Behavior-Based Neuro-Fuzzy Controller for Mobile Robot Navigation

Petru Rusu, Emil M. Petriu, *Fellow, IEEE*, Thom E. Whalen, Aurel Cornell, and Hans J. W. Spoelder

***Abstract*—This paper discusses a neuro-fuzzy controller for sensor-based mobile robot navigation in indoor environments. The control system consists of a hierarchy of robot behaviors.**

***Index Terms*—Behavior, mobile robot, neuro-fuzzy control, sensors.**

I. Introduction

FUZZY logic control (FLC) is well suited for controlling a robot because it is capable of making inferences even under uncertainty [1]. Having a hierarchical architecture that divides the FLC into several smaller subsystems will reduce the negative effect that a large rule-base may have on real-time performance, and the problem of insufficient knowledge for designing the rule base can be solved by using a neuro-fuzzy controller [2] and [3].

Learning allows autonomous robots to acquire knowledge by interacting with the environment and subsequently adapting their behavior. Behavior learning methods are used to solve complex control problems that autonomous robots encounter in an unfamiliar real-world environment. Neural networks, fuzzy logic, and reinforcement- and evolutionary-learning methods can be used to implement basic behavioral functions [4]–[7].

This paper discusses an experimental neuro-fuzzy controller for sensor-based mobile robot navigation in indoor environments. The autonomous mobile robot uses infrared and contact sensors for detecting targets and avoiding collisions. The control system is organized in a top-bottom hierarchy of various tasks and behaviors. When multiple low-level behaviors are required, command fusion is used to combine the output of several neuro-fuzzy sub-systems. A switching coordination technique selects a suitable behavior from the set of possible higher level behaviors.

II. Mobile Robot

The circular shaped mobile robot has a differential steering system. Two dc motors independently control two wheels on a common axis. A third wheel (caster) is provided for support. The distance between wheels is 0.70 m, the wheel radius is 0.075 m, the wheel thickness is 0.03 m, and robot's speed is 0.4–0.6 m/s.

The *exteroceptor* system consists of a total of nine Sharp—GP2D12 infrared (IR) range sensors and four Sharp—GP2D15 contact sensors, as shown in Fig. 1. Using IR range sensors instead of sonar sensors allows avoiding some of the drawbacks that the later are usually suffering from multiple reflections, limited speed of firing, and little energy returned to the transducer if sensor does not point normal to the target surface. These sensors are purposefully arranged to cover the whole field around the robot. Four forward facing range sensors are used for collision avoidance. Two lateral range sensors placed at a 90° angle relative to the forward moving direction are used to control smooth contour following movements. Two backward looking range sensors are used while turning corners, when no other sensor can provide useful data. The four contact sensors are placed diagonally at 45° angles relative to the robot's forward moving direction. They provide supplementary information in situations when the range sensors fail to detect obstacles. Such a situation may occur, as an example, when the distance to an obstacle is less than the minimum detection range, (e.g., 0.1 m for a Sharp GP2D12 range sensor).

III. Behavior-Based Control System

We have adopted Mataric's definition for the *robot behavior*, "A behavior is a control law that satisfies a set of constraints to achieve and maintain a particular goal" [8].

The hierarchical organization of the behaviors used for the control of the robot is shown in Fig. 2. Each primitive behavior is self-contained and reacts to data from specific sensors. For instance, *Go-Tangent* behavior reacts to forward facing sensors, *Follow-Wall* behavior reacts to wall sensors, and *Turn-Corner* behavior reacts to backward facing sensors.

The architecture of the behavior-based controller of the mobile robot is shown in Fig. 3. Using the information received from the *exteroceptors* and from its dead-reckoning *proprioceptors*, once every control cycle (100 ms) the controller recalculates the turn angle to be assumed by the moving robot.

All but two primitive behaviors are implemented as *Fuzzy Inference Systems* (FISs). Only *Go-Tangent* primitive behavior is implemented as an *Adaptive Neuro-Fuzzy Inference System* (ANFIS) [3]. The *Turn-Around* primitive is implemented as a hard coded ±180° switch. When multiple primitive behaviors are activated, a command fusion technique is used to combine several fuzzy outputs into a single fuzzy output, which is then *defuzzified*.

