Sz. Kovács: *Interpolative Fuzzy Reasoning and Fuzzy Automata in Adaptive System Applications*, Proceedings of the IIZUKA2000, 6th International Conference on Soft Computing, October 1-4, Iizuka, Fukuoka, Japan, pp.777-784, (2000). – Draft version.

Interpolative Fuzzy Reasoning and Fuzzy Automata in Adaptive System Applications

Szilveszter Kovács
Gifu Prefectural Research Institute of Manufactural Information Technology
4-179-1 Sue Kakamigahara Gifu 509-0108 Japan
Tel: +81-583-79-2207; Fax: +81-583-79-2208; E-mail: szkszilv@gold.uni-miskolc.hu

Key Words: fuzzy automata, interpolative fuzzy reasoning and behaviour-based control

Abstract – One view of hybrid control, or multiple objective, behaviour-based control systems is a kind of strategy reconfiguration – an intelligent adaptation of the system to the actual situation. In case of hybrid control it is an actual situation dependent discrete strategy (controller) changing, while in behaviour-based control systems it is a discrete switching to the most appropriate strategy or a kind of fusion of the strategies appeared to be the most appropriate ones in an actual situation. In this paper we suggest a strategy reconfiguration structure, which based on fuzzy interpolative fusion of different strategies in the function of the actual necessity of the corresponding strategies approximated by a fuzzy automata. For introducing some of the possible application areas of the proposed structure, a behaviour-based vehicle control, a fault tolerant control and a user adaptive emotion-based (Kansei) selection system application is also introduced shortly in this paper.

1 Introduction

In behaviour-based control systems (a good overview can be found in [1]), the actual behaviour of the system is formed as one of the existing system behaviours (which fits best the actual situation), or a kind of fusion of the known behaviours appeared to be the most appropriate to handle the actual situation. This structure has two main task. The first is a decision, which behaviour is needed in an actual situation, or the levels of their necessities in case of behaviour fusion, the second is the way of the behaviour fusion.

A different view of the same problem is a kind of situation adaptive strategy reconfiguration. The main idea of this view is the following: “More similar the actual situation to one of the known partial strategy prerequisites, more similar must be the strategy used to that strategy”. We call this view *similarity based strategy reconfiguration*. The main tasks of the similarity based strategy reconfiguration – as it is basically the same problem as the behaviour-based control using behaviour fusion - the decision about the levels of necessities of the different strategies, and the way of the strategy fusion.

The first task can be viewed as an actual system state approximation, where the actual system state is the approximated level of similarities of the actual situation to the prerequisites of all the known strategies (the level of necessity and the type of the strategy needed to handle the actual situation). The second is the fusion of the existing partial strategies based on these similarities. For the first task, we suggest the adaptation of finite state fuzzy automata, where the state variables are the corresponding similarities, and the state transitions are driven by fuzzy reasoning (State Transition Rulebase on fig.2.). A similar idea - adapting crisp finite state automata, and crisp states for the actual situation approximation - *Discrete Event Systems* is introduced in [2]. In our case the adaptation of finite state fuzzy automata supports the potentialities of strategy fusion, instead of the crisp strategy switching.

For the second task, the application of interpolative fuzzy reasoning is suggested. Having the approximated similarities of the actual situation to the prerequisites of all the known strategies, the conclusions of the different strategies could be simply combined as an upper level interpolative fuzzy reasoning in a function of the corresponding similarities to get the actual final conclusion (Interpolative Fuzzy Reasoning on fig.2.). A similar idea - adapting fuzzy rulebase for conclusion fusion – fuzzy metarules for activating (fusing) control schemes is introduced in [3], and a fuzzy inference method for conclusion fusing "*Fuzzy Damn*" is introduced in [4]. Both method is based on classical fuzzy reasoning methods. In our case the adaptation of interpolative fuzzy reasoning gives the benefit of simple built conclusion fusing rulebase (in case of interpolative fuzzy reasoning the rulebase is not needed to be complete [5]) and the needless of defuzzification (in case of some interpolative fuzzy reasoning methods [6]).

The proposed similarity based strategy reconfiguration structure is based on the premise, that the interpolative combinations of the partially valid strategies are also partially valid strategies (at least in the area covered by the original partially valid strategies). This case having some relevant, but only partially valid strategies (with respect to the state space of the system), they can simply combined in interpolative manner to form one strategy, which has an extend area of validity (at least in a part of the area covered by the original partially valid strategies). Moreover because of the similarity based interpolative manner of strategy combination, there are some chance to get valid strategy in the area outside the area covered by the original strategies too. Usually it is not fulfilled in the case of combining neighbouring, or overlapping, but contradictive, or alternative strategies. The problem of behaviour-based control in case of alternative strategies is studied in [4], [1].

