Production Systems and Information Engineering
Volume 5 (2009), pp. 151-166.

BEHAVIOUR BASED CONTROL WITH FUZZY AUTOMATON IN VEHICLE NAVIGATION

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

SZILVESZTER KOVÁCS
University of Miskolc, Hungary
Department of Information Technology,
Technical University of Kosice, Slovakia
Department of Cybernetics and AI
szkovacs@iit.uni-miskolc.hu

[Received January 2009]

Abstract. From the viewpoint of Behaviour based Control many control tasks can be divided into separate behaviour components. By defining the relevant behaviour components, the actual control action can be constructed based on the individual control actions of the component behaviours. In this case the control action is either related to an individual behaviour component or to a fusion of behaviour components based on their relevance to the actual situation. This paper adapts the concept of fuzzy automaton for achieving the decision related to the relevance of the behaviour components in the task of the navigation of an autonomous vehicle. In the structure applied, the relevance of the behaviour components is approximated by a fuzzy rule interpolation (FRI, namely the FIVE method) based fuzzy automaton. The main reason for the FRI application is the state-transition rule-base simplification of the fuzzy automaton. In case of FRI, sparse rule bases (incomplete rule bases) are acceptable, because derivable rules can be omitted intentionally, saving construction time and reducing the complexity of the state-transition rule-base. The paper also provides a brief overview of Behaviour based Control and fuzzy rule interpolation (FRI). For demonstration purposes the paper gives a simple example of state-transition rule-base construction in case of the vehicle navigation task mentioned.

Keywords: behaviour based control, fuzzy automaton, fuzzy rule interpolation, FIVE, vehicle navigation control

1. Introduction

The main building blocks of Behaviour based Control (BBC, a comprehensive overview can be found in [14]) are the behaviour components themselves. The behaviour components can be copies of typical human or animal behaviors, or can be artificially created behaviours. The actual behaviour response of the system can be formed as one of the existing behaviour components, which gives the best match for the actual situation, or a fusion of the behaviour components based on their suitability for the actual situation. Encoding the behaviour components can be realized with simple reflexive agents, which assign an output response to each input situation.

In the case when more than one behaviour components are simultaneously competing for the same actuator an aggregation or selection of the behaviour components is necessary. Handling multiple behaviour components in a BBC system can be done in two ways. The first is the competitive way, when the behaviour components are assigned priorities, and the behaviour component with the highest priority takes precedence, while the behaviours with lower priorities are simply ignored. The second is the cooperative way when the outputs are fused based on various criteria.

