A Flexible Fuzzy Behaviour-based Control Structure

Szilveszter Kovács

Department of Information Technology, University of Miskolc,
Miskolc-Egyetemváros, Miskolc, H-3515 Hungary,
szkovacs@iit.uni-miskolc.hu

Abstract: The main idea of behaviour-based control structures is to handle partially known complex situations by a set of known behaviours. By discrete switching to the behaviour seems to be the most appropriate one, or by fusing the behaviours appeared to be the most appropriate ones. These structures has two main tasks to solve. The decision about the level of suitability of the known behaviours in handling the actual situation, and the way of their fusing to form the actual behaviour. In this paper, for these tasks, a flexible structure, the fuzzy automaton based system state approximation and the interpolative fuzzy reasoning based fusion is suggested. For demonstrating the applicability of the suggested structure, a path tracking and collision avoidance navigation control of a simulated automated guided vehicle (AGV) is also introduced briefly in this paper.

Keywords: behaviour-based control, interpolative fuzzy reasoning, fuzzy automaton.

1 Introduction

In behaviour-based control systems (a good overview can be found in [3]), the actual behaviour of the system is formed as one of the existing 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 tasks. The first is a decision, which behaviour is needed in an actual situation, and the levels of their necessities in case of behaviour fusion. The second is the way of the behaviour fusion. The first task can be viewed as an actual system state approximation, where the actual system state is the set of the necessities of the known behaviours needed for handling the actual situation. The second is the fusion of the known behaviours based on these necessities.

The applicability of the behaviour-based control structures is based on the premise, that all the situations could possibly occur can be handled by a behaviour formed as a convex combination of the known (existing) behaviours. This case having the relevant behaviours (control strategies in case of control application) the behaviour-based control structure has the chance to form a suitable actual behaviour (control strategy).

In the followings, first a simple and flexible fuzzy behaviour-based control platform based on the fuzzy automaton and the hierarchical interpolative fuzzy reasoning, then as one of its possible application areas, a vehicle navigation control will be briefly introduced.

2 The Suggested Fuzzy Behaviour-based Structure

The first task of the behaviour-based control is to determine the necessities of the known behaviours needed for handling the actual situation. In the suggested behaviour-based control structure, for this task the finite state fuzzy automaton is adapted (Fig.1.) [4]. This solution is based on the heuristic, that the necessities of the known behaviours for handling a given situation can be approximated by their suitability. And the suitability of a given behaviour in an actual situation can be approximated by the similarity of the situation and the prerequisites of the behaviour. (Where the prerequisites of the behaviour is the description of the situations where the behaviour is valid (suitable itself)). This case instead of determining the necessities of the known behaviours, the similarities of the actual situation to the prerequisites of all the known behaviours can be approximated.

