Control and Navigation with Knowledge Bases

Miloš Šeda and Tomáš Březina

Abstract—In this paper, we focus on the use of knowledge bases in two different application areas – control of systems with unknown or strongly nonlinear models (i.e. hardly controllable by the classical methods), and robot motion planning in eight directions. The first one deals with fuzzy logic and the paper presents approaches for setting and aggregating the rules of a knowledge base. Te second one is concentrated on a case-based reasoning strategy for finding the path in a planar scene with obstacles.

Keywords—fuzzy controller, fuzzification, rule base, inference, defuzzification, genetic algorithm, neural network, case-based reasoning

I. INTRODUCTION

THE classical literature, dealing with automatic control, describes many sophisticated methods of designing these controllers in accordance with their dynamic behaviour. Mostly, these methods require a precise mathematical model of the controlled system. However, for complex systems, such a model may be difficult or even impossible to find or it may be strongly nonlinear [2], [15], which causes many difficulties in designing a controller. A promising way is to use the fuzzy logic approach to control [6], [20], [23]. Fuzzy logic uses the interval from 0 (False) to 1 (True) to describe human reasoning making it possible to formulate a rule base which is a major tool used to generate the values of the manipulated variable of the controller by its input values

II. FUZZY LOGIC

While the values of algebraic variables are usually given by numbers (in the Boolean algebra, for example, by the logical 1 and 0), in fuzzy logic [1], [13], [17], [22] we work with *linguistic variables* whose values may be words or sentences. A linguistic variable is determined by: 1) name, 2) set of values that it can have (these values are called *terms* being represented by fuzzy sets), 3) universe defined for all terms and 4) function that maps words into values of a universe.

For example, a linguistic variable x with label "Age" [7] may have terms from the set T = {very old, old, not so old, more or less young, young, very young}, the universe is the interval [0, 120] and the functions mapping word values to the universe have the shapes from Fig. 3. The simple fuzzy sets are mostly not sufficient and thus we use their composition. In

Miloš Šeda and Tomáš Březina work in the Institute of Automation and Computer Science, Faculty of Mechanical Engineering, Brno University of Technology, Technická 2896/2, CZ 616 69 Brno, Czech Republic (phone: +420-54114 3332; fax: +420-54114 2330; e-mail: seda@fme.vutbr.cz, brezina@fme.vutbr.cz.

fuzzy controllers, which can be studied in the next sections, combined conditions occur and an output variable is given by a combination of several rules. However, we must only determine one membership function from these combinations of rules. This operation is called *inference* being performed by an inference engine.

Let us define the basic operations over fuzzy sets – union, intersection and complement. The logical operators for disjunction and conjunction will be denoted by the symbols $\wedge$ (conjunction, logical AND) and $\vee$ (disjunction, logical OR) to avoid misunderstanding.

The *intersection* of two fuzzy sets A, B in X, $A \cap B$, is defined as

$$\mu_{A \cap B}(x) = \mu_A(x) \wedge \mu_B(x), \ \forall x \in X \tag{1}$$

where "$\wedge$" is the minimum, it corresponds to the connective AND.

The *union* of two fuzzy sets A, B in X, $A \cup B$, is defined as

$$\mu_{A \cup B}(x) = \mu_A(x) \vee \mu_B(x), \ \forall x \in X \tag{2}$$

where "$\vee$" is the maximum, it corresponds to the connective OR.

The *complement* of a fuzzy set A in X, $\neg A$, is defined as

$$\mu_{\neg A}(x) = 1 - \mu_A(x), \ \forall x \in X \tag{3}$$

corresponding to the negation NOT.

In fuzzy logic, an analogy to the complement law and contradiction law in classical logic is not satisfied. This is caused by the fact that some elements partially belong to a fuzzy set and partially to its complement. Therefore, e.g., the intersection of a fuzzy set and its complement is not empty. Each element belongs to this intersection with a grade of membership lower or equal to 0.5.

Apart from (1) and (2), there are also the other definitions for the evaluation of the membership functions of intersection, and union based on the minimum and maximum operations. Instead of the intersection, a *triangular norm* (shortly *t-norm*) can be used, and instead of the union, a *triangular conorm* (shortly *t-conorm* or *s-norm*) can be used. These norms must satisfy several axioms (boundary condition, monotonicity, commutativity and associativity), see [13].

In Table I and Table II, basic t-norms and t-conorms (s-norms) are summarised. The first row of this table corresponds to Equations (1) and (2).

Like in the classical logic, we can generalise binary relations to higher order relations.

The key operation in the fuzzy control is the fuzzy implication, which is a base of rules in a knowledge base.

