Interpolative Fuzzy Reasoning and Fuzzy Automata in Adaptive System Applications

Szilveszter Kovács

Gifu Prefectural Research Institute of
Manufacturing Information Technology
4-179-1 Sue Kakamigahara Gifu,
509-0108 Japan
E-mail: szkovacs@gifu-irtc.go.jp

On leave from:
Department of Information Technology,
University of Miskolc
Miskolc-Egyetemváros,
Miskolc, H-3515, Hungary
E-mail: szkszilv@gold.uni-miskolc.hu

Introduction

Difficulties: Control **partially known** complex systems.

Solution: Fusing partially valid control strategies in the function of their **suitability**:

Behaviour-based control structures

An interpolative view: "**The more similar the actual situation** to one of the **known partial strategy prerequisites**, **the more similar the strategy used to that strategy must be**."

Similarity based strategy reconfiguration

The main tasks to solve:

***The actual system-state approximation**:* The actual system-state – 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* – must be determined.

***The fusion of the existing partial strategies**:* based on the approximated system-state - *in the function of their suitability* – the *conclusions* (proposed actions) of the known partial strategies *must be fused*.

For the first task, we propose the adaptation of ***fuzzy automata***, where the state variables are the corresponding similarities, and the state-transitions are driven by fuzzy reasoning.

For the second task, the application of ***interpolative fuzzy reasoning*** is suggested.

Introduction

One view of **hybrid control** or **behaviour-based control** systems is a kind of **strategy reconfiguration** – an **intelligent adaptation of the system to the actual situation**.

Hybrid control –

Actual situation dependent discrete strategy (controller) **changing**.

Behaviour-based control systems –

Discrete **switching** to the **most appropriate strategy** or a kind of **fusion** of the strategies, which appeared to be the **most appropriate ones** in an actual situation.

Similarity based strategy reconfiguration – (behaviour based structure)

The **main idea** of this view is the following:

"The **more similar** the actual situation to one of the known partial strategy prerequisites, the **more similar** the strategy used to that strategy **must be**".

Known partial strategy, is an existing strategy (controller, or perception-action units in behaviour-based view), which has a partial validity only – with respect to the whole state space of the system.

Similarity based strategy reconfiguration

Main tasks to solve (similarly to behaviour-based control using behaviour fusion)

– Decision about the **levels of necessities of the different strategies**, which startegies are needed in an actual situation (it is also called the “action selection” or “behaviour co-ordination problem”).

– The **way of the startegy 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**. These similarities can be interpreted as the level of necessities of the corresponding strategies needed to handle the actual situation.

The **second task** is the conclusion fusion of the known partial strategies based on their necessities (their prerequisite’s similarities to the actual situation).

Here we suggest a strategy reconfiguration structure, which is based on **fuzzy interpolative fusion** of different strategies in the function of their actual necessity approximated by a **fuzzy automata**.

The system-state approximation

To approximate the **level of similarities** of the **actual system behaviour** to the **prerequisites of all the known strategies** (the level of necessity and the type of the strategy needed to handle the actual system behaviour)

This step is a kind of ***symptom evaluation***:

Calculating the **similarity of the actual symptom** to the **known symptoms patterns** - to the **prerequisites of the known partially valid strategies**. (Symptom patterns characterising the systems states where the corresponding partial strategies are valid.)

Some possible methods for fuzzy logic symptom evaluation:

– Adopting fuzzy classification methods e.g. the **Fuzzy c-Means** fuzzy clustering algorithm (Bezdek, 1981), 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.

– In simple situations, the fuzzy logic symptom evaluation could be a **fuzzy rule based reasoning system** itself.

<u>Difficulties</u>: 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**.

<u>Solution</u>: the **adaptation of *fuzzy automata***.

Fuzzy Automata in system-state approximation

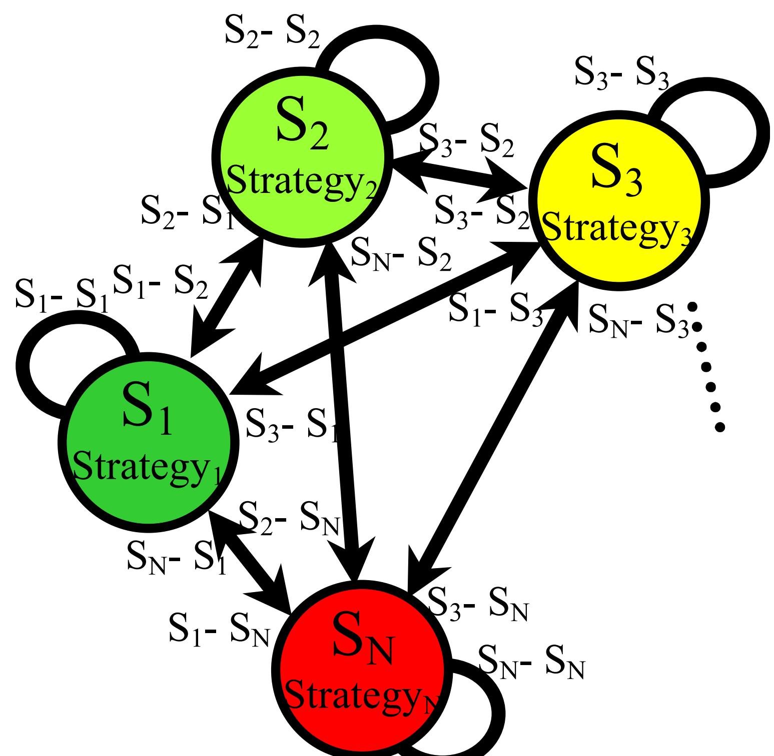
One solution of solving the problem of **state - actual control strategy – dependent symptom evaluation** is the **adaptation of *fuzzy automata***.

The **state vector** of the automata is the **approximated system-state**. (The vector of the approximated similarities of the actual situation to the prerequisites of the known strategies.) The **state transitions** are driven by **fuzzy reasoning**, based on the conclusions of the symptom evaluation.

The **basic structure of the rulebase** applied for the state-transitions of the fuzzy automata (rules for interpolative fuzzy reasoning) for the ith state S_i (R_{Ai}):

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.



This automata is a **fuzzy automata**, because its **state variables are fuzzy membership values** (similarities – infinite values) and the **state-transitions are driven by fuzzy reasoning**.

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.

The simplest way for such a combination, is the application of the **interpolative fuzzy reasoning**. The main idea of the similarity based strategy reconfiguration – *"The more similar the actual situation to one of the known partial strategy prerequisites, the more similar the strategy used to that strategy must be"* – can be directly translated to a fuzzy rulebase:

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** $\boldsymbol{y}=y_2$

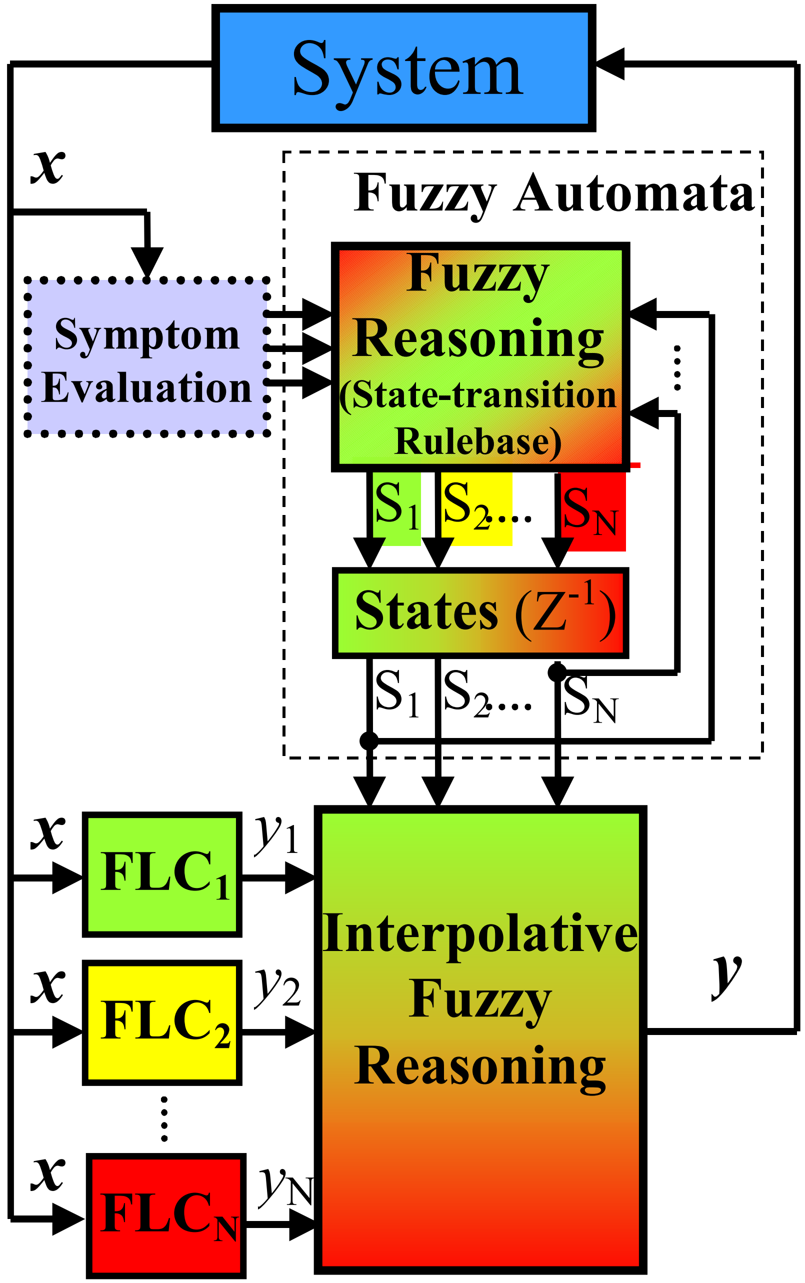
...

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

Benefits of interpolative fuzzy reasoning: **simple built** conclusion fusing rulebase (the rulebase is not needed to be complete) and the **needlessness of defuzzification** (in some cases).

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).

Basic structure of the Similarity Based Strategy Reconfiguration



Application examples

From the side of possible application areas, the view of situation adaptive strategy reconfiguration – the **situation adaptivity** can be very simply explained as ***environment adaptivity***.

This case the "environment" could have many practical interpretations.

From the **view of the controller** this **environment can be interpreted** as the

– ***current situation*** (common behavioural-based applications),

– or the ***controlled system*** itself (e.g. fault tolerant control applications),

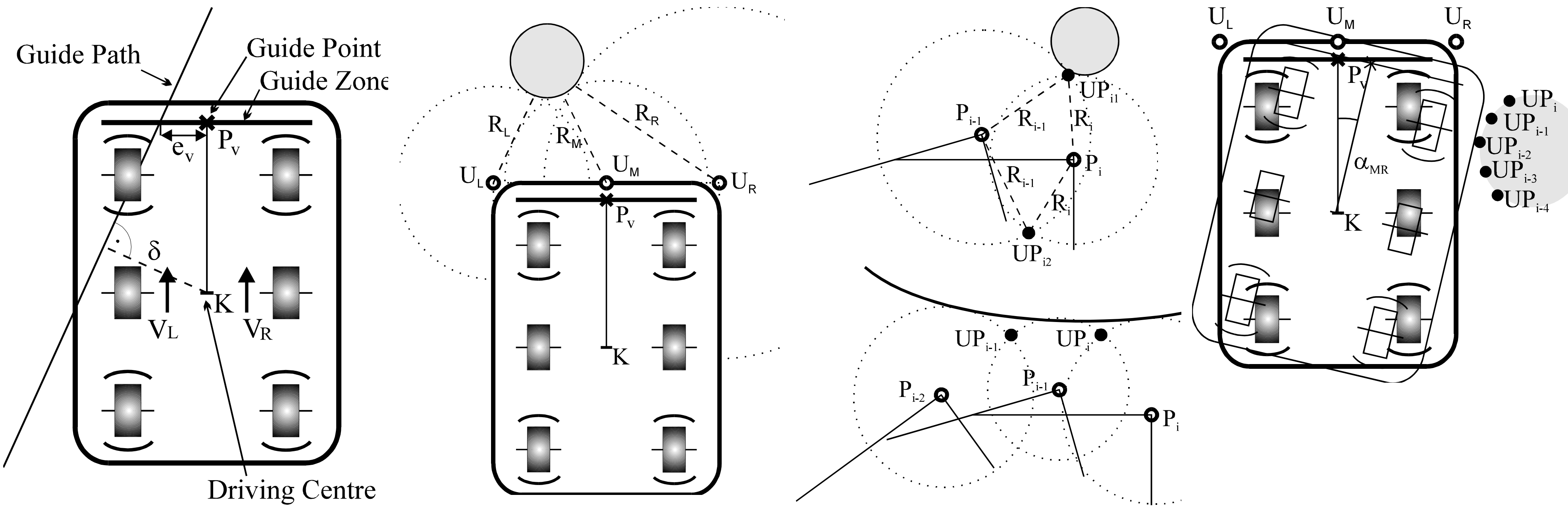
– or even the ***actual user*** himself/herself (user adaptive applications).

These interpretation differences can extend the application areas of the behavioural-based control structures to "common sense" adaptive system applications.

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 introduced in the followings.

Application example: Automated Guided Vehicle navigation

For checking the efficiency of the proposed similarity based strategy reconfiguration structure in an application example, a simulated **steering control of an automated guided vehicle (AGV)** is introduced. The steering control has **two main goal**s, 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.



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), UP is the *unsafe (risky) point*, α_{MR} is the *maximal right turning angle without side collision*.

The known partial strategies

The **first step** of similarity based reconfiguration is to **build the component partially valid strategies**:

Path tracking and restricted collision avoidance strategy. Its main goal is the path tracking (to follow a guide path) and as a sub goal, a restricted collision avoidance ("avoiding obstacles without risking the chance of loosing the guide path"). This simple strategy needs seven observations: the *estimated momentary path tracking error* (e_v), the *distance between the guide path and the guide point* (δ), 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 it has two conclusions, the *speed* (V_a - $\mathbf{R_{Va}}$ rulebase) and the *steering* (V_d - $\mathbf{R_{Vd}}$ rulebase).