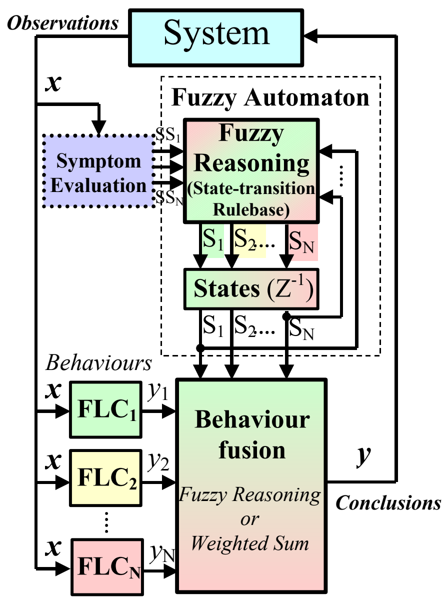

Fig. 1. The applied behaviour-based control structure

Thus the first step of the system state approximation is determining the similarities of the actual situation to the prerequisites of all the known behaviours – applying the terminology of fault classification, it is the symptom evaluation (see e.g. Fig.1.). The task of symptom evaluation is basically a series of similarity checking between an actual symptom (observations of the actual situation) and a series of known symptoms (the prerequisites – symptom patterns – of the known behaviours). These symptom patterns are characterising the systems states where the corresponding behaviours are valid. Based on these patterns, the evaluation of the actual symptom is done by calculating the similarity values of the actual symptom (representing the actual situation) to all the known symptoms patterns (the prerequisites of the known behaviours). There are many methods exist for fuzzy logic symptom evaluation. For example fuzzy classification methods e.g. the Fuzzy c-Means fuzzy clustering algorithm [1] can be adopted, 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 behaviours are strongly dependent on the actual behaviour of the system. Each behaviour 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 automaton. This case the state vector of the automaton is the approximated system state, and the state-transitions are driven by fuzzy reasoning (Fuzzy state-transition rulebase on Fig.1.), as a decision based on the previous actual state (the previous iteration step of the approximation) and the results of the symptom evaluation. The basic structure of the rulebase applied for the state-transitions of the fuzzy automaton (rules for interpolative fuzzy reasoning) for the ith state S_i (R_{Ai}) can be the following:

$$
\begin{array}{llll}
\textbf{If } S_i=\text{One} & \textbf{And } S_i\text{-}S_i=\text{One} & \textbf{Then } S_i=\text{One} \\
\textbf{If } S_i=\text{One} & \textbf{And } S_i\text{-}S_k=\text{One} & \textbf{Then } S_i=\text{Zero} \\
\textbf{If } S_k=\text{One} & \textbf{And } S_k\text{-}S_i=\text{One} & \textbf{Then } S_i=\text{One} \\
\textbf{If } S_k=\text{One} & \textbf{And } S_k\text{-}S_i=\text{Zero} & \textbf{Then } S_i=\text{Zero}
\end{array}
\tag{1}
$$

where $S_i\text{-}S_k$ is the conclusion of the symptom evaluation about the state-transition from state i to k, $\forall k \in [1, N]$, N is the number of known behaviours (state variables). The structure of the state-transition rules is similar for all the state variables. `Zero` and `One` are linguistic labels of fuzzy sets (linguistic terms) representing high and low similarity. The interpretations of the `Zero` and `One` fuzzy sets can be different in each S_i, S_i-S_k universes. The reason for the interpolative manner of fuzzy reasoning is the incompleteness of state-transition rulebase [2].

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 automaton (as it was done in the example application of this paper).

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 and the prerequisites of the known behaviours. Applying these similarities as the level of necessities for fusing the known behaviours, the actual behaviour can be formed.

In case of fuzzy behaviour fusion, the following rulebase can be used for the fusion of the conclusions of the different behaviours: (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** $\boldsymbol{y}=y_N$

where S_i is the i[th] state variable, y_i is the conclusion of the i[th] behaviour and $\boldsymbol{y}$ is the fused conclusion. `Zero` and `One` are linguistic labels of fuzzy sets (linguistic terms) representing high and low similarity. The interpretations of these fuzzy sets can be different in each S_i universes.

Instead of 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). E.g.:

$$\boldsymbol{y} = \frac{\sum_{i=1}^{N} w_i \cdot y_i}{\sum_{i=1}^{N} w_i}, \tag{3}$$

where $w_i = S_i$ is the weight and y_i is the conclusion of the i^{th} behaviour, $\boldsymbol{y}$ is the fused conclusion.

3 A vehicle navigation control example

For introducing some of the possible application areas of the proposed fuzzy behaviour-based control structure, a simulated steering control of an automated guided vehicle (AGV) [6], [7], [8] is briefly introduced. In the example application the steering control has two main goals, 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.2.).

