Fuzzy Modelling Operator Navigation Behaviors

A. García-Cerezo, A. Mandow and M. J. López-Baldán

Departamento de Ingeniería de Sistemas y Automática. Universidad de Málaga.
Plaza el Ejido s/n, 29013 Málaga (Spain).
e-mail: amandow@ctima.uma.es
Fax: (+34) 5 213 14 13- Tel: (+34) 5 213 14 07

Abstract

The paper presents a fuzzy modelling method for static and dynamic systems and its application as a tool for learning navigation behaviors from operator's actions. The modelling process starts from a set of experimental input-output data, and automatically produces a set of fuzzy rules, whose number depends on the desired degree of model's accuracy. The method is based on least squares techniques, and can be used in association with the most common fuzzy operators and sets. The paper illustrates the application of the method to model the behaviors of a human expert operator driving a real mobile robot equipped with ultrasonic sensors. Learning the fuzzy behaviors makes possible the adaptation of the robot to the environment, and saves design time and efforts. The resulting rulebases can be used as building blocks in executing purposeful and robust autonomous navigational tasks.

1. Introduction

Mobile robotics is a field where the application of fuzzy control techniques is beneficial on account of the non-linearities posed by the control system as well as the imprecision of vehicle models and sensor measurements. In general, there are two ways to generate a fuzzy controller. One is to obtain the rulebase directly from the experience of a seasoned operator or a knowledge engineer. The other consists on fuzzy modelling the operator's control actions or the process itself, which becomes necessary in complex systems where rules are difficult to state.

Fuzzy modelling techniques represent a good strategy to obtain reasonable nonlinear models. These can be achieved by means of a variety of different approaches, like least squares [7][13] and another regression techniques [6], gradient descent learning methods, which lie on some traditional methods used in neural nets [2] [10], and the application of stability criteria over experimental measures [1].

Unfortunately, many of these techniques are difficult to apply in the real world on account of their inherent limitations in some aspects like the type of fuzzy operators, defuzzification techniques, shapes of fuzzy sets and, more frequently, their important computational requirements.

The method presented in this paper makes use of efficient least squares techniques to identify the parameters of the rule consequents, as well as improved algorithms to determine the antecedent fuzzy partitions [4]. The result is an automatic fuzzy modelling method that achieves efficient and accurate models at a relatively low computational cost in comparison with existing techniques.

The paper presents results on how to use the identification method as a tool for the design of behavioral control for a real mobile robot, where input-output measures are obtained during human-driven experiments. Behaviors for navigation control have been widely and successfully applied over the last decade, following Brooks' subsumption architecture [3]. Basically, in most applications, behaviors are defined as specialized functions that establish a relationship between input values and control actions in order to achieve some particular objectives of autonomous navigation. Appropriately combining and arbitrating the activation of behaviors results in robust navigation in the face of uncertain sensors, unpredictable environments and a dynamic world. Fuzzy logic can be used in the frame of behavioral control for designing individual behaviors [11][15], processing input data from disparate sensors [9][14], or blending outputs from several active behaviors [11].

The paper is organized as follows. Section 2 presents the modelling procedure. Section 3 discusses the use of fuzzy logic for navigation behaviors. Section 4 describes the application of the method for the design of the fuzzy controller from human driven experiments, and section 5 presents the conclusions. Appendix A shows how the least squares techniques have been applied to obtain rule consequents.

2. Identification Algorithm

The rule base generated automatically by the modelling method consists of rules of the type:

$$R^k\!: IF\ (x_1\ is\ A_{1i}\ and\ x_2\ is\ A_{2j}\ and \ldots and\ x_n\ is\ A_{nm})\ THEN\ y\ is\ C_k$$

where k is the rule number; $i, j,.., m$ vary from 1 to the number of membership functions of the variables $x_1, x_2,..., x_n$ respectively; and C_k is the consequent parameter of the k-th rule, which is a numeric constant.

The fuzzy model is obtained through the following steps:

0-7803-3796-4/97/$10.00©1997IEEE

1. Input the desired accuracy of the fuzzy model.
2. Set the initial fuzzy inference rules.
3. Derive the inference error from the input-output data.
4. Repeat until the inference error falls within the desired accuracy:
4.1. Select the appropriate region to be divided.
4.2. Produce and add the corresponding new fuzzy inference rules to the existing ones.
4.3. Obtain the consequent parameters by the recursive least squares algorithm.
4.4. Derive the inference error from the input-output data.