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** $V_d=B_i$.

$\mathbf{R_{Vd}}$:	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 **labels of fuzzy sets** (linguistic terms).

The **interpretation** of these fuzzy sets can be different in each antecedent, consequent universe.

The collision avoidance 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 (**R_{Vd}**) and the speed (**R_{Va}**):

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*

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. These 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 (**R_{Vd}**):

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:

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

The symptom evaluation and the fuzzy automata

The example application is so simple, that 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: 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}). 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 partial strategies, the state variables themselves (S_P,S_C,S_{CR},S_{CL}), and an 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_C
					V			*S*			*L*
					V			*L*			*Z*
					NV						*Z*

$\mathbf{R_{SCR}}$:

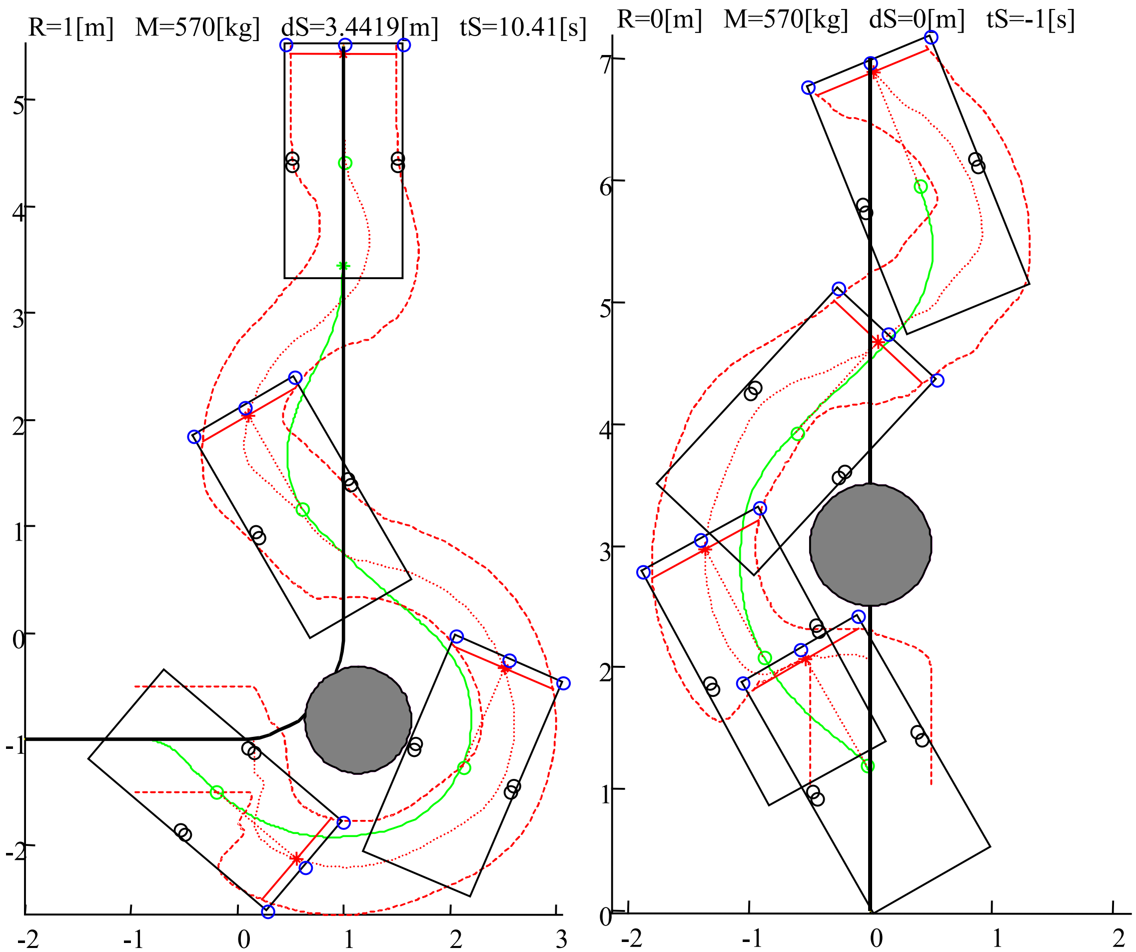
S_P	S_C	S_{CR}	S_{CL}	e_v	PV	R_L	R_R	R_M	α_{ML}	α_{MR}	S_{CR}
L				*NVL*	*V*						*L*
		L			*NV*						*L*
				Z	*V*			*L*			*Z*
			L								*Z*

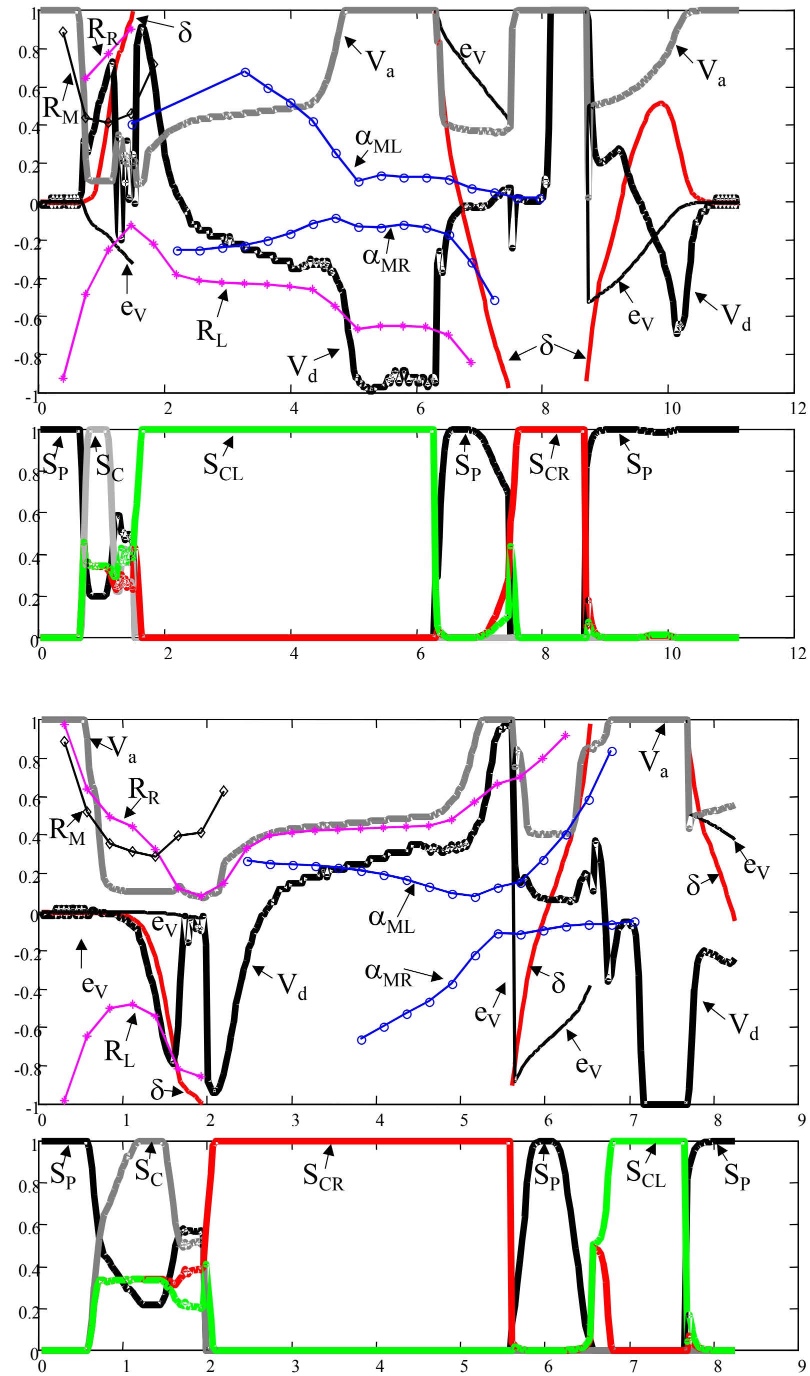
$\mathbf{R_{SCL}}$:

S_P	S_C	S_{CR}	S_{CL}	e_v	PV	R_L	R_R	R_M	α_{ML}	α_{MR}	S_{CL}
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.

Simulated results

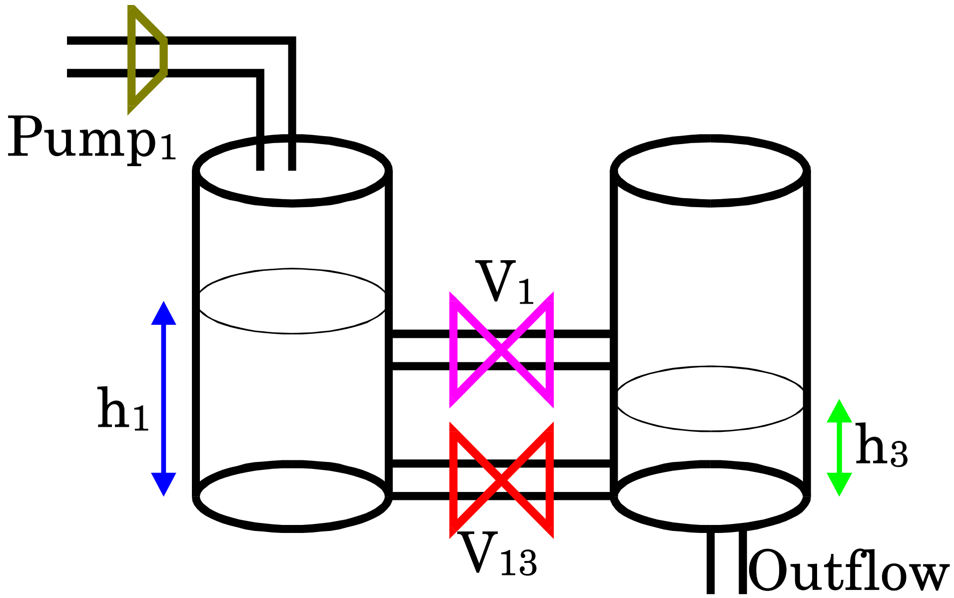




Path, and time function of observations, conclusions and system-state values

Fault diagnosis and reconfiguration of the three tank benchmark

As a simple demonstration, a simplified configuration (two tanks only) of the three tank benchmark was chosen.



Normal behaviour: The **goal of the control system** is to keep the water levels in tank$_1$ and tank$_3$ $\boxed{h_1 = 0.5}$ and $\boxed{h_3 = 0.1}$ by controlling the **valve$_{13}$** and the **pump$_1$** at a **constant value of outflow from tank$_3$**. Faults of the **valve$_{13}$**:

Fault no.1.: **valve$_{13}$** is **opened and blocked**, the water level in tank$_3$ $\boxed{h_3 = 0.1}$ could be controlled by **pump$_1$** (this case h_1 is changed)

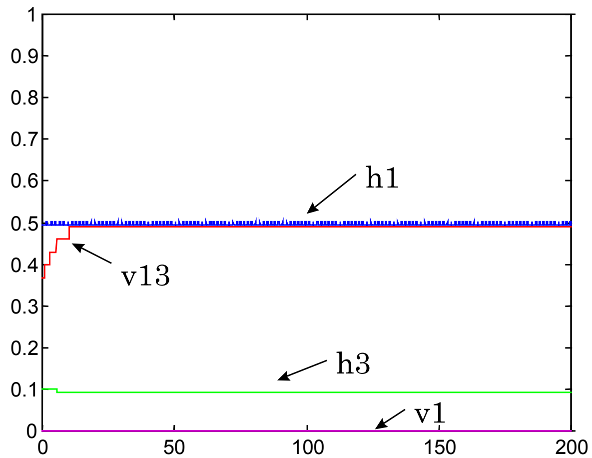
Fault no.2.: **valve$_{13}$** is **closed and blocked**, the water levels in tank$_1$ and tank$_3$ could be controlled by the **valve$_1$** and the **pump$_1$**

The main steps of Similarity Based System Reconfiguration:

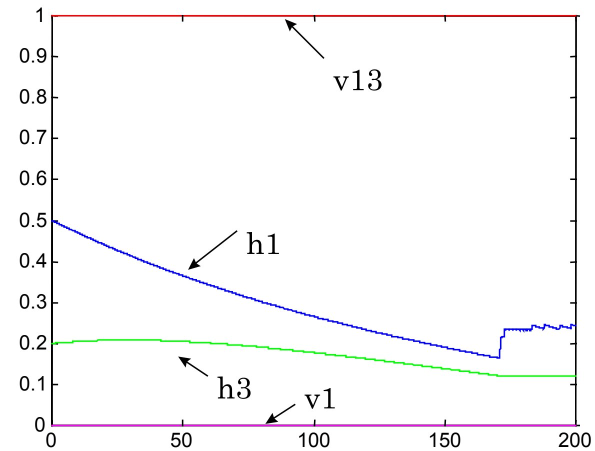
- producing the controllers for **handling the separate system behaviours by separate controllers** (interpolate fuzzy logic controllers),
- **collecting the syndromes characterising all the system behaviour classes - separately for all the studied control states** and **state transitions**,
- **generating the relevant symptom patterns** characterising the main system behaviour classes (applying the Fuzzy c-Means algorithm),
- **comparing and unifying the fuzzy partitions** (relevant symptom patterns),
- producing the **interpolate fuzzy logic controllers for the fuzzy automata and for the second hierarchical reasoning level**.

