Algorithmic Aspects of Fuzzy Control

László T. Kóczy

Department of Telecommunication and Telematics
Technical University of Budapest
Sztoczek u. 2
H-1111 Budapest, Hungary

ABSTRACT

Fuzzy control is at present still the most important application of fuzzy theory. It is a
generalized form of expert control using fuzzy sets in the definition of vague/linguistic
predicates, modeling a system by If... then rules. In the classical approaches (Zadeh,
Mamdani) the essential idea is that a fact (observation) known concerning the actual
state of the system will match with one or several rules in the model to some positive
degree, the conclusion will be calculated by the evaluation of the degree of these
matches, and the matched rules themselves. In these approaches, the rules contain
linguistic terms, i.e., fuzzy sets in the consequent parts, and these terms, weighted with
their respective degrees of matching, will be combined in order to obtain a fuzzy
conclusion—from which the crisp action is obtained by defuzzification, as e.g. the
center of gravity method. This paper summarizes these classical methods and turns
attention to their weak point: the computational complexity aspect. As a partial
solution, the use of sparse rule bases is proposed and rule interpolation as a fitting
inference engine is presented. The problem of preserving or not preserving linearity is
discussed when terms in the rules are restricted to piecewise linear.

KEYWORDS: fuzzy control, rule interpolation, preservation of piecewise
linearity, preservation of normality, hierarchical rule base

1. INTRODUCTION

Fuzzy control, as it was introduced by Zadeh [1] and then by Mamdani
[2], is a powerful tool for systems where the exact model is not known, or

*This article contains partly the topics discussed at CIFT '93 in the author's plenary paper
"State of the Art of Fuzzy Control Algorithms," and in addition includes some further results
on the linear interpolation algorithm.

E-mail: koczy@ttt-202.ttt.bme.hu.

Received September 1994; accepted November 1994.

Address correspondence to Dr. László T. Kóczy, Department of Telecommunications and Telematics,
Technical University of Budapest, Sztoczek u. 2, H-1111 Budapest, Hungary.
where it is too complex to be tractable in real time. However, its applicability has strict limitations. A serious problem is high computational time and space complexity of the rule base describing the system with multiple state variables at proper accuracy level. The “exponential explosion” allows no real time application of the classical fuzzy control algorithms where the number of state variables exceeds 8 or 10. An exponential expression can be reduced in two ways: decreasing the base or decreasing the exponent. The former method leads to the use of sparse rule bases, and all kinds of rule interpolation and approximation in the general sense. The main topic of this paper is related to this group of methods. Some new techniques for identifying rules with very large support size fall in the other category, where the total number of rules is not much higher than the number of variables. These algorithms belong to the second group: they reduce the exponent. Unfortunately, the possibility of finding such rule bases is very much system dependent. Nevertheless, this technique will also be addressed. It is possible to combine both methods, but at present this is still an open problem.

Another approach to fuzzy control is the Takagi-Sugeno model [3], where the consequents are given in the form of linear or, more generally, arbitrary functions. The (crisp) conclusion can be obtained by the weighted average of the substitution values of these functions. This model copes with the complexity problem very well; however, it allows less flexibility in the consequent part. It is possible to extend the idea to fuzzy functions instead of ordinary ones, which eliminates the latter problem, but introduces a new question: how the difficulty of computing with too many \( \alpha \)-levels can be overcome.

2. THE LINGUISTIC APPROACH TO FUZZY CONTROL

Fuzzy control in the linguistic rule based sense was first applied in 1974; a new approach based on the theory of fuzzy sets was proposed in 1965 and then further developed by L. A. Zadeh. His crucial paper, published in 1973 [1], essentially contains the ideas of modeling and controlling very complex, in most cases nonlinear, systems through an expert-system-like model consisting of linguistically formulated “rules of thumb,” containing fuzzy sets for the mathematical representation of the linguistic, vague, or imprecise concepts describing approximate values or sets of values of the state variables.

The essence of this idea is that the knowledge is given as a set of rules in the form

"If \( x \) is \( A \) then \( y \) is \( B \),"
where $x$ stands for the input state variable, usually a compound variable

$$x = (x_1, x_2, \ldots, x_k);$$

$y$ is the output state variable (or action variable), which can always be considered as a scalar (otherwise the rules can be decomposed); and $A$ and $B$ are linguistic terms like "positively big" or "very tall" which are represented by fuzzy sets of the universe of discourse, $X = \{x\}$ and $Y = \{y\}$, respectively. Such a rule can be represented mathematically by a fuzzy relation of the cross product space $X \times Y$, i.e., it is a fuzzy set of the latter. The considerable advantage of this method over the classical expert control approach is that the fuzzy terms $A$ cover a wide area of the input space (with different degrees of validity, i.e., membership degrees), and so a relatively small number of rules might cover all possible situations at least to some degree. The definition of rules by general relations in the form

$$R : X \times Y \rightarrow [0, 1]$$

might reflect the real life situations in the most adequate manner; however, the input variables are supposed to be independent, and the relations decomposable in the form

$$R = S_1 \times S_2 \times \cdots \times S_k \times S_{k+1},$$

where

$$S_i : X_i \rightarrow [0, 1] \text{ for } i = 1, \ldots, k \text{ and } S_{k+1} : Y \rightarrow [0, 1]$$

or

$$R = \prod_{i=1}^{k} \text{proj}_{X_i}(R) \times \text{proj}_{Y}(R)$$

is true. Such a rule can be formulated like the following: "If the amount of dirt is very high and the mass of the water is medium then the necessary amount of detergent to add is high" (a possible rule for a fuzzy washing machine).

Having a knowledge base with a set of production rules like $R$, an observation on $X$ in the form of "$x$ is $x^*$" or, more generally, "$x$ is $A^*$" can be combined with them by relational composition. (In the former, the observation is a singleton value in $X$, while the latter is a fuzzy term, similar to $A_i$.) An example could be "The amount of dirt is rather high and the amount of water is low." The result is a fuzzy conclusion on $Y$:

$$B^* = R \circ A^*,$$

where

$$R = \bigvee_{i=1}^{r} R_i$$
(there are \( r \) rules in the base), from which the concrete (crisp) action can be calculated by defuzzification as

\[
y^* = \text{defuzz}(B^*).
\]

Examples for the \text{defuzz}() operator are the \textit{center of gravity} method

\[
\text{defuzz}(S) = \frac{\int yS \, dy}{\int S \, dy},
\]

the \textit{first maximum} method

\[
\text{defuzz}(S) = \inf \{ Y_{\max} | Y_{\max} = \{ y_j | \exists y \in Y : S(y) > S(y_j) \} \}
\]

etc.

As we shall see, the method in the form as it was originally proposed involves still a rather high computational complexity, and soon a simplified version was developed by E. H. Mamdani, which produced very good results in the control of highly nonlinear systems [2]. The essential difference is that the composition is executed in the projections, and so

\[
B^* = \prod_{i=1}^{k} B_i^*, \quad \text{where} \quad B_i^* = \bigcup_{j=1}^{r} \left[ \text{height} \left( \bigcap_{i=1}^{k} \left( A_i^* \cap A_{ji} \right) \right) \right] \cap B_j.
\]

In these formulae, \( \cap \) is the \( t \)-norm and \( \cup \) the \( s \)-norm. Originally they were \( \min \) and \( \max \); however, in a later version of the Mamdani technique, Larsen proposed the algebraic \( t \)-norm rather than \( \min \) [4]. The method is illustrated in Figure 1.

The success of this family of control algorithms led more than 10 years later to the "fuzzy boom" in Japan, of which the start could be observed in 1987 during the Second IFSA World Congress, held in Tokyo. A large number of industrial applications have appeared since then, in the first years mainly in Japan, but soon also in Korea and the USA, and recently also in Europe, first of all in Germany. Applications include home electronics goods and other electric home appliances; train systems, automobiles, and other vehicles; various kinds of industrial processes; robots; space research systems; etc. With the single exception of the unmanned helicopter control by M. Sugeno [5, 6], all these applications are more or less founded on the Zadeh-Mamdani(-Larsen) algorithm.

A common feature of almost all of these applications (except the helicopter) is that the number of state variables (inputs) is small. The reason is that even the simplified computational approach leads to exponential space and time complexity in terms of the state variables. In this article, the computational complexity aspects of these various control algorithms will be studied, and some further methods will be discussed, from the point of view of their suitability for complexity reduction.
Before that, however, it is necessary to mention that there is another approach to fuzzy rules, the Takagi-Sugeno model [3]. In this model, the typical form of a rule is the following:

"If x is A then y = ax + b,"

or, in a somewhat extended form,

"If x is A then y = f(x)."

The essential idea is that if an approximate model of the system is known, but is nonlinear (or, more generally, too complicated to be used directly), it can be approximated by partitioning the input domain into (partially overlapping) fuzzy areas and defining linear (or simpler) functions for each domain, which express the connection between output and input quasianalytically. It should be noted that a might be a vector of coefficients, each one corresponding to one of the input variables $x_i$. If an observation on $X$ is known, the conclusion will be calculated by taking into consideration all the functions $f_i(x)$ for which $A^* \land A_i \neq \emptyset$, weighting each $f_i(x)$ by e.g. the same $w_i$ as in the other algorithms. As this type of the rules gives crisp
values for $y$ directly, there is no defuzzification phase involved if the observation is a singleton. Otherwise, the observation itself must be defuzzified, and the weighted substitution value belonging to the observation centroid will be calculated. For an illustration, see Figure 2.

The Takagi-Sugeno model can be further extended, to fuzzy functions in the consequent part. Then, however, defuzzification is necessary. In this paper the topic will not be discussed in detail.

To conclude this section, a few recent monographs are referred to that give good overviews of both the Mamdani and the Takagi-Sugeno (TS) control, discussing in some detail the "intersection" of the two, namely the "Sugeno controller," this latter being the special Mamdani controller where $B_i$ are crisp singletons $b_i$, or where in the TS model has no $a_i x$ part, which leads to identical rules [7–10].

Figure 2. A version of Takagi-Sugeno type control.
3. ANALYSIS OF THE BASIC LINGUISTIC CONTROL ALGORITHMS

Let us review briefly the basic method. The model of the system is given first in a linguistic form. Rules like

"If $x_1$ is negative medium and $x_2$ is negative very large and ... and $x_k$ is negative zero (i.e. close to zero but on the negative side of the real line) then $y$ is positive small"

describe the behavior of the system by approximate fuzzy values for $x_i$ and $y$. The individual rules can be formulated quite independently of each other. It must be mentioned that there is no difficulty in having more than one variable also on the right hand side of the rules (the "then" part), but it is always possible to decompose the rules into as many single output ones as there are output variables involved, which fact reduces the complexity. So, from now on, we always suppose that the output is one dimensional—i.e. there exist separate submodels for every output variable.

The crucial new element in a rule like the one above is the involvement of the linguistic expressions $A_i$ and $B$ in contrast with the classic expert system approach using only exact values. An expression like "negative small" can be represented by a fuzzy set of the universe of all possible values for $x_i$ (which is usually a bounded set of ordered values: an interval or a set of points of an interval in practical applications); even more, it is in most cases a convex and normal fuzzy set, or briefly a CNF set (i.e., all of its $\alpha$-level cuts are connected, and its core is not empty). In practice, the membership functions for representing rules are usually piecewise linear, as it is much easier to calculate with such functions. Trapezoidal and triangular membership functions are rather common. Figure 3 presents two possible sets of terms, one containing 11 linguistic terms from "negative very big" to "positive very big," the other one only 7, from "negative big" to "positive big." It is important to observe that both these term sets form 0.5-covers of the universe $X$ in the sense that for every $x \in X$ it is true that

$$\exists A_i : A_i(x) \geq 0.5.$$  

Of course, it is not necessary that these terms be represented by symmetrical triangular membership functions. In practice, smooth "bell shaped" or "S-shaped" functions are always replaced with piecewise linear membership shapes—in the simplest case, by trapezoidal or even triangular functions. The use of equal and symmetric triangles is common in some simple applications, but more complex covers require other (often not uniform) shapes.
An observation \( A^* \) on \( X \) can be expressed by a similar membership function. It often happens (indeed, it is almost always true) that the observation is not identical with any of the antecedents \( A_i \) in the cover. In some applications the observation is a crisp (singleton) value, so its membership function is equal to 1 for a particular \( x^* \) and 0 for every other \( x \).

In both cases concerning the type of \( A^* \) (fuzzy or crisp), the Zadeh-Mamdani technique and its variants involve the use of the weighting factors

\[
 w_{ij}^* = \max_{x_j} \left\{ \min \{ A^*(x_j), A_{ij}(x_j) \} \right\},
\]

either as a delimiter (min \( t \)-norm) or (in the case of the Larsen version) as a shrinking factor (algebraic \( t \)-norm) for the corresponding consequent. For every rule, a single weight is calculated e.g. by

\[
 w_i^* = \min_j \{ w_{ij}^* \}.
\]

In the Zadeh-Mamdani version, the conclusion is obtained by the formula

\[
 B^* = \max \{ \min\{ w_i^*, B_i \} \};
\]

in the Larsen version, by

\[
 B^* = \max \{ w_i^* B_i \}.
\]

For the first version of the algorithm see Figure 4.

This kind of calculation is considered to be a kind of interpolation, as the conclusion is derived from several (at least two) rules, and the individual consequents in the fired rules are weighted according to their...
similarity or degree of overlapping with the corresponding antecedents. By 
this interpolation, the number of rules might be significantly reduced, 
while in the classic approach, for every essentially different case, another 
rule is needed [as, e.g., the state space must be divided into q intervals, 
and for every $x_i (i = 1, \ldots, q)$ another rule is constructed: "If $x$ is $x_q$ 
then ... "]]. Here a reasonably dense cover offers full information on the 
system—provided that it has continuous behavior.