The final model accuracy is expressed by the user in the first step in terms of an estimation of inference error, i.e. the difference between the real input/output data and the values generated by the model. Particularly, the root mean squared error has been used in the implementation:

$$D = \sqrt{\frac{\sum_{p=1}^{N_d} (y^*_p - y^r_p)^2}{N_d}}, \qquad (1)$$

where N_d is the number of available data, and y^*_p, y^r_p are the model-estimated output and the real output, respectively, for the p-th set of data.

In *step 2*, the rulebase is initialized with a minimum set of membership functions for each variable (e.g., 2 sets per variable). The sets must comply with the two-overlapped condition, i.e. they satisfy, for every value v of the input variable $x_i = v$, the following relations:

$$\mu_{Aj}(v) + \mu_{Aj+1}(v) = 1, \qquad \mu_{Ak}(v) = 0,$$

with $k \neq j, j+1$, where $\mu_{Aj}(x_i)$ and $\mu_{Aj+1}(x_i)$ are the membership functions of two consecutive fuzzy sets corresponding to two consecutive linguistic labels in the ordered sets. The consequent parameter of each of these rules is then computed by using the input/output data with the recursive least squares algorithm, as described in Appendix A.

Equation (1) is then used in *step 3* to estimate the inference error, which is used as the ending condition of the loop comprising *steps 4.1 to 4.4*.

The universe of discourse of each input variable is divided into regions, which are defined as the space where two consecutive fuzzy sets are overlapped. For instance, the example on Fig. 1.(a) has four regions (two for each variable), of which the one defined by sets A_{12} and A_{13} is shaded. In order to refine the rulebase, the method decides in *step 4.1* which of all these regions needs to be more finely adjusted to its input/output data. It must be noticed that this choice is essential because it affects the accuracy of the resulting model as well as the increase of the number of rules. In general, the selection should balance simultaneously the following priorities:

• Regions with a high deviation from the original input-output data.

• Regions with a higher number of input-output data, because their adjustment will have more effect over the general inference error.

• Wider regions, in systems with uneven distribution of input-output data, to avoid over-adjustment in small regions with great concentration of data.

The choice of the region might depend on a definition of inference error, like in [2], where the mean deviation is considered, leading to the selection of the region with the highest mean absolute error, regardless of its size or its number of measured data. This approach makes possible the division of regions defined by few data, which may be undesirable, since it might produce little improvement of accuracy for the whole system and even over-adjustment to noisy input-output data.

Alternatively, the proposed method contemplates the three conditions exposed above by selecting the region that maximizes the following expression:

$$E = \sqrt{\frac{w \cdot \sum_{p=1}^{N_d} (y^*_p - y^r_p)^2}{N_d}}, \qquad (2)$$

where w stands for the width of the region, i.e. the difference between the upper limit of the region and the lower one. This expression takes into account the fact that not all the regions have the same number of data because of their size, without abandoning the influence of the concentration of data N_d inside the region. The equilibrium between the concentration of data in a region and the importance of their accuracy in relation with its relative size is obtained by w/N_d Comparative studies have revealed that the use of (2) for region selection instead of an estimation of inference error (such as (1)) significantly reduces the ratio of number of rules to accuracy.

Then, a twofold partition of the selected region (see the example in Fig. 1) increases the number of rules (*step 4.2*). The limits of the two membership functions that originally defined the region must be modified so that the two-overlapped condition is maintained when the new one is

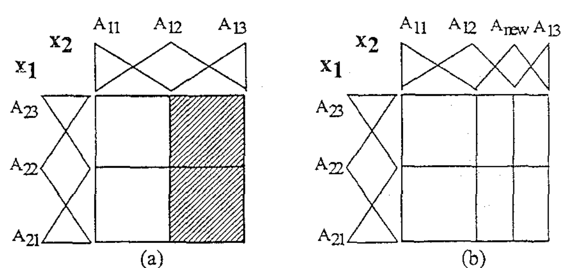

Fig. 1. The selected region (a) is subdivided and a new membership function is introduced (b).

inserted. The rule consequents corresponding to the new set of rule antecedents are computed in *step 4.3* by the least squares algorithm (see Appendix A).