The simulated results of the different system behaviours (states and state transitions) direct handled by the specific controllers:

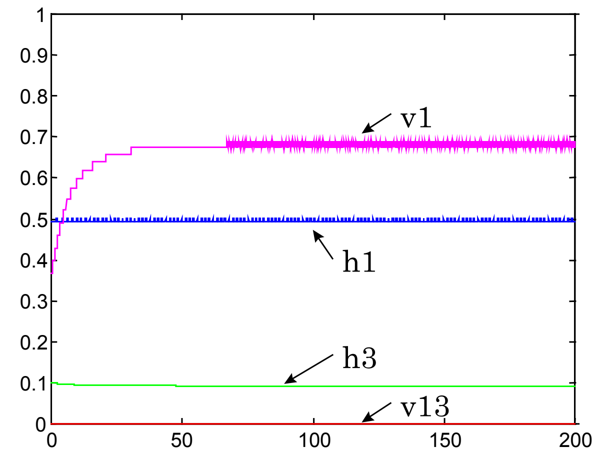
Handling normal behaviour



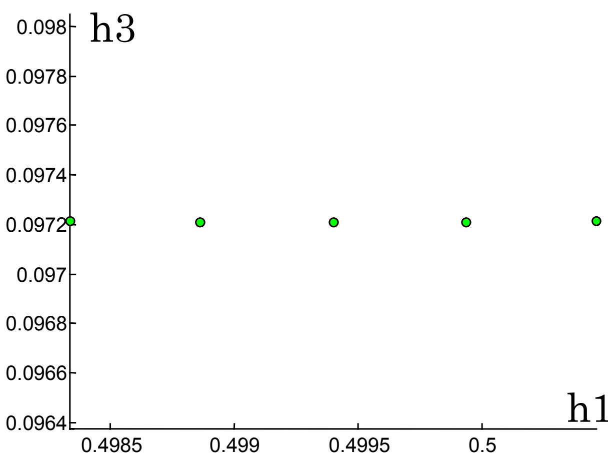
Handling fault 1 behaviour

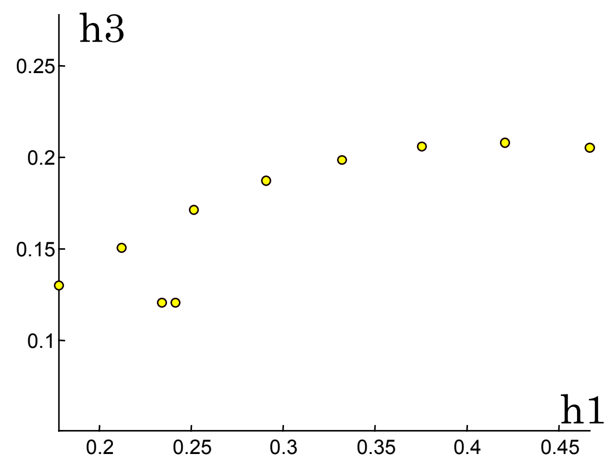


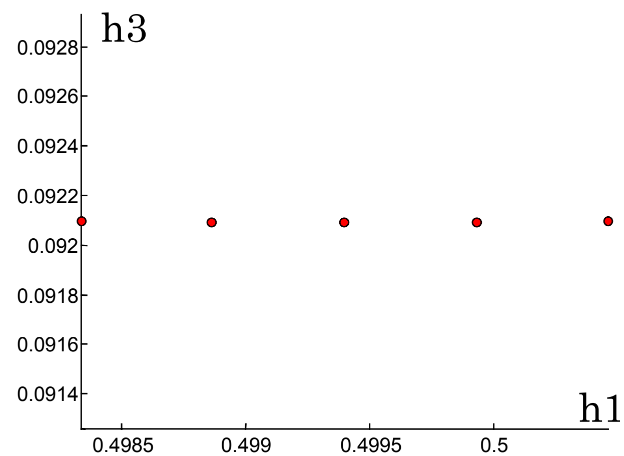
Handling fault 2 behaviour



The relevant symptoms:

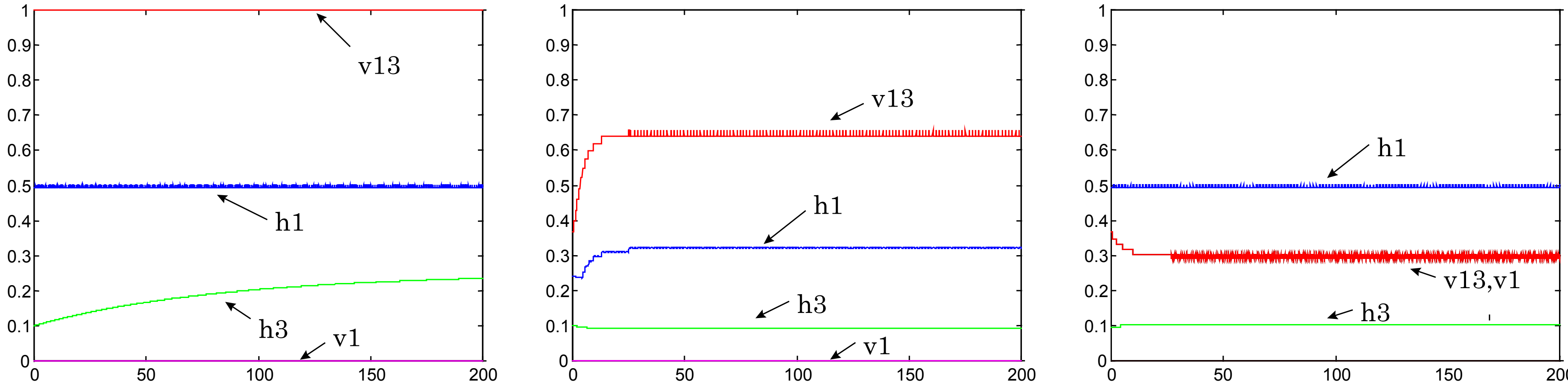




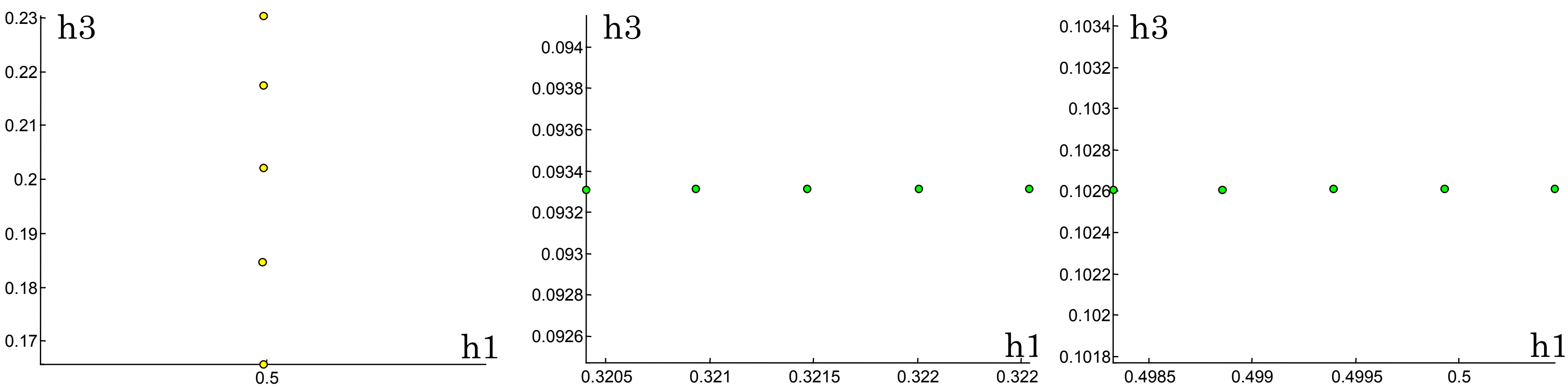


State transitions:

Fault 1 during normal control | Normal during fault 1 control | Normal during fault 2 control

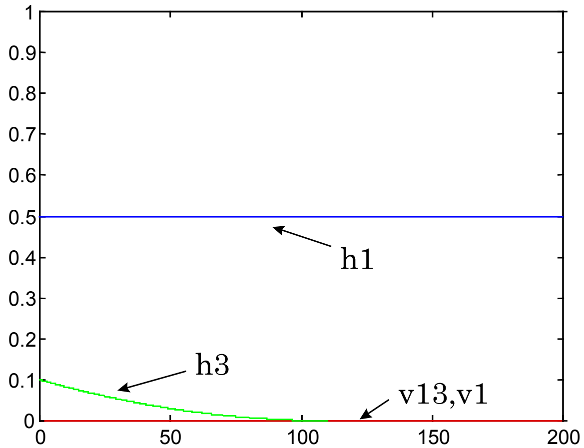


The relevant symptoms:

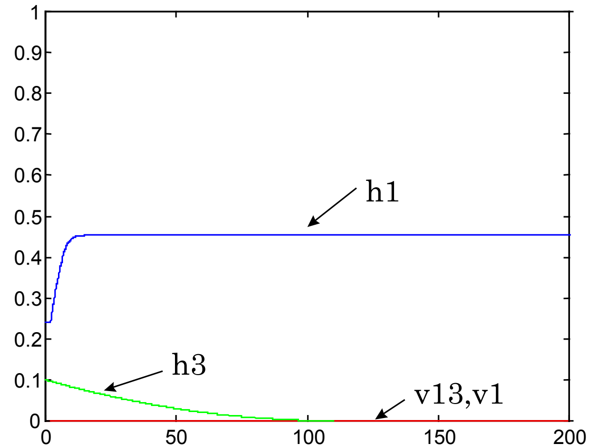


State transitions:

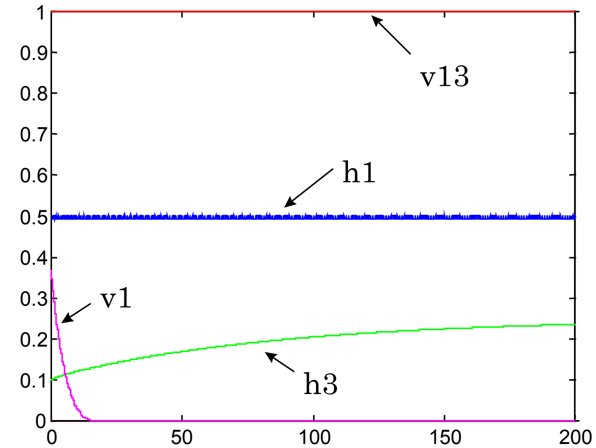
Fault 2 during normal control



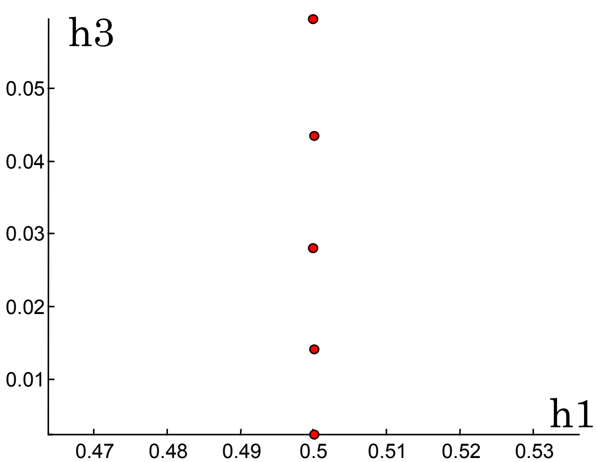
Fault 2 during fault 1 control

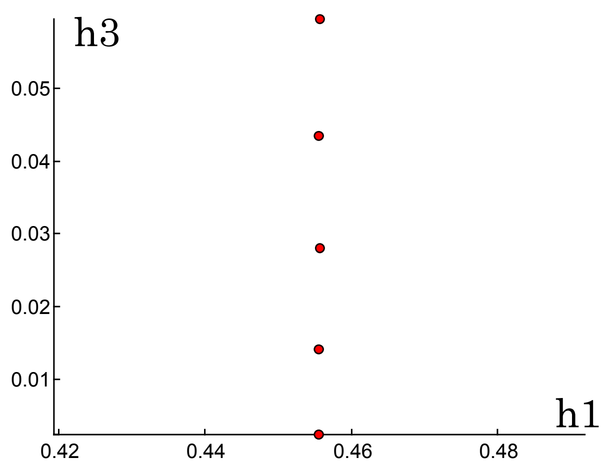


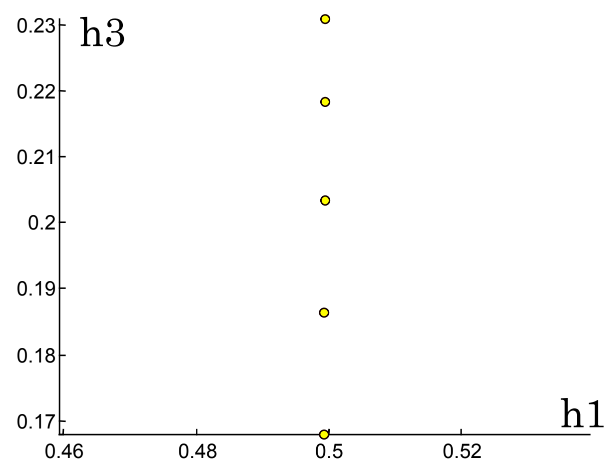
Fault 1 during fault 2 control



The relevant symptoms:



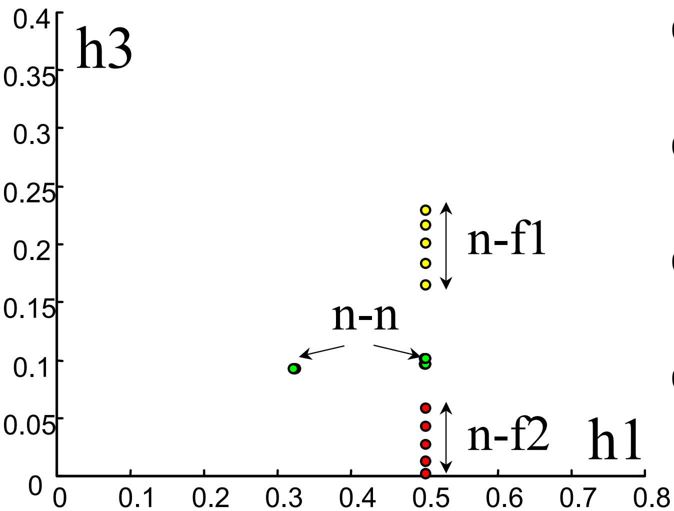


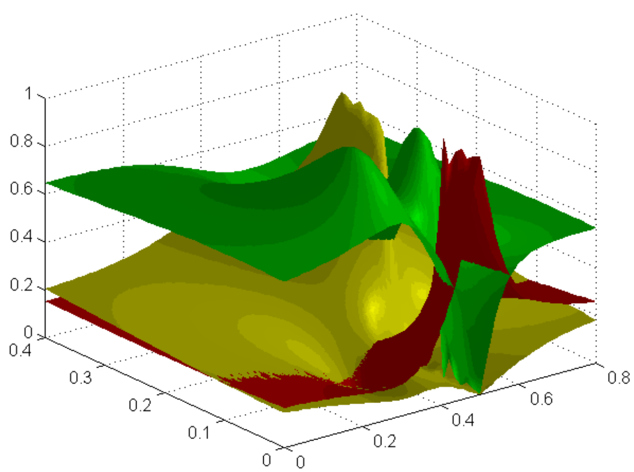


The relevant symptoms (unified clusters):

