Transparent Fuzzy Systems in Modelling and Control

Andri Riid and Ennu Rüstern

Department of Computer Control, Tallinn Technical University, Ehitajate tee 5, 19086, Tallinn, Estonia
e-mail: andri@dcc.ttu.ee.

1 Introduction

First fuzzy controllers based on Zadeh's studies [1] of human-machine interaction, were simple expert systems designed on the basis of human operator experience as, for example, the steam engine controller of Mamdani and Assilian [2]. Although this approach has produced many successful applications e.g. [3], the common complaint is that the design procedure relies heavily on human judgement and is therefore more heuristics that exact science.

The important step toward the automation of fuzzy system design was taken in [4] where Takagi-Sugeno (TS) rules and least squares procedure for identifying fuzzy system parameters from data were simultaneously introduced. Later on, a gradient descent backpropagation technique adopted from neural network research was proposed for parameter adaptation [5], creating a new concept of "neuro-fuzzy" systems. Jang combined these techniques into ANFIS [6] that remains one of the most effective approximation algorithms today. Recently, application of clustering methods and genetic algorithms as well as further combinations of different algorithms for fuzzy system design have become popular [7,8,9].

Data-driven optimisation of fuzzy systems and consequent reduction of human role in the design process has resulted in greatly improved accuracy. One, however, cannot ignore the fact that on linguistic level these numerically improved fuzzy models and controllers are usually completely meaningless to an human observer.

In fact, surprisingly little attention (see section 3 for details) is devoted to the fact that perhaps the most attractive property of fuzzy systems that lies in the capacity to process information in linguistic terms is somewhat neglected and sacrificed to numerical accuracy. The aim of the present paper is to establish the mechanisms that would preserve the semantics of fuzzy systems even with the application of data-driven optimisation techniques instead of letting them to be destroyed.

The paper is organised as follows: in next section the definitions of main types of fuzzy systems are given. 3rd section presents the definition of transparency for standard and 0th order TS systems and transparency conditions/constraints based on this definition are derived. Transparency problem of 1st order TS systems is discussed. In section 4 the issue of transparency protection in fuzzy modelling is considered and in the 5th section relevance of transparency in fuzzy control is discussed. In section 6 several of previously described techniques are applied for the control of truck backer-upper demonstrating the advantages of transparent control for this particular problem.

2 Fuzzy systems

Presently, two main types of fuzzy systems are distinguished:

- Standard (linguistic, Mamdani) fuzzy systems [1,2]

Standard fuzzy systems consist of a number of rules that specify linguistic relation between the linguistic labels of input and output variables of the system. A fuzzy rule (1) is a statement where the premise and the consequent consist of fuzzy propositions with A_{ir} and B_{jr} denoting the linguistic labels of i^{th} input variable x_i and output variable y ($i = 1 \ldots N$), respectively, associated with the r^{th} rule ($r = 1 \ldots R$).

$$\text{IF } X_1 \text{ is } A_{1r} \text{ AND} \ldots \text{ AND } X_i \text{ is } A_{ir} \ldots \text{ AND } X_N \text{ is } A_{Nr} \text{ THEN } Y \text{ is } B_{1r} \tag{1}$$

Note that (1) expresses the input-output relationship in linguistic terms. To give the relationship in numerical terms, a special inference function (2) is used

$$y = \boldsymbol{Y}\left(\bigcup_{r=1}^{R}\left(\bigcap_{i=1}^{N}\mu_{ir}(x_i)\right)\cap\gamma_r\right), \tag{2}$$

where μ_{ir} and γ_r, denote normal and convex fuzzy subsets or membership functions (MFs) having one-to-one correspondence with the respective linguistic labels in (1); x_i denotes the numerical value of the i^{th} input variable; $\cap$ and $\cup$ denote the t- and s-norms that act as inference operators, respectively, and *Y(*)* denotes the defuzzification function (centre-of-gravity (CoG), mean-of-maximum (MoM), etc.) used.

- (First-order) Takagi-Sugeno systems [4]

In TS rules (3) the consequent fuzzy proposition is replaced by a linear combination of inputs, thus the rules are to be interpreted in terms of local linear models (y_r).

IF X_1 is A_{1r} AND X_2 is A_{2r} … AND X_i is A_{ir} … AND X_N is A_{Nr} THEN (3)
$y_r = p_{0r} + p_{1r}x_1 + \ldots + p_{ir}x_i + \ldots + p_{Nr}x_N$

p_{ir} (i = 0…N) in (3) denote the consequent coefficients. Because t-norm and s-norms in first-order TS (3) systems are commonly product and sum, inference function (2) reduces to

$$y = \sum_{r=1}^{R} \left(\prod_{i=1}^{N} \mu_{ir}(x_i) \right) (p_{0r} + \sum_{i=1}^{N} p_{ir} x_i) \Big/ \sum_{r=1}^{R} \prod_{i=1}^{N} \mu_{ir}(x_i) \qquad (4)$$

A special case of TS systems called 0^{th} order TS systems is obtained if the consequent function is a constant ($\forall p_{ir} = 0$, $i = 1 \ldots N$, $r = 1 \ldots R$):

IF U_1 is A_{1r} … AND U_i is A_{ir} … AND U_N is A_{Nr} THEN $y_r = p_{0r}$ (5)

$$y = \sum_{r=1}^{R} \prod_{i=1}^{N} \mu_{ir}(x_i) p_{0r} \; / \sum_{r=1}^{R} \prod_{i=1}^{N} \mu_{ir}(x_i) \qquad (6)$$

0^{th} order TS systems can also be regarded as a special case of standard fuzzy systems (with consequent fuzzy sets defined as fuzzy singletons) and semantically their interpretation is more closer to standard fuzzy systems than to 1^{st} order TS systems (3-4).

3 Fuzzy system transparency

The use of the term (transparency) in present paper is based on [10] where transparency is defined as a property that enables us to understand the influence of each system parameter on the system output as well as on [11] where fuzzy systems are characterised as being transparent to interpretation.

Fuzzy system transparency is closely related to the concept of linguistic interpretability but these are not matching terms and, in our opinion, it is very important to see the distinction. Interpretability is a property of fuzzy systems (1-6) that exists by default, being established with linguistic rules and fuzzy sets associated with these rules, even the rules of 1^{st} order TS systems can be interpreted. Transparency, on the other hand, is not a default property of fuzzy systems and should be regarded as a measure of how valid or how reliable is the linguistic interpretation of the system. It will be shown shortly that for standard fuzzy systems and 0^{th} order TS systems, transparency has binary character, for 1^{st} order TS systems it is a continuous variable.

Most authors, however, do not make this distinction; some of them do not pay attention to transparency at all and consequently assume that transparency like interpretability is a default property of fuzzy systems (sometimes regarded characteristic to standard and 0^{th} order TS systems only as in [12]); others do emphasise that transparency of fuzzy systems is not guaranteed by default [13, 14] but use the terms in parallel.

There are two aspects of transparency of fuzzy systems. First one is related to the readability of rules that basically boils down to the overall complexity of the system. Improvement of readability through the use of moderate number of variables, rules and fuzzy subsets or by avoiding the inconsistency of the rule base, however, does not provide the solution to the problem of destroyed semantics. To solve the problem, one should concentrate on low-level transparency that grows out from conformity between the linguistic layer and the inference function of a fuzzy system.