In classical logic, we can express the implication by the equivalent formulas as follows:

$$p \rightarrow q = \neg p \vee q = \neg p \vee (p \wedge q) = (\neg p \wedge \neg q) \vee q =$$

$$= \max \ \{x \in \{0,1\} \mid (p \wedge x) \leq q\} \qquad (4)$$

TABLE I
T-NORMS

name	formula
standard intersection	$i(a,b) = \min(a,b)$
algebraic product	$i(a,b) = ab$
bounded difference	$i(a,b) = \max(0, a+b -1)$
drastic intersection	$i(a,b) = \begin{cases} a, \text{ if } b = 1 \\ b, \text{ if } a = 1 \\ 0, \text{ otherwise} \end{cases}$

TABLE II
T-CONORMS (S-NORMS)

name	formula
standard union	$u(a,b) = \max(a,b)$
algebraic sum	$u(a,b) = a + b - ab$
bounded sum	$u(a,b) = \min(1, a+b)$
drastic union	$u(a,b) = \begin{cases} a, \text{ if } b = 0 \\ b, \text{ if } a = 0 \\ 1, \text{ otherwise} \end{cases}$

Now we define a fuzzy relation and related notions, which will be necessary for the following considerations.

If $X = \{x\}$ and $Y = \{y\}$ are two universes, then a *fuzzy relation R* is defined as a fuzzy set in the Cartesian product $X \times Y$, characterised by its membership function $\mu_R : X \times Y \rightarrow [0,1]$; $\mu_R(x,y) \in [0,1]$ reflects the *strength of relation* between $x \in X$ and $y \in Y$.

Note that, for finite, small enough X and Y, a fuzzy relation may be evidently shown in the matrix form.

Let R be a fuzzy relation in $X \times Y$ and S be a fuzzy relation in $Y \times Z$. Their (*max-min*) *composition* is a fuzzy relation R o S in $X \times Z$ defined by

$$\mu_{R \circ S}(x, z) = \sup_{y \in V} \left[\mu_R(x, y) \wedge \mu_R(y, z) \right] \qquad (5)$$

If we interpret conjunction, disjunction and negation by means of *t*-norm, *t*-conorm and fuzzy complement, then, combining them, we get a large number of different fuzzy implications. Some authors such as Mamdani suggested other formulations derived from (5) and therefore the total number of implications is equal to 72 [7].

Let A be a fuzzy set in X and B be a fuzzy set in Y. The following is a list of membership function formulas of some common fuzzy implications $R(A,B)$:

Kleene-Dienes
$$\mu_{A \rightarrow B}(x, y) = \max \ \{1 - \mu_A(x), \mu_B(y)\} \qquad (6)$$

Lukasiewicz
$$\mu_{A \rightarrow B}(x, y) = \min \ \{1, 1 - \mu_A(x) + \mu_B(y)\} \qquad (7)$$

Zadeh (Willmott)
$$\mu_{A \rightarrow B}(x, y) = \max \ \{\min [\mu_A(x), \mu_B(y)], 1 - \mu_A(x)\} \qquad (8)$$

stochastic implication

$$\mu_{A \rightarrow B}(x, y) = \min \ \{1, 1 - \mu_A(x) + \mu_A(x). \ \mu_B(y)\} \qquad (9)$$

Reichenbach
$$\mu_{A \rightarrow B}(x, y) = 1 - \mu_A(x) + \mu_A(x).\mu_B(y) \qquad (10)$$

Mamdani
$$\mu_{A \rightarrow B}(x, y) = \min \ \{\mu_A(x), \mu_B(y)\} \qquad (11)$$

Larsen
$$\mu_{A \rightarrow B}(x, y) = \mu_A(x).\mu_B(y) \qquad (12)$$

Rescher-Gaines
$$\mu_{A \rightarrow B}(x, y) = 1, \text{ if } \mu_A(x) \leq \mu_B(y); 0, \text{ otherwise} \qquad (13)$$

Gödel
$$\mu_{A \rightarrow B}(x, y) = 1, \text{ if } \mu_A(x) \leq \mu_B(y); \mu_B(y), \text{ otherwise} \qquad (14)$$

Goguen
$$\mu_{A \rightarrow B}(x, y) = \min \ \{1, \mu_A(x)/\mu_B(y)\}, \text{ if } \mu_B(y) \neq 0 ; \\ 1, \text{ otherwise} \qquad (15)$$

Sharp (Gorgen)
$$\mu_{A \rightarrow B}(x, y) = 1, \text{ if } \mu_A(x) < 1 \text{ or } \mu_B(y) = 1; 0, \text{ otherwise} \qquad (16)$$

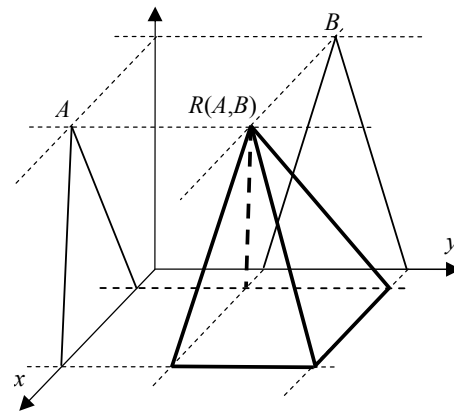Figure 1 shows a fuzzy set A in X, fuzzy set B in Y and Mamdani fuzzy implication $R(A,B)$.

