Input-Output Fuzzy Modelling Applied to a Mobile Robot

LOPEZ-BALDAN, M.J., RUIZ-GOMEZ, J., FERNANDEZ, R., GARCIA-CEREZO, A.
System Engineering and Automation Department
University of Málaga
E. T. S. Ingeniería Industrial. Campus El Ejido. 29071 Málaga.
SPAIN

Abstract: - Two modelling techniques have been used in order to obtain path curvature fuzzy models for a mobile robot. It has been applied a design process based on experimental input-output data. Two types of rule bases have been obtained depending on the structure of the generated fuzzy models (Mamdani and Sugeno based). The first approach is an inductive learning based on classical Quinlan ID3 method. The second one performs least squares identification and an automatic rule generation by minimizing an error index. The results obtained in both cases lead us to conclude about the satisfactory behaviour of these kind of modelling techniques. This sort of tool is especially useful to practitioners and not expert modelling people in order to get the description of system behaviour and the subsequent controller design.

Keywords: Fuzzy Modelling, Inductive Learning, Qualitative Modelling, IEEE CSCC'99 Proc.pp. 6001-6006

1 Introduction

Model Based Fuzzy Control (MBFC) deals with the design of a fuzzy controller given a conventional, non-linear open loop model of the system under control. An approach to the design of fuzzy controllers needs more depth knowledge than the common used *heuristic approach* based on a shallow knowledge and the experience of the user. MBFC works on the stability, robustness and performance analysis of the resulting closed loop system [11]. To build a fuzzy controller it looks easier to use an approach that derives fuzzy control rules from fuzzy model rules.

The aim of this work is the application of two modelling approaches in order to obtain a mobile robot model, based on fuzzy rules. These fuzzy modelling techniques achieve automatic rule generation from experimental input-output data.

The paper layout is: first, a brief introduction about related work in fuzzy modelling, next an overview of the two methods and, finally, both applications to the RAM mobile robot dynamics.

2 Fuzzy Modelling

Modelling, by means of fuzzy rules, a non linear process from a set of data that reflect past behaviour of the plant is an alternative approach to conventional I/O modelling of dynamic systems. And, based on a fuzzy model of a process, a non-linear control system can be designed.

Fuzzy identification is adopted due to the difficulty of conventional mathematical modelling on certain systems, specially non-linear ones, and due to the ability of approximating any real function by a set of fuzzy rules [2][19].

Fuzzy identification has been treated from different viewpoint. First approaches, based on Sugeno inference method, arise on 1985 [17].

There are several methods based on inductive learning: Sison and Chong approach [16] applies ID3 to generate a fuzzy rulebase, and it also allows to select the relevant inputs, though it does not build a sequence of models as the Sugeno approach; Tani *et al*. [18] identify the premises structure and select the effective input variables by means of ID3; Delgado and Gonzalez, [3] work on an inductive learning process based on the frequency of appearance of certain patterns from raw data.

Close to the inference error learning described on section 4.2, a self-tuning method of Nomura *et al* [9] is accomplished through the gradient descent method for parameter identification and Sugeno type rules. The procedure of building a fuzzy model from process data and the subsequent controller design based on this model is presented in [1], where a Sugeno-Takagi model is constructed from data and, then, derived into a Mamdani model.

3 Mobile Robot RAM

The dynamical system of the RAM-2 mobile robot, developed in the System Engineering and

Automation Department at Malaga University, has been considered to obtain the fuzzy models. The vehicle (figure 1) was developed for application on indoor and outdoor industrial environments [8][10]. It has four wheels located in the vertices of a rhombus. The front and rear wheels in the longitudinal axis are steered at the same time by a DC motor with a rigid link. The two parallel wheels are driven independently by DC motors.

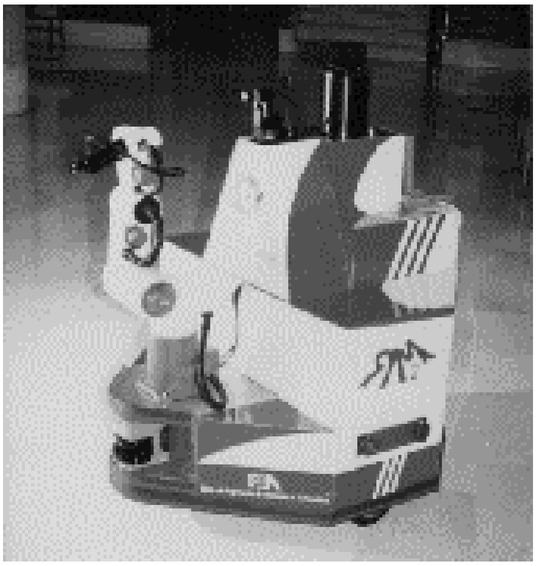

Fig. 1. RAM-2 Mobile Robot.

