# Using fuzzy rule interpolation based automata for controlling navigation and collision avoidance behaviour of a robot

Dávid Vincze*, Szilveszter Kovács*

* Department of Information Technology, University of Miskolc, Miskolc, Hungary
david.vincze@iit.uni-miskolc.hu
szkovacs@iit.uni-miskolc.hu

*Abstract*—**Relatively few Fuzzy Rule Interpolation (FRI) techniques can be found among the practical fuzzy rule based applications. Many of them have limitations from the direct application point of view, for example they can be applied only in one dimensional case, or defined based on the two closest surrounding rules of the actual observation. Additionally the FRI methods can dramatically simplify the building of fuzzy rule bases by enabling the application of sparse rule bases. FRI methods can provide reasonable (interpolated) conclusions even if none of the existing rules fires under the current observation. These methods can help the expert to concentrate on the cardinal actions only. Compared to the classical fuzzy CRI, by omitting the derivable rules, the number of the fuzzy rules needed to be handled during the design process could be dramatically reduced. This paper provides a brief overview of several FRI methods and in more detailed an application oriented simple and quick FRI method "FIVE" will be introduced. For the demonstration of the benefits of the interpolation-based fuzzy reasoning as systematic approach, a robot navigation application is presented, where the robot is able to cycle through waypoints while avoiding collision with obstacles and walls. All the controlling parts were accomplished with fuzzy rule bases of the "FIVE" FRI method.**

## I. Introduction

Traditional fuzzy reasoning methods (e.g. the Zadeh-Mamdani compositional rule of inference (CRI)) are demanding "complete rule bases", and hence the construction of a classical rule base requires a special care for filling all the possible rules. A fuzzy rule base is called sparse or incomplete when an existing observation does not hit any of the rules in the rule base. Accordingly there can be observations, where no conclusion can be gained, and having no conclusion in a fuzzy control structure is hard to explain. For example, a solution could be to use the last real conclusion instead of the missing one, but applying historical data automatically to fill unintentionally missing rules could cause unpredictable side effects. Another solution for the same problem is the application of fuzzy rule interpolation (FRI) methods, where the derivable rules are intentionally missing, as FRI methods are capable of providing reasonable (interpolated) conclusions even if none of the existing rules fires under the current observation. The rule base of an FRI controller is not complete by necessity, therefore it could contain the most significant fuzzy rules alone without risking the chance of having no conclusion for some of the observations. In this case a considerable amount of unnecessary work can be avoided during the rule bases set up. On the other hand most of the FRI methods are sharing the burden of high computational demand, e.g. the task of searching for the two closest surrounding rules to the observation, and calculating the conclusion at least in some characteristic α-cuts. Moreover in some methods the interpretability of the fuzzy conclusion gained is also not straightforward [7]. Even if there have been a lot of efforts to rectify the interpretability of the interpolated fuzzy conclusion [17]. In [1] Baranyi *et al.* give a comprehensive overview of the recent existing FRI methods. Beyond these problems, some of the FRI methods are originally defined for one dimensional input space, and need special extension for the multidimensional case (e.g. [2]-[3]). In [22] Wong *et al.* gave a comparative overview of the multidimensional input space capable FRI methods. In [2] Jenei introduced a way for axiomatic treatment of the FRI methods. In [14] Johanyák *et al.* introduces an automatic way for direct sparse fuzzy rule base generation based on given input-output data. The high computational demand, mainly the search for the two closest surrounding rules to an arbitrary observation in the multidimensional antecedent space makes many of these methods hardly suitable for real-time applications. Some FRI methods, (e.g. the method introduced by Jenei *et al.* in [3], or FRIPOC [25], LESFRI [26], VEIN [27]), eliminate the search for the two closest surrounding rules by taking all the rules into consideration, hence speeding up the reasoning process. On the other hand, keeping the goal of constructing fuzzy conclusion, and not simply speeding up the reasoning process, they still require some additional (or repeated) computational steps for the elements of the level set (or at least some relevant α levels). A rather different application oriented aspect of the FRI emerges in the concept of "FIVE" (Fuzzy Interpolation based on Vague Environment). In the followings the method "FIVE" will be introduced in more details.

## II. A Brief Overview of Several FRI Techniques

One of the first FRI techniques was published by Kóczy and Hirota [5]. It is usually referred as *KH method*. It is applicable to convex and normal fuzzy (CNF) sets. It determines the conclusion by its $\alpha$-cuts in such a way that the ratio of distances between the conclusion and the consequents should be identical with

978-1-4244-2875-5/08/$25.00 ©2008 IEEE

the ones between the observation and the antecedents for all important $\alpha$-cuts. The applied formula

$$d(A^*, A_1) : d(A^*, A_2) = d(B^*, B_1) : d(B^*, B_2),$$

can be solved for $B^*$ for relevant $\alpha$-cuts after decomposition.