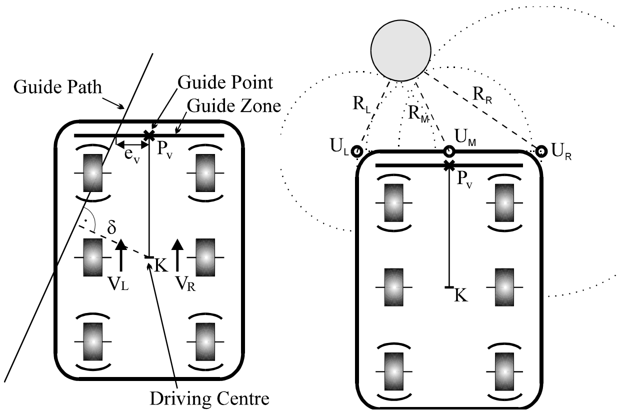

Fig. 2. Differential steered AGV with guide zone, δ is the path tracking error, ev is the distance of the guide path and the guide point, Pv is the guide point, K is the driving centre, RL, RR, RM are the distances measured by the left, right and middle ultrasonic sensors (UL, UR, UM).

In the collision avoidance strategies, two different collision situations, the frontal and the side collision are distinguished. 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.2.)) to approximate both the collision conditions [7]. Having the preconditions of motionless obstacles, the obstacle distance measurements of the near past can be used 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), the boundaries of the obstacles can be approximated by discrete points [7]. These points are called *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.3.). Having more measurements and more positions, the boundaries of the obstacles can be traced by the pair by pair point of intersections of these circles (called “unsafe” points [7], see e.g. on Fig.3.). 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 numerous unsafe points, the actual maximal left and right turning angle without side collision (α_{ML}, α_{MR}) is calculated [7].

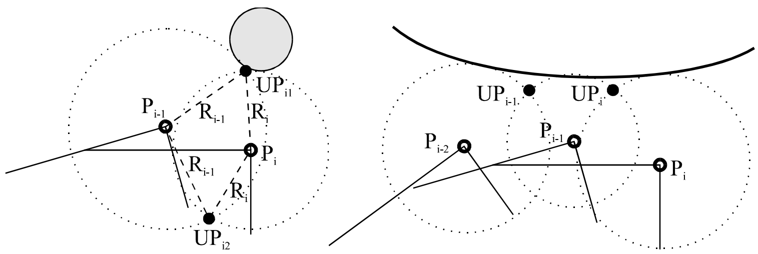

Fig. 3. The obstacles boundaries approximated by discrete unsafe points, R is the distance measured by the sensor P, and UP is the unsafe (risky) point.

The first stage of building the suggested behaviour-based control structure is to build the component behaviours. The simplest way of defining these strategies is based on describing the operator's control actions. These control actions could form a fuzzy rulebase. In the example – using interpolative fuzzy reasoning for direct fuzzy control – constructing the fuzzy rulebase is very simple. It is not necessary to build a complete fuzzy rulebase; 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 following simulated example, all the rulebases introduced, and the corresponding fuzzy partitions (not introduced) were generated in such a manner. Starting from a heuristic rulebase and fuzzy partition structure, after some trial and error style modification, a "working" strategy was achieved (the first strategy fulfilling the task of the studied strategy). Then the working strategy was tuned in its own environment. For tuning the working strategy, a simple genetic method was adapted, for modifying the fuzzy partitions only (see more detailed in [7]), to get an at least locally better solution than the original one.

In the example, there are four different known behaviours [8]:

Path tracking and restricted collision avoidance strategy: The main goal of this strategy is the path tracking (to follow a guide path) and as a sub goal, a kind of restricted (limited) collision avoidance [7]. (Here the restricted collision avoidance means, "avoiding obstacles without risking the chance of loosing the guide path".) The basic 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, then 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 (e_v), and the distance between the guide path and the guide point (δ). And 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 are needed: the speed (V_a) of the vehicle and the level of steering (V_d). In implementation it takes two rulebases, one for the steering R_{Vd} and one for the speed R_{va}.

