Proc. of the Fourth IEEE Int. Conf. on Fuzzy Systems, pp. 83 - 86, 20 - 24.03.1995, Yokohama, Japan.
The Robot MORIA was distinguished with the Intelligence Award in the Robot Competition.

A Fuzzy System for Indoor Mobile Robot Navigation

Hartmut Surmann, Jörg Huser and Liliane Peters

German National Research Center for Computer Science, 53757 St. Augustin, Germany
Tel: + 49 2241 14-2518, E-mail : surmann@gmd.de

Abstract — **An autonomous mobile robot (AMR) has to cope with uncertain, incomplete or approximate information. Moreover it has to identify sudden perceptual situations to manoeuvre in real time. This paper describes a fuzzy rule based system (FRBS) approach controlling the movement of an autonomous mobile robot (MORIA). Difficult guiding and controlling properties of the robot are achieved by combining local actions and global strategies within the fuzzy controller. Different behaviors and perceptions are detected with the help of fuzzy rules and stored in fuzzy state variables (FSV). These state variables activate different fuzzy rule sets which in turn change the behavior of the fuzzy controller.**

I. Introduction

It has already been shown that fuzzy rule based systems (FRBS's) [1] are important tools for modelling complex problems. The knowledge base is acquired from human experts or from a referential data set usually optimized by means of neural or genetic algorithms [2]. These concepts have been successfully applied in nearly all fields of control applications [3, 4], including autonomous robot control [5 - 10]. There exist distinct fuzzy control methodologies for autonomous robot control because of the large variety of existing sensors and the various methods to recognize different perceptual situations. Therefore in our approach we used a fuzzy rule based controller (FRBC) of an autonomous mobile robot (MORIA).

An autonomous robot is a dynamic system of higher order i.e. the output mapping is not only dependent on the current input but also on the previous inputs. Such systems are more difficult to approximate and to control than first-order-processes. To realize a short term memory we introduced in our fuzzy control system internal state variables. Internal variables have to be defined for the input and output sections of a fuzzy controller [11]. Different behaviors and perceptions are identified with the help of fuzzy rules and stored in these fuzzy state variables (FSV). By this means the system behavior is adapted to the dynamic changing environment. Furthermore, difficult guiding and controlling properties of the robot are achieved by the combination of local actions, e.g. avoiding obstacles, and linguistic instructions, e.g. turning left at the next junction.

This paper is structured as follows. In the next section we present the new algorithm which is based on a first order FRBS. In the third section we describe the control architecture of the robot MORIA. The robot behavior on an example situation is presented in section four. The last section summarizes with concluding remarks.

II. Basic Terms of Fuzzy Rule Based Systems

As shown by Castro [12] FRBSs are universal approximators , i.e.:

For each $a, b \in \Re$ with $a<b$ let $\mu_{a,b}: \Re \rightarrow \Re$ be a membership function such that $\mu_{a,b}(x) \neq 0$ *if* $x \in [a,b]$. Moreover let T_1 and T_2 be t-norms and I a fuzzy implication verifying $I(a,0) = 0$ *if* $a \neq 0$ (for example a R-implication or a t-norm implication $I(a,b)=T_3(a,b)$). Let G be a t-conorm and S a defuzzification method.

The 6-tupel $FRBS=(T_1,T_2, I, G, S, \mu_{a,b})$ is a family of FRBS's with:

1) A fuzzy rule base composed by a finite number k of rules of the form

$$R_j: IF\ x_1\ is\ A_{1j}\ and\ ...\ x_n\ is\ A_{nj}\ THEN\ y\ is\ B_j \qquad j=1..k$$

2) The membership function of each A_{ij} is of the form $\mu_{a_{ij}^1,\, a_{ij}^2}(x)$ for $a_{ij}^1 < a_{ij}^2,\ a_{ij}^1, a_{ij}^2 \in \Re$, i.e.

$$\mathrm{A_{ij}}(x) = \mu_{a_{ij},b_{ij}}(x)$$

3) The membership function of each B_j is also of the form $\mu_{a,b}$ for some $a<b,\ a,b \in \Re$, i.e.