1.1 The system state approximation

The first step of the system state approximation is the symptom evaluation. The task of symptom evaluation is basically a series of similarity checking between an actual symptom and a series of known symptoms (in this case all the prerequisites - symptom patterns - of the known strategies). The prerequisites of the known partially valid strategies are symptom patterns. These symptom patterns are characterising the systems states where the corresponding partial strategy is valid. Based on these patterns, the evaluation of the actual symptom is done by calculating the similarity values of the actual symptom (the actual system behaviour) to all the known symptoms patterns (the prerequisites of the known partially valid strategies). There are many methods exist for fuzzy logic symptom evaluation. For example we can adopt fuzzy classification methods e.g. the Fuzzy c-Means fuzzy clustering algorithm [7], where the known symptoms patterns are the cluster centres, and the similarities of the actual symptom to them can be fetched from the fuzzy partition matrix. On the other hand, having a simple situation, the fuzzy logic symptom evaluation could be a fuzzy rule based reasoning system itself. One of the main difficulties of the system state approximation is the fact, that most cases the symptoms of the prerequisites of the known partially valid strategies are strongly dependent on the actual control strategy of the system. Each control strategy has its own symptom structure. In other words for the proper system state approximation, the approximated system state is needed itself. A very simple way of solving this difficulty is the adaptation of fuzzy automata. This case the state vector of the automata could be the approximated system state (it is the vector of the approximated similarities of the actual system behaviour to the prerequisites of the known strategies), and the state transition could be driven by fuzzy reasoning, based on the conclusions of the symptom evaluation (see fig.2.). Having an approximated system state and the conclusions of the symptom evaluation, a decision about the new approximated system state can be evaluated. More precisely, considering the description of the actual system behaviour to be the state variables of the automata (a set of similarities - fuzzy membership values), based on the actual state and the conclusions of the symptom evaluation, the state transition decision can be done by fuzzy reasoning (state-transition rulebase). This automata is a fuzzy automata, because its state variables are fuzzy membership values (similarities) and the state-transitions are driven by fuzzy rules.

The basic structure of the rulebase applied for the state-transitions of the fuzzy automata (rules for interpolative fuzzy reasoning) for the i^{th} state S_i (R_{Ai}) are the following: **(1)**

If S_i=One **And** S_i-S_i=One **Then** S_i=One
If S_i=One **And** S_i-S_k=One **Then** S_i=Zero
If S_k=One **And** S_k-S_i=One **Then** S_i=One
If S_k=One **And** S_k-S_i=Zero **Then** S_i=Zero

where S_i-S_j is the conclusion of the symptom evaluation about the state-transition from state i to j. The structure of the state-transition rules are similar for all the strategies (see e.g. fig.1.). Because of the possibly incomplete structure [5] of state-transition rulebase, we applied interpolative fuzzy reasoning for the state transition decisions.

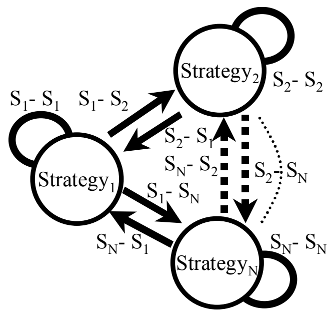

Fig. 1: Basic structure of the state-transition rules.

Comments: Please note, that the state variables of the automata are level of similarities (as many as known strategies we have), so it is a finite state fuzzy automata. The states shown on fig.1., are only some discrete points of the infinite similarity state space.

In case of having a simple situation, where fuzzy logic rule based symptom evaluation can be applied, the fuzzy symptom evaluation (rulebase) could be integrated to the state transition rulebase of the fuzzy automata (as it was done in the first example application of this paper).

1.2 The strategy fusion

The conclusion of the system state approximation (the approximated state itself) is a set of similarity values, the level of similarities of the actual situation to all the prerequisites of the known partially valid strategies. Having all the conclusions of the different known partially valid strategies (FLC_i - y_i), the actual conclusion (y) could be simply combined from them in the function of the corresponding similarities (S_i), as an upper level interpolative fuzzy reasoning [6] (see fig.2.).

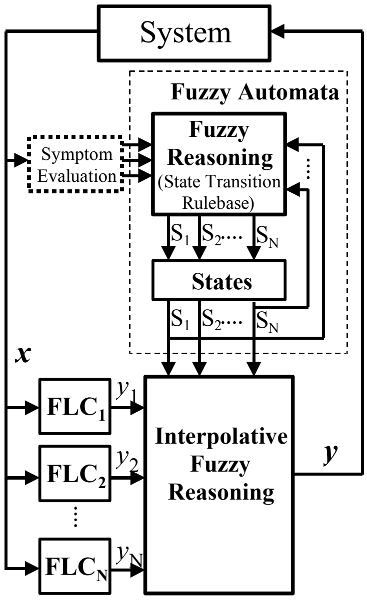

Fig.2 : Structure of the proposed Similarity based control strategy reconfiguration.

