Production Systems and Information Engineering
Volume 2 (2004), pp. 53-71

INTERPOLATION-BASED FUZZY REASONING IN BEHAVIOUR-BASED CONTROL STRUCTURES

SZILVESZTER KOVÁCS*
Department of Information Technology, University of Miskolc,
Miskolc-Egyetemváros, Miskolc, H-3515, Hungary
szkovacs@iit.uni-miskolc.hu

*Intelligent Integrated Systems Japanese Hungarian Laboratory,
Budapest University of Technology and Economics, Hungary

[Received November 2004 and accepted January 2005]

Abstract. Some difficulties emerging during the construction of fuzzy behaviour-based control structures are inherited from the type of the applied fuzzy reasoning. Classical fuzzy reasoning methods need a complete fuzzy rule base. In case of fetching fuzzy rules directly from expert knowledge e.g. for the behaviour coordination module, the way of building a complete rule base is not always straightforward. One simple solution for overcoming the necessity of the complete rule base is the application of interpolation-based fuzzy reasoning methods, since interpolation-based fuzzy reasoning methods can serve usable (interpolated) conclusion even if none of the existing rules is hit by the observation. These methods can save the expert from dealing with derivable rules and help to concentrate on cardinal actions only. For demonstrating the applicability of the interpolation-based fuzzy reasoning methods in behaviour-based control structures a simple interpolation-based fuzzy reasoning method and its adaptation for behaviour-based control will be discussed briefly in this paper.

Keywords: Interpolation-based Fuzzy reasoning, Behaviour-based Control

1. INTRODUCTION

In behaviour-based control systems (a good overview can be found in [3]), the actual behaviour of the system is formed as one of the existing behaviours (which fits best the actual situation), or as a kind of fusion of the known behaviours appeared to be the most appropriate to handle the actual situation. Beyond the construction of the behaviours, this structure has two other important tasks. The first is the decision, which behaviour is needed, or in case of behaviour fusion the determination of the necessity levels for each behaviour in solving the actual situation. The second is the way of the behaviour fusion. The first task, the behaviour coordination can be viewed as an actual system state approximation, where the actual system state is the set of the necessities of the known behaviours needed for handling the actual situation. The second is the fusion of the known behaviours based on their necessities. In case of fuzzy behaviour based control structures both tasks are solved by fuzzy logic controllers. If the behaviours are also implemented on direct fuzzy logic controllers, the behaviours together with the behaviour fusion modules form a hierarchical fuzzy logic controller. Since the classical fuzzy reasoning methods (e.g. compositional rule of inference) demand complete rule base, all these rule bases have to be built taking care of filling all the possible rules. In case if there is some missing rule, there are observations may exist which hit no rule in the rule base and therefore no conclusion is obtained. Having no conclusion at any level of the fuzzy behaviour based control structure is hard to explain. E.g. one solution could be to keep the last real conclusion instead of the missing one, but applying historical data automatically to fill undeliberately missing rules could cause unpredictable side effects. Another solution for the same problem is the application of the interpolation-based fuzzy reasoning methods, where the derivable rules are deliberately missing. Since the rule base of a fuzzy interpolation-based controller, is not necessarily complete, it could contain the most significant fuzzy rules only without risking the chance of having no conclusion for some of the observations. In other words, during the construction of the fuzzy rule base, it is enough to concentrate on the cardinal actions; the "filling" rules (rules could be deduced from the others) can be deliberately omitted.

In the followings, first an approximate fuzzy reasoning method based on interpolation in the vague environment of the fuzzy rule base [4], [5], [6] will be reviewed. The main benefit of the proposed method is its simplicity, as it could be implemented to be simple and quick enough to be applied in practical direct fuzzy logic control too. Then its adaptation to behaviour-based control structures together with two simple examples will be discussed briefly.

2. INTERPOLATION-BASED FUZZY REASONING

One way of interpolative fuzzy reasoning is based on the concept of vague environment [2]. Applying the idea of the vague environment the linguistic terms of the fuzzy partitions can be described by scaling functions [2] and the fuzzy reasoning itself can be replaced by classical interpolation. The concept of vague environment is based on the similarity or indistinguishability of the elements. Two values in the vague environment are ε-distinguishable if their distance is grater than ε. The distances in vague environment are weighted distances. The weighting factor or function is called *scaling function (factor)* [2]. Two values in the vague environment X are ε-distinguishable if

$$\varepsilon > \delta_s(x_1,x_2) = \left| \int_{x_2}^{x_1} s(x)dx \right| \tag{1}$$

where $\delta_s(x_1,x_2)$ is the vague distance of the values x_1, x_2 and $s(x)$ is the scaling function on X. For finding connections between fuzzy sets and a vague environment the membership function $\mu_A(x)$ can be introduced as a level of similarity **a** to x, as the degree to which x is indistinguishable to **a** [2]. The α-cuts of the fuzzy set $\mu_A(x)$ are the sets which contain the elements those are (1−α)-indistinguishable from **a** (see Fig.1):

$$\delta_s(\mathrm{a,b}) \le 1-\alpha \qquad \mu_A(x) = 1-\min\{\delta_s(\mathrm{a,b}),1\} = 1-\min\left\{\left|\int_{\mathrm{a}}^{\mathrm{b}} s(x)dx\right|,1\right\} \tag{2}$$