The large number of industrial applications based on these algorithms is 
due to their simplicity, and especially their low complexity, compared to 
the classical expert control systems. Nevertheless, it is obvious that al-
though applying fuzzy sets as input terms reduces the number of rules in 
the knowledge base drastically, even so, the method has rather clear limits 
concerning the number of input state variables.

The limitation of the domain of application of these algorithms is in 
their inherent complexity, a result of the obvious fact that even if only a 
very few fuzzy terms are necessary to cover all possible situations in a 
given input, the number of possible combinations grows very fast with the 
number of variables. Taking the most extreme situation, when for all input 
variables the distinction between only two states is necessary (if there were
no two different states the "variable" could be considered as effectless for
the given input, and could be eliminated from the model), the number of
possible combinations for all input variables can be still expressed by an
exponential function, the power of two. In general, having \( k \) input state
variables and supposing that for the \( \alpha \)-cover of \( x \) in the sense shown in
Figure 3 for \( \alpha = 0.5 \), for every input variable (i.e., in every dimension of
\( X \)), \( T \) [or approximately \( T, \vartheta(T) \)] different linguistic terms are necessary.
Then the number of total rules covering \( X \) at least to \( \alpha \) is

\[ |R| = \vartheta(T^k), \]

which is very high if \( k \) is not very small. There are two possibilities to
decrease this value. One is the reduction of \( T \), which does not eliminate
the inherent exponentiality of the expression, though it does extend the
applicability to higher \( k \); and the other is to reduce \( k \) itself, but even here,
the exponentiality remains, although with lower \( k \), much higher \( T \) might
be tractable. Of course, a combination of the two techniques is also
possible. In the next sections, examples of both techniques and of combi-
nations will be shown. Before that, however, we briefly investigate the
computational complexities of the classical fuzzy control algorithms. In our
calculations of the time complexity we adopt the notion of uniform
complexity (cf. [11]), in the sense that every step (min, max, multiplication,
addition, etc.) is considered to be one unit. This method is justifiable in
that all these operations have an upper bound on their execution time, and
the maximal upper bound is a good estimate for the worst case of the total
execution time. In some calculations we will be satisfied to express only the
order of the complexity (disregarding constant factors), while in some
other cases (especially if a given method enables the reduction of time by a
constant factor), more exact formulae will be given.

The calculations in this section follow basically the calculations in [12],
but they are more precise and include some of the more recent results
obtained in [13].

Let us examine first the Zadeh algorithm in its basic form, which works
with the rules as fuzzy relations in the \((k + 1)\)-dimensional hyperspace,
and in which defuzzification is done directly for the hyperbody represent-
ing the union of all modified conclusions. This algorithm has a very high
complexity:

\[ \vartheta_Z = (r + 3)T^{k+1} + T^2 + kT + k = \vartheta(rT^{k+1}). \]

As has been stated already, the system is covered by rules—i.e., the rule
base is dense—if

\[ r = |R| = \vartheta(T^k). \]
If \( r \) is replaced by the order of the size of a dense rule base, we obtain
\[
\mathcal{C}_Z = \Theta(T^{2k+1}).
\]
The input (rule base size) and observation complexities are
\[
\mathcal{J} = r(k + 1)T = \Theta(r(k + 1)T) = \Theta((k + 1)T^{k+1})
\]
and
\[
\mathcal{A}^* = kT = \Theta(kT).
\]

From the uniform complexity viewpoint, there is no difference between the two methods in calculating with the projections. Both the Mamdani and the Larsen algorithm have a considerably lower complexity, having
\[
\mathcal{C}_{ML} = 2r(k + 1)T + 3kT + k = \Theta(r(k + 1)T) = \Theta((k + 1)T^{k+1})
\]
for the number of steps, while \( \mathcal{J} \) and \( \mathcal{A}^* \) are the same. Comparing the two results, we find that
\[
\frac{\mathcal{C}_Z}{\mathcal{C}_{ML}} = \Theta\left(\frac{T^k}{k + 1}\right).
\]
The Mamdani algorithm is illustrated with a simple base, consisting of only two rules, in Figure 4.

The reduction of the number of steps is roughly by the square root. It is not surprising that all real applications use some version of the latter approach. Of course, the essential exponentiality did not disappear; as was stated earlier, exponentiality cannot be eliminated completely.

When comparing these two approaches, besides complexity it should be investigated whether the two are equivalent concerning their behavior from the control point of view. How well do these methods work as models and as control techniques? In order to be able to give a simple evaluation, a kind of sensitivity measure will be considered. It is defined as the minimal \( \Delta y \) for a given \( \Delta x \). In this sense, low sensitivity will mean that for a large change in the input, the reaction of the system is little or nothing. If the sensitivity of the two models in this sense is compared, it is found that the original Zadeh algorithm is remarkably more sensitive. In contrast, it is rather easy to construct cases where the observation is considerably changed and still the Mamdani algorithm does not indicate it, i.e. it calculates exactly the same conclusion. The reason is that only the heights of the intersections of the observation and the antecedents are taken into consideration, and the height may remain the same while the centroid (peak, core, etc.) of the observation shifts and at the same time the width (support size, expressing a degree of vagueness) also changes. The Larsen
method is slightly better in this respect, as using the algebraic $t$-norm for weighting the consequents allows more flexible information transmission from the inputs to the outputs. However, compared to the Zadeh algorithm, the sensitivity is less, as the latter almost always reacts more intensely in such cases. No analytical examination of this problem is known; however, a large number of experimental data support the statements about sensitivity.

An obvious question was how to preserve the good sensitivity of the Zadeh algorithm while reducing its very high complexity. A compromise solution (however limited in its applicability because of the strong restrictions concerning the antecedents and consequents of the rules) was proposed in [12, 14], but even so, only the exponentiality caused by the size of the state space could be eliminated:

$$\mathcal{C}_K = rL^k + rL^{k+1} + L^k + 2T + 1 = \mathcal{O}(rL^{k+1} + T),$$

where $L$ is a small bounded constant (the maximum support size in one dimension). If $L^{k+1}$ can be kept low, this is near to $\mathcal{O}(r)$, i.e. similar to the Mamdani algorithm. For this, however, a very high price must be paid: the higher $k$ is, the smaller $L$ can be; the denseness of the rule base cannot be kept. Then it is also useless that $r = \mathcal{O}(T^k)$ is kept; by having less and less rules covering a smaller and smaller part of the state space, the validity of the model becomes low, and moreover, the algorithm will be inapplicable, as the default observation will not overlap with any of the antecedents.

It is clear that such compromise reductions of the complexity do not extend the applicability of the two classical algorithms significantly.

Finally, a brief remark on the complexity of the Takagi-Sugeno algorithm is necessary. Depending on the type of $f_i(x)$,

$$\mathcal{C}_{TS} = c_1r + c_2 = \mathcal{O}(T^k)$$

where $c_1$ and $c_2$ are suitable constants expressing the number of steps necessary to obtain $y_i$ from each rule (including the weighting step and the construction of the weighted sum of consequents, as well as the sum of the weights themselves) and the defuzzification of the observation if it is not a singleton value, and $c_2$ refers to the number of steps necessary to obtain the final result for $y^*$ (this time simply executing a division, and eventually a few administrative steps). This complexity is slightly better than for the previous approach; however, for its use, it is necessary to have another type of knowledge of the system than in the case of the Zadeh-Mamdani type rules. Figure 2 illustrates a version of the Takagi-Sugeno algorithm [$x^* = \text{defuzz}(A^*)$].
4. SPARSE RULE BASES AND INTERPOLATION ALGORITHMS

As mentioned in the previous section, there are various possibilities to reduce \( r \) and \( \mathcal{E} \), one of them being the reduction of \( T \). It is interesting to mention that besides the goal of reducing complexity, there are many other motivations leading to sparse models. A good example is where the starting term set is of the \( \alpha \)-cover type but by tuning the rules, the antecedents are partially shifted and shrunk so that the tuned model will contain "gaps" where no antecedent is present to any positive degree [15].

It is worthwhile to discuss rule tuning in general. When no exact model of a system to control is available, it is not easy to determine the membership functions of \( A_i \) (and \( B_i \)). Even if the possible shape of these terms is strongly restricted (as e.g. to triangular or trapezoidal), it takes a long training period to find the suitable breakpoints, unless some primary information is available. In many earlier applications these functions were determined by some kind of iteration. The starting point was a "neutral," essentially linear rule system, which usually gave a very bad model of the given specific system. By successive iteration, the shape of the rules (including their positions along the variable axes) was altered so that the reactions of the system confirmed more and more the newly obtained rules as parts of an adequate model. By tuning the rules, it is possible to obtain a certain degree of finer interpolation for the areas between the core areas of the antecedents, as long as the \( \alpha \)-cover property is satisfied. In the case where by tuning the cover becomes only partial, the applicability of the classical methods is restricted to the areas where the antecedents still overlap.

Of course, if the model is constructed directly on the basis of some experts' knowledge, it is also likely that there will be gaps in that knowledge, especially in those areas where the states are assumed by the system only in very rare cases. (These cases should not be overlooked, however, as the lack of an acceptable behavior of the controller at the assumption of such a state might lead to catastrophes in the system.)

Before dealing in more detail with sparse rule bases, it must be mentioned that recently some very interesting results have been published which construct the fuzzy model without the subjectivity of the human expert, by evaluating a large amount of measured data on the system's input-output states. The methods aim at obtaining as few as possible terms (with as large as possible supports) while not losing the property of adequate modeling. The terms are determined on the basis of clusters detected in the data, the areas of high density around the centroids indicating the cores, and lower and lower densities of elements in the clusters indicating the slopes of the terms. A very rich paper (containing
many other aspects, as e.g. methods for detecting and eliminating irrelevant variables in the initial hypothetical model) is [16], which applies the fuzzy c-means method for clustering the state data [17]. In [16] some examples are shown; e.g., it is possible to construct a good model consisting of only seven rules for a system with five input variables.

While discussing the construction of models through clustering, it must be mentioned, too, that it is possible to apply the same approach to input-output data pairs, for obtaining a Takagi-Sugeno style model [18]. In a similar way, clusters with quasilinear behavior are searched for, and so the functions $f_i$ are determined.

Let us return now to the problem of sparse rule bases. To cope with rule bases containing gaps, i.e. fuzzy models where typical cases are represented by the rules but the rule base does not contain and will never contain full information on the system, needs completely new techniques of reasoning and control. The essential idea is that in the gaps, approximation methods are used for estimating the conclusion by the examples of the surrounding known rules. This approach is called rule interpolation, and should be clearly distinguished from the interpolative reasoning done by parallel evaluation of several partially overlapping rules, where the interpolativity is an indirect consequence of the algorithm. The method of rule interpolation itself has been discussed in detail in this journal [19], so it is only briefly summarized. It is necessary to see clearly that rule interpolation works only if the state space has some "nice" properties; these properties will also be discussed. Furthermore, the interpolation algorithm is the most advanced member of a family of approximate reasoning algorithms, which all depend on constructing the conclusion by the analogy of at least one known rule, and referring to some kind of analogy or similarity among the antecedent(s) and the observation, establishing a similar relation among the consequent(s) and the conclusion.

For the interpolation it is necessary that the system does not behave too unexpectedly in the areas where the model does not really cover it; in this sense it can be said that the full information provided by the dense cover would be highly redundant. Luckily enough, in practice, such nice behavior can be expected in most cases: even though many systems are not linear, it can be expected that they are smooth in some general sense and they do not have very sudden and very large changes in their reactions to small changes of the input. At this point we must stress that the previous statement is not some euphemism for mere continuity: a system might be continuous (i.e., the mapping from the inputs to the outputs might be a continuous function) and still very rhapsodic. For example, if our system has the analytic model

$$Y = x + \sin(1000x) \quad \text{where} \quad X = [0, 2\pi],$$
it will hardly be possible to model the system by e.g. 20 fuzzy rules, although it is continuous enough! A good term for the class of systems where the following algorithms are applicable is interpolative system, referring to the interpolative approach in the next section. It must be seen clearly that interpolativity is not an absolute idea; it always depends on the fineness of the model. Every system with finite zeros of the first derivative in the (bounded) universe of discourse \( X \) is interpolative for even sparse rule bases with a large enough number of rules. It is obvious that even dense rule bases cannot be dealt with by any of the classical algorithms if they are not interpolative, as both the Zadeh and the Mamdani algorithms use interpolativity in a broader sense. The Takagi-Sugeno model supposes some kind of function interpolation in the overlapping domain areas, too.

Interpolativity can be formulated more precisely by Shannon's sampling theorem (cf. [21]): For a given (sparse) rule base a system is called interpolative if it is reconstructible in the Shannon sense from the samples represented by the centroids of the rules. This means essentially that the (local) density of the rules (frequency of the antecedents, each antecedent represented by some characteristic point, as e.g. the centroid of the antecedent, usually having the nature of a vague cluster of crisp values) should be at least twice as high as the highest frequency component of the system descriptor function.

Let us now investigate the rules from the point of view of their semantic meaning and how it affects the possible reasoning algorithms. There are two essentially different ways to interpret the semantics of fuzzy If... then rules and rule bases. These two ways can be called implication and mapping (or graph) interpretation. An analysis of both methods with comparison of their behavior can be found in [20].

If the rules are decoded as implications and the observation as a simple logical statement itself, then modus ponens is suitable for obtaining conclusions—supposing that the observation fits one of the antecedents. The rule

"If \( x \) is \( A \) then \( y \) is \( B \)"

combined with a fact

"\( x \) is \( A \)" \( (A^* = A) \)

is simply interpreted as the logical formula

\[ A \land (A \rightarrow B), \]

which implies \( B \).