The simplest way for such a combination, is the application of the interpolative fuzzy reasoning. The main idea of the proposed similarity based strategy reconfiguration – "More similar the actual situation to one of the known partial strategy prerequisites, more similar must be the strategy used to that strategy" – can be directly translated to an interpolative fuzzy rulebase. (Applying interpolative fuzzy reasoning the completeness off the fuzzy rulebase is not necessary.) The rulebase applied for the interpolative fuzzy reasoning to combine the conclusions (y_i) of the known partial strategies in a function of the corresponding similarities is the following (see fig.2. for notation): **(2)**

If S_1=One **And** S_2=Zero **And** ... **And** S_N=Zero **Then** $\boldsymbol{y}=y_1$

If S_1=Zero **And** S_2=One **And** ... **And** ... S_N=Zero **Then** $\boldsymbol{y}=y_2$

If S_1=Zero **And** S_2=Zero **And** ... **And** S_N=One **Then** $\boldsymbol{y}=y_N$

Comments: instead of interpolative fuzzy reasoning a kind of weighted average, (where the weights are functions of the corresponding similarities) is also applicable (even it is not so flexible in some cases).

2 Application examples

For introducing some of the possible application areas of the proposed similarity based reconfiguration structure, a behaviour-based vehicle control, a fault tolerant control application and a user adaptive emotion-based (Kansei) selection system is shortly introduced in the followings.

2.1 AGV navigation control example

Our first application example is a simulated steering control of an automated guided vehicle (AGV) [8], [9]. In the example application the steering control has two main goal, the path tracking (to follow a guide path) and the collision avoidance. The simulated AGV is first trying to follow a guide path, and in the case if it is impossible (because of the obstacles) leave it, and as the collision situation is avoided try to find the guide path and follow it again. A simulated path sensing system senses the position of the guide path by special sensors (guide zone) tuned for the guide path. The goal of the path tracking strategy is to follow the guide path by the guide zone with minimal path tracking error on the whole path (see fig.3.).

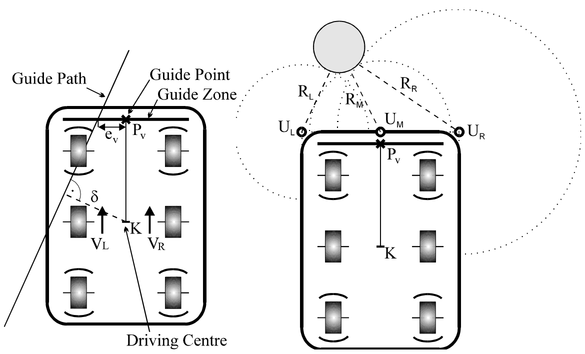

Fig. 3: Differential steered AGV with guide zone, δ is the path tracking error, e_v is the distance of the guide path and the guide point, P_v is the guide point, K is the driving centre, R_L, R_R, R_M are the distances measured by the left, right and middle ultrasonic sensors (U_L, U_R, U_M).

For defining the collision avoidance strategies, the types of the possible collision situations must be studied. There are two different collision situations, the frontal and the side collision. Having the preconditions of motionless obstacles, it is sufficient to have three ultrasonic distance sensors on the front of the AGV, one in the middle (U_M) and one-one on both sides (U_L, U_R) (see fig.3.) to approximate both the collision conditions [9]. Having the preconditions of motionless obstacles, we can use the obstacle distance measurements of the near past for scanning the boundaries of the obstacles. Collecting the previous measurements of the left and right obstacle sensors and the corresponding positions of the AGV (measured by the motion sensors on the wheels), we can approximate the boundaries of the obstacles by discrete points (see fig.4.) [9]. These points are the unsafe or risky points. The main idea of the side collision avoidance part of the strategies is to avoid side collisions to obstacles by avoiding side collisions to unsafe points. For having observations easier to handle then a set of unsafe points, the calculated actual maximal left and right turning angle without side collision (α_{ML}, α_{MR}) was used instead (see e.g. on fig.5.).

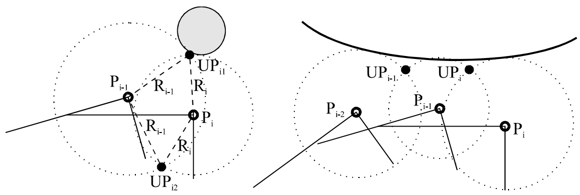

Fig. 4: The obstacles boundaries approximated by discrete unsafe points, where R is the distance measured by the ultrasonic sensor P, and UP is the unsafe (risky) point.

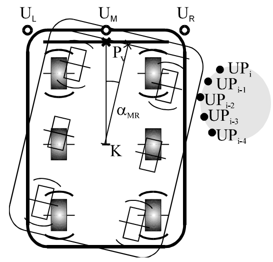

Fig. 5: α_{MR} is the maximal right turning angle without side collision.