The locomotion system can provide a zero turning radius and a top speed of 1.6 m/sec when the vehicle moves along a straight path, although this speed decreases when the curvature of the vehicle is increased.

RAM-2 is equipped with different types of sensors: sonars, lasers and cameras. It also includes a robot arm to achieve specific tasks. The robot control is done by a PC-based system, under the Lynx real-time operating system.

4 Description of the methods

This section describes the procedures of building two different fuzzy models from input-output data. The modelling methods generate the rule bases trying to reduce an error among the real output data and the output generated by the fuzzy models. This estimation is computed by using expression (1).

4.1 Inductive modelling

The first method [15] for the robot curvature modelling is twofold based in a classical inductive learning method as Quinlan ID3 [7] and the well known Takagi-Sugeno modelling approach [17].

From both methods several outstanding features are taken. From Takagi-Sugeno, the partitioning of variables ranges and data fuzzification by means of these partitions. From ID3 (a classifying method based on entropy or information gain) the decision tree building technique.

The goal is to obtain a set of fuzzy rules that describes the behaviour hidden in the experimental data. The raw data obtained by experimentation are converted into a set of ‘experiences’ prior to the application of the ID3 inductive inference procedure. Consequents are in Mamdani format.

Fuzzy partitioning is built by means of equally spaced triangular fuzzy sets overlapped in a variable value between adjacent sets.

Though it shares with the classical Takagi-Sugeno method two main features of clustering approach: the partition of variable ranges, and data fuzzification by means of these partitions. An outstanding feature is that the rules are not stated in the Sugeno format, with the consequents as linear function of the input variables. The consequents are performed in the Mamdani type, i.e. as fuzzy sets. This is an advantage that improves the model comprehension or transparency. But this method differs from that of Takagi-Sugeno in the building of the decision tree, in the input spaces partitioning techniques and in the rules generation. A detailed description of this approach can be found in [14].

As the measured data from a physical system are usually continuous ones, the problem of translating them into fuzzy sets is critical. Two important points on numeric-fuzzy conversion of data arise: the number of sets to split every variable range, and the way the intersection between two adjacent fuzzy sets is treated.

To begin with, every variable range have been divided in two equal sets, by the middle of the range value, with a variable overlap between adjacent sets varying from 0 to 100 per cent. Also it is possible to perform a second way to split the variable range on the basis of critical points of the I/O curve [11], owing a derivative of zero value or not possessing derivative at all.

As ID3 is a learning from classes method, we need first a classification of all variables. As facts or «experiences» we consider a set of data resulting in one measurement, i.e. a row in a numerical data table. All the numerical data are fuzzified. The raw data obtained by experimentation give raise to a set of fuzzy experiences as these shown in Table 1.

Table 1. Fuzzy Experiences

Experience #	Attribute **x1**	Attribute **x2**	Consequent **y**
1	NB	NS	NS
2	PS	Z	PS
3	Z	PS	Z
4	...	...	..

NB: negative big; NS; negative small; PS: positive small; Z: zero.

Afterwards, a link between «experiences» and rules must be established, in order to apply ID3 to a fuzzy rules generation process.

For every input variable, we find the consequents that reflect the action of every fuzzy set of the selected input. The probability P_i is obtained by dividing the number of experiences in the input related with a same consequent, by the total number of times the antecedent is in the table.

After computing the amount of information, we make a tree based on the attributes or input variables. The tree development continues by the node with greater entropy. In a leaf node all its elements belongs to the same class. A rule is given by the path from the root node to a leaf node. The models sequence stops when the performances for a certain model achieves a desired value.

Model validation can be accomplished by an numeric index based on root mean squares error (1) between experimental data and modelled data, also by using an error histogram or last by viewing a 3D graphic of experimental and modelled surfaces.

4.2 Inference error learning

In this subsection, a modelling method, based on least squares identification, is outlined. A detailed description can be found on [4].

The rule base, automatically generated 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 ...*

... and x_n *is* A_{nm}) THEN y is y_{CG_k}

where k is the rule number; i, j,.., m the membership function of x_1, x_2,..., x_n respectively and y_{CG_k} is the k-th rule consequent parameter.