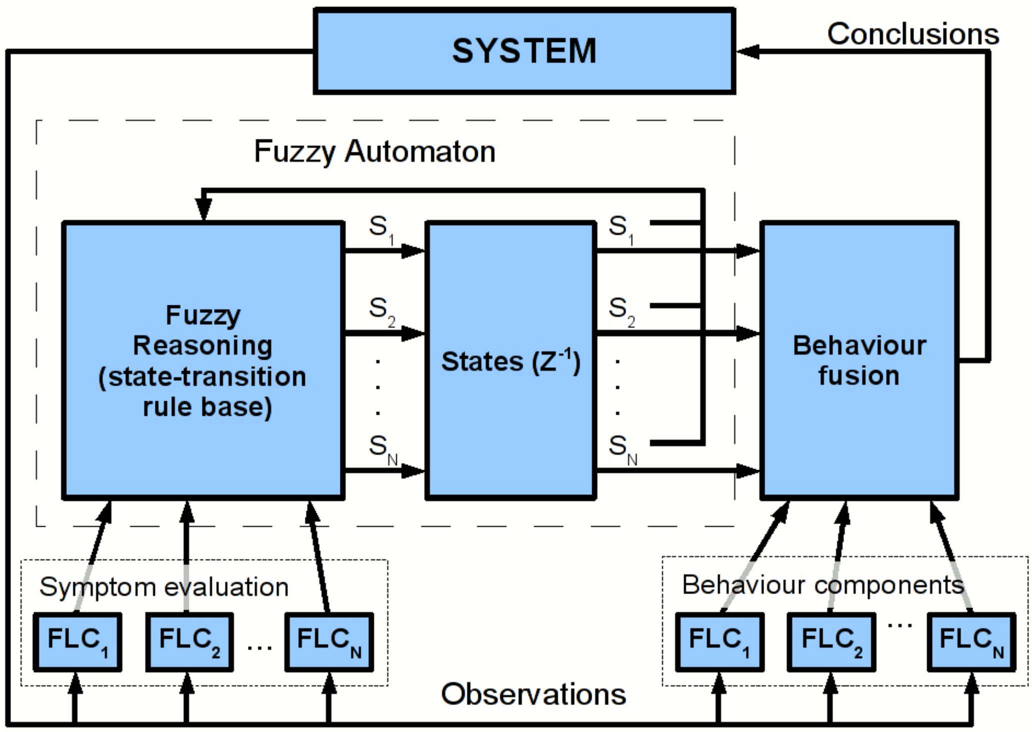

Figure 1. Diagram of the fuzzy automaton

For achieving the decision related to the relevance of the behaviour components this paper adapts the concept of fuzzy automaton. (See the diagram of the suggested fuzzy automaton based system in Fig. 1). The system consists of not only the automaton but the behaviour fusion component and various component behaviours implemented as fuzzy logic controllers (FLC). The state variables characterize the relevance of the component behaviours. The state-transition rule base of the automaton applies fuzzy rule interpolation (namely the FIVE method) for state-transition evaluation. The previous states are fed back to the automaton and the conclusion given by the automaton is used as a weight in the behaviour fusion component for determining the final conclusion of the BBC. The conclusion of the fuzzy automaton will be the new system state for the next step of the behaviour fusion. The behaviour fusion component can also be implemented by fuzzy reasoning (e.g. using fuzzy rule interpolation), or simply as a weighted sum. The symptom evaluation components provide a kind of preprocessing for the automaton based on the observations gathered. These components can also employ FRI techniques.

Embedding fuzzy rule interpolation the model always gives an usable conclusion even if there are no rules defined for the actual observations. Hence the application of sparse rule bases (not complete) can be beneficial, because derivable rules can be omitted intentionally, radically simplifying the rule base creation, saving time-consuming work. The example application of the paper is also based on sparse (not a complete) rule bases. The main reason of applying sparse rule bases and FRI in this case is the simple adaptation of expert knowledge to the system. The existing knowledge is naturally sparse as the experts concentrate on giving the main state-action rules only. On the other hand, having sparse rule bases also helps the final parameter optimization process, as it has usually fewer tunable parameters than complete rule bases.

In the next section, the FRI method FIVE will be introduced in more detail, as it is a quick and simple FRI method. It has the speed benefit against other FRI methods in the price of handling crisp observations and crisp conclusions only. (This makes no real drawback in the example.) Next a simple application example will be presented, which applies the proposed FRI based BBC structure for autonomous vehicle navigation. The vehicle follows pre-defined waypoints while avoiding collision with obstacles and walls. The vehicle navigation example includes competitive and cooperative behaviour components as well.

2. Fuzzy Rule Interpolation

2.1. FRI Introduction

Traditional fuzzy reasoning methods (e.g. the Zadeh-Mamdani compositional rule of inference (CRI) and the Takagi-Sugeno reasoning method) are demanding complete rule bases, and hence the construction of a classical rule base requires extensive work to define all the required rules. In contrary, the application of fuzzy rule interpolation (FRI) methods, where the derivable rules are missing on purpose (as FRI methods are capable of providing reasonable (interpolated) conclusions even if none of the defined rules fire under the current observation) allows avoiding a considerable amount of unnecessary work in the construction of the rule bases, because the rule base of an FRI controller can contain the most significant fuzzy rules only. 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. Additionally, in some methods interpreting the fuzzy conclusion gained is not straightforward [8] even if there has been a great deal of effort to rectify the interpretability of the interpolated fuzzy conclusion [16]. In [1] Baranyi *et al.* give a comprehensive overview of recent existing FRI methods. Moreover, some of the FRI methods need special extension for the multidimensional case (e.g. [2]-[3]) because they are originally defined for one dimensional input space. In [19] Wong et al. gave a comparative overview of the multidimensional input space capable FRI methods and in [2] Jenei introduced a way for axiomatic treatment of the FRI methods. In [6] Johanyák *et al.* introduce an automatic way for direct sparse fuzzy rule base generation based on given input-output data. Many of these methods are hardly suitable for real-time applications due to the high computational demand (notably the search for the two closest surrounding rules to an arbitrary observation in the multi-dimensional antecedent space). Some FRI methods, e.g. LESFRI [7] or the method introduced by Jenei *et al.* in [3], eliminate the search for the two closest surrounding rules by taking all the rules into consideration, and therefore speed up the reasoning process. An application oriented aspect of the FRI emerges in the concept of FIVE (Fuzzy Interpolation based on Vague Environment), where the fuzziness of the antecedent and consequent fuzzy partitions is replaced by the concept of vague environment. This makes a speed benefit against other FRI methods, but it has the price of handling crisp observations and crisp conclusions only. It is a real disadvantage of FIVE, but in many direct FRI control applications, like the example in this paper, where the fuzzy conclusion is not required, it has no effect. In the followings the method FIVE will be introduced briefly.