Handling normal behaviour

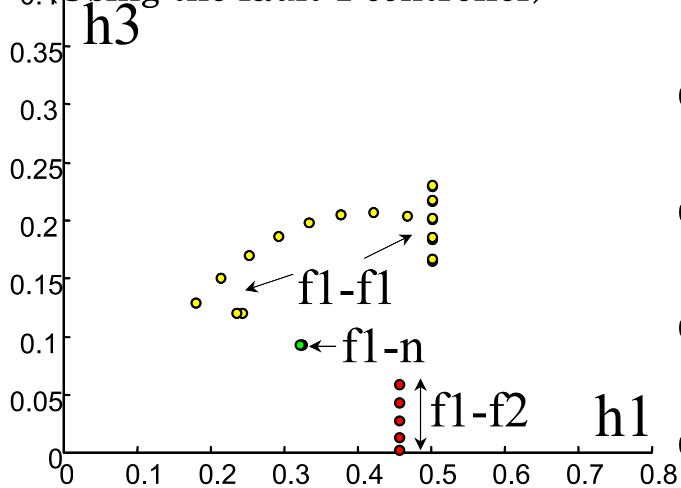
(Using the normal controller)

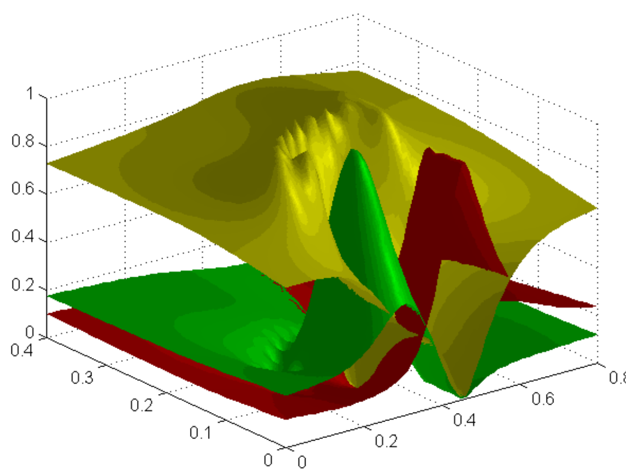




Handling fault 1 behaviour

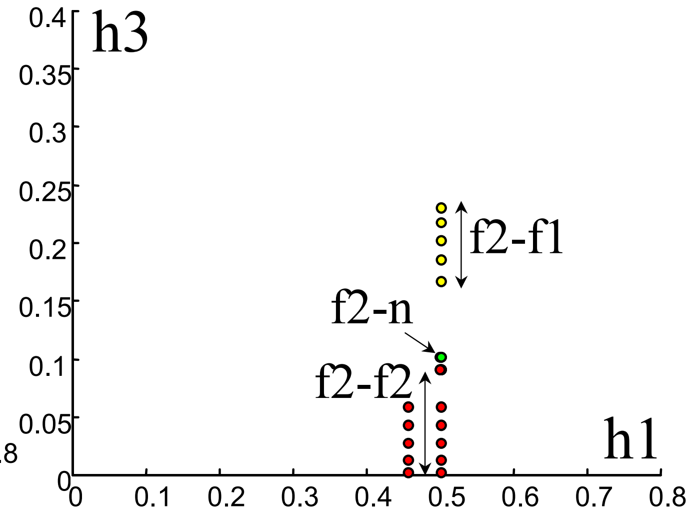
(Using the fault 1 controller)

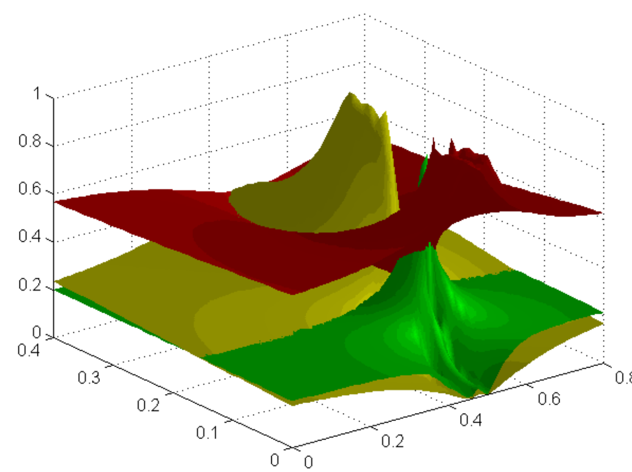




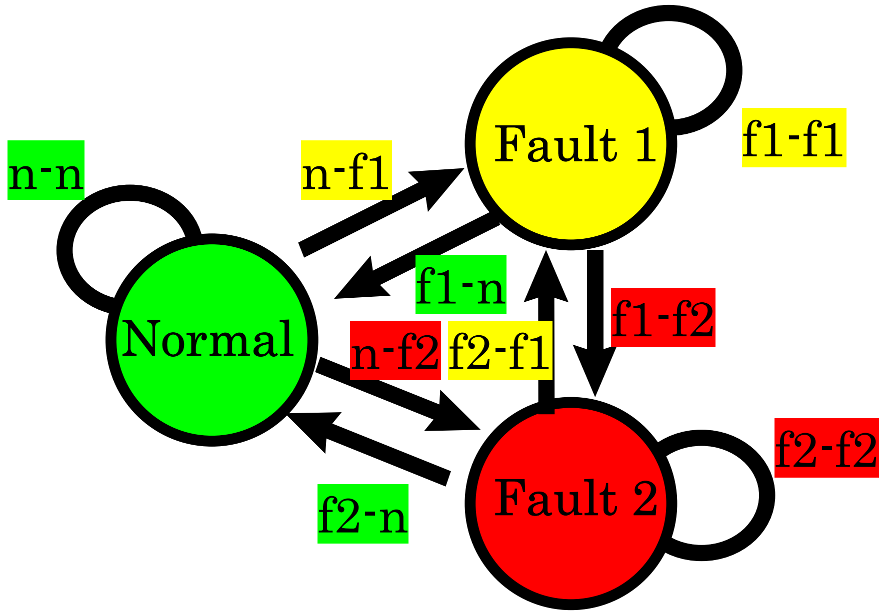
Handling fault 2 behaviour

(Using the fault 2 controller)





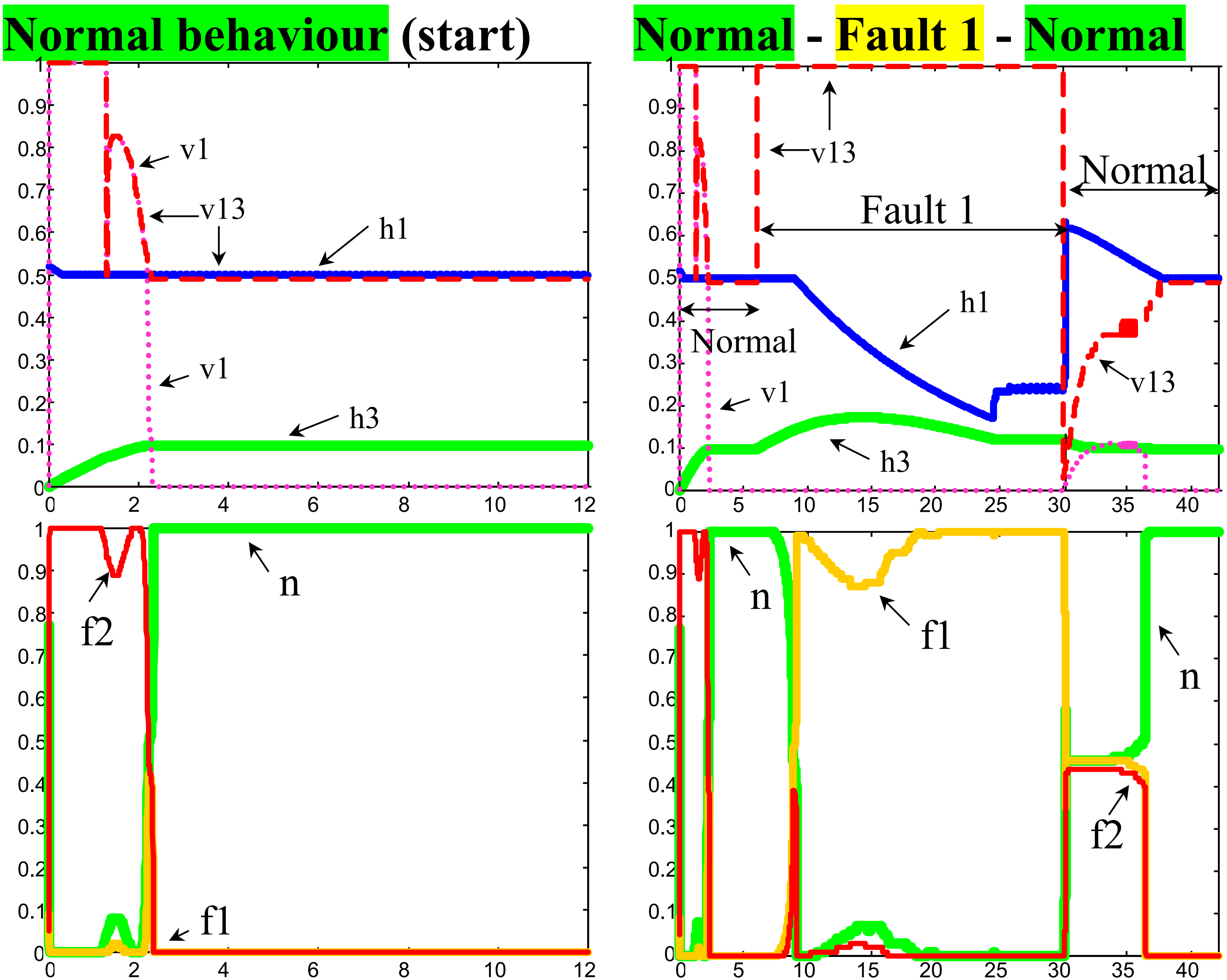
The state-transition diagram of the interpolate fuzzy automata:



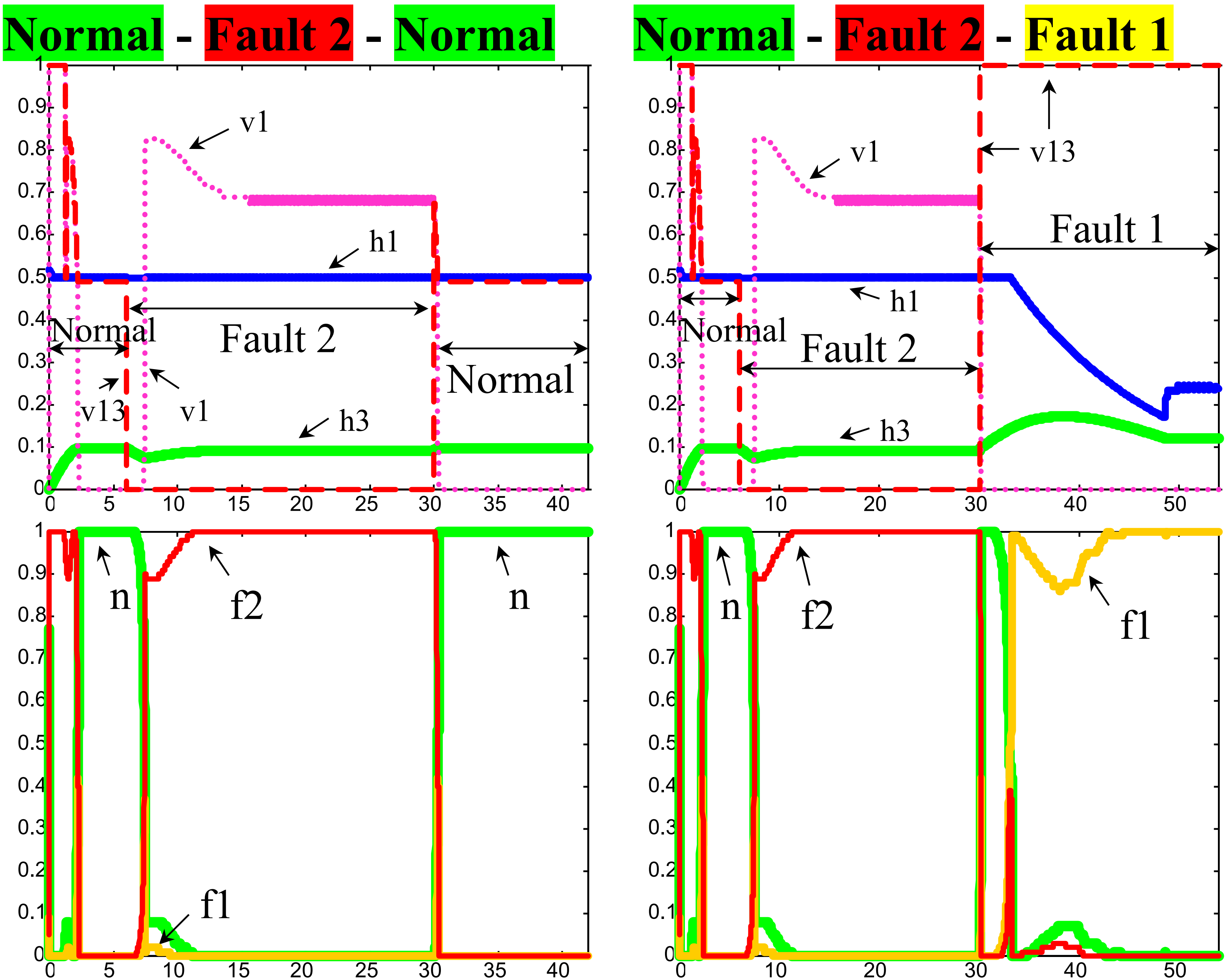
The rulebase applied for the fuzzy automata for the Normal state N (R_{AN}):