$$\mathrm{B_j}(x) = \mu_{a_j,b_j}(x)$$

4) T_1 is the fuzzy conjunction operation. The generalised modus ponens is constructed with the other t-norm T_2 and the implication I:

a) A rule R_j: *IF* x_1 *is* A_{1j} *and ...* x_n *is* A_{nj} *THEN* y *is* B_j will be applied only if the n-dimensional input vector $\underline{x}$ matches with the antecedent, i.e. if $A_j(\underline{x}) \neq 0$, being $A_j(\underline{x}) = T_1(A_{1j}(x_1), ..., A_{nj}(x_n))$.

b) If the input vector $\underline{x}$ matches with the antecedent then the inference is

IF x_1 is A_1 and ... x_n is A_n THEN y is B
x is A'
is B'
with $B'(y) = Sup\{T_2(A'(x), I(A(\underline{x}),B(y))) \mid \underline{x} \in \Re^n\}$ and $A(\underline{x}) = T_1(A_1(x_n),...A_n(x_n))$

In control applications the input $\underline{x} = \underline{x}^0$ is a point, so

$$A'(\underline{x}) = \begin{cases} 1, & if \ \underline{x}^0 = \underline{x} \\ 0, & otherwise \end{cases}$$

and $B'(y) = T_2(1, I(A(\underline{x}),B(y)))=I(A(\underline{x}),B(y))$.

c) In general, for the input $\underline{x} = \underline{x}^0$ the inference algorithm of the rule R_j is expressed by:

$$B_j'(y) = \begin{cases} 0, & if\, A(\underline{x}_j^0) = 0 \\ I(A_j, B_j(y)), & otherwise \end{cases}$$

which is in the case of a t-norm implication

$$B'(y) = I(A_j(\underline{x}^0),B_j(y)).$$

5) The composition of all fuzzy rules is made by the t-conorm G:

$$B'(y)=G(\{B_j'(y)\}), j=1..k.$$

6) The defuzzification method S is the center of area method:

$$y^0 = S(x^0) = \frac{\int y\, B'(y)\, \mathrm{dy}}{\int B'(y)\ \mathrm{dy}}$$

The important parameters for the FRBS are the number of fuzzy rules k and the positions and widths of the input and output membership functions expressed by a_{ij}^1, a_{ij}^2, b_j^1, b_j^2.

THEOREM. Let $f: U \subseteq \Re^n \rightarrow \Re$ be a continuous function defined on a compact U. For each $\varepsilon > 0$ there exists a $FRB_\varepsilon \in FRBS$ such that

$$sup\{\mid f(\underline{x})\text{-}FRB_\varepsilon(x) \mid \ \mid \underline{x} \in U\} \leq \varepsilon .$$

Proof in [14].

The Mamdani fuzzy controller [13] expressed by $FRB_{Mam} = (MIN, MIN, MIN, MAX, COG, \mu_{a,b})$ is a universal approximator. In this paper we use a variant of the Mamdani controller which is often applied in the design system from Togai InfraLogic. The FRBS uses triangular and gaussian membership functions which are either linear approximated by 2 or 6 straight lines or directly stored in look-up tables. For a fast defuzzification, the centroids M_i and areas A_i of the output membership functions are calculated apriori instead of during the evaluation process.