It is a more complicated situation if

\[ A' \land (A \rightarrow B) \]
is known, where $A^* = A'$ fits $A$ only partially. Theoretically, it is possible to obtain $B'$ as a conclusion by using an appropriate method of finding $B'$ that is in some sense in a similar relation to $B$ as $A'$ is to $A$. This kind of inference is often called *modified* or *extended modus ponens*, but it is a very controversial question whether the extension of a strict Boolean logical tautology in this direction has the correct theoretical foundation at all, and what is the meaning of “similarly modified” when the new version of $B$ is calculated to obtain the conclusion.

It can be generally observed that the implication approach is not very popular among fuzzy control engineers because of the technical difficulties with handling the implications in $X \times Y$. Just one (rather subjective) aspect is mentioned: In the whole area of $\overline{A}$ the logic implication $A \rightarrow B$ is automatically true, as nothing is known there that contradicts the truth of the logical formula represented by that particular implication. As $A$ has usually a comparatively small support, that means that the logic function representing a single rule is almost everywhere equal to 1. It is somewhat disturbing that everywhere where there is no information (no rule), the membership function is 1 and obtaining information means “taking away” from the already existing membership value. There are however more serious difficulties with the implication view.

As mentioned above, there exist some exact techniques which offer ways of constructing $B'$ from $B$ and the known difference of $A'$ and $A$. These are referred to in the literature as *gradual reasoning*, *the analogical reasoning*, *revision principle*, etc. [22-26]. A common feature is that they work with observations where

$$\text{supp}(A^*) \cap \text{supp}(A_i) \neq \emptyset.$$ 

The idea of *gradual rules* supposes that the If...then rules implicitly contain the semantics, that the properties (terms) in the rule may be satisfied in different degrees, and that the more the term $A$ is true in the case of the observation, the more should $B$ also be satisfied in the calculated conclusion. **Analogous** or **analogical reasoning** is slightly different, but has nevertheless a deep connection with the above. Analogies are sought in the sense that the observation is evaluated as to its similarity with the antecedent in question. The similarity is based on the notion of (crisp) distance. The distance of two fuzzy sets can be defined in many ways. The paper [24] gives a good overview of many such distance definitions and finally selects one as the best fitting for a definition of similarity. (The farther, the less similar, is rather obvious.) A common feature of all these definitions of distances is that they are crisp (single) values, calculating the shortest or the average distance of the two fuzzy sets. In the normalized case the
chosen distance is 0 if and only if the two fuzzy sets are equal, and is maximal (1) if their supports are disjoint. This latter fact restricts the usability of this definition of similarity considerably. If the observation falls completely into a gap, no similarity is discovered, i.e., no analogical conclusion can be obtained. A considerable extension of this idea will be discussed later.

The revision principle is the most advanced technique of this family of analogical reasoning, as it is suitable for the pointwise construction of the membership function of the conclusion. For any rule, a semantic curve is determined by geometrical methods, and this curve is used for reference when a new conclusion is sought for a given observation. The method is presented for overlapping antecedent and observation, but it is not impossible to extend the technique to the more general case. The algorithm in its original form is not very time efficient, but it is possible to construct very fast versions for piecewise linear membership function shapes.

All three approaches point to the second type interpretation of the semantics of rules, which considers every rule essentially as a representative “point” of a fuzzy mapping in the sense

\[ R : \mathcal{P}(x) \to \mathcal{P}(Y). \]

Here \( \mathcal{P}(S) \) denotes the fuzzy power set, the set of all fuzzy subsets, of the universe \( S \), so the mapping assigns a fuzzy set of \( Y \) to every fuzzy set of \( X \), at least theoretically. This view is called the “graph view” in [20], a name that is somewhat ambiguous but refers to a very essential point: In practical systems, the mapping is usually of a very special type in that it can be represented by a kind of “graph,” a curve (or hypersurface) with fuzzy borderlines. If such a special property is accepted, it is rather obvious to introduce some restrictions on the elements of \( \mathcal{P}(X) \) and \( \mathcal{P}(Y) \). We suggested \( \mathcal{E}(X) \), etc.: the subsets of \( \mathcal{P}(X) \), etc. containing only the convex and normal fuzzy sets (CNF sets) of \( S \). This restriction guarantees that the “graph” will be unambiguous in the sense that it has a connected central area (core) which absolutely belongs to the mapping (even if that is not a single curve—or hypersurface—but maybe a “band”), and the farther a point in \( X \times Y \) is from that core, the less its membership degree belongs to the “graph” (mapping). There are no unconnected core areas in any sense, and it cannot even happen that somewhere the mapping has a local maximum in the membership function (though less than 1).

According to this interpretation, \( R \) assigns a CNF set of \( Y \) theoretically to every CNF set of \( X \). Instead of a full identification of the system (clearly, an impossible aim), several representative “points” (i.e. pairs of CNF sets of \( X \) and \( Y \), respectively) are given: the rules in the base. The number and especially the denseness of these points is characteristic of the
goodness of the approximation of $R$. Figure 5 presents both a dense and a sparse rule base in the “support view” (seen from “above,” i.e. from the direction of the $\mu$-axis) in the direct product space of $X$ and $Y$; the oblongs represent the projections of the individual membership functions to $X \times Y$. Each oblong is a fuzzy point, altogether featuring the fuzzy graph.

In this mapping interpretation of the semantics, the problem of fuzzy control can be defined as searching for the substitution value $B^* = R(A^*)$ if $A^*$ is given. Regardless of what algorithm is used, the probability that $A^*$ exactly fits one of the known $A_i$'s is equal to zero. So, virtually always, only an approximation can be sought for, instead of the exact substitution value of $R(A^*)$.

In most of the practical applications, luckily enough, the systems involved can be approximated quite well by not too complicatedly shaped...
fuzzy mappings (in the sense of the sampling theorem as mentioned above). The behavior of the mapping is not too unpredictable in any system that a human operator can control with an acceptable level of accuracy. Practical results have shown that fuzzy control is usually more adequate than manual control even then if the fuzzy rules are constructed by the experience of the controlling operator(s)! (For this see e.g. a series of papers in the contributed volume [27].)

For the interpolation algorithm, it is necessary to take the extension principle and the resolution principle into consideration. The latter describes the decomposition of fuzzy sets to $\alpha$-cuts:

$$F = \bigcup_{\alpha \in [0, 1]} \alpha F_{\alpha}.$$  

(Here union means the maximum.) The former states that the solution of a problem for fuzzy sets can be found by solving first for arbitrary $\alpha$-cuts (N.B. these are crisp sets) and then extending the solution to the fuzzy case. Interpolation is done for every level independently.

In this approach, besides allowing only CNF sets in the rules, some further conditions must be fulfilled. The state variables (including $X_i$ and $Y$ as well) must be bounded and gradual in the sense of [22], which fact guarantees that a full ordering in each of them exists. If all dimensions of $X$ (and also $Y$ itself) are ordered, it is possible to introduce a partial ordering (po) among all elements of $X$. With the help of the $\alpha$-cuts, it is possible to introduce a po among the CNF sets of each $X_i$ (or $Y$, denoted by $<$, or simply $\prec$). This po is similar to the ordering among fuzzy numbers, as in [28]. If

$$\forall \alpha \in [0, 1]: \inf\{F_{\alpha}\} \leq \inf\{G_{\alpha}\} \& \sup\{F_{\alpha}\} \leq \sup\{G_{\alpha}\}$$

then $F$ and $G$ are comparable, i.e., $F \prec G$. The convexity of the sets guarantees that the cuts are connected, and this means that $F \prec G$ is stated only if for all levels, every point in the corresponding cut of $F$ is below the maximal point (or upper bound) of the cut of $G$, and also every point in the cut of $G$ is above the minimal point (or lower bound) of the corresponding cut of $F$, both in the sense of $< \prec$ in $x_i$. The normality is necessary, as otherwise the $\alpha$-cuts above a certain bound would not exist at all.

Among the comparable fuzzy sets, a new concept of distance can be introduced. The two extremal points of the cuts represent the whole cut (because of the convexity of the set), and their pairwise distances are
defined as the *lower* and the *upper distance* at the given \( \alpha \)-level. These distances form together two families. In the general case, it is impossible to represent the family of \( \alpha \)-distances by fuzzy sets. The reason is that sometimes for \( \alpha_1 < \alpha_2 < \alpha_3 \); \( d_{L,\alpha_1}(A, B) < d_{L,\alpha_2}(A, B) \) but \( d_{L,\alpha_3}(A, B) < d_{L,\alpha_2}(A, B) \), too; then for an imaginary fuzzy set of distances one "slope" would change sign—obviously an impossible shape. (See Figure 6.)

The lower distance is defined as the (extended) fuzzy set constructed from the \( \alpha \)-cuts obtained as the distances of the infima of the \( \alpha \)-cuts of the two fuzzy sets, and the upper distance similarly with respect to from the suprema:

\[
\mu_{d_{L,F,G}}(z_i) = \sum_{\alpha \in [0,1]} \frac{\alpha}{D(\inf\{A_\alpha\}, \inf\{B_\alpha\})},
\]

\[
\mu_{d_{U,F,G}}(z_i) = \sum_{\alpha \in [0,1]} \frac{\alpha}{D(\sup\{A_\alpha\}, \sup\{B_\alpha\})},
\]

where \( z_i \in Z \) is the variable representing the possible values of distance in \( X_i \). In multivariable state spaces, it is necessary to normalize \( Z_i \), i.e.,

![Figure 6. Comparable fuzzy sets A < B < C and examples for upper and lower distances among them \( |d \cup (B, C)\) is not a fuzzy set!][1]
The multidimensional distance might be calculated e.g. as a Euclidean distance, or, more generally, as an arbitrary Minkowski distance. This definition has some relation to the extended Hausdorff distance according to [29], but it does not involve its high complexity. Figure 6 depicts some examples for the lower and upper distances of comparable CNF sets.

By the concept of fuzzy (lower and upper) distance, the closeness of two comparable fuzzy sets can be determined even if their supports are disjoint. The distance of two fuzzy sets will be zero (i.e. the crisp distance set with the characteristic function 1 over \( z = 0 \) and zero elsewhere) if the two fuzzy sets are identical, and will be maximal (if using the Euclidean distance, \( \sqrt{k} \)) if the two sets are the first and the last element of the lattice defined by the po (the two farthest crisp sets). A detailed discussion of ordering, distance, and closeness of CNF sets can be found in [30].

Having the concept of "fuzzy distance," the classical methods of function approximation are applicable on almost arbitrary rule bases. Using the resolution principle, a rule base is represented by a family of hyperintervals in \( X \times Y \): namely, for every \( \alpha \), the points \( A_{i\alpha} \) and \( B_{i\alpha} \) form a hyperinterval for CNF sets. The domain \( A_{i\alpha} \times B_{i\alpha} \) is unambiguously represented by only two points in \( X \times Y \): the minimal and maximal points. Figure 7 gives an example for the representation of a rule by four points. In the example, the rules consist of trapezoidal membership function terms which have breakpoints only at 0 and 1. The rule is represented in the Cartesian product space by the four points \( P(0, L) \) and \( P(0, U) \) (for the support of the rule) and \( P(1, L) \) and \( P(1, U) \) (for the core of the rule).

It is not obvious (and also not true) that if the original membership functions are piecewise linear then applying some simple linear transformations on them in order to approximate the conclusion the result is also linear. A detailed examination of this problem follows later in this paper.

The simplest method for approximating the behavior of the fuzzy mapping between two neighboring rules is linear interpolation. This can be applied if the observation is located so that

\[
A_{i1} < A^* < A_{i2} \quad \text{and} \quad B_{i1} < B_{i2}
\]

For an arbitrary observation it is reasonable to select the closest comparable antecedent on each side.

The fundamental equation of linear interpolation (referring to the "fuzzy distance") is

\[
d(A^*, A_{i1}) : d(A^*, A_{i2}) = d(B^*, B_{i1}) : d(B^*, B_{i2}).
\]
Figure 7. Representation of a trapezoidal rule by its support and core in $X \times Y$.

(This equation is in reality shorthand for an infinite family of equations, one for every $\alpha \in [0, 1]$.) From here, the following solution is obtained for $B_{\alpha}^*$:

$$\min\{B_{\alpha}^*\} = \frac{1}{d_L(A_{\alpha}^*, A_{i1})} \inf\{B_{i1}\} + \frac{1}{d_L(A_{\alpha}^*, A_{i2})} \inf\{B_{i2}\}$$

$$\max\{B_{\alpha}^*\} = \frac{1}{d_U(A_{\alpha}^*, A_{i1})} \sup\{B_{i1}\} + \frac{1}{d_U(A_{\alpha}^*, A_{i2})} \sup\{B_{i2}\}$$

As mentioned, in multiple input cases the resulting distance is obtained by taking the Euclidean (or Minkowski) sum. An illustration for interpolation with triangular terms is depicted in Figure 8 (results obtained by the RULEINT program [32]).

The principle of interpolating two rules can be extended to many different algorithms. Several software packages have been developed, and the behavior of various methods has been experimentally analyzed with their help. The most obvious extension of the interpolation of two rules is
Figure 8. Two examples for multiple dimensional linear rule interpolation by RULEINT.

The interpolation of $2n$ rules ($n$ and $n$ flanking the observation in the sense of $\prec$) where pairs of flanking rules are considered and the farther the elements of the pair are located from the observation, the less weight the respective consequents play in the construction of the conclusion. The formulae for this type of interpolation are as follows:

$$
\min\{B^*_\alpha\} = \frac{\sum_{i=1}^{2n} \frac{1}{d_L(A^*_\alpha, A_i \alpha)} \inf\{B_{i \alpha}\}}{\sum_{i=1}^{2n} \frac{1}{d_L(A^*_\alpha, A_i \alpha)}},
$$
\[
\max(B^*_\alpha) = \frac{\sum_{i=1}^{2n} \frac{1}{d_U(A^*_\alpha, A_{i\alpha})} \sup\{B_{i\alpha}\}}{\sum_{i=1}^{2n} \frac{1}{d_U(A^*_\alpha, A_{i\alpha})}}.
\]

More details on this method can be found in [19, 33].

