Sz. Kovács: *Similarity based Control Strategy Reconfiguration by Fuzzy Reasoning and Fuzzy Automata*, Proceedings of the IECON-2000, IEEE International Conference on Industrial Electronics, Control and Instrumentation, October 22-28, Nagoya, Japan, pp.542-547, (2000). – Draft version.

Similarity based Control Strategy Reconfiguration by Fuzzy Reasoning and Fuzzy Automata

Szilveszter Kovács

Gifu Prefectural Research Institute of Manufacturing Information Technology
4-179-1 Sue Kakamigahara Gifu 509-0108 Japan
E-mail: szkszilv@gold.uni-miskolc.hu

Abstract

In case of control partially known complex systems, a kind of control strategy could be built simply by combining some known, partially valid control strategies. These known strategies are covering only a part of the whole state space of the possible control strategies. In some situation, these strategies could be combined in a hierarchical fuzzy way to form a strategy covering at least the same state space as all the attendant strategies.

There are two main questions of this kind of strategy reconfiguration. The first is the decision about the level of necessity and the type of the strategy (rulebase in the case of fuzzy control) needed to use by the actual system. The second is the way of generating the actual strategy, as a combination of the different existing known partially valid strategies. One solution for these questions could be the combination of the fuzzy automata based system state approximation and the hierarchical interpolative fuzzy reasoning - the similarity based strategy reconfiguration.

As an example for the practical application of the proposed structure, path tracking a collision avoidance strategy of a simulated AGV is introduced in this paper.

Keywords: hierarchical interpolative fuzzy reasoning, fuzzy automata, strategy reconfiguration

1 Introduction

The main idea of the suggested similarity based strategy reconfiguration – "More similar the actual system behaviour to one of the known partial strategy prerequisites, more similar must be the strategy used to that strategy" – 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, we can simply combine them in interpolative manner to get one strategy, which has an extend area of validity (at least in a part of the area covered by the original strategies). Moreover, because of the similarity based interpolative manner of strategy combination, we have 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 main tasks of the proposed similarity based strategy reconfiguration are twofold. The first is *the actual system state approximation*. We have to make a decision about the level of similarities 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). The second is the *combination of the existing partial strategies* based on these similarities. For the first task we suggest to adapt a *fuzzy automata*, where the state variables are the corresponding similarities (approximated similarities of the actual system behaviour to the prerequisites of all the known strategies) and the state transition are driven by fuzzy reasoning (see fig.2.). For the second task, we suggest to use *interpolative fuzzy reasoning*. Having the approximated similarities of the actual system behaviour to the prerequisites of all the known strategies, the conclusions of the different strategies could be simply combined as an upper level interpolative reasoning in a function of the corresponding similarities (see fig.2.).

2 The system state approximation

The first step of the system state approximation is the *symptom evaluation*. The task of symptom classification is a series of similarity checking between an actual symptom and the known symptoms (in our 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, evaluating the actual symptom in our case is nothing else than calculating the similarity values of the actual symptom to all the known symptoms patterns (the prerequisites of the known partially valid strategies). There are many methods exist for fuzzy logic symptom evaluation. E.g. we can adopt fuzzy classification methods e.g. the Fuzzy c-Means fuzzy clustering algorithm [1], where the known symptoms patterns are the cluster centres and the similarities of actual symptom to them can be fetched from the fuzzy partition matrix. On the other hand, having a simple situation 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, we need the approximated system-state itself. A very simple way of solving this difficulty is the adaptation of *fuzzy automata*. In this case, the state vector of the automata could be the vector of approximated similarities of the actual system behaviour to the prerequisites of the known strategies. Moreover, the state-transitions 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, we can make a decision about the new approximated system-state. 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), the state-transition decision can be done by fuzzy reasoning based on the actual state and the conclusions of the symptom evaluation. The automata is a fuzzy automata, because of the state variables are fuzzy membership values (similarities – infinite states) 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 ith state S_i (R_{Ai}) are the following:

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. One and Zero are the labels of fuzzy sets (linguistic terms) representing high and low similarity. The interpretation of these fuzzy sets can be different in each S_i, S_i-S_k universes. The structures of the state-transition rules are similar for all the strategies (see e.g. fig.1.). Because of the possibly incomplete structure [3] of state-transition rulebase, we applied interpolative fuzzy reasoning for the state transition decisions.

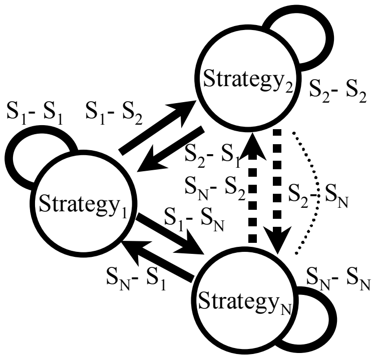

Fig.1. Basic structure of the state-transition rules.