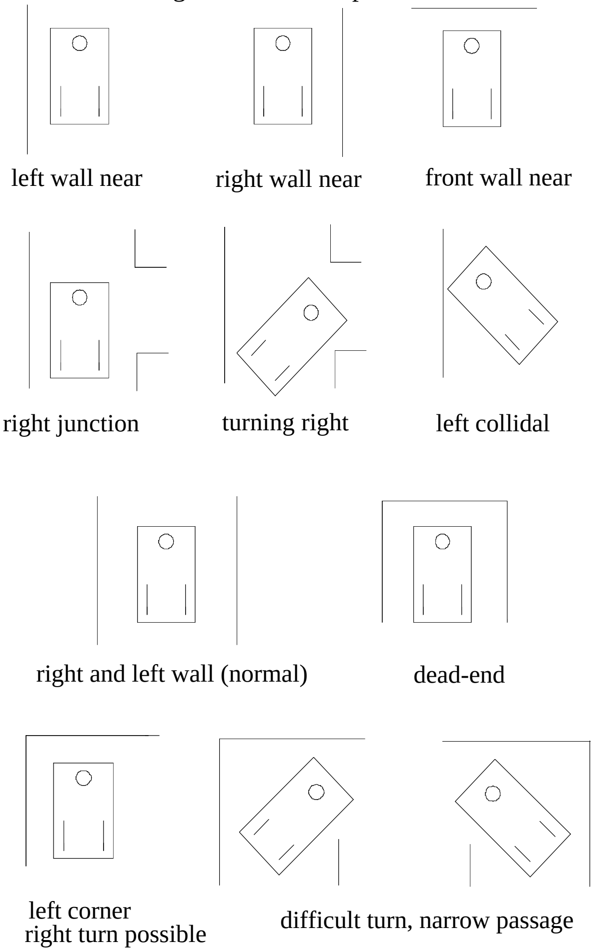

Figure 1: Some examples of perceptual situations

The processing of multiple activated output membership function and the output composition is achieved by scaling the centroids M_i and A_i with the truth value and addition:

$$y^0 = \frac{\sum y * M_i}{\sum y * A_i}, \text{ i=1..\#OMF}$$

Since FRBS's are universal approximators and define an n-dimensional function we regard the design of a FRBS as a user friendly, easy to model technique. Especially the realization of the function, which is defined by the FRBS, can be independent from the rule base structure. The fuzzy controller realizes an n-dimensional non-linear function. The I/O behavior of the first order FRBS depends only on the current input vector because the first order fuzzy algorithms have no storage or delay elements.

Now, for a dynamic system such as the autonomous robot, the mapping of the output - speed and

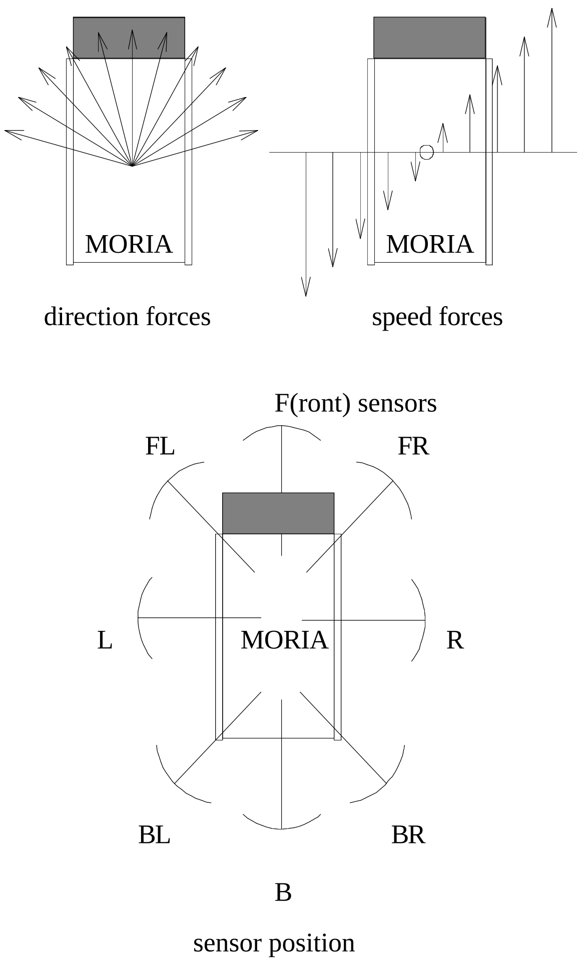

Figure 3: Sensor position and possible driving commands for the MORIA robot

steering angle - depends not only on the current input - sonar values -, but also on different perceptual situations (Figure 1) and user or planner commands (Table 1).