In order to reach its goal, *Reaching Two-Dimensional (2-D) location (XY)*, the controller will continuously switch to the

Manuscript received June 15, 2002; revised March 13, 2003. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada and in part by the Communications Research Centre Canada.

P. Rusu and E. M. Petriu are with the School of Information Technology and Engineering, University of Ottawa, Ottawa, ON, K1N 6N5 Canada.

T. E. Whalen is with the Communications Research Centre Canada, Ottawa, ON, K1N 6N5 Canada.

A. Cornell is with the Department of Computer Science, Brigham Young University, Provo, UT 84601 USA.

H. J. W. Spoelder is with the Division Physics and Astronomy, Vrije Universiteit, Amsterdam, The Netherlands.

Digital Object Identifier 10.1109/TIM.2003.816846

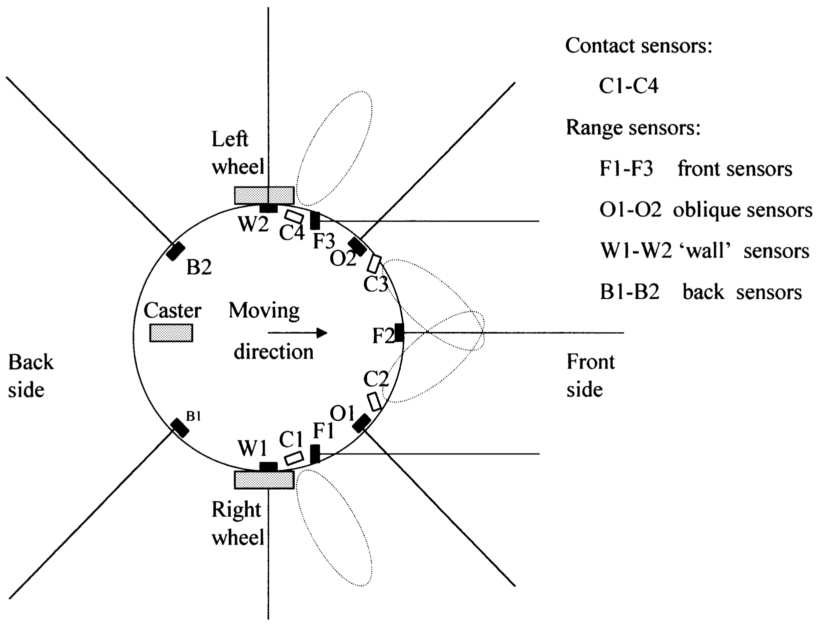

Fig. 1. Sensors of the mobile robot.

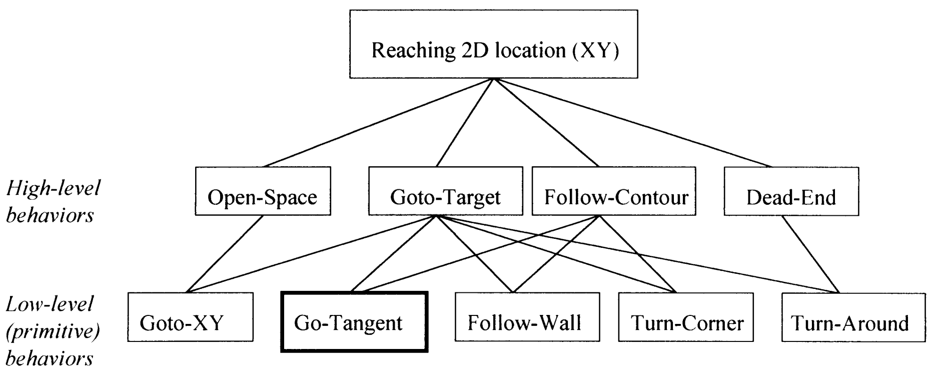

Fig. 2. Hierarchy of the sensor-based robot behaviors.

high-level behaviors that are currently activated by the sensory information. For example, if none of the sensors detects something, the *Open-Space* high-level behavior will be selected. If there is information coming from all the three sides, front, left and right, the *Dead-End* high-level behavior will be activated. If the controller receives only information from the sensors on one of its sides, the *Goto-Target* high-level behavior or *Follow-Contour* high-level behavior will be selected. Choosing between these two high-level behaviors depends on the total rotation angle of the robot relative to the direction toward the target. When this angle passes a certain threshold, the robot is considered 'lost', and it is only allowed to follow a contour but not to go directly to its target at the *XY* location.