In fact, very few authors [13, 14, 15, 16] have investigated the latter issue in any detail. The most important of these works is perhaps [16] that lists a set of properties (moderate number of MFs; natural zero positioning, normality, coverage and distinguishability of MFs) that fuzzy systems should meet and proposes mathematically formulated constraints for preserving the last two, incorporated into the cost function of the gradient descent algorithm. These works dealing with low-level transparency, however, aim for certain balance between transparency and accuracy and the results can be generally applied only to a limited class of systems/algorithms.

It is claimed that "currently there exists no well-established definition of transparency of a fuzzy system" and "there are no definite criteria for the distinguishability of a fuzzy partition" [13]. Hopefully, solutions proposed to these problems in [17], further developed in [18] and [19] being summarised here, help to fill the void.

3.1 Transparency of standard fuzzy systems

Let us consider the properties listed in [16]. It is arguable if coverage and natural zero positioning have anything to do with transparency [14]. Normality on the other hand, is the standard assumption in fuzzy systems. Distinguishability of input MFs (directly related to the overlap of input MFs) is, however, vital to transparency as shown in the following.

The effect of overlap of input MFs to system output can be most conveniently observed in two-dimensional space that we do by constructing five otherwise equivalent SISO fuzzy systems, made up of 6 rules with 0%, 25%, 50%, 75% and 100% overlap degree, respectively. Although other system parameters (including minimum t-norm, maximum s-norm and CoG defuzzification) are fixed, in each case quite a different result is obtained (Fig. 1). With 0% overlap, no interpolation occurs, the system behaves as a multi-level relay and its output abruptly switches from one rule centroid to another. With 25% overlap the input intervals where the output has constant value, are still present but some interpolation between the neighbouring rules occurs.

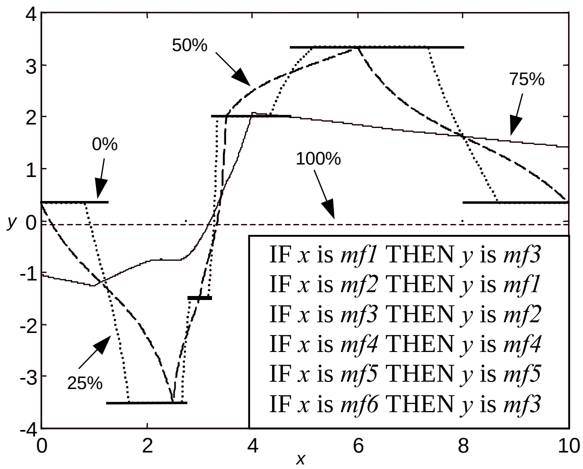

Fig. 1. Numerical input-output mapping of five fuzzy systems.

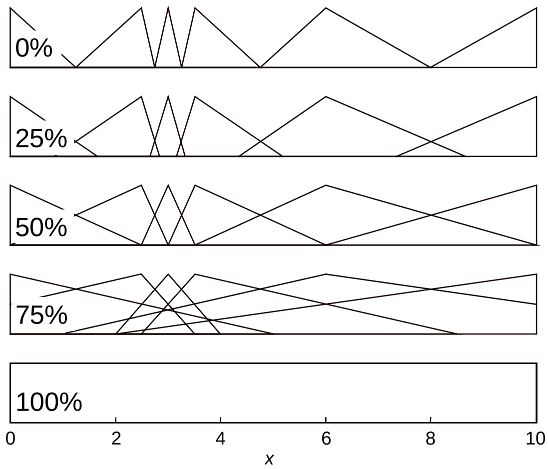

Fig. 2. Input MFs of observed systems.

With 50% overlap, the interval where the system output is the explicit contribution of a given rule is reduced to a single point. With larger overlap, however, at least two rules contribute simultaneously for any given input, thus system output is always the result of interpolation. This makes the contribution of the observed rule invisible in system output. We suggest that such feature is undesirable. The phenomenon is driven to extreme with 100% overlap where all rules are fully activated simultaneously and system output has constant value, equalling to the centroid of the union of output fuzzy sets.

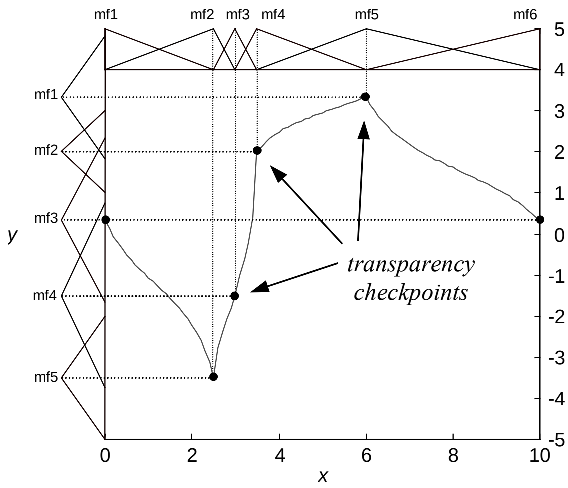

Fig. 3. Transparent fuzzy system.

Let us consider again the case of 50% overlap and let us refer to the point in input-output space where the explicit contribution of a given rule takes place and the rule under observation is fully activated as transparency checkpoint. When the overlap is equal or smaller than 50%, transparency checkpoints do exist. Closer inspection reveals that the input co-ordinate of the transparency checkpoint is equal to the centre of the fired MF (where $\mu(x) = 1$). Building up on the analogy, the desired output y for the transparency checkpoint would be the centre of the respective output MF, where $\gamma(y) = 1$. This ensures that the interpretation of the rule that we are able to obtain by combining the information from the rule base and MF definition base has good correspondence with the inferred numerical values (conformity!). This is exactly what we call transparency. The ideology of transparency checkpoints extends to MISO (and MIMO) systems and is covered by the following definition.

Definition: r^{th} rule of the standard MISO fuzzy system (1) is transparent if it's activation degree

$$\tau_r = \bigcap_{i=1}^{N} \mu_{ir}(x_i) = 1, \tag{7}$$

results in the system output

$$y = \text{core}(\gamma_r)\,, \tag{8}$$

where $\text{core}(\gamma_r) = \{y \in Y \,|\, \gamma_r(y) = 1\}$

A standard fuzzy system (1-2) can thus be regarded transparent only if all its rules are transparent (Fig.3).

In order to satisfy (7-8), certain conditions concerning input and output MFs of the system must be satisfied. The condition for input MFs, is given:

$$\forall x_i \in X_i : \sum_{s=1}^{S_i} \mu_i^s(x_i) \le 1\,, \tag{9}$$

where S_i denotes the number of fuzzy subsets defined for x_i. (9) implies that overlap of input MFs should not exceed 50%. Note that if (9) is strictly equal to 1, a fuzzy partition (alternatively termed Ruspini partition) is established.