User / Planers Command	Fuzzy number
Stop	0 ∈ [-0.5,1.5]
Straight ahead	1 ∈ [-1.5,2.5]
Take next left	2 ∈ [-1.5,2.5]
Take next right	3 ∈ [-3.5,3.5]
Go backwards	4 ∈ [-3.5,4.5]
Turn left (≈ 90^{o})	5 ∈ [-4.5,5.5]
Turn right (≈ 90^{o})	6 ∈ [-5.5,6.5]

Table 1: Example for different high level (global) commands.

The high level commands as well as the fuzzy state variables corresponding to different perception situations (Figure 1) are given to the fuzzy navigator via fuzzy numbers. For example, the command "straight ahead" is realized as a gaussian function $\mu_{a,b}(x)$ with a = 0,5 and b = 1,5.

III. The Architecture of the Robot MORIA

The autonomous platform, "MORIA" (Figure 2) is a mobile vehicle, driven by two motors one for forward / backward movement and the other for turning. The vehicle has a length of 175 cm (including the bumper 45 cm), a width of 73 cm and a height of 60 cm. It can carry and move a payload of 100 kg by a natural weight of 400 kg. For the strategy we present only 8 sonar sensors devices were used (Figure 3). Computational capabilities of MORIA are based on an industrial PC (486/33 MHz, 4 MBytes) with extended I/O possibilities. The PC board collects the output of the sonar sensors and the drives the two motors. A communication link to other mobile platforms or remote mainframe are possible through the on board installed infrared sensor.

The basic architecture of the autonomous platform is shown in figure 4. Based on a reduced topological map a high level planner or a human operator selects a list of linguistic driving commands e.g. straight, next left, straight, next right, etc. (see also Table 1).

The actual command is forwarded to the fuzzy controller (navigator) and represent the global driving direction. The response of the fuzzy navigator is dependent on the incoming sensor values and the input state variables and the global driving direction. The actual fuzzy input state variable reflects the latest recognized environment and allows a switch between different navigation strategies. The output of the fuzzy controller (navigator) gives the new driving speed and steering angle for the robot and estimates the current state variable. On its turn this FSV is