The *Very-Close* primitive behavior is an "internal behavior" used only by the *Command fusion module* to invalidate commands that could lead to collisions in situations when the robot is very close to an obstacle. For example, the controller activates the *Goto-XY* primitive behavior based on dead-reckoning information from *proprioceptors*. However, it may happen that the robot actually is too close to an obstacle because dead-reckoning is prone to error. In this case the *Very-Close* internal behavior is activated, which cancels the original command from the *Goto-XY* primitive behavior.

IV. Neuro-Fuzzy Implementation of *Go-Tangent* Behavior

The *Go-Tangent* behavior calculates a turn angle that puts the robot on a trajectory parallel with the surface of the obstacle reflecting the sensor signal. Although real life obstacles have irregular shapes, we consider the object surface detected around the sensor's impact point as planar and normal to the main sensor axis. The angle between this plane and the current robot direction is the angle required for steering the robot on a trajectory parallel with this reflection plane. Because the reflection plane can be assumed tangent locally to the irregular surface of the obstacle, the behavior has been named *Go-Tangent*.

There are two cases to be considered for this behavior.

1) The obstacle is detected by only one sensor oblique to the robot's direction, Fig. 4. Based on the sequence of measured range values to the obstacle we can also calculate the speed of approach. The distance and speed are fed to a