If	N = One	**And**	N-N = One	**Then**	N = One
If	N = One	**And**	$N\text{-}F_1$ = One	**Then**	N = Zero
If	N = One	**And**	$N\text{-}F_2$ = One	**Then**	N = Zero
If	F_1 = One	**And**	$F_1\text{-}N$ = One	**Then**	N = One
If	F_2 = One	**And**	$F_2\text{-}N$ = One	**Then**	N = One
If	F_1 = One	**And**	$F_1\text{-}N$ = Zero	**Then**	N = Zero
If	F_2 = One	**And**	$F_2\text{-}N$ = Zero	**Then**	N = Zero

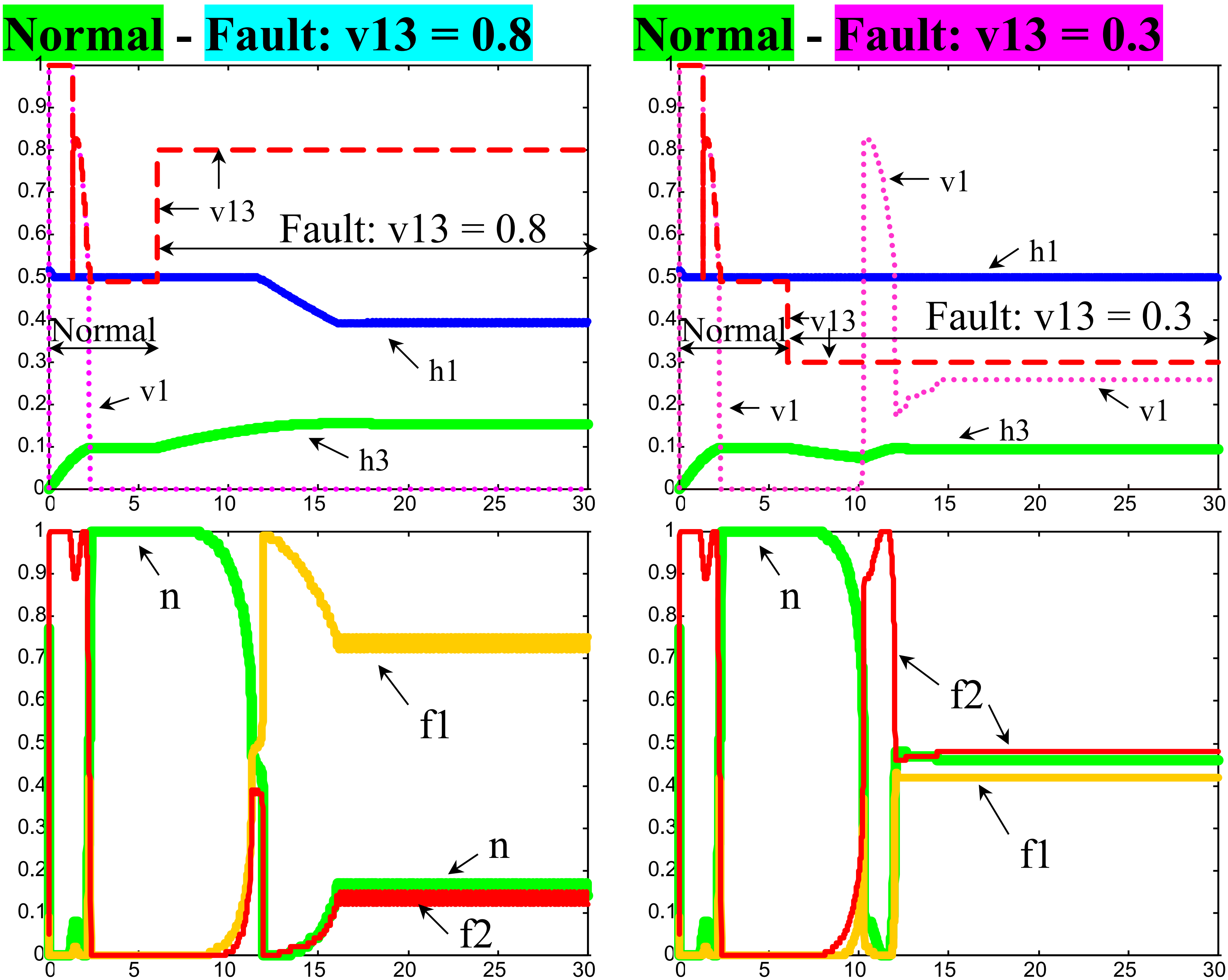
Where $\mathbf{F_i}$ **is the grade value of the** $\mathbf{F_i}$ **state** (the **approximated level of similarity of the actual system behaviour to the** $\mathbf{F_i}$ **state),** $\mathbf{F_i\text{-}N}$ **is the membership value of the from state** $\mathbf{F_i}$ **to state N state-transition** calculated by the fuzzy syndrome evaluation unit (Fuzzy c-Means)



Normal behaviour (start)
Normal - Fault 1 - Normal
v1
v13
h1
h3
n
f1
f2
Fault 1
Normal



Normal - Fault 2 - Normal
Normal - Fault 2 - Fault 1
v1
h1
Normal
Fault 2
Normal
v13
v1
h3
n
f2
n
f1
v13
v1
Fault 1
h1
Normal
Fault 2
h3
n
f2
f1



Normal - Fault: v13 = 0.8
Normal - Fault: v13 = 0.3
v13
Fault: v13 = 0.8
Normal
h1
v1
h3
n
f1
n
f2
v1
h1
Fault: v13 = 0.3
Normal
v13
v1
v1
h3
n
f2
f1

The adaptive Kansei user model

Application of **interpolative fuzzy reasoning** and **fuzzy automata** in **Kansei Technology** gives a simple way for adding **user adaptivity to emotion-based selection systems**.

One way of handling user adaptivity, is a kind of **combination of existing** (off-line collected valid) **human opinions** in the **function of the approximated similarity** to the actual user opinions **to get the approximated actual user opinions**.

Main idea

More similar the actual user to one of the existing user models, more similar must be the actual user model to that user model.

Tasks to solve

- Approximating the similarities of the actual user opinions to the off-line collected user opinions (existing user models) - **fuzzy automata**
- Combine the off-line collected opinions (existing user models) in the function of the corresponding approximated similarities - **interpolative fuzzy reasoning**

Approximating the similarities of the actual user to the existing user models

$\Rightarrow$ **Fuzzy automata**

State: A **set of similarity values**, the actual approximated similarities of the actual user and the existing user opinions.

State-transitions: Are driven by **fuzzy reasoning (Fuzzy state transition rulebase)** as a decision based on the **previous actual state** (similarities) and the **similarities** of an editing actual user opinion to the existing user opinions.

State-transitions rulebase 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**	SS_i=One	**And**	S_k=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**	SS_i=One	**And**	S_k=Zero	**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 rules is similar for all the states.

(Incomplete rulebase - interpolative fuzzy reasoning)

Combining the existing Kansei user models based on the actual similarities

$\Rightarrow$ Interpolative fuzzy reasoning

Main idea

More similar the actual user to one of the existing user models, more similar must be the actual user model to that user model

- **can be directly translated to an interpolative fuzzy rulebase**:
  (Completeness off the fuzzy rulebase is not necessary)

$$\textbf{If } S_1=\text{One} \textbf{ And } S_2=\text{Zero} \textbf{ And } \ldots \textbf{ And } S_N=\text{Zero} \textbf{ Then } KD=KD_1$$

$$\textbf{If } S_1=\text{Zero} \textbf{ And } S_2=\text{One} \textbf{ And } \ldots \textbf{ And } S_N=\text{Zero} \textbf{ Then } KD=KD_2$$

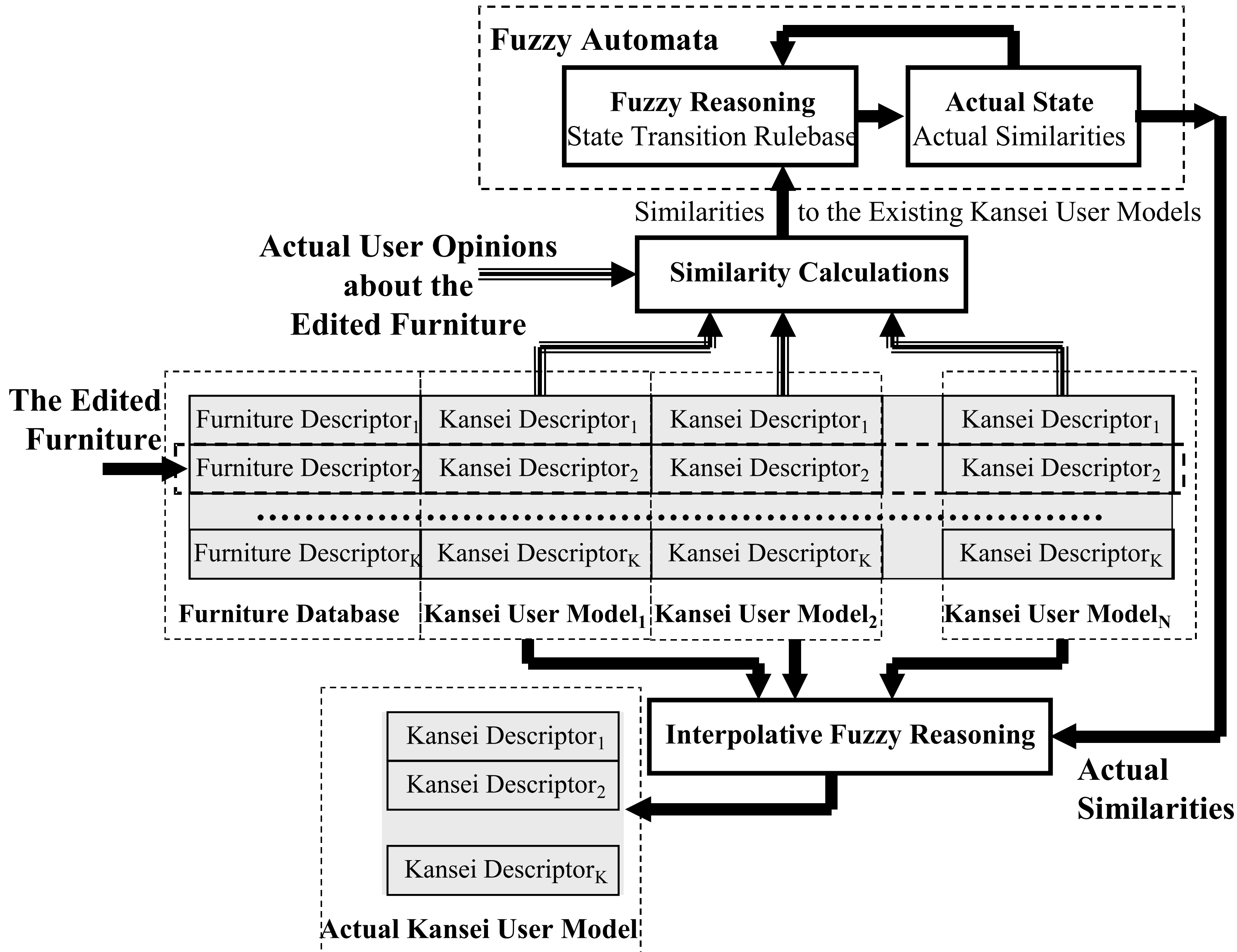
$$\ldots$$

$$\textbf{If } S_1=\text{Zero} \textbf{ And } S_2=\text{Zero} \textbf{ And } \ldots \textbf{ And } S_N=\text{One} \textbf{ Then } KD=KD_N$$

for all the Kansei descriptors in a user model, where KD_i is the set of Kansei descriptors in the i^{th} user model, and KD is the set of Kansei descriptors of the actual Kansei user model we are searching for.

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).

Structure of the proposed adaptive Kansei user model generation



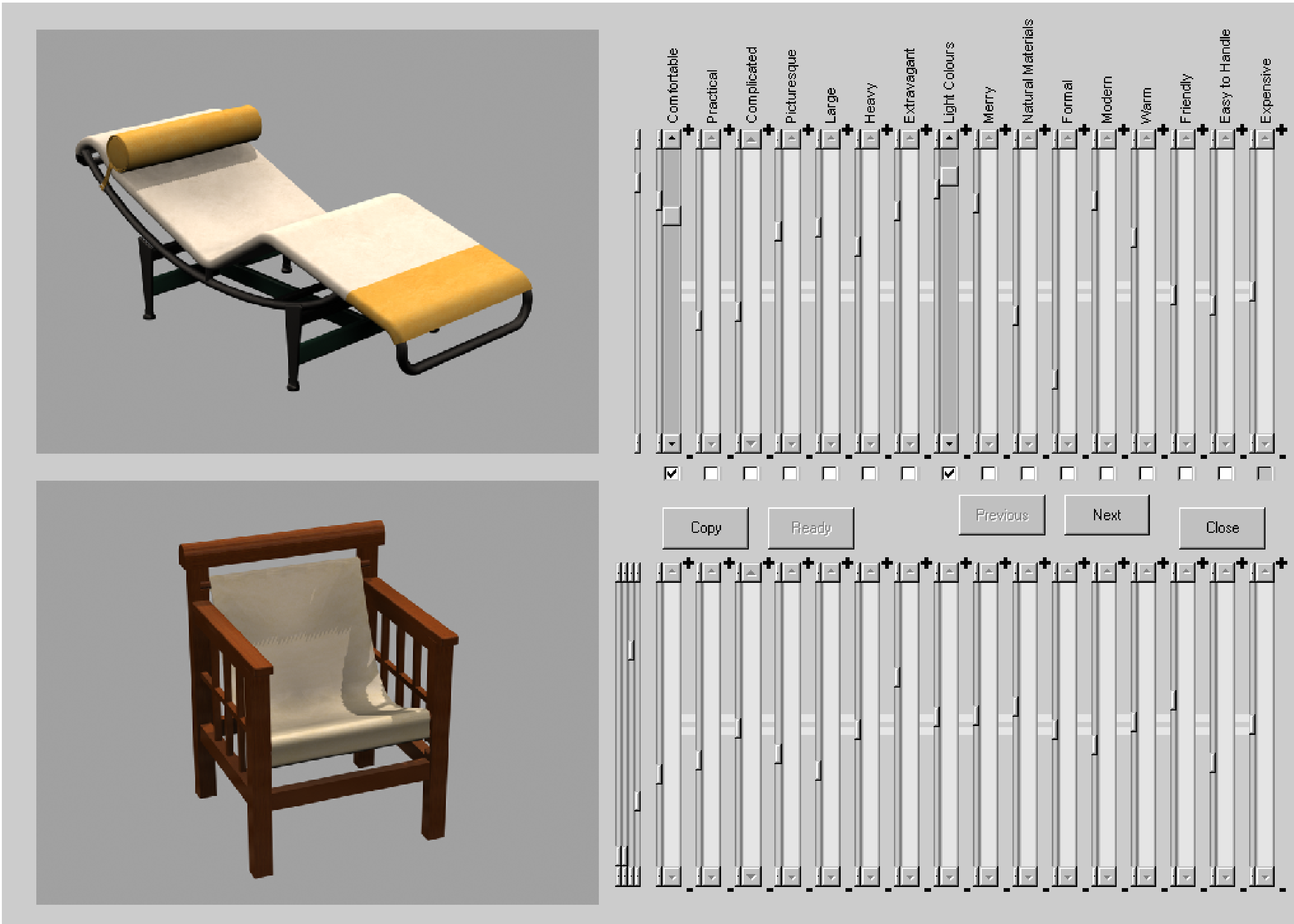
The user adaptive furniture selection

As an example of the proposed adaptive Kansei user model structure, a Kansei furniture selection system was developed:

The goal of the selection system:

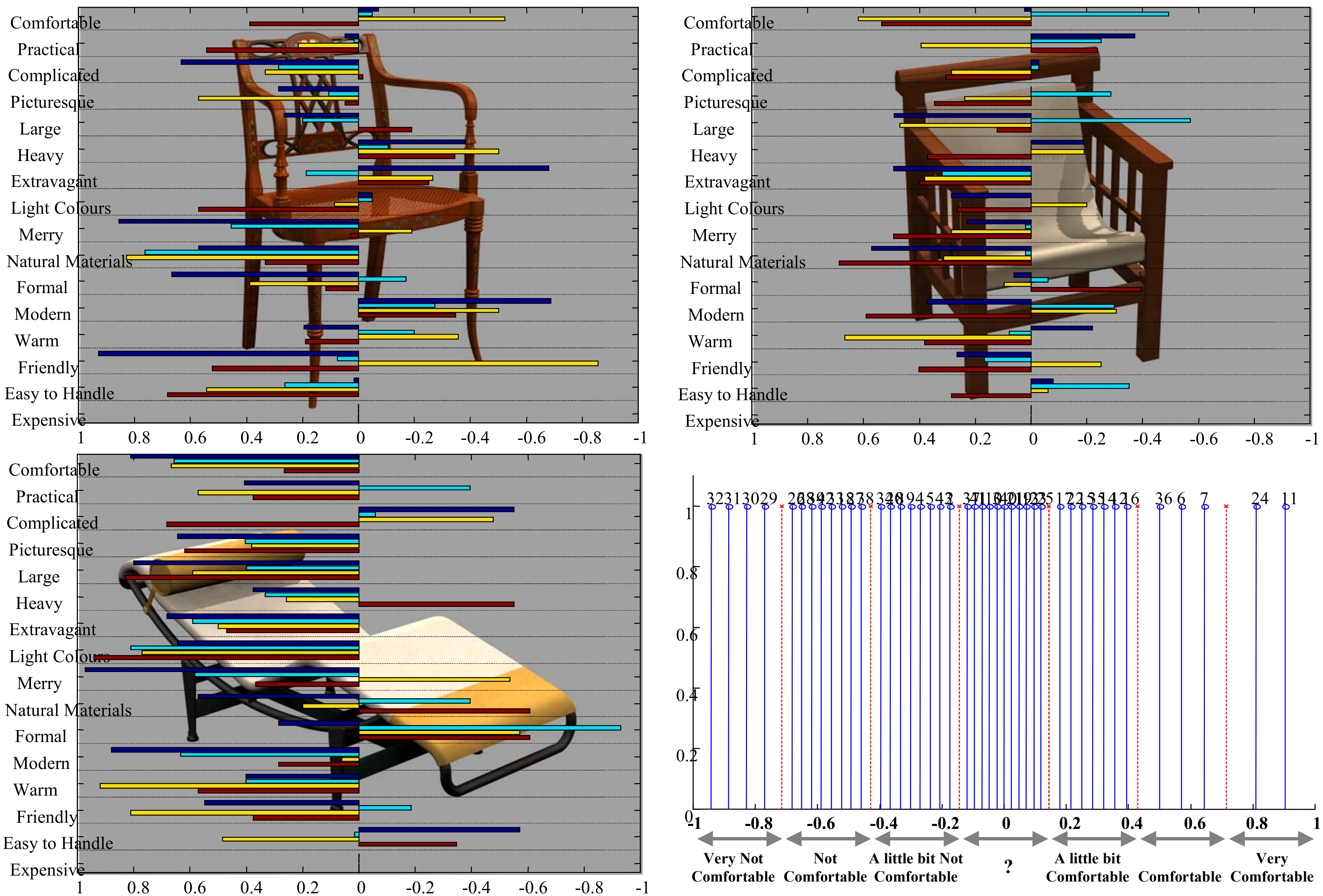
- To aid furniture (chair) selection by giving the chance to the user to **express his/her requirements through emotional (Kansei) levels**.
- The set of handled emotions is fixed to **16 emotional words related to chairs**.
- The user is **giving the requirements** by selecting some of the emotional words and **adjusting the corresponding sliders**. (On the sliders the “+”, “0”, “-” symbols are appearing only, to inspire the user to **give his/her feelings in a scaleless manner**.)
- As a response of the user intervention, **the best fitting chair is appearing** in the working window. The same time **the system gives all the Kansei values** (16 in our case) related to the furniture on screen, fetched from the actual Kansei user model. **These values are appearing the same manner, on sliders** (side by the user sliders), as the user was giving his/her requirements. **This method inspires the user to make modifications in more/less, small/big differences manner – relative to the furniture on screen**.
- In the case the **user is disagree with the evaluation given by the system**, **he/she can give his/her opinions** by copying the actual furniture to the editing window and adjusting some of the bottom sliders. Pressing the Ready button, the system recalculates the actual similarities.

Screenshot of the Kansei furniture selection system



The Kansei user models

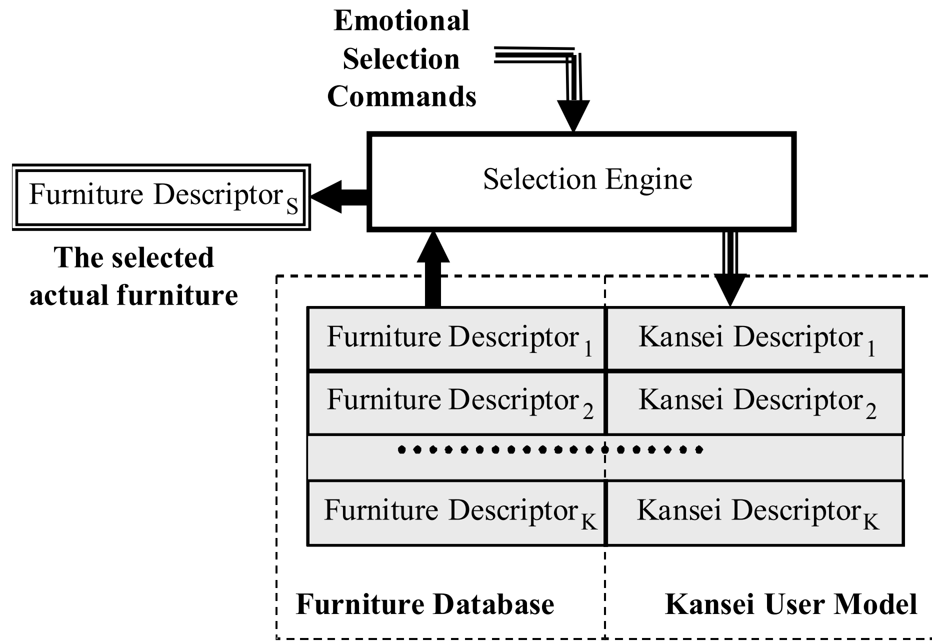
- The existing Kansei user models were generated **based on questionnaires**.
- The inquired persons had to make a **partial ordering of a set of pictures** of 43 different chairs.
- For each emotional (Kansei) attributes in the questionnaire, the inquired persons were **first asked to make a rough order of the pictures into seven groups: very ~, ~, a little bit ~, ?, a little bit not ~, not ~, very not ~ - where ~ is the actual Kansei attribute**. Than he/she was asked to **partially order the pictures of the same groups**. (Partially ordering was meant as ordering in the case of the pictures are distinguishable in respect to the Kansei attribute, and signing equality, if they are indistinguishable.)
- The answers than **translated to real values of the [-1,1] interval**, according to equal width of the seven attribute group, and equal distances of the elements of the same group in the manner of partial ordering (equal values for the indistinguishable ones). These values are forming the **Kansei descriptors**.



Comfortable
Practical
Complicated
Picturesque
Large
Heavy
Extravagant
Light Colours
Merry
Natural Materials
Formal
Modern
Warm
Friendly
Easy to Handle
Expensive
1 0.8 0.6 0.4 0.2 0 -0.2 -0.4 -0.6 -0.8 -1
-1 -0.8 -0.6 -0.4 -0.2 0 0.2 0.4 0.6 0.8 1
0 0.2 0.4 0.6 0.8 1
Very Not Comfortable
Not Comfortable
A little bit Not Comfortable
?
A little bit Comfortable
Comfortable
Very Comfortable
36 6 7 24 11

The selection system

- The task of the selection engine is to select the **furniture descriptors** from the furniture database which have the **closest actual Kansei descriptor to the user requirements**.
- The similarities are calculated as distances in **Euclidean sense**. Having a user selection command, the best fitting (closest) furniture is put on screen. Than the user can use the Next (Previous backward) button to view the next best fitting furniture.



The similarity calculations

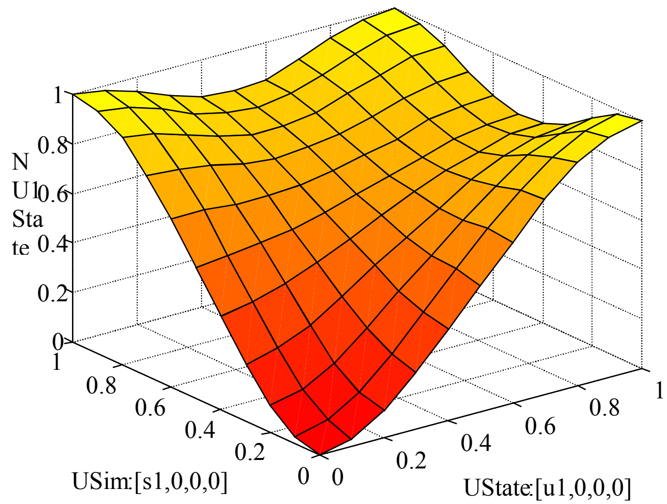
- In the case the user is disagree with the evaluation given by the system**, the similarities (*SS_i*) of the given user opinions and the ith existing Kansei user model** is calculated using the following formula (applying functions of the **Fuzzy c-Means fuzzy clustering algorithm**):

$$SS_i = \frac{1}{\sum_{j=1}^{N} \left(\frac{d_i}{d_j} \right)^{\frac{2}{m-1}}}$$

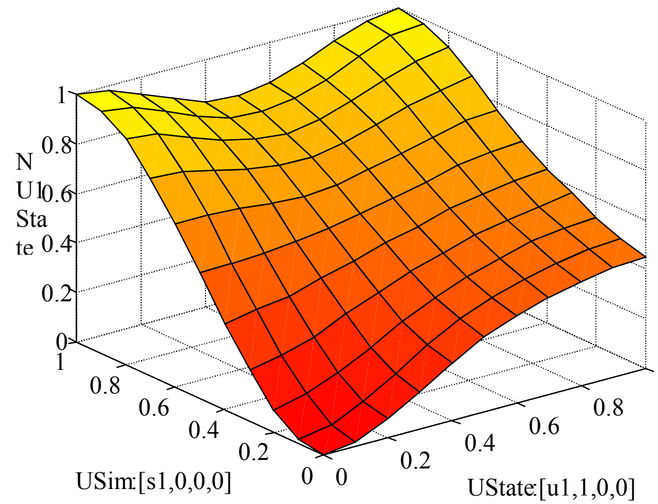
where $d_k = \|\mathbf{x} - \mathbf{v}_k\|$, the **distance** (**measure of dissimilarity**) of the user opinions (Kansai descriptors) **x** and the Kansei descriptors of the edited furniture in the kth existing Kansei user model $\mathbf{v}_k$, *m* is a weighting exponent (usually *m*=2).

