

# Fuzzy Logic Based Intelligent Agents for Reactive Navigation in Autonomous Systems

Antonio Gómez Skarmeta and Humberto Martínez Barberá  
Dep. de Informática y Sistemas. Universidad de Murcia

**Abstract:** *This paper deals with the application of fuzzy logic to the navigational component of an indoor autonomous system implemented by means of intelligent agents. The number of applications of fuzzy logic to mobile robots and autonomous vehicles control has increased significantly in the last years. The most well known arguments supporting fuzzy control are the ability to cope with imprecise information in heuristic rule based knowledge and sensor measurements, the interpolative nature of the fuzzy controllers, and the flexibility in the definition of non linear control laws.*

**Keywords:** *Autonomous Systems, Fuzzy Control, Intelligent Agents, Reactive Navigation*

## Introduction

The operation of an autonomous mobile robot in a real-world unstructured environment requires consideration of multiple issues. Principally, the controller must be able to operate under conditions of imprecision and uncertainty. For example, prior knowledge about the environment is, in general, incomplete, uncertain, and approximate. Perceptually acquired information is also typically noisy and incomplete. Furthermore, the execution of control commands is not completely reliable while the dynamics of real-world environments is complex and unpredictable. To cope with these difficulties, the controller must be able to respond reactively to unforeseen events as soon as they are perceived.

The very nature of the behaviour on an autonomous systems seems to be adequate to the use of intelligent and cooperative agents which implement the navigation control using fuzzy logic. In one hand, the autonomous robotics literature contains numerous examples of approaches that seek to deal with these problems by decomposition of the overall control function into a number of simple units, called behaviors (Brooks, 1986). Each

behavior is responsible for producing certain actions (typically, motion and perception actions), aimed at achieving or maintaining a particular goal such as staying away from obstacles. This interpretation fits with the notion of agents. Agents are autonomous entities capable of carrying out specific tasks by themselves, or through cooperation with other agents

In the other hand, intelligence is the degree of reasoning and learned behavior: the agent's ability to accept the user's statement of goals and carry out the task delegated to it. At a minimum, there can be some statement of preferences, perhaps in the form of rules, with an inference engine or some other reasoning mechanism to act on these preferences. If we also want to be able to operate under conditions of imprecision and uncertainty the intelligence of this agents needs to use fuzzy logic. Higher levels of intelligence include a user model or some other form of understanding and reasoning about what a user wants to do, and planning the means to achieve this goal.

Finally networked agent applications add a third dimension to the picture. Mobility is the degree to which agents themselves travel through the network.

Some agents may be static, either residing on the client machine (to manage a user interface, for instance) or instantiated at the server. Mobile scripts may be composed on one machine and shipped to another for execution in a suitable secure environment; in this case, the program travels before execution, so no state data need to be attached.

To summarize, the combination of these three components let us have a multiagent system offering a decentralized control model that can be used in the development of the reactive navigation of the autonomous systems.

In the following section we will describe the architecture we propose to address with intelligent multi-agents. Then, we will state the control systems, that is, how agents will be instantiated and how they will cooperate with each others. In the next section we will show the hardware we are using and our current agents configuration. Last, we will present some conclusions and future trends.

### **Description of Agents Architecture**

A multiagent systems is composed of largely autonomous and decentralised components, called agents, cooperating together in performing complex solving problem tasks. The use of multiagent systems creates a framework which allows the inter-operation of a vast set of heterogeneous solutions to carry out the complex network management tasks. Deterministic agents, knowledge based agents, neural agents, etc. can coexist and even cooperate inside this framework, in what can be addressed as a "natural way". That is, offering black box interfaces, and thus, high degrees of encapsulation and modularity, and of course, supporting several interaction schemes.

In order to achieve a correct agent behaviour, it needs knowledge about itself.

This kind of knowledge includes information on the skills it has, and thereby the tasks it is able to perform. It also includes knowledge on goals it has to achieve and whether they could be done concurrently with communication tasks or not. An agent also has information on what to do when messages (requests from other agents) arrive. The agent also needs knowledge on the global problem to be solved, or, at least, the part of the problem relevant to it.

Much of the work on reactive navigation has been inspired by the layered control system of the subsumption architecture (Brooks, 1986), which tightly couples sensing and action. The emergent behavior of the robot is the result of the cooperation of independent reactive modules, each one specialised in a particular basic behaviour. This approach of competing and cooperative agents has also been applied in a fuzzy frame (Goodridge and Luo, 1994). However, since reactivity alone does not suffice to meet the needs of real world tasks, hybrid solutions have been devised that blend low level reactive behaviours with a high level plan, used to supervise overall mission execution by activating and deactivating behaviours either sequentially (Ollero et al., 1995) or by means of fuzzy transitions (Ruspini and Saffioti, 1993).

