examples above *radius_cluster* was always fixed to the appropriate value of 0.3.

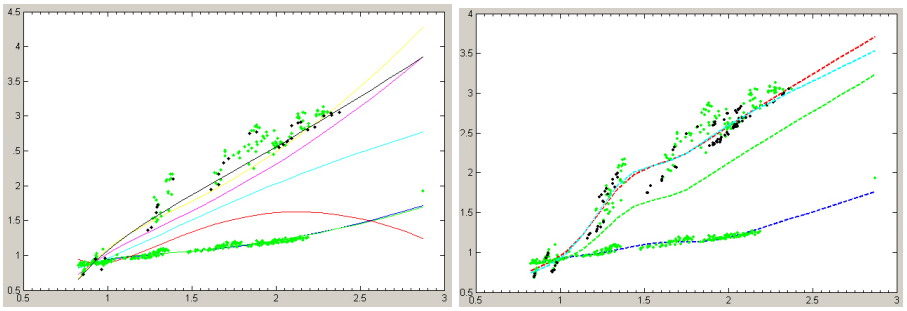

Figure 6: Left image: *global correlation model* generated and adapted with forgetting factor 0.98 -> shape is lost, right image: fuzzy model generated and adapted with forgetting factor 0.98 -> shape is preserved

5.2 Adaptation Results with Forgetting

All the above examples in the previous chapter are based on the assumption that the point of time when a data point is recorded plays no role for the generation of a *Takagi-Sugeno fuzzy inference system*. For practical reasons (see chapter 1) this circumstance eventuates, if the test object changes its behaviour completely over a very short time (for example in the case of a high speed elevator) or an auto-adaptation of the model to a new adjacent test object should be carried out. In these cases older data points have to be forgotten over time. In figure 6 an adaptation with a forgetting factor of 0.98 is compared between the generation and adaptation of a *global correlation model* (left image) with the generation and adaptation of a *Takagi-Sugeno fuzzy inference system* (right image).

From figure 6 it can be easily seen, that the drawbacks of *global correlation models* regarding the shape-holding problem (see [11], chapter 7) can be overcome by using fuzzy inference systems. This property of FIS stems from the fact, that the outlook and location of the fuzzy sets over the input space and their conjunction via rules determine the most significant part of the curvature of the model's surface. So, in the case of an auto-adaptation to an adjacent test object, where preserving the shape of the model is the crucial point, only an adaptation of the rules consequent parameters is performed while letting the nonlinear premise parameters fixed, if the range of all input variables is the same; if not, a shifting of the centers of the fuzzy sets to the new range(s) or even a stretching or contracting of the widths, in order to ensure the same model's curvature, has to be carried out before adaptation.

6 Conclusion and Outlook

Opposed to *correlation* and *regression models* and their adaptation as they are described in [10] and [11], fuzzy inference systems overcome the drawbacks with respect to the shape-holding problem when adapting to a new adjacent test object — centers of local MSFs can be shifted, while letting the range of the influence of the MSFs fixed —, with respect to the localization deficit (by nature) and with respect to the inflexibility of the model structure when applying local estimation and adaptation of the rules consequents (see chapter 3.2).

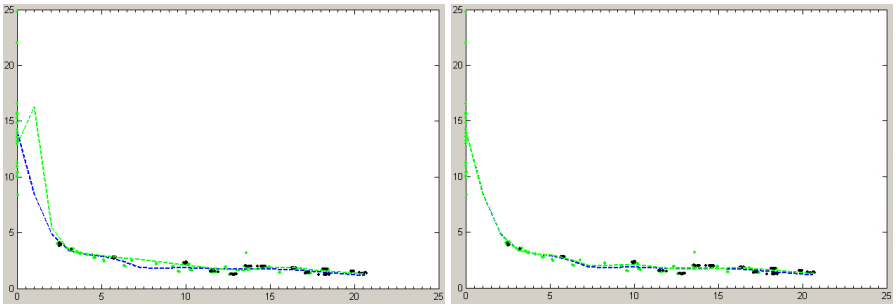

Figure 7: Left image: premise parameters adapted with 8 iterations of nonlinear optimization technique -> edgy shape in area where no new data points came in (lower region with empty load); right image: premise parameters adapted with only 2 iterations -> correct shape over the whole region; the fuzzy model represents the famous LAMBDA-PE(torque) relationship of a diesel engine

Beyond that, some aspects are not satisfying enough till now, namely:

- The computation time for updating premise parameters is by far not satisfying, as it lies above a predefined upper bound in the case of online adaptation, where data points are recorded with a frequency of 1 Hz or even faster. Additionally, the problem of choosing an appropriate value for the number of iterations arises when adjusting nonlinear parameters (see figure 7). In our examples above, this number was tuned manually, i.e. by trying to improve bad resulting model surfaces; in practice, the evaluation of an appropriate number has to be done automatically due to data's nature. As this is not a trivial task, alternatives such as adaptive clustering algorithms should be developed and applied, which also overcome the computational time deficit. Of course, adaptive clustering algorithms can only be applied, when generating the initial model with a cluster-based generation algorithm such as *genfis2*, which uses *subtractive clustering* for obtaining clusters and as a projection onto each axis fuzzy sets for each dimension. *ANFIS* and *RENO* as described in chapter 2 are not suitable for that task.

- Extensions to a new operating conditions are not practically implemented and tested till now, only ideas and hints for the enlargement of fuzzy models are given in chapter 4. Probably, these ideas need to be refined in the future.

- Shifting of complete fuzzy models, when input range of new adjacent test object is different, is not tested; open questions concerning this task would be: is a contraction or expansion of the widths of the fuzzy sets necessary?, is a shifting of the centers of the fuzzy sets enough or should be also an adaptation of the premise parameters after shifting? etc.

For all these reasons mentioned above various extensions for updating Takagi-Sugeno models have to be considered...

References

[1] Paul T. Boggs, Jon W. Tolle, "Sequential Quadratic Programming", Acta Numerica (1996), pp. 1-000

[2] Adolf Grauel, "Fuzzy Logik", Universität Paderborn, BI-Wiss.-Verl., 1995, ISBN 3-411-16801-3

[3] Werner Groissboeck, Edwin Lughofer, "A Theoretical and Practical Comparison of Variable Selection Methods with the Main Focus on Orthogonalization", Technical Report FLLL-TR-0211

[4] Josef Haslinger, Ulrich Bodenhofer,Martin Burger, "Data-Driven Construction of Sugeno Controllers: Analytical Aspects and New Numerical Results", Software Competence Center Hagenberg, 2000

[5] Jyh-Shing Roger Jang, "ANFIS: Adaptive Network-based Fuzzy Inference Systems", Department of Electrical Engineering and Computer Science, University of California, Berkeley, CA 94720

[6] Jyh-Shing Roger Jang, "Self-Learning Fuzzy Controllers based on Temporal Backpropagation", Department of Electrical Engineering and Computer Science, University of California, Berkeley, CA 94720

[7] Jyh-Shing Roger Jang, Chuen-Tsai San, "Neuro-Fuzzy Modelling and Control"

[8] Lennart Ljung, "System Identification: Theory for the User", Second Edition, Prentice Hall PTR, Prentice-Hall Inc., Upper Saddle River, New Jersey 07458

[9] Edwin Lughofer, Werner Groissboeck, "Dimension Reduction: A Comparison of Methods with respect to Fault Diagnosis of Engine Test Bench Data", Technical Report FLLL-TR-0210

[10] Edwin Lughofer, Werner Groissboeck, "Generating Correlation and Regression Models from High Dimensional Measurement Data - Advanced Aspects, Strategies and Validation", Technical Report FLLL-TR-0212

[11] Edwin Lughofer, Werner Groissboeck, "Online Adaptation of Correlation and Regression Models", Technical Report FLLL-TR-0213

[12] Hung T. Nguyen, Michio Sugeno, Richard Tong and Ronald R. Yager, "Theoretical Aspects of Fuzzy Control", John Wiley and Sons, Inc.

[13] Oliver Nelles, "Nonlinear System Identification - From Classical Approaches to Neural Networks and Fuzzy Models", ISBN 3-540-67369-5 Springer-Verlag Berlin Heidelberg New York

[14] H.Schwetlick and Thorsten Schütze, "Least Squares Approximation by Splines with Free Knots", BIT 35(3):361-384, 1995.

[15] Robert P. Velthuizen, Lawrence O. Hall, Laurence P. Clark and Martin L. Silbiger, "An Investigation of Mountain Method Clustering for large Data Sets", *Pattern Recognition*, Vol. 30, No. 7, pp. 1121-1135, 1997

[16] Shiqian Wu, Meng Joo Er and Yang Gao, "A Fast Approach for Automatic Generation of Fuzzy Rules by Generalized Dynamic Fuzzy Neural Networks", IEEE Transactions on Fuzzy Systems, Vol. 9, No. 4, August 2001

[17] Ronald R. Yager and Dimitar P. Filev, "Generation of Fuzzy Rules by Mountain Clustering", Technical Report MII-1318R, Machine Intelligence Institute, Iona College, New Rochelle, NY 10801