For output MFs the following condition applies:

$$\text{Y}_{cog}(\gamma_r(y)) = \frac{\int_{y_{\min}}^{y_{\max}} y\gamma_r(y)dy}{\int_{y_{\min}}^{y_{\max}} \gamma_r(y)dy} = \text{core}(\gamma_r(y)) \tag{10}$$

It must be taken into account that with several MF types (e.g. Gaussian), (9) cannot be satisfied because of non-compact support of the MFs. Input MFs must therefore be "local" according to the following definition.

A MF $\mu_A(x)$, defined by three (or four) parameters a, b, c (or d), ($x, a, b, c, d \in X$), is said to be local if the following conditions hold:

$$\begin{cases} a \le b \le c \\ a = \min(\text{supp}(A)) \\ b = \text{core}(A) \\ c = \max(\text{supp}(A)) \end{cases} \quad \text{or} \quad \begin{cases} a \le b \le c \le d \\ a = \min(\text{supp}(A)) \\ b = \min(\text{core}(A)) \\ c = \max(\text{core}(A)) \\ d = \max(\text{supp}(A)) \end{cases}, \tag{11}$$

where $\text{supp}(A) = \{x \in X \,|\, \mu_A(x) > 0\}$.

It is easy to see that commonly used MFs such as triangular or trapezoid satisfy the respective conditions. Other examples of local MFs that can be found from literature are squared-cosine and cubic spline MFs [20].

The conditions (9,10) also apply for 0^{th} order TS systems. Note that output MFs of 0^{th} order TS systems are symmetric by definition.

For transparent fuzzy systems, we are able to predict its output at transparency checkpoints. Between these points the output is the result of interpolation that takes place between individual rules. The nature of interpolation is determined by fuzzy system parameters - defuzzification method, inference operators, shape of membership functions [17].

3.2 Transparency of 1st order TS systems

1st order TS systems are interpreted in terms of local linear models [11]. Overall system output, however, is interpolated from individual rules and quite often local models cannot be recognised in system output because of non-transparency. Interpolation issues of 1st order TS systems are considered in detail in [7] and two kinds of interpolation are distinguished: (i) S-type interpolation that produces intuitively expected results; (ii) V-type interpolation that has some undesirable properties but is more suited for continuous, smooth function approximation.

For two types of interpolation that can be distinguished, thus, an a priori order of preference cannot be given.

Let us consider another example where five otherwise equivalent 1st order TS systems are obtained by varying the overlap and the magnitude of the cores of input MFs (Fig. 4). We construct separate examples for V-type and S-type interpolation (Figs. 5 and 6, respectively).

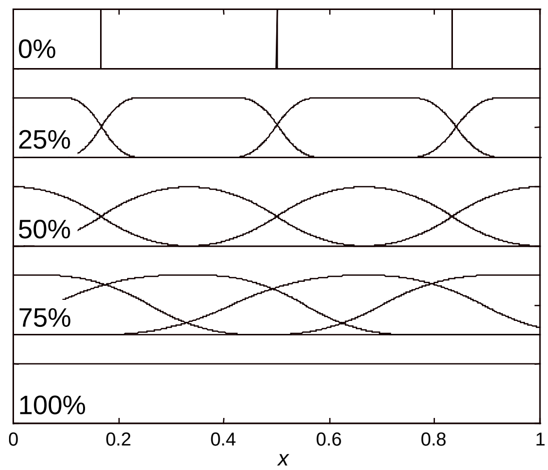

Fig. 4. Input-output relationship of TS systems with S-type interpolation.

Note that in the case of 100% overlap system rule base is replaced by single "average rule". In case of 0% overlap, on the other hand, there occurs no interpolation and 1st order TS system is a perfect piecewise linear system.

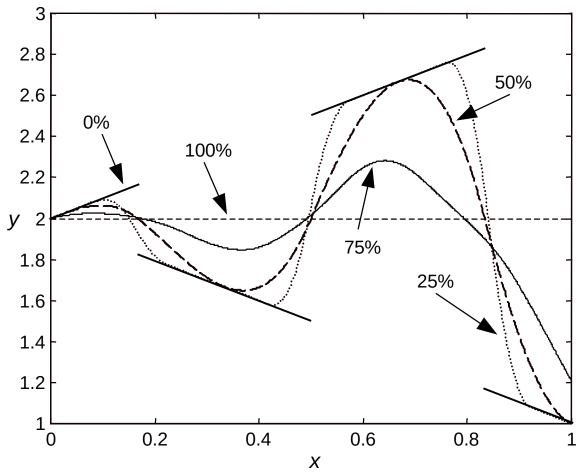

Fig. 5. Input-output relationship of TS systems with S-type interpolation.

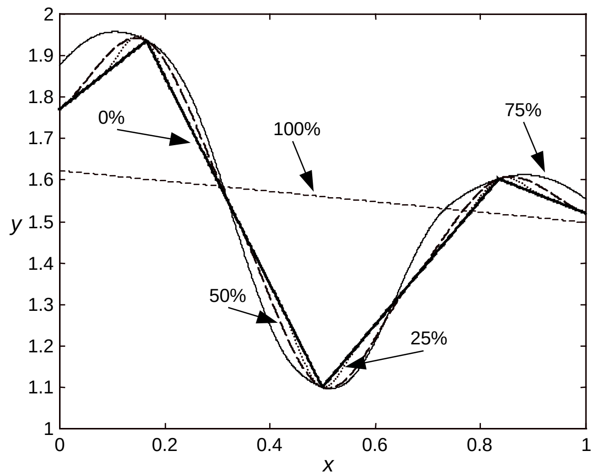

Fig. 6. Input-output relation of TS systems with V-type interpolation depending on the overlap of input MFs.

If the overlap of input MFs equals 50%, and the core of the r^{th} rule is a single point, the existence of the transparency checkpoint in output space where $y = y_r$ is guaranteed. With smaller overlap and larger cores, the region where system output is the contribution of a single rule increases and vice versa; consequently, the relationship between interpolation and "transparency error" is rather straightforward.

It turns out then that on these conditions, V-type interpolation (Fig.5) ensures smaller transparency error than S-type interpolation (Fig. 6).

For evaluation of this varying degree of transparency error, a measure (12), based on the deviation of the global output y from local models y_r of the system, was proposed in [19]:

$$\varepsilon_{tr} = \sqrt{\frac{\sum_{k=1}^{K}((y(k) - \underset{\max \tau_r(k)}{y_r}(k)))^2}{K}}, \tag{12}$$

where $\underset{\max \tau_r(k)}{y_r}$ denotes the output of the r^{th} rule with the highest activation degree for the k^{th} input-output pair and $y(k)$ is the corresponding global output.

Although V-interpolation yields smaller transparency error, it is shown in [19] that generally we have no reliable means for controlling the type of interpolation. Input MF constraint (9) remains relevant, by using MFs with multi-point cores, ε_{tr} can further reduced but this, however, does not always ensure low transparency error (particularly in case of S-interpolation). Moreover, we are not able to derive explicit transparency constraints for consequent parameters that would fulfil the purpose. Further improvement of transparency is, however, possible but depends on the identification algorithm and is therefore considered in the next section.