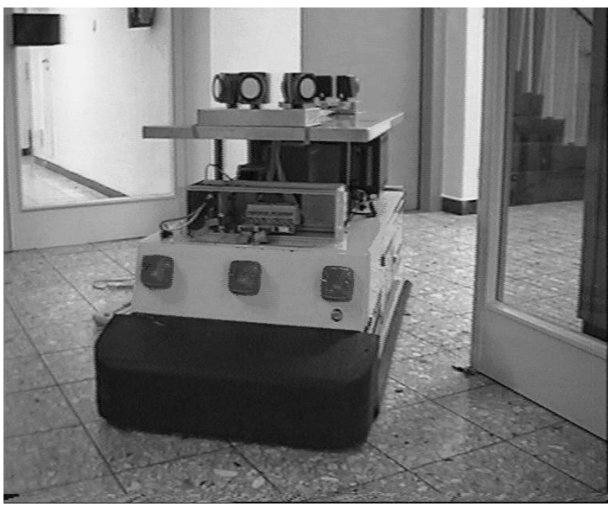

Figure 2: The mobile robot MORIA

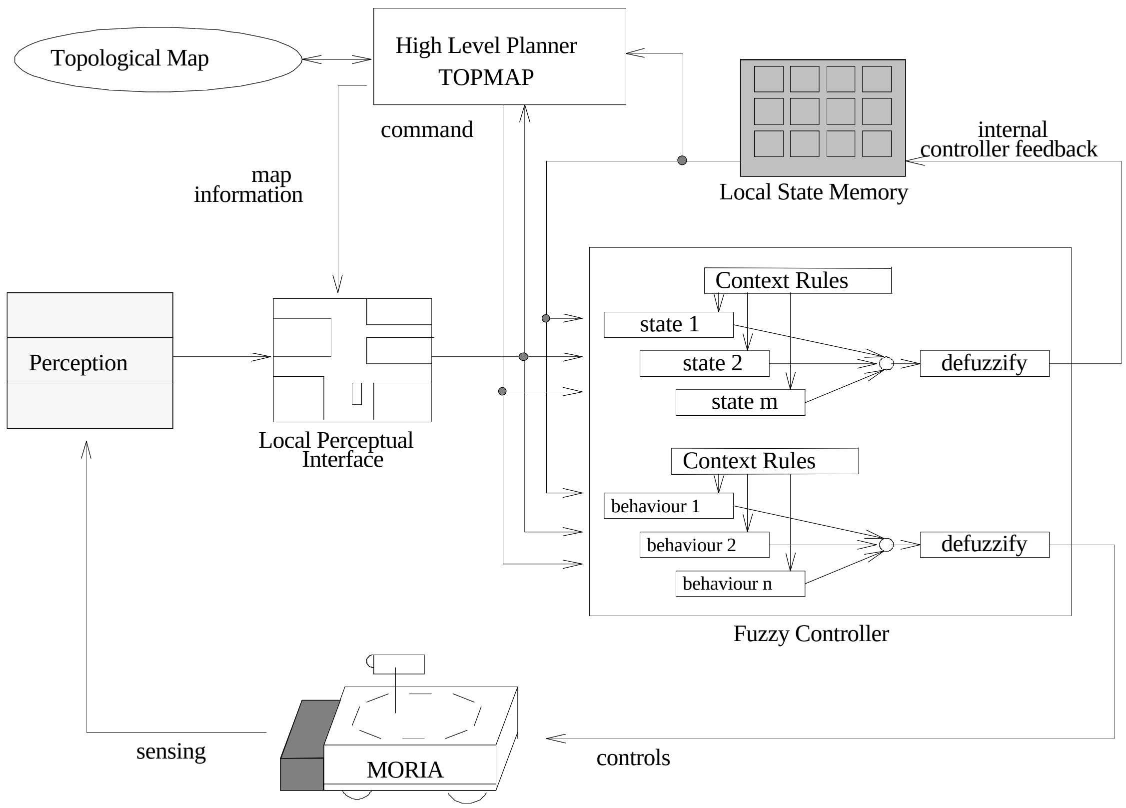

Figure 4: Architecture of the autonomous platform

looped back to the input of the fuzzy navigator (controller) and simultaneously given back to the planner as control values. The planner creates or updates the topological map and checks the route in accordance with the given global command. The planner can give exploring commands, where the goal is to detect environment structure or task commands.

The heart of the system is a set of context dependent fuzzy rules. Two distinct rule blocks are defined. One for the driving of the robot motors (velocity, direction, angle) and the other for the recognition of the perceptual environment which gives the actual FSV (Figure 1). Map informations, like the actual corridor width or length are used for the local perceptual interface. To give a better understanding of the MORIA behavior we present in the next section a perception and reaction.

IV. Example

Let's suppose the robot is in exploring mode and has the global direction command "take a right when ever you can". The detection of a dead-end corridor (Figure 6a) is absolutely necessary in order to change the exploring strategy. We present an example where the robot has to detect first the dead-end and then change the exploring command to " take a left when ever you can". The detection of the environment structure and the driving commands are explained in detail in the next paragraph. Figure 6b shows a snapshot of the control window during the robot exploration phase. The behavior of the robot in this situation is essentially determined by three fuzzy rules. At point a) the sonar values of the 3 front sensors (front left FL, front F, front right FR) give a fuzzy value of small (see Figure 5). The following rule fires:

RULE 1: IF (FSV_2 [direction] is forward) and (FL is small) and (F is small) and (FR is small) THEN (velocity is ositive small) and (steering angle is small left).