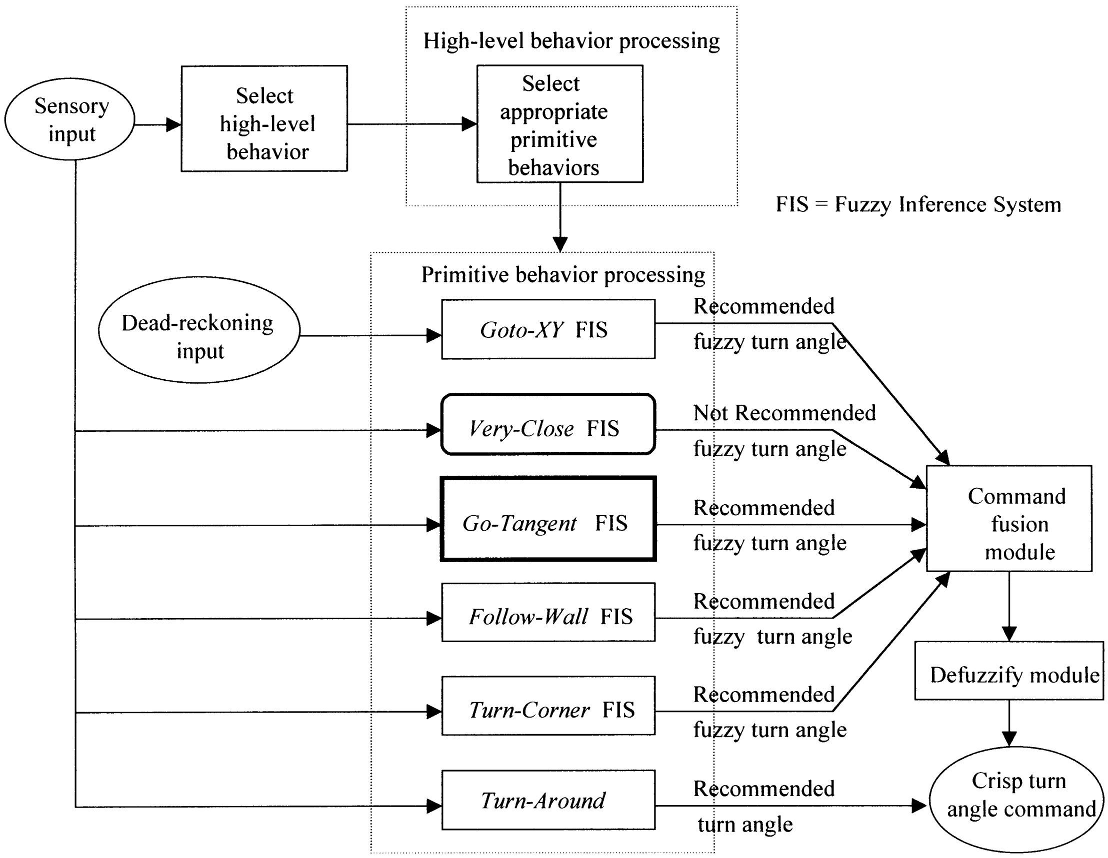

Fig. 3. Architecture of the behavior-based controller.

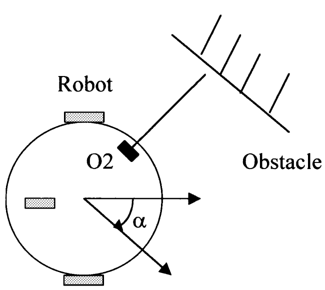

Fig. 4. Sensors used in *Go-Tangent-Oblique-Sensor* case.

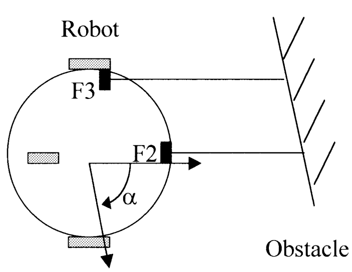

Fig. 5. Sensors used in *Go-Tangent-Front-Sensor* case.

Sugeno-type *Go-Tangent-Oblique-Sensor* ANFIS, which calculates the proper turn angle α.

2) The obstacle is detected by two sensors, Fig. 5. The range data provided by the pair of active sensors ($F2$ and $F1$, or $F2$ and $F3$, in Fig. 1) are fed to a Sugeno-type *Go-Tangent-Front-Sensor* ANFIS, which calculates the proper turn angle α.

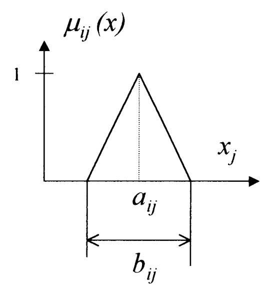

Fig. 6. Generic triangular membership functions for both *Go-Tangent-Oblique-Sensor* and *Go-Tangent-Front-Sensor* cases.

The general form of a fuzzy rule for a 2-input and 1-output first-order Sugeno fuzzy controller is

$$\begin{aligned}&\text{IF } x \text{ is } A_i \text{ AND } y \text{ is } B_j \text{ THEN } F_k = p_k x + q_k y + r_k;\\&\text{for } i = 1, \ldots, L;\ \ j = 1, \ldots, M;\\&k = 1, \ldots, N;\ \ N = L \times M\end{aligned} \tag{1}$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ are the linguistic variables, $\boldsymbol{F}_k$ is the output for the k-th rule, L is the size of the fuzzy set A, M is the size of the fuzzy set B, and N is the size of the rule base.

For *Go-Tangent-Oblique-Sensor* ANFIS, input $\boldsymbol{x}$ is the distance to the obstacle with the corresponding fuzzy set $A = \{short,\ medium\ long\}$, input $\boldsymbol{y}$ is the speed

