# A mixed analogue/digital fuzzy system for indoor mobile robot navigation

**Trevor Lund†, Hartmut Surmann, Jörg Huser and Liliane Peters**

† Faculty of Information Sciences and Engineering,
University of Canberra,
PO Box 1, Belconnen ACT, 2616 Australia
Tel: +61 6(06)201-2379, Fax¿ -5231
E-mail: tlund@ise.canberra.edu.au

German National Research Center for Information Technology,
53754 St. Augustin, Germany,
Tel: 0049/2241/14-2518, Fax: -2342
E-mail: surmann@gmd.de

*Abstract* – **Analogue techniques for the implementation of fuzzy control seem to be taken in competition with digital techniques. Since both methods have their own strengths, an attempt has been made to marry the two to produce a navigation system for an autonomous mobile robot. The system developed takes advantage of the speed and compactness of analogue fuzzy controllers to implement some of the tasks, especially where a fast quasi-continuous response is required. The versatility, adaptability and memory capability of a general purpose microcontroller are then used to perform the tasks for which the analogue controllers available are not suitable. The partitioning of the tasks, and the structuring of the rule base are discussed.**

## 1 Introduction

An autonomous mobile robot (AMR) has to cope with uncertain, incomplete or approximate information when going about its tasks as a transportation or cleaning robot in administrative buildings, factories or hospitals. Moreover, it has to identify sudden changes that it perceives in its surroundings, and react and maneuver in real time [1]. A control system utilising fuzzy rules has recently been developed in this laboratory, and has been demonstrated to be able to guide a real AMR through a network of corridors of varying widths, containing occasional random obstructions such as fire extinguishers [2] (Robot Intelligence Award).

It is our desire to make use of some of the new concepts proved in that system, but produce a smaller, faster system, which will possibly run on a small embedded microcontroller, leaving more processing power in the central CPU of the robot available for other tasks, such as the reading of labels on doors or packages.

A purely analogue-based fuzzy control system for an AMR has also been developed in this laboratory [3, 4, 5], which uses just two fuzzy inference chips and some opamps. This controller can successfully guide the robot through a simple network of corridors, avoiding collisions with walls or other obstacles, responding to simple commands from the user such as “go straight ahead” or “take next left”, but has not the path-finding sophistication of the purely digitally implemented system.

To combine the analogue and digital approaches, to make use of the compact size and fast response of the analogue system whilst using a microcontroller to add on a more varied, versatile and tunable response, is the objective of the work described in this paper.

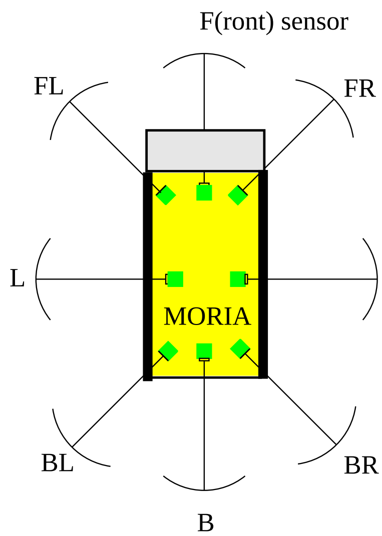


Figure 1: Sensor position of the MORIA robot

The vehicle used in our experiments, MORIA, is 175cm x 90cm x 75cm in size, and has a weight of 400kg with a payload of 150kg. It is equipped with 8 ultrasonic sensors which supply the distance information required for navigation. The orientation and naming of these sensors is shown in figure 1. The vehicle is driven by two motors acting on a single wheel, one motor (reversible) to provide the driving torque, and the other motor to steer the vehicle by changing the orientation of the driving wheel. The external appearance of the robot is shown in figure 2.

Figure 2: The autonomous vehicle MORIA

The control system required for this AMR is expected to respond to high level, or "goal-oriented" commands which will be issued by a route planning program, or by a user, and steer the AMR in such a way as to achieve the goal smoothly without colliding with any features (*eg* walls, packages, fire extinguishers, other robots or humans) in the surroundings. The set of commands adopted is shown in Table 1, together with the number associated with each command in the fuzzy rule base.

| Planner/User Command | Fuzzy number |
|---|---|
| Stop (s) | $0 \in [-0.5, 0.5]$ |
| Straight Ahead (a) | $1 \in [0.5, 1.5]$ |
| Take Next Left (l) | $2 \in [1.5, 2.5]$ |
| Take Next Right (r) | $3 \in [2.5, 3.5]$ |
| Go Backwards (b) | $4 \in [3.5, 4.5]$ |
| Turn Left Immediately (li) | $5 \in [4.5, 5.5]$ |
| Turn Right Immediately (ri) | $6 \in [5.5, 6.5]$ |
| Go Forwards (f) | $7 \in [6.5, 7.5]$ |
| Change Direction (cd) | $8 \in [7.5, 8.5]$ |

Table 1: The high-level, or goal-oriented commands supplied by the "Planner" routine or the user to the fuzzy controller. (Commands 4-7 are "manual" commands, where the response contains no automatic collision avoidance strategy.)

