ELSEVIER

Expert Systems with Applications 14 (1998) 179–186

Expert Systems with Applications

Fuzzy knowledge-based controller design for autonomous robot navigation

Marc Boumedine Montaner* and Alejandro Ramirez-Serrano

Instituto Tecnológico y de Estudios Superiores de Monterrey, Campus Ciudad de México, Calle del Puente 222