It is shown in, e.g. in [7], [8] that the conclusion of the KH method is not always directly interpretable as fuzzy set. This drawback motivated many alternative solutions. A modification was proposed by Vass, Kalmár and Kóczy [20] (*VKK method*), where the conclusion is computed based on the distance of the centre points and the widths of the α-cuts, instead of lower and upper distances. VKK method decreases the applicability limit of KH method, but does not eliminate it completely. The technique cannot be applied if any of the antecedent sets is singleton (the width of the antecedent's support must be nonzero). In spite of the disadvantages, KH is popular because its simplicity that infers its advantageous complexity properties. It was generalized in several ways. Among them the *stabilized KH interpolator* is emerged, as it is proved to hold the universal approximation property [19], [16]. This method takes into account all flaking rules of an observation in the calculation of the conclusion in extent to the inverse of the distance of antecedents and observation. The universal approximation property holds if the distance function is raised to the power of the input's dimension.

Another modification of KH is the modified alpha-cut based interpolation (*MACI*) method [17], which alleviates completely the abnormality problem. MACI's main idea is the following: it transforms fuzzy sets of the input and output universes to such a space where abnormality is excluded, then computes the conclusion there, which is finally transformed back to the original space. MACI uses vector representation of fuzzy sets and originally applicable to CNF sets [23]. These latter conditions (CNF sets) can be relaxed, but it increases the computational need of the method considerably [18]. MACI is one of the most applied FRI methods [22], since it preserves advantageous computational and approximate nature of KH, while it excludes its abnormality.

Another fuzzy interpolation technique was proposed by Kóczy *et al.* [6]. It is called conservation of "relative fuzziness" (*CRF*) method, which notion means that the left (right) fuzziness of the approximated conclusion in proportion to the flanking fuzziness of the neighboring consequent should be the same as the (left) right fuzziness of the observation in proportion to the flanking fuzziness of the neighboring antecedent. The technique is applicable to CNF sets. An improved fuzzy interpolation technique for multidimensional input spaces (*IMUL*) was proposed in [21], and described in details in [22]. IMUL applies a combination of CRF and MACI methods, and mixes advantages of both. The core of the conclusion is determined by MACI method, while its flanks by CRF. The main advantages of this method are its applicability for multi-dimensional problems and its relative simplicity.

Conceptually different approaches were proposed by Baranyi *et al* [1] based on the relation and on the semantic and inter-relational features of the fuzzy sets. The family of these methods applies "General Methodology" (GM); this notation also reflects to the feature that these methods are able to process arbitrary shaped fuzzy sets. The basic concept is to calculate the reference point of the conclusion based on the ratio of the distances between the reference points of the observation and the antecedents. Then, a single rule reasoning method (revision function) is applied to determine the final fuzzy conclusion based on the similarity of the fuzzy observation and an "interpolated" observation. Several methods follow the two-step idea of the GM. For example FRIPOC (Johanyák and Kovács [25]) extends the range of the applicable membership functions types by introducing the concept of polar cuts using a polar coordinate system, LESFRI (Johanyák and Kovács [26]) preserves the characteristic shape type of the antecedent and consequent partitions by applying the method of least squares, and VEIN [27] solves the task of rule interpolation in the vague environment by the help of the set interpolation method VESI proposed by Johanyák in [12].

## III. Fuzzy Interpolation based on Vague Environment

In the demonstrative example of the paper the FRI method "FIVE" (Fuzzy Interpolation based on Vague Environment) was chosen, because it is an application oriented, i.e. it is quick and simple, and hence it can be easily embedded into a direct robot control. In the concept of "FIVE" an application oriented aspect of the fuzzy rule interpolation emerges. It was originally introduced in [9], [10] and [11]) and it was developed to fit the speed requirements of direct fuzzy control, where the conclusions of the fuzzy controller are applied directly as control actions in a real-time system.

The main idea of the FIVE is based on the fact that most of the control applications serves crisp observations and requires crisp conclusions from the controller. Adopting the idea of the vague environment (VE) [4], FIVE can handle the antecedent and consequent fuzzy partitions of the fuzzy rule base by scaling functions [4] and therefore turn the fuzzy interpolation to crisp interpolation. The idea of a VE is based on the similarity (in other words: indistinguishability) of the considered elements. In VE the fuzzy membership function $\mu_A(x)$ is indicating level of similarity of x to a specific element a that is a representative or prototypical element of the fuzzy set $\mu_A(x)$, or, equivalently, as the degree to which x is indistinguishable from a [4]. Therefore the α-cuts of the fuzzy set $\mu_A(x)$ are the sets which contain the elements that are $(1\text{-}\alpha)$-indistinguishable from $a$. Two values in a VE are ε-distinguishable if their distance is greater than ε. The distances in a VE are weighted distances. The weighting factor or function is called scaling function (factor) [4]. If VE of a fuzzy partition (the scaling function or at least the approximate scaling function [9], [11]) exists, the member sets of the fuzzy partition can be characterized by points in that VE (see e.g. scaling function $s$ on Fig. 1). Therefore any crisp interpolation, extrapolation, or regression method can be adapted very simply for FRI [9], [11]. Because of its simple multidimensional applicability, in FIVE the Shepard operator based interpolation (first introduced in [15]) is adapted (see e.g. Fig. 1).