Fig. 1 Mamdani implication

III. FUZZY CONTROL AND MAMDANI MODEL

A basic feedback connection of a regulation circuit with a fuzzy controller is shown in Fig. 2 [8]. The inputs are given by crisp data resulting from a measurement. Since, in fuzzy logic, we work with values from the interval [0,1], these fuzzy controller input data must be first transformed into this interval. This operation is called *normalisation*. If we know that the observed variable x may have values between a lower boundary l and an upper boundary u, its current value x_a can be easily converted to a value in interval [0, 1] using Equation (17).

$$\frac{x_a - l}{u - l} \qquad (17)$$

The measured values are normalised in a *preprocessing block* where the signal is filtered to eliminate the noise impact.

In the *fuzzification block*, which is the first block of a fuzzy

controller, each measured value is converted to a grade of membership of one or more *membership functions*, which correspond to the terms of a linguistic variable. For simplicity, we select as membership functions mainly the functions from Fig. 3, which consist of linear segments and are described by Equations (18)-(21).

$$L(x,a,b) = \begin{cases} 1 & \text{for } x < a \\ \dfrac{b-x}{b-a} & \text{for } a \le x \le b \\ 0 & \text{for } x > b \end{cases} \quad (18)$$

$$\Lambda(x,a,b,c) = \begin{cases} 0 & \text{for } x < a \\ \dfrac{x-a}{b-a} & \text{for } a \le x \le b \\ \dfrac{c-x}{c-b} & \text{for } b < x \le c \\ 0 & \text{for } x > c \end{cases} \quad (19)$$

$$\Pi(x,a,b,c,d) = \begin{cases} 0 & \text{for } x < a \\ \dfrac{x-a}{b-a} & \text{for } a \le x \le b \\ 1 & \text{for } b < x \le c \\ \dfrac{c-x}{d-c} & \text{for } c < x \le d \\ 0 & \text{for } x > d \end{cases} \quad (20)$$

$$\Gamma(x,a,b) = \begin{cases} 0 & \text{for } x < a \\ \dfrac{x-a}{b-a} & \text{for } a \le x \le b \\ 1 & \text{for } x > b \end{cases} \quad (21)$$

In the next step, the fuzzy controller will determine, on the basis of expert knowledge, the word values of the manipulated variables (e.g., a manipulated variable may be positive big). This is the function of the *fuzzy controller*. Finally, the word terms are converted to crisp number values in a *defuzzification* process.

A block diagram of the fuzzy control is shown in Fig. 4.

The block diagram of a discrete regulation circuit is shown in Fig. 4. It is a slightly modified version of Fig. 2 extended by analogue/digital (A/D) and digital/analogue (D/A) converters.

Let w be command variable, e be error, ∇e be change in error, δe be accumulated error, u manipulated variable and y controlled variable. The error, change in error and accumulated error are the input values of the fuzzy controller.

If k is the time, the input values in a digital form can be expressed as follows:

$$e(k) = w(k) - y(k) \quad (22)$$

$$\nabla e(k) = e(k) - e(k-1) \quad (23)$$

$$\delta e(k) = \sum_{i=1}^{k} e(i) \quad (24)$$

The rules in the rule base use these input values and generate the value of the manipulated variable. Basically, a fuzzy controller contains rules in the IF THEN format.

The rules of a fuzzy PSD controller will have the following form

IF e=value1 **AND** ∇e=value2 **AND** δe=value3
   THEN u:=value4 $\quad (25)$

The values can usually be specified by one of the linguistic terms: NB (Negative Big), NM (Negative Medium), NS (Negative Small), ZO (Zero), PS (Positive Small), PM (Positive Medium), PB (Positive Big).

The *fuzzy PSD controller* approximates Equation (26) or (27).

$$u(k) = k_P e(k) + k_S \sum e(k) + k_D \nabla e(k) \quad (26)$$

$$\nabla u(k) = k_P \nabla e(k) + k_S e(k) + k_D \nabla^2 e(k) \quad (27)$$

The rules and equations for P, PS, and PD fuzzy controllers are given by a straightforward simplification of (26) and (27).

The execution of each rule is based on the way the conditions are aggregated in the antecedent of the rule and the selection of implication between antecedent and consequent.

Determining a relation between input and output values forms rules that serve both for situations included in a rule base and for situations that are not explicitly described in it. Fuzzy regulation is based on the following steps: Within a certain time from the measurement, the crisp values of all the input values are available. These values differ from the linguistic terms in rules. The similarity of the crisp value and the linguistic term in the fuzzification step for each input variable is determined. Next we must determine the grade of membership for the aggregation of conditions in the antecedent and, using an inference engine, determine the grade of membership for the consequent. If the same output is a consequent of more rules, then these consequents must also be aggregated and we determine only one grade of membership for this output value. In the defuzzification block, this fuzzy set is converted to a crisp value (e.g. voltage).