This is a Sugeno-type rule in which the consequent parameter stands for the COG of the consequent fuzzy set. The membership functions $A_{nm}(k)$ are triangle sets. The *and* connective corresponds to the algebraic product, and the final output of the fuzzy model inferred from the whole set of implications is given by a modified average defuzzification due to the overlapped condition [7].

The fuzzy model is obtained through the following steps:

1. Input the desired accuracy of the fuzzy model.
2. Set the initial fuzzy inference rules.
3. Derive inference error from input-output data.
4. Repeat until the inference error falls down 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 inference error from I/O data.

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

$$e = \sqrt{\frac{\sum_{p=1}^{N_d} (y_p^* - y_p^r)^2}{N_d}} \tag{1}$$

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 rule base is initialised with minimum set of membership functions for each variable (i.e., 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:

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

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 these rules is then computed by using the I/O data into a recursive least squares algorithm [5]. The equation to solve is given by (2).

$$\begin{bmatrix} u_{(1)} \\ u_{(2)} \\ \vdots \\ u_{(k)} \\ \vdots \\ u_{(N)} \end{bmatrix}_{N\times 1} = \begin{bmatrix} a_1(\bar{x}_{(1)}) & a_2(\bar{x}_{(1)}) & \dots & a_m(\bar{x}_{(1)}) \\ a_1(\bar{x}_{(2)}) & a_2(\bar{x}_{(2)}) & \dots & a_m(\bar{x}_{(2)}) \\ \vdots & \vdots & & \vdots \\ a_1(\bar{x}_{(k)}) & a_2(\bar{x}_{(k)}) & \dots & a_m(\bar{x}_{(k)}) \\ \vdots & \vdots & & \vdots \\ a_1(\bar{x}_{(N)}) & a_2(\bar{x}_{(N)}) & \dots & a_m(\bar{x}_{(N)}) \end{bmatrix}_{N\times m} \cdot \begin{bmatrix} u_{CG_1} \\ u_{CG_2} \\ \vdots \\ u_{CG_m} \end{bmatrix}_{m\times 1} \tag{2}$$

where $\overline{y}(\overline{u})$ are the N experimental output data, $\overline{a} = [a_1(\overline{x}), a_2(\overline{x}), \ldots, a_m(\overline{x})]$ is calculated with the experimental input data and the parameter vector to be identified by least squares algorithm is $[u_{CG_1} \quad u_{CG_2} \quad \ldots \quad u_{CG_m}]^T$.

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

The universe of discourse of each input variable is divided into regions, which are defined as the area where two consecutive fuzzy sets are overlapped. For instance, the four regions (two for each variable) of figure 2 (a).

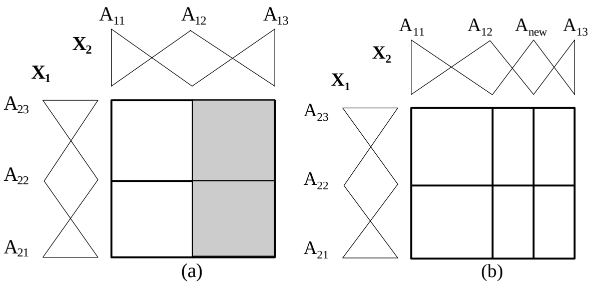

Fig. 2. Fuzzy regions.

In order to refine the rule base, the method calculates in *step 4.1* the adjustment of the regions to its input/output data, and selects the one with larger differences. It must be noticed that this choice is essential because it affects the accuracy of the whole resulting model as well as the increase in 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 [4]. The three conditions above exposed are comply with the selection of the region that maximizes the following expression:

$$e = \sqrt{\frac{w \cdot \sum_{p=1}^{N_d} (y_p^* - y_p)^2}{N_d}} \qquad (3)$$

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.

Then, a twofold partition of the selected region (see the example in figure 2) increases the number of membership functions (*step 4.2*). The rule consequents corresponding to the new set of rule antecedents are computed in *step 4.3* by the least squares algorithm [5].

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.

5 Application results

The described methods have been used to obtain the curvature modelling of a mobile robot [13]. The table of measured I/O data is all the information about this real system needed to be introduced as input in the modelling process. Input variables are the desired velocity and curvature values, and the output variable is the actual path curvature taken by the mobile robot, measured as the difference between the length travelled by the left and right wheel and also the heading wheel angle.