## 2 Overview of Approach

Various analogue fuzzy chips have been designed at GMD, and manufactured through EUROCHIP, and some of these have been used in experiments to provide a collision avoiding guidance system for the robot MORIA, with some success. For these experiments, chips with a fixed rule-base were used, and differing behaviour patterns (*eg* go straight ahead, follow left wall) for the guidance system were obtained by summing the outputs of two or more chips with differing weightings according to the behaviour required. An operator could cause the switching from one behaviour pattern to another whilst the robot was in motion, and thus be the source of the "goal-oriented" commands, whilst the analogue chips took care of the low level navigation and collision avoidance strategy.

In order to construct a system with the capability of following more varied and more finely tuned behaviour patterns, it was decided to make use of a chip with a dynamically re-configurable rule-base. One such chip available has 3 inputs, one output, and 13 rules. Each input has 3 bell-shaped membership functions as shown in figure 3(a), and each rule can select one of five singletons, as shown in figure 3(b), as its output. Which consequent is selected for a particular rule is determined by the magnitude of the voltage applied to the appropriate pin on the chip. Of the 27 possible rules for such a fuzzy system with 3 inputs each with 3 membership functions, the location of the 13 used in this chip is shown in figure 4, where the entries correspond to the behaviour pattern "turning right".

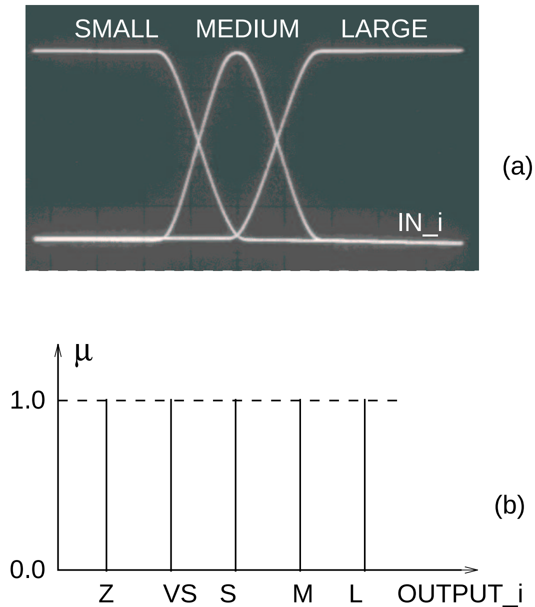


Figure 3: Fuzzy set definitions of (a) the input and (b) the output variables for the analogue controller.

The lack of a capability to implement rules for the blank squares of figure 4 imposes no limitation on this system, since, with the overlapping bell-shaped functions for the inputs, there will normally be two or more rules firing, whilst in the worst case, on some cell boundaries, a solitary rule will fire. Indeed, the availability of only 13 rules per chip has simplified the rule generation process. Two of these chips are needed for the robot MORIA - one to control the speed, and one to control the steering angle. Since all of the signals are in the analogue domain, no A/D or D/A converters are needed between the sensors and the analogue fuzzy controllers or between the controllers and the motor driver circuit.

(a)

F: small

| FL \ FR | s | m | l |
|---|---|---|---|
| s | Z | | VS |
| m | | | |
| l | VS | | S |

F: medium

| FL \ FR | s | m | l |
|---|---|---|---|
| s | | S | |
| m | S | M | M |
| l | | M | |

F: large

| FL \ FR | s | m | l |
|---|---|---|---|
| s | M | | L |
| m | | | |
| l | L | | L |

(b)

F: small

| FL \ FR | s | m | l |
|---|---|---|---|
| s | Z | | PL |
| m | | | |
| l | NL | | PL |

F: medium

| FL \ FR | s | m | l |
|---|---|---|---|
| s | | PS | |
| m | NS | Z | PS |
| l | | Z | |