Let us consider two rules where the antecedent will be given by a conjunction of two conditions and consequents will be specified by the same output variable. If we want to process such rules, we must decide how to interpret the rules and their components and how to aggregate their consequents.
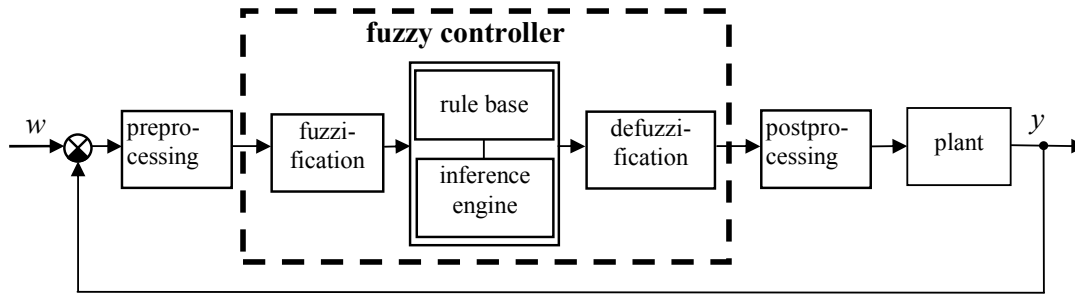
Fig. 2 Feedback control with a fuzzy controller
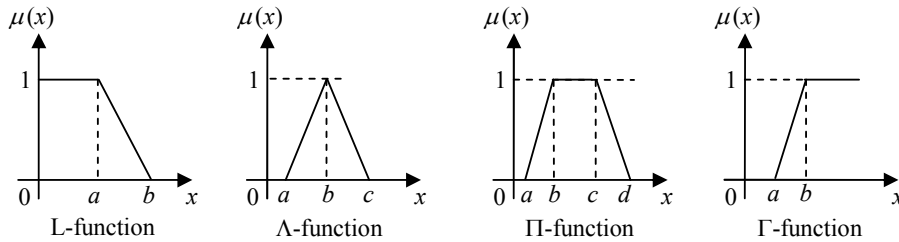

Fig. 3 Typical membership functions
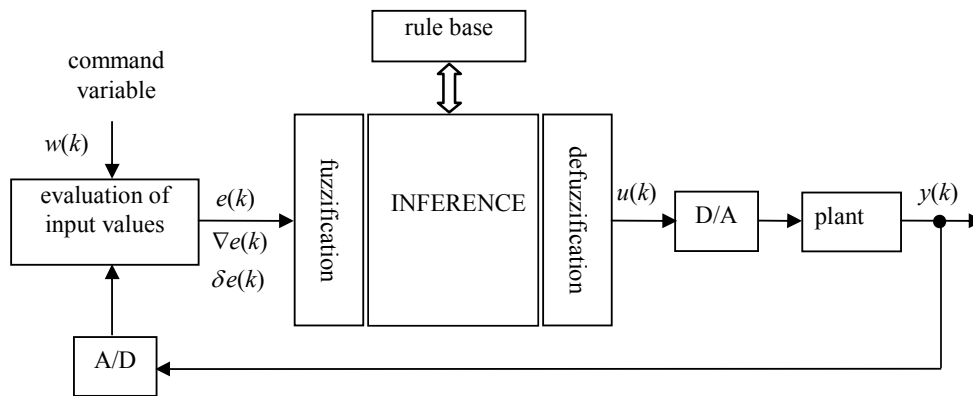

Fig. 4 A block diagram of fuzzy control

1. The interpretation of rules is unique because
   IF <fuzzy statement 1> **THEN** <fuzzy statement 2> is the fuzzy implication, however we must select its type.
2. A conjunction in the antecedent is realised by a *t*-norm and must be selected, too. Similarly we would select a *t*-conorm (*s*-norm) for a disjunction in the antecedent.
3. For the aggregation of consequents, we select a disjunction, which means that we must select a *t*-conorm (s-norm).

Let us consider the Mamdani implication, conjunction (AND) realised as minimum and disjunction realised as maximum. From Equation (11), we see that the Mamdani implication uses the operation of minimum. Although it is a very simple operation, graphically expressed, it represents a 3D object as shown in Fig. 1

The starting mechanism of the evaluation is given by the measurements of the crisp values of the input variables.

In Fig. 6, it can be seen how the result of the Mamdani implication from Figure 1 will change if variable x will have a crisp value of a_0.

In Fig. 5, the crisp value $x=a_0$ in the tetrahedral pyramid will mark off a trapezoid that is projected in the universe Y of variable y. Thus, the evaluation can be simplified to a planar mapping.