The interpolation algorithms have the common feature that they use formulae that can be calculated once the rule base is given. Suppose that the denseness of rules is approximately equal for every \(X_i\), and that there is a certain regularity in the distribution. Then, in every dimension of \(X\), there are approximately

\[r_i = \sqrt[2]{r}\]

different antecedents. For simplicity, this value will be denoted also by \(T\) (although it has little to do with the number of terms forming in \( \alpha \)-cover). The total number of hyperintervals flanked by a different set of antecedents is \((T - 1)^k\), somewhat less than \(r\). The situation is similar if the antecedents are less regularly distributed, except that in such a case only a small fraction of the total state space is covered by these flanked intervals, i.e., the model is rather incomplete. Anyway, in the worst case, there are \(r - 1\) flanked intervals, so \(r\) is an upper bound for the number of different interpolation functions occurring in the control of a certain system. All these functions must be calculated in advance.

Independently of the type of interpolation (two or more rules, linear or nonlinear, spline interpolation, regression, etc.), the calculation of the function (or, more exactly, its coefficients for the given hyperinterval) takes a constant number of steps. In this sense, the preparation phase for the algorithm needs \(\Theta(r)\) steps. Even if \(r\) is still high (it should be in the order of \(T^k\), unless the system conforms very well with the function type chosen for interpolation; e.g., it should be close to linear if linear interpolation is applied), these steps must be done before the actual control starts, i.e., not in real time. As the result of these steps, a new model is obtained, a kind of extended Takagi-Sugeno model, which might be formulated by the following type rules:

"If \(A^*\) is in the interval determined by \(\{A_{i_{11}}, A_{i_{12}}, \ldots, A_{i_{1n}}\}\)
then \(B^* = F(A^*)\),"

where \(F\) denotes a fuzzy mapping, i.e., a set of functions (theoretically a different one for every \(\alpha\), and the lower or upper bound), and \(F(A^*)\) is shorthand for the family of different substitution values in the sense of the resolution principle. The indicated antecedents are the closest flanking
ones. As the interval determined by \( \{A_{ij,1}, A_{ij,2}, \ldots, A_{ij,n}\} \) is a fuzzy domain itself, and in the case of well-chosen \( A_{ij} \) the whole set of these intervals covers the input space, these rule differ from the TS type ones only in the consequent part, where \( F \) replaces \( f \), the latter being the crisp special case of the former.

In order to obtain an estimation for the complexity of the interpolation algorithm, the number of steps calculated in real time for every observation must be taken. Once the interpolation function is given, it takes a constant number of steps to obtain the result for a given \( \alpha \) (for linear interpolation and only one input, seven steps, plus a few more for comparisons).

For determining into which of the maximally \( r - 1 \) hyperintervals the actual observation \( A^* \) fits, the selection of the proper interpolation function is done in \( r - 1 = \mathcal{O}(r) \) steps. Then \( \leq c_1 k + c_2 \) steps are necessary for the substitution of a single end point of one of the \( \alpha \)-cuts (in general, it is reasonable to suppose that each of the \( k \) inputs will occur in the formula, and the number of operations will be proportional to this number; a few additional steps must be executed, like the division in the linear case), and all these steps must be repeated for the other end point as well [altogether \( \leq 2(c_1 k + c_2) \) steps]. Finally, this procedure must be done for all important \( \alpha \)-levels \( \lambda \), which gives the final result

\[
\mathcal{C}_1 \leq r - 1 + 2 \lambda (c_1 k + c_2),
\]

\[
\mathcal{C}_1 = \mathcal{O}(r + \lambda k).
\]

There are two critical components in this equation: \( r \) and \( \lambda \). While \( r \) is reduced by accepting sparse rule bases, in the case of general style systems, it cannot be expected that the size of it is less than \( \mathcal{O}(2^k) \) (as at least two different antecedents are necessary in all of the \( k \) dimensions of \( X \) in order to have the possibility of interpolation). It is thus of interest whether \( \lambda \) is high or not. Especially in the case of not too many input variables, where \( r \) may be not very high, or if the system is close to linear (when, in the extreme case, two rules altogether, located at the minimal and maximal corners of the state space, might be enough for interpolation), or if it can be partitioned into a few approximately linear components, the \( \lambda k \) component of \( \mathcal{C}_1 \) may become the dominant one. The possibilities of reducing \( \lambda \) will be discussed later in this paper.

If the fuzzy distance of CNF sets is defined in a somewhat different way, i.e., instead of taking the infimum and supremum of \( \alpha \)-cuts, the center and the length of the \( \alpha \)-cuts are considered (the latter meaning the width of the fuzzy set), then the solutions of the fundamental equation will take the
following form:

\[
\text{centr}\{B^*_a\} = \frac{1}{d(A^*_a, A_{i1a})} \text{centr}\{B_{i1a}\} + \frac{1}{d(A^*_a, A_{i2a})} \text{centr}\{B_{i2a}\},
\]

\[
\text{width}\{B^*_a\} = \frac{1}{d(A^*_a, A_{i1a})} \frac{\text{width}\{B_{i1a}\}}{\text{width}\{A_{i1a}\}} + \frac{1}{d(A^*_a, A_{i2a})} \frac{\text{width}\{B_{i2a}\}}{\text{width}\{A_{i2a}\}}
\]

\[
\times \text{width}\{A^*_a\},
\]

where

\[
\text{centr}\{S_a\} = \frac{\inf\{S_a\} + \sup\{S_a\}}{2},
\]

\[
\text{width}\{S_a\} = \sup\{S_a\} - \inf\{S_a\},
\]

\[
d(S_{1a}, S_{2a}) = \text{centr}\{S_{2a}\} - \text{centr}\{S_{1a}\},
\]

and the distance of the widths is expressed by their ratio. This approach is suitable for extension to a general interpolation and extrapolation algorithm where an arbitrary number of arbitrarily located rules is taken into consideration:

\[
\text{centr}\{B^*_a\} = \sum_{i=1}^{r} \frac{1}{d(A^*_a, A_{i1a})} \left( \text{centr}\{B_{i1a}\} + d(A^*_a, A_{i1a}) \frac{\text{width}\{B_{i1a}\}}{\text{width}\{A_{i1a}\}} \right)
\]

\[
\sum_{i=1}^{r} \frac{1}{d(A^*_a, A_{i1a})}
\]

\[
\text{width}\{B^*_a\} = \sum_{i=1}^{r} \frac{1}{d(A^*_a, A_{i1a})} \frac{\text{width}\{B_{i1a}\}}{\text{width}\{A_{i1a}\}}
\]

\[
\sum_{i=1}^{r} \frac{1}{d(A^*_a, A_{i1a})} \frac{\text{width}\{A^*_a\}}{\text{width}\{A_{i1a}\}}.
\]

In this general approach it is necessary, however, that the weights be provided with signs according to their position. This method was experimentally examined by the RULEMINT2 software [34], and the results are promising so far; however, an analytical treatment has not been done yet.
As one of the most common methods of interpolation, we mention finally the Lagrange interpolation fitting polynomial curves to the rule points, which has proved to be useful for reducing \( r \), especially when piecewise quadratic Lagrange polynomials were used [37, 31]. Further details on rule interpolation, including not exactly fitting approximations (regression lines, Bézier curves, etc.), and some results concerning applications can be found in [13, 19, 35, 36, 40, 43].

5. THE SHAPE OF \( B^* \) AND THE VALUE OF \( \lambda \)

We have seen that rule interpolation combined with small enough sparse rule bases offers a new perspective on fuzzy control algorithms, even though it helps radically only if the rule base can be reduced to a size \( o(T^k) \), possibly to \( o(2^k) \). Then, however, the second component of \( \varrho_i \) might dominate, and the next question is whether its "unknown" member \( \lambda \) can be kept low. \( (k \) is considered to be fixed for a given application, even though before constructing the model, it might be possible to eliminate some of the initially occurring variables—as mentioned earlier.) \( \lambda \) depends on the shape of the terms in the rules, and on the behavior of the interpolation as regards the shape of the generated conclusion.

In all real applications, the shape of \( A_i \) and \( B_i \) has been considered to be piecewise linear. In the first algorithms of Mamdani [2], the shapes of the terms "negatively big," "positively very small," etc. were determined by about half a dozen typical points, and so their shape could be considered as a "curve" consisting of that many linear pieces. In later applications, more and more exclusively trapezoidal shapes were used (or, as a special case, triangular ones), meaning that for determining the terms, only four points—the two ends of the support, and the two ends of the core (in triangular functions, identical with each other)—were used, so that all terms were constructed of five (or even four) linear pieces (two of these being the horizontal lines outside the support, i.e. the areas where \( \mu = 0 \)).

Clearly, it is sufficient to deal with the three (or two) lines which determine the positive part of the membership functions. What will be the shape of the conclusion generated from trapezoidal (triangular) rules by linear (or other type) interpolation? The answer to this question will deeply influence the value of \( \lambda \); e.g., if the shape of \( B^* \) could be identified as trapezoidal (triangular) in all cases, then \( \lambda = 2 \) (the values \( \alpha = 0, 1 \)—obviously the minimal theoretically possible \( \lambda \)—could be applied, through which the best possible time complexity

\[
\varrho_i = r + 4k = \mathcal{O}(R + k)
\]
could be achieved, by applying the rule interpolation algorithm for control. On the other hand, if the shape of the conclusion is very nonlinear, it will be necessary to approximate its curve by many points, for many different levels, and that will increase the total computational time needed.

As the shape of the conclusion is investigated, it is natural to extend these investigations to another aspect. It is an important restriction that all terms should be convex and normal; otherwise the \( \alpha \)-cuts are not connected or do not exist at all for certain \( \alpha \)'s, which makes the application of this approach senseless. Multistep inference (i.e. the use of the conclusion as the "observation" of the next step) is possible only if \( B^* \) itself is CNF. The preservation of this latter property will also be discussed.

It is clear from the previous sections that the two rules \( R_1 \) and \( R_2 \) must be comparable both in their antecedents and consequents and that they should flank the observation (also in the sense of \( < \)):

\[
A_1 < A^* < A_2 \quad \text{and} \quad B_1 < B_2,
\]

otherwise the interpolation algorithm does not work.

Comparable rules may partially overlap, including the case where there is no real gap between the sets (terms) and so interpolation will not be necessary. A strictly ordered situation is when the rules and \( A^* \) are well separated, that is,

\[
\inf\{\text{core}(A_1)\} \leq \inf\{\text{supp}(A^*)\},
\]

\[
\sup\{\text{core}(A_1)\} \geq \sup\{\text{supp}(A^*)\}.
\]

Of course, the interpolation algorithm is really important if the rules and the observation are completely separated, i.e.

\[
\text{supp}(A^*) \cap \text{supp}(A_i) = \emptyset.
\]

Figure 9 summarizes the interpolation method, with the distances on level \( \alpha = 1 \). This time the shape of \( B^* \) is strongly distorted and nonlinear, indicating that only the support and core points of the conclusion are directly calculated, and the exact nature of the connecting lines, i.e. the points for \( \alpha \in (0, 1) \), is unknown.

In applications, \( \lambda \) should be small, i.e., calculations should be restricted to a small finite set of levels, which will be called the important cuts. For piecewise linear membership functions, an obvious assumption is to define the set of important cuts by the united breakpoint set \( \Lambda \). For trapezoidal and triangular sets \( \Lambda = \{0, 1\} \). The value \( \lambda = |\Lambda| \) can be chosen if the conclusion always has a linear shape between two breakpoint levels. As we shall see, this is not true in general.

The next sections are based on the results in [31].
6. THE SHAPE OF THE CONCLUSION BETWEEN TWO BREAKPOINT LEVELS

Without hurting generality, in the next calculations we suppose that the two neighboring \( \alpha \)-levels in the breakpoint set \( \Lambda \) are 0 and any \( \alpha \) (e.g. 1). All other cases can be obtained from this special case by simply scaling the flanks of the membership functions along the \( Y \)-axis. In this sense the left and right flanks of trapezoids are suitable for the general analysis of any piece in the piecewise linear membership function of the terms. The rules to be interpolated will be denoted by

"If \( x \) is \( A_1 \) then \( y \) is \( B_1 \)" \hspace{1em} (briefly \( A_1 \rightarrow B_1 \))

and

"If \( x \) is \( A_2 \) then \( y \) is \( B_2 \)" \hspace{1em} (briefly \( A_2 \rightarrow B_2 \));

the observation is "\( x \) is \( A^* \)" (referred to as \( A^* \)).
The membership function of $A_i$ is defined by the four points $P(a_{i1}, 0)$, $P(a_{i2}, \alpha)$, $P(a_{i3}, \alpha)$, $P(a_{i4}, 0)$, and similar notation is applied for $A^*$ and $B_i$. Then the equations of the left and right flanks of $A_i$ are

$$x_{i\alpha L} = \alpha(a_{i2} - a_{i1}) + a_{i1},$$
$$x_{i\alpha U} = \alpha(a_{i3} - a_{i4}) + a_{i4},$$

and similarly the flanks of the consequents are

$$y_{i\alpha L} = \alpha(b_{i2} - b_{i1}) + b_{i1},$$
$$y_{i\alpha U} = \alpha(b_{i3} - b_{i4}) + b_{i4}.$$ 

The observation has

$$x_{\alpha L}^* = \alpha(a_{2}^* - a_{1}^*) + a_{1}^*,$$
$$x_{\alpha U}^* = \alpha(a_{3}^* - a_{4}^*) + a_{4}^*.$$ 

From this and the solutions for $B_i^*$ from the fundamental equations of the interpolation, the equations of the flanks of $B_i^*$ can be determined.