2.2. The FRI "FIVE"

The FIVE method was originally introduced in [9], [10] and [11] and it was developed to fulfill the speed requirements of direct fuzzy control. In this case the conclusions of the fuzzy controller are applied directly as control actions in a real-time system, so the concept of the FIVE method is an application oriented aspect of the FRI techniques. Most of the control applications serve crisp observations and require 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], therefore it can turn the task of fuzzy interpolation to crisp interpolation. The idea of a vague environment is based on the similarity or in other words the indistinguishability of elements. In a vague environment the fuzzy membership function $\mu_A(x)$ indicates the level of similarity of x to a specific element a which is a representative or prototypical element of the fuzzy set $\mu_A(x)$, or it can be interpreted as the degree to which x is indistinguishable from a [4]. Two values in a vague environment are ε-distinguishable if their distance is greater than ε, where the distances are weighted distances. The weighting factor or function is called scaling function [4]. The scaling function serves the purpose of describing the shapes of the fuzzy sets in the partition. After determining the vague environment of both the antecedent and consequent part universes (the scaling function or at least the approximate scaling function [9], [11]), every member set of the fuzzy partition can be characterized by points in that vague environment (e.g. the approximated scaling function s shown in Fig. 3).

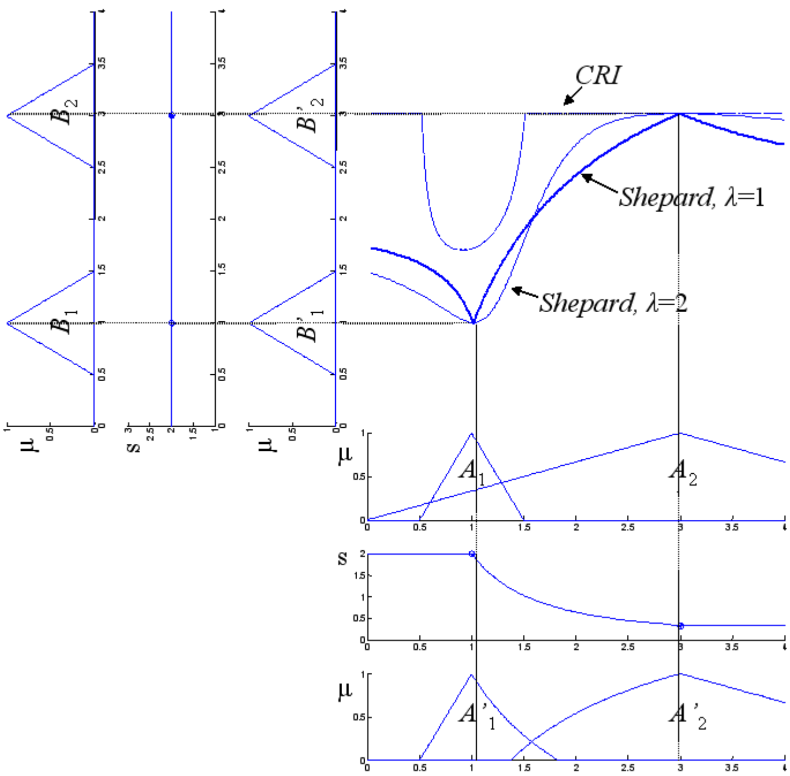

Figure 2. Interpolation of two fuzzy rules (R_i: $A_i \rightarrow B_i$), by the Shepard operator based *FIVE*, and for comparison the min-max *CRI* with COG defuzzification. λ is a parameter of the Shepard operator