In our example with two rules of the following form

$$\text{IF } x_1 = A_{11} \text{ AND } x_2 = A_{12} \text{ AND THEN } y = B_1$$
$$\text{IF } x_1 = A_{21} \text{ AND } x_2 = A_{22} \text{ AND THEN } y = B_2,$$

where A_{11}, A_{12}, B_1, A_{21}, A_{22}, B_2 are terms with triangular membership function, the Mamdani implication, minimum operation for conjunction, and maximum for aggregation disjunction are used. If the crisp values $x_1 = a_{10}$ and $x_2 = a_{20}$ are measured, then we get the result by Fig. 6.
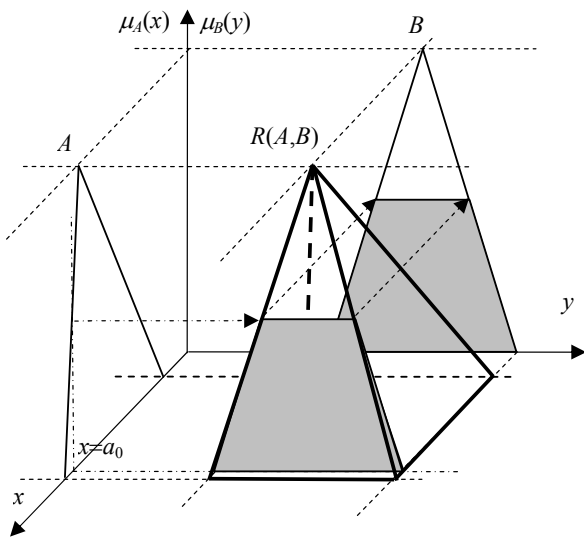
Fig. 5 Inference mechanism for the Mamdani implication
and crisp input

Finally, we must assign a crisp value to the area given by aggregating the rules in the inference engine. Again, this defuzzification step is not unique and there may be numerous approaches to implementing it. Most of them are empirical.

The most frequent is the *centre of gravity* (COG) method. The crisp value y_0 is evaluated by Equation (28).

$$y_0 = \frac{\int_U y\,\mu_A(y)\,dy}{\int_U \mu_A(y)\,dy}, \qquad (28)$$

where U is continuous universe.

For discrete universe, we calculate the resulting value of the output variable by Equation (29).

$$y_0 = \frac{\sum_{i=1}^{n} y_i\,\mu_A(y_i)}{\sum_{i=1}^{n} \mu_A(y_i)} \qquad (29)$$

A drawback of this evaluation is that it does not consider overlapping areas. Therefore many other methods for the evaluation of the defuzzificated value are used, e.g. *bisector of area* (BOA), *mean of maxima* (MOM), *leftmost maximum* (LM), *rightmost maximum* (RM). More details can also be found in [3], [6], [7], [9], [18].

In the general scheme of fuzzy regulation circuit, the defuzzification block is followed by a *postprocessing* block that, from the interval of the values normalised to [0, 1], provides its conversion into a domain that uses the controlled system.

IV. TAKAGI-SUGENO FUZZY MODEL

In the previous section we studied the Mamdani fuzzy model whose rules can be generally expressed by (30)

IF $x_1=A_{1r}$ **AND** $x_2=A_{2r}$ **AND … AND** $x_n=A_{nr}$
    THEN $y:=B_r$, $r=1, \dots , R$,   (30)

where $x_1, x_2, \dots , x_n$ are inputs, R is the number of rules and values $A_{1r}, A_{2r},\dots , A_{nr}, B_r$ are specified by linguistic terms.

The second type is represented by the Takagi-Sugeno fuzzy model where the antecedents of rules have the same form as in the Mamdani model but the T-S model differs in the consequents of rules and aggregation methods. Instead of using fuzzy sets, the consequent part of rules in the Takagi-Sugeno model is a linear combination of inputs $x_1, x_2, \dots , x_n$. Then, the rules in the T-S model can be expressed by (31)

IF $x_1=A_{1r}$ **AND** $x_2=A_{2r}$ **AND … AND** $x_n=A_{nr}$
    THEN $y:=f_r(x_1, x_2, \dots , x_n)$, $r=1, \dots , R$,   (31)

where R is the number of rules and the values $A_{1r}, A_{2r},\dots , A_{nr}$ are specified by linguistic terms.

The defuzzified output (i.e. the aggregated crisp value of the output) is then given as the weighted average of the contributions from each rule by (32):

$$y = \frac{\sum_{r=1}^{R} w_r f_r(x_1,\dots,x_n)}{\sum_{r=1}^{R} w_r} \qquad (32)$$

If **AND** is realised by min t-norm and $x_{10}, x_{20}, \dots , x_{n0}$ are the measured values of inputs, then the weights of rules are given by (33)

$$w_r = \min\{\mu_{A_{1r}}(x_{10}), \mu_{A_{2r}}(x_{20}), \dots , \mu_{A_{nr}}(x_{n0})\}, r=1,...,R \quad (33)$$

The Takagi-Sugeno fuzzy model is also presented in simplified versions called *zero-order* model and *first-order* model.

The zero-order model is identical to a singleton output rule and has the following form:

IF $x_1=A_{1r}$ **AND** $x_2=A_{2r}$ **AND … AND** $x_n=A_{nr}$
    THEN $y := c_r$ (34)
where c is a crisp constant.
The first-order model with two inputs is given by (35).

    IF $x_1=A_{1r}$ **AND** $x_2=A_{2r}$ **THEN** $y := a*x_1+b*x_2+c$ (35)
where a, b, c are constants.