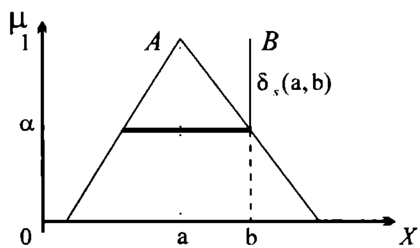

Figure 1: The α-cuts of $\mu_A(x)$ contains the elements that are (1−α)-indistinguishable from a

This case (Fig.1) the vague distance of points a and b ($\delta_s(\mathrm{a,b})$) is the *Disconsistency Measure* (S_D) of the fuzzy sets A and B (where B is a singleton):

$$S_D = 1-\sup_{x\in X} \mu_{A\cap B}(x) = \delta_s(\mathrm{a,b}) \text{ if } \delta_s(\mathrm{a,b}) \in [0,1], \tag{3}$$

where $A\cap B$ is the min t-norm, $\mu_{A\cap B}(x) = \min[\mu_A(x), \mu_B(x)]\ \forall\ x \in X$.

From the viewpoint of fuzzy reasoning and fuzzy rule bases, where an observation fuzzy set is needed to be compared to rule antecedents built up member fuzzy sets (linguistic terms) of the antecedent fuzzy partitions (2) and (3) means that the disconsistency measures between member fuzzy sets of a fuzzy partition and a singleton, can be calculated as vague distances of points in the vague environment of the fuzzy partition. The main difference between the disconsistency measure and the vague distance is, that the vague distance is a value in the range of [0,∞], while the disconsistency measure is limited to [0,1].

Therefore if it is possible to describe all the fuzzy partitions of the primary fuzzy sets (the antecedent and consequent universes) of the fuzzy rule base by vague environments, and the observation is a singleton, the "extended" disconsistency measures of the antecedent primary fuzzy sets of the rule base, and the "extended" disconsistency measures of the consequent primary fuzzy sets and the consequence can be calculated as vague distances of points in the antecedent and consequent vague environments.

The vague environment is described by its scaling function. For generating a vague environment of a fuzzy partition, an appropriate scaling function is needed to be find, which describes the shapes of all the terms in the fuzzy partition. A fuzzy partition can be characterised by a single vague environment if and only if the membership functions of the terms fulfil the following requirement [2]:

$$s(x)=\left|\mu'(x)\right|=\left|\frac{d\mu}{dx}\right| \text{ exists if } \min\{\mu_i(x),\mu_j(x)\}>0 \Rightarrow \left|\mu'_i(x)\right|=\left|\mu'_j(x)\right|, \quad \forall i,j\in I \tag{4}$$

where $s(x)$ is the vague environment.

Generally the above condition is not fulfilling, so the question is how to describe all fuzzy sets of the fuzzy partition with one "universal" scaling function. For this task the concept of *approximate scaling function*, as an approximation of the scaling functions which describe the terms of the fuzzy partition separately [4], [5], [6] is proposed. If the vague environment of a fuzzy partition (the scaling function or the approximate scaling function) exists, the member sets of the fuzzy partition can be characterised by points in the vague environment. (These points are characterising the cores of the fuzzy terms, while the membership functions are described by the scaling function itself.) If all the vague environments of the antecedent and consequent universes of the fuzzy rule base exist, all the primary fuzzy sets (linguistic terms) used in the fuzzy rule base can be characterised by points in their vague environment. Therefore the fuzzy rules (build on the primary fuzzy sets) can be characterised by points in the vague environment of the fuzzy rule base too. In this case the approximate fuzzy reasoning can be handled as a classical interpolation task. Applying the concept of vague environment (the distances of points are weighted distances), any interpolation, extrapolation or regression method can be adapted very simply for approximate fuzzy reasoning [4], [5], [6].

Because of its simple multidimensional applicability, for interpolation-based fuzzy reasoning in this paper the adaptation of the *Shepard operator* based interpolation (first introduced in [16]) is suggested. Beside the existing deep application oriented investigation of the Shepard operator e.g. [17], it was also successfully applied in the *Kóczy-Hirota fuzzy interpolation* [15]. (The stability and the approximation rate of the Shepard operator based Kóczy-Hirota fuzzy interpolation is deeply studied in [7] and [8].) The Shepard interpolation method for arbitrarily placed bivariate data was introduced as follows [16]:

$$S_0(f,x,y)=\begin{cases} f_k & \text{if } (x,y)=(x_k,y_k) \text{ for some } k, \\ \left(\sum_{k=0}^{n} f(x_k,y_k)/d_k^\lambda\right)\Big/\left(\sum_{k=0}^{n} 1/d_k^\lambda\right) & \text{otherwise,} \end{cases} \tag{5}$$

where measurement points x_k, y_k ($k \in [0,n]$) are irregularly spaced on the domain of $f \in \Re^2 \to \Re$, $\lambda > 0$, and $d_k = \left|(x-x_k)^2+(y-y_k)^2\right|^{1/2}$. This function can be typically used when a surface model is required to interpolate scattered spatial measurements.