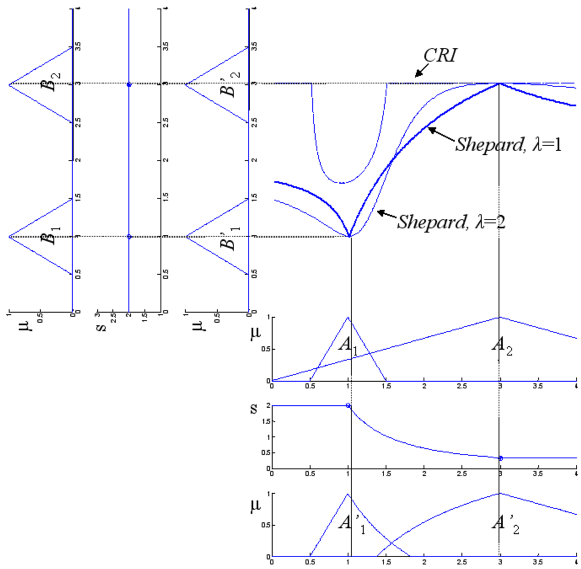


Fig.1: Interpolation of two fuzzy rules ($R_i$: $A_i \rightarrow B_i$), by the Shepard operator based *FIVE*, and for comparison the min-max *CRI* with COG defuzzification.

The code of the FIVE FRI together with other FRI methods as a freely available as a MatLab FRI Toolbox [24], and it can be downloaded from [28] and [29].

## IV. Application Example: Room Surveillance with Collision Avoidance

The example application of the paper is a room surveillance navigation control of a mobile robot. The goal of the navigation is to control an unmanned robot capable of room surveillance by cycling through given waypoints within a room (exploration) with walls and moving obstacles avoidance.

When the way of the robot seems to be blocked by an obstacle or by a wall, then the robot is capable of turning around and head in the opposite direction as a last resort. The order of the waypoints is a fixed sequence. The example configuration has four waypoints which correspond to the four corners of the room, where the shape of the room is a rectangle having a 4:3 side to side ratio (see Fig. 2.).

The navigation control is built from three separate controlling components: the *selection of the next waypoint to approach*, the *wall and obstacle avoidance*, and the *heading direction change*.

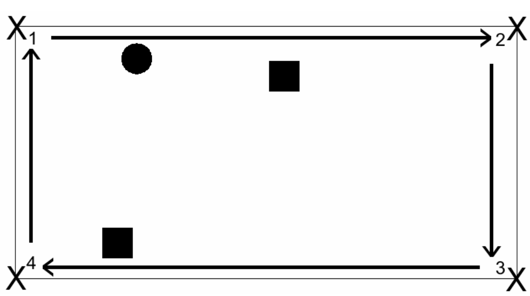


Fig.2: The room with the waypoints and obstacles where the robot navigates. The circle shaped object symbolizes the robot, and the two rectangles are the obstacles.

The *selection of the next waypoint to approach component* is determined according to the followings: the current position of the robot (described by the distances from the four waypoints), the heading weights of the four waypoints, the need for direction changing and the current direction of the robot. The method for the selection is simple: assign the waypoint in the predefined sequence which follows the nearest waypoint to the robot. The observations needed for this component are the measured distances of the robot from each of the defined waypoints ($dw_1$, $dw_2$, $dw_3$, $dw_4$) and the selection state, which waypoints ($sw_1$, $sw_2$, $sw_3$, $sw_4$) is the robot heading towards. States characterize that whether the waypoint is actively selected or not. For expressing the distance from an arbitrary waypoint in the fuzzy rule base, the linguistic terms for the antecedent universes are given as the following: zerus (*Z*), large (*L*) and for expressing the state of the heading waypoint, there are only two antecedent linguistic terms: true (*T*), false (*F*). The consequent of the rule base expressing the weight of the selected next waypoint (*WW*) it has only two linguistic terms: zerus (*Z*), large (*L*). The selection of the next waypoint to approach component has as many separate rule bases as the count of the pre-defined waypoints, and they have similar structure. In the example case this means four rule bases (having four waypoints), each needs to be evaluated with the same measured distances and state variables. Every conclusion is a normalized weight which is used to scale a vector pointing towards the corresponding waypoint. When these four scaled vectors are summarized the result will be the movement vector of the *exploration controller* component.