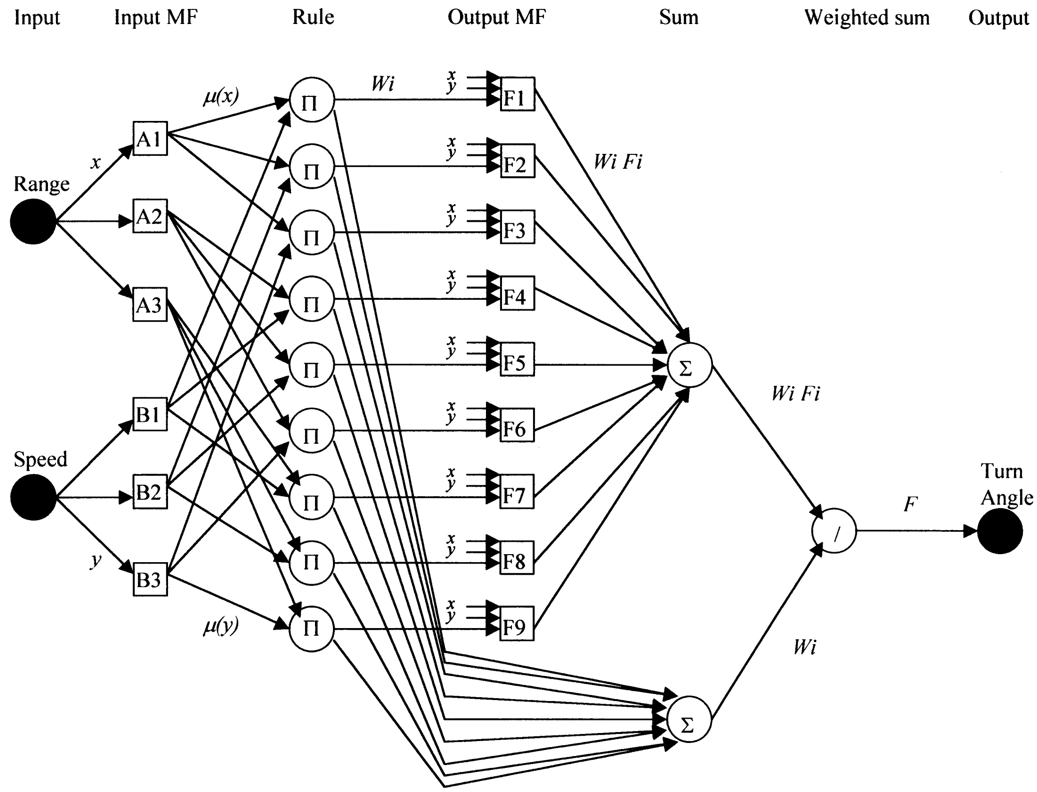

Fig. 7. Neural network identification of the Sugeno ANFIS parameters $\{a_{ij}, b_{ij}\}$ and $\{p_k, q_k, r_k\}$.

toward the obstacle with the corresponding fuzzy set $B = \{slow,\ medium,\ fast\}$. Ouput $\boldsymbol{F}_k$ is the turn angle for the k-th rule, and $N = 9$ is the size of the rule base.

Go-Tangent-Front-Sensor ANFIS is described by a similar Sugeno model with the single difference that in this case both inputs, $\boldsymbol{x}$ and $\boldsymbol{y}$, are range values. The fuzzy set for x being $A = \{short,\ medium\ long\}$, and fuzzy set for y being $B = \{short,\ medium\ long\}$. Ouput $\boldsymbol{F}_k$ is the turn angle for the k-th rule, and $N = 9$ is the size of the rule base. Because of this similarity we will limit our discussion to the *Go-Tangent-Oblique-Sensor* ANFIS.

We have experimented with several types of membership functions for the fuzzy sets A and B and with various sizes for L and M. The *triangular membership functions* and a size 3 for each of the two fuzzy sets, $L = M = 3$, were found to be the simplest and best suited for this case. Fig. 6 illustrates a generic triangular membership function described by the following equation:

$$\boldsymbol{\mu}_{ij}(\boldsymbol{x}_j) = \begin{cases} 1 - \frac{2|x_j - a_{ij}|}{b_{ij}}, & \text{for } a_{ij} - \frac{b_{ij}}{2} < x_j \leq a_{ij} + \frac{b_{ij}}{2} \\ 0, & \text{otherwise} \end{cases}. \tag{2}$$

Fig. 7 shows the structure of the Neural Network (NN) used for the identification of the parameters $\{a_{ij}, b_{ij}\}$ in (2) and $\{p_k, r_k\}$ in (1).