The adaptation of the Shepard interpolation method for interpolation-based fuzzy reasoning in the vague environment of the fuzzy rule base is straightforward by substituting the Euclidian distances d_k by scaled distances $\delta_{s,k}$

$$\delta_{s,k} = \delta_s(\mathbf{a}_k,\mathbf{x}) = \left[\sum_{i=1}^{m}\left(\int_{a_{k,i}}^{x_i} s_{X_i}(\mathrm{x}_i)\,d\mathrm{x}_i\right)^2\right]^{1/2} \tag{6}$$

where s_{X_i} is the i^{th} scaling function of the m dimensional antecedent universe, $\mathbf{x}$ is the m dimensional crisp observation and $\mathbf{a}_k$ are the cores of the m dimensional fuzzy rule antecedents A_k.

Thus in case of singleton rule consequents the fuzzy rules R_k has the following form:

$$\textbf{If } \mathrm{x}_1 = A_{k,1} \textbf{ And } \mathrm{x}_2 = A_{k,2} \textbf{ And } \quad \textbf{And } \mathrm{x}_m = A_{k,m} \textbf{ Then } \mathrm{y} = \mathrm{c}_k \tag{7}$$

by substituting (6) to (5) the conclusion of the interpolative fuzzy reasoning can be obtained as:

$$\mathrm{y}(\mathbf{x})=\begin{cases} \mathrm{c}_k & \text{if } \mathbf{x}=\mathbf{a}_k \text{ for some } k, \\ \left(\sum_{k=1}^{r} \mathrm{c}_k/\delta_{s,k}^\lambda\right)\Big/\left(\sum_{k=1}^{r} 1/\delta_{s,k}^\lambda\right) & \text{otherwise.} \end{cases} \tag{8}$$

The interpolative fuzzy reasoning (8) can simply extend to be able to handle fuzzy conclusions by introducing the vague environment (scaling function) of the consequence universe. This case the fuzzy rules R_k has the following form:

$$\textbf{If } \mathrm{x}_1 = A_{k,1} \textbf{ And } \mathrm{x}_2 = A_{k,2} \textbf{ And } \quad \textbf{And } \mathrm{x}_m = A_{k,m} \textbf{ Then } \mathrm{y} = B_k. \tag{9}$$

By introducing vague distances on the consequence universe:

$$\delta_s(\mathrm{b}_k,\mathrm{y}) =: \left[\left(\int_{\mathrm{b}_k}^{y} s_{\mathrm{Y}}(\mathrm{y})d\mathrm{y}\right)^2\right]^{1/2} \tag{10}$$

where s_{Y} is the i^{th} scaling function of the one dimensional consequent universe, b_k are the cores of the one dimensional fuzzy rule consequents B_k.

Introducing the first element of the one dimensional consequence universe b_0 the (Y: $\mathrm{b}_0 < \mathrm{y} \ \ \forall\ \mathrm{y} \in \mathrm{Y}$), based on (8) and (10) the requested one dimensional conclusion $\mathrm{y}(\mathbf{x})$ can be obtained from the following formula:

$$\delta_s(\mathrm{y}(\mathbf{x}),\mathrm{b}_0) = \begin{cases} \delta_s(\mathrm{b}_k,\mathrm{b}_0) & \text{if } \mathbf{x} = \mathbf{a}_k \text{ for some } k, \\ \left(\sum_{k=1}^{r} \delta_s(\mathrm{b}_k,\mathrm{b}_0)/\delta_{s,k}^{\lambda}\right) \Big/ \left(\sum_{k=1}^{r} 1/\delta_{s,j}^{\lambda}\right) & \text{otherwise.} \end{cases} \tag{11}$$

A simple one-dimensional example for the approximate scaling function and the Shepard operator based interpolation (11) is introduced on Fig. 2 and on Fig. 3.

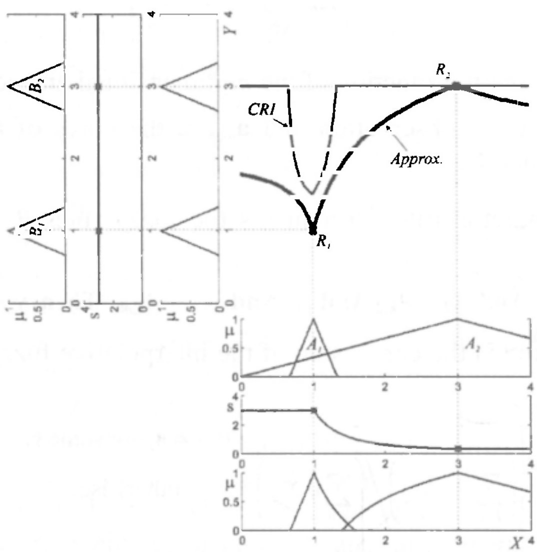

Figure 2: Interpolation of two fuzzy rules (R_i: $\mathrm{A}_i \rightarrow \mathrm{B}_i$) (see fig. 3 for notation)

For comparing the crisp conclusions of the interpolation-based fuzzy reasoning and the classical methods, the conclusions generated by the max-min compositional rule of inference (CRI) and the centre of gravity defuzzification for the same rule base is also demonstrated on the example figures (Fig. 2, Fig. 3). More detailed description of the proposed approximate fuzzy reasoning method can be found in [4], [5], [6].