The consequent and antecedent sides of the vague environment and scaling functions can be precalculated and cached, which provides the fastness of the method. Fig. 2 presents an example of a one-dimensional antecedent and consequent system with two fuzzy rules. Therefore if the observation is a singleton, any crisp interpolation, extrapolation, or regression method can be adapted very simply for FRI [9], [11]. In method FIVE, because of its simple multi-dimensional applicability, the Shepard operator based interpolation (first introduced in [15]) was adapted (see e.g. in Fig. 2). The Shepard operator based interpolation also appeared in other FRI methods like the stabilized KH interpolator which is proved to hold the universal approximation property in [17] and [18]. Beside its simplicity and therefore high reasoning speed, the original FIVE method has obvious drawbacks: the lack of the fuzziness on the observation side and on the conclusion side. The explanation is that this deficiency is inherited from the nature of the vague environment applied, which describes the indistinguishability of two points and therefore the similarity of a fuzzy set and a singleton only. The lack of fuzziness on the conclusion side has a little influence on common applications where the next step after the fuzzy reasoning is the defuzzification. On the other hand, the lack of fuzziness on the observation side can restrict applicability of the method. Furthermore, an extension of the original FIVE method was suggested in [12], where the question of fuzzy observation is handled by merging vague environments of the antecedent universes and the fuzzy observation. An implementation of FRI FIVE as a component of the FRI Matlab Toolbox [5] can be downloaded from [20] and [21].

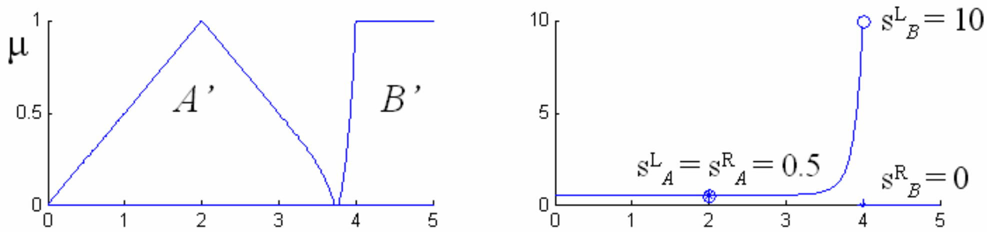

Figure 3. Approximate scaling function *s* generated by non-linear interpolation, and the partition as described by the approximate scaling function (A', B')

3. Vehicle Navigation Example

The example application of the paper is an autonomous vehicle navigation simulation which demonstrates the benefits of the proposed FRI based BBC structures. The goal of the application is to navigate the vehicle around given waypoints in a pre-defined order, while the vehicle should avoid collision with obstacles and the walls of the room. The vehicle can detect whether some obstacle is standing in its way, and hence whether the planned path of the vehicle seems to be blocked. In this case the vehicle can turn back and head in the opposite direction by reversing the sequence of the waypoints. The example waypoint configuration has four members which correspond to the four corners of the room.

3.1. Circular Waypoint Navigation and Collision Avoidance

For the navigation control the previously proposed BBC structure is adapted (see Fig. 1). The actual states, observations and symptom evaluation and behaviour components of the example are shown in Fig. 4. The suggested BBC has homogeneous FRI knowledge representation. In the following the rule bases for all the BBC components, the symptom evaluation, the state-transition and the behaviour component rule bases will be described and explained in more detail.