2.1.1 The known partial strategies

The first step of similarity based strategy reconfiguration is to build the component partially valid strategies. The simplest way of defining these strategies is based on describing the operator's control actions. These control actions could form a fuzzy rule base. In the example - using interpolative fuzzy reasoning for direct fuzzy control - constructing the fuzzy rule base is very simple. It is not necessary to build a complete fuzzy rule base; it is enough to concentrate on the main control actions, by simply adding rules piece by piece. Having the simulated model of the controlled system, the performance of the controller can be checked after each step. In the simulated example, there are four different partial strategies. These strategies are the followings:

Path tracking and restricted collision avoidance strategy. The main goal of the path tracking and restricted collision avoidance steering strategy control is the path tracking (to follow a guide path) and as a sub goal, a kind of restricted (limited) collision avoidance [9]. (Here the restricted collision avoidance means, “avoiding obstacles without risking the chance of loosing the guide path”.) The base idea of the path tracking strategy is very simple: keep the driving centre of the AGV as close as it is possible to the guide path, than if the driving centre is close enough to the guide path, simply turn the AGV into the new direction. Adding the collision avoidance, this simple strategy needs seven observations: Two for the path tracking, the distance between the guide path and the driving centre (*estimated momentary path tracking error* (e_v)), and the *distance between the guide path and the guide point* (measured by the guide path (δ)). Five for the collision avoidance, the *distances measured by the left middle and right ultrasonic sensors* (R_L, R_M, R_R) and the *approximated maximal left and right turning angle without side collision* (α_{ML}, α_{MR}). Based on these observations two conclusions, the *speed* (V_a) and the *steering* (V_d) are needed. This means two single conclusion rulebases, one for the steering $\mathbf{R_{Vd}}$ and one for the speed $\mathbf{R_{Va}}$ of the AGV.

The i^{th} rule of the steering rulebase has the following form: $R_{Vd,i}$:

If $e_v = A_{1,i}$ **And** $\delta = A_{2,i}$ **And** $R_L = A_{3,i}$ **And** $R_R = A_{4,i}$ **And** $R_M = A_{5,i}$ **And** $\alpha_{ML} = A_{6,i}$ **And** $\alpha_{MR} = A_{7,i}$ **Then** $V_d = B_i$.

Having a simulated model of the AGV and a trial guide path, there are 12 rules needed for controlling the steering ($\mathbf{R_{Vd}}$) and 5 for the speed ($\mathbf{R_{Va}}$):

$\mathbf{R_{Vd}}$:	e_v	δ	R_L	R_R	R_M	α_{ML}	α_{MR}	V_d
1.,	*NL*							*PL*
2.,	*PL*							*NL*
3.,	*NM*	*Z*					*L*	*PL*
4.,	*PM*	*Z*				*L*		*NL*
5.,	*NM*	*PM*	*L*		*L*	*L*		*Z*
6.,	*PM*	*NM*		*L*	*L*		*L*	*Z*
7.,	*Z*	*PM*	*L*		*L*	*L*		*NS*
8.,	*Z*	*NM*		*L*	*L*		*L*	*PS*
9.,	*Z*	*PM*	*S*		*S*			*PL*
10.,	*Z*	*NM*		*S*	*S*			*NL*
11.,	*Z*	*Z*	*L*	*S*	*S*			*NL*
12.,	*Z*	*Z*	*S*	*L*	*S*			*PL*

$\mathbf{R_{Va}}$:	e_v	δ	R_L	R_R	R_M	V_a
1.,	*Z*	*Z*	*L*	*L*	*L*	*L*
2.,	*NL*	*PL*				*Z*
3.,	*PL*	*NL*				*Z*
4.,	*NL*	*Z*				*Z*
5.,	*PL*	*Z*				*Z*

where *N*: negative, *P*: positive, *L*: large, *M*: middle, *Z*: zero.

The collision avoidance strategy. The second partially valid strategy is a simple collision avoidance steering strategy. Its only goal is to avoid collisions. Having a simulated model of the AGV after some trial, the following rules are needed for controlling the steering ($\mathbf{R_{Vd}}$) and 5 for the speed ($\mathbf{R_{Va}}$):

$\mathbf{R_{Vd}}$:	R_L	R_R	R_M	α_{ML}	α_{MR}	V_d
1.,		*Z*		*L*		*NL*
2.,	*Z*				*L*	*PL*
3.,		*Z*	*L*	*S*		*NVS*
4.,	*Z*		*L*		*S*	*PVS*

$\mathbf{R_{Va}}$:	R_L	R_R	R_M	V_a
1.,	*L*	*L*	*L*	*L*
2.,			*S*	*S*

where *N*: negative, *P*: positive, *L*: large, *M*: middle, *S*: small, *VS*: very small, *Z*: zero.