The square elements represent the *adaptive nodes* depending on the parameter set of the adaptive network. The circles represent *fixed nodes*, which are independent of the parameter set. The first layer is composed of adaptive nodes representing the membership functions associated with each linguistic value in (2). The second layer implements the fuzzy rules. It includes only fixed nodes implementing a product operation Π between the membership degrees of the two inputs, $\mu(x)$ and $\mu(y)$, corresponding to the two propositions in the antecedent of each fuzzy rule. The third layer consists of adaptive nodes, which include the output membership function. The other two layers consist of fixed nodes that implement the weighted average procedure for the output $\boldsymbol{F}$ representing the turn angle to be assumed by the robot

$$F = \sum_{k=1}^{N} (\overline{W}k \bullet x) \bullet pk + \sum_{k=1}^{N} (\overline{W}k \bullet y) \bullet qk + \sum_{k=1}^{N} \overline{W}k \bullet rk. \tag{3}$$

As the size of the rule base of the Sugeno FIS is $N = 9$, we will have to identify 27 consequent parameters $\{p_1, \ldots, p_9, q_1, \ldots, q_9, r_1, \ldots, r_9\}$. This will be done by NN using a training set $\{x, y, F\}$ of size P.

We use a *back-propagation* learning algorithm to identify these parameters in two steps. In the *forward pass,* the input membership functions are fixed and the consequent parameters associated with the output are calculated by applying the least square estimation method. Using these parameters, the NN generates an estimate of the turn angle. The difference between this estimate and the turn angle's value from the training set is then *back-propagated* in a *second pass* when the premise parameters associated with the input membership functions are calculated.

The data set we used for training *Go-Tangent-Oblique-Sensor* ANFIS consisted of 29 measurement data points (range, speed, turn angle). We have also used a checking data set obtained by applying a 10% random noise to the training data set. The root mean squared errors of the output over 25 training epochs are plotted in Fig. 8.

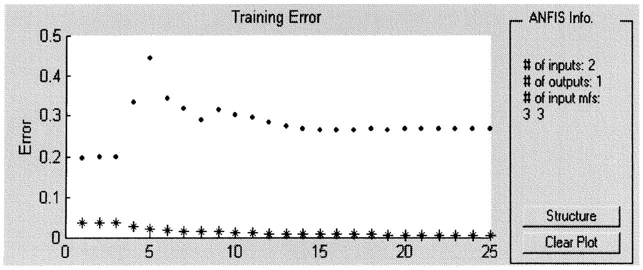

Fig. 8. Plot of the errors during the training of *Go-Tangent-Oblique-Sensor ANFIS*.

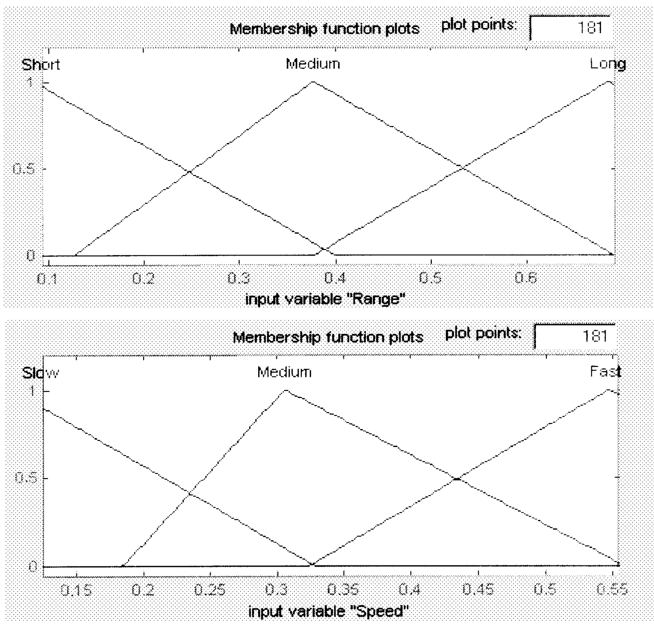

Fig. 9. Membership functions for the two inputs (range and speed) of *Go-Tangent-Oblique-Sensor ANFIS*; the measurement unit for "Range" is the *meter* and for "Speed" is the m/s.

One result of the training is the generation of the set of premise parameters $\{(a_{ij}, b_{ij}) | i = 1, 2, 3; j = 1, 2, 3\}$ from the (2) of the triangular membership functions.