F: large

| FL \ FR | s | m | l |
|---|---|---|---|
| s | Z | | PS |
| m | | | |
| l | NS | | PS |

Figure 4: Fuzzy rule set for “turning right” behaviour pattern (a) speed (b) steering angle

To obtain the degree of sophistication of the navigation system demonstrated by Surmann et. al. [2], some level of memory is necessary, and this can be provided by a digitally based controller which can affect the behaviour pattern of the analogue fuzzy controllers by altering the rule base of those controllers “on the fly”. The purpose of this digitally based controller is then to

i.) identify the nature of the surroundings
ii.) remember the current behaviour pattern
iii.) recognise the current goal-oriented command
iv.) combine the above knowledge to select the next behaviour pattern.

Having been relieved of the detail of the motion control, which will be operating in the analogue chips, providing quasi-continuous collision avoidance, the time constraints on the processing of these rules will be less severe than in a purely digital system. Adding this fact to the likelihood that the rule set to be processed by the digital controller should be smaller than previously (again because the motion control is taken care of by the analogue chips), the possibility of using a small embedded microcontroller to host the fuzzy inference engine arises.

The partitioning of the control tasks between the analogue and the digital parts of the control system, and the structuring of the framework for the decision making within the digital controller are then central concerns of this work. As a starting point the rules can be grouped into sets and arranged in a hierarchy as shown in figure 5.

## 3 Strategy

Since each analogue fuzzy controller chip has only three inputs, it is appropriate to implement the motion controlling rules (fig. 5) (the part of the rule base which will reside on the analogue chips) using only the three forward facing sensors as inputs. The widths of the membership functions can be scaled (in hardware) to suit the problem at hand *ie* the dynamics of the robot and the dimensions of the environment in which it must operate.

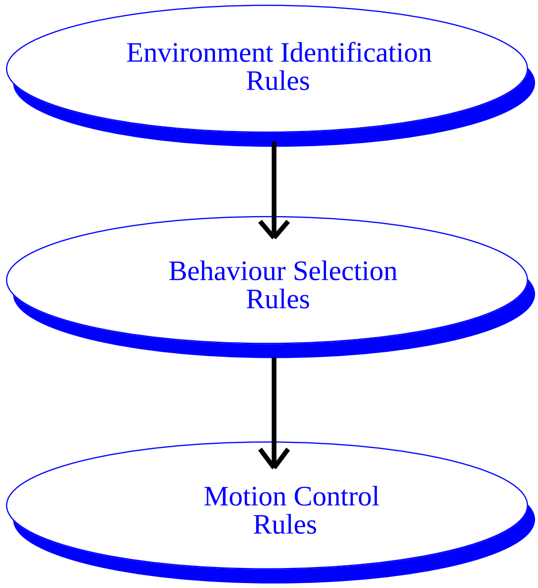


Figure 5: Hierarchy of the rule-sets within the total rule base

It then remains to devise sets of rules which will determine the possible behaviour patterns of the robot, and which can be switched into operation by the digital controller, controlling the set of voltages applied to the consequent determining pins of the analogue chip. Though not all of the rules will need to be changed to switch from one behaviour pattern to another, it is useful in the design process to think in terms of complete sets of rules, so that any one time the robot is operating with a complete self-consistent set of rules, which will produce a particular behaviour, such as turning a corner, rather than in terms of switching the rules one at a time in response to some perceived change in operating conditions.

Thus the rules controlling the motion of the robot can be compartmentalised, and each behaviour pattern can be independently tuned and optimised, *eg* any change to the set of rules controlling the robot's motion through a right turn will have no effect on the rules which allow the robot to steer a course along a corridor, avoiding obstacles and *vice versa*. Designing for complete sets of motion rules produces insignificant extra overhead in both hardware and software, and ensures maximum flexibility for future modifications.

Hence, rules must be developed in sets of 13 at a time, to suit the configuration of the analogue chip. Furthermore, since only the three forward facing sensors can be connected to the analogue controllers, it is necessary to implement a switching mechanism, similar to that used in the purely digitally implemented forerunner of this system [2], to connect the backward facing sensors in place of the forward facing sensors whenever the robot is commanded to go into reverse. In this fashion, the rules developed for forward motion will be equally useful for reversing, with no need to generate further modifications.