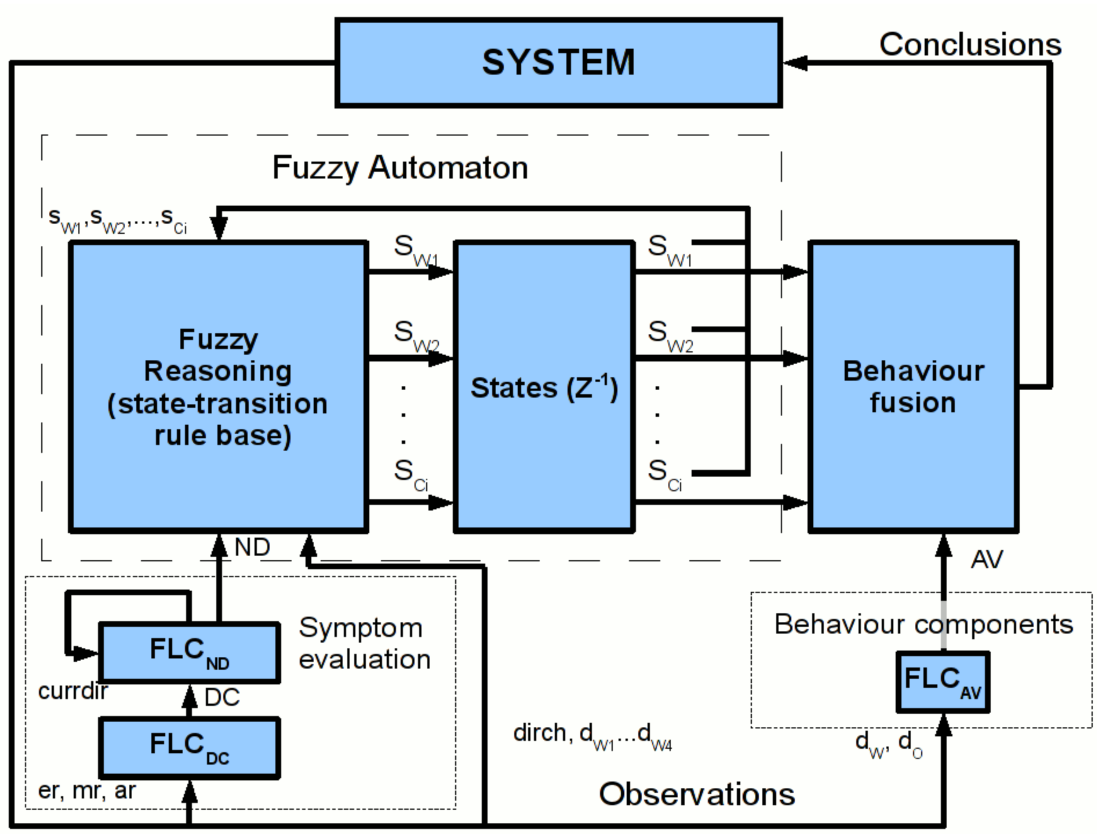

Figure 4. Diagram of the actual fuzzy automaton for the demonstration example

The navigation control is built of the following components: *waypoint approach (one for each waypoint)*, *wall avoidance, obstacle avoidance*, and the *heading direction change*.

The *waypoint approach component* (which is a component of the 'Fuzzy Automaton' labeled block in Fig. 4) partly determines the current state vector, the selection weights of the waypoints. (Then these weights will be applied as selection strengths of the corresponding waypoint directions.) The approximation of the waypoint selection weights is based on the following input parameters: the current position of the vehicle (described by the distances from the four waypoints; denoted respectively: dw_1, dw_2, dw_3, dw_4), the previous selection weights of the four waypoints (namely sw_1, sw_2, sw_3, sw_4 – from the previous state of the automaton), the need for direction changing and the current direction of the vehicle. The need for direction changing component calculates a weight, and if this weight is beyond the value of an adjustable parameter, then a direction change is necessary. The state-transition rule base is very simple, it assigns the highest waypoint selection weights in a predefined sequence which follows the nearest waypoint to the vehicle. A high level of a waypoint selection weight means that the vehicle is mainly heading towards the corresponding waypoint. 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*). For the state variables related to the waypoint selection weights (*WW* in Table 4-7), there are only two linguistic terms defined: true (*T*) and false (*F*) for the antecedent partitions and zerus (*Z*) and large (*L*) for the consequents. Each element of the waypoint selection weight (partly the state) vector has a separate state-transition rule base, and a similar structure. In the example case it means four state-transition rule bases (equal to the number of the pre-defined waypoints). Each rule base needs to be evaluated with the same measured distances and previous state variables. The conclusion is partly the new state, the normalized weight of the behaviour components heading for the corresponding waypoints, which is used to scale a vector pointing towards the corresponding waypoint.

The collision avoidance strategy consists of as many behaviour components as the number of the walls and obstacles and hence the same number of state values (the weight of the corresponding collision avoidance component) in the state vector. By definition walls are the borders of the room and obstacles are objects which can move freely inside the room. The *wall avoidance components* are very simple. There are as many normalised movement vectors as the number of the walls having a perpendicular direction to the corresponding wall. The state variables are the corresponding repulsion rates, one for each wall avoidance component. The state variables (repulsion rates, S_{Ci} in Fig. 4) are calculated based on the distance from the corresponding wall. The structures of the rule bases are similar and introduced in Table 1. Obstacle avoidance is solved in the same manner as wall avoidance. It has as many component behaviours as the number of the obstacles. They are normalised movement vectors having a direction opposite to the resultant waypoint movement vector. Similarly to wall avoidance the corresponding state variables are their weights calculated based on the distance between the vehicle and the obstacle in the same way as the states of wall avoidance (see again Table 1). Observations of the *wall and obstacle avoidance components* are the measured distances from each of the walls (denoted: d_w), and the measured distances from each of the objects inside the room (denoted: d_o). The linguistic terms of the antecedent universes are: zerus (*Z*), small (*S*), medium (*M*), large (*L*), and for the consequent universes (*AV*): zerus (*Z*), small (*S*), large (*L*).