Thus the velocity of the robot is reduced and hence it tends to move to the left direction. This is because of its normal strategy to follow the right wall. Next step is the detection of the dead-end

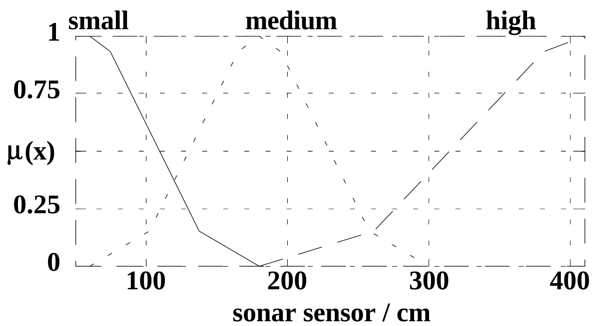

Figure 5: Membership functions for the sonar sensors

(point b). For the given situation the following rule will fire:

RULE 2: IF (FL is small) and (F is small) and (FR is small) THEN (FSV_1[environment] *is* dead-end corridor*) and (*FSV_2[direction] *is backward).*

As soon as the dead-end corridor structure is detected the FSV_2 is changed and with it the exploring strategy. The transition from one state to another is rather smooth because of the overlapping membership functions and the selected defuzzification method (section 2). Note that although the state variables are fuzzy in nature, it is also possible to incorporate binary variables.

In the next step the rear sensors (rear left BL, rear B and rear right BR) become active and the following rule is firing:

RULE 3: IF (FSV_2*[direction] is backward) and (BL is not small) and (B is not small) and (BL is not small) THEN (velocity is negative medium) and (steering angle is zero).*

If a turn forward is possible the robot will change its driving direction to forward (FSV_2) and reset all the state variables (Figure 6d). In the presented example the exploring strategy will change to "take a left when possible".

In this situation, one real life problem can occur. The robot goes into a deadlock situation, if the fuzzy state variables are not properly reset. The robot cradles to and fro. It is the task of the planner to recognize this deadlock situation reset the fuzzy state variable and give a new command.

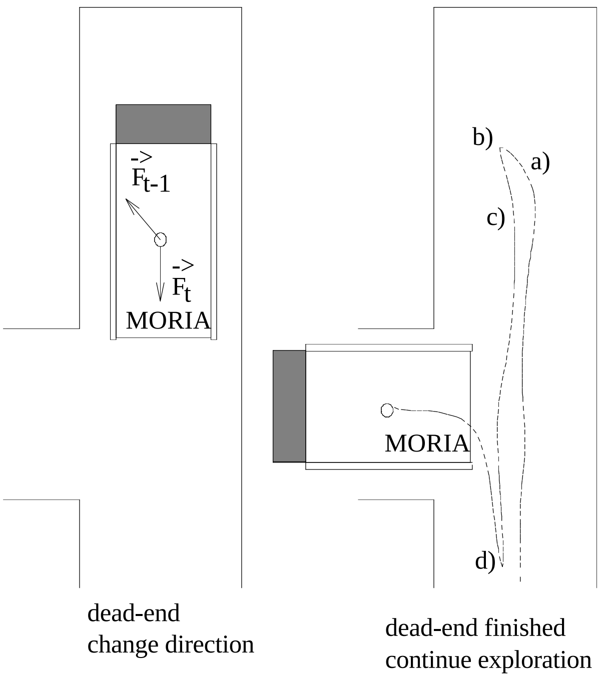

Figure 6: Snapshot of the control window during the detection of a dead-end corridor

The basic reactivity of the robot is achieved by only 27 fuzzy rules (3 membership functions for each sensor). The whole implemented fuzzy control system has about 180 fuzzy rules with 30 inputs and 11 outputs.

V. CONCLUSION

As presented in this paper an autonomous navigation system for unstructured real world environment can achieve real-time reactivity through the implementation of a goal oriented behavior and the detection of various perceptual situations. The presented strategy showed that through the implementation of a FRBS controller a flexible behavior can be achieved and the inherent existing imprecision in knowledge and execution can be tolerated.

Consequently several conclusions may be drawn from these investigations. The adopted fuzzy techniques provide a flexible behavior strategy. With help of the additional introduced fuzzy state variables different perceptual situations are identified and a local memory map is created. through this approach local reativities (e.g. obstacle avoidance) are smoothly blended together with the high level instructions (e.g. turn left at the next junction) to create a real-time exploring and executing robot system.

An additional advantage of the strategy used is the possibility to build a topological map of the environment and update it. The memory space needed for such a map is reduced compared to the use of an exact geometric description of the environment. The proposed architecture was implemented on an indoor autonomous vehicle, MORIA. The short modelling time was possible by employing our modelling system FUNNYLAB [14].

Future work will concentrate on the automatic learning and improvement of fuzzy rules as well as membership functions tuning.

Acknowledgements Our special thanks to Klaus Beck, Shuwei Guo and Helmut Podworni, who contributed to the software and hardware development of MORIA. And last but not least our special thanks to TZN Forschung- und Entwicklungszentrum Unterlüß GmbH for the given possibility to test our approach on the their autonomous vehicle MORIA.

REFERENCES

[1] M. Jamshidi et. al Ed. Fuzzy Logic and Control: Software and Hardware Applications, Prentice Hall, Englewood Cliffs, New Jersey, 1993.

[2] H. Surmann, A. Kanstein, and K. Goser. Self-Organizing and Genetic Agorithms for an Automatic Design of Fuzzy Control and Decision systems. In Proceedings of the First European Congress on Fuzzy and intelligent Technologies, EUFIT'93, Aachen, pages 1097 - 1104, 7-10 Sept.1993.

[3] M. Sugeno Ed.. Industrial applications of fuzzy control. North-Holland, 1985.
[4] H. J. Zimmermann. Fuzzy set theory and its applications, 2nd edition. Kluwer Academic Publishers, 1990.
[5] Alessandro Saffiotti, Enrique H. Ruspini, and Kurt Konolige. A Multivalued Logic Approach to Integrating Planning and Control, Artificial Intelligence Center, SRI International, Technical Note No. 533, Menlo Park,CA 94025, USA, 4/1994.
[6] Changman Son and George Vachtsevanos. A Fuzzy Expert Organizer / Robust Control Strategy for a Moving Base Robotic Part Assembly. In IEEE International Conference on Robotics and Automation, San Diego/California, pages 3290 - 3295, 8.5. - 13.5.1994.
[7] H.R. Beom and H.S. Cho. A Sensor-based Obstacle Avoidance Controller for a Mobile Robot Using Fuzzy Logic and Neural Networks. In IEEE international conference on intelligent robots and systems, pages 1470 - 1475, 6.1992.
[8] Edward Tunstel and Mo Jamshidi. Fuzzy Logic and Behavior Control Strategy for Autonomous Mobile Robot Mapping. In FUZZ-IEEE World Congress on Computional Intelligence, Orlando, pages 514 - 517, 26.6. - 2.7.1994.
[9] Yoichiro Maeda, Minoru Tanabe, Morikazu Zuta, and Tomohiro Takagi. Hierarchical Control for Autonomous Mobile Robots with Behavior-Decision Fuzzy Algorithm. In Proceddings of the IEEE International Conference on Robotics and Automation, Nice/France, pages 117 - 122, 12.5. - 14.5.1992.
[10] Patrick Reignier. Fuzzy logic techniques for mobile robot obstacle avoidance. Robotics and Autonomous Systems, 12:143 - 153, 1994.
[11] Vittorio Gorrini and Hugues Bersini. Recurrent Fuzzy Systems. In FUZZ-IEEE World Congress on Computional Intelligence, Orlando, pages 193 - 198, 26.6. - 2.7.1994.
[14] J. Castro; Fuzzy logic controllers are universal approximators. University of Grenada, Department of Computer Science and Artificial Intelligence, Technical Report #DECSAI-93101, 6/1993.
[13] E.H. Mamdani. Application of fuzzy algorithms for control of simple dynamic plant. Proc. IEE, Vol. 121, No. 12, Dec. 1974.
[14] Hartmut Surmann, Ansgar P. Ungering, Thorsten Kettner, and Karl Goser. What kind of hardware is necessary for a fuzzy rule based system. In FUZZ-IEEE World Congress on Computional Intelligence, Orlando, pages 274 - 278, 26.6. - 2.7.1994.