Finally, expression (1) is used again to check the model accuracy. After a model is available, it can be further refined with new data (or improve its adjustment to the original data) by simply taking up the modelling loop again.

3. Fuzzy logic for the design of navigational behaviors

The behavioral control approach to purposeful mobile robots deals with the decomposition of complex tasks into simple units called behaviors. Thus, appropriately sequencing or blending the outputs of different predefined behaviors results on an "emergent behavior" that meets the mission's objectives. Depending on implementations, the complexity of these units may vary from low level actions like "approaching to the left" or "going ahead" to higher level activities such as "following a corridor."

The design of autonomous navigation behaviors for real world mobile robots is prone to a fuzzy approach. This is a nonlinear control problem in which the relationship between the real control variables and values perceived by the robot is unknown (due to imperfect sensors and/or actuators), and the models of the environment and the performing system are, at most, partially known. Moreover, behaviors are mostly reactive, i.e. they provide an output control value resulting from some input variables, namely sensor information about the robot and its surroundings. This trait allows for a mapping of inputs and outputs to antecedents and consequents of fuzzy control rules.

Directly programming the sets of rules is possible for relatively simple control actions, like following a wall [9] or a road [15]. Nevertheless, even in those cases, where advantage can be taken from the linguistic flavor of fuzzy systems, the definition of the rule base and the scaling factors can be a cumbersome time-consuming work, which requires successive refinements over a simulated system and/or the real robot. Moreover, more complex behaviors can be too hard to program, given the imprecisions mentioned above. Consequently, effort must be devoted to the development of techniques that facilitate fuzzy modelling approaches.

4. Identification of fuzzy behaviors

In their pioneering paper about fuzzy control for navigation, Sugeno and Nishida [12] demonstrated the usefulness of control rules obtained from an experienced operator's control actions for the particular activity of driving a model car along a crank-shaped corridor.

The Fuzzy Modelling Method presented above has been

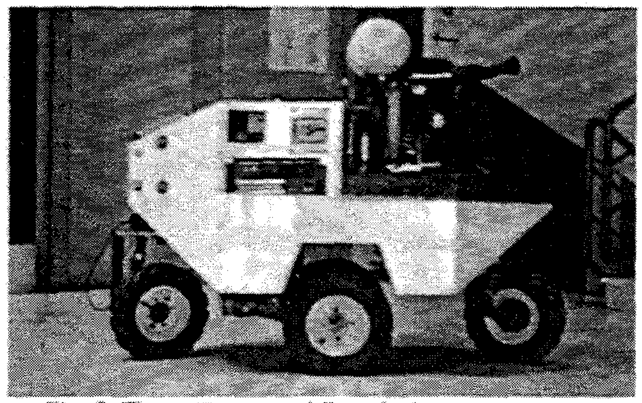

Fig. 2. The AURORA mobile robot.

used to implement fuzzy behaviors on the AURORA mobile robot (see Fig. 2), a four wheeled robot equipped with two ultrasonic sensors at the front and another one at each side, with a ranging distance of about 1.3 meters (see Fig. 3). This robot is 1.4 meters long by 0.8 meters wide [8].

The fuzzy behaviors have been incorporated within the navigation architecture depicted in Fig. 4. Basically, a plan, generated either by a user or an automatic planner, is a sequence of behaviors and their finishing conditions. A supervisor module is in charge of activating and deactivating the behaviors, as the conditions are detected through the sensing system. Sensor information is also used

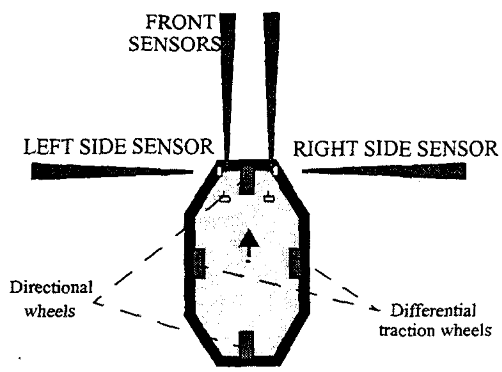

Fig. 3. AURORA's sensing ranges and locomotion.