The *wall and obstacle avoidance components* use the same rule base structure (see Table 1) for all the required conclusions; only the input distances differ within every evaluation. The conclusions are the state variables (component weights) related to the wall and obstacle avoidance components and applied in the behaviour fusion component in the same manner as for the waypoint direction components.

The structure of the wall and obstacle avoidance state rules are defined in the following form:

RColl_i: **If** $d_w = \mathrm{A}_i$ **Then** $AV = \mathrm{B}_i$

Table 1. 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

The behaviour fusion part of the example is a simple convex combination of the component behaviours with the corresponding weights (state variables).

3.2. Heading Direction Change Extension

As already mentioned earlier in the case when the way of the vehicle seems to be blocked in the current direction, the vehicle can change its heading, by assigning the waypoints in the reverse order. This direction change decision is made by the *heading direction change* symptom evaluation (see Fig. 4) component. The observations needed for this component (see Fig. 5) are the sum of movement rates of the vehicle and the collision avoidance vector (denoted: *mr*), the summarized rate of the length of the wall and obstacle avoidance vectors (denoted: *ar*). In a hierarchical navigation control, the vehicle could do some other types of movements beyond navigation among the waypoints ('exploration'), hence an 'exploration rate' observation could be also added (*er* in Table 2) to control the level of our example navigation strategy as a component behaviour itself in a more complex system.

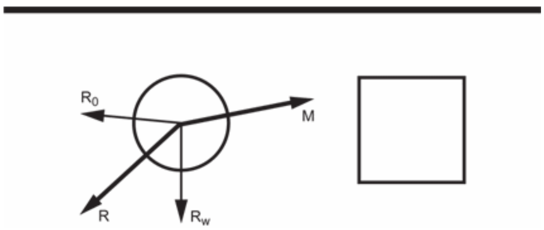

Figure 5. *M* is the movement vector of the vehicle towards the next waypoint, R_o is the repulsion vector of the obstacle, R_w for the wall and *R* is their sum

The linguistic terms of the two antecedent universes of the *heading direction change component* are: zerus (*Z*) and large (*L*). For the conclusion universe (*DC*), which tells whether to change the direction of the vehicle or not, the linguistic terms are: false (*F*) and true (*T*). The rule base consists only of three rules, which can be seen in Table 2. 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 2. Direction changing behaviour component decision rule base

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

One more rule base is used to determine the new heading direction for the vehicle. Two observations are required for this subcomponent, which is also a fuzzy automaton in the symptom evaluation component, with one state variable (see in Fig. 4): a value which tells whether a direction heading change is necessary (denoted: *dirchg*) (this is the conclusion above, see Table 2) and the current heading direction state (denoted: *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 conclusion of the symptom evaluation will be the new state of the fuzzy automaton (direction).

The state-transition rule base of the fuzzy automaton embedded into the symptom evaluation can be seen in Table 3, and the rules can be interpreted according to the following form:

$\mathrm{RNewDir}_i$: **If** *dirchg* = $A_{1,i}$ **and** *currdir* = $A_{2,i}$ **Then** *ND* = B_i

Table 3. 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

Having the rule bases for direction changing decision the state-transition rule base of the waypoint selection weights can also be extended with direction changing. Some new observations should be added to the waypoint selection weights state-transition rule base which were introduced earlier: the current heading direction (denoted: *dir*) and a parameter expressing whether the heading direction was changed (denoted: *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*).

