# Fuzzy Behaviors and Behavior Arbitration in Autonomous Vehicles

António Abreu[1,2] and Luís Correia[2]

[1] Escola Superior de Tecnologia, Instituto Politécnico de Setúbal
Campus do IPS, Setúbal 2914-508, Portugal
aabreu@est.ips.pt
[2] CENTRIA - Dept. Informática, Fac. Ciências e Tecnologia,
Universidade Nova de Lisboa
Quinta da Torre, 2825-114 Caparica, Portugal
Luis.Correia@di.fct.unl.pt, http://ssdi.di.fct.unl.pt/~lc

**Abstract.** In this paper a fuzzy behavior based decision control is presented. A behavior is described by one or more fuzzy controllers reacting to external stimulus, and a state variable (denoted behavior activity) measuring how evident are the conditions of application of the knowledge coded in the fuzzy controllers. The autonomous vehicle decision control is composed by several behaviors of this kind, each one feeding an arbitrator with the activity and action values. The arbitrator evaluates the final action to send to the actuators taking into consideration each behavior activity and action values. Two arbitration methods are proposed. Results of simulations are presented and conclusions are drawn.

**Keywords**: autonomous vehicles, Khepera, fuzzy controller, fuzzy behavior, fuzzy behavior arbitration.

## 1 Introduction

The domain of application of fuzzy control is now large. It can be applied in situations where smoothness of actuation is important (e.g. Anti-Block Breaking Systems in automobiles), where it is difficult to express knowledge in an analytical way (high system complexity), where the variables domain are qualitative in nature (e.g. water purification), etc. In all these situations the gains are: robustness (in the sense that the controller is applicable to situations not explicitly declared), project efficiency (the rule base determination is less complicated than the determination of some analytical model, e.g., high order differential equations), wide application domain (the generality of application of fuzzy control is now accepted) and "handiness" of thinking (the rule set partly implements a divide-and-conquer problem resolution approach), among others.

All these issues are related to the control of autonomous vehicles at several levels of design. The simpler one is piloting in environments with obstacles. In fact, the rate of wheels curvature depends in a qualitatively way on the rate and the angle of approach of the vehicle to some obstacle. As another example,

Progress in Artificial Intelligence, 9th Portuguese Conference on Artificial Intelligence, EPIA '99, Évora, Portugal, September 21-24, 1999, in Pedro Barahona, José Júlio Alferes (Eds.), Lecture Notes in Computer Science 1695 Springer 1999, ISBN 3-540-66548-X

the linear and differential velocity of some transportation vehicle depends on the weight and shape of the load and how the objects are accommodated in the vehicle. For some external observer and for both illustrative situations, it seems that this is the humans *modus faciendi*. Informally, this last statement shows the adequacy of fuzzy logic to guiding/driving control.

The control of an autonomous vehicle cannot be reduced to the guiding task ([3]). There are other modules that must be addressed, which can be quite complex ([17], [7]). Let's assume that the objectives of some vehicle are: to have a wander behavior, and to have a self-looking-for-energy behavior. Consider also that the energy source is somewhere near the walls. It can be associated a wall following behavior to the self-looking-for-energy behavior. The projected global behavior is as follows: the vehicle wanders when the energy level is "high" enough, and starts looking for energy supply when needed, i.e. look for and follow walls until it finds energy. The question is: is it possible to have a single fuzzy controller taking care of everything? Set aside the operators (such as aggregation and composition operators, defuzzification, etc.), the main component of a fuzzy controller is the knowledge base, which is composed of rules. In a typical situation, all rules depend on the same input variables and influence the same output variables; they are one of a kind. The input variables domain imposes different contexts, and as so, some rules will have more importance on the outputs than others. Even if one uses both positive (conditions to satisfy) and negative knowledge (conditions that must not be satisfied), in practice such a centralized controller will have such a dimension that it will be difficult to design, either in terms of knowledge elicitation or performance tuning.

In fact, in the traditional approach to fuzzy control ([10],[11]), the controller is made up of a knowledge base which is usually a monolithic computational entity; it works like a mathematical function where one feeds the inputs and receives the outputs. The controller design is finished as soon as there is some kind of satisfaction about its performance. As so, it can be seen as an indivisible entity, because there is no use for the discrete parts (the rules). This does not imply that the knowledge base (or other parameters, e.g., membership functions) could not be modified. This is indeed the case when there is the need to adjust on-line the performance of the controller ([2],[16]). It happens in situations where some parameters of the system change, and the controller has to be modified to track the new behavior of the system.

The size of the controller depends heavily on the quantity (number of input variables) and granularity (number of membership functions per variable) of the inputs. Although the process of rule determination is modular, meaning that when "thinking" in some rule one should not be too much concerned about the others (one thinks in each rule separately), it is known that the expressive power of fuzzy logic decreases with the increase in the knowledge base size.

Building a big centralized monolithic machine that has the capacity of doing many things might not be the best way to address a difficult problem. On the other hand, the decentralized (modular) way of thinking, or the bottom-up approach ([14]) advocated by artificial life, could facilitate the construction of such a controller, leading to a behavior based architecture. So, “keep the knowledge base modular” and “find a fuzzy controller for each behavior” (or module of the application) can be seen as good heuristics for solving large problems through fuzzy logic.

A new question then arises: how can simultaneous knowledge bases be articulated, each one showing some kind of competence different from each other? This is the issue of the arbitration problem, which is central in behavior based autonomous vehicles and also central in the study reported in this paper. In concrete, the experience about controlling a Khepera mobile robot is presented (see Fig. 1).

The organization of the document is as follows. In Sect. 2 some approaches to the application of fuzzy logic in autonomous vehicles are referenced, pointing out the connection to the work reported here, when applicable. In Sect. 3 the fuzzy behavior thematic is introduced; mainly, the constitution of a fuzzy behavior is presented. In Sect. 4 the integration of several fuzzy behaviors or the arbitration problem is considered. In Sect. 5 the results of the conduced experiments are presented. Finally, in Sect. 6 some conclusions are drawn and directions for future research are pointed out.

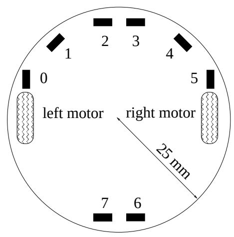


**Fig. 1.** Mobile robot Khepera with its eight infrared distance sensors (labeled from 0 to 7) and two motors

## 2 Relations to Other Works

Good references to start on fuzzy logic and to revisit still are [10] and [11].

An utilization of fuzzy control to the steering of a mobile vehicle can be found in [3]. Other utilizations of fuzzy control in the classical sense, with a comparison with neural control, for navigation of a mobile platform can be found in [8] and [19].

Implementations of adaptive fuzzy control applied to motor control and to navigation can be found in [2] and [16], respectively.

An implementation of distributed fuzzy control with applications to mobile robotics can be found in [6], with more focus on physical distributedness of computation rather than competence distributedness, as is favored here. Similarities with this work can be found in [5] and [7]. Namely, the work reported in [5] embraces a larger domain of application than the reported here. A combination between two separated fuzzy controllers (one for each actuator) and non-fuzzy controllers in a navigation system can be found in [9]. The major resemblance is with the work of Saffiotti, Ruspini and Konolige, which can be appreciated in several references, such as [1], [17] and [18]. This work extends into other problems beyond navigation, as is task planning.

## 3 Fuzzy Behaviors

A behavior is considered here as the reaction of the vehicle in the presence of some condition (environmental, in the typical situation). Why should fuzzy behaviors be used? First of all the term *fuzzy* relates with the fuzzy logic scientific topic, i.e. how the behaviors are implemented, and not to the behaviors itself. Fuzzy logic is used for several reasons; among them are:

- a way to get a smooth reaction to the environment (the vehicle turns as it gets closer to some obstacle);
- it is relatively easy to express the knowledge to do just that;
- an adequate representation for declaring knowledge which is:
  - uncertain (the sensors do not deliver correct data due to obstacle color and texture, functioning temperature dependency, etc.),
  - incomplete (the information captured by sensors is local and not global, due to environment scanning inside a tight 3-D cone), and
  - approximate (if there is imprecision, then the data is of an approximate nature only).

In spite of these attributes, there are not so many works in decision control in autonomous vehicles using fuzzy logic as are without it. This may happen because the main problems are the autonomy of the vehicle decision control and its level of competence, and not only the trajectory smoothness, for instance. However, the bigger the vehicle, the more concern should be put in the smoothness of the trajectories, if one wishes to have a safe and good use of batteries and not to break any electrical and mechanical part. In small vehicles there is no inconvenience in using threshold values to switch from small velocities to higher ones, to make tight bends or to stop suddenly; this is the case of the Khepera mobile robot.

One should not mix the anticipation of a bend (smoothness in turning) with fuzzy logic. This anticipation will be achieved if the sensor types and disposition allows so. In the case of Khepera, the sensors are not very good concerning correctness, range and linearity, but they are good enough for the goal at hand.

Fuzzy logic allows to have a smoothness in actuation, in the sense that no matter the amount of stimulus, there is always a corresponding amount of actuation. To this corresponds, most of the time, some degree of trajectory smoothness. As a consequence, the sensibility of the reaction is higher with fuzzy logic.

### 3.1 Constitution of a Fuzzy Behavior

For the time being the simplest form of behavior representation is considered, whose block diagram is illustrated on the left side of Fig. 2. It is similar to the one proposed in [12]. The fuzzy controller (FC) has the ability to create action, according to the designers intention. The knowledge declared in FC should code some particular competence necessary to the performance of the vehicle; examples used in this work are: how to drive in the direction of light and how to avoid left, right and front obstacles. FC receives input from the environment through the sensors and creates action through the actuators. It is also possible to use the notion of fuzzy state of the vehicle either as input or output variables.

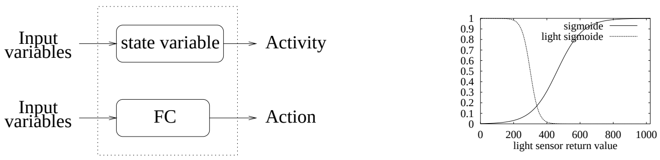


**Fig. 2.** Left: Behavior block diagram. Right: Shapes used in the experiments for the state variable membership functions

It is a special feature of the proposed approach that each behavior does not depend on others, at least during the design process. In this sense, each behavior knowledge base is determined only in terms of the type of (qualitative) response the behavior shall produce for each (qualitative) environmental condition. When constructing the rule base it should be clear for the designer what the vehicle must do when the behavior being constructed is active.

The state variable block in the left side of Fig. 2 is composed by a fuzzy predicate and the corresponding membership function whose intention is to measure how evident are the conditions of application of the behavior action in the present state of the environment, by returning a value belonging to the interval [0,1]. In other words, if it is considered that the behavior fuzzy controller codes some competence or know-how, then the state variable measures the degree of membership of the environment state to the conditions of application of the fuzzy controller. The environment conditions to satisfy are the ones that matter from the point of view of the behavior competence. The block input could refer to sensor values (reflecting the state of the environment) or to internal conditions (reflecting the state of the vehicle). By measuring how important are some environmental or internal conditions, the activity value is informing the final actuator evaluation module (the arbitrator) for how much the action values supplied by the behavior should be taken into account. It is up to the arbitrator to decide taking in consideration the action and activity value delivered by each behavior. Note that the state variable and FC do not influence each other.

It is also desired that the global conduct of the vehicle be dependent on some condition. Assume that this condition is internal to the vehicle, like in the following situation: as soon as the energy level drops below the energy threshold value, the vehicle should look for energy. This is accomplished by knowing that the recharging station is somewhere near a wall; so the vehicle starts to follow walls as soon as it finds one. It can be said that the vehicle's global behavior shifted from a *wander* behavior to a *wall follow* behavior. What does this situation have to do with the fuzzy behavior proposed so far? The issue is that the vehicle should consider left, right and front obstacles in a different way as it did before (when it was wandering). So the fuzzy controllers in avoid-left-obstacles, avoid-right-obstacles and avoid-front obstacles must be different. This is achieved by letting more than one fuzzy controller reside inside the action module. Finally, there is the need to switch between the several FC's available. This kind of rotary switch depends on some variable; in this case an internal state variable as it is energy. The new block diagram of a fuzzy behavior with more than one way of implementing its competence can be found in Fig. 3. It should be noted that the capacity to choose the active fuzzy controller resides in the arbitration module (the details will be presented in Sect. 4).

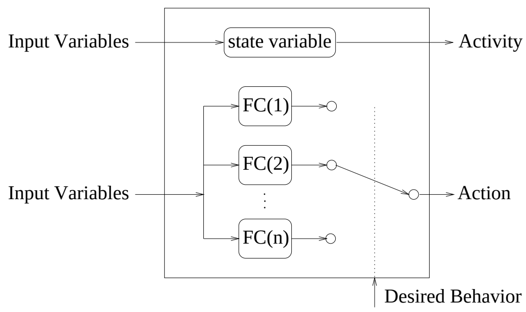


**Fig. 3.** Block diagram of the extended behavior

### 3.2 A Fuzzy Behavior Example

An example of an avoid-front-obstacles behavior is considered. The fuzzy controller is characterized by having two input variables (front-distance and difference-in-front-readings) and two output variables (left-motor and right-motor). Each variable is defined by having three fuzzy terms, which are: *far*, *medium* and *close* for input variable front-distance; *negative*, *zero* and *positive* for input variable difference-in-sensor-readings; and *forward*, *stop* and *backward* for both left-motor and right-motor output variables.

The membership functions of each fuzzy term associated to each input variable are presented in parts a) and b) of Fig. 4 (observe that sensor values range from 0 to 1023). The same happens in part c) in relation to the output variables (observe that motors should receive any integer value from -10 to 10). In what respects the shape of the membership functions, they where adjusted by hand in a way to compensate the non-linearity of the infrared distance sensors in Khepera.

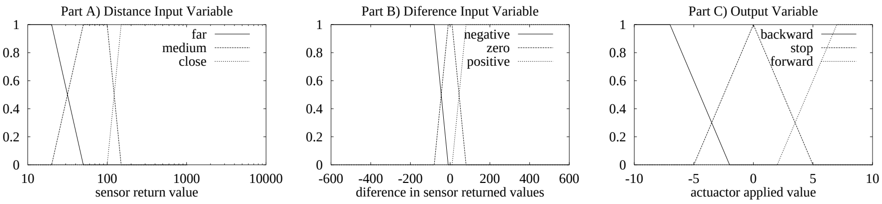


**Fig. 4.** Membership function definitions: part a) *far*, *medium* and *close* for input variable front-distance (note the log scale on the $x$-axis), part b) *positive*, *zero* and *negative* for input variable difference-in-sensor-readings and, finally, part c) *backward*, *stop* and *forward* for output variables left-motor and right-motor

The crisp input values are evaluated from the values of sensors '2' and '3' (recall Fig. 1) using expressions 1 and 2. In the case of Khepera, the two front sensors are parallel, and lower sensor values represent longer distances. So when there is the need to know the front distance it is safer to take the maximum of both front sensors. Also, the right side of expression 2 is a simple way to have a measure of the approach angle to an obstacle.

$$v_{\mathrm{dist}} = max(sensor[2], sensor[3]) \ . \tag{1}$$

$$v_{\mathrm{dif}} = sensor[2] - sensor[3] \ . \tag{2}$$

The activity value is a function of sensors '2' and '3' because these are the only ones pointing to the front. The activity is evaluated by applying the returned value from equation 1 to a sigmoide like expression, which returns a value in the [0,1] interval. This function is labeled *sigmoide* in the right side of Fig. 2. Thus the satisfaction of every predicate is measured in each iteration step.

Finally, the knowledge base is represented in tabular form in Fig. 5. As an example of how to read the table, consider the rule in the top-left rectangle: "if distance is *far* and difference-in-sensor-readings is *negative*, then left-motor is *forward* and right-motor is *forward*".

## 4 Fuzzy Behavior Arbitration

Behavior arbitration is a central topic in autonomous vehicles, but the corresponding in fuzzy logic, which can be called as controller arbitration, is not so common. The question behind arbitration is how to decide when there are several actions that can be performed.

In the non-fuzzy approach to behavior arbitration (the crisp way), the switching process depends on some threshold values, which means that the vehicle tends

| *dif* \ *dist* | *far* | *medium* | *close* |
|---|---|---|---|
| *negative* | $F_l$ $F_r$ | $S_l$ $F_r$ | $B_l$ $F_r$ |
| *zero* | $F_l$ $F_r$ | $F_l$ $F_r$ | $F_l$ $F_r$ |
| *positive* | $F_l$ $F_r$ | $F_l$ $S_r$ | $F_l$ $B_r$ |

**Fig. 5.** Knowledge base for avoid-front-obstacles fuzzy controller, where $F$, $S$ and $B$ represent *forward*, *stop* and *backward*, respectively; the subscript $l$ denotes left and $r$ denotes right

to react the same way in some subset of input conditions. In a sense, the threshold values voids the richness of the environment, meaning that the increasing evidence of some environmental feature is neglected until it is considered to be strong enough. This is where fuzzy logic is considered.

What is *fuzzy* behavior arbitration? It is considered as a way to decide what the actuators shall receive taking into account the action and the relative importance of each behavior (the activity value presented earlier), and because these values carry no information about the outcome of each action (namely some contradictory actions could be sent to the arbitrator), some internal state of the vehicle is also considered in the decision process. This is a (too much) global definition that will be clarified in a short notice. In the time frame of this work, two methods of behavior arbitration were evaluated: "weight" and "max", which are in essence defuzzification operators. These are addressed shortly. The effect of the vehicle's internal state will be considered in Sub-sects. 4.2 and 4.3 and also in Sect. 5.

### 4.1 Two Arbitration Methods

A kind of defuzzification process is used in two different ways, whose expressions can be formulated as follows: let $N$ denote the number of behaviors, $M^j$ denote the $j$-th motor ($j$=left, right), $S_i$ be the value of the $i$-th behavior state variable and $A_i^j$ be the $i$-th behavior action for $M^j$, then:

- Arbitration by Weight

$$M^j = \frac{\sum_{i=1}^{N} S_i \times A_i^j}{\sum_{i=1}^{N} S_i} : \quad j = \text{left, right} \ . \tag{3}$$

- Arbitration by Max

$$M^j = A_I^j : \qquad S_I = \max_{i=1 \ldots N} S_i, \quad j = \text{left, right} \ . \tag{4}$$

The intuitive arguments behind each mode are the ones used for the defuzzification methods in fuzzy logic ([11]). In the first case, it is considered that every behavior is important, and so they must be considered in the actuators final value. In the last case, it is given total importance to the behavior showing maximum activity, as some research work based on ethology seems to consider ([20], [13] and [15] are some examples). The results of both evaluation methods are presented in Sect. 5.

### 4.2 Behavior Inhibition

As soon the behavior activity value is different from zero, it can be said that the behavior is active. Depending on the arbitration method, this behavior activity could influence how the arbitrator commands the actuators. Apart from these two facts, nothing has been said about the number of behaviors that are permitted to be active independently of their activity values. In other words, a situation in which there is the need to disregard some behavior could exist. Take the following example. The vehicle should look for a re-charging station (simulated by a light) when the level of energy drops below some predetermined value. The presence of light should be taken in consideration in case the vehicle needs to re-charge, and should be disregarded otherwise. This kind of conduct is achieved by letting the arbitration module to impose a zero on the look-for-light behavior activity value (in the case it does not need to re-charge), or to consider the activity value in the actuator evaluation (otherwise).

### 4.3 Behavior Incompatibility

When more than one behavior with the same intention or purpose exists, there is the need to decide which behavior is allowed to influence the actuators, because there must be exclusivity in the decision. The same argument holds for fuzzy controller switching inside a behavior, because this implements the same competence in different ways; and as so they are incompatible or could be contradictory. In the present approach and as mentioned before, there must be a state variable upon which to ground the decision to choose another fuzzy controller. In the illustrative example that has been used in the explanations, this state variable is energy; namely, when the available energy is below some threshold value the piloting mode is to follow walls, as opposed to wander which is the mode when the available energy is high enough.

## 5 Experiments and Results

Lets make a short recall of the experiment. Khepera wanders until it falls short of energy. As soon this happens, it looks for walls, because it is known that these are the re-charging station probable locations. When it finds one, it stops for a while (simulating re-charging time) and resumes movement in a different direction than before (selected randomly).

There are four fuzzy behaviors: 1) avoid-left-obstacles, 2) avoid-front-obstacles, 3) avoid-right-obstacles and 4) look-for-energy. Behavior 1 input is connected to sensors 0, 1 and 2[1] of Khepera (recall Fig. 1), behavior 2 input is connected to sensors 2 and 3 (according to expressions 1 and 2), behavior 3 input is connected to sensors 3, 4 and 5, and behavior 4 input is connected to sensors 1, 2, 3 and 4.

Behaviors 1, 2 and 3 have two fuzzy controllers in the FC block: one allowing wandering behavior and the other allowing wall following behavior. In each case, the state variable membership function used is the one labeled "sigmoide" in Fig. 2. In the case of look-for-energy behavior, the state variable membership function is the one labeled "light-sigmoide".

Each fuzzy variable is defined by three fuzzy terms, with the output fuzzy variables of wall following mode being the exception: there are four fuzzy terms in left-motor and right-motor variables. Since it is not easy to manually tune fuzzy models, the use of four fuzzy terms per variable was a simple way to make Khepera contour the obstacles. Concerning the fuzzy controller definition, the following operators were used: aggregation by min, composition by max and defuzzification by centroid.

Energy drops linearly with time from the maximum value (MaxEnergy) to '0'. Note that $\mu(\text{MaxEnergy}) = 1$ and $\mu(0) = 0$ for the energy state variable membership function. The threshold value to switch between wandering and following piloting modes is $\mu(energy) = 0.2$; which is an experimental value giving Khepera enough time to find the light source before the energy fades way.

Two kinds of experiments are presented: one where the behavior arbitration is made by "weight" and the other where the "max" method is used. The results of the simulation are presented in Figs. 6 and 7. The environment is composed by walls and by one light source representing the re-charging station (denoted by the symbol 'L'). The legend is as follows: the symbol '+' denotes walls, light lines represent trajectories where the vehicle is wandering and dark lines represent trajectories where the vehicle is following walls. Figure 7 illustrates the trajectory broken in a eight figure set for sake of clarity; it should be observed from left to right and top to bottom. There is an arrow showing the starting point and the heading of Khepera. The starting behavior is wander. Note that the lines denote the trajectory of the center of the robot, which means that it approaches more the walls than the impression given by the figures.

As a first observation, note that in the left side of Fig. 6 Khepera misses the presence of the light source because the evidence of light is always less than the evidence of a wall by its right side (this fact is asserted in the right side of Fig. 6, where the activity values for both behaviors are shown). In a real dependency of energy, the khepera would "die". It must be said that this is not always the case. If the distance to the wall increases by a small amount then it could be enough for the evidence of light to increase and the evidence of the wall to decrease, having as result Khepera finding the light source. Note that this could happen because the Khepera distance sensors are nonlinear and noisy; as a consequence

[1] It could be argued that sensor '2' is not on the left side of the vehicle, and as so it should not make part of behavior 1. This is partly true. The fact is that the presence of sensor '2' is most help-full in realizing the contour (left) obstacles mode.

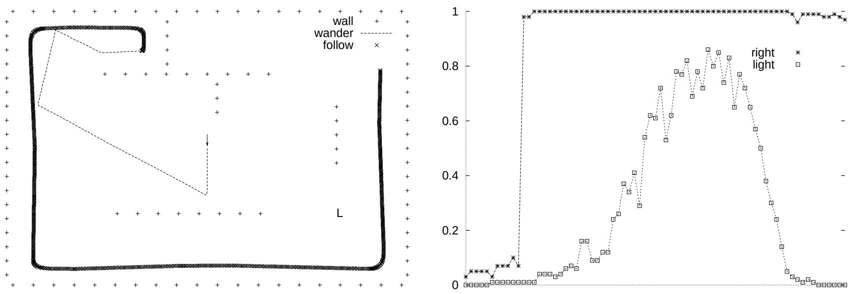


**Fig. 6.** Left: Simulated trajectory resulting from the “max” arbitration. Right: State variable values in wall following mode for fuzzy behaviors avoid-right-wall and look-for-light, near the vicinity of the light

the trajectories are not always equal. This is only a local proof and not a general demonstration of the inadequacy of the “max” arbitration method. However, this fact shows a drawback of this method when in comparison to arbitration by “weight” (in Fig. 6 the light position is the same as in Fig. 7).

As a second experiment, the “max” method is used and the light source is moved closer to the wall, as is illustrated in Fig. 8. Now an informal comparison can be made between both arbitration methods. This comparison is not straightforward, however, since the system shows dependence on initial conditions. Notwithstanding, it is observable that in the “weight” method walls act as more repulsive obstacles to the vehicle than what results from the “max” method, i.e., the vehicle spends more time turning. In fact, the amount of time that each behavior is influencing the actuators is larger in the “weight” method. Additionally, the “weight” method shows a better performance when guiding in corridors because it tends to cancel trajectory oscillations. Because this effect is not very clear in Figs. 7 and 8, the result of an experiment where the vehicle travels in a small corridor is provided in Fig. 9. It should be stressed that the knowledge bases used in each behavior are not changed from one arbitration method to another (i.e., from Fig. 7 to Fig. 8), which is an important feature of the proposed method.

## 6 Conclusion

This papers reports a work on autonomous vehicles, where piloting is accomplished with fuzzy logic. Our main motivation was to conclude over the adequacy of fuzzy logic to support a behavior based approach to autonomous vehicle piloting. Each behavior is composed by an action module, implemented by at least one fuzzy controller, and an activity value which is (for now) implemented by a membership function, supported on the environment state.

Each behavior is connected to the arbitrator and there is no hierarchy among behaviors. The arbitrator role is to create action from the action and activity val-

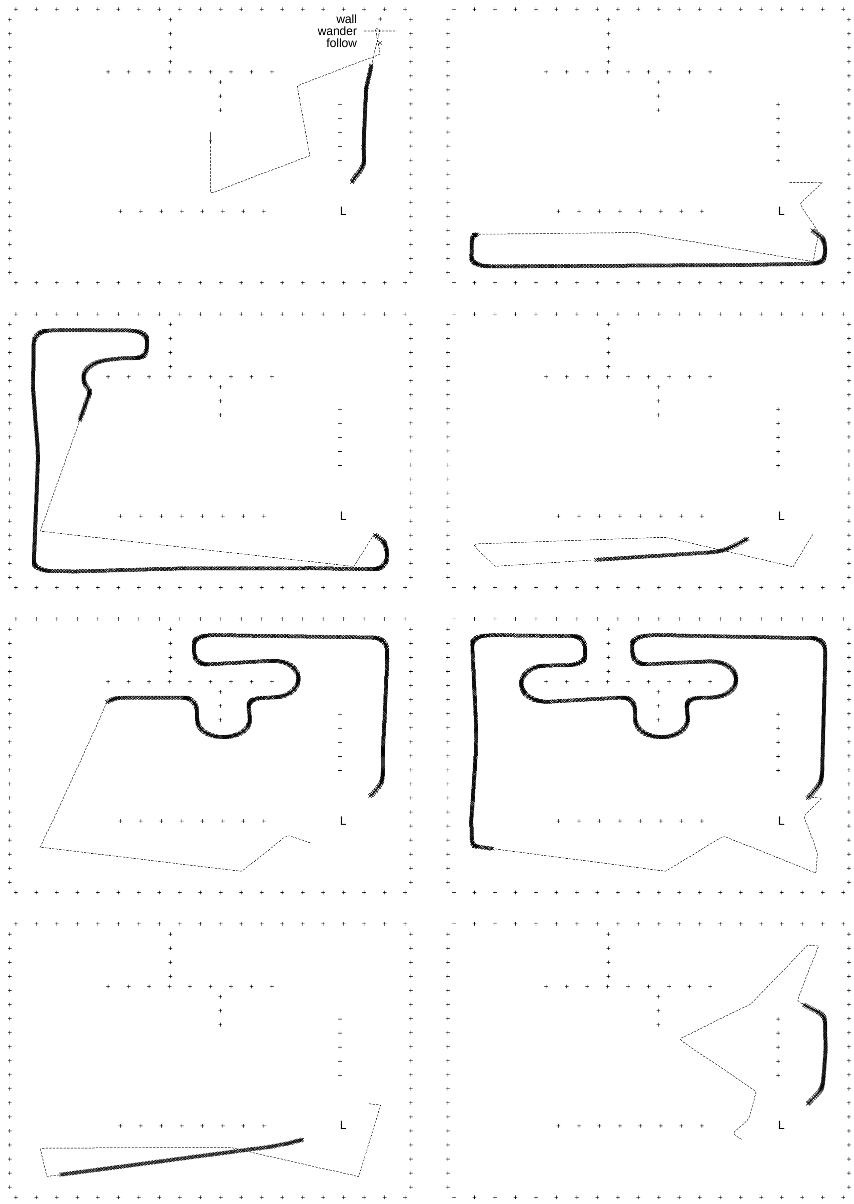


**Fig. 7.** Total simulated trajectory resulting from the "weight" arbitration (the trajectory is separated in pieces for sake of clarity; time flows from left to right and top to bottom)

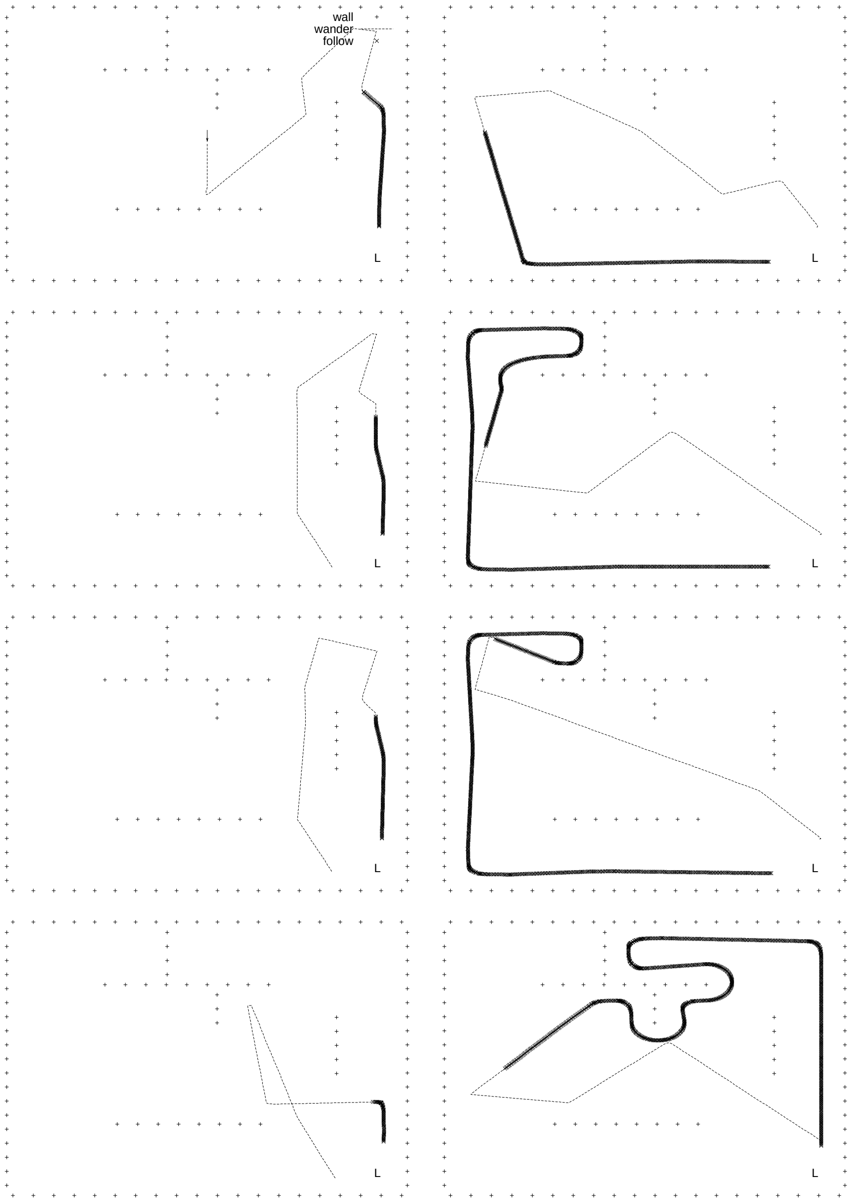


**Fig. 8.** Total simulated trajectory resulting from the "max" arbitration (the trajectory is separated in pieces for sake of clarity; time flows from left to right and top to bottom)

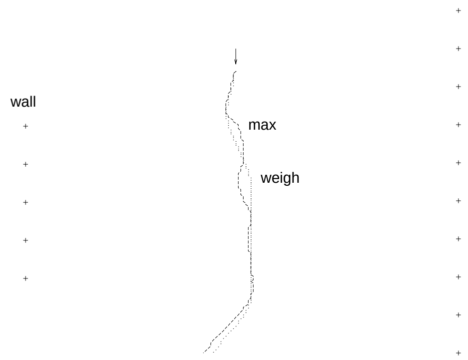


**Fig. 9.** Simulated piloting in a corridor with arbitration by "weight" and by "max", where the arrow indicates the starting point and the initial direction of the movement

ues supplied by each behavior. Two ways for realizing arbitration were proposed, which in essence are defuzzification methods: "weight" and "max".

Results of simulation experiments were presented, where the adequacy of the proposed approach to the piloting task was shown.

Not much attention was purposely given to find how optimal are the knowledge bases of the presented fuzzy controllers. This constitutes an important point for performance tuning, and as so it should be addressed, for instance with a genetic algorithm ([4]).

The arbitration problem is also a challenge, and as so it deserves future efforts.

Other open issues worthing future research are: i) creation of other useful behaviors as are piloting in tight corridors, piloting in agglomerated or overcrowded areas; ii) creation of monitoring behaviors with the capability of avoiding stagnation or trajectory pattern repetition, as happens when the vehicle contours an isolated obstacle; iii) architecture extensibility to many behaviors; iv) map building with the intent to know the re-charging station location; and v) development of strategies for performance measuring of long range autonomy.

## References

[1] Alessandro Saffiotti, Enrique H. Ruspini, and Kurt Konolige. Blending reactivity and goal-directedness in a fuzzy controller. In *Proceedings of the Second IEEE International Conference on Fuzzy Systems*, pages 134–139, San Francisco, CA, mar 1993.

[2] E. Cerruto, A. Consoli, A. Raciti, and A. Testa. Adaptive fuzzy control of high performance motion systems. In Robert J. Marks II, editor, *Fuzzy Logic Technology and Applications*, New York, 1992. IEEE.

[3] Filipe Cardoso, Luís Custódio, and Carlos Pinto-Ferreira. Fuzzy logic steering controller for a guided vehicle. In *Proceedings of the IEEE* $7^{th}$ *Mediterranean Electrotechnical Conference, MELECON'94*, pages 711–714, Turkey, apr 1994.

[4] Filipe Cardoso, Luís Custódio, and Carlos Pinto-Ferreira. Supervision of fuzzy controllers using genetic algorithms. In *Proceedings of the 1998 IEEE International Conference on Fuzzy Systems, FUZZ'98*, pages 1241–1246, Alaska, may 1998.

[5] Hartmut Surmann, Jörg Huser, and Liliane Peters. A fuzzy system for indoor mobile robot navigation. In *Proceedings of the IEEE International Conference on Fuzzy Systems*, pages 83–86, Yokohama, Japan, 1995.

[6] J. François Josserand, Gilles Mauris, Eric Benoit, and Laurent Foulloy. Fuzzy components network for distributed intelligent systems. In *Proceedings of the International Symposium on Intelligent Robotic Systems '94*, Grenoble, jul 1994.

[7] Jörg Huser, Hartmut Surmann, and Liliane Peters. A fuzzy system for realtime navigation of mobile robots. In *19th Annual German Conference on Artificial Intelligence, KI-95*, pages 170–172, Bielefeld, 1995.

[8] Jos Nijhuis, Stefan Neußer, Jörgen Heller, and Jochen Spünnemann. Evaluation of fuzzy and neural vehicle control. In Robert J. Marks II, editor, *Fuzzy Logic Technology and Applications*, pages 50–55, New York, 1992. IEEE.

[9] Kai-Tai Song and Jen-Chau Tai. Fuzzy navigation of a mobile robot. In *Proceedings of the 1992 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Raleigh, NC, jul 1992.

[10] Chuen Chien Lee. Fuzzy Logic in Control Systems: Fuzzy Logic Controller - Part I. *IEEE Transactions On Systems, Man, and Cybernetics*, 20(2):404–418, mar/apr 1990.

[11] Chuen Chien Lee. Fuzzy Logic in Control Systems: Fuzzy Logic Controller - Part II. *IEEE Transactions On Systems, Man, and Cybernetics*, 20(2):419–435, mar/apr 1990.

[12] Luís Correia and A. Steiger-Garção. A useful autonomous vehicle with a hierarchical behavior control. In *ECAL'95 European Conference on Artificial Life*, pages 625–639. Springer, 1995.

[13] Pattie Maes. A bottom-up mechanism for behaviour selection in an artificial creature. In Jean-Arcady Meyer and Stewart Wilsom, editors, *From Animals to Animats*. MIT Press, 1991.

[14] Pattie Maes. Behavior-based artificial intelligence. In Jean-Arcady Meyer, Herbert Roitblat, and Stewart Wilsom, editors, *From Animals to Animats 2*. MIT Press, 1993.

[15] David McFarland. The biology of behavior - criteria for success in animals and robots. In Luc Steels, editor, *The Biology and Technology of Intelligent Autonomous Agents*, volume 144 of *NATO ASI Series, Series F: Computer and System Sciences*. Springer, 1995.

[16] Nobuhiko Yuasa, Mitsuo Shioya, and Gunji Kimura. Follow-up characteristics of a small automatic guided vehicle system with fuzzy control. In Robert J. Marks II, editor, *Fuzzy Logic Technology and Applications*, pages 56–61, New York, 1992. IEEE.

[17] Alessandro Saffiotti. Some notes on the integration of planning and reactivity in autonomous mobile robots. In *AAAI Spring Symposium on Foundations of Planning*, pages 122–126, Stanford, CA, mar 1993.

[18] Alessandro Saffiotti. Fuzzy logic in autonomous robot navigation: A case study. In `http:\\iridia.ulb.ac.be/FLAR/HFC/home.html`, aug 1997.

[19] Seong-Gon Kong and Bart Kosko. Adaptive fuzzy systems for backing up a truck-and-trailer. *IEEE Transactions on Neural Networks*, 3(2), mar 1992.

[20] P. J. B. Slater. *An introduction to Ethology*. Cambridge University Press, 1985.