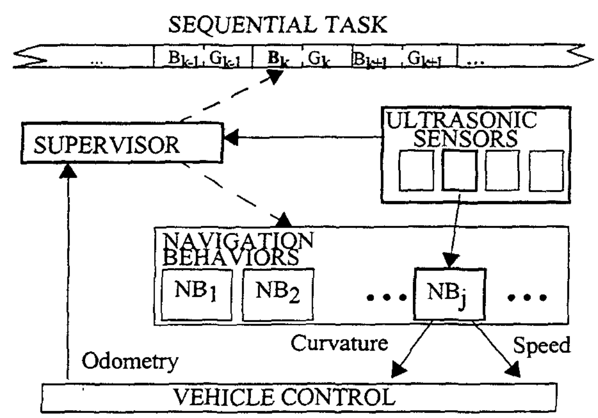

Fig. 4. Architectural framework.

by the behaviors to produce the control action, namely the desired speed and curvature.

The modelling tool has been used to model some of the behaviors needed to make up purposeful tasks in indoor environments, which are sketched in Fig. 5: following walls, following corridors, aligning to walls, and inside corners. The first two behaviors are characterized by tracking a continuous natural feature of the environment (walls and corridors), and the variables needed to accomplish them are just the relative position to the landmark and its first derivative. Details about the wall following behavior model can be found in section 4.1. Aligning to walls and inside corners are more complex behaviors; both are similar in that the robot reaches a wall perpendicularly and has to align with it without collisions. Their input variables involve complex relations between several sensors. The inside corner implementation is described in section 4.2.

In general, a number of teaching experiments (ranging from 15 to 40) where performed for each of the behaviors. The input-output data were collected at each control interval (every 0.15 s). Only the sensors directly involved in the behavior were considered (the corresponding side sensor in the case of walls, both sides for the corridor, and the two front sensors together with the inner side sensor for wall alignment and inside corners). In any case, it is important to consider for the model the least possible number of input variables, because of the size and complexity of the resulting rulebase.

Regarding the output variables, only curvature has been considered, whereas speed has been kept at a constant value appropriate for each behavior. Speed control can be dealt with independently [5], because it involves consideration of several kinds of restrictions which depend not only on the status of the current behavior, but also on vehicle constraints, operational requirements, security in the face of moving or unexpected obstacles, etc.

During the measurement experiments, the vehicle was driven by the operator through a joystick connected directly to it (see Fig. 6). One of the advantages of fuzzy identification of the driver's behavior is that as long as the available sensors provide enough information to achieve the task, the information perceived by the operator for training

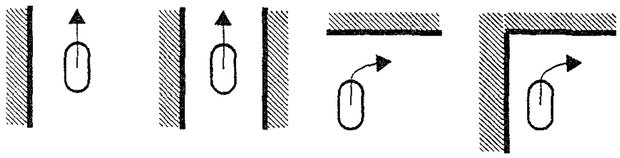

Fig. 5. Some useful behaviors for indoor navigation. A) Following wall, B) Corridor, C) Align with wall, D) Inside Corner.

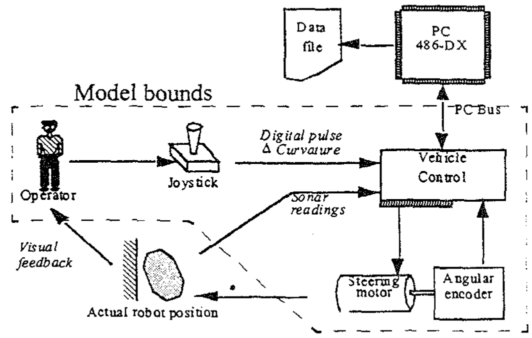

Fig. 6. Operator control and model bounds.

the robot does not necessarily have to be the same. In fact, the driver's experience and ability to guide the vehicle rests on first-hand visual perception of the operation rather than the interpretation a flow of sensor readings, even if these are presented graphically.

Another important decision in the design of the teaching experiments refers to where the output control value is actually measured from. Model bounds in Fig. 6 show how instead of measuring digital pulses from the joystick in order to obtain a "pure" model of the operator, output curvature has been measured from the system response. Thus, a high degree of dynamic and kinematic admissibility is obtained for the modelled control actions.

4.1. Wall and Corridor Following.

Tracking a natural landmark like a corridor or a wall is one of the most common behaviors for indoor navigation. Obtaining a curvature control value for these cases is possible by alternative methods, like PID control [15] or direct fuzzy programming [9]. Nevertheless, there are two interesting reasons for acquiring a fuzzy model from operator guidance: first, obtaining the rulebase directly by the modelling tool saves time and developing effort in parameter tuning; second, obtaining a controller that is adapted to the working environment, with the capability of reacting to real conditions, like wall irregularities.