4 Modelling with transparency protection

With some identification algorithms transparency constraints (9-10) can be satisfied easily by a suitable a priori selection of MF parameters. This is for example true for Wang-Mendel method [21] or Babuska's combined approach of Gustafson-Kessel (GK) clustering and least square estimation (LSE) for 0^{th} order TS systems [7] where input fuzzy sets extracted from GK clusters form a fuzzy partition and output MFs identified through LSE are symmetric by definition. Note also that for 1^{st} order TS systems local least squares method [22] enhances transparency. The issue of transparency protection, however, specifically arises with iterative learning algorithms such as gradient descent or genetic algorithms where MFs undergo many modifications and in unconstrained mode this generally results in a non-transparent model.

The problem can be solved by (i) imposing constraints on membership functions that prevent the system from becoming non-transparent (ii) employing special membership functions that make transparency a default property of a fuzzy system (iii) multi-objective optimisation [16]. First two methods are applicable to standard fuzzy systems where we the transparency constraints have binary nature;

third approach is more suitable for 1st order TS systems where a certain balance between accuracy and transparency is sought. Transparency protection, generally, deteriorates the approximation capabilities of adaptation algorithms that is not a complete surprise as trade-off between accuracy and interpretability is a long known fact

4.1 Gradient descent

Gradient descent is based on the minimisation of the cost function

$$\varepsilon = \frac{1}{2}\left[y - \tilde{y}\right]^2 , \tag{13}$$

where y denotes the output of the fuzzy model (e.g. 2,4 or 6) and $\tilde{y}$ is the reference output.

To minimise the cost function (13) through the modifications of MF parameters, differential calculus is used that computes the necessary updates of optimised parameters c

$$\Delta c = -\alpha \frac{\partial \varepsilon}{\partial c} , \tag{14}$$

where α is the learning rate.

(14) implies that inference function y must be differentiable. This restricts gradient descent to TS systems (3-6).

One possibility to protect transparency of a fuzzy system trained with gradient descent is to verify the fulfilment of transparency conditions before every parameter update. If the new parameter value violates transparency conditions, the update will not be applied to the given parameter.

More efficient way to protect system transparency is to use such definition of MFs that make transparency a default property of the system.

Note that output MFs of 0th order TS systems satisfy (10) by definition, hence transparency preservation problem reduces to (9). To protect input transparency in the similar manner one can use Jager (neighbour-oriented) definition of triangular fuzzy sets [23], where each fuzzy subset is defined so that its edge parameters b_i^j, c_i^j equal the centres of the neighbouring sets, a_i^{j-1}, a_i^{j+1}, respectively (Fig. 10). Thus, a fuzzy partition is permanently maintained. Another advantage of Jager partition (15) is that the number of adjustable antecedent parameters is reduced.

$$\mu_i^j(x_i) = \begin{cases} \dfrac{x_i - a_i^{j-1}}{a_i^j - a_i^{j-1}}, & a_i^{j-1} < x_i < a_i^j \\ \dfrac{a_i^{j+1} - x_i}{a_i^{j+1} - a_i^j}, & a_i^j < x_i < a_i^{j+1} \\ 0, & a_i^{j+1} < x_i < a_i^{j-1} \end{cases} \tag{15}$$

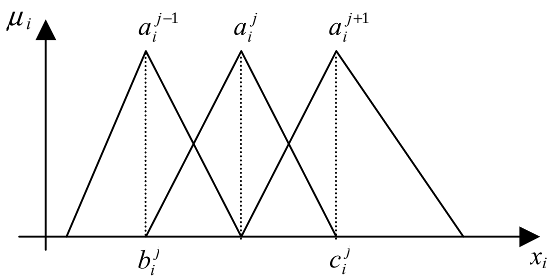

Fig. 7. Neighbour-oriented definition of triangular membership functions.

Usage of (15) means that traditional gradient descent based update formulas given in [5,24,25] are no longer valid and one must use the update rules derived by Jager [23]. Jager designed the algorithm originally for triangular MFs but expressions similar to (15) for other types of transparent MFs and respective update rules for their parameters can easily be derived.

For 1st order TS systems, input transparency protection alone does not ensure low transparency error and consequent parameters are constrained through multi-criterion optimisation where some form of transparency error (12) weighted with term λ is included in the optimisation cost function J, e.g.

$$J = J_1 + J_2 = \frac{1}{2}(\tilde{y} - y)^2 + \lambda \frac{1}{2}(\tilde{y} - \underset{\max \tau_r}{y_r})^2 \tag{16}$$

4.2 Genetic algorithms

Genetic algorithms (GAs) are stochastic search techniques that operate without knowledge of the task domain, utilising only the fitness of evaluated individuals. This feature makes the method extremely universal and allows training of fuzzy systems of arbitrary configuration and of arbitrary sets of fuzzy system parameters. Interested reader may refer to [8] for more information because in the following, we concentrate on MF parameter training schemes, because of their direct relation to transparency.

It is assumed that a potential solution to a problem may be represented by a set of parameters. These parameters (usually encoded into binary alphabet) are joined together to form a chromosome. The initial population consisting of N chromosomes evolves to the next generation through the genetic operations of crossover and mutation. After each step, chromosomes are decoded, fuzzy systems corresponding to those chromosomes are reconstructed and the fitness of these systems is evaluated. Chromosomes with better fitness have higher probability to survive to the next generation. If the algorithm is properly designed it will converge to an optimal solution although it usually requires many cycles (generations).

Transparency protection mechanisms are similar to gradient descent. MF parameters can be validated when system is being reconstructed from a chromosome. Note that because GAs allow training of standard fuzzy systems, output MF constraint must also be taken into account.

Alternatively, we can make use of Jager partition for input MFs and some definition of symmetric output MFs, e.g triangular MFs.

$$\gamma^t(y) = \begin{cases} -(s^t/2)(y_j - a^t) + 1, & a^t - s^t/2 < y_j < a^t \\ (s^t/2)(y_j - a^t) + 1, & a^t < y_j < a^t + s^t/2 \\ 0, & a^t + s^t/2 < y_j < a^t - s^t/2 \end{cases}, \tag{17}$$

where s^t and a^t are the centre and the spread of (17), respectively.

The chromosome configuration of a transparent fuzzy system corresponding to (15) and (17) is depicted in Fig. 8.

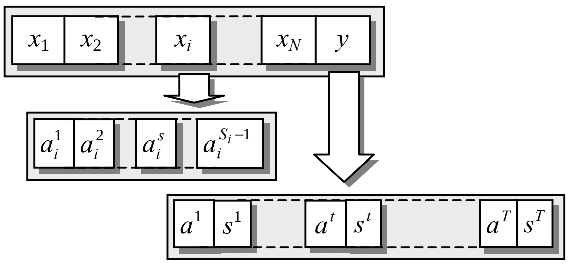

Fig. 8. Chromosome configuration.

For 1st order TS systems, multi-criterion optimisation is considered again. Fitness function of the GA may be formulated as:

$$\text{Fitness(chromosome)} = 1/(\varepsilon + \lambda\varepsilon_{tr}) \tag{18}$$

5 Transparent fuzzy systems in control

Traditional classification of fuzzy control [26] divides fuzzy controllers into four main categories. We briefly describe these four categories and discuss the relevance of transparency in each case.