The user similarities approximation, the fuzzy automata

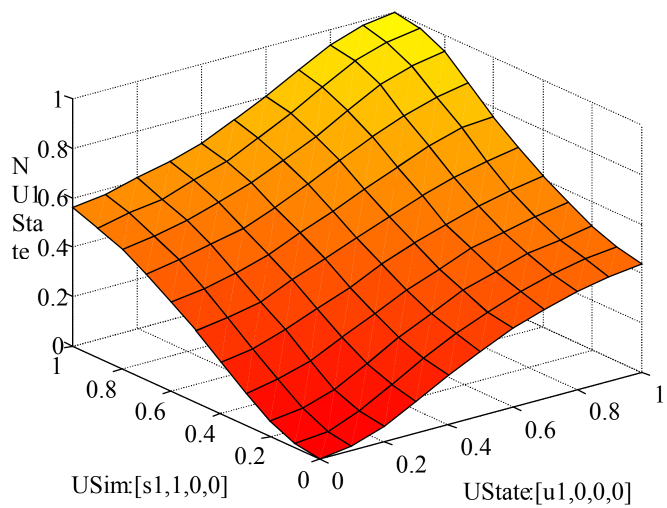
- The **initial state** (initial value of the actual similarities) is **a vector of 0.5**.
- Some **i^th^ state-transition surfaces** of the fuzzy automata $\mathbf{S_i(SS_i,S_{i-1})}$:



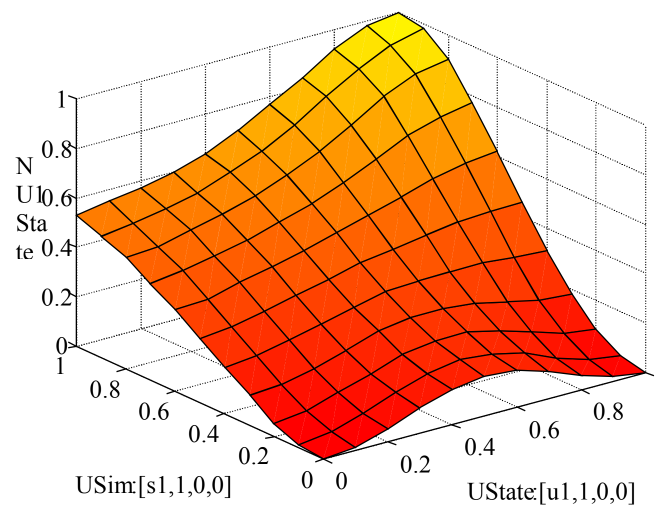
$S_{k-1}=0,\ SS_k=0,\ \forall k \neq i$



$S_{k-1}=0,\ SS_k=1,\ \exists k \neq i$



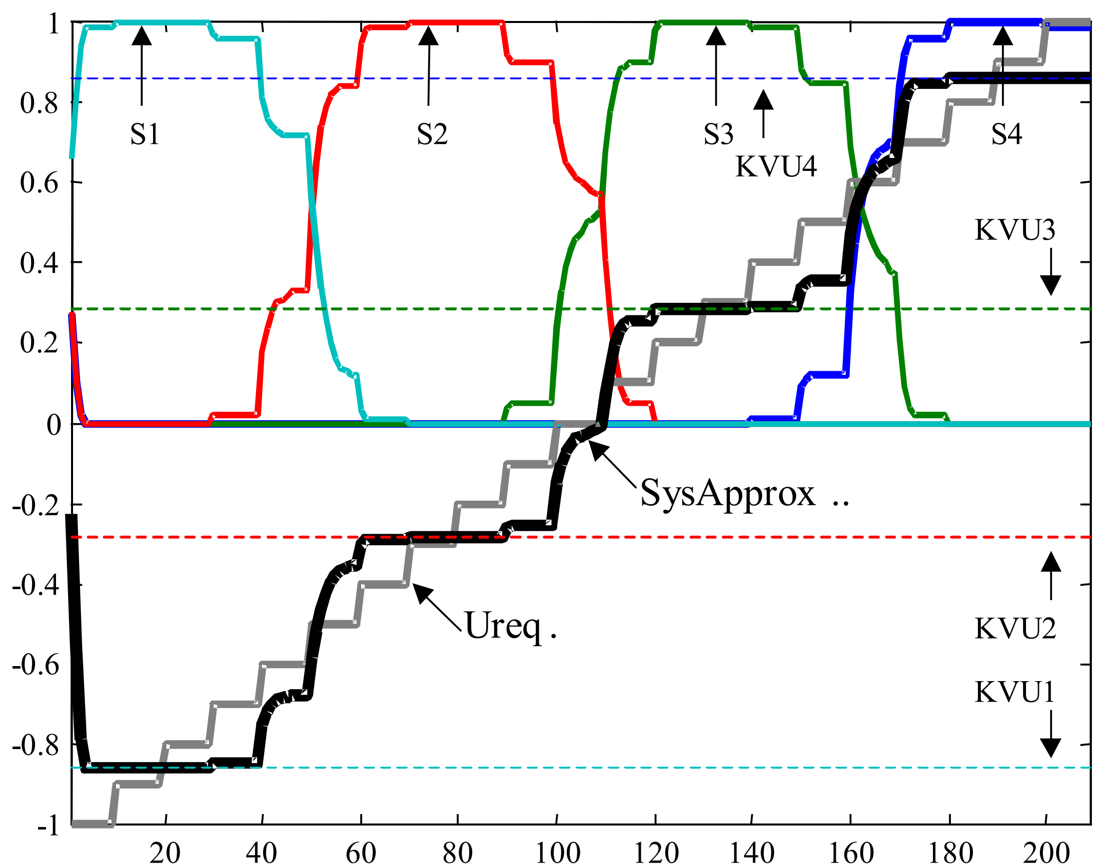
$S_{k-1}=1,\ SS_k=0,\ \exists k \neq i$



$S_{k-1}=1,\ SS_k=1,\ \exists k \neq i$

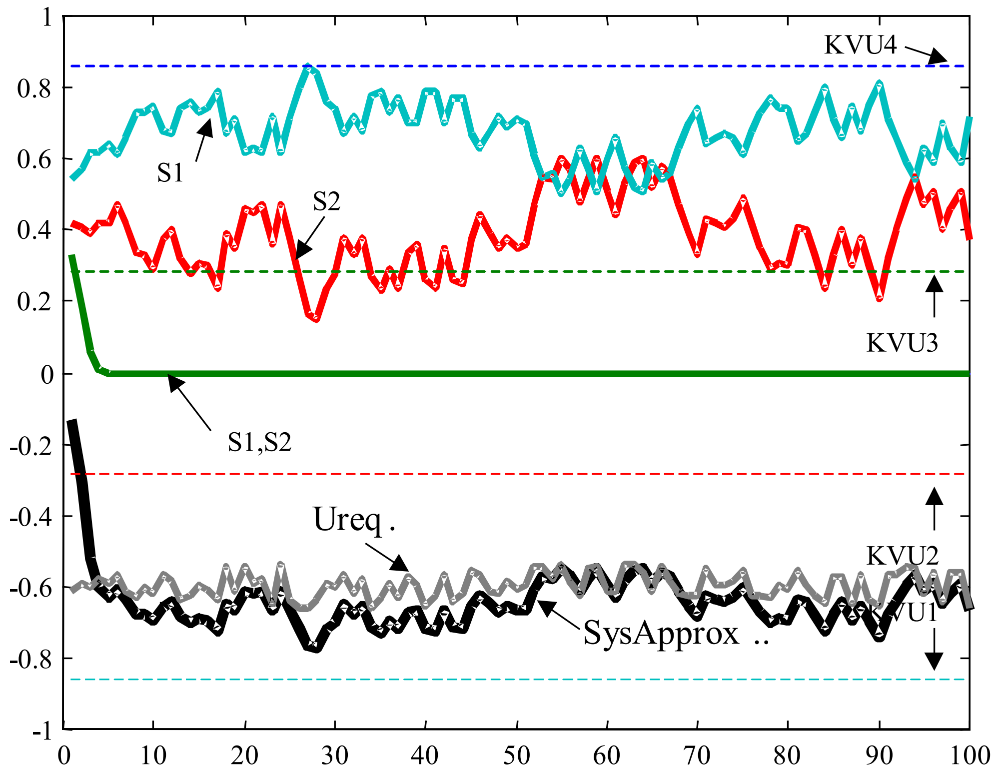
Experiences

- **Test user model set for checking the approximation ability**: By the first user model (KVU1), this furniture is very not ~, by the second (KVU2) a little bit not ~, by the third (KVU3) a little bit ~, by the fourth (KVU4) very ~. (These user models are containing only one Kansei descriptor and one furniture.)
- The **actual Kansei user model in case of step function user requirements**:

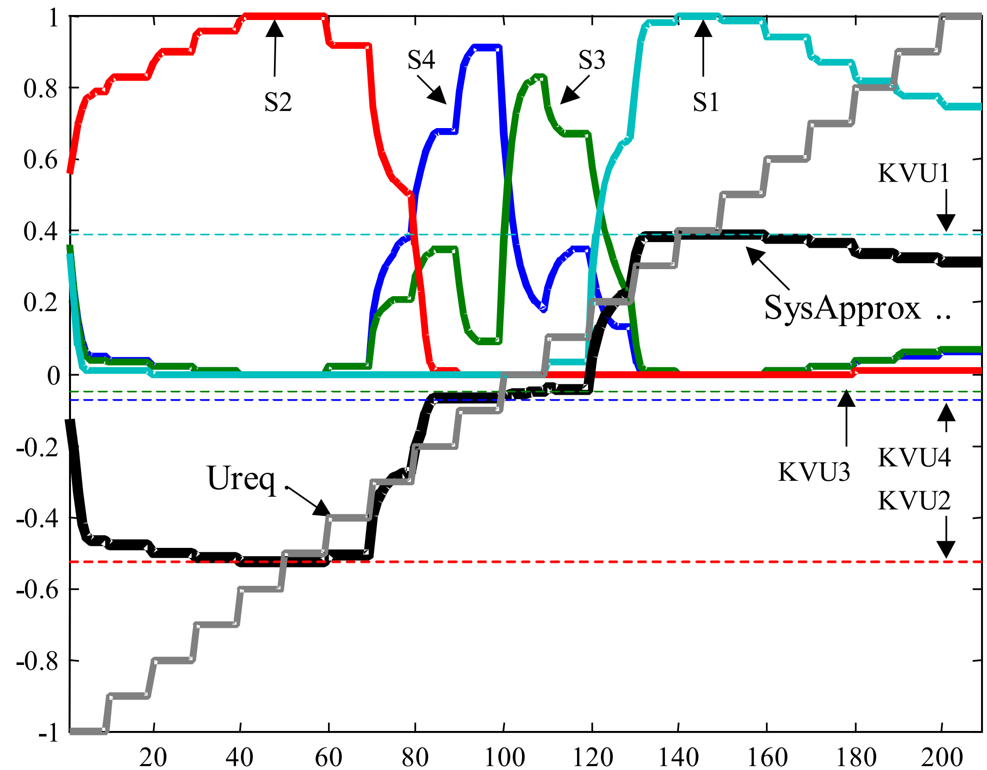


where Si is the i^th element of the state vector (actual level of similarity to the i^th user model), Ureq. is the user requirement, and SysApprox is the actual Kansei user model (only one Kansei descriptor).

Test user model set, constant noisy user requirements (robustness against noise):



Real Kansei user model set, step function actual user opinions:



where KVUi is the i^{th} user model, Si is the i^{th} element of the state vector (actual level of similarity to the i^{th} user model), Ureq. is the user requirement (only one Kansei descriptor), and SysApprox is the actual Kansei user model (only one Kansei descriptor).

Conclusions

- The simulated results shows, that in the tested situation the proposed similarity based reconfiguration method was **able unify the relevant, but only partially** (with respect to the state space of the system) **valid strategies**.
- 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**. This hierarchy can build 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. E.g. we can study and handle only the relevant characteristic situations of a system (situations need significantly different handling) and let the similarity based reconfiguration to handle all the other situations by interpolation.
- 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** – two or more strategies fitting the same situation have very different (competitive, not co-operative) conclusions.

Conclusions

- The main benefit of the proposed structure is to give a **simple way for adding user adaptivity to emotion-based selection systems**.
- Having different existing Kansei user models, it achieves user adaptivity simply by combining them (in interpolative manner) in the approximated best fitted way to the actual user: **“The more similar the actual user to one of the existing user models**, **the more similar the actual user model in use to that user model must be**.”
- The proposed structure **can handle many different user model parallel**, even if they are in **contradiction with each other**. These differences are **possible alternatives, not errors**.
- The “**adaptive knowledge**” of the system related to the actual user is not a new adapted user model, but **a set of approximated similarities**, the similarities of the actual user to the existing user models. (The model of the user from the viewpoint of the existing knowledge.)
- We hope, this kind of structure, the global similarity based combination of existing user models, is **able to avoid incoherence** could caused by step by step partial modifications of the user model. (Limited online user interaction, related to one or a few Kansei descriptors usually)
- 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**.
- Adopting **interpolative fuzzy reasoning for user model combination**, and **fuzzy automata for user similarity approximation** makes the proposed structure **very flexible, simple to build, and easily adjustable**.

Conclusions

- The goal of this paper beyond the introduction of a flexible behaviour-based control structure is to **introduce some generalised application areas**.

- The suggested structure, the **strategy reconfiguration**, is based on **fuzzy interpolative fusion** of **different existing strategies** in the **function of their actual necessity approximated by fuzzy automata**.

- This is a very easily built and simply adaptable structure for many application areas. Its **environment adaptivity** is easily adaptable to the current **situation adaptivity** (common behavioural-based applications), or **adaptivity to the controlled system** itself (e.g. fault-tolerant control applications), or even **adaptivity to the actual user** himself/herself (user-adaptive applications).

- The main benefits, both the **simplicity** and the **adaptivity** of the proposed structure are inherited from its **hierarchical construction**. This hierarchy can have 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). In this way **a rather complicated strategy can be modularly built**. Moreover, because of the similarity based interpolative manner of strategy combination, **there is some chance to get valid strategy in the area outside the area covered by the original strategies, too**.