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, only 12 rules are sufficient for controlling the steering ($\mathbf{R_{Vd}}$) and 5 is needed 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 the meanings of the linguistic labels are: *N*: negative, *P*: positive, *L*: large, *M*: middle, *Z*: zero.

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

$\mathbf{R_{Vd}}$:	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 the meanings of the linguistic labels are: *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 behaviours 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 ($\mathbf{R_{Vd}}$) is the following:

$\mathbf{R_{Vd}}$:	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*

The example application is so simple, that it does not need separate symptom evaluation. The function of the symptom evaluation is built to the state-transition rulebase of the fuzzy automaton. Having four known behaviours, the automaton 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 are 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. The state-transition rulebases for interpolative fuzzy reasoning are the followings [8]:

$\mathbf{R_{SP}}$:

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 the meanings of the linguistic labels are: *N*: negative, *P*: positive, *VL*: very large, *L*: large, *S*: small, *Z*: zero, *V*: path valid, *NV*: path not valid.

The conclusions of the four known behaviours are fused by the rulebase introduced in (2). Having the behaviours alike the behaviour fusion implemented on interpolative fuzzy reasoning, the behaviours together with the behaviour fusion forms a hierarchical interpolative fuzzy reasoning structure.

Conclusions

The goal of this paper was to introduce a simple and flexible fuzzy behaviour-based control structure through its vehicle navigation control application example.

The suggested structure is based on fuzzy interpolative fusion of different known behaviours in the function of their actual necessities approximated by fuzzy automaton. It is an easily built and simply adaptable structure for many application areas (see e.g. [9] as an application area in user adaptive emotional and information retrieval systems).

Fig.4-7. are introducing some results of the simulated vehicle navigation application. The results show, that in the tested situation the suggested structure was able to fuse the known behaviours in the expected manner.

The main benefits, both the simplicity and the situation adaptivity of the behaviour-based control structures are inherited from their hierarchical construction. 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 suggested fuzzy behaviour-based control structure is simply fusing these strategies to form one strategy, which has an extended area of validity. This way a rather complicated strategy can be modularly built.

The benefit of adapting fuzzy automaton for system state 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; 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).

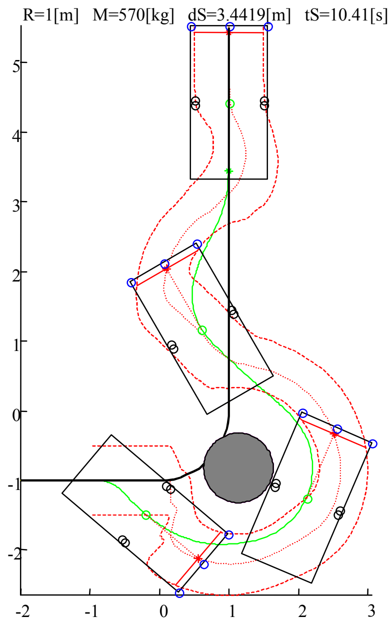

Fig. 4. Track of a single run in case of curved guide path and one obstacle.

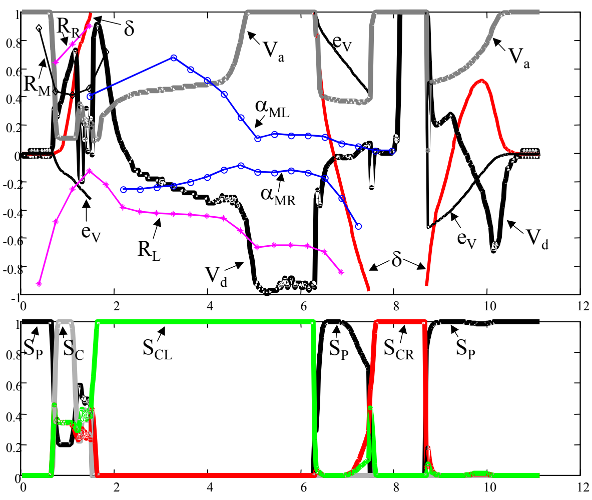

Fig. 5. Time function of observations, conclusions and system state values (S_P, S_C, S_{CL}, S_{CR}) related to the track of Fig.4. (See strategy descriptions above for notation.)