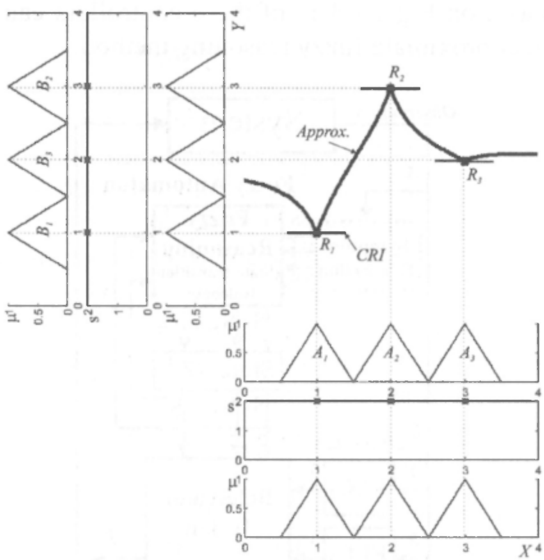

Figure 3: Interpolation of three fuzzy rules (R_i: $A_i \rightarrow B_i$) in the approximated vague environment of the fuzzy rule base. Using the Shepard operator based interpolation (p=1) (*Approx.*), and the min-max CRI with the centre of gravity defuzzification (CRI). Where μ is the membership grade and s is the scaling function.

3. THE APPLIED FUZZY BEHAVIOUR-BASED STRUCTURE

The main benefit of the interpolation-based fuzzy reasoning method, discussed in the previous chapter, is its simplicity. Applying look-up tables for pre-calculating the vague distances, it could be implemented to be simple and quick enough to fit the speed requirements of practical real-time direct fuzzy logic control systems, e.g. the requirements of fuzzy behaviour-based control too. The calculation efforts of many other interpolation-based fuzzy reasoning methods "wasted" for determining the exact membership shape of the interpolated fuzzy conclusion prohibits their practical application in real-time direct fuzzy logic control. The lack of the fuzziness in the conclusion is a disadvantage of the proposed method, but it has no influence in common applications where the next step after the fuzzy reasoning is the defuzzification.

In the followings a pure fuzzy behaviour-based control structure and the adaptation of the proposed interpolation-based fuzzy reasoning method will be discussed more detailed.

In case of pure fuzzy behaviour-based control structures all the main tasks of the behaviour-based control – the behaviour coordination, the behaviour fusion, and the behaviours themselves – are implemented on fuzzy logic controllers. (Such a structure is introduced on Fig.4.) Any of these controllers can apply the proposed interpolation-based approximate fuzzy reasoning method.

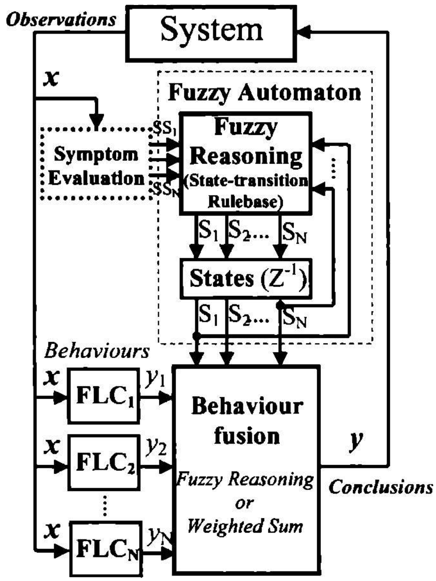

Figure 4: The suggested fuzzy behaviour-based control structure

For demonstrating the main benefits of the interpolation-based fuzzy reasoning in behaviour-based control, this paper concentrates on the many cases most heuristic part of the structure, on the behaviour coordination.

The task of behaviour coordination is to determine the necessities of the known behaviours needed for handling the actual situation. In the suggested behaviour-based control structure, for this task the finite state fuzzy automaton is adapted (Fig.4) [9]. This solution is based on the heuristic, that the necessities of the known behaviours for handling a given situation can be approximated by their suitability. Moreover the suitability of a given behaviour in an actual situation can be approximated by the similarity of the situation and the prerequisites of the behaviour. (Where the prerequisites of the behaviour is the description of the situations where the behaviour is valid (suitable itself)). In this case instead of determining the necessities of the known behaviours, the similarities of the actual situation to the prerequisites of all the known behaviours can be approximated.