The speed for these behaviors was 0.3 m/s, and it has been observed that the resulting behavior works well with speeds of at least +/- 0.1m/s. In these experiments, the operator tried to drive the vehicle along a line on the floor, drawn in the middle of a corridor or parallel to the wall (0.85 meters from it). Most of the interest in these training experiments was on starting from a variety of different initial postures, so that the state space was covered as much as possible with input/output data. On the other hand, no attention was paid as to obtain an homogeneous input/output sample. In fact, data is mostly concentrated around the equilibrium status (centered along the corridor or aligned to the wall at the desired distance), since all training experiments eventually reach it.

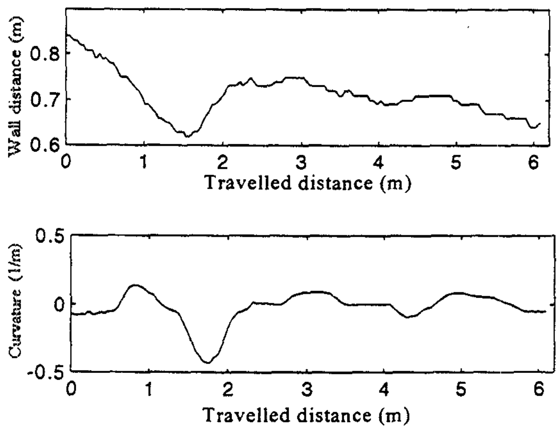

Fig. 7. Curvature and distance to wall in one of the teaching experiments.

Two input variables were used in both behaviors: the measured distance and its first difference for the wall behavior, and the difference between both sides and its first difference in the case of the corridor behavior. The resulting rulebases contain 12 and 18 rules respectively. It must be mentioned that for the corridor case, more input variables were tried (considering independently the measures from both sides and their first differences), but the result was not good enough to compensate for an increase in the number of rules.

Fig 7 shows data from one of the corridor training experiments. The control action produced by the human operator was quite nervous and over-actuated, so it takes some time for the robot's dynamics to react completely. When the equilibrium is reached, oscillating (and contradictory) actions were taken. All of these were filtered by the fuzzy model, whose results on a similar experiment are presented in fig. 8. The controller's actions are smoother, and they do not oscillate so much when the center is reached. These results are similar to what could be expected if alternative control methods were used, but at a lesser development cost.

4.2. Inside Corner and Wall Alignment.

These behaviors, in which the robot reaches more or less perpendicularly a wall and alings with it, are much more complex than tracking a single landmark. In the first place, they involve several different inputs, not just the deviation from an objective. Then, several important decisions must be taken based on inaccurate input data: when to start turning, and what curvature exhibit at each moment. Sensor inaccuracies arise not only from operational errors, but also about the moment in which the data were obtained, since the robot is moving while sensors are being automatically fired. While turning, some of the sonars lose contact with the references because of the incidence angle of the beam. The possibility of a collision if improper actions are taken is

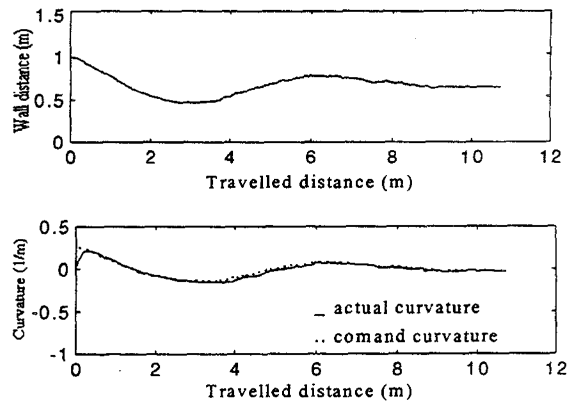

Fig. 8. Performance of the wall fuzzy controller.

rather high.

Taking into account these circumstances, an analytical solution for the controller is very difficult to obtain, and programming the rules by eliciting operator knowledge is inefficient. Fuzzy modelling actual operation clearly emerges as an effective way for coping with this kind of problem.