- Controllers designed on the basis of expert experience and control engineering knowledge

Knowledge based controllers are typically open loop controllers (there is no feedback involved), state feedback controllers or set point controllers with and without additional inputs. The design methodology is ill-defined and problem dependent. Two more problems associated with this approach can be pointed out. One is the possible inadequacy of the expert as the controller cannot be better than expert's knowledge. Another and even more serious issue is the expert's possible inability to express his control experience or general knowledge effectively with the tools of fuzzy logic, either because he/she does not understand the properties of fuzzy systems very well or cannot formulate the control rules verbally because it is only his/her body that knows how to control the process/system, not the mind.

By definition, transparency is vital to this type of controllers, otherwise the expert knowledge appears in distorted form and consequently the controller performance is sub-optimal.

- Controllers modelled on the existing controllers

This type of fuzzy controllers try to mimic some other working controller. During the training it is connected so that it has access to the inputs and outputs of the working controller. After it is found to have learnt the expected task, it is put online and replaces the original controller. The approach is very useful if the controller to be emulated is a human being (who is unable to express his control knowledge verbally) or if the original control algorithm is very expensive to implement. The possible disadvantage of the controller is that it cannot be better than the original controller and often may be worse because there always exists certain modelling error.

Transparency of the controller is not the necessary requirement, as we are primarily concerned with the numerical performance of the controller. Transparency, however, may be useful as it allows the validation of the controller by an expert.

- Model-based fuzzy control

Model-based fuzzy control uses a given (typically fuzzy) open loop model of the plant under control to derive the set of fuzzy rules for the fuzzy controller and is therefore principally different from previous two approaches where it is implicitly assumed that no model exists. Examples of model-based fuzzy control are model-based predictive control [27,28], inverse fuzzy process model based control

[29,30] and fuzzy gain-scheduling methods [31]. As stated, this type of control involves the generation of a fuzzy model of the controlled process as the preliminary step and typically numerical accuracy of the model is the primary concern with the exception of [32] where the linguistic inversion of the controlled process is proposed.

With this type of control we are again typically more interested in the numerical properties of the controller (and of the model) that makes transparency unnecessary. The exception is the linguistic inversion of the process model [32] where transparency of the model plays the key role and it is interesting to note that exact model inversion technique proposed by Babuska [7] assumes certain properties of a model (including transparency).

- Self-learning fuzzy controllers (adaptive fuzzy control)

In adaptive fuzzy control, the focus is on the automatic on-line synthesis and tuning of fuzzy controller parameters which will ensure that the performance objectives are met even if the plant parameters change in time. Generally, these techniques can be split into two categories: direct and indirect adaptive fuzzy control. In indirect adaptive fuzzy control, there is an identifier mechanism that produces a model of the plant which is then used to specify the controller e.g. [33]. Thus the distinction between model-based and adaptive fuzzy control is sometimes imaginary. In direct adaptive control, a model of the plant is not estimated; instead, we tune the controller parameters directly using plant data [34, 35].

Presently, the role of transparency in adaptive fuzzy control is unexplored (but very interesting) research topic.

6 Application of transparent fuzzy modelling and control

Truck backer-upper problem was first presented in [36] where Nguyen and Widrow amply demonstrated the learning potential of neural networks applied for the tuning of self-learning controller based on temporal backpropagation. In more recent works, several authors have replaced or complemented neural network with genetic algorithms, e.g [37]. The basic shortcoming of all these data-driven techniques is the computational cost.

On the other hand, the problem is an ideal test bed for fuzzy control systems because nearly anyone is able to drive the truck to the desired position given some time to adjust himself to the controls and the potential of fuzzy logic for implementing expert knowledge is well-known. Several applications can be found from literature [21,38]. The following material is based on the results of [39], putting more stress on transparency analysis.

6.1 Truck backer-upper system

The system used in the simulations is supplied with MATLAB as a demo. The truck as in [21,38] corresponds to the cab part of the Nguyen-Widrow's truck and trailer, referred to as simplified Nguyen-Widrow problem. The truck position is determined by the three state variables x = [-20, 20], y = [0, 25], and, Φ_c = [-90°, 270°] - the angle between truck's onward direction and the x-axis (Fig. 9). The width and length of the truck are 4 and 2 meters, respectively.

Truck must arrive from the initial position (x_0, y_0, Φ_0) to the loading dock (x_f = 0, y_f = 0) at a right angle (Φ_f = 90°). Truck only moves backward with the fixed speed 2m/s. To control the truck at every stage appropriate steering angle θ = [-45°, 45°] must be provided. Thus controller is a function of state variables

$$\theta = f(x, y, \Phi_c)\,. \tag{19}$$

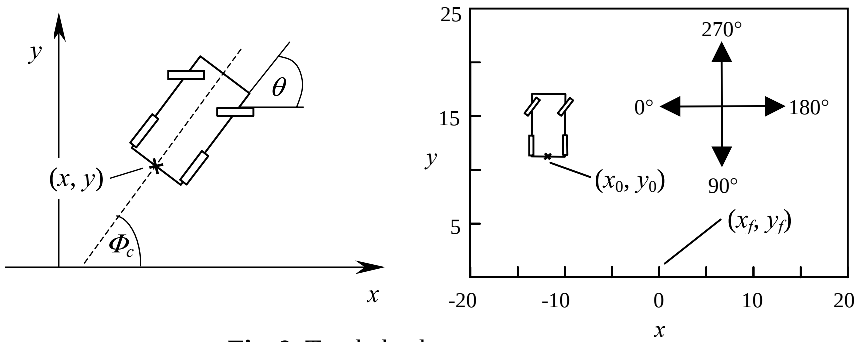

Fig. 9. Truck backer-upper system.

Typically, it is assumed that enough clearance between the truck and the loading dock exists so that the truck y-position co-ordinate can be ignored, simplifying (19) to:

$$\theta = f(x, \Phi_c)\,. \tag{20}$$

6.2 Truck backer-upper controllers

We consider two modes of fuzzy control described in the section 5: knowledge-based control and mimicking control.

6.2.1 Knowledge-based controllers

Selection of the controller function (19) implies that the following rule base format should be used

IF *x* is *A* and *y* is *B* and Φ_c is *C* THEN θ is *D*, (21)

where *A*, *B*, *C* and *D* are the linguistic labels of the system variables associated with the corresponding fuzzy sets.

Design of the fuzzy controller includes the definition of input-output domains, partitions and fuzzy sets, and the contents of the rule base. The only source of that information in present case is human understanding of the driving process. The major problem with (21) is so-called curse of dimensionality. Employing the input partition {5 3 7}, for example, results in 105 rules that all must be derived from experience. Although we can drive the car to the loading dock manually from almost any position, design of the fuzzy controller that would achieve the same goal is not a trivial task. Though fuzzy logic is a good interface for man-machine interaction, the problem in present case is that we do not know exactly how we are able to drive the car. For this reason, tuning of the controller is not a trivial task and whole design procedure becomes time-consuming and frustrating when the number of tuning parameters is large. Consequently the design task of (21) is extremely difficult and therefore the state variable selection (19) was replaced with (20). This enabled us to use the control rules given in [38]. Some readjustment of the controller parameters was necessary though, because the truck backing systems are not identical.