In our case we have defined an architecture based on several agents that implement low level reactive behaviours using fuzzy rules and that are supervised by a blackboard controller that decide which behaviour/intelligent agent has priority in each instant. In general the proposed architecture has been designed according to the following principles:

- Use of the mechanisms of encapsulation, isolation, and local control: each agent is a semiautonomous, independent entity.

- No assumptions are made about the individual agent's knowledge or their problem solving methods.
- Flexible and dynamic organization is allowed.

Any agent may include goals (that is, processes which start when the agent is "born"), services offered (the agent offers a set of services to the rest of the agents, and these services may be executed in a concurrent - as an independent process- or non-concurrent way) and services required (a list with the names of the services that this agent may need).

Finally in order to improve the performance of the system, the developing process is done in a simulation framework, where the fuzzy rules associated to the intelligent agents are learned using sample data, and once they are tuned the agents are sent to the autonomous systems where they are executed. In this way the agents correspond to the idea of Mobile Scripts.

## Control Systems

The basic building block of the fuzzy controller is a behavior. A behavior implements a motor skill of the agent, aimed at achieving a given goal. Behavioral skills are expressed as preferences over possible control actions from the perspective of achieving that behavior's goal. For example, a behavior aimed at following a given wall could prefer actions that keep the agent parallel to that wall at a "safe" distance.

Following Ruspini's semantic characterization, each behaviour is described in term of a desirability function:

$$Des_B : State \times Control \rightarrow [0,1]$$

that measures, for each state vector  $x$  and control vector  $c$ , the desirability  $Des_B(x,c)$  of applying the control  $c$  when the state  $x$  from the point of view of attaining goals

associated with  $B$ . Equivalently, we can say that  $Des_B$  associates each situation  $x$  with the fuzzy set  $C$  of control values characterized by the membership function  $\mu_C(c) = Des_B(x,c)$ .

Each intelligent agent corresponding to a behaviour is described by a collection of fuzzy rules like:

$$IF A_i THEN C_i, i = 1, \dots, n$$

where  $A_i$  is composed of fuzzy predicates and fuzzy connectives, and  $C_i$  is a fuzzy set of control vectors.

For each possible control value  $c$ ,  $C_i(c)$  quantifies the extent by which  $c$  is a good instance of  $C_i$ . From these fuzzy rules, a desirability function is computed by:

$$Des_B(x,c) = [A_1(x) \wedge C_1(c)] \vee \dots \vee [A_n(x) \wedge C_n(c)]$$

Intuitively, this equation characterizes a control  $c$  as being desirable in the state  $x$ , if there is some rule in  $R$  that supports  $c$  and whose antecedent is true in  $x$ . This interpretation of a fuzzy rule set is that of a classical (Mamdani type) fuzzy controller.

Regardless of its implementation approach, a desirability function specifies, for each input variable value, a ranking over possible controls rather than a unique control value to apply in that situation. The robot eventually employs this ranking to choose one specific control that is sent. A possible mechanism to accomplish that selection is centroid defuzzification, which computes the mean of possible control values, weighted by their degree of desirability. Another alternative is to use this information by a blackboard control mechanism in order to decide which action need to be applied or if it is possible a combination of them is the best alternative.

As we have previously indicated, we have used the blackboard approach in order to decide which behaviour/intelligent agent has priority in each instant. Next we describe the architecture of the system and how the different agents interact through the blackboard.

### Hardware and Agents Configuration

An AA-Kidsmate BMW Sport RC (radio-controlled in hobbyist use) 1:10 scale model car, powered by a 12V DC motor, serves as the robotic platform. (Figure 1). It has two degrees of control: steering (to turn wheels to left or right), and traction (to control sense to forward or backward). It has no brakes, so to stop the car it is needed to reverse the traction motor for a given amount of time (depending on the speed). The total weight of the robot is about two kilograms.

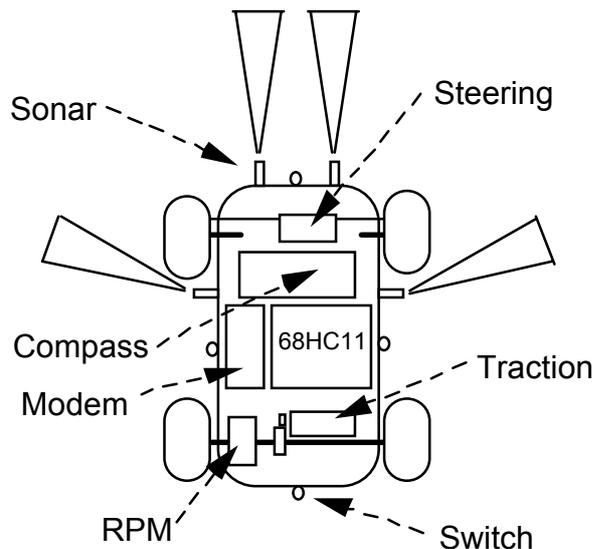


Figure 1: Hardware arrangement.

A variety of sensors (Figure 2) are mounted on the car: a custom Hall effect 3D compass for measuring direction and elevation, four ultrasonic sonar sensors (two facing forward and the other two one on each side), a shaft encoder sensor mounted on the main axle for measuring car speed, and four switches for detecting when the car has crashed (one forward, one backward, and the other two one on each side).

The robot contains a Motorola 68HC11 microcontroller based board (the HandyBoard), with a set of A/D-D/A input/output converters. The board has an integrated RS-232 serial port, where we have attached a custom bidirectional wireless 1200 bauds modem. This modem with the RC emitter/receptor is used to manually control the car and transfer sensor data to a 68040 Macintosh computer to perform fuzzy cluster analysis. The board is powered by a 9.6V NiCad rechargeable battery and the motors by a 12V AA/R6 battery pack.

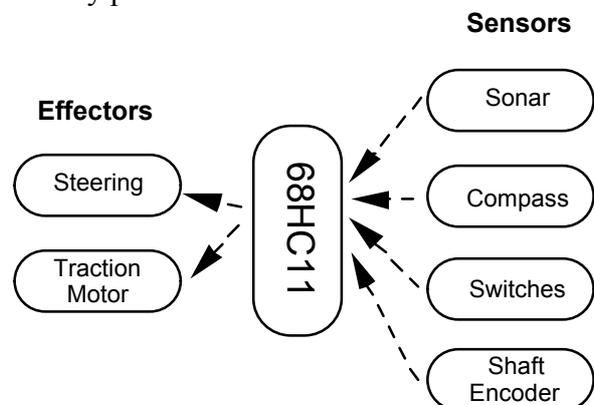


Figure 2: Effectors and sensors.

To achieve the goal of getting to a given point in an unstructured environment using a reactive control, the agents are organized (Figure 3) in the architecture described before. The sensor data is put on the blackboard as well as the knowledge obtained via inference by the agents. The control of the agents is carried out by a FIFO queue based scheduler with two priorities (normal and dangerous conditions). Each agent, based on the information available on the blackboard has the ability to instantiate other agents that it knows are suitable for processing its results.

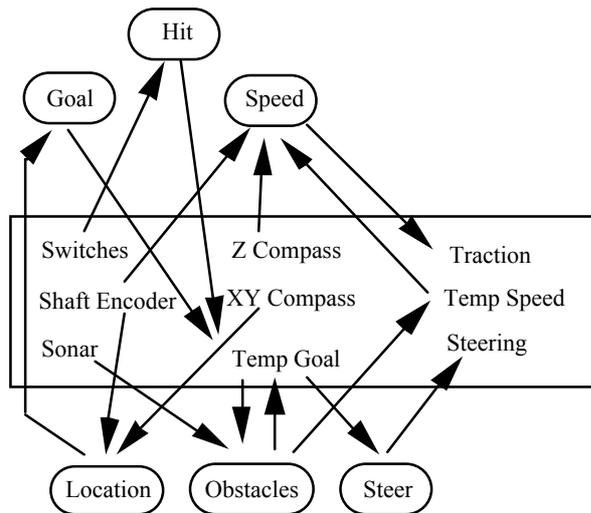


Figure 3: Intelligent Agents.

To illustrate the behaviour of the agents used, some of them will be described. There is an agent that controls the *speed* of the system. It reads data from the shaft encoder, the inclination and the temporal goal speed. With these data, by way of some fuzzy rules, it modifies the current sense and power transmitted to the traction motor. Other agents may cooperate with it using the blackboard, for example the *obstacles* avoidance agent, that reads data from the sonar and the temporal goal and modifies the new temporal goal and the new temporal speed as well. By this way, it is clear that agents can cooperate sharing information, and agents running with the highest priorities void other agents (and so considered less important at this very moment).

The learning process for the behaviour of agents is done using a simulation framework. In order to generate the fuzzy rules associated to each agent, different learning techniques are applied to sample input-output data obtained from the system behaviour in different real situations (Gásos and Martín, 1996). The use of fuzzy clustering techniques in order to define a first prototype of the fuzzy rules and its subsequent tuning using genetic algorithms (Delgado et al., 1995; Gómez et al., 1996), let us work with the system in an off-line manner, hence once the control system is

improved, the different agents can be loaded into the autonomous system.

## Conclusions and Future Trends

Although full evaluation of the system has not yet been completed, some preliminary conclusions can be advanced. In comparison to the conventional (centralized) architecture previously used, the distributed solution shows evident advantages:

- Interfaces are simpler, thus speeding up the development phase of the system's life-cycle.
- Systems are more reliable in terms of fault tolerance and protection against noise.
- Control is more versatile, in the sense that this approach facilitates the simultaneous use of several controllers based on different techniques (each with its own errors and response time depending on the problem state).

As a consequence, if the system has several controllers to perform a particular action or process, the error estimation received can be used to improve system accuracy by selecting the best controller.

## References

- [Beo95] Beom, H.R., Cho, H.S., "*A Sensor-Based Navigation for a Mobile Robot using Fuzzy Logic and Reinforcement Learning*", IEEE Trans. Syst. Man Cyber., vol 25, no. 3, pp. 464-477, 1995
- [Bon96] Bonarini, A., "*Learning Dynamic Fuzzy Behaviours from Easy Missions*", Proc. Sixth Int. Conf. Information Processing and Management of Uncertainty in Knowledge-Based Systems IPMU'96", Granada, Spain, pp. 1223-1228, 1996
- [Bro86] Brooks, R.A., "*A robust Layered Control System for a Mobile Robot*", IEEE J. Robotics and Automat., vol. RA-2, no. 1, pp. 14-23. 1986

- [Del95] M. Delgado, A.F. Gómez Skarmeta, F. Martín, "Generating Fuzzy Rules Using Clustering Based Approach", Third European Congress on Fuzzy and Intelligent Technologies and Soft Computing, August. 1995, pp. 810-814, Aachen, Germany
- [Gas96] Gásos, J., Martín, A., "Mobile Robot Localization using Fuzzy Maps", in Lectures Notes in Artificial Intelligence (Ralescu A. and Martín T. Eds.), Springer-Verlag, 1996
- [Gom96] A.F. Gómez Skarmeta, M. Delgado, F. Martín, "Using Fuzzy Clustering in a Descriptive Fuzzy Modeling Approach", Sixth International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems IPMU'96, July 1996, pp. 563-569, Granada, Spain
- [Hay85] Hayes-Roth, B.A., "Blackboard for Control", Artificial Intelligence, vol. 26, pp.251-324, 1985.
- [Hay95] Hayes-Roth, B. et al., "A Domain-Specific Software Architecture for Adaptive Intelligent Systems", IEEE Trans. on Robotics and Automation, vol. 21, no. 4, pp. 288-301, 1995
- [Hof96] Hoffmann, F., Pfister, G., "Evolutionary Learning of Fuzzy Control Rule Base for an Autonomous Vehicle", Proc. Sixth Int. Conf. Information Processing and Management of Uncertainty in Knowledge-Based Systems IPMU'96", Granada, Spain, pp. 1235-1240, 1996
- [Mat88] Matthies, L. and Elfes, A., "Integration of Sonar and Stereo Range Data using a Grid-Based Representation", Proc. IEEE Int. Conf. on Robotics and Automation, 1988.
- [Oll96] Ollero, A., García-Cerezo, A., Martínez, J.L., Mandow, A., "Fuzzy Tracking Methods for Mobile Robots", in Applications of Fuzzy Logic: Towards High Machine Intelligence Quotient (Jamshidi M. Ed.), Prentice-Hall, 1996
- [Pay86] Payton, D.W. "An Architecture for Flexible Autonomous Vehicle Control", IEEE Int. Conf. on Robot. and Automation. pp. 1838-1845, 1986.
- [Saf94] Saffiotti, A. et al., "Robust Execution of Robot Plans using Fuzzy Logic", in Fuzzy Logic in Artificial Intelligence: IJCAI'93 Workshop. Lectures Notes in Artificial Intelligence 847 (Ralescu A. Ed.). Springer-Verlag, pp. 24-37, 1994
- [Saf97] Saffiotti, A., Ruspini, E.H. and Konolige, K., "Using Fuzzy Logic for Mobile Robot Control", in International Handbook of Fuzzy Sets and Possibility Theory (Dubois D, Prade H. and Zimmermann H.J. Eds.), Kluwer Academic, 1997.
- [Vel94] Velasco, J.R., and Ventero, F.E., "Some Applications of Fuzzy Clustering to Fuzzy Control Systems", Proceedings of the 3rd Int. Conference of Fuzzy Theory and Technology, Durham, USA.
- [Vra95] Vranes, S., and Stanojevic, M., "Integrating Multiple Paradigms within the Blackboard Architecture", IEEE Transactions on Software Engineering, vol. 21, no. 3, pp 244-262, 1995