Thus the first step of this kind of behaviour coordination is determining the similarities of the actual situation to the prerequisites of all the known behaviours – applying the terminology of fault classification; it is the symptom evaluation (see e.g. Fig.4). The task of symptom evaluation is basically a series of similarity checking between an actual symptom (observations of the actual situation) and a series of known symptoms (the prerequisites – symptom patterns – of the known behaviours). These symptom patterns are characterising the systems states where the corresponding behaviours are valid. Based on these patterns, the evaluation of the actual symptom is done by calculating the similarity values of the actual symptom (representing the actual situation) to all the known symptoms patterns (the prerequisites of the known behaviours). There exist many methods for fuzzy logic symptom evaluation. For example fuzzy classification methods e.g. the Fuzzy c-Means fuzzy clustering algorithm [1] can be adopted, where the known symptoms patterns are the cluster centres, and the similarities of the actual symptom to them can be fetched from the fuzzy partition matrix. On the other hand, having a simple situation, the fuzzy logic symptom evaluation could be a fuzzy rule based reasoning system itself.

One of the main difficulties of the system state approximation in behaviour coordination is the fact that in most of the cases the symptoms of the prerequisites of the known behaviours are strongly dependent on the actual behaviour of the system. Each behaviour has its own symptom structure. In other words for the proper system state approximation, the approximated system state is also needed. A very simple way of solving this difficulty is the adaptation of fuzzy automaton. In this case the state vector of the automaton is the approximated system state, and the state-transitions are driven by fuzzy reasoning (Fuzzy state-transition rule base on Fig.4), as a decision based on the previous actual state (the previous iteration step of the approximation) and the results of the symptom evaluation.

4. APPLICATION EXAMPLES

For demonstrating the simplicity of defining the rule base for interpolation-based fuzzy reasoning, as the first example, the state-transition rule base of the previously studied fuzzy automaton style behaviour coordination module applied for user adaptive information retrieval system in [10] and [11] will be discussed briefly in the followings. In this user adaptive information retrieval system example (introduced in [10] and [11] in more details) the user adaptivity is handled by combination of existing (off-line collected) human opinions (user models) in the function of their approximated similarity to the actual user opinions. As an analogy to the behaviour-based control applications, the different behaviours are the different existing user models, and the actual situation is the similarity of the actual user to the evaluators, originally gave the existing user models. Based on the observations (inputs) the conclusion of the user feedback (the symptom evaluation about the state-transition to state i, SS_i for all the possible states $\forall i \in [1,N]$) and the previous state values S_i – the new state values (i.e. the vector of the suitability of the existing user models) are needed to somehow be estimated. The suggested heuristic in this example is very simple. If a suitable model (S_i) is already found and the user feedback is still supporting it (SS_i), it is needed to be kept even if the user feedback began to support some other models too. If there were no suitable model, but the user feedback began to support one, it has to be picked it at once. In case of interpolation-based fuzzy reasoning, the above heuristic can be simply implemented by the following state-transition rule base [10], [11]. For the i[th] state variable S_i, $i \in [1,N]$ of the state vector **S**:

If S_i=One **And** SS_i=One **Then** S_i=One (12.1)

If S_i=Zero **And** SS_i=Zero **Then** S_i=Zero (12.2)

If S_i=One **And** SS_i=Zero
And SS_k=Zero **Then** S_i=One $\forall k \in [1,N], k \neq i$ (12.3)

If S_i=Zero **And** SS_i=One
And S_k=Zero **And** SS_k=Zero **Then** S_i=One $\forall k \in [1,N], k \neq i$ (12.4)

If S_i=Zero **And** SS_i=One
And S_k=One **And** SS_k=One **Then** S_i=Zero $\exists k \in [1,N], k \neq i$ (12.5)

where SS_i is the conclusion of the symptom evaluation about the state-transition to state i, $\forall i \in [1,N]$; N is the number of known behaviours (state variables). The structure of the state-transition rules is similar for all the state variables. Zero and One are linguistic labels of fuzzy sets (linguistic terms) representing high and low similarity. The interpretations of the Zero and One fuzzy sets can be different in each S_i, SS_i universes.

Please note that rule base (12) is sparse. It contains the main rules for the following straightforward goals only: Rule (12.1) simply keeps the previously chosen state values in the case if the symptom evaluation also agrees. The rule (12.2) has the opposite meaning, if the state values were not chosen, and moreover the symptom evaluation also disagrees, then the state value should be suppressed. The rule (12.3) keeps the already selected state values (previous approximation), even if the symptom evaluation disagrees, if it has no better "idea" Rules (12.4) and (12.5) have the task of ensuring the relatively quick convergence of the system to the sometimes unstable (changeable) situations, as new state variables which seem to be fit, can be chosen in one step, if there is no previously chosen state, which is still accepted by the symptom evaluation (12.4). (Rule (12.5) has the task to suppress this selection in the case if exists a still acceptable state, which has been already chosen.) The goal of this heuristic is to gain a relatively quick convergence for the system to fit the opinions of the actual user, if there is no state value high enough to be previously accepted. This quick convergence could be very important in many application areas e.g. in case of an on-line user adaptive selection system introduced in [10], where the user feed-back information needed for the state changes are very limited.