The collision avoidance with left/right tendency strategy. The next two partially valid strategies are basically the same as the collision avoidance steering strategy, expect the left or right turning tendencies in case of no left or right turning difficulties. These strategies are needed to aid finding the path after leaving it (because of the fail of the first strategy). Their rulebases are the same as the rulebases of the collision avoidance strategies, except one additional rule, which causes the left/right turning tendencies in collision free situations. The additional rule for the right tendency to the collision avoidance steering strategy ($\mathbf{R_{Vd}}$):

$\mathbf{R_{Vd}}$:	R_L	R_R	R_M	α_{ML}	α_{MR}	V_d
1-4.,	...	...	...	...	...	...
5.,		*L*	*L*		*L*	*PL*

The additional rule for the left tendency to the collision avoidance steering strategy ($\mathbf{R_{Vd}}$):

$\mathbf{R_{Vd}}$:	R_L	R_R	R_M	α_{ML}	α_{MR}	V_d
1-4.,	...	...	...	...	...	...
5.,	*L*		*L*	*L*		*NL*

2.1.2 The symptom evaluation and the fuzzy automata

The example application is so simple, that it does not need separate symptom evaluation. The function of the symptom evaluation can be built to the state-transition rulebase of the fuzzy automata. Having four partial known strategies, the automata has four state variables. These are the approximated level of similarity of the actual system to the prerequisites of the path tracking and restricted collision avoidance strategy (S_P), to the prerequisites of the collision avoidance strategy (S_C), to the prerequisites of the collision avoidance strategy with right tendency (S_{CR}), and left tendency (S_{CL}). Having four conclusions, four state transition rulebases is needed. The $\mathbf{R_{SP}}$ state transition rulebase is determining the next value of the S_P state variable, $\mathbf{R_{SC}}$ is for determining S_C, $\mathbf{R_{SCR}}$ for S_{CR}, and $\mathbf{R_{SCL}}$ for S_{CL}. The observations of the state transition rulebases are the observations introduced in the path tracking and partial collision avoidance strategy, the state variables themselves (S_P,S_C,S_{CR},S_{CL}), and a new observation (P_V), signing if the path sensing is available (valid), or not:

$\mathbf{R_{SP}}$:

S_P	S_C	S_{CR}	S_{CL}	e_v	PV	R_L	R_R	R_M	α_{ML}	α_{MR}	S_P
				Z	*V*			*L*			*L*
				PL	*V*					*S*	*Z*
				NL	*V*				*S*		*Z*
					NV						*Z*

$\mathbf{R_{SC}}$:

S_P	S_C	S_{CR}	S_{CL}	e_v	PV	R_L	R_R	R_M	α_{ML}	α_{MR}	S_P
					V			*S*			*L*
					V			*L*			*Z*
					NV						*Z*

$\mathbf{R_{SCR}}$:

S_P	S_C	S_{CR}	S_{CL}	e_v	PV	R_L	R_R	R_M	α_{ML}	α_{MR}	S_P
L				*NVL*	*V*						*L*
		L			*NV*						*L*
				Z	*V*			*L*			*Z*
			L								*Z*

$\mathbf{R_{SCL}}$:

S_P	S_C	S_{CR}	S_{CL}	e_v	PV	R_L	R_R	R_M	α_{ML}	α_{MR}	S_P
L				*PVL*	*V*						*L*
			L		*NV*						*L*
				Z	*V*			*L*			*Z*
		L									*Z*

where *N*: negative, *P*: positive, *VL*: very large, *L*: large, *S*: small, *Z*: zero, *V*: path valid, *NV*: path not valid.

2.1.3 The conclusions of the example

Fig.6. introduces some results of the simulated application of the proposed method for the task of path tracking and collision avoidance control of an AGV. The results shows, that in the tested situation the proposed similarity based strategy reconfiguration method was able unify the relevant, but only partially (with respect to the state space of the system) valid strategies.

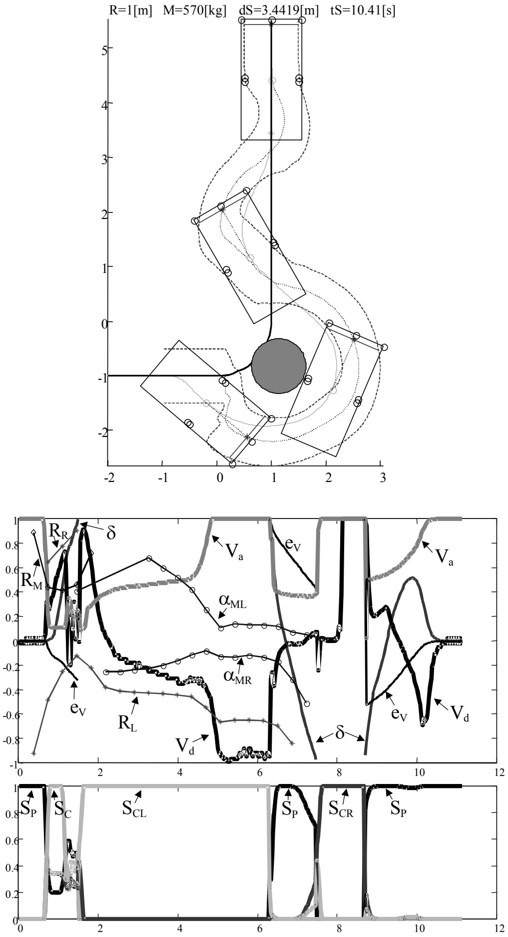

Fig. 6: Path, and time function of observations, conclusions and system state values (simulated results, one obstacle).