As mentioned, four rule bases are required in this particular case. E.g. having four waypoints in case the direction is clockwise: first to calculate the weight needed to take the vehicle towards the 2nd waypoint, second to direct the vehicle to the 3rd waypoint, third to take the vehicle to the 4th waypoint, and a fourth rule base to navigate the vehicle back to the 1st waypoint. E.g. in case of the first waypoint the waypoint selection weights state-transition rule base has the following meaning (see Table 4): the first rule means that when the corresponding waypoint (1st) is reached by the vehicle then that waypoint (1st) should be abandoned, hence the weight of the waypoint (1st) will be zerus (*Z*). The second rule keeps the vehicle coming to the waypoint (1st) if it has been selected earlier. The third rule stops the vehicle 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 (2nd 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 4th). The sixth rule serves the purpose of keeping down the weight when the vehicle is going to the next (2nd) waypoint, so do the seventh and eighth rules, but for the remaining two waypoints (4th and 3rd respectively). The ninth means that when the vehicle reaches the previous waypoint in the sequence (4th), it should head for the current waypoint (1st). The meaning of the last rule is very similar to the previous one, but for the opposite heading direction. The rule bases for the 2nd, 3rd and 4th waypoints contain similar rules, the differences are only the rotational numbering of the corresponding next and previous waypoints numbers.

The extended waypoint selection weights state-transition rules are defined in the following form:

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 $WW = B_i$

With the rule bases above described the vehicle can cycle around the given waypoints, with direction change in blocked situations, while still avoiding obstacles and walls.

Table 4. 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 5. 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 6. 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 7. 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

3.3. Implementation Remarks

It is recommended 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 vehicle. 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 vehicle. 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 FRI model of vehicle navigation and collision avoidance.

With a simple algorithm the waypoint selection rule bases can be generated based on the number of the defined waypoints. This was implemented as a standalone script written in Python programming language, which can be found at [22]. Using this script, dynamic waypoint insertion/deletion could be achieved with regeneration of the waypoint selection rule base every time the count of waypoints has been modified. A drawback is that this feature implies modifications not only in the rule bases (and the number of rule bases also), but in the rule base evaluation procedures. Achieving the latter requires further research.

For non-commercial purposes the Matlab source of the example of the paper can be accessed free of charge at [22].

Conclusion

Some details of an autonomous surveillance vehicle implementation based on fuzzy automaton and behaviour-based control navigating were examined in the paper. The knowledge representation of the behaviour components, and the state-transition rule-base of the system state approximation were implemented as sparse fuzzy rule bases of the “FIVE” fuzzy rule interpolation method. Beyond the successful application, a notable conclusion of the paper is that by using FRI methods the rule base sizes can be considerably reduced to a fraction of the original sizes. Building a complete fuzzy rule base for the behaviour components introduced in the paper with the same strategies, but with complete rule-base could require approximately a thousand fuzzy rules. On the other hand applying sparse fuzzy rule bases (and fuzzy rule interpolation, in case of 4 waypoints, 4 walls and 2 obstacles) only 71 rules are sufficient. This rule base size is easily implementable even in an embedded FRI fuzzy logic controller. The implementation also proves the real-time suitability of the FIVE fuzzy rule interpolation method itself (in application areas where as a restriction of the method crisp observation and crisp conclusion are sufficient). For non-commercial purposes an implementation of FRI FIVE as a component of the FRI Matlab Toolbox [5] can be downloaded from [20] and [21].

Acknowledgements

This research was partly supported by the Hungarian National Scientific Research Fund grant no: OTKA K77809.

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] 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.

[6] ZS. CS. JOHANYÁK, R. PARTHIBAN, AND G. SEKARAN: *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.

[7] ZS. CS. JOHANYÁK, SZ. 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.

[8] 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.

[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] KOVÁCS, SZ.: *Extending the Fuzzy Rule Interpolation "FIVE" by Fuzzy Observation*. Theory and Applications, Springer Berlin Heidelberg, 2006, pp. 485-497.

[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] PIRJANIAN, P.: *Behavior Coordination Mechanisms - State-of-the-art, Tech-report* IRIS-99-375, Institute for Robotics and Intelligent Systems, School of Engineering, University of Southern California, October (1999)

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

[16] D. TIKK AND P. BARANYI: *Comprehensive analysis of a new fuzzy rule interpolation method*. IEEE Trans. Fuzzy Systems, vol. 8, No. 3, June, 2000, pp. 281-296.

[17] 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, 2002, 105-119.

[18] D. TIKK: *Notes on the approximation rate of fuzzy KH interpolator.* Fuzzy Sets and Systems, (138) 2, 2003, pp. 441-453.

[19] 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.

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

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

[22] The example application can be found at: http://www.iit.uni-miskolc.hu/~vinczed/vehnav/