Generally, a description of relations between the input and output variables is easier for the Mamdani models than for the Sugeno model because it is sufficient to decompose the input and output space into fuzzy regions and approximate each region by linear models, typically covering intervals of possible values by fuzzy sets with triangular membership functions. In the T-S model, we must get an expert knowledge for consequents of rules.
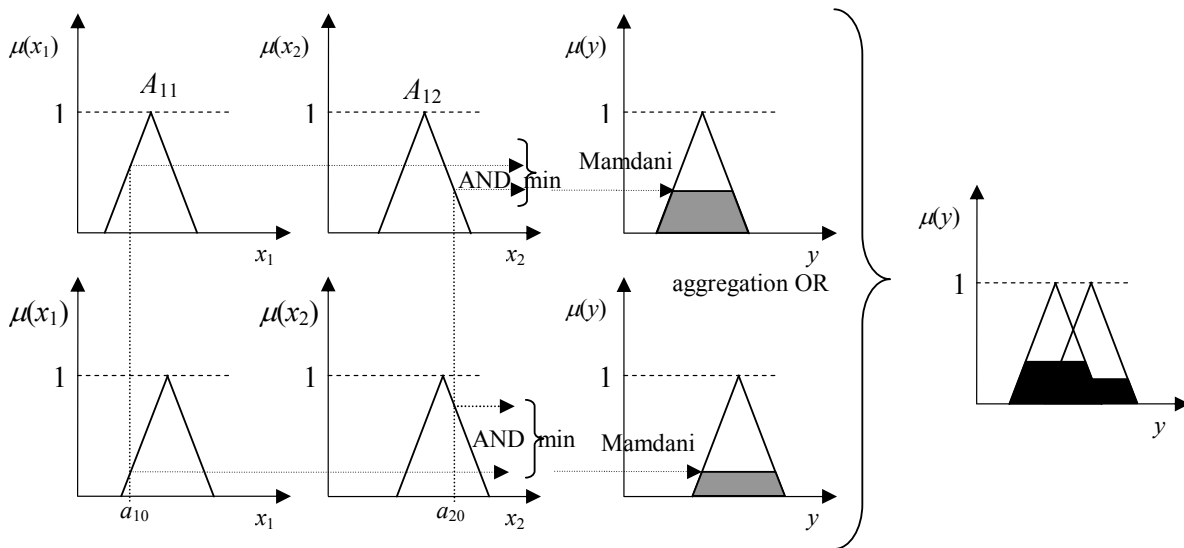
Fig. 6 Inference mechanism for the Mamdani implication and min-max operations

V. NEURAL NETWORKS AND GENETIC ALGORITHMS

In addition to the above-mentioned steps selecting the components of a fuzzy controller, we may use modified approaches that may enable to change the initial rule base and fine tune the fuzzy membership function parameters of the predefined sets so as to avoid performance with a large overshoot and long settling time. While the fuzzy controller design is based on intuition and experience, the rule base and the shape of the membership functions are refined through simulation and testing.

Genetic algorithms are often applied to the if-then linguistic knowledge base of a controlled process and it is assumed that this base has already been designed, either by a human expert or by a prior learning process [11], [12], [21]. The chromosome codes membership functions (and/or scaling functions) for input and output variables. If we have the Mamdani fuzzy model with two inputs (e.g. error and change in error) and one output, both inputs are represented by n linguistic terms (e.g. 7 – NB, NM, NS, ZO, PS, PM and PB) and the output by m linguistic terms, there are $n \times n$ rules in all that form the fuzzy control rule base. These pairs of input values need not be stored in chromosomes because they are fixed and only outputs are recomputed by a genetic algorithm. If 4 bits are sufficient for encoding the output linguistic terms, each chromosome will have $4n^2$ bits for the outputs relating to all pairs of inputs. In Fig. 7 each four consecutive bits are coded to represent the output for each rule.

The fitness function is defined according to the controlled process and is given by the performance of the plant for the computed output, e.g. by its transient response or steady-state.

The generated groups of 4 bits in the chromosomes are decoded by their transformation in the range of the output variable.
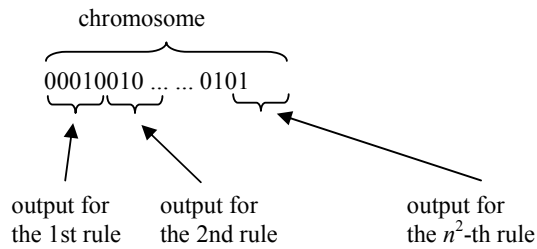

Fig. 7 Coding of rules

For example, if the range of output variable PS is [0.15, 0.25] and its binary string is 0101, then the output variable PS is computed as

$$PS = 0.15 + 1/15 \times (0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0) \times (0.25 - 0.15) =$$
$$= 0.15 + 1/15 \times 5 \times 0.1 \approx 0.183$$

Another approach to improving the dynamic properties of a control algorithm is to use a more complex feedback circuit, e.g. the two-loop model-based predictive control with parallel distributed compensation [5] or a more general shape of membership functions [10].