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

3 Strategy combination

The conclusion of the system state approximation (the approximated state itself) is a set of similarity values, the level of similarities of the actual system behaviour 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 their corresponding similarities (S_i), as an upper level interpolative fuzzy reasoning [4] (see Fig.2.). The simplest way for such a combination is the application of the interpolative fuzzy reasoning [4]. The main idea of the proposed similarity based strategy reconfiguration – "More similar the actual system behaviour 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:

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** $\boldsymbol{y}=y_N$

(See fig.2. for notation.)

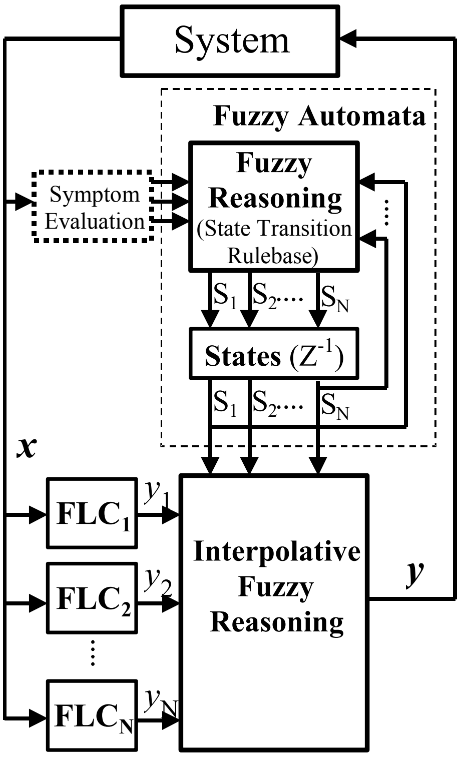

Fig.2. Structure of the proposed Similarity based control strategy reconfiguration

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).

4 Application example

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) [5,6] is introduced. In our example application the steering control has two main goal, the path tracking (to follow a guide path) and the collision avoidance. Our 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.

The simulated path sensing system senses the position of the guide path by special sensors (*guide zone*) tuned for the guide path. The *guide zone* is a section of the AGV determined by the guide path sensor (or raw of sensors). 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.).

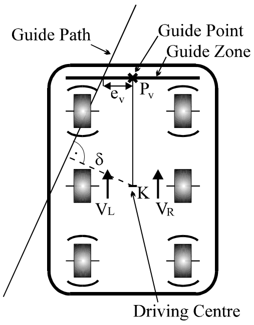

Fig.3. Differential steered AGV, δ is the path tracking error, ev is the distance of the guide path and the guide point, Pv is the guide point, K is the driving centre

For defining the collision avoidance strategies, we have to study the types of the possible collision situations. There are two different collision situations, the frontal and the side collision. We need the simplest obstacle sensor configuration giving enough information for both the avoidable situations. Having the preconditions of motionless and avoidable 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.4.) [6]. The three distances (R_L, R_R, R_M), measured by the three obstacle sensors (R_L, R_R, R_M) gives sufficient information for finding a strategy to be able to avoid the frontal collision situations. The sufficiency of the measurements of these sensors for generating observations for avoiding the side collisions is not so simple. Having the preconditions of motionless and avoidable obstacles, we have a chance to 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. We call these points *unsafe*, or *risky points*. The distance measured by an obstacle sensor means the existence of a potential obstacle outside the circle defined by the position of the sensor and the measured value (see e.g. on fig.5.). Having more measurements and more positions we can approximate the boundaries of the obstacles by the pair by pair point of intersection of these circles (see e.g. on fig.5.). 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 unsafe points, we calculate the actual maximal left and right turning angle without side collision (α_{ML}, α_{MR}) (see e.g. on fig.4.).

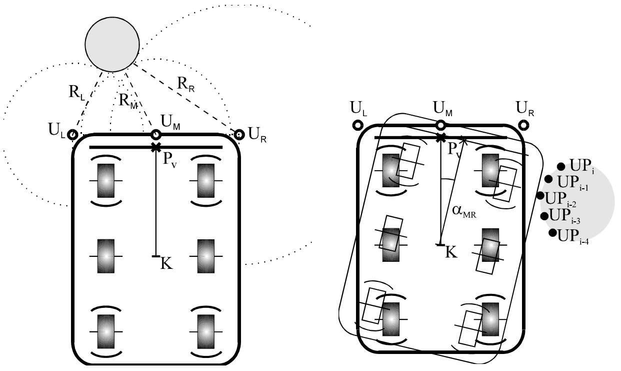

Fig.4. RL, RR, RM are the distances measured by the left, right and middle distance sensors (UL, UR, UM), α_{MR} is the maximal right turning angle without collision.

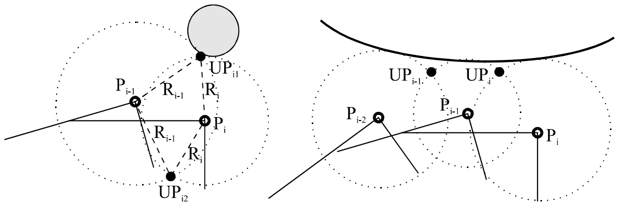

Fig.5. The obstacles boundaries approximated by discrete unsafe points, R is the distance measured by the sensor P, and UP is the unsafe (risky) point.