Having the required observations and the strategy described above, the fuzzy rule bases can be constructed. As mentioned, four rule bases are required in this particular case: first to calculate the weight needed to take the robot towards the second waypoint (see Table I), second to direct the robot to the third waypoint (see Table II), third to take the robot to the fourth waypoint (see Table III), and a fourth rule base to navigate the robot back to the first waypoint (see Table IV).

TABLE I.
First Waypoint Selection Weight Rule Base

| RW1 | $dw_1$ | $dw_2$ | $dw_3$ | $dw_4$ | $sw_1$ | $sw_2$ | $sw_3$ | $sw_4$ | *WW* |
|---|---|---|---|---|---|---|---|---|---|
| Rule 1 | Z | | | | | | | | Z |
| Rule 2 | L | | | | T | | | | L |
| Rule 3 | | L | | | | T | | | Z |
| Rule 4 | | | | L | | | | T | Z |
| Rule 5 | | | L | | | | T | | Z |
| Rule 6 | | | | Z | | | | | L |

TABLE II.
Second Waypoint Selection Weight Rule Base

| RW2 | $dw_1$ | $dw_2$ | $dw_3$ | $dw_4$ | $sw_1$ | $sw_2$ | $sw_3$ | $sw_4$ | *WW* |
|---|---|---|---|---|---|---|---|---|---|
| Rule 1 | | Z | | | | | | | Z |
| Rule 2 | | L | | | | T | | | L |
| Rule 3 | | | L | | | | T | | Z |
| Rule 4 | L | | | | T | | | | Z |
| Rule 5 | | | | L | | | | T | Z |
| Rule 6 | Z | | | | | | | | L |

TABLE III.
THIRD WAYPOINT SELECTION WEIGHT RULE BASE

| RW3 | $dw_1$ | $dw_2$ | $dw_3$ | $dw_4$ | $sw_1$ | $sw_2$ | $sw_3$ | $sw_4$ | *WW* |
|---|---|---|---|---|---|---|---|---|---|
| Rule 1 | | | Z | | | | | | Z |
| Rule 2 | | | L | | | | T | | L |
| Rule 3 | | | | L | | | | T | Z |
| Rule 4 | | L | | | | T | | | Z |
| Rule 5 | L | | | | T | | | | Z |
| Rule 6 | | Z | | | | | | | L |

TABLE IV.
FOURTH WAYPOINT SELECTION WEIGHT RULE BASE

| RW4 | $dw_1$ | $dw_2$ | $dw_3$ | $dw_4$ | $sw_1$ | $sw_2$ | $sw_3$ | $sw_4$ | *WW* |
|---|---|---|---|---|---|---|---|---|---|
| Rule 1 | | | | Z | | | | | Z |
| Rule 2 | | | | L | | | | T | L |
| Rule 3 | L | | | | T | | | | Z |
| Rule 4 | | | L | | | | T | | Z |
| Rule 5 | | L | | | | T | | | Z |
| Rule 6 | | | Z | | | | | | L |

The rules are defined in the following form:

$RW_i$:

**If** $dw_1 = A_{1,i}$ **and** $dw_2 = A_{2,i}$ **and** $dw_3 = A_{3,i}$ **and** $dw_4 = A_{4,i}$ **and** $sw_1 = A_{5,i}$ **and** $sw_2 = A_{6,i}$ **and** $sw_3 = A_{7,i}$ **and** $sw_4 = A_{8,i}$ **Then** $WW = B_i$

The rules in the first rule base (see Table I) have the meanings as follows: The first rule means that when the corresponding waypoint (first) is reached by the robot then that waypoint (first) should be abandoned, hence the weight of the waypoint (first) will be zerus (*Z*). The second rule keeps the robot coming to the waypoint (first) if it has been selected earlier. The third rule serves the purpose of keeping down the weight when the robot is going to the next (second) waypoint, so do the fourth and fifth rules, but for the remaining two waypoints (fourth and third respectively). The sixth, last rule means that when the robot reached the previous waypoint in the sequence (fourth), it should head to this very waypoint (first).

With the above described rule bases the robot can cycle around the given waypoints, but when obstacles stand in its way further rule bases are required to handle the situation.

The applied collision avoidance strategy consists of two parts: *wall avoidance* and *obstacle avoidance*. By the definition walls are the borders of the room and the obstacles are objects which can move freely inside the room.