For the more effective utilisation of existing control experience, however, a hierarchical control system is more suitable as demonstrated in [39]. It is observed that in interaction with the real world we are engaged in a continuous process of constructing representations of that environment and our experience of it. Some of these representations are very simple, others highly specific. They may also operate at different levels of consciousness. The driving process (how we handle the car controls) is generally carried out on subconscious level. To obtain that skill one usually needs many hours of extensive training and when asked to explain the control principles, one cannot provide adequate answers. Conscious models, on the other hand are based on “common sense”, can be easily expressed in linguistic form and can therefore be modelled with fuzzy logic. Trajectory planning of the truck is an example of a conscious model.

This two-level control model can be effectively modelled with the proposed control system where the control block consists of a fuzzy supervisor and PD controller (Fig. 10). The task of the supervisor that implements the high-level control strategy is to provide setpoint Φ_r for the given state, PD controller has to determine then appropriate steering angle. Although extra effort is required to determine the parameters of PD controller, it can be considered a bargain price for the exclusion of one state variable from the input of the fuzzy block.

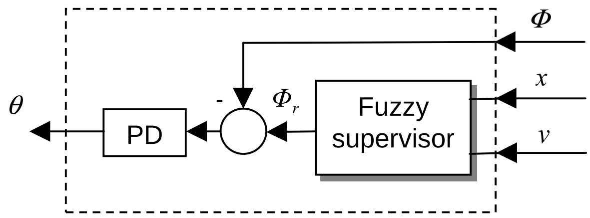

Fig. 10. Block diagram of fuzzy supervisory control system.

The rule base of the supervisor is therefore easily configured, e.g. grey region in Fig. 11 reads as

$$\text{IF } x \text{ is } mf4 \text{ AND } y \text{ is } mf3 \text{ THEN } \Phi_r \text{ is } 90°, \quad (22)$$

(22) is in good accordance with the general idea what angle the truck in this particular area should maintain. The rest of the rules are based on the same analogy.

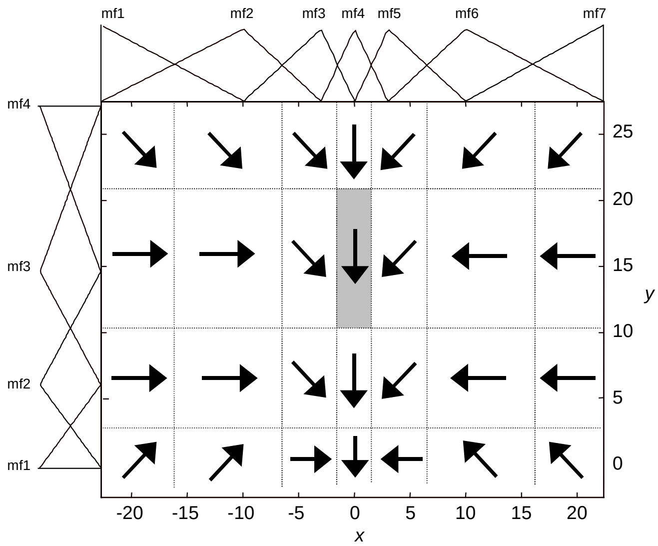

Fig. 11. Rule base of the two-input fuzzy supervisor.

The advantages of such decomposition are thus (i) simplified design and (ii) better compatibility with the actual control principles of human beings.

6.2.2 Controllers modelled on human operator

The crucial problem with data-driven techniques is the selection criterion of training data. In theory, we need a sufficient amount of data that would give good representation of operator actions. In practice, our resources are always limited.

The problem with multidimensional systems is that some rules created in the initialisation phase remain uncovered by data, implying that the rule base of the controller will be sparse. This may result in unexpected behaviour. On the other hand, large amount of training data slows down the learning process. Another data-driven modelling issue is that modelling algorithms available are not perfect; there always exists modelling error.

For modelling we used ANFIS [5] and Gustafson-Kessel clustering in combination with least squares procedure [7]. The ANFIS algorithm lacks transparency protection and was employed because of its excellent approximation properties shown in [5], application of GK/LSE on the other hand would result in transparent model of human controller.

Data used in modelling was collected from 31 truck backing experiments with 8 upward, 6 leftward, 6 rightward and 11 downward initial angles. Starting positions were chosen so that different backing trajectories would be present (Fig. 12). To reduce the computational load, most of data was filtered out so that the final data set, consisting of 642 input-output pairings, corresponds to the situation as if information had been available every third second only (normal sampling interval is 0.1s).

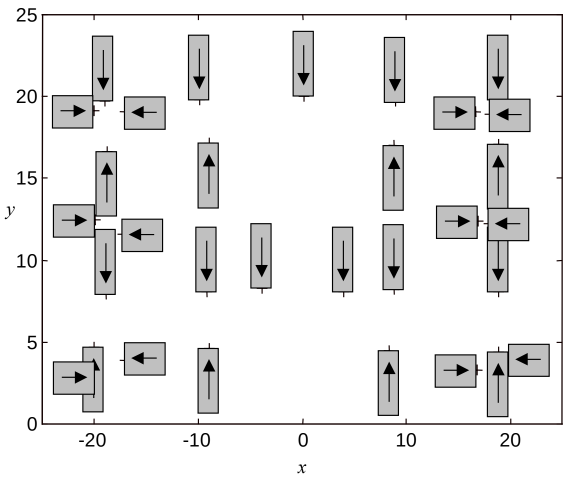

Fig. 12. The initial positions of backing trajectories in modelling data (left)

The number of parameters that influence the approximation error and must be determined prior to training is quite large and all of them cannot be determined automatically. Therefore, the determination of training parameters was based on trial and error, until the configuration by what "reasonably low" approximation error could be achieved, was established. Very soon, the necessity for modelling the control law (20) was confirmed because with (19), results of any acceptable accuracy could not be obtained. ANFIS was then applied to 1st order Takagi-Sugeno system with input partition of {7 3 9} and GK/LS model was initialised as a 0^{th} order Takagi-Sugeno system with the same partition. Final modelling root mean square errors for ANFIS (2500 epochs, RMSE = 0.2129) and GK/LS (RMSE = 0.2048) are quite similar, though.

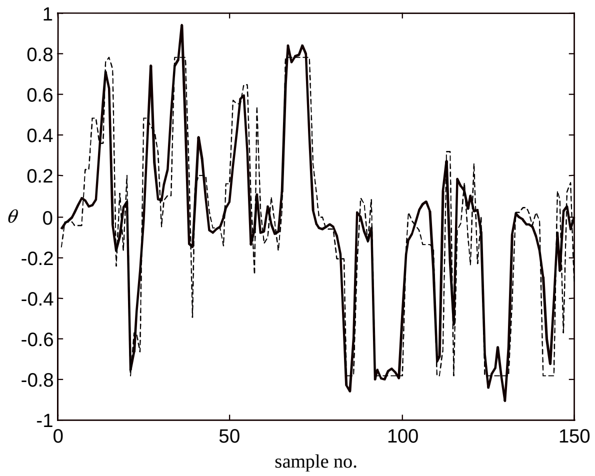

Fig. 13. Modelling results (excerpt) - target steering angle (dashed line), modelled steering angle (bold line) in the right side of the figure.