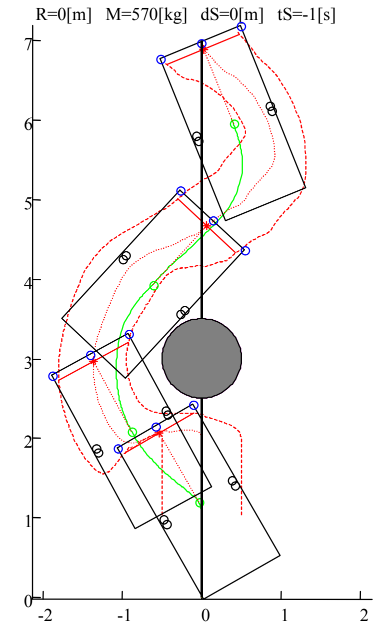

Fig. 6. Track of a single run in case of straight guide path and one obstacle in the path centre.

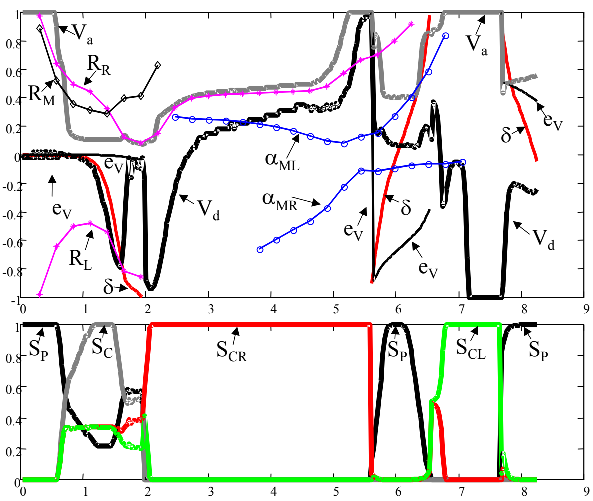

Fig. 7. Time function of observations, conclusions and system state values (S_P, S_C, S_{CL}, S_{CR}) related to the track of Fig.6. (See strategy descriptions above for notation.)

Acknowledgement

This research was partly supported by the Japan Gifu Prefectural Research Institute of Manufacturing Information Technology and the Intelligent Integrated Systems Japanese Hungarian Laboratory. Szilveszter Kovács is supported by the György Békésy Postdoctoral Scholarship.

References

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

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

[3] 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, http://www-robotics.usc.edu/~paolo/publications/ bcm.ps.gz, October, (1999).

[4] Sz. Kovács, “Interpolative Fuzzy Reasoning and Fuzzy Automaton in Adaptive System Applications”, *Proceedings of the IIZUKA2000*, 6th International Conference on Soft Computing, October 1-4, Iizuka, Fukuoka, Japan, (2000), pp.777-784.

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

[6] 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, Germany, (1997), pp.456-467.

[7] 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, (1998), pp.67-72.

[8] Sz. Kovács: “Similarity based Control Strategy Reconfiguration by Fuzzy Reasoning and Fuzzy Automaton”, *Proceedings of the IECON-2000, IEEE International Conference on Industrial Electronics, Control and Instrumentation*, October 22-28, Nagoya, Japan, (2000), pp.542-547.

[9] Sz. Kovács: “Fuzzy Reasoning and Fuzzy Automata in User Adaptive Emotional and Information Retrieval Systems”, *Proceedings of the 2002 IEEE International Conference on Systems, Man and Cybernetics*, October 6-9, Hammamet, Tunisia, 02CH37349C, ISBN: 0-7803-7438-X, WP1N5, p.6, (2002).