Avoiding walls is a simple procedure. Based on the distance from the four walls, a repulsion rate is calculated, which then can be used to compute a vector perpendicular to the corresponding wall. Observations of the *wall and obstacle avoidance component* are the measured distances from each of the walls ($d_w$), and the measured distances from each of the objects inside the room ($d_o$). Only one coordinate is sufficient to calculate the distance between the robot and a wall: difference of the horizontal coordinates for vertical walls, and the difference of the vertical coordinates for horizontal walls. The linguistic terms of the antecedent universes are: zerus (*Z*), small (*S*), medium (*M*), large (*L*), and for the consequent universe (*AV*): zerus (*Z*), small (*S*), large (*L*). Obstacle avoidance follows the same strategy. Summarizing the normalized wall and obstacle avoidance repulse vectors the result can overrun the maximum. In this case the length of the vector should be cut to the maximum allowed value.

Based on the above described technique a simple fuzzy rule base can be built (see Table V). The *wall and obstacle avoidance component* uses the same rule base for all the required conclusions only the input distances differ within every evaluation. The rules are defined in the following form:

$RColl_i$:

**If** $d_w = A_i$ **Then** $AV = B_i$

TABLE V.
WALL AND OBSTACLE AVOIDANCE WEIGHT RULE BASE

| RColl | $d_w$, $d_o$ | *AV* |
|---|---|---|
| Rule 1 | Z | L |
| Rule 2 | S | S |
| Rule 3 | M | Z |
| Rule 4 | L | Z |

In the case if the way of the robot seems to be blocked in the current exploration direction, the robot can change its heading, by assigning the waypoints in the reverse order. This direction change decision is made by the *heading direction change component*.

The observations needed for this component (see Fig.3) are the sum of movement rate of the robot and the collision avoidance vector (*mr*), the summarized rate of the length of the wall and obstacle avoidance vectors (*ar*), and finally a rate of exploration is added (*er*). It serves as a movement component weight. Since the robot could do some other types of movements than exploring (e.g. directly heading towards a specified waypoint, maybe to the exit), and because probably this is the only component requires direction changing decision, an exploration rate value should be also considered.

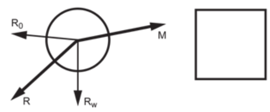


Fig.3: The repulsion vectors of the obstacle ($R_o$), the wall ($R_w$) and their sum (R). M is the movement vector of the robot towards the next waypoint.

The linguistic terms of the two antecedent universes of the *heading direction change component* are: zerus (*Z*) and large (*L*). The conclusion universe (*DC*), which tells whether to change the direction of the robot, the linguistic terms are: false (*F*) and true (*T*). The rule base consists only of three rules, which can be seen in Table VI. The rules are defined in the following form:

$RDirCh_i$:

**If** $er = A_{1,i}$ **and** $mr = A_{2,i}$ **and** $ar = A_{3,i}$ **Then** $DC = B_i$

TABLE VI.
DIRECTION CHANGING DECISION RULE BASE

| RDirCh | *er* | *mr* | *ar* | *DC* |
|---|---|---|---|---|
| Rule 1 | Z | | | F |
| Rule 2 | L | Z | L | T |
| Rule 3 | L | L | | F |

Another rule base can be used to determine the new heading direction for the robot. For this subcomponent two observations are required: a value which tells whether a direction heading change is necessary (*dirchg*) (this is the conclusion above, see Table VI.) and the current heading direction (*currdir*). The linguistic terms for the antecedent universes are the following: for expressing the need of direction changing: true (*T*), false (*F*), for expressing the current direction and also for the consequent universe, which gives the new direction: clockwise (*C*), counter clockwise (*CC*). The rule base is shown on Table VII, and the rules can be interpreted according to the following form:

$\text{RNewDir}_i$:

**If** *dirchg* = $A_{1,i}$ **and** *currdir* = $A_{2,i}$

**Then** *ND* = $B_i$

Having the rule bases for collision avoidance, direction changing decision and new heading direction, the original waypoint selection rule bases (Table I. - Table IV.) should be extended. New observations will be added: the current heading direction (*dir*) and a parameter expressing whether the heading direction was changed (*dirchg*). The newly added antecedent linguistic terms for the necessity of reversing the direction are: true (*T*), false (*F*). For the current direction: clockwise (*C*), counter clockwise (*CC*). The extended rule bases are shown on Table VIII. - Table XI. The rules are defined in the following form:

$\text{RWX}_i$:

**If** $dw_1 = A_{1,i}$ **and** $dw_2 = A_{2,i}$ **and** $dw_3 = A_{3,i}$ **and** $dw_4 = A_{4,i}$ **and** $sw_1 = A_{5,i}$ **and** $sw_2 = A_{6,i}$ **and** $sw_3 = A_{7,i}$ **and** $sw_4 = A_{8,i}$ **and** $dir = A_{9,i}$ **and** $dirch = A_{10,i}$