**STATEMENT 1.** The equation of the left slope of the conclusion calculated from the linear interpolation of the two rules $A_1 \rightarrow B_1$ and $A_2 \rightarrow B_2$ and the conclusion $A^*$ between the two breakpoint levels 0 and $\alpha$ is

$$y_{\alpha L}^* = \frac{C_1 \alpha^2 + C_2 \alpha + C_3}{c_9 \alpha + c_{10}},$$

where

$$C_1 = c_3 c_5 + c_1 c_7, \quad C_2 = c_3 c_6 + c_4 c_5 + c_1 c_8 + c_2 c_7, \quad C_3 = c_4 c_6 + c_2 c_8,$$

in which

$$c_1 = a_{2}^* - a_{1}^* - a_{12} + a_{11}, \quad c_2 = a_{1}^* - a_{11},$$
$$c_3 = a_{22} - a_{21} - a_{2}^* + a_{1}^*, \quad c_4 = a_{21} - a_{1}^*,$$
$$c_5 = b_{12} - b_{11}, \quad c_6 = b_{11}, \quad c_7 = b_{22} - b_{21}, \quad c_8 = b_{21},$$
$$c_9 = a_{11} - a_{12} + a_{22} - a_{21}, \quad c_{10} = a_{21} - a_{11}.$$ 

The right slope has a symmetrical equation where the parameters indexed by 1 are replaced by those with 4, those indexed by 2 are replaced by those with 3, and the sign in $X$ is replaced by its opposite (negative direction tangents).
Proof. We prove it only for the infima. For simplicity, the symbols $c_1, \ldots, c_{10}$ will be used.

$$y_{\alpha_L} = \frac{\alpha c_5 + c_6}{\alpha(a_{22} - a_1^*) + a_1^* - \alpha(a_{12} - a_{11}) - a_{11}} + \frac{\alpha c_7 + c_8}{\alpha(a_{22} - a_{21}) + a_{21} - \alpha(a_{22} - a_1^*) - a_1^*}.$$

The common denominator in the numerator and the denominator of this expression will be the product of the denominators of the two fractions occurring in them:

$$D = D_1 D_2 = (\alpha c_1 + c_2)(\alpha c_3 + c_4),$$

and the sum of these two is

$$D_1 + D_2 = \alpha(c_1 + c_3) + (a_1^* - a_{11} + a_{21} - a_1^*) = \alpha c_9 + c_{10}.$$

From this,

$$y_{\alpha_L} = \frac{(\alpha c_5 + c_6)(\alpha c_3 + c_4) + (\alpha c_7 + c_8)(\alpha c_1 + c_2)}{\alpha c_9 + c_{10}}.$$

By simple rearrangement of this, the formula in the statement is straightforwardly obtained.

The result in the statement implies that piecewise linearity is generally not preserved in the conclusion. Let us examine now under which special restrictions on the shape of the terms in the rules the functions $y_{\alpha_L}$ and $y_{\alpha_U}$ become polynomial or even linear.

The condition of polynomiality is very simple: $c_9 = 0$. From that we obtain:

**Corollary 1.** The flanks of $B^*$ are piecewise polynomial (quadratic) if and only if the two antecedents $A_1$ and $A_2$ have equivalent piecewise linear slopes, obtainable from each other by geometric translations:

$$a_{12} - a_{11} = a_{22} - a_{21}.$$

If we require linearity of the pieces, one more condition needs to be satisfied: $C_1 = 0$. A consequence of it is the second corollary of the statement:

**Corollary 2.** The slopes of $B^*$ are piecewise linear either if both the antecedents $A_i$ and the consequents $B_i$ of the rules are equivalent pairwise and piecewise or if the antecedents and the observation $A^*$ are all equivalent piecewise.
Proof Although the above-mentioned two cases are interesting because they have very obvious meanings, we show how the general formula for linearity is obtained:

\[ C_1 = c_3c_5 + c_1c_7 \]

\[ = (a_{22} - a_{21} - a_2^* + a_1^*)(b_{12} - b_{11}) \]

\[ + (a_2^* - a_1^* - a_{12} + a_{11})(b_{22} - b_{21}) \]

\[ = 0. \]

But the slope is linear only if it is polynomial, i.e., by Corollary 1,

\[ a_{22} - a_{21} = a_{11} - a_{12} = d. \]

Let us denote similarly

\[ a_2^* - a_1^* = d^*. \]

Then we obtain

\[ (d - d^*)(b_{12} - b_{11}) + (d^* - d)(b_{22} - b_{21}) \]

\[ = (d - d^*)(b_{12} - b_{11} - b_{22} + b_{21} = 0. \]

Similar conditions apply for the right flanks as well.

There are infinitely many different combinations for the parameters \( b_{ij} \) where this equation is satisfied, but two cases deserve particular interest:

(1) \( d = d^* \) (then the \( b_{ij} \)'s are arbitrary).
(2) \( b_{12} - b_{11} = b_{22} - b_{21} \) (\( d \) and \( d^* \) are indifferent).

In the first case, there is no restriction on the shape of the observation (e.g., it might be a singleton value, as in many industrial applications), and in the second case there is no restriction on the consequents. Although the conditions of preserving piecewise linearity are rather strict, they are satisfied in many practical cases, as e.g. when the state variables are covered by "equidistant" terms.

Despite the importance of the special cases satisfying the conditions of Corollary 2, the result obtained is somewhat disappointing, as it indicates that in the general case interpolation only for the support and the core (\( \alpha = 0, 1 \) in the general case for the breakpoint set \( \alpha \in \Lambda \)) may not be satisfactory. If the nonlinearity of the rational function obtained for the general case is strong, it will be necessary to calculate for a much larger number of \( \alpha \)'s, and this increases the computational time.
7. NUMERICAL EXAMPLES FOR NONLINEAR $B^*$

In order to obtain a feeling for the degree of nonlinearity in the general case, let us discuss now some numerical examples, where the conditions of the corollaries are not satisfied.

EXAMPLE 1. Let the input and output universes be $X = [0, 7]$ and $Y = [0, 5]$. The rules will be discussed only from the point of view of the left flanks. The parameters $(a_{i1}, a_{i2}), \ldots$ describing these rules for $\alpha = 0, 1$ are $A_1: (0, 1); A_2: (5, 7); B_1: (0, 2); B_2: (4, 5); A^*: (2, 3)$ (see Figure 10). Clearly, in this example neither condition is satisfied, as

$$7 - 5 \neq 1 - 0$$

(i.e., the slope of $B^*$ cannot even be polynomial); moreover

$$5 - 4 \neq 2 - 0$$

(the consequents are also not uniform).

![Figure 10. Example for interpolation with trapezoidal terms.](image-url)
By substituting these values, we get

\[
\begin{align*}
    c_1 &= 0, & c_2 &= 2, & c_3 &= 1, & c_4 &= 3, & c_5 &= 2, & c_6 &= 0, \\
    c_7 &= 1, & c_8 &= 4, & c_9 &= 1, & c_{10} &= 5; \\
    C_1 &= 2, & C_2 &= 8, & C_3 &= 8,
\end{align*}
\]

from which the equation obtained for the left slope of $B^*$ is

\[
y_{aL}^* = \frac{2\alpha^2 + 8\alpha + 8}{\alpha + 5}.
\]

It is not surprising that this function is neither linear nor even quadratic. If interpolation is done for the two breakpoint levels 0 and 1, and linearity is supposed, the approximation of this equation will be

\[
y_{aL}^* = 1.4\alpha + 1.6.
\]

Obviously,

\[
\begin{align*}
    y_{aL}^*(0) &= y_{aL}^*(0) = 1.6, \\
    y_{aL}^*(1) &= y_{aL}^*(1) = 3.0,
\end{align*}
\]

as at these two points the exact values of $B^*$ have been calculated.

The following table shows the substitution values for every step of 0.1 in $\alpha$:

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>$y_{aL}^*$</th>
<th>$Y_{aL}^*$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.60</td>
<td>1.608</td>
</tr>
<tr>
<td>0.1</td>
<td>1.73</td>
<td>1.74</td>
</tr>
<tr>
<td>0.2</td>
<td>1.86</td>
<td>1.88</td>
</tr>
<tr>
<td>0.3</td>
<td>2.00</td>
<td>2.02</td>
</tr>
<tr>
<td>0.4</td>
<td>2.13</td>
<td>2.16</td>
</tr>
<tr>
<td>0.5</td>
<td>2.27</td>
<td>2.30</td>
</tr>
<tr>
<td>0.6</td>
<td>2.41</td>
<td>2.44</td>
</tr>
<tr>
<td>0.7</td>
<td>2.56</td>
<td>2.58</td>
</tr>
<tr>
<td>0.8</td>
<td>2.70</td>
<td>2.72</td>
</tr>
<tr>
<td>0.9</td>
<td>2.85</td>
<td>2.86</td>
</tr>
<tr>
<td>1.0</td>
<td>3.00</td>
<td>3.00</td>
</tr>
</tbody>
</table>

The curve of $y_{aL}^*$ is depicted in Figure 11.

The results show that the obtained rational function of $y_{aL}^*$ is astonishingly close to linear in the interval $\alpha = [0, 1]$: the largest deviation is 0.03, i.e. less than 1.5% of the exact value.

**EXAMPLE 2.** Let the input and output universes this time be $X = [0, 100]$ and $Y = [0, 11]$. Again we discuss only the left flanks. The parameters
(a_{i1}, a_{i2}), \ldots$ describing the rules for $\alpha = 0, 1$ are $A_1: (0, 1)$; $A_2: (10, 100)$; $B_1: (0, 10)$; $B_2: (10, 11)$; $A^*: (1, 10)$ (see Figure 12). In this example, both the antecedents and the consequents are very different:

$1 - 0 \ll 100 - 10$

and

$10 - 0 \gg 11 - 10$.

(The differences are one order of magnitude different, and in opposite directions.)

We obtain

$c_1 = 8, \quad c_2 = 1, \quad c_3 = 81, \quad c_4 = 9, \quad c_5 = 10, \quad c_6 = 0,$

$c_7 = 1, \quad c_8 = 10, \quad c_9 = 89, \quad c_{10} = 10;$

$C_1 = 818, \quad C_2 = 171, \quad C_3 = 10,$

and the equation obtained for the left slope of $B^*$ is

$y^*_{\alpha L} = \frac{818 \alpha^2 + 171 \alpha + 10}{89 \alpha + 10}$.

This result is even farther from being linear or quadratic. If we interpolate only for the two breakpoint levels 0 and 1, and we suppose linearity, the approximation of it will be

$y^*_{\alpha L} = 9.09 \alpha + 1.00$.

Obviously the end points fit here, too:

$y^*_{\alpha L}(0) = y^*_{\alpha L}(0) = 1.00,$

$y^*_{\alpha L}(1) = y^*_{\alpha L}(1) = 10.09,$

as these are the exact values of $B^*$.
The following table shows the substitution values for every step of 0.1 in $\alpha$:

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>$y_{\alpha L}^*$</th>
<th>$Y_{\alpha L}^*$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>0.1</td>
<td>1.91</td>
<td>1.87</td>
</tr>
<tr>
<td>0.2</td>
<td>2.82</td>
<td>2.77</td>
</tr>
<tr>
<td>0.3</td>
<td>3.73</td>
<td>3.68</td>
</tr>
<tr>
<td>0.4</td>
<td>4.64</td>
<td>4.59</td>
</tr>
<tr>
<td>0.5</td>
<td>5.55</td>
<td>5.50</td>
</tr>
<tr>
<td>0.6</td>
<td>6.45</td>
<td>6.42</td>
</tr>
<tr>
<td>0.7</td>
<td>7.36</td>
<td>7.34</td>
</tr>
<tr>
<td>0.8</td>
<td>8.27</td>
<td>8.25</td>
</tr>
<tr>
<td>0.9</td>
<td>9.18</td>
<td>9.17</td>
</tr>
<tr>
<td>1.0</td>
<td>10.09</td>
<td>10.09</td>
</tr>
</tbody>
</table>

The graph of $y_{\alpha L}^*$ is depicted in Figure 13.

Even with the very different antecedents, etc., the largest deviation is 0.05, less than 2% of the exact value. Some more numerical examples show
similar results. The empirics seems to be more promising than the analytic results concerning the approximate linearity of the slopes of $B^*$. 

8. SOME CONSIDERATIONS ON THE QUESTION WHY $B^*$ IS APPROXIMATELY LINEAR

The results obtained by many numerical examples seem to be in contradiction with the analytical results indicating the clear nonlinearity of $B^*$ in the general case. In the following, we attempt to examine the value of the maximal error if linear slopes are considered, and to explain why the numerical results point towards the good approximation of linearity.

The two end points of the left flank of $B^*$ are

$$y_{0L} = \frac{c_3}{c_{10}} = \frac{(a_{21} - a^*_{11})b_{11} + (a^*_{1} - a_{11})b_{21}}{a_{21} - a_{11}}$$

and

$$y_{1L} = \frac{C_1 + C_2 + C_3}{c_9 + c_{10}} = \frac{(a_{22} - a^*_{2})b_{12} + (a^*_{2} - a_{12})b_{22}}{a_{22} - a_{12}}.$$

From this, the equation of the linear approximation of the left flank is

$$y'_{aL} = \alpha (y_{1L} - y_{0L}) + y_{0L} = \left( \frac{C_1 + C_2 + C_3}{c_{9} + c_{10}} - \frac{C_3}{c_{10}} \right) \alpha + \frac{C_3}{c_{10}}.$$
From this formula the difference of the approximated (linear) slope and the real (rational function) one can be expressed:

**STATEMENT 2.** The error of approximating the nonlinear slope of the calculated conclusion by a linear slope between 0 and 1 expressed in terms of \(\alpha\), the membership degree running through \([0, 1]\) is

\[
\Delta y_{aL} = \frac{C_1}{c_9 \alpha + c_{10}} \alpha^2 + \left( \frac{C_2}{c_9 \alpha + c_{10}} + \frac{C_3}{c_{10}} - \frac{C_1 + C_2 + C_3}{c_9 + c_{10}} \right) \alpha \\
+ \left( \frac{C_3}{c_9 \alpha + c_{10}} - \frac{C_3}{c_{10}} \right).
\]

A similar expression can be given for the right side approximation.