To control highly uncertain and nonlinear dynamical systems, fuzzy systems-based adaptive methodologies in combination with neural networks are frequently used [16]. This approach makes it possible to use the linguistic rule base of fuzzy systems together with the learning capabilities of neural networks which may bring a synergistic effect. The neuro-fuzzy architecture is similar to classical feed-forward multilayer neural networks with an input layer, one or more hidden layers, and an output layer. However, its neurons are more general. Instead of inputs represented by real numbers and real weights, here we have fuzzy inputs and/or fuzzy weights. A 3-layer feedforward neural network is shown in Fig. 8.
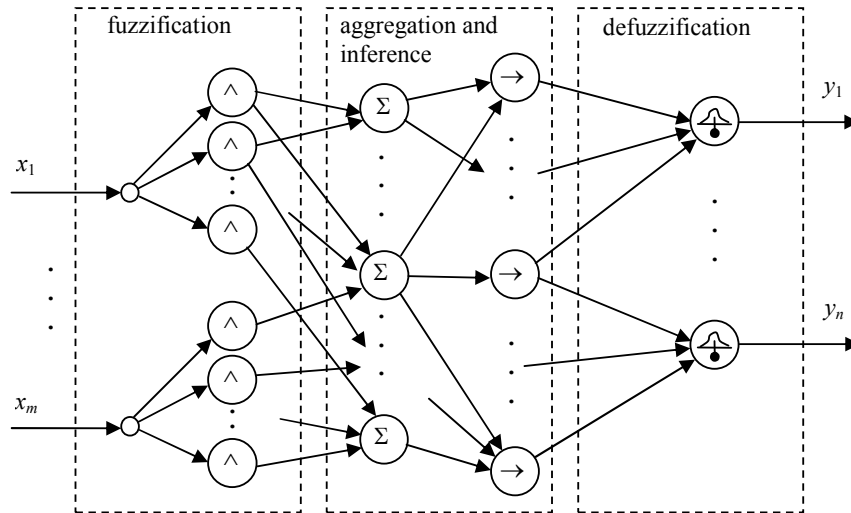
Fig. 8 Neuro-fuzzy structure with 3 layers, *m* inputs and *n* outputs

In this hybrid architecture, the neural network is initialized by fuzzy knowledge. After the learning phase using experimental data (e.g. based on step functions), its result is mapped back into fuzzy rules.

The weights and biases in the neural network can also be computed using genetic algorithms [4]. In this case, GA chromosome is created from potential weights and biases. The GA cost function computes the mean square difference between the current guess of the function and the exact function evaluated at specific points

VI. ROBOT MOTION PLANNING

There are three basic types of robot motion planning algorithms [19].

1. *Potential field method*. The robot moves in the direction of the gradient of a potential field produced by the goal configuration and the obstacles.
2. *Cell decomposition method*. Here, the scene is decomposed into cells and the outcome of the search is a sequence of adjacent cells between start and target from which a continuous path can be computed. The square cell decomposition can be used for 8-directional (horizontal, vertical and diagonal) robot motion in the plane with static rectangular obstacles.
3. *Roadmap methods*. The roadmap is built by a set of paths where each path consists of collision free area connections. There are several different methods for developing the roadmap such as visibility graphs and Voronoi diagrams [19].

We will not deal with a comparison of these methods and their drawbacks and only concentrate on a case-based reasoning procedure [14], which can be applied for as an additional method for the cell decomposition.

Case-based reasoning (CBR) is based on the retrieval and adaptation of old solutions to new problems.

A general CBR cycle may be given by the following steps:

- Retrieve the most similar case or cases;
- Reuse the information and knowledge in that case to solve the problem;
- Revise the proposed solution;
- Retain the parts of this experience likely to be useful for future problem solving.

If, for a given start cell c_s^0 and a given goal cell c_g^0, the case-base does not contain a path leading from c_s^0 to c_g^0, a similar path is retrieved according to the formula

$$P(c_s', c_g') = \arg\min \left\{ \begin{array}{l} F\left(P(c_s, c_g)\right), \\ d(c_s^0, c_s) \le \delta, \ d(c_g^0, c_g) \le \delta \end{array} \right\} \quad (36)$$
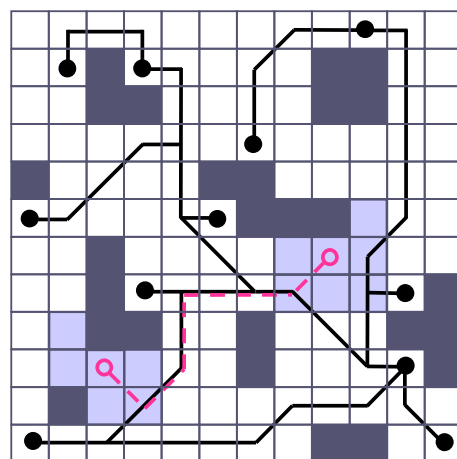

Fig. 9 "Neighbour-intersection" strategy

The problem is that the new solution gained as an adaptation of the most similar case in old solutions can be worse than a new computation that is not based on the stored cases. If a new problem is not very similar to the solutions stored in a database, then we first try to investigate the neighbouring cells of the new start and target. If these neighbours intersect some of the paths stored in a database, then these intersections are used for joining to the old solutions. This is shown in Fig. 9.