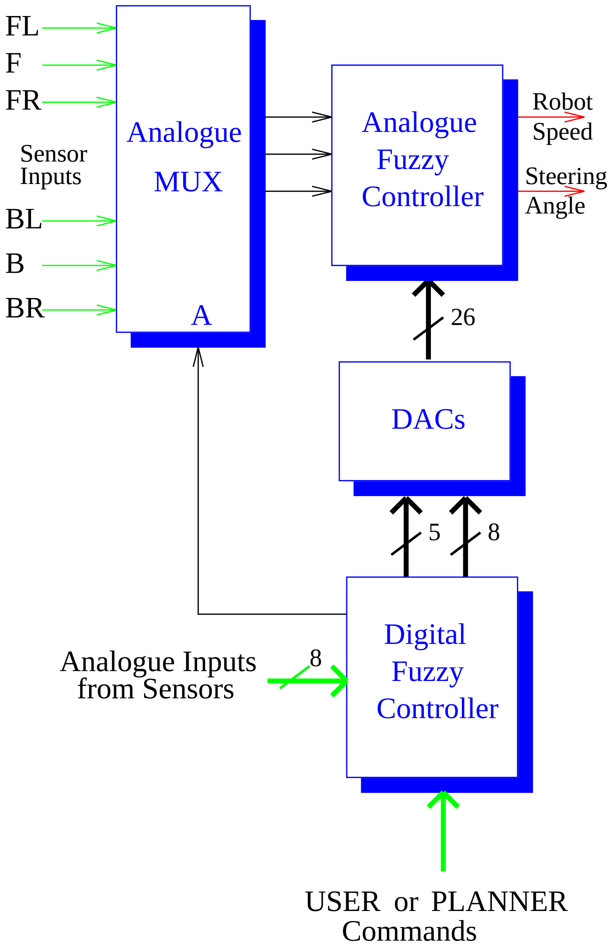


Figure 6: Target hardware configuration

A complete fuzzy inference using the analogue fuzzy controller takes less than a microsecond, so the update rate for the motion control system is essentially determined by the update rate for the information from the sensors.

The digital controller, on the other hand, can, and indeed needs to, make use of the information from all 8 sensors. It is not hard to provide the hardware capability for monitoring the 8 sensors, since a variety of microcontrollers come with an 8-channel A/D converter built in. However, when the only information on the robot's surroundings comes from these sensors, the identification of the nature of these surroundings (*eg* whether there is a branch off the corridor to the right or not) is complicated by the motion of the robot itself. For instance, if the robot is not aligned parallel to a corridor wall, then the relative values read from the sensors will have a different pattern from when the robot is aligned, or if the robot is turning for some reason (such as to avoid an obstacle) then changes in the readings from an individual sensor will not have the same meaning as when the robot is proceeding in a straight line.

To allow for these effects, the fuzzy inference engine in the digital controller needs more inputs than just the sensor inputs; it needs also fuzzy variables representing the current state of the controller and the robot, such as whether a branch has been detected, whether the robot is in the process making a turn and so on. By the same token, the fuzzy controller needs more outputs than just the identification of the set of rules to be used by the analogue controller; it needs to output variables to signal other outcomes of its inference process (*eg* has a turn been completed, has a branch been passed without being taken *etc*) for use by the higher level route planner, or for use by itself as inputs in its next inference round. This strategy has previously been shown to be effective [2]. The above considerations lead to the hardware configuration shown in figure 6.

# 4 Development and simulation environment

The development and simulation environment used for this project was basically that developed for the purely digitally implemented fuzzy control system reported in [2]. The modeling system FunnyLab [6] was used for the creation and editing of the membership functions and rule base. The knowledge base file produced by FunnyLab was read into a simulation environment developed in-house at GMD, which includes not only a fuzzy inference engine, but also a motion simulator which incorporates a model of the real, measured dynamics of the robot MORIA, and a graphical user interface which shows a plan view of the robot's progress through the test environment, displays the values of certain key parameters, and allows the instantaneous entry of the planner/user commands given in table 1.

The motion simulator has been demonstrated to produce an accurate representation of the dynamics of MORIA during the earlier development work, and allows a realistic assessment of the effects of the fuzzy controller, given that it includes the effects of the inertia and finite response time of the robot.

The user interface includes a feature which allows single-stepping. At each step it is possible to ascertain which rules are firing, what distance values are being returned by the sensors, and the values of some of the important fuzzy variables. This feature is a powerful tool for examining the operation of the rule base over critical stretches of travel, and facilitates the rapid identification of which rules are serving the purpose for which they were intended, which are firing in unexpected circumstances, and which need tuning to produce smoother, more reliable behaviour.