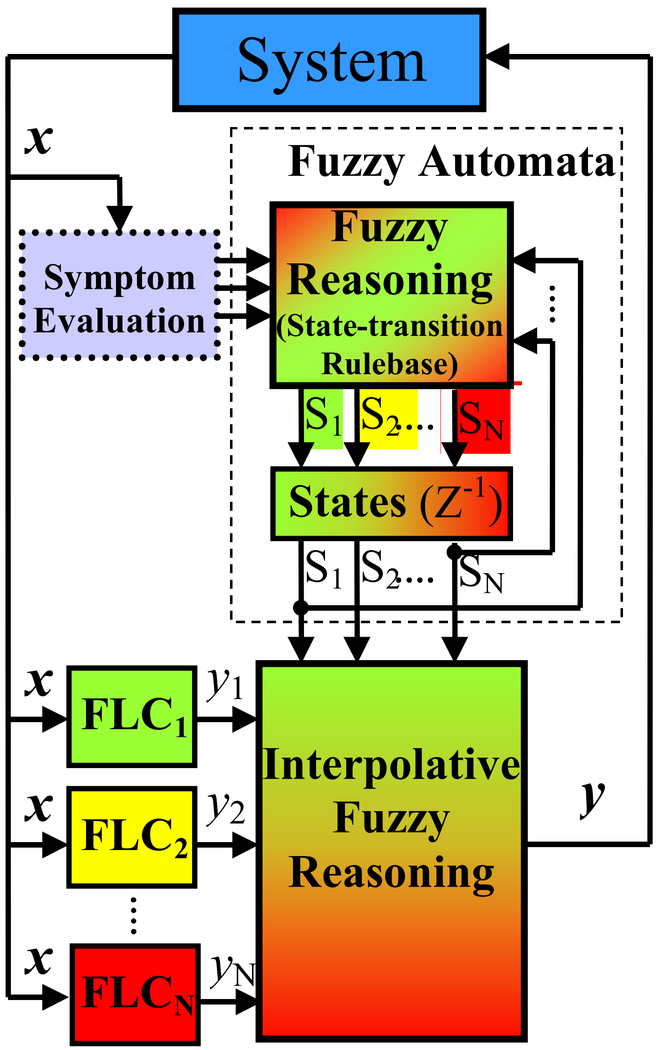
- The **benefit of adapting fuzzy automata** for system state (similarity) approximation in the proposed structure is to **give** (state) **memory to the system**. On one hand this memory is needed for the **correct symptom evaluation**, or it is able to **hold a kind of "history" information**. On the 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**. (E.g. 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 situations it can be built to **take the function of the symptom evaluation**, or to be able to handle the "history" of the system.

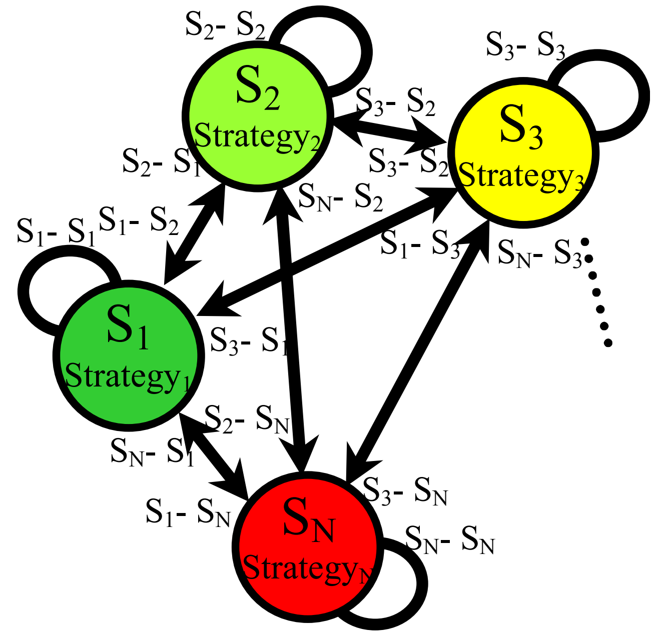
- The **main drawback** of the proposed structure is the **lack of alternative strategies handling ability**. From the viewpoint of interpolative conclusion fusion, the **existence of alternative strategies is a kind of contradiction** – two or more strategies fitting the same situation have very different (**competitive, not co-operative**) **conclusions**.

- A simple solution of this problem is **to design the fuzzy automata** to avoid situations where the suitability of alternative strategies is high at the same time – **avoiding situations of ambiguous selection among alternatives**. Or by **extending the state-space** by adding some additional "hidden" state-variables to the fuzzy automata **to track the alternatives** and **make the critical decisions** of selecting the suitable strategy from the alternatives **unambiguous**.

- A similar, but different problem is if one of **the known partial strategies contains contradiction** (alternative rules in its rulebase). This situation can be handled by **decomposing the original partial strategy** (which contains alternatives) **to a set of "contradiction free" strategies** and **handling them** the same manner, as they were **separate partial strategies**.

Similarity Based Strategy Reconfiguration





The basic structure of the state-transitions rulebase for the ith state S_i (R_{Ai}):

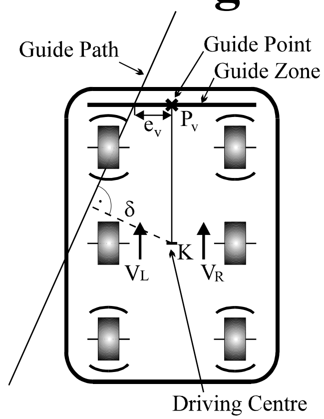
If	S_i=One	**And**	S_i-S_i=One	**Then**	S_i=One
If	S_i=One	**And**	S_i-S_k=One	**Then**	S_i=Zero
If	S_k=One	**And**	S_k-S_i=One	**Then**	S_i=One
If	S_k=One	**And**	S_k-S_i=Zero	**Then**	S_i=Zero

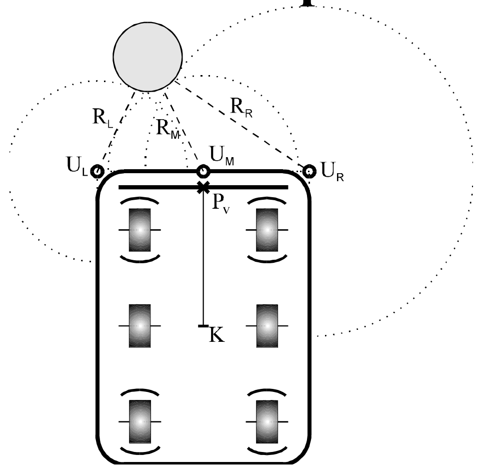
where S_i-S_j is the conclusion of the symptom evaluation about the state-transition from state i to j.

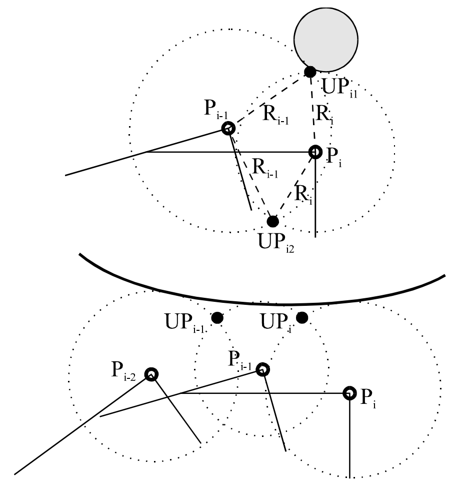
Strategy fusion rulebase:

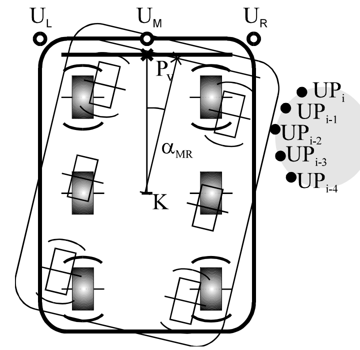
If	S_1=One	**And**	S_2=Zero	**And**	...	**And**	S_N=Zero	**Then**	$y=y_1$
If	S_1=Zero	**And**	S_2=One	**And**	...	**And**	S_N=Zero	**Then**	$y=y_2$
	...								
If	S_1=Zero	**And**	S_2=Zero	**And**	...	**And**	S_N=One	**Then**	$y=y_N$

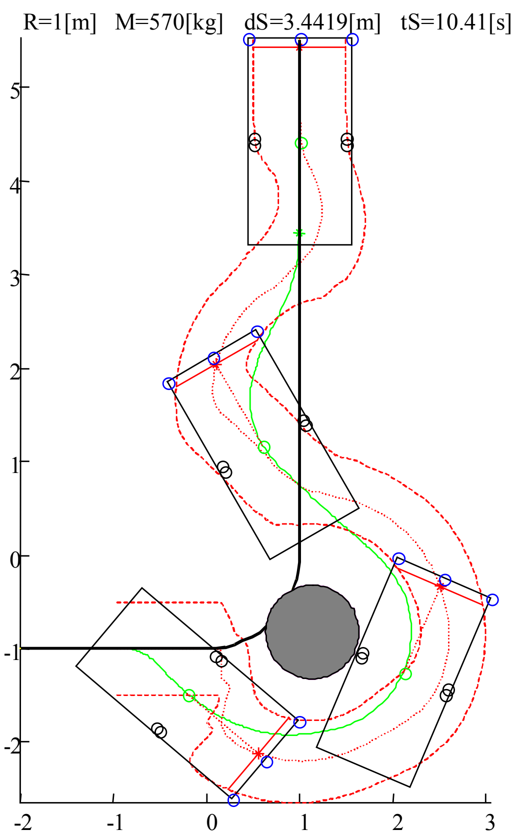
AGV navigation control example





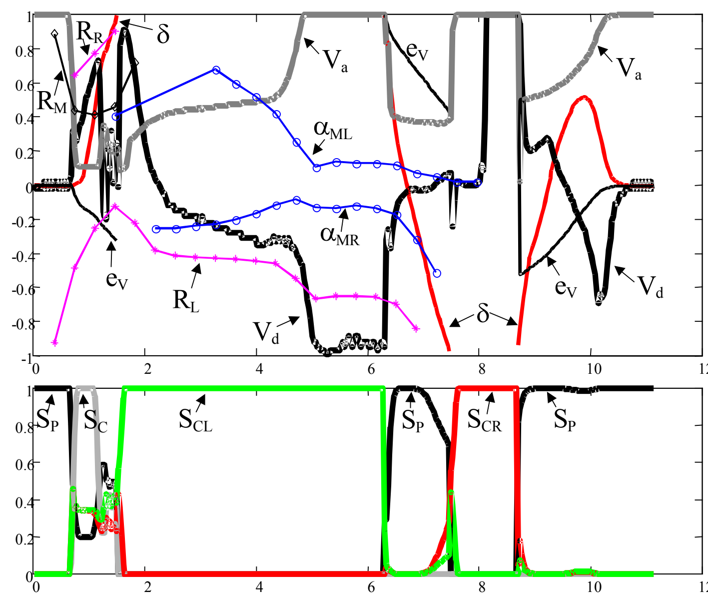




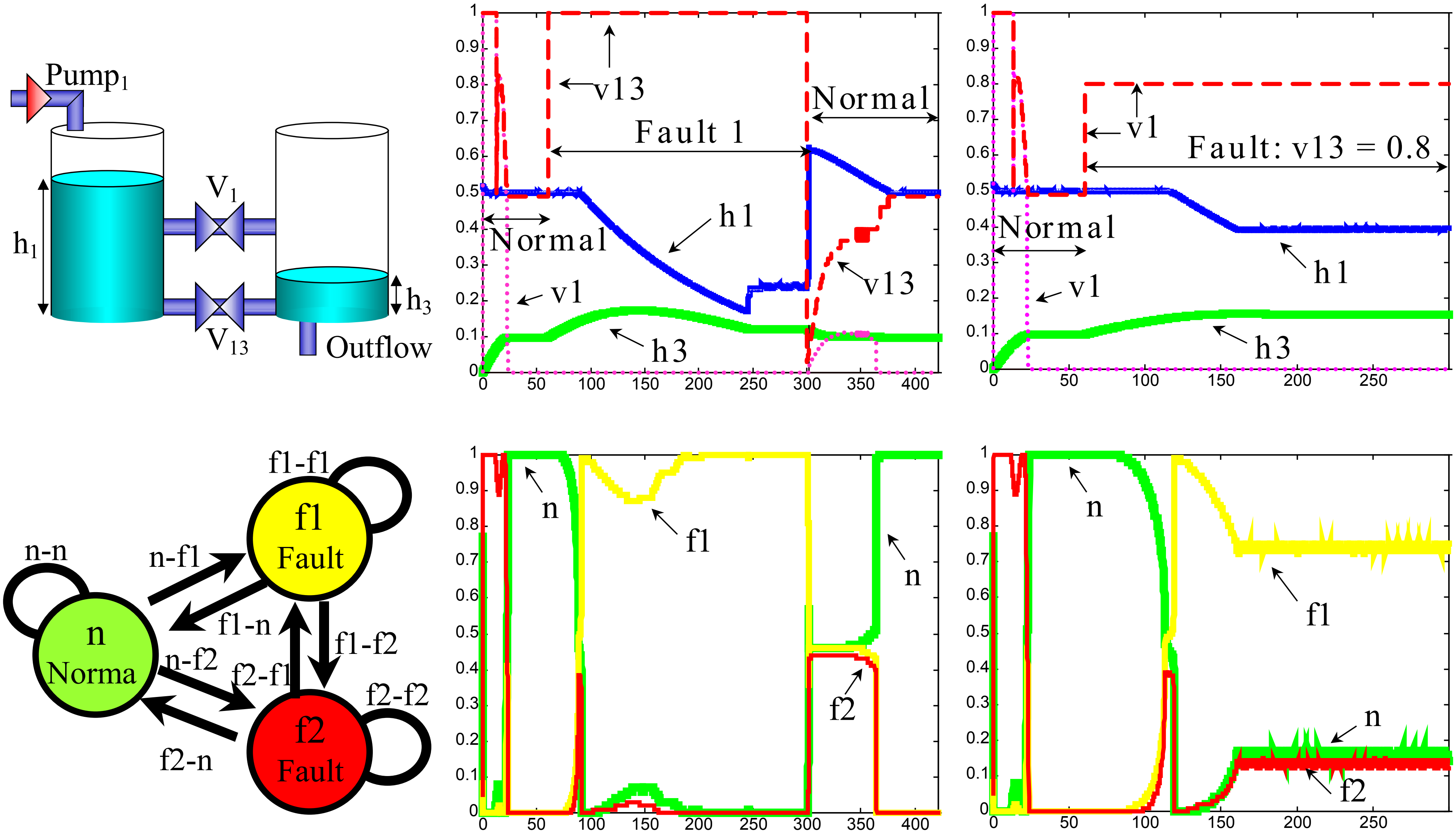


The known partial strategies:

- Path tracking and restricted collision avoidance strategy
- The collision avoidance strategy
- The collision avoidance with left/right tendency strategy



Fault tolerant control example



The structure was able to follow the studied operational modes (states) and state-transitions, even in some cases to approximate the unstudied situations, too.

User adaptive emotion-based system example