**Proof** Straightforward by rearranging:

\[
\Delta y_{aL} = \frac{C_1 \alpha^2 + C_2 \alpha + C_3}{c_9 \alpha + c_{10}} - \left( \frac{C_1 + C_2 + C_3}{c_9 + c_{10}} - \frac{C_3}{c_{10}} \right) \alpha + \frac{C_3}{c_{10}}.
\]

The obtained result is not a very convenient expression for further evaluation. Nevertheless, an upper bound of the error was estimated and the following result was obtained:

**COROLLARY 3.** The error of replacing the real nonlinear flank by a linear approximation in the case of well-separated rules and observation is bounded by

\[
\Delta y_{aL} < \frac{E}{e} (E + 2b_{11} + 3F) - \frac{e}{E} (b_{11} + f) = L,
\]

where

\[
E = \max_{i=1}^5 e_i, \quad e = \min_{i=1}^3 e_i; \quad F = \max_{i=4}^6 e_i, \quad f = \min_{i=4}^6 e_i
\]

and

\[
e_1 = a_{12} - a_{11}, \quad e_2 = a_{11}^* - a_{12}, \quad e_3 = a_{21}^* - a_{11}^*, \\
e_4 = a_{21} - a_{21}^*, \quad e_5 = a_{22} - a_{21}, \quad e_6 = b_{12} - b_{11}, \\
e_7 = b_{21} - b_{12}, \quad e_8 = b_{22} - b_{23}.
\]

**Proof** \(c_{10}\) is always positive. The extrema for \(c_9 \alpha - c_{10}\) are

\[c_{10} = e_1 + e_2 + e_3 + e_4 > 0\]
and
\[ c_9 + c_{10} = e_4 - e_1 + e_2 + e_3 + e_4 > 0, \]
and so both \( c_9 + c_{10} \) and \( c_9 + c_{10} \) are always positive. Further,
\[
\begin{align*}
\max\{c_{10}\} &= 4E, & \min\{c_{10}\} &= 4e, \\
\max\{c_9 + c_{10}\} &= 4E, & \min\{c_9 + c_{10}\} &= 4e, \\
\max\{c_9 + c_{10}\} &= 4E, & \min\{c_9 + c_{10}\} &= 4e.
\end{align*}
\]
Similarly (using the definitions of \( C_i \)),
\[
\begin{align*}
\max\{C_1\} &= 2E^2, & \min\{C_1\} &= -2E^2, \\
\max\{C_2\} &= 2b_{11}E + 6EF, & \min\{C_2\} &= -2b_{11}E - 2EF, \\
\max\{C_3\} &= 4b_{11}E + 4EF, & \min\{C_3\} &= 4b_{11}e + 4ef.
\end{align*}
\]
From these 12 bounds the following is obtained:
\[
\Delta y_{\alpha L} < \frac{2E^2}{4e} + \frac{2b_{11}E + 6EF}{4e} - \frac{-2E^2 - 2b_{11}E - 2EF}{4e} + \frac{4b_{11}E + 4EF}{4e} - \frac{4be + 4ef}{4E},
\]
from which the statement of the Corollary is straightforward.

It must be stressed that Corollary 3 is valid only if the fuzzy sets in the rules are well separated according to the definition in Section 1.

Unfortunately, the expression obtained by using upper and lower bounds everywhere is rather rough. If \( R = E/e \) is introduced, it can be written as
\[
\Delta y_{\alpha L} < R(E + 2b_{11} + 3F) - \frac{b_{11} + f}{R},
\]
and clearly, if the antecedents are located equidistantly, all \( e_i \) in \( X \) and in \( Y \) are equal; consequently \( E = e, f = F, \) and \( R = 1, \) and so
\[ L = E + b_{11} + 2F \]
is obtained, although it is clear from Corollary 2 that in this case \( \Delta y_{\alpha L} = 0. \) Even in normed universes, this lower bound of the upper bound \( L \) is too high. Unfortunately, we could not find a better upper bound by using similar algebraic methods of estimation.
Now the function of \( y_{aL}^* \) will be examined from another point of view. Let us analyze the function of \( y_{aL}^* \) qualitatively. By completing the polynomial division, the function can be rewritten in the form

\[
y_{aL}^* = \frac{C_3 c_9^2 - C_2 c_9 c_{10} + C_1 c_{10}^2}{c_9^3 (c_9 \alpha + c_{10})} + \left( \frac{C_1}{c_9} \alpha + \frac{C_2 c_9 - C_1 c_{10}}{c_9^2} \right)
\]

where

\[
A = \frac{C_3 c_9^2 - C_2 c_9 c_{10} + C_1 c_{10}^2}{c_9^3}, \quad B = \frac{c_{10}}{c_9}, \quad C = \frac{C_1}{c_9}, \quad D = \frac{C_2 c_9 - C_1 c_{10}}{c_9^2}.
\]

This form shows clearly that the curve of \( y_{aL}^* \) is the superposition of a straight line \( y_L^* \) (for which, by the way, \( C = C_1/c_9 > 0 \) always holds) and a hyperbola \( y_H^* \). This latter has a horizontal asymptote at \( \alpha = -B = c_{10}/c_9 \) and a vertical one at \( y = 0 \). The combination has the same horizontal asymptote and another one at the linear component of the function. The graphs of the two components, and the combination can be seen in Figure 14. The left slope of \( B^* \) is given by the section of the curve from \( \alpha = 0 \) to \( \alpha = 1 \). (In the figure, the negative half of the hyperbola and the corresponding resulting curve are not depicted, as they never play a role in the slope of \( B^* \).) If \( \alpha \to \pm \infty \), the curve converges to its asymptote, i.e., for large \( \alpha \)’s it is approximately linear. The larger \( B \) is, the farther \( \alpha = 0 \) is from the neighborhood of the focus, i.e. the area where the curve is very nonlinear. So one of the qualitative statements we can obtain is:

**Fuzzy Statement.** For large positive values of \( c_{10}/c_9 \), i.e. where the ratio

\[
\frac{a_{21} - a_{11}}{a_{11} - a_{12} + a_{22} - a_{21}} = \frac{1}{\frac{a_{22} - a_{12}}{a_{21} - a_{11}} - 1}
\]

is high, the slope of \( B^* \) is rather close to linear.

The problem is that \( A \) also influences the behavior of the hyperbola and \( A \) is not independent of \( B \).

It is worthwhile mentioning why we prefer to express \( y_{aL}^* \) in terms of \( \alpha \) rather than the other way around. From the graph it is clear that \( \alpha \) is not
a single-valued function: for almost $\alpha$'s it has two values. The explicitly expressed form is obtained from the equation quadratic in $\alpha$, a rearrangement of the expression

$$C_1 \alpha^2 + (C_2 - c_9 y_{aL}^*) \alpha + (C_3 - c_{10} y_{aL}^*) = 0$$

in Statement 1 for $y_{aL}^*$ as

$$\alpha = \frac{(c_9 y_{aL}^*)^2 \pm \sqrt{(C_2 - c_9 y_{aL}^*)^2 - 4C_1(C_3 - c_{10} y_{aL}^*)}}{2C_1}. $$

For $\alpha \in [0, 1]$, the positive half is valid, so

$$\alpha = \frac{(c_9 y_{aL}^*)^2 + \sqrt{(C_2 - c_9 y_{aL}^*)^2 - 4C_1(C_3 - c_{10} y_{aL}^*)}}{2C_1}. $$

This formula is much more difficult to handle than its inverse.
It might be interesting to investigate the curve's behavior with respect to its asymptote \((C\alpha + D)\). Another style of upper bound for the linearity error can be given by calculating its difference from its asymptote, a straight line itself. As the hyperbola is monotonically decreasing with increasing \(\alpha\), it is sufficient to check whether the curve's distance from the asymptote at \(\alpha = 0\) is less than a given \(\epsilon\); for all other \(\alpha \in [0, 1]\) it will be even smaller. The result can be formulated in

**STATEMENT 3a.** The deviation of \(y^*_{al}\) from its linear asymptote \(C\alpha + D\) does not exceed \(\epsilon > 0\) if

\[
\frac{A}{\epsilon} \leq B.
\]

Proof By rearranging

\[
\frac{A}{\alpha + B} \leq \epsilon
\]

and substituting \(\alpha = 0\).

The statement can be expressed in terms of the \(c_i\) only:

\[
(c_4c_6 + c_2c_8)c_9^2 + (c_3c_5 + c_1c_7)c_{10}^2 \\
\leq (c_3c_6 + c_4c_5 + c_1c_8 + c_2c_7)c_9c_{10} + c_{10}c_5^2\epsilon.
\]

Unfortunately, this form does not offer any serious advantage in checking the condition faster, and the bound is also not very sharp. A somewhat better estimation of the linearity error takes both end points of the \(B^*\) segment into consideration, and uses the fact that the hyperbola is monotonically decreasing and so in the interval \([0, 1]\) its distance from the straight line passing through \(y^*_{al}\) and parallel with the asymptote will be everywhere within

\[
y^*_H(0) - y^*_H(1) = \frac{A}{B(1 + B)}.
\]

From this, a somewhat sharper version of the previous statement follows:

**STATEMENT 3.** The linearity error of \(y^*_{al}\) does not exceed \(\epsilon > 0\) if

\[
\frac{(C_2 + c_{10}\epsilon) + \sqrt{(C_2 + c_{10}\epsilon)^2 - 4C_1(C_3 - c_{10}\epsilon)}}{2(C_3 - c_{10}\epsilon)} \\
\leq \frac{c_0}{c_{10}} = \frac{1}{B}
\]

\[
\leq \frac{(C_2 + c_{10}\epsilon) - \sqrt{(C_2 + c_{10}\epsilon)^2 - 4C_1(C_3 - c_{10}\epsilon)}}{2(C_3 - c_{10}\epsilon)}.
\]
Proof Substituting $C_i$ and rearranging the equation for $y^*_H(0) - y^*_H(1)$, we obtain

$$\frac{C_3 c_9^2 - C_2 c_9 c_{10} + C_1 c_{10}^2}{c_9^3 c_{10}} \times \frac{c_9 + c_{10}}{c_9} \leq \epsilon,$$

$$(C_3 - c_{10} \epsilon)c_9^2 - (C_2 c_{10} + c_{10}^2 \epsilon)c_9 + C_1 c_{10}^2 \leq 0.$$

If $\epsilon$ is small, $C_3 - c_{10} \epsilon$ is always positive, and so the inequality is satisfied in the interval between the two solutions of the corresponding quadratic equation. Getting these solutions and dividing by $c_{10}$, the statement is obtained. ⊓⊔

It seems reasonable to express the condition for $c_9 = (a_{22} - a_{21}) - (a_{12} - a_{11})$ in a form relative to $c_{10} = a_{21} - a_{11}$, as in this form it is much clearer that the difference of the degree of fuzziness between the two antecedents (measured in units of their distance from each other) must be limited. The result in Corollary 2 intuitively coincides with this fact.

It might be interesting to calculate this bound in the case of the two numerical examples treated earlier in this report. In the case of the first one we had

$$c_9 = 1, \quad c_{10} = 5; \quad C_1 = 2, \quad C_2 = 8, \quad C_3 = 8,$$

and so the estimated error (upper bound) is

$$E = \frac{A}{B(1 + B)} = \frac{C_3 c_9^2 - C_2 c_9 c_{10} + C_1 c_{10}^2}{c_9^3 c_{10}} \times \frac{c_9 + c_{10}}{c_9} = \frac{C_3 c_9^2 - C_2 c_9 c_{10} + C_1 c_{10}^2}{c_9 c_{10}(c_9 + c_{10})} = \frac{18}{30} = 0.6.$$

This estimate for the error is considerably (approximately 20 times) higher than the real maximal deviation from linearity.

Next we calculate the estimate for the second numerical example:

$$c_9 = 89, \quad c_{10} = 10; \quad C_1 = 818, \quad C_2 = 171, \quad C_3 = 10;$$

$$E = \frac{C_3 c_9^2 - C_2 c_9 c_{10} + C_1 c_{10}^2}{c_9 c_{10}(c_9 + c_{10})} = \frac{8820}{88110} \approx 0.1.$$
This result is considerably better; it is only about twice the size of the real error.

Although the estimates obtained in Statements 2 and 3 are not very sharp, they still give a certain insight into the reason for having usually very close to linear slopes of $B^*$, a fact that helps to reduce enormously the computational complexity of the interpolation based reasoning. Summarizing our results, we can state that in some special but practically important cases the slopes of the calculated conclusion are exactly linear, while in the general case they are hyperbolic. However, in most cases they are rather close to linear, so that in the case of piecewise linear rules and observation it is sufficient to calculate only the breakpoints of $B^*$—i.e., in the trapezoidal or triangular case, for $\alpha = \{0, 1\}$, where $\lambda = 2$ is obtained.

There is an important conclusion to the considerations in the previous sections. Even though the inherent exponentiality of the problem cannot be eliminated, by using sparse rule bases the number of terms per variable can be reduced. Let us suppose that $T$ is reduced by a factor $s > 1$, so in every dimension, the maximal number of terms is $T/s$. Consequently, the number of rules in the rule base (with $k$ inputs) is $\mathcal{O}((T/s)^k)$, and the time complexity of the interpolative algorithm is

$$\mathcal{E}_i = \mathcal{O}\left(\lambda(k + 1)\left(\frac{T}{s}\right)^{k+1}\right).$$

However, as $|\lambda|$ was shown to be in every practical case $\leq 4$, this is identical with

$$\mathcal{E}_i = \mathcal{O}\left((k + 1)\left(\frac{T}{s}\right)^{k+1}\right).$$

This is nevertheless exponential, and the exponent is identical with that of the Mamdani (or Larsen) algorithm. However, the gain of using interpolation instead of the Mamdani method is expressed by the following ratio:

$$\frac{\mathcal{E}_{ML}}{\mathcal{E}_i} = \mathcal{O}(s^{k+1}),$$

which might amount to a considerable increase in the control speed.