The data used as inputs of the modelling methods are the results of several experiments realised with the RAM-2 autonomous robot. The experiments to get the data were made by controlling in a manual way the robot RAM-2 introducing curvature and speed commands and collecting by the odometric system the real curvature and speed achieved, these data were collected every 50 msc., although only the stabilised data were considered. The commands of speed introduced varied between -1.6 to 1.6 m/sc. with increments of 0.1 m/sc., and the curvature commands between 0.5 and -0.5 m^{-1} with increments of 0.02 m^{-1}.

5.1 Model 1

From the 388 randomly chosen data, a subset of 70 data has been selected, newly by using a random selection. From these data, 55 form the training set and 15 the test set. The values have been scaled to avoid negative curvatures.

In this application the model is obtained by using ID3 learning, middle point splitting, centre of gravity, 5% overlapping and searching for a minimum number of rules. It can be seen that in the generated model the input velocity variable is considered as a single fuzzy set (i.e. the influence of this input it is not relevant to the output). The input curvature is split in four fuzzy sets and the output variable in seven sets. The model has ten rules. Fuzzy set splitting is depicted in figure 3.

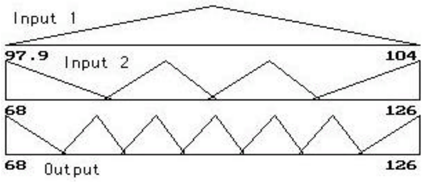

Fig. 3. Model fuzzy inputs and output.

A report showing the numerical values of the triangular fuzzy sets, the performance index is 1,0549 (about 1,07 %) and the resulting rules is in the following Table 2. The first column value 1 in inputs and output tables means the set in that row is active. The same meaning has values 10.

Table 2. Fuzzy model report.

Report: NDAT55
Model error index: 1.0549
Date:25/ 3/99, H: 10:13:28
MFS Matrix (I/O fuzzy sets):
Input 1:

1	97.9	101.1	101.1	104.2

Input 2:

1	68	68	68	83.22
1	81.78	90.47	90.47	97.72
1	96.28	105	105	112.2
1	110.8	126	126	126

Output:

1	67.97	67.97	67.97	76.67
1	75.84	80.82	80.82	84.96
1	84.13	89.1	89.1	93.25
1	92.42	97.39	97.39	101.5
1	100.7	105.7	105.7	109.8
1	109	114	114	118.1
1	117.3	126	126	126

Model Rules (RLS Matrix):

10	10	10	10
10	1	1	1
10	1	1	2
10	1	2	2
10	1	2	3
10	1	2	4
10	1	3	4
10	1	3	5
10	1	3	6
10	1	4	6
10	1	4	7

5.2 Model 2

By applying the method described in section 4.2, the fuzzy model of Table 3 has been obtained. Three blocks represent each rule: the first ones for the antecedent part, while the last one is the column representing the centres of gravity of the consequent sets. The parameters of the antecedent membership functions correspond to the triangular fuzzy sets for the input variables (V1 for the desired curvature and V2 for the desired velocity) in the rule. Only four rules have been considered because of the difficulty in the reduction of the inference error obtained with this model. An inference error of 0.0281 has been reached for 337 data, computed by the mean square error detailed in expression (1). It means that the adjustment is almost exact and, therefore, it is not possible more improvement due to the good result of this index.

Table 3. Set of rules corresponding to Model 2.

V1					V2	y_{CG_k}
-3.5	-3.5	3.0	-0.48	-0.48	0.9	-3.5151
-3.5	-3.5	3.0	-0.48	0.90	0.9	-3.4451
-3.5	3.0	3.0	-0.48	-0.48	0.9	3.0064
-3.5	3.0	3.0	-0.48	0.90	0.9	2.9572

The distribution of the errors can be seen in the histogram of figure 4. In this graphic, a set of 51 test data has been used to obtain the error between the real output data and the fuzzy output data. The total of errors is close to 0, even spurious data, such as indicates the index of the fuzzy model inference error. The few data little more outlying coincide with spurious data corresponding to bad lectures or disturbances.

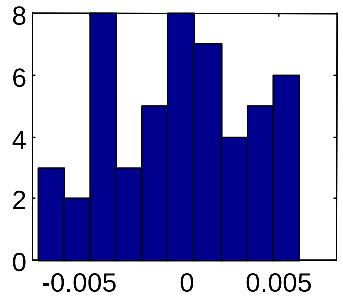

Fig. 4. Histogram of data test errors.

6 Conclusions

Two different methods of fuzzy modelling have been applied to the RAM mobile robot curvature modelling.

Both methods works well to obtain a set of fuzzy rules that models the robot behaviour. Therefore, it can be stated that fuzzy modelling is useful to develop models of complex non-linear systems by using experimental input-output data.