2.2 Fault tolerant control example

As a second simple application example of the proposed similarity based reconfiguration structure, fault diagnosis and reconfiguration of a simplified configuration (two tanks only) of the three tank benchmark [10] (fig.7.) is introduced in the followings.

The goal of the control system is to keep the water levels in tank_1 and tank_3 h_1=0.5 and h_3=0.1 by controlling the valve_{13} and the pump_1 at a constant value of outflow from tank_3 (*normal* behaviour of the system).

The example is concentrating of the faults of the valve_{13}. Were this valve opened and blocked, the water level in tank_3 h_3=0.1 could be controlled by pump_1 (this case h_1 is changed) - this is the *fault condition no.1*. Were this valve closed and blocked, the water levels in tank_1 and tank_3 h_1=0.5 and h_3=0.1 could be controlled by the valve_1 and the pump_1 - this is the *fault condition no.2*.

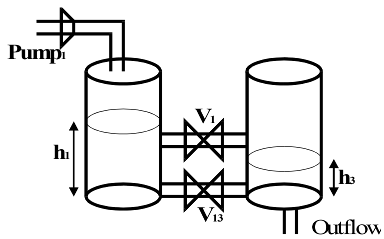

Fig. 7: Simplified configuration of the three tank benchmark.

The way of the implementation of the proposed similarity based reconfiguration structure is very similar to the previous example. However having a little bit more complicated symptom structure, after generating the controllers handling the separate situations (one controller for the normal, one for handling fault 1 and one for fault 2), we have to build a separate symptom evaluation module. In this example we applied fuzzy clustering for this task. The symptom evaluation module have to be able to characterise all the system situations (state-transitions) in all the studied control states. The state-transition diagram for the studied states of the interpolative fuzzy automata is shown on fig.8. (Please note, that the states are fuzzy membership values.) The structures of the state-transition rulebase (1) and the strategy fusion rulebase (2) are the same as introduced in section one.

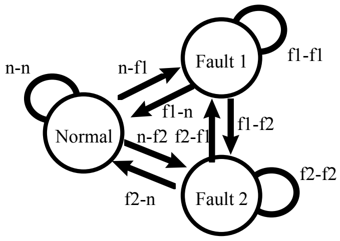

Fig. 8: State-transition diagram of the fuzzy automata, where S_i-S_j is the symptom evaluation conclusion about the - from i to j state-transition.

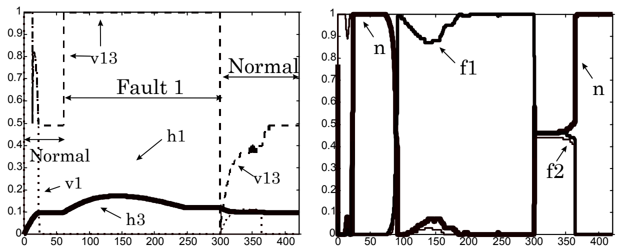

Fig. 9: The simulated results and the approximated fuzzy states of the complete control system (Normal - Fault 1 – Normal behaviour).

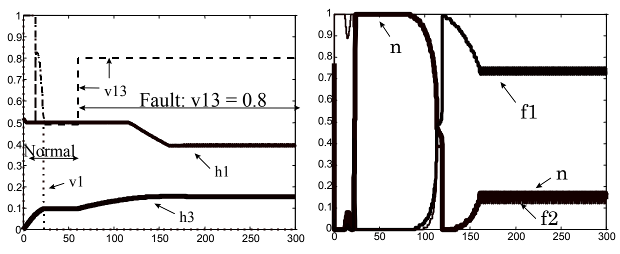

Fig. 10: The simulated results and the approximated fuzzy states of the complete control system (Normal - Fault: v_{13} = 0.8 unstudied situation).

2.3.1 The conclusions of the example

The simulated example application demonstrated, that the fuzzy automata is able to follow the studied relevant states and state-transitions (e.g. normal-fault1-normal on fig.9.). Moreover, because of the interpolative properties of the fuzzy automata in some cases it is able to approximate the unstudied situations too. Based on the similarities of the actual situation to the studied situations, in some cases, the fused conclusions are able to handle unknown (unstudied) fault situations (see e.g. fig.10., where v_{13}=0.8 open is an unstudied fault situation).

2.3 User adaptive emotion-based system example

The last simple application example of the proposed similarity based reconfiguration structure, is a user adaptive emotion-based (Kansei) system – an interactive furniture selection system [11]. The example application is based on the idea, that from the viewpoint of the system, the user adaptivity is basically the same task as the situation (environment) adaptivity, introduced in the previous examples. It is handling user adaptivity, as a kind of combination of existing (off-line collected) human opinions (user models) in the function of the approximated similarity to the actual user opinions. As an analogy to the previous applications, the different existing strategies are the different existing user models, and the actual situation is the similarity of the actual user to the evaluators, gave the existing user models.