9. ON THE PROBLEM OF THE CNF PROPERTY OF $B^*$

As was mentioned, the validity of the linear interpolation method is restricted to CNF sets. It is interesting to check whether it preserves convexity and normality, i.e. whether $B^*$ itself is always a CNF set and so is
suitable for further interpolation in a multiple level reasoning algorithm, where $B^*$ might be the "observation" for the next level in a scheme

\[ A_1 \rightarrow B_1, \quad A_2 \rightarrow B_2, \quad A^*, \]
\[ B'_1 \rightarrow C_1, \quad B'_2 \rightarrow C_2 \]

such that

\[ A_1 < A^* < A_2 \quad \text{and} \quad B'_1 < B_1 < B_2 < B'_2 \]

If all $A_i$ etc. are CNF and $B^*$ is CNF too, then these conditions are sufficient for the existence of $C^*$, the second level conclusion calculated by two interpolations. Clearly, the second part is not a necessary condition, as in any case

\[ B'_1 < B^* < B'_2 \]

is enough for the executability of the second step calculations.

Let us check first the convexity. $B^*$ is convex if all its $\alpha$-cuts are connected. The interpolation method applied never produces other than connected cuts, as they are expressly defined as intervals, by their minimum and maximum. So convexity is automatically satisfied.

It is much more complicated to describe the situation from the point of view of normality. $B^*$ is normal if the membership function of $B^*$ assumes all values in $[0, 1]$. Obviously, the method of generating $B^*$ is such that the two points for the intervals should be always ordered (in the sense of $<_\prec$) in the proper way, i.e.,

\[ \forall \alpha : b^*_2 = \inf\{B^*_\alpha\} \leq \sup\{B^*_\alpha\} = b^*_3. \]

If the above condition is not fulfilled, the "membership function" wraps around itself (forms a loop). For any $\alpha$ where this condition is not satisfied, a real membership function does not exist, i.e.,

\[ \text{height}(B^*) = \max_{[0,1]} \{\alpha : \inf\{B^*_\alpha\} \leq \sup\{B^*_\alpha\}\}. \]

Using the equations and notation of the previous sections, we can formulate the condition of normality:

**STATEMENT 4.** With the notation of Statement 1, $B^*$ is normal if and only if

\[ b^*_2 = \frac{(a_{22} - a^*_2)b_{12} + (a^*_2 - a_{12})b_{22}}{a_{22} - a_{12}} \leq b^*_3 \]
\[ = \frac{(a_{23} - a^*_3)b_{13} + (a^*_3 - a_{13})b_{23}}{a_{23} - a_{13}}, \]
that is,

\[
\frac{(a^*_2 - a_{12})(b_{22} - b_{12})}{a_{22} - a_{12}} \leq (b_{13} - b_{12}) + \frac{(a^*_3 - a_{13})(b_{23} - b_{13})}{a_{23} - a_{13}}.
\]

Proof  Let us substitute \( \alpha = 1 \) into the expressions for \( b^*_2 \) and \( b^*_3 \) (the latter is obtained from the former in a symmetrical way by replacing the second subscripts 2 with 3 everywhere, and changing the sign). So we obtain

\[
b^*_2 = y^*_{\alpha L} (\alpha = 1) = \frac{C_1 + C_2 + C_3}{c_9 + c_{10}}
\]

\[
= \frac{c_3 c_5 + c_1 c_7 + c_3 c_6 + c_4 c_5 + c_1 c_8 + c_2 c_7 + c_4 c_6 + c_2 c_8}{c_9 + c_{10}},
\]

and from here the left hand side of the first formula in the statement is obtained. Similarly, the expression for the maximum of the core is

\[
y^*_{\alpha U} (\alpha = 1) = \frac{(a_{23} - a^*_3)b_{13} + (a^*_3 - a_{13})b_{23}}{a_{23} - a_{13}},
\]

and so the first formula follows.

For a better explanation of the second formula, let us introduce some shorthand notation (see Figure 15):

\[
\Delta L_1 = a^*_2 - a_{12}, \quad \Delta L_2 = a_{22} - a^*_2, \quad \Delta U_1 = a^*_3 - a_{13},
\]

\[
\Delta U_2 = a_{23} - a^*_3, \quad \Delta b = b_{13} - b_{12}, \quad \Delta b_2 = b_{22} - b_{12},
\]

\[
\Delta b_3 = b_{23} - b_{13}, \quad \Delta B = b_{22} - b_{13}.
\]

With its help, we can rewrite

\[
b^*_2 \leq b^*_3
\]

as

\[
\frac{\Delta L_2 b_{12} + \Delta L_1 b_{22}}{\Delta L_1 + \Delta L_2} \leq \frac{\Delta U_2 b_{13} + \Delta U_1 b_{23}}{\Delta U_1 + \Delta U_2},
\]

and, since

\[
b_{22} = b_{12} + \Delta b_2, \quad b_{13} = b_{12} + \Delta b, \quad b_{23} = b_{13} + \Delta b_3,
\]
we obtain
\[
\frac{\Delta L_2 b_{12} + \Delta L_1 (b_{12} + \Delta b_2)}{\Delta L_1 + \Delta L_2} \leq \frac{\Delta U_2 b_{13} + \Delta U_1 (b_{13} + \Delta b_3)}{\Delta U_1 + \Delta U_2},
\]
\[
b_{12} + \frac{\Delta L_1 \Delta b_2}{\Delta L_1 + \Delta L_2} \leq b_{13} + \frac{\Delta U_1 \Delta b_3}{\Delta U_1 + \Delta U_2},
\]
\[
\frac{\Delta L_1 \Delta b_2}{\Delta L_1 + \Delta L_2} \leq \Delta b + \frac{\Delta U_1 \Delta b_3}{\Delta U_1 + \Delta U_2},
\]
which is the second formula with our shorthand.

This form makes it clear that the problem of obtaining subnormal \(B^*\) appears basically if both the relative lower distance of the cores of \(A^*\) and of \(A_1\) (compared to the distance of \(A_1\) and \(A_2\)) and the lower distance of the cores of \(B_1\) and \(B_2\) are high, while the upper distances of the same terms are low, i.e., the core of the observation is small compared to the core of \(A_1\). This problem is eliminated, however, if the core of \(B_1\) is high enough. Clearly, a similar formula referring to the core of \(B_2\) and
expressing the relative distances of $A^*$ from $A_2$ could be obtained, i.e., the problem can be formulated in a more general form: The calculated conclusion is normal if the core of the observation is large enough, or if the cores of the consequents are large enough. As often the observation is triangular, or even a singleton, (i.e., it has a zero length core), it is worthwhile examining the conditions in these cases. Before that, however, let us take an example to show that subnormality might very easily occur.

The universes are real intervals, $X = [\leq 0, \geq 11]$, $Y = [\leq 0, \geq 10]$. Suppose that the core points of the rules are

$$a_{12} = a_{13} = 1, \quad a_{22} = 9, \quad a_{23} = 11,$$
$$b_{12} = b_{13} = 1, \quad b_{22} = b_{23} = 10$$

and the observation is $a^*_1 = a^*_3 = 5$. [It is enough to examine the core; if it exists (i.e. $b^*_1 = b^*_3 \geq 0$), the result is normal.] The rules and the observation are triangular with the exception of $A_2$. Substituting the values into the condition, we find that

$$\Delta L_1 = 5 - 1 = 4, \quad \Delta L_2 = 9 - 5 = 4, \quad \Delta U_1 = 5 - 1 = 4,$$
$$\Delta U_2 = 11 - 5 = 6; \quad \Delta b = 1 - 1 = 0, \quad \Delta b_2 = 10 - 1 = 9,$$
$$\Delta b_3 = 10 - 1 = 9,$$

and so the following inequality should hold:

$$\frac{4 \times 9}{4 + 4} = \frac{9}{2} \leq 0 + \frac{4 \times 9}{4 + 6} = \frac{18}{5}.$$ 

This is not true, so $B^*$ is subnormal. The example can be seen in Figure 16; it is clear that the "membership function" is wrapped and its really existing part is subnormal. Let us find now the minimal length of $\Delta b$ for which the result would be normal. $\frac{9}{2} - \frac{18}{5} = \frac{9}{10}$, so if the core of $B_1$ is elongated by this value, and all other points $b_{ij}$ are transformed, too, so that the other differences do not change, then the result will become normal triangular. The new values for the consequents will be

$$b_{12} = 1, \quad b_{13} = \frac{19}{10}, \quad b_{22} = 10, \quad b_{23} = \frac{109}{10},$$

and so

$$\Delta b = \frac{19}{10}, \quad \Delta b_2 = 9, \quad \Delta b_3 = \frac{109}{10} - \frac{9}{10} = 9,$$

for which

$$\frac{4 \times 9}{4 + 4} = \frac{9}{2} \leq \frac{9}{10} + \frac{4 \times 9}{4 + 6} = \frac{9}{2}.$$
where equality holds. The so modified rules and conclusion can be seen in Figure 17.

From the example, the important role played by the core lengths is obvious. Let us examine now some special cases, first where all the cores are disjoint, i.e.,

\[ a_{13} < a^*_2, \ a^*_3 < a_{22}, \quad b_{13} < b_{22}. \]

Once more we introduce new notation, this time for all core lengths and the distances between the cores of the neighboring sets:

\[ K_{a1} = a_{13} - a_{12}, \quad K_{a2} = a_{23} - a_{22}, \quad K^*_a = a^*_3 - a^*_2; \]

\[ K_{b1} = b_{13} - b_{12}, \quad K_{b2} = b_{23} - b_{22}; \]

\[ d_{a1} = a^*_2 - a_{13}, \quad d_{a2} = a_{22} - a^*_3, \quad d_b = b_{22} - b_{13}. \]

where the last three are positive. In this case, the necessary and sufficient condition for the normality of \( B^* \) is

\[
\frac{(K_{a1} + d_{a1})(K_{b1} + d_b)}{K_{a1} + K^*_a + d_{a1} + d_{a2}} \leq K_{b1} + \frac{(K^*_a + d_{a1})(K_{b2} + d_b)}{K_{a2} + K^*_a + d_{a1} + d_{a2}}.
\]
From this we get

\[(K_{a1} + d_{a1})(K_{b1} + d_b)(K_{a2} + K_a^* + d_{a1} + d_{a2})\]

\[\leq K_{b1}(K_{a1} + K_a^* + d_{a1} + d_{a2})(K_{a2} + K_a^* + d_{a1} + d_{a2})\]

\[+ (K_a^* + d_{a1})(K_{b2} + d_b)(K_{a1} + K_a^* + d_{a1} + d_{a2}).\]

Executing all multiplications and canceling the identical members on both sides, this reduces to

\[d_bK_{a1}K_{a2} + d_{a1}d_bK_{a2} + d_{a2}d_bK_{a1}\]

\[\leq K_{a2}K_a^*K_{b1} + d_{a2}K_a^2K_b1 + K_a^*2K_{b1} + d_{a2}K_a^*K_{b1}\]

\[+ d_{a1}K_a^*K_{b1} + d_{a2}K_a^*K_{b1} + d_{a1}d_{a2}K_{b1} + d_{a2}^2K_{b1}\]

\[+ K_a^*K_{a1}K_{b2} + d_{a1}K_a^2K_{b2} + K_a^*2K_{b2} + d_{a1}K_a^*K_{b2}\]

\[+ d_bK_a^* + d_{a1}d_bK_a^* + d_{a1}K_a^*K_{b2} + d_{a2}^2K_{b2}\]

\[+ d_{a2}K_a^*K_{b2} + d_{a1}d_{a2}K_{b2} + d_{a2}d_bK_a^*.\]
From this, after some more rearrangements, we obtain the following:

\[ d_b[(K_{a1} + d_{a1})(K_{a2} + d_{a2}) - (K_a^* + d_{a1})(K_a^* + d_{a2})] \leq (K_{a1} + d_{a1})(K_{b2}d_{a1} + K_a^*K_{b2}) + (K_{a2} + d_{a2})(K_{b1}d_{a2} + K_a^*K_{b1}). \]

In this, \( K_a^* = 0 \) in the worst case. Substituting this value and returning to the original notation, we get a form that is the statement of

**Corollary 4.** For rules and observation with disjoint cores, \( B^* \) will be normal if and only if

\[ (b_{22} - b_{13})[(a_{22}^* - a_{12})(a_{23} - a_{32}^*) - (a_{22}^* - a_{12})(a_{22} - a_{32}^*)] \leq (a_{22} - a_{12})(a_{22}^* - a_{13})(b_{23} - b_{22}) + (a_{23} - a_{13})(a_{22} - a_{32}^*)(b_{13} - b_{12}). \]

The differences in this form have some plausible geometrical meaning.

Let us check some more special cases, first if all membership functions are triangular. As here \( K_{ai} = K_{bi} = K_a^* = 0 \), the former condition is considerably simplified to

\[ \frac{d_{a1}d_{b}}{d_{a1} + d_{a2}} \leq \frac{d_{a1}d_{b}}{d_{a1} + d_{a2}}, \]

which is obviously always true. So we have the next corollary:

**Corollary 5a.** If the rules and the observation contain only triangular membership functions, the conclusion is always normal.

We check now the trapezoidal case, where however the core width is uniform for the antecedents and (separately) also for the consequents (in general, every cut should have the same length), i.e., \( K_{ai} = K_a, K_{bi} = K_b \).