**Then** $WW = B_i$

Compared to the original waypoint selection rule base, four new rules were added. In the following the new rules are explained, according to the first rule base (see Table VIII.). The first two rules serve the same purpose as in the original rule base. Rule 3 stops the robot when a direction change is necessary. The fourth rule changes the direction if needed and if the previous heading was towards the next waypoint in the defined sequence (second in this particular case). The fifth rule is similar to the fourth one, it changes the direction if required and if the previous heading was the previous waypoint in order (in this case the fourth). Rule 6, 7 and 8 are the same as the third, fourth and fifth rule in the original waypoint selection rule base. Rule 9 means that when the robot reaches the previous waypoint in the sequence (fourth), it should head to the waypoint (first). The last rule is very similar to Rule 9, but for the opposite heading direction.

TABLE VII.
SELECTION OF CURRENT DIRECTION DECISION RULE BASE

| RNewDir | *dirchg* | *currdir* | *ND* |
|---|---|---|---|
| Rule 1 | F | C | C |
| Rule 2 | F | CC | CC |
| Rule 3 | T | C | CC |
| Rule 4 | T | CC | C |

TABLE VIII.
FIRST WAYPOINT SELECTION WEIGHT WITH DIRECTION CHANGING RULE BASE

| RW1 | $dw_1$ | $dw_2$ | $dw_3$ | $dw_4$ | $sw_1$ | $sw_2$ | $sw_3$ | $sw_4$ | *dir* | *dirch* | *WW* |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Rule 1 | Z | | | | | | | | | | Z |
| Rule 2 | L | | | | T | | | | | F | L |
| Rule 3 | L | | | | T | | | | | T | Z |
| Rule 4 | | | L | L | | T | | | CC | T | L |
| Rule 5 | | L | L | | | | | T | C | T | L |
| Rule 6 | | L | | | | T | | | | F | Z |
| Rule 7 | | | | L | | | | T | | F | Z |
| Rule 8 | | | L | | | | T | | | | Z |
| Rule 9 | | | | Z | | | | | C | F | L |
| Rule 10 | | Z | | | | | | | CC | F | L |

TABLE IX.
SECOND WAYPOINT SELECTION WEIGHT WITH DIRECTION CHANGING RULE BASE

| RW2 | $dw_1$ | $dw_2$ | $dw_3$ | $dw_4$ | $sw_1$ | $sw_2$ | $sw_3$ | $sw_4$ | *dir* | *dirch* | *WW* |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Rule 1 | | Z | | | | | | | | | Z |
| Rule 2 | | L | | | | T | | | | F | L |
| Rule 3 | | L | | | | T | | | | T | Z |
| Rule 4 | L | | | L | | | T | | CC | T | L |
| Rule 5 | | | L | L | T | | | | C | T | L |
| Rule 6 | | | L | | | | T | | | F | Z |
| Rule 7 | L | | | | T | | | | | F | Z |
| Rule 8 | | | | L | | | | T | | | Z |
| Rule 9 | Z | | | | | | | | C | F | L |
| Rule 10 | | | Z | | | | | | CC | F | L |

TABLE X.
THIRD WAYPOINT SELECTION WEIGHT WITH DIRECTION CHANGING RULE BASE

| RW3 | $dw_1$ | $dw_2$ | $dw_3$ | $dw_4$ | $sw_1$ | $sw_2$ | $sw_3$ | $sw_4$ | *dir* | *dirch* | *WW* |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Rule 1 | | | Z | | | | | | | | Z |
| Rule 2 | | | L | | | | T | | | F | L |
| Rule 3 | | | L | | | | T | | | T | Z |
| Rule 4 | L | L | | | | | | T | CC | T | L |
| Rule 5 | L | | | L | | T | | | C | T | L |
| Rule 6 | | | | L | | | | T | | F | Z |
| Rule 7 | | L | | | | T | | | | F | Z |
| Rule 8 | L | | | | T | | | | | | Z |
| Rule 9 | | Z | | | | | | | C | F | L |
| Rule 10 | | | | Z | | | | | CC | F | L |

TABLE XI.
FOURTH WAYPOINT SELECTION WEIGHT WITH DIRECTION CHANGING RULE BASE

| RW4 | $dw_1$ | $dw_2$ | $dw_3$ | $dw_4$ | $sw_1$ | $sw_2$ | $sw_3$ | $sw_4$ | *dir* | *dirch* | *WW* |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Rule 1 | | | | Z | | | | | | | Z |
| Rule 2 | | | | L | | | | T | | F | L |
| Rule 3 | | | | L | | | | T | | T | Z |
| Rule 4 | | L | L | | T | | | | CC | T | L |
| Rule 5 | L | L | | | | | T | | C | T | L |
| Rule 6 | L | | | | T | | | | | F | Z |
| Rule 7 | | | L | | | | T | | | F | Z |
| Rule 8 | | L | | | | T | | | | | Z |
| Rule 9 | | | Z | | | | | | C | F | L |
| Rule 10 | Z | | | | | | | | CC | F | L |