The similarity based reconfiguration structure implementation is very similar to the previous examples (see fig.11.). The main differences (compare fig.2.-fig.11.) are the substitution of the known strategy controllers (FLC_i) by existing user models (Furniture Descriptor – Kansei descriptors), and the direct similarity checking (similarities of the actual user opinions to the content of the existing models) instead of symptom evaluation only.

Using the selection system, the user can search a furniture database by giving emotion-related requests (like “friendly” or “convenient”). These requests are translated to physical parameters (characterising the real furniture objects) by the actual user (Kansei) model. The user adaptivity of the actual user (Kansei) model (see fig.11.) (please note, that the physical meanings of the emotional words are highly user dependent) is provided by the similarity based reconfiguration – by the similarity based existing user model fusion. This case the state of the fuzzy automata (actual similarities, see fig.11.) is interpreted as, the actual approximated similarities of the actual user and the existing user opinions. The state-transitions rulebase is a slightly different to the rulebase (1) suggested in the first section. This is because the direct similarity checking gives the similarities of the actual user opinions to the content of the existing models, instead of the state-transition decisions of symptom evaluation (the similarities are actual state/model independent).

The rulebase applied for the state-transitions of the fuzzy automata for the i^{th} state S_i (R_{Ai}):

If S_i=One **And** SS_i=One **Then** S_i=One
If S_i=Zero **And** SS_i=Zero **Then** S_i=Zero
If S_i=Zero **And** S_k=One **And** SS_i=One
And SS_k=One **Then** S_i=Zero
If S_i=One **And** SS_i=Zero
And SS_k=Zero **Then** S_i=One
If S_i=Zero **And** S_k=Zero **And** SS_i=One
And SS_k=Zero **Then** S_i=One

where SS_i is the calculated similarity of the actual user opinion to the i^{th} existing user opinion, $k \in [1, N], k \neq i$. The structure of the state-transition rules is similar for all the states. For the conclusion (user model) fusion, both interpolative fuzzy reasoning and kind of weighted average (where the weights are functions of the corresponding similarities) were tested [11]. In the final example application, because of the simplicity of the conclusion fusion rulebase (no need for the flexibility of the rule-based fusion), and the need of the quick response of the interactive program, weighted average were used.

2.3.1 The conclusions of the example

The goal of the actual user model modifications from the actual user side is to tune the system to be closer to his/her opinions. Practically the system is starting from an initial stage (where the similarities to the existing models are equal), and in the case the user is disagree with the evaluation of the actual object (furniture) given by the system, he/she has the possibility to modify the actual user model by giving his/her opinions. In most cases the given opinions are related to one or a few Kansei descriptors (emotional words) of the edited object (furniture in our case). But because of the proposed structure, all the changes are done globally (all the Kansei descriptors of an existing user model has the same weights "globally" in the actual model – not only the descriptor weights related directly to the given user opinion are "locally" modified). We hope that this kind of adaptation strategy keeps the actual user model coherent, and able to avoid incoherence could caused by step by step partial modifications of the user model. E.g. if one of the users have exactly the same opinions as one of the existing user model (even his opinions were given through some of the Kansei parameters only), then (after a few modification, detection steps) as the best fitting existing user model, the system will use it exactly. Basically the "adaptive knowledge" of the system related to the actual user is not a new adapted user model, but the actual system state, a set of approximated similarities of the actual user to the existing user models. Because of the interpolative properties of the user model combination, the proposed system is unable to follow user requirements outside the area covered by the existing user models. In other words, the system cannot go beyond its existing "knowledge". The only solution of this problem is extending the number and the variety of the existing user models, to cover the state space by user models as much as it is possible.

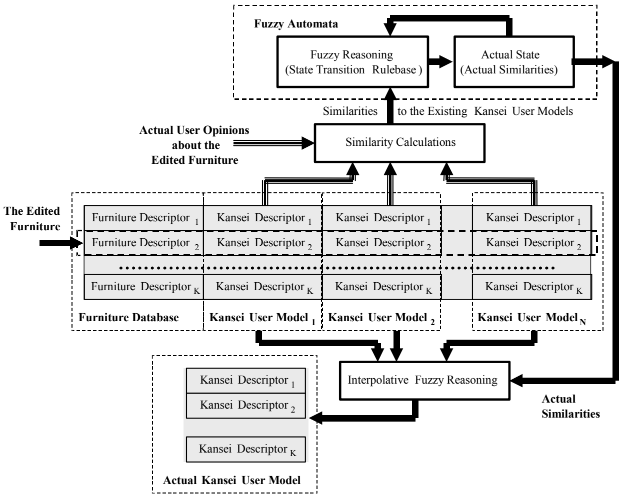

Fig. 11: Structure of the proposed adaptive user model generation.