VII. Conclusion

In this chapter, we presented a fuzzy logic approach to controlling complex systems and processes where classical PID/PSD controllers cannot be used because their mathematical model is unknown, partially known, strongly nonlinear or in situations in which all the knowledge about the controlled process is given in a linguistic form.

In the last section we studied possible improvements of the cell decomposition method by means of a case-based reasoning approach that takes into consideration the start and target positions of the old solutions.

In the future, we will investigate in more detail other stochastic heuristic techniques for tuning fuzzy rules, and compare the cell decomposition method combined with the case-based reasoning procedure and sophisticated roadmap methods, e.g. generalised Voronoi diagrams.

References

[1] P. Bosc and J. Kacprzyk (eds.), *Fuzziness in Database Management Systems*. Physica-Verlag, Heidelberg, 1995.

[2] A. Boulkroune, M. Tadjine, M. M'Saad, M. Farza, "Fuzzy Adaptive Controller for MIMO Nonlinear Systems with Known and Unknown Control Direction," *Fuzzy Sets and Systems*, vol. 161, no. 6, pp. 797-820, 2010.

[3] R. Hampel, M. Wagenknecht and N. Chaker, *Fuzzy Control*. Physica-Verlag, Heidelberg, 2000.

[4] R.L. Haupt and S.E. Haupt, *Practical Genetic Algorithms*, John Wiley & Sons, New Jersey, 2004.

[5] G. Huang.and & A.L. Dexter, "Realization of Robust Nonlinear Model Predictive Control by Offline Optimisation," *Journal of Process Control*, 8 pp., 2007.

[6] J. Jantzen, *Foundations of Fuzzy Control*. John Wiley & Sons, New York, 2007.

[7] J. Jantzen, Tutorial on Fuzzy Logic. Technical Report No. 98-E 864 (logic), Technical University of Denmark, Department of Automation, Lyngby, 1998.

[8] J. Jantzen, Design of Fuzzy Control. Technical Report No. 98-E 864 (design), Technical University of Denmark, Department of Automation, Lyngby, 1998

[9] J. Jantzen, Tuning of Fuzzy Controllers. Technical Report No. 98-H 871 (fpid), Technical University of Denmark, Department of Automation, Lyngby, 1998.

[10] Y.-T. Juang, Y.-T. Chang and C-P. Huang, "Design of Fuzzy PID Controllers Using Modified Triangular Membership Functions," *Information Sciences*, vol. 178, pp. 1325-1333, 2008.

[11] F. Jurado and M. Valverde, "Genetic Fuzzy Control Applied to the Inverter of Solid Oxide Fuel Cell for Power Quality Improvement," *Electric Power Systems Research*, vol. 76, pp. 93-105, 2005.

[12] D.H. Kim, A. Abraham and J.H. Cho, "A Hybrid Genetic Algorithm and Bacterial Foraging Approach for Global Optimization," *Information Sciences*, vol. 177, pp. 3918-3937, 2007.

[13] G.J. Klir, B. Yuan, *Fuzzy Sets and Fuzzy Logic. Theory and Applications*. Prentice Hall, New Jersey, 1995.

[14] M. Kruusmaa and J. Willemson, "Covering the Path Space: A Casebase Analysis for Mobile Robot Path Planning," *Knowledge-Based Systems*, vol.16, pp. 235-242, 2003.

[15] Y.-G. Leu, "Mean-based Fuzzy Identifier and Control of Uncertain Nonlinear Systems," *Fuzzy Sets and Systems*, vol. 161, no. 6, pp. 837-858, 2010.

[16] S.P. Moustakidis, G.A. Rovithakis and J.B. Theocharis, "An Adaptive Neuro-Fuzzy Tracking Control for Multi-Input Nonlinear Dynamic systems," *Automatica*, 8 pp., 2007.

[17] V. Novák, *Fundamentals of Fuzzy Modelling* (in Czech). BEN – technická literatura, Praha, 2000.

[18] P. Pivoňka, "Fuzzy PI/PD/PID controllers (in Czech)," *Automatizace*, vol. 41, no. 4, 1998.

[19] M. Šeda, "A Comparison of Roadmap and Cell Decomposition Methods in Robot Motion Planning," *WSEAS Transactions on Systems and Control*, vol. 2, no. 2, pp. 101-108, 2007.

[20] K. Tanakay, and H.O. Wang, *Fuzzy Control Systems*. Design and Analysis. John Wiley & Sons, New York, 2001.

[21] G. Yan and L.L. Zhou, "Integrated Fuzzy Logic and Genetic Algorithms for Multi-objective Control of Structures Using MR Dampers,"*Journal of Sound and Vibration*, vol. 296, pp. 368-382, 2006.

[22] L.A. Zadeh, "Is There a Need for Fuzzy Logic?," *Information Sciences*, 29 pp., 2008.

[23] H. Zhang and D. Liu, *Fuzzy Modelling and Fuzzy Control*. Birkhäuser, Boston, 2006.