4.1 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 the fuzzy rule base. In our case - using interpolative fuzzy reasoning for direct fuzzy control - constructing the fuzzy rule base is very simple. We do not have to bother with building 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, we can check the performance of the controller after each step. In our simulated example, All the rulebases introduced, and the corresponding fuzzy partitions (not introduced) in the followings were generated in such a manner. First having heuristic rulebase and fuzzy partition structure, after some trial and error style modification, a "working" strategy was got (the first strategy was fulfilling the task of the strategy). Then the working strategy was tuned in its own environment. For tuning the working strategy, a simple genetic method was adapted. It was modified the fuzzy partitions only (see more detailed in [6]), to get an at least locally better solution than the original one. In this example, we have four different partial strategies:

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 [5,6]. (In our case, restricted collision avoidance means "avoiding obstacles without risking the chance of loosing the guide path".) The basic idea 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 we need two conclusions, the *speed* (V_a) and the *steering* (V_d). S we have two rulebases, one for the steering R_{Vd} and one for the speed R_{va} of the AGV.
The i^th rules 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, we got only 12 rules 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

Our 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, we have got the following rules for controlling the steering ($\mathbf{R_{Vd}}$) and 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 he 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 obstacles). 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*

4.2 The fuzzy automata

Our example application is so simple, that it does not need separate symptom evaluation. The function of symptom evaluation is built in 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, we need four state-transition rulebases. We have the $\mathbf{R_{SP}}$ state transition rulebase for determining the next value of the S_P state-variable, $\mathbf{R_{SC}}$ for 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, 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.

5 Conclusions

The goal of this paper was the introduction of a flexible behaviour-based control structure through its application example. (A good overview behaviour-based control structure can be found in [2].) 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. Fig.7-8. are introducing some results of the simulated application. The results shows, that in the tested situation the proposed structure was able unify the relevant, but only partially valid (with respect to the state space of the system) strategies. Both the simplicity and the adaptivity of the proposed structure are inherited from its hierarchical construction. The benefit of applying fuzzy automata for system-state approximation is to give (state) memory to the system. On one hand, this memory is needed for the correct symptom evaluation; on the other hand, it is able to hold a kind of "history" information (e.g. left, or right turning tendency strategy decision of the 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. A simple solution of this problem is to design the fuzzy automata to avoid situations of ambiguous selection among alternatives, by extending the state-space. For example 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. Similar, but different problem if one of the partial strategies contains contradiction (e.g. if there are contradicting, or 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.

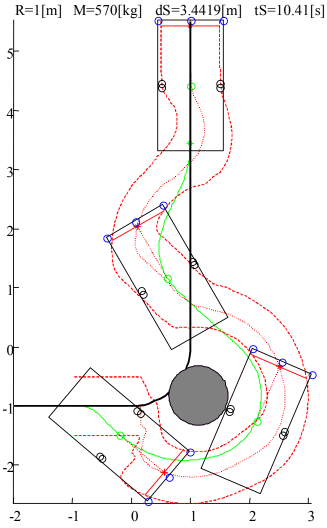

Fig.7. AGV path tracking and collision avoidance strategy implemented using similarity based control strategy reconfiguration (simulated results, one obstacle)

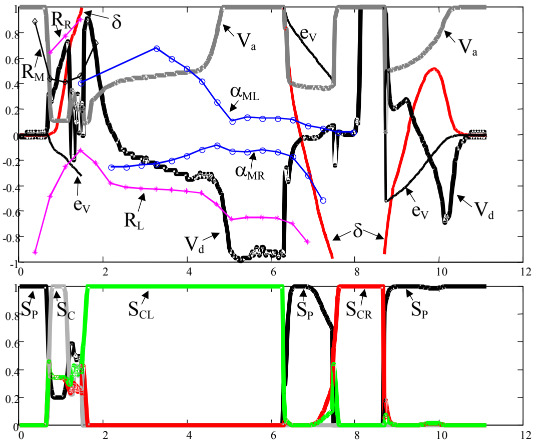

Fig.8. Time function of observations, conclusions (see 4.1.) and system state values (S_P,S_C,S_{CL},S_{CR}) for the simulation on fig.7.

Acknowledgement

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] J.C. Bezdek, *Pattern Recognition with Fuzzy Objective Function*, Plenum Press, New York, 1981.

[2] P. Pirjanian, *Behavior Coordination Mechanisms - State-of-the-art*, Tech-rep. IRIS-99-375, Institute for Robotics and Intelligent Systems, School of Engineering, Univ. of Southern California, 1999.

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

[4] Sz. Kovács, 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.

[5] 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, Springer, pp.456-467, 1997.

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