The model was obtained from experiments in which several almost-perpendicular incidence angles to the wall were considered. The operator eventually aligned the robot to the wall at approximately the same distance as in the wall following behavior, so that an eventual merging of behaviors is possible when combining them along a more complex mission.

The input/output values of one of the teaching experiments for the inside corner behavior are shown in Fig. 9. The model receives three inputs: the two front sensors and the wall side sensor. Null values from the sensor have been omitted in the figures, but have actually been considered in the model. In particular, the sensors provide a

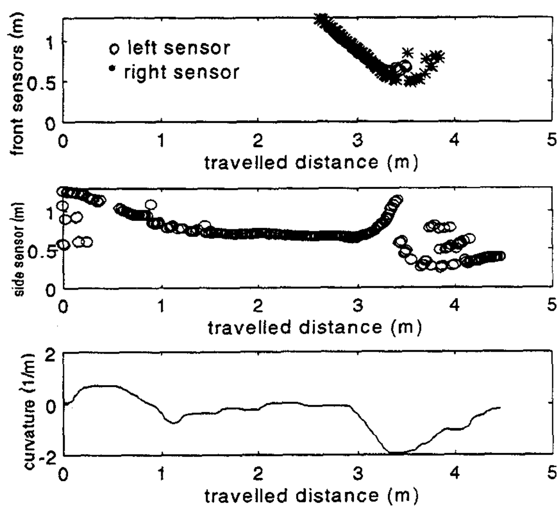

Fig. 9. Input-output measures while following an inside corner.

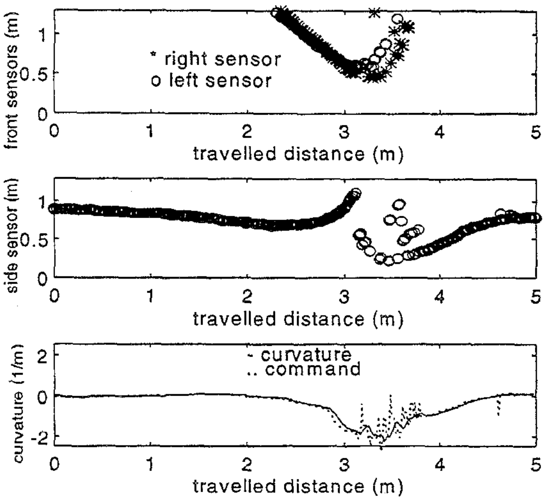

Fig. 10. Fuzzy model performance for the inside corner behavior.

value of approximately their upper limit value (which in this case is 1.3 m) when they fail to detect anything. The speed for these experiments was kept constant at 0.1 m/s. The resulting rule base has 36 rules . In the first flight of these experiments (up to 3 meters in Fig. 10) the behavior is like wall following. Note that during the turn, control commands are not very smooth on account of the scarce amount of information used (with a relatively small number of rules), but they are filtered by the system, which produces a good performance. Results for the wall alignment behavior are very similar to the ones presented for the inside corner.

5. Conclusions and future work

The proposed modelling method identifies the consequent parameters by using the recursive least squares algorithm. The use of the least squares method results in a reduction of computing time, since it produces all the rule consequents at once, instead of using a consequent refining iteration, typical of another methods. Its flexibility allows the use of a variety of inference mechanisms. Moreover, consequent sets of any shape can be defined, as long as they have the same area. The identification algorithm defines the shape and position of the input membership functions and can be incrementally improved with further examples.

The fuzzy modelling methodology has been successfully applied to learning navigation behaviors from a human operator, overcoming the limitations of real-world sensors and actuators. The control actions issued by the navigation behaviors are admissible from the dynamic and kinematic viewpoint, since they are directly modeled from the reaction of the robot system, thus filtering sharp control actions of the driver. Contradictory actions produced by a "bad teacher", like the oscillations around the equilibrium status, are dealt with by the least squares method, which optimizes the output value for each combination of antecedents.

Other useful behaviors for indoor navigation could also be considered for fuzzy modelling. Turning around a corner would involve modelling the delay elapsed between the detection of a turning corner by the side sensors (i.e. a free space) and the moment when the robot actually reaches the turning position. Another relevant behavior in real-world robots is obstacle avoidance, which is suitable for fuzzy modelling [2] because of its complexity. Operator control actions can be directed to modifying either the original path or the speed. However, although some interesting results have been obtained for the presented robot configuration, further work is needed with a more comprehensive sensor system (like a laser rangefinder or a sonar ring).