It is practical to arrange the evaluation of these rule bases and observation calculations in a loop. First the waypoint selection conclusions should be calculated, the result vector should be added to the current position of the robot. With this new position the distances from the walls and obstacles should be computed, then the wall and obstacle avoidance fuzzy rule bases should be evaluated, these results should be summarized with the current position. This will be the next valid position of the robot. Finally we have all the required data to get the conclusion for direction changing. If the direction has to be changed, the direction state variable should be inverted and in the next iteration it should take effect. Following this procedure gives a working model of surveillance navigation and collision avoidance.

## V. CONCLUSION

Applying the "FIVE" fuzzy rule interpolation method and sparse fuzzy rule bases described as an example of this paper, a room surveillance navigation control strategy was implemented [28]. Compared to the classical complete fuzzy rule base solutions, the main benefit of this approach is the reduced rule base size. For building a complete fuzzy rule base with the same strategies, $2^{(2n+2)}+8+4+4$ (where $n$ is number of the defined waypoints) fuzzy rules would be needed, which is 1040 with the four waypoints of the given example. But the fuzzy rule interpolation and sparse fuzzy rule base solution of this paper has only $n*(6+n)+3+4+4$, which is only 51 with four waypoints. This rule base size is easily implementable even in embedded FRI fuzzy logic controllers. Hence the main conclusion in the paper, that there are application areas, where FRI methods and the corresponding sparse rule bases turns strategies to be tractable sizes even with numerous input dimensions.

## REFERENCES


[1] P. Baranyi, L. T. Kóczy, and Gedeon, T. D., "A Generalized Concept for Fuzzy Rule Interpolation", IEEE Trans. on Fuzzy Systems, vol. 12, No. 6, 2004, pp 820-837.

[2] S. Jenei, "Interpolating and extrapolating fuzzy quantities revisited – an axiomatic approach", Soft Comput., vol. 5., 2001, 179-193.

[3] S. Jenei, E. P. Klement and R. Konzel, "Interpolation and extrapolation of fuzzy quantities – The multiple-dimensional case", Soft Comput., vol. 6., 2002, 258-270.

[4] F. Klawonn, "Fuzzy Sets and Vague Environments", Fuzzy Sets and Systems, 66, 1994, pp. 207-221.

[5] L. T. Kóczy and K. Hirota, "Rule interpolation by α-level sets in fuzzy approximate reasoning", In J. BUSEFAL, Automne, URA-CNRS. Vol. 46. Toulouse, France, 1991, pp. 115-123.

[6] L.T. Kóczy, K. Hirota, and T. D. Gedeon, "Fuzzy rule interpolation by the conservation of relative fuzziness", Technical Report TR 97/2. Hirota Lab, Dept. of Comp. Int. and Sys. Sci., Tokyo Inst. of Techn., Yokohama, 1997.

[7] L. T. Kóczy and Sz. Kovács, "On the preservation of the convexity and piecewise linearity in linear fuzzy rule interpolation", Tokyo Inst. Technol., Yokohama, Japan, Tech. Rep. TR 93-94/402, LIFE Chair Fuzzy Theory, 1993.

[8] L. T. Kóczy, and Sz. Kovács, "Shape of the Fuzzy Conclusion Generated by Linear Interpolation in Trapezoidal Fuzzy Rule Bases," in Proceedings of the 2nd European Congress on Intelligent Techniques and Soft Computing, Aachen, 1994, pp. 1666–1670.

[9] Sz. Kovács, "New Aspects of Interpolative Reasoning", Proceedings of the 6th. International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, Granada, Spain, 1996, pp. 477-482.

[10] Sz. Kovács, and L.T. Kóczy, "Approximate Fuzzy Reasoning Based on Interpolation in the Vague Environment of the Fuzzy Rule base as a Practical Alternative of the Classical CRI", Proceedings of the 7th International Fuzzy Systems Association World Congress, Prague, Czech Republic, 1997, 144-149.

[11] Sz. Kovács, and L.T. Kóczy, "The use of the concept of vague environment in approximate fuzzy reasoning", Fuzzy Set Theory and Applications, Tatra Mountains Mathematical Publications, Mathematical Institute Slovak Academy of Sciences, Bratislava, Slovak Republic, vol.12, 1997, pp. 169-181.