Some state changes of the fuzzy automaton in the function of the conclusion of the symptom evaluation (SS_1, SS_2) for the two states case (applying the state-transition rule base (12)) are visualised on Fig.5 and Fig.6.

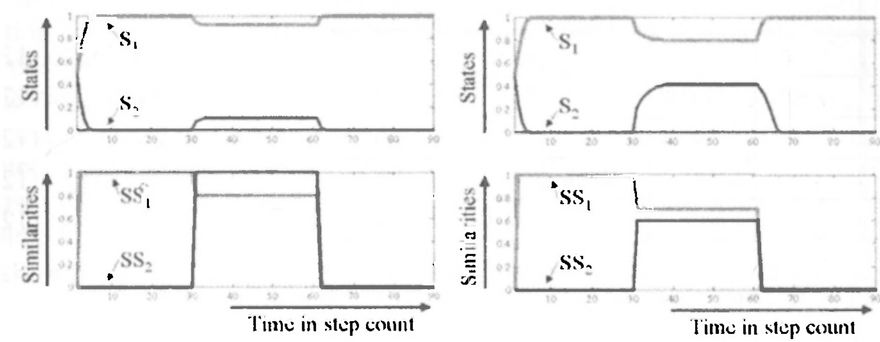

Figure 5: Do not "pick up" a new state if the previous approximation is still adequate

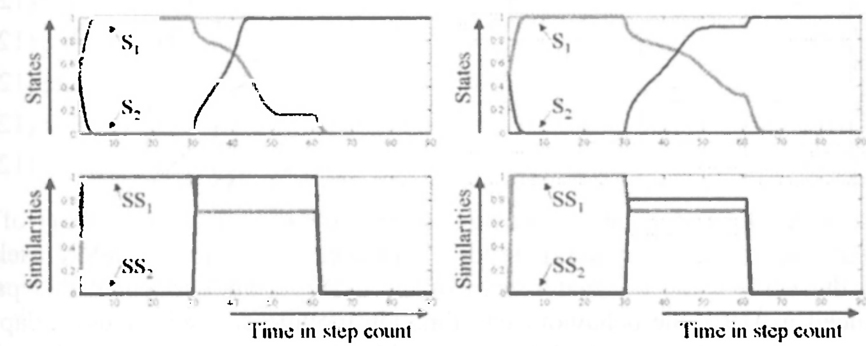

Figure 6: But "pick it up" if it seems better, or at least as good as the previous was

Counting the rules of the classical (e.g. compositional) fuzzy reasoning for the same strategy, in the two state case the complete rule base needs 16 rules (four observation universes (S_1, SS_1, S_2, SS_2) each with two terms fuzzy partitions (Zero, One) - 2^4 rules), while the sparse rule base (12) contains 5 rules only (see table 1 for the state-transition rule base of state S_1). Taking into account that in the proposed behaviour-based control structure a separate rule base is needed for each state variable, the behaviour coordination needs 32 rules, while 10 is enough in case of applying the proposed interpolation-based fuzzy reasoning method. Increasing the number of the state variables, the situation becomes even worse. In case of three state variables (S_1, S_2, S_3) the rate become $3 \cdot 2^6$ ($n \cdot 2^{2 \cdot n}$, where n is the number of the states) and $3 \cdot 6$ ($n \cdot (n+3)$) up to the interpolation-based method (see table 2).

Table 1: State-transition rule base of state S_1 in case of two state variables (S_1,S_2) according to rule base (12)

R_{S1}:	S_1	SS_1	S_2	SS_2	S_1	
1.,	One	One			One	according to (12.1)
2.,	Zero	Zero			Zero	according to (12.2)
3.,	One	Zero		Zero	One	according to (12.3)
4.,	Zero	One	Zero	Zero	One	according to (12.4)
5.,	Zero	One	One	One	Zero	according to (12.5)

Table 2: State-transition rule base of state S_1 in case of three state variables (S_1,S_2,S_3) according to rule base (12)

R_{S1}:	S_1	SS_1	S_2	SS_2	S_3	SS_3	S_1	
1.,	One	One					One	see (12.1)
2.,	Zero	Zero					Zero	see (12.2)
3.,	One	Zero		Zero		Zero	One	see (12.3)
4.,	Zero	One	Zero	Zero	Zero	Zero	One	see (12.4)
5.,	Zero	One	One	One			Zero	see (12.5)
6.,	Zero	One			One	One	Zero	see (12.5)

The exponential rule number "explosion" in case of increasing the number of the input variables makes many heuristic ideas unimplementable and therefore useless. E.g. in the case of the original source of the example application of this paper (introduced in [10]), the behaviour coordination module applied for user adaptive information retrieval system had 4 state variables (one for each emotional model), which makes this simple rule base (12) practically unimplementable as a complete rule base ($4 \cdot 2^8 = 1024$ rules). While the working demonstrational example (which can be downloaded from [18]) had only 28 rules thanks to the applied interpolation-based fuzzy reasoning method.