Fig. 9 shows the resulting membership functions for the *range* input and respectively the *speed* input. It may be interesting to note that, although the training started with symmetric membership functions, their shape eventually became asymmetric after training.

The second result of the training is the generation of the set of consequent parameters $\{(p_k, q_k, r_k) | k = 1, 2, , 9\}$ in (1). With these parameters the Sugeno fuzzy rules are as follows.

IF (*Range* is Short and *Speed* is Slow)
THEN $(TurnAngle = -3.253x - 8.858y + 1.561)$
IF (*Range* is Short and *Speed* is Medium)
THEN $(TurnAngle = 4.215x - 8.077y + 1.133)$
IF (*Range* is Short and *Speed* is Fast)
THEN $(TurnAngle = 0.006x - 6.715y + 2.008)$
IF (*Range* is Medium and *Speed* is Slow)
THEN $(TurnAngle = 2.967x - 9.659y + 3.276)$
IF (*Range* is Medium and *Speed* is Medium)
THEN $(TurnAngle = 2.241x - 8.763y + 0.933)$

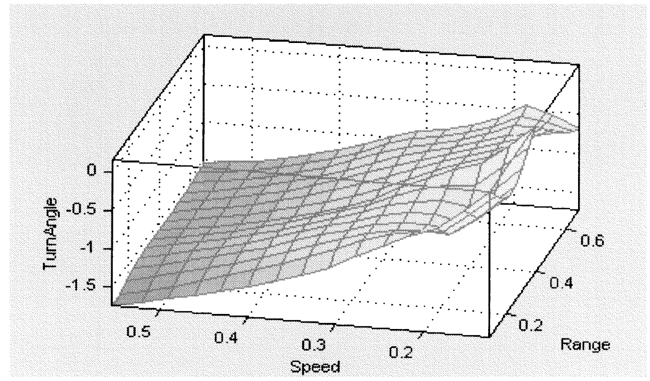

Fig. 10. Input-output characteristics of *Go-Tangent-Oblique-Sensor ANFIS*; the measurement unit for "Range" is the *meter*, for "Speed" is the *meter/second*, and for "Turn Angle" is the *radian*.

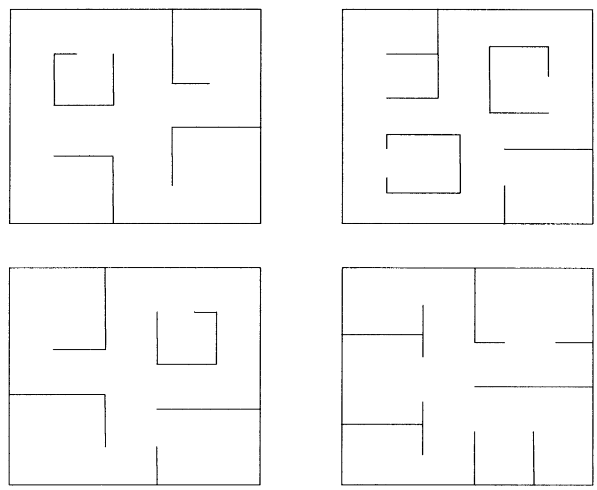

Fig. 11. Shape of the four test settings.

IF (*Range* is Medium and *Speed* is Fast)
THEN $(TurnAngle = -1.113x - 7.021y + 2.762)$
IF (*Range* is Long and *Speed* is Slow)
THEN $(TurnAngle = 3.907x + 4.525y - 3.954)$
IF (Range is Long and *Speed* is Slow)
THEN $(TurnAngle = 1.249x + 3.560y - 2.943)$
IF (*Range* is Long and *Speed* is Slow)
THEN $(TurnAngle = -0.951x + 4.639y - 3.477)$.

Fig. 10 shows the input-output characteristics of the resulting ANFIS controller. It has a good continuity as is normally expected from a Sugeno fuzzy controller.