6. Acknowledgments

This work has been done within the framework of the project TAP96-1184-C04-02 of the C.I.C.Y.T. (Spain). We are grateful to our colleague Dr. Jorge L. Martínez for his insightful comments on this paper.

7. References

[1] J. Aracil, A. García-Cerezo, A. Barreiro, A. Ollero, "Stability Analysis Of Fuzzy Control Systems Based On The Conicity Criterion". *Fuzzy Logic: State Of The Art*, pp. 487-496. Kluwer Academic Publishers. 1993.

[2] S. Araki, H. Nomura, I. Hayashi, N. Wakami, "A Self-Generating Method of Fuzzy Inference Rules," *IFES'91 Fuzzy Engineering toward Human Friendly Systems, Part VIII, Application,* (1991), pp. 1047-1058

[3] R. A. Brooks, (1986), "A Robust Layered Control System For a Mobile Robot". *IEEE Journal on Robotics and Automation.*Vol. RA-2, No. 1, pp. 14-23.

[4] A. García-Cerezo, M. J. López-Baldán, A. Mandow, "An Efficient Least Squares Fuzzy Modelling Method for Dynamic Systems." CESA'96 IMACS Multiconference, Symposium on Modelling, Analysis and Simulation. Lille, France, 1996.

[5] K. Kant, S.W. Zucker, "Toward Efficient Trajectory Planning: The Path-Velocity Decomposition." *The Int. J. of Robotics Research*, Vol. 5, No. 3, Fall 1986, pp. 72-89.

[6] Y. Lee, C. Huwang, Y. Shih, "A Combined Approach to Fuzzy Model Identification," IEEE Transactions on Systems, Man, and Cybernetics, Vol. 24, No. 5, May. 94, pp. 736-744.

[7] E. H. Mamdani, "Application of Fuzzy Algorithm for Control of Simple Dynamic Plant," *Proc. IEEE*, 121 (12), (1974), pp. 1585-1888.

[8] A. Mandow, J. M. Gómez de Gabriel, J. L. Martínez, V. F. Muñoz, A. Ollero, "The Autonomous Mobile Robot AURORA for Greenhouse Operation" *IEEE Robotics and Automation Magazine.* Vol. 3, No. 4, December 1996.

[9] A. Ollero, A. García-Cerezo, J.L. Martínez, A. Mandow.

"Fuzzy Tracking methods for Mobile Robots", *Applications of Fuzzy Logic: Towards High Machine Intelligence Quotient Systems*, Vol. 17, Ch. 17. Prentice Hall series on Environmental and Intelligent Manufacturing Systems, forthcoming 1997.

[10] S. M. Smith, D. J. Comer, "Automated Calibration of a Fuzzy Logic Controller Using a Cell State Space Algorithm," *IEEE Control Systems* (1991) pp. 18- 28.

[11] A. Saffiotti, E. H. Ruspini, K. Konolige. "Using Fuzzy Logic for Mobile Robot Control," *International Handbook of Fuzzy Sets and Possibility Theory*, D. Dubois, H. Prade and H.J. Zimmermann, Eds. Kluwer Academic, forthcoming in 1997.

[12] "M Sugeno and M. Nishida,"Fuzzy Control of a Model Car", *Fuzzy Sets and Systems* 16 (1985) 103-113. North Holland

[13] T. Takagi, M. Sugeno, "Fuzzy Identification of Systems and Its Applications to Modeling and Control," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-15, No. 1, January/February 1985

[14] G. Ulivi, "Emergent Techniques for Mobile Robots and Autonomous Vehicles." ICASAV-95 IFAC Workshop Toulouse, France, October, 1995

[15] H. Wu, W. Chang, H. He, P. Xi. "A Fuzzy Control Method for Lateral Control of Autonomous Land Vehicle", *Mobile Robots X*, W. J. Wolfe and C. H. Kenyon, editors, SPIE Proceedings, Vol. 2591, pp. 125-131, SPIE, Oct. 1995.

Appendix A: Applying least squares to fuzzy consequents.

The development that follows leads to an expression of the output value u that allows the use of the least squares optimization in the proposed modelling method. It assumes a Center of Sums (COS) defuzzification approach in which the final output set is obtained through summed aggregation, as well as a constraint that makes all the fuzzy consequents have the same area.