**Corollary 6.** If the corresponding cores in the rules have uniform length, the conclusion will be normal if and only if

\[ d_b(K_a - K_a^*) \leq K_b(d_{a1} + d_{a2} + 2K_a^*). \]

In the worst case \( K_a^* = 0 \) and the inequality can be replaced by an even stricter one:

\[ d_b(K_a - K_a^*) \leq K_b(d_{a1} + d_{a2} + K_a^*) = K_bd_a, \]
where $d_a = a_{22} - a_{13}$ is a similar notation in $X$ to $d_b$ in $Y$. This form can be formulated in a rather plausible way:

**Corollary 7.** In the case of uniform core length in both the antecedent and the consequent parts of the rules, the conclusion will always be normal if the ratio of the distances of the rule cores of the consequents and of the antecedents does not exceed the ratio of the core lengths themselves.

If the scale in both universes is locally normalized by the distance between the two rule cores (obtaining the normalized cores $k_a = K_a/d_a$ and $k_b = K_b/d_b$), the conclusion will always be normal if the consequents have a not shorter core, i.e., the consequents are not less fuzzy then the antecedents themselves. (The opposite is not true: if the consequents are less fuzzy, still a fuzzy enough observation might save the normality of the conclusion.) In the normalized scale we have

$$k_a \leq k_b + k^*$$

as a sufficient (but still not necessary) condition ($k^*_a = K^*_a/d_a$). It is enough for the observation to be at least as fuzzy as the antecedents in order to guarantee the normality of the conclusion.

**Corollary 5.** If $K_{ai} = K^*_a = K_a$ and $K_{bi} = K_b$, the conclusion is always normal.

(Corollary 5a for triangular membership functions is a special case of this.)

To conclude, it is mentioned that the subnormality obtained by the omission of the wrapped part of the membership function can be normalized in two different ways: either by taking the convex hull of the conclusion (so that every cut is contained in every lower $\alpha$-cut), or by expanding the subnormal set by the appropriate factor. Figure 16 helps to understand both solutions: either the upper loop should be simply omitted (and then the remaining triangular function be scaled appropriately), or the convex hull of the set should be taken instead of the wrapped one. If the trapezoidal shape is required, the convex hull can be replaced by a “trapezoidal hull” connecting the two support points with the wrapped core points. Normalization is necessary if further stages of (e.g. interpolative) reasoning follow.

The above analysis does not cover all possible problems that might occur with abnormal conclusions in rule interpolation. In some extreme cases a resulting “membership function” might be obtained where

$$\inf\{B^*_0\} > \inf\{B^*_1\}$$

or

$$\sup\{B^*_0\} < \sup\{B^*_1\}.$$
Further study of such abnormalities is going on; some initial results have been shown in [41, 42].

The statements in the analysis above refer only to the linear interpolation technique using infima and suprema for the construction of the conclusion. Other techniques like center and width interpolation [13, 35] avoid the problem automatically; however, some of the original information gets lost in that case.

10. STRUCTURED RULE BASES

Even though one is using sparse rule bases, in the case of many state variables the complexity of the rule base may be too high. Interpolation and sparse rule bases decrease the run time drastically only if a very low $r \Theta(2^k)$ fits the system quite well. Otherwise, only the reduction of $k$ might lead to low run times. Of course, $k$ really can be decreased if some of the variables turn out not to affect the output essentially. This case can be eliminated now, as we suppose that the effective number of variables cannot be reduced further.

A completely new aspect is offered if more radical changes in the structure of the rule base are considered. An essentially new way of dealing with this kind of problems was shown by M. Sugeno in the unmanned helicopter control application [5, 6].

From the point of view of the computational complexity reduction, the essence of the idea of structured rule bases is that even though a system needs a very complex description, and there are too many input variables affecting the behavior of the system in nonnegligible ways, the model might be dominated locally by only a subset of variables (this subset changing with the change of the state space area in question). The total state space must be subdivided into domains, and for every domain another “local model” (with possibly very restricted validity) can and must be constructed. In the lucky case, every local model contains a much smaller number of variables and consequently also a much smaller number of rules. The same partition of the space might be given naturally by the intended action to be done on the system as it is in the case of the helicopter control application (“hovering,” “forward flight,” etc.).

Every local model is determined by a sub-rule-base. On the higher level(s), called metalevel(s), it must first be decided which sub-rule-base is adequate for the neighborhood of the observation or the desired action of the system. In order to do so, a set of metarules is necessary, of which the output variable is the domain variable. The rules decide on either some input variables essentially separating the different submodels, or the special variables expressing the type of action taken by the system locally.
In the basic form of this method, the domains form a classical (crisp) partition of the state space, so for every situation exactly one domain is adequate; consequently, always exactly one sub-rule-base must and can be used. However, the principle can be extended farther. The partition can be replaced by some more general cover of the state space, this allowing more flexibility in the determination and control of the validity domains for the individual sub-rule-bases. Even the cover might be fuzzy, so that its elements fade as the observation gets farther from the typical validity area (core) of that particular sub-rule-base. This means however that in the general case, usually several among the (presumably numerous) available sub-rule-bases must be combined, most of them playing probably a less important role, which fact is indicated by their lower membership degree in the calculation of the conclusion.

It is possible to involve several levels in the model, such that all levels except the lowest one are metalevels refining the actual domain more and more. On the lowest level the sub-rule-bases for the actual output variable(s) are always to be found.

A very simple example for a structured rule base with a single metalevel is the following:

(1) Metalevel \( (R_0) \):

"If \( x_1 \) is \( A_{11} \) and \( x_2 \) is \( A_{21} \) then take domain \( D_1 \)."

"If \( x_1 \) is \( A_{12} \) and \( x_2 \) is \( A_{22} \) then take domain \( D_2 \)."

(2) Sub-rule-base \( R_1 \) for \( D_1 \):

"If \( x_3 \) is \( A_{31} \) and \( x_4 \) is \( A_{41} \) then \( Y \) is \( B_1 \)."

"If \( x_3 \) is \( A_{32} \) and \( x_4 \) is \( A_{42} \) then \( Y \) is \( B_2 \)."

(3) Sub-rule-base \( R_2 \) for \( D_2 \):

"If \( x_5 \) is \( A_{51} \) then \( Y \) is \( B_3 \)."

"If \( x_5 \) is \( A_{52} \) then \( Y \) is \( B_4 \)."

Here, instead of \( \Theta(T^5) \) rules only \( \Theta(T^2) \) are necessary; if \( T \) is e.g. 7, than instead of \( \approx 16,807 \) only \( < 3 \times 49 = 147 \) are needed. In general, if the \( k \) input variables can be divided into \( n \) groups, the first one \( (k_1) \) being the first metalevel variables, the second one \( (k_2) \) the second metalevel, etc., and the last one \( (k_n) \) the control level inputs, in the worst case the total number of rules in the base is

\[
r = \Theta(T^{k_1} + T^{k_1}T^{k_2} + \cdots + T^{k_1}T^{k_2} \cdots T^{k_n}) = \Theta(T^k).
\]
Even so, for any concrete observation, only
\[ r_{A^*} = O(T^{k_1} + T^{k_2} + \cdots + T^{k_n}) = O(nT^{\max\sum k_i}) \]
rules must be processed, which is an effective decrease in the run time. Moreover, if the individual sub-rule-bases contain only a reduced number of variables, the role of \( \max_{i=1}^n k_i \) is taken over by the largest number of variables occurring in the same sub-rule-base.

Of course, in real applications, the number of variables and the size of the rule base might be much bigger than in the small example above.

11. EXTENDED HIERARCHICAL CONTROL AND INTERPOLATION

If a fuzzy cover is taken instead of a crisp partition, a new problem emerges: How to combine sub-rule-bases with different subsets of variables? There might be even more complicated situations. What if the available information does not cover the whole state space, except in a sparse way? The problem of sparse rule bases might appear on the metalevel, too. To present a general algorithm for all these various cases, the idea of using structured rule bases can be developed further by combining it with the principle of interpolation—this time on the metalevel. Then it will be possible to consider sparse partitions of the state space, i.e. to allow gaps between the (possibly fuzzy) domains of validity for the individual sub-rule-bases. In such a model, in a certain situation the observation does not always identify the actual domain unambiguously: when it falls into a gap (between two or more domains), several "neighboring" (flanking) domains must be taken into consideration. By making the model even more general, overlapping of the domains and contradicting metarules can be also allowed, so in a general situation no sub-rule-base at all or even several (partially) contradicting sub-rule-bases might be "fired." if also fuzzy domains are allowed, the membership degree to which a certain sub-rule-base is taken into consideration might weight the importance of the sub-rule-bases in the calculation of the conclusion, leading to different and possibly (partially) contradicting sub-rule-bases, referring to different subsets of variables weighted by different degrees (determined either by the membership function of the corresponding fuzzy domain or by the degree of closeness determined by the fuzzy distances of the observation from the flanking fuzzy domains, calculated by the \( \inf \sup \) or the centr width pairs).

The idea of interpolating and approximating among sub-rule-bases is discussed in some more detail in [44]. It must be seen clearly, however, that this kind of general approach raises a large number of serious mathematical difficulties. The sub-rule-bases refer to different subsets of
the state variables, maybe partially overlapping, maybe completely disjoint. If for the interpolation of these sub-rule-bases they have to be united, it is necessary to find first the smallest containing subset of variables, and all rules in the rule bases must be transformed into the new, extended space. We suggested here the *cylindric extension* (i.e. to suppose that the rules as relations have a projection equal to 1 in every dimension where the rule has to be extended). It is still an open question if it is better to take the cylindric extensions of all sub-rule-bases rule by rule and so to form a big conglomerate sub-rule-base by the union of all flanking sub-rule-bases, or to calculate first a single representative rule for every sub-rule-base (by evaluating the relative position of the observation with that particular rule base), and then the union of these representative rules in the extended rule base.

The behavior of disjoint rule bases is quite different from that of bases with partially common variables. The mathematical problems are partially discussed in [44, 45] but need further study.

12. CONCLUSIONS ON THE ALGORITHMIC ASPECTS

The paper has discussed a variety of fuzzy rule based control algorithms. The classical method based on relations has a very high complexity. The simplified techniques occurring in almost all applications and working with the projections to the $X_i$ axes are somewhat better, although even here, with a somewhat larger number of variables, the problem becomes intractable.

When using sparse rule bases and interpolation algorithms, the complexity is further decreased; depending on the type of the system, it can be very low even with very many variables—if the system tends to be close to linear. Special treatment was given to the problem of the number of interpolation levels and, in general, to the shape of the interpolated conclusion, as this group of questions has not yet been treated in the literature in much detail.

Finally, hierarchical structured control was discussed briefly. It is the only algorithm so far that is able to reduce effectively the exponent in the expression for uniform steps, provided the system is decomposable and it is possible to reduce the number of variables locally.

13. FUZZY CONTROL AND APPROXIMATION

To conclude this paper it seems desirable to refer to another very important dual aspect of fuzzy control. While the examination from the
algorithmic point of view offers some answers to the problem “If the fuzzy model of the control system is given, how can this model be treated in the most effective, possibly approximative way so that essential information on the control system is not lost?” There is another question: “If the exact model of the control system is known, or at least, is supposed to exist theoretically, what are the limits of approximating this model via the use of fuzzy controllers?” In other words, this paper has treated the problem of approximation in fuzzy control systems, while there remains the problem of approximation by fuzzy control systems.

Although this second topic would exceed the goals set in this paper, which discusses only the algorithmic question, it might be interesting to have a brief look at the second question, especially as in the past few years more and more results have been published, and more discussion is going on concerning the universality of fuzzy controllers. The first results in this direction were published in 1991 and 1992, and several parallel results showed that fuzzy controllers with crisp singleton inputs, consisting of a degree of matching unit, the rule base, an inference engine, and a defuzzification unit, can be interpreted as crisp to crisp mappings, i.e., function generators that have the surprising property of being universal approximators under rather mild conditions (the function to be approximated should be continuous, and the domain of approximation should be compact) if certain properties of the rules are satisfied, and some version of the Mamdani algorithm (e.g. the Sugeno model) is used (see e.g. [46–49]). Even the interpolation algorithm proved to be universal and suitable to approximate the optimal control algorithm arbitrarily well, at least for known linear systems (cf. [50]). These facts seemed at first glance to be very strong arguments for the widespread application of fuzzy controllers. In addition to these “universal approximator theorems,” it was recently shown that with proper rules it is possible to generate any “reasonable” function exactly with the Sugeno controller [51], i.e., fuzzy controllers are even universal function generators.

From the point of view of real applications, the weak point of these results lies in the fact that the algorithmic aspect is left out of consideration. Some of these results use the Stone-Weierstrass theorem, and it is necessary that all antecedents in the rules have unbounded supports, but in every case, the number of rules in the rule base is unbounded. (With such conditions, even starting with sparse rule bases, the limit of the rule base as the approximation becomes better and better is a dense rule base, so the real advantage of interpolation is lost in the limit.) In the case of the universal function generator, the knowledge of the exact function is necessary for the determination of the membership functions in the rules, i.e., the advantage of reducing the complexity of calculating the substitution values of very complicated functions is lost.
If the algorithmic (especially complexity) preferences are taken into consideration, the universality disappears: with a bounded number of bounded support rules and simple membership functions for the terms, no universality can be achieved. The classes of functions that can be generated this way are far from being very suitable for arbitrarily good approximation, as can be observed when the actual explicit function classes are determined [52]. Some theoretical results supporting these observations will be published soon. However, it seems that the use of fuzzy controllers is much better motivated by the possibility of rough approximation with conveniently low complexity than by the theoretical possibility of fine approximation or exact function generation. An analysis of the problem of approximation by fuzzy controllers is given in [53].

References