[12] Johanyák, Zs. Cs.: Vague Environment Based Set Interpolation, A GAMF Közleményei, Kecskemét, XXI. évfolyam (2006-2007), ISSN 0230-6182, pp. 33-44.

[13] Sz. Kovács and L. T. Kóczy, "Application of an approximate fuzzy logic controller in an AGV steering system, path tracking and collision avoidance strategy", Fuzzy Set Theory and Applications, In Tatra Mountains Mathematical Publications, Mathematical Institute Slovak Academy of Sciences, vol.16, Bratislava, Slovakia, 1999, pp. 456-467.

[14] Johanyák, Zs. Cs., Parthiban, R, and Sekaran, G.: Fuzzy Modeling for an Anaerobic Tapered Fluidized Bed Reactor, SCIENTIFIC BULLETIN of "Politehnica" University of Timisoara, ROMANIA, Transactions on AUTOMATIC CONTROL and COMPUTER SCIENCE, ISSN 1224-600X, Vol: 52(66) No: 2 / 2007, pp. 67-72.

[15] D. Shepard, "A two dimensional interpolation function for irregularly spaced data", Proc. 23rd ACM Internat. Conf., 1968, pp. 517-524.

[16] D. Tikk: Notes on the approximation rate of fuzzy KH interpolator. Fuzzy Sets and Systems, 138(2), Sept., 2003, pp. 441–453.

[17] D. Tikk and P. Baranyi, "Comprehensive analysis of a new fuzzy rule interpolation method", In IEEE Trans. Fuzzy Syst., vol. 8, No. 3, June, 2000, pp. 281-296.

[18] D. Tikk, P. Baranyi, T. D. Gedeon, and L. Muresan, "Generalization of a rule interpolation method resulting always in acceptable conclusion", Tatra Mountains Math. Publ., 21, 2001, pp. 73–91.

[19] D. Tikk, I. Joó, L. T. Kóczy, P. Várlaki, B. Moser, and T. D. Gedeon, "Stability of interpolative fuzzy KH-controllers", Fuzzy Sets and Systems, 125(1), January, 2002, pp. 105–119.

[20] G. Vass, L. Kalmár, and L. T. Kóczy, "Extension of the fuzzy rule interpolation method", in Proc. Int. Conf. Fuzzy Sets Theory Applications (FSTA'92), Liptovsky M., Czechoslovakia, 1992, pp. 1-6.

[21] K. W. Wong, T. D. Gedeon, and D. Tikk: "An improved multidimensional α-cut based fuzzy interpolation technique", In Proc. Int. Conf Artificial Intelligence in Science and Technology (AISAT'2000), Hobart, Australia, 2000, pp. 29–32.

[22] K. W. Wong, D. Tikk, T. D. Gedeon, and L. T. Kóczy, "Fuzzy Rule Interpolation for Multidimensional Input Spaces With Applications", IEEE Transactions on Fuzzy Systems, ISSN 1063-6706, Vol. 13, No. 6, December, 2005, pp. 809-819.

[23] Y. Yam, and L. T. Kóczy, "Representing membership functions as points in high dimensional spaces for fuzzy interpolation and extrapolation", Dept. Mech. Automat. Eng., Chinese Univ. Hong Kong, Tech. Rep.CUHK-MAE-97-03, 1997.

[24] Zs. Cs. Johanyák, D. Tikk, Sz. Kovács, K. W. Wong: Fuzzy Rule Interpolation Matlab Toolbox – FRI Toolbox, Proc. of the IEEE World Congress on Computational Intelligence (WCCI'06), 15th Int. Conf. on Fuzzy Systems (FUZZ-IEEE'06), July 16--21, Vancouver, BC, Canada, Omnipress. ISBN 0-7803-9489-5, 2006, pp. 1427-1433.

[25] Johanyák, Zs. Cs., Kovács Sz.: Fuzzy Rule Interpolation Based on Polar Cuts, Computational Intelligence, Theory and Applications, Springer Berlin Heidelberg, 2006, pp. 499-511

[26] Z. C. Johanyák, S. Kovács, "Fuzzy Rule Interpolation by the Least Squares Method", 7th International Symposium of Hungarian Researchers on Computational Intelligence (HUCI 2006), November 24-25, 2006 Budapest, pp. 495-506.

[27] Johanyák, Zs. Cs., Kovács, Sz.: Vague Environment-based Two-step Fuzzy Rule Interpolation Method, 5th Slovakian-Hungarian Joint Symposium on Applied Machine Intelligence and Informatics (SAMI 2007), January 25-26, 2007 Poprad, Slovakia, pp. 189-200.

[28] The FRI Toolbox is available at: http://fri.gamf.hu/

[29] Some FRI applications are available at: http://www.iit.uni-miskolc.hu/~szkovacs/