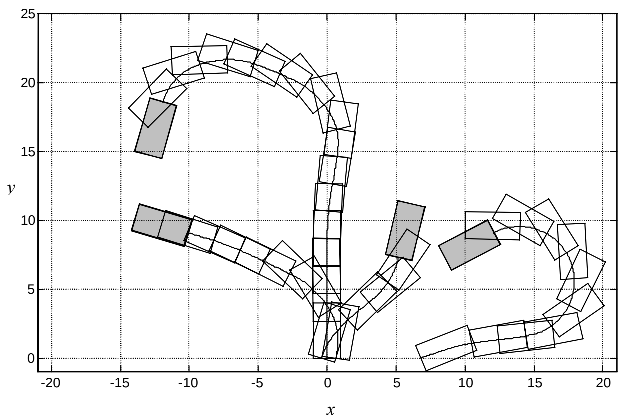

Fig. 14. Backing up with knowledge-based controller (initial positions of the truck are indicated with grey colour).

6.3 Control results

The comparison of different controllers is based on backing the truck from randomly chosen initial positions (we present here only a few of the experiments conducted in [39]). The backing trajectories are depicted in Figs. 14-17.

Expert defined controller shows the weakest performance - partly because it does not make account of *y* co-ordinate (consequently it cannot guarantee success for initial positions close to the loading dock), partly because tuning of the controller is based on more on trial-and-error than on efficient knowledge translation.

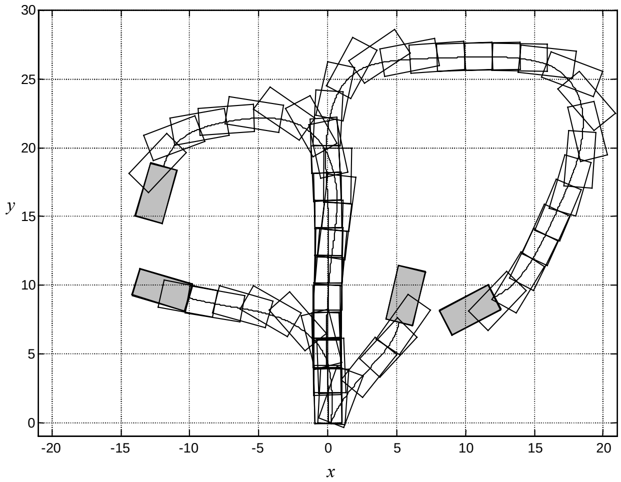

Fig. 15. Backing up with human operator model (ANFIS).

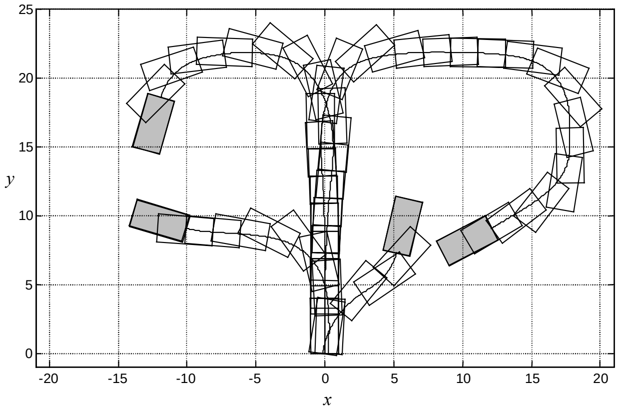

Fig. 16. Backing up with human operator model (GK/LSE).

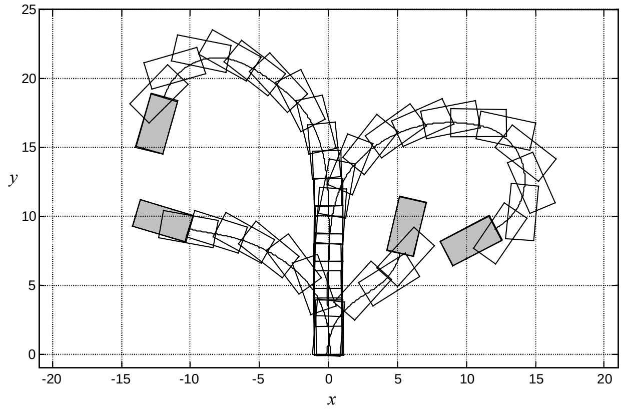

Fig. 17. Backing up with supervisory control system.

Controllers modelled on human operator show better performance (Figs. 15-16). Due to approximation error, however, their performance is sub-optimal. We saw that the approximation error of both algorithms was in the same range, ANFIS-approximated controller, however, shows erratic behaviour on some occasions (Fig. 15). Due to non-transparency of the controller we are not able to validate the rules and on the other hand, this non-transparency may be the reason why truck occasionally goes "berserk".

Finally, due to enhanced transparency, supervisory control system allows more efficient design and provides smooth and economic truck trajectories with superior control accuracy compared to other approaches (Fig. 17).

7 Concluding remarks

In this paper the systematic approach to transparency problem of fuzzy systems was presented. Transparency that is distinguished from linguistic interpretability (the latter is considered a default property of the observed classes of fuzzy systems) measures validity or reliability of the linguistic interpretation. Transparency as defined for standard and 0^{th} order TS systems (8-9) implies that fuzzy system transparency is of binary character for these types of systems. Taking the transparency definition as the basis, transparency constraints for standard and 0^{th} order TS systems were derived and mechanisms for preserving transparency in iterative modelling were discussed.

For 1^{st} order TS systems the situation is a bit different. Although transparency checkpoints can be similarly defined, this does not guarantee low transparency error (12) because interpolation in 1^{st} order TS systems has undesirable properties from transparency viewpoint. Additional means for improving transparency of 1^{st}

order TS systems were discussed including the use of MFs with multi-point cores (e.g. trapezoid MFs) and transparency-sensitive identification algorithms for consequent parameters (local least squares, gradient descent and genetic algorithms with multi-objective optimisation criterion).

Transparency is of primary importance in linguistic analysis and synthesis of control systems [32]. The applications of transparent control presented in this paper clearly demonstrate that transparency is vital to this branch of intelligent control that seeks solutions by emulating the mechanisms of reasoning and decision processes of human beings. It must be stressed that best results are obtained if besides transparency preservation other aspects of linguistic interpretability such as complexity reduction are taken into account. Possible implication to presently black-box techniques of fuzzy control where numerical accuracy is the primary concern is presently unclear, however, this line of research will be our first concern in near future.