3 Conclusions

The goal of this paper was to introduce a simple and flexible similarity based reconfiguration structure and some of its possible application areas, a behaviour-based vehicle control, a fault tolerant control and a user adaptive emotion-based (Kansei) selection system application. The main benefits, both the simplicity and the situation adaptivity of the proposed structure (similarly to some other behaviour-based control structures) are inherited from its hierarchical construction. In case of our first and second application example, this hierarchy has the meaning of building a (more) global strategy from some relevant, but only partially valid (with respect to the state space of the system) strategies. The proposed structure is simply combining (fusing) these strategies in interpolative manner to form one strategy, which has an extended area of validity (at least in a part of the area covered by the original partially valid strategies). This way a rather complicated strategy can be modularly built. Moreover because of the similarity based interpolative manner of strategy combination, there are some chance to get valid strategy in the area outside the area covered by the original strategies too (as it was showed in our second example application). E.g. we can study and handle only the relevant characteristic situations of a system (some kind of situations need significantly different handling) and let the similarity based reconfiguration to handle all the other situations by interpolation. Usually it is not the case when we try to combine neighbouring, or overlapping, but alternative (from the view of interpolation - contradictive) strategies. The problem of behaviour-based control in case of alternative strategies is studied in [4], [1]. The benefit of adapting fuzzy automata for system state (similarity) approximation in the proposed structure is to give (state) memory to the system. On one hand this memory is needed for the correct symptom evaluation (see the second example), or able to hold a kind of "history" information as it was introduced in the first example (left, or right turning tendency strategy decision). On the other hand, in case of adaptive applications, the system state can be viewed as the model of the actual situation, or the surrounding environment of the system from the viewpoint of the adaptive strategy. In the case of the third example the system state is the model of the actual user with respect to the existing user models. Having rule-based state-transitions of the fuzzy automata, it is very simple to built even relatively complicated state-transition structures. In some sample situations it can be built to take the function of the symptom evaluation (in case of rule-based symptom evaluation it could be simply combine with the state-transition rulebase), or to be able to handle the "history" of the system (as it was done in the first example). The main drawback of the proposed structure is the lack of alternative strategies handling ability. This problem is inherited from the similarity based interpolative manner of strategy combination. Having more, but different valid strategy, fitting the same situation, means a kind of contradiction from the viewpoint of interpolative conclusion fusion. On the other hand this problem is never appearing in contradiction, or alternative solution free situations, as we had in the sample applications introduced in this paper.

Acknowledgements

This research was partly supported by the Hungarian National Scientific Research Fund grant no: F 029904, and the Hungarian Ministry of Culture and Education grant no 0422/1997.

References

[1] P. Pirjanian, "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).

[2] J. Kosecka and R. Bajcsy "Discrete Event Systems for Autonomous Mobile Agents", Proceedings of Intelligent Robotic Systems '93, Zakopane, Czechoslovakia, pp.21-31, (1993).

[3] A. Saffiotti, K. Konolige and E. H. Ruspini, "A multivalued logic approach to integrating planning and control", Artificial Intelligence, **76**, pp.481-526, (1995).

[4] J. Yen and N. Pfluger, "A Fuzzy Logic Based Extension to Payton and Rosenblatt's Command Fusion Method for Mobile Robot Navigation", IEEE Transactions on Systems, Man, and Cybernetics, **25**(6), pp.971-978, (1995).

[5] L.T. Kóczy and K. Hirota, "Interpolative reasoning with insufficient evidence in sparse fuzzy rule bases", Information Sciences, **71**, pp. 169-201, (1992).

[6] 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, pp.144-149, (1997).

[7] J.C. Bezdek, "Pattern Recognition with Fuzzy Objective Function", Plenum Press, New York, (1981).

[8] Sz. Kovács and L.T. Kóczy, "Application of the Approximate Fuzzy Reasoning Based on Interpolation in the Vague Environment of the Fuzzy Rulebase in the Fuzzy Logic Controlled Path Tracking Strategy of Differential Steered AGVs", Lecture Notes in Computer Science, **1226**, pp.456-467. Springer, Germany, (1997).

[9] Sz. Kovács and L.T. Kóczy, "Path Tracking and Collision Avoidance Strategy of an AGV Implemented on Interpolation-based Fuzzy Logic Controller", Proceedings of the INES'98 IEEE International Conference on Intelligent Engineering Systems, Vienna, Austria, pp.67-72, (1998).

[10] Sz. Kovács, "Similarity Based System Reconfiguration by Fuzzy Classification and Hierarchical Interpolate Fuzzy Reasoning", Lecture Notes in Computer Science, **1625**, Springer, pp.12-19, (1999).

[11] Sz. Kovács, N. Kubota, K. Fujii and L.T. Kóczy, "Interpolative Fuzzy Reasoning and Fuzzy Automata in Kansei Technology", To appear in the proceedings of the AFSS2000, the Fourth Asian Fuzzy Systems Symposium, Tsukuba, Japan, p.6, (2000).