It was possible to include the simulation of the effect of the analogue part of the system in the existing simulator because the update rate of the commands from the analogue controller is limited by the update rate of the sensor information, and the simulator is effectively already clocked according to this update rate.

# 5 Results

## 5.1 Motion Rules

Five complete sets of motion rules (rules which will be implemented by the analogue fuzzy controllers in the target system) have been developed, and have been found to be sufficient to provide the robot with the ability to:

i.) go straight ahead, avoiding branches
ii.) take next left branch
iii.) take next right branch,

all with varying corridor widths, and the capability of avoiding obstacles within the corridors. These rule sets have been given the names:

- proceeding - the default set, for going straight ahead
- turning left - for performing a turn into a branch on the left
- turning right - for performing a turn into a branch on the right
- avoid left branch to maintain a straight course
- avoid right branch past an unwanted branch,

and they are associated with the "states" of the same name shown in the state diagram of figure 7 (see next paragraph), such that each time the digital controller decides to enter a particular state it switches in the rule set of that name to the analogue controllers. In particular, by partitioning the rules into sets in this fashion, it has been found possible to tune the sets of "turning" rules so that only a single set for each direction is necessary, however tight the turn is, despite expectations to the contrary [2].

## 5.2 Behaviour Selection

The digital fuzzy controller makes use of a fuzzy variable "state" to keep track of the behavioural state that it has summoned up in the analogue controller, and to aid in decisions about the next state. The interrelationships between the possible behavioural states are shown in figure 7, and each of the states depicted there is assigned a fuzzy number to identify it within the fuzzy variable "state". Here the system is no longer properly fuzzy, but takes on to a certain extent, a discrete character. The states depicted in figure 7 do not overlap, but because of the nature of the switching of the rule-sets to the analogue chips, must be mutually exclusive - the controller can be in only one state at a time. Nevertheless, fuzzy rules are used to decide on the next state to be entered, and the membership functions have been arranged so that a genuinely fuzzy decision can be made within the group alb-proc-arb. The value output by the fuzzy inference engine for the variable "state" has to be rounded off by the supervisory controller before it is used as an input variable in the next round of fuzzy inference, or to activate the switching of motion rule-sets.

## 5.3 Environment Identification

When the switching between these behavioural states is clean and reliable, *eg* the controller enters a turning state just once to perform a turn, and stays in that state just long enough to complete the maneuver, then the switching between the states can be used as cues to the high-level planner [2] to enable it to time the issuing of its commands (go straight ahead, take next left *etc.*) correctly. Of paramount importance for this clean switching is the reliable detection of a branch - when a branch is within view of the (forward) sensors, and when a branch has been passed (either avoided, or turned into).

Two fuzzy variables are used for this purpose: Left Branch Detected and Right Branch Detected, but again these variables have a certain discreteness in their nature. Although fuzzy inference is used to determine a value for each of these variables, the value is rounded off by the supervisory controller before being passed back in to the fuzzy inference engine for its next round of inference. These two variables are the main cause of switching between the behavioural states, and much care has been put into developing the rules to set their value so that in the simulation environment, as the robot makes its way through a maze of corridors of varying widths,