By using Mamdani type format, the comprehension of a set of not many rules is much more intuitive to a human agent than any other model formulation (mathematical, neural, or genetic algorithm based). It offers a transparency feature helpful in control design.

Due to the distribution of the curvature data measured from the mobile robot, the Sugeno based approach reaches least number of rules and lower performance index.

References:

[1] Babuska, R., Sousa, J. and Verbruggen, H.B. Model-based design of fuzzy control systems. *Proceedings of 3rd. European Congress on Fuzzy and Intelligent Technologies*, Aachen (Germany), 1995.

[2] Buckley, J. and Hayashi, Y., Fuzzy input-output controllers are universal approximators, *Fuzzy Sets and Systems* 58, 1993, pp. 273-278.

[3] Delgado, M. and González, A., An Inductive learning procedure to identify fuzzy systems", *Fuzzy Sets and Systems 55*, 1993, pp. 121-132.

[4] Garcia-Cerezo, A. J., Lopez-Baldan, M. J. and Mandow, A. An Efficient Least Squares Fuzzy Modelling Method for Dynamic Systems. *Proceed. CESA'96. IMACS Multiconf. Symp. Modelling, Analysis and Simulatio*n. Lille (Fr), 1996, pp. 885-890.

[5] García Cerezo, A., Mandow, A. and López Baldán M.J., Fuzzy Modelling Operator Navigation Behaviors, *Proc. of 6th IEEE International Conference on Fuzzy System*, Barcelona (España), 1997.

[6] Hellendoorn, H. and Driankov. D. (Eds.), *Fuzzy Model Identificationl*, Springer, 1997.

[7] López-Baldán, M.J., Contribuciones al modelado y control de sistemas dinámicos mediante lógica borrosa, Ph. D. Thesis, 1997.

[8] Muñoz, V., Ollero A., Prados M. and Simon A., Mobile Robot Trajectory Planning with Dynamic and Kinematic Constraints. *Proc. IEEE Int. Conf. Robotics and Automation*, San Diego (CA), May 1994, pp. 2802-2807.

[9] Nomura, H., Hayashi, I., and Wakami, N., A self-tuning method of Fuzzy Control by Descent Method, *Proc. of IFSA'91*, 1991, pp. 155-158.

[10] Ollero, A., A. Simon, F. García and V. Torres, Integrated Mechanical Design and Modelling of a New Mobile Robot, *Preprints of the SICICA'92 IFAC Symposium on Intelligent Components and Instruments for Control Applications*, Malaga (Spain), May. 1992, pp.557-562.

[11] Palm, R. Driankov. D. and Hellendoorn, H., *Model Based Fuzzy Control*, Springer, 1996.

[12] Quinlan, J. R. et al., Discovering rules by induction from large collections of examples. *Expert Systems in the microelectronic age* Michie,D. (Ed.). Edinburgh Univ. Press, 1979.

[13] Ruiz-Gómez J., García-Cerezo, A. and Fernández-Ramos, R., Curvature Fuzzy Modelling for a Mobile Robot, Ollero, A. (Ed.): *Intelligent Components for Vehicles*. Pergamon Press. 1998.

[14] Ruiz-Gomez, J. and A. Garcia-Cerezo, Inductive Learning Fuzzy Modelling. *Proc. INCON´97 Intern. Workshop on Intelligent Control*. Sofia (Bulgaria), 1997, pp. 23-28.

[15] Ruiz-Gomez, J. and Garcia-Cerezo, A., An Inductive Inference Rule Generation Tool for Fuzzy Modelling, *Proc. IDEA'97, Intelligent Design in Engineering Application Symp.*, Aachen (Germany), 1997, pp. 113-117.

[16] Sison, L. S. and Chong, E., Fuzzy modeling by induction and pruning of decision trees, *Proc. IEEE Int. Symp. Intelligent Control*. Columbus. (OH.) 1994, pp. 166-171.

[17] Takagi, T. and M. Sugeno, Fuzzy Identification of Systems and Its Applications to Modeling and Control. *IEEE Trans. Syst., Man, Cybernetics*, 15, 1985, pp. 116-132.

[18] Tani, T. *et al.*, Fuzzy Modelling by ID3 algorithm and its application to prediction of a heather outlet temperature, *Proc. IEEE Int. Conf. on Fuzzy Systems*, San Diego (CA), 1992, pp. 1719-1723.

[19] Wang, L-X., Fuzzy systems are universal approximators, *Proc. Int. Conf. Fuzzy Systems*, San Diego, 1992, pp. 1163-1170.