4.1. Vehicle navigation control example

For another example of the interpolation-based fuzzy rule base definition simplicity, in the followings, the behaviour coordination module of an automated guided vehicle (AGV) steering control [12], [13] will be discussed briefly.

In this example application the steering control has two main goals, the path tracking (to follow a guide path) and the collision avoidance. The simulated AGV is first trying to follow a guide path, and in the case if it is impossible (because of the obstacles), it leaves it, and as the collision situation is avoided, it tries to find the guide path and follow it again.

The AGV has two simulated sensor systems. The path sensing system senses the position of the guide path by special sensors (guide zone) tuned for the guide path. The obstacles are sensed directly by three ultrasonic distance sensors (on the front of the AGV, one in the middle (U_M) and one-one on both sides (U_L, U_R) (see Fig.7) and the obstacle boundaries are approximated based on dead reckoning and previous obstacle distances [12]. The global goal of the path tracking strategy is to follow the guide path by the guide zone with minimal path tracking error on the whole path (see Fig.7).

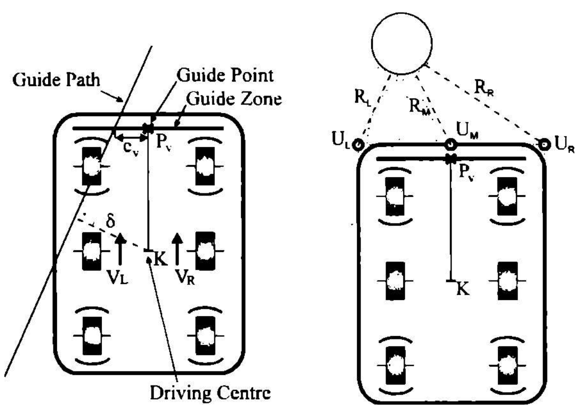

Figure 7: Differential steered AGV with guide zone, δ is the path tracking error, e_v is the distance of the guide path and the guide point P_v, K is the driving centre, R_L, R_R, R_M are the distances measured by the left, right and middle ultrasonic sensors (U_L, U_R, U_M).

Because of the requirement of being able to find the guide path after leaving it, the complete path tracking and the collision avoidance strategy needs four component behaviours:

Path tracking and restricted collision avoidance strategy: The main goal of this strategy is the path tracking (to follow a guide path) and as a sub goal, a kind of restricted (limited) collision avoidance [13]. (Here the restricted collision avoidance means, "avoiding obstacles without risking the chance of loosing the guide path".)

The collision avoidance strategy: The second known behaviour is a simple collision avoidance steering strategy. Its only goal is to avoid collisions.

The collision avoidance with left/right tendency strategy: The next two behaviours are basically the same as the collision avoidance steering strategy, expect the left or right turning tendencies in case of no left or right turning difficulties. These strategies are needed to help finding the path after leaving it (because of the fail of the first strategy).

In this vehicle navigation control example (introduced in [13] in more details) the studied behaviour coordination module has the task of determining the necessities of the four component behaviours. Having four known behaviours, the automaton has four state variables (see Fig.4).

These are the necessity of the path tracking and restricted collision avoidance strategy (S_P), the necessity of the collision avoidance strategy (S_C), and the necessities of the collision avoidance strategies with right tendency (S_{CR}), and left tendency (S_{CL}) in solving the actual situation.

Having four necessities (four conclusions), four state transition rule bases are required. The $\mathbf{R_{SP}}$ state transition rule base is determining the next value of the S_P state variable, $\mathbf{R_{SC}}$ is for determining S_C, and the $\mathbf{R_{SCR}}$ and $\mathbf{R_{SCL}}$ are determining the next values of S_{CR} and S_{CL}. The available observations [13] of the state transition rule bases are the distance between the guide path and the driving centre (e_v), the distances measured by the left middle and right ultrasonic sensors (R_L, R_M, R_R), the approximated maximal left and right turning angle without side collision (α_{ML}, α_{MR}), the availability of the path sensing (P_V), and the state variables themselves (S_P,S_C,S_{CR},S_{CL}).