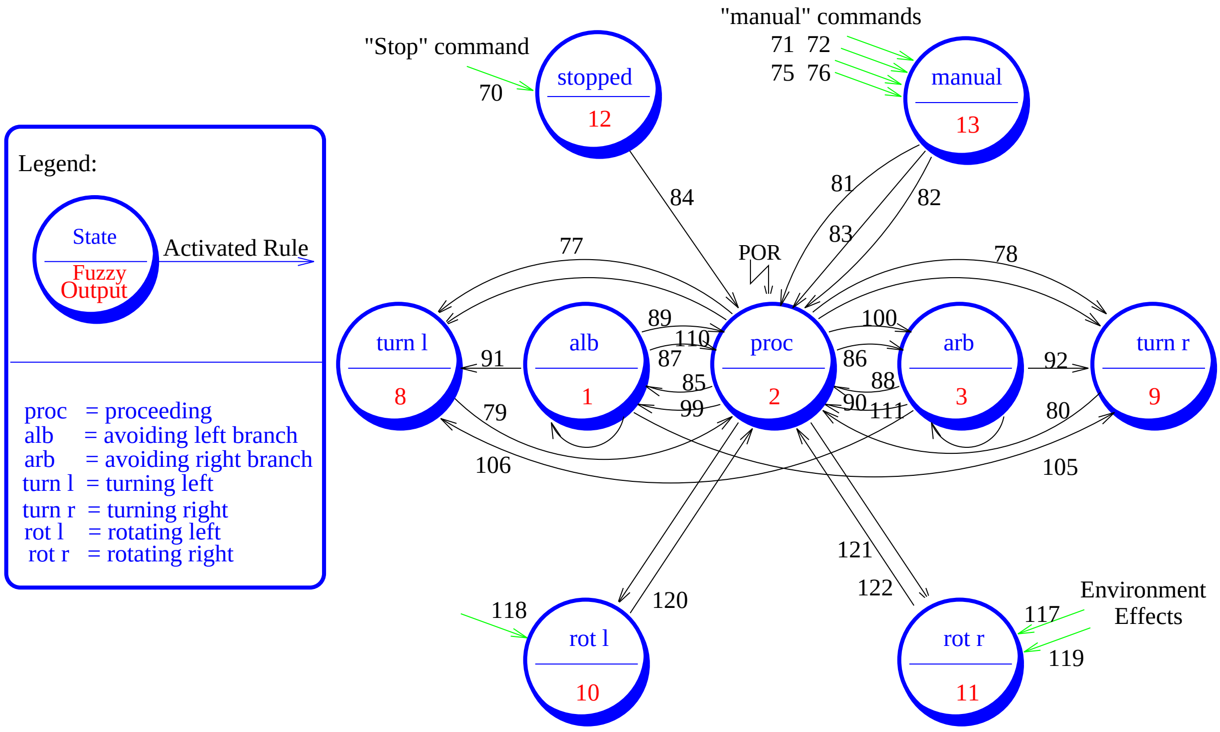


Figure 7: Transitions between the behavioural states

the controller switches reliably and cleanly between the behavioural states, ignoring spurious identifications that may occur when the sensors sweep down a corridor or across obstacles when the robot is maneuvering around a turn, or avoiding an obstacle etc..

The rules and structures described so far are sufficient to enable a "planner" to guide the robot through an imperfectly known environment, using just the goal-oriented commands outlined in the introduction. However, in the exploration phase, the controller needs to know when the robot has come to a dead-end, or some other environmental configuration which would otherwise cause the robot to halt, when under the control of the motion rules as they currently are written. Work is currently underway on the structure of a fuzzy variable "environment" which will enable the controller to identify the salient features of the robot's surroundings, so that an appropriate behaviour pattern can be adopted to allow the exploration to continue. To facilitate such an escape, provision has already been made in the behavioural "states" for the states "rotating left" and "rotating right", as can be seen from figure 7.

**Acknowledgements** This work was done whilst Trevor Lund was absent from the University of Canberra, supported by the University on an "Outside Studies Programme".

# References

[1] A. Saffiotti, E. H. Ruspini, and K. Konolige, *A Multivalued Logic Approach to Integrating Planning and Control.* Menlo Park, CA 94025, USA: Artificial Intelligence Center, SRI International, Technical Note No. 533, 4/1994.

[2] H. Surmann, J. Huser, and L. Peters, "Guiding and controlling mobile robots with a fuzzy controller," in *Fourth IEEE International Conference on Fuzzy Systems, Yokohama, Japan*, pp. 83 – 88, 20. - 24.3.1995, distinguished with Robot Intelligence Award.

[3] S. Guo and L. Peters, "A Reconfigurable Analog Fuzzy Logic Controller," in *FUZZ-IEEE World Congress on Computional Intelligence, Orlando*, pp. 124 – 128, 26.6. - 2.7.1994.

[4] S. Guo, L. Peters, and H. Surmann, "Design and Application of an Analog Fuzzy Logic Controller," *IEEE Transaction on Fuzzy System, Special Issue on Fuzzy Hardware*, vol. submitted paper, 1995.

[5] L. Peters, S. Guo, and R. Camposano, "A novel analog fuzzy controller for intelligent sensors," *Fuzzy Sets and Systems*, vol. 70, no. 2-3, pp. 235 – 247, 1995.

[6] H. Surmann, K. Heesche, M. Hoh, K. Goser, and R. Rudolf, "Entwicklungsumgebung für Fuzzy-Controller mit neuronaler Komponente," *VDE-Fachtagung: „Technische Anwendungen von Fuzzy-Systemen", Dortmund*, pp. 288 – 297, 12. - 13.11.1992.