References

1. L.A. Zadeh. Outline of a New Approach to the Analysis of Complex Systems and Decision Processes. *IEEE Trans. Systems, Man and Cybernetics*, 3:28-44, 1973.
2. E.H. Mamdani and S. Assilian. An experiment in linguistic synthesis with a fuzzy logic controller. *Int. J. Man-Machine Studies*, 7:1-13, 1975.
3. L.P. Holmblad and J.J. Ostergaard. Control of Cement Kiln by Fuzzy Logic. In M.M-Gupta and E. Sanchez, editors, *Approximate Reasoning in Decision Analysis*, pages 389-400. Amsterdam, Holland, 1983.
4. T. Takagi and M. Sugeno. Fuzzy identification of systems and its applications to modeling and control. *IEEE Trans. Systems, Man and Cybernetics*, SMC-15(1):116-132, 1985.
5. L.-X. Wang and J.M. Mendel. Back-propagation fuzzy system as nonlinear dynamic system identifiers. *Proc. 1st IEEE Int. Conf. on Fuzzy Systems*. pages 1409-1416, San Diego, CA, USA, 1992.
6. J.-S.R. Jang. ANFIS: Adaptive-network-based fuzzy inference system. *IEEE Trans. Systems, Man and Cybernetics,* 23(3):665-685, 1993.
7. R. Babuska. *Fuzzy Modeling and Identification*, Ph.D. dissertation. Technical University of Delft, Delft, Holland, 1997.
8. J. Liska and S.S. Melsheimer. Complete design of fuzzy logic systems using genetic algorithms. *Proc. 3rd IEEE Int. Conf. on Fuzzy Systems*, pages 1377 – 1382, Orlando, FL, USA, 1994.

9. J.A. Roubos and M. Setnes. Compact fuzzy models through complexity reduction and evolutionary optimization. *Proc IEEE Int. Conf. Fuzzy Systems*, pages 762-767. San Antonio, TX, USA, 2000.

10. M. Brown and C.J. Harris. *Neurofuzzy Adaptive Modelling and C*ontrol, Prentice Hall, Englewood Cliffs. 1994.

11. M. Setnes, R. Babuska and H.B. Verbruggen. Rule-Based Modeling: Precision and Transparency. *IEEE Trans. Systems, Man and Cybernetics*. 28(1):165-169, 1998.

12. D. Nauck, F. Klawonn and R. Kruse. *Foundations of Neuro-Fuzzy Systems*. Wiley, Chichester, UK, 1997.

13. Y. Jin. Fuzzy modeling of high-dimensional systems: Complexity reduction and interpretability improvement. *IEEE Trans. Fuzzy Systems*, 8(2):212-220, 2000.

14. R. Babuska. Construction of fuzzy systems - Interplay between precision and transparency. *Proc. European Symp. Intelligent Techniques (ESIT)*, pages 445-452, Aachen, Germany, 2000.

15. A. Lotfi, H.C. Andersen and A.C. Tsoi. Interpretation preservation of adaptive fuzzy inference systems. *Int. J. Approximate Reasoning*, 15(4):379-394, 1996.

16. J.V. de Oliveira. Semantic constraints for membership function optimization. *IEEE Trans. Systems, Man and Cybernetics*, 29(1):128-138, 1999.

17. A. Riid and E. Rüstern. Transparent fuzzy systems and modeling with transparency protection. *Proc. IFAC Symp. on Artificial Intelligence in Real Time Control*, pages 229-234. Budapest, Hungary, 2000.

18. A. Riid, P. Jartsev and E. Rüstern. Genetic algorithms in transparent fuzzy modeling. *Proc. 7th Biennal Baltic Electronic Conf. 2000*, pages 91-94. Tallinn, Estonia, 2000.

19. A. Riid, R. Isotamm and E. Rüstern. Transparency analysis of 1st order Takagi-Sugeno systems. *Proc 10th Int. Conf. System-Modeling-Control*, 2:165-170. Zakopane, Poland, 2001.

20. M. Bikdash. A highly interpretable form of Sugeno inference systems. *IEEE Trans. Fuzzy Systems*, 7(6):686-696, 1999.

21. L.-X. Wang and J.M. Mendel. Generating fuzzy rules by learning from examples. *IEEE Trans. Systems, Man, and Cybernetics*, 22(6):1414-1427, 1992.

22. J. Yen, L. Wang, and W. Gillespie (1998), "Improving the Interpretability of TSK Fuzzy Models by Combining Global Learning and Local Learning," *IEEE Transactions on Fuzzy Systems,* Vol. 6, No. 4, pp. 530-537, November 1998.

23. R. Jager. *Fuzzy Logic in Control*, Ph.D. dissertation. Technical University of Delft, Delft, Holland, 1995.

24. F. Guely and P. Siarry. Gradient descent method for optimizing various fuzzy rule bases. *Proc. 2nd IEEE Int. Conf. Fuzzy Systems*, pages 1241-1246. San Francisco, CA, USA, 1993.

25. H. Nomura, I. Hayashi and N. Wakami. A learning method of fuzzy inference by descent method. *Proc. 1st IEEE Int. Conf. on Fuzzy Systems*, pages 485-491. San Diego, CA, USA, 1992.

26. C.C. Lee. Fuzzy logic in control systems: Part I and Part II. *IEEE Trans. Systems, Man and Cybernetics*, 20(2):404-435, 1990.

27. J. J. Espinosa, M. L. Hadjili, V. Wertz and J. Vandewalle. Predictive control using fuzzy models - Comparative study. *Proc. European Control Conference*. Karlsruhe Germany, 1999.

28. R. Babuska, M. A. Botto, J.S. da Costa, H.B. Verbruggen. Neural and fuzzy modeling in nonlinear predictive control: a comparative study. *Proc*. *Computational Engineering in Systems Applications: Symposium on Control, Optimization and Supervision*, pages 1049-1054. Lille, France, 1996.

29. J. Abonyi, L. Nagy and F. Szeifert. "Indirect Model Based Control Using Fuzzy Model Inversion. *Proc. IEEE Int. Conf. Intelligent Systems*, pages 951-956. Vienna, Austria, 1998.

30. K.J. Hunt and T.A. Johansen. Design and analysis of gain-scheduled local controller networks. *International Journal of Contro*l, 66(5):619–651, 1997.

31. P. Korba and P.M. Frank. An applied optimization-based gain-scheduled fuzzy control. *Proc. American Control Conference*, pages 3383–3387, Chicago, CH, USA, 2000.

32. M. Braae and D.A. Rutherford. Theoretical and Linguistic Aspects of the Fuzzy Logic Controller. *Automatica*, 15(5):553-577, 1979.

33. J-S. R. Jang. Self-learning fuzzy controllers based on temporal back propagation. *IEEE Trans. Neural Networks*, 3(5):714-723, 1992.

34. T.J. Procyk and E.H. Mamdani. A linguistic self-organizing process controller. *Automatica*, 15:15-30, 1979.

35. H.R. Berenji and P. Khedkar. Learning and tuning fuzzy logic controllers through reinforcements. *IEEE Trans. Neural Networks*, 3(5):724-740, 1992.

36. D. Nguyen and B. Widrow. Neural networks for self learning control systems. *IEEE Contr. Syst. Mag.*, 10(2): 18-23, 1990.

37. M. Schoenauer and E. Ronald. Neuro-genetic truck backer-upper controller. *Proc. First Int. Conf. Evolutionary Comp*., pages 720-723. Orlando, USA, 1994.

38. S.-G. Kong and B. Kosko. Adaptive fuzzy systems for backing up a truck-and-trailer. *IEEE Trans. Neural Networks*, 3(5):211-223, 1992.

39. A. Riid and E, Rüstern. Fuzzy logic in control: Truck backer-upper problem revisited. *Proc. IEEE 10th Int. Conf. Fuzzy Systems*, Melbourne, Australia, 2001 (to be published).