Based on heuristic considerations and simulated experiments the four state-transition rule bases became the following ones:

$\mathbf{R_{SP}}$:

S_P	S_C	S_{CR}	S_{CL}	e_v	PV	R_L	R_R	R_M	α_{ML}	α_{MR}	S_P	
				Z	V			L			L	(13.1)
				PL	V					S	Z	(13.2)
				NL	V				S		Z	(13.3)
					NV						Z	(13.4)

$\mathbf{R_{SC}}$:

S_P	S_C	S_{CR}	S_{CL}	e_v	PV	R_L	R_R	R_M	α_{ML}	α_{MR}	S_C	
					V			S			L	(14.1)
					V			L			Z	(14.2)
					NV						Z	(14.3)

$\mathbf{R_{SCR}}$:

S_P	S_C	S_{CR}	S_{CL}	e_v	PV	R_L	R_R	R_M	α_{ML}	α_{MR}	S_{CR}	
L				NVL	V						L	(15.1)
		L			NV						L	(15.2)
				Z	V			L			Z	(15.3)
			L								Z	(15.4)

$\mathbf{R_{SCL}}$:

S_P	S_C	S_{CR}	S_{CL}	e_v	PV	R_L	R_R	R_M	α_{ML}	α_{MR}	S_{CL}	
L				PVL	V						L	(16.1)
			L		NV						L	(16.2)
				Z	V			L			Z	(16.3)
		L									Z	(16.4)

where the linguistic labels of fuzzy sets (linguistic terms) stand for N: negative, P: positive, VL: very large, L: large, S: small, Z: zero, V: path valid, NV: path not valid.

The heuristic considerations laying behind the state-transition rule base of the path tracking and restricted collision avoidance strategy ($\mathbf{R_{SP}}$) are quite straightforward: Rule (13.1) simply takes the path tracking strategy in case if there is a valid path with no path tracking error (e_v=Z) and there is no collision situation. The rest of the rules are suppressing the path tracking strategy in case of collision situation (13.2)-(13.3), or if the path sensing is not available (13.4).

The state-transition rule base of the collision avoidance strategy ($\mathbf{R_{SC}}$) is also straightforward: Rule (14.1) calls the collision avoidance strategy in case of valid path sensing and collision situation. Rule (14.2) suppresses the collision avoidance strategy if there is no collision situation (the distance of the obstacle and the middle sensor is large) and the path sensing is valid. Rule (14.3) suppresses the collision avoidance strategy if the path sensing is not valid (PV=NV), as these situations are handled by the collision avoidance strategies with left and right tendencies.

The remaining two state-transition rule bases are serving the requirement of being able to find the guide path after leaving it. They are symmetric in the sense of the left and right directions. The right turning tendency is called if the vehicle leaves the guide path on the left side (see rule (15.1)) and left turning tendency is called if the vehicle left on the right (rule (16.1)). Rules (15.2) and (16.2) have the task to keep the already selected right or left direction tendency if the path sensing is still not available (PV=NV). Rules (15.3) and (16.3) are suppressing the strategies serving the free run if the guide path is found again (valid path PV=V with no path tracking error e_v=Z). Rules (15.3) and (16.3) are serving of the mutual exclusion of the two contradictive (left or right turning tendencies) strategies.

Figure 8 introduces some results of the simulated AGV steering application.

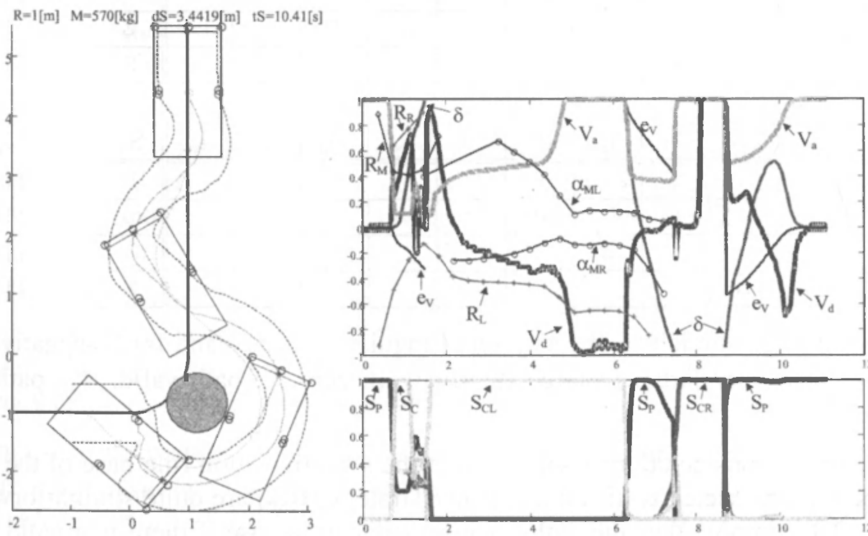

Figure 8: Track of a single run in case of one obstacle and the time function of observations, conclusions and system state values (S_P, S_C, S_{CL}, S_{CR}).

A downloadable and runable code of the application examples and the code of the interpolation-based fuzzy reasoning method studied in this paper can be found at [18].

5. CONCLUSIONS

The goal of this paper was to review an interpolation-based fuzzy reasoning method, which could be implemented to be simple and quick enough to fit the requirements of behaviour-based control structures in real-time direct fuzzy logic control systems. The suggested approximate fuzzy reasoning method based on interpolation in the vague environment of the fuzzy rule base gives an efficient way for designing direct fuzzy logic control applications. The lack of the fuzziness in the conclusion is a disadvantage of the proposed method, but it has no influence in