V. Experimental Results

The behavior-based robot controller has been tested in four simulated maze-like indoor environments, Fig. 11, for over 2,000 endpoints (start point, target point). We used the Rossum's Playhouse simulation environment based on a 2-D robot simulator developed by Lucas [9]. This is a tool specially designed for testing navigation algorithms and control logic for robots and it is based on a client-server architecture. We developed our own client application for the behavior-based neuro-fuzzy controller. For a given set of endpoints, the simulation environment monitors the time needed for each target search, i.e., the time elapsed between the start point and the target point.

The speed of the robot was set at 0.4 m/s. The four test environments varied in size from 10×10 m to 13×13 m. Traveling between two opposite corners of any environment following a path parallel with the walls, has required 60 s. None of the target searches required more than 200 s.

References

[1] A. Saffiotti, "The uses of fuzzy logic in autonomous robot navigation: a catalogue raisonne," *Soft Comput.*, vol. 1, no. 4, pp. 180–197, 1997.

[2] M. Figueiredo and F. Gomide, "Design of fuzzy systems using neuro-fuzzy net-works," *IEEE Trans. Neural Networks*, vol. 10, no. Aug., pp. 815–827, 1999.

[3] J.-S. R. Jang, "ANFIS: Adaptive-network-based fuzzy inference systems," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, pp. 665–685, June 1993.

[4] R. Brooks, "A robust layered control system for a mobile robot," *IEEE J. Robot. Automat.*, vol. 2, pp. 14–23, Feb. 1986.

[5] R. C. Arkin, *Behavior-Based Robotics*. Cambridge, MA: MIT Press, 1998.

[6] H. Danny, T. Lippincott, M. Jamshidi, and E. Tunstel, "Autonomous navigation using an adaptive hierarchy of multiple fuzzy behaviors," in *Proc. Int. Symp. Comput. Intell. Robot. Automat.*, Monterey, CA, 1997, pp. 276–281.

[7] E. Tunstel, "Behavior hierarchy for autonomous mobile robots: Fuzzy-behavior modulation and evolution," *Intl. J. Intell. Automat. Soft Comput., Special Issue: Auton. Contr. Eng.*, vol. 3, no. 1, pp. 37–50, 1997.

[8] M. J. Mataric, "Behavior-based control: Examples from navigation, learning and group behavior," *J. Experimental Theoretical Artif. Intell., Special Issue: Software Architecture Phys. Agents*, vol. 9, pp. 46–54, 1997.

[9] G. W. Lucas. (2000) Rossum's Playhouse RP1—User's Documentation Guide 0.46. [Online]. Available: http://rossum.sourceforge.net

Petru Rusu, photograph and biography not available at the time of publication.

Emil M. Petriu (M'86–SM'88–F'01) received the Dipl. Eng. and Dr. Eng. degrees from the Polytechnic Institute of Timisoara, Romania, in 1969 and 1978, respectively.

He is currently a Professor with the School of Information Technology and Engineering, University of Ottawa, Ottawa, ON, Canada, where he has been since 1985. His research interests include intelligent sensors, robot sensing and perception, test and measurement systems, interactive virtual environments, neural-networks, and fuzzy control.

Dr. Petriu is a Registered Professional Engineer in the province of Ontario, Canada. He is a Fellow of the Engineering Institute of Canada and Fellow of the Canadian Academy of Engineering. He is currently serving as member of the AdCom, Co-Chair of TC-15 Virtual Systems Committee, and Co-Chair of TC-28 Instrumentation and Measurement for Robotics and Automation of the IEEE Instrumentation and Measurement Society. He is an Associate Editor of the IEEE Transactions on Instrumentation and Measurement.

Thom E. Whalen received the doctorate degree in experimental psychology from Dalhousie University, Halifax, NS, Canada, in 1979.

He conducts research in human-computer interactions at the Communications Research Centre (CRC), a research institute of the Government of Canada. His current research interests include access to information through natural language queries, retrieval of images using visual features, Web-based training, and the control of avatars in shared virtual environments. He is also an Adjunct Professor in the Psychology Department of Carleton University and in the Department of Management Science and Finance at St. Mary's University.

Aurel Cornell, photograph and biography not available at the time of publication.

Hans J. W. Spoelder, photograph and biography not available at the time of publication.