The output value u, in the discrete universe case, is given by the equation that represents the center of the area of the final output fuzzy set $\mu_{c'}(u)$ through summed aggregation:

$$u = \frac{\sum_{j=1}^{d} \mu_{c'}(u_j) \cdot u_j}{\sum_{j=1}^{d} \mu_{c'}(u_j)}, \qquad U = \{u_1, \ldots, u_d\}, \tag{3}$$

where $\mu_{c'}(u_j)$ represents the membership function associated with the output inferred by the set of rules defined in the rule base. For the Sum operator, $\mu_{c'}(u)$ is calculated as:

$$_{c'}(u) = \sum_{r=1}^{m} \mu_{c'_r}(u) \tag{4}$$

In the case of *scaled* fuzzy outputs the output activation level α_i is used to scale the fuzzy consequent of rule R_i. Thus, the membership function of the scaled output fuzzy set is defined by:

$$\mu_{c'_i}(u) = \alpha_i(\bar{x}) \cdot \mu_{c_i}(u) \qquad \forall u \in U \tag{5}$$

Combining (4) with (5):

$$\mu_{c'}(u) = \sum_{r=1}^{m} \mu_{c'_r}(u) = \sum_{r=1}^{m} \alpha_r(\bar{x}) \cdot \mu_{c_r}(u \tag{6}$$

If all the consequent fuzzy sets are defined to have the same area, the following condition applies:

$$\forall r \qquad \sum_{j=1}^{d} \mu_{c_r}(u_j) = C, \tag{7}$$

where C is a constant. Defining

$$a_i = \frac{\alpha_i}{\sum_{r=1}^{m} \alpha_r},$$

then $a_i = a_i(\bar{x})$ (because of α_i) and $0 <= a_i <= 1$, so the value of the control action can be obtained as:

$$u = \begin{bmatrix} a_1(\bar{x}) & a_2(\bar{x}) & \ldots & a_m(\bar{x}) \end{bmatrix} \begin{bmatrix} u_{CG_1} \\ u_{CG_2\Upsilon} \\ \ldots \\ u_{CG_m} \end{bmatrix} \tag{8}$$

It must be considered that any type of sets that satisfies the condition expressed in (7) can be used for the consequents whenever their centroids coincide with the u_{CG_r} in the parameter vector.

This method is applied in the presented problem conditions to an algebraic equation of the type:

$$\bar{y} = A \cdot \bar{p}$$

where $\bar{y}$ represents measured output data, A represents input data, and $\bar{p}$ is the vector of consequent rule parameters to be identified.

If antecedents are fixed, the least squares method can be easily adopted by using (8), where the set of input data is given by vector $\bar{a} = [a_1(\bar{x}), a_2(\bar{x}), \ldots, a_m(\bar{x})$, $\bar{u}$ represents the vector of measured outputs, and the parameter to be identified is given by vector $[u_{CG_1} \ldots u_{CG_m}]^T$, which represents the centers of gravity of the consequents.

With N sets of measured input-output data, the algebraic equation to be solved can be expressed as:

$$\begin{bmatrix} u_{(1)} \\ u_{(2)} \\ \ldots \\ u_{(k)} \\ \ldots \\ u_{(N)} \end{bmatrix} = \begin{bmatrix} a_1(\bar{x}_{(1)}) & a_2(\bar{x}_{(1)}) & \ldots & a_m(\bar{x}_{(1)}) \\ a_1(\bar{x}_{(2)}) & a_2(\bar{x}_{(2)}) & \ldots & a_m(\bar{x}_{(2)}) \\ \ldots & \ldots & \ldots & \ldots \\ a_1(\bar{x}_{(k)}) & a_2(\bar{x}_{(k)}) & \ldots & a_m(\bar{x}_{(k)}) \\ \ldots & \ldots & \ldots & \ldots \\ a_1(\bar{x}_{(N)}) & a_2(\bar{x}_{(N)}) & \ldots & a_m(\bar{x}_{(N)}) \end{bmatrix} \begin{bmatrix} u_{CG_1} \\ u_{CG_2} \\ \ldots \\ \ldots \\ \ldots \\ u_{CG_m} \end{bmatrix}$$

and consequently, the vector $\bar{p}$ of u_{CG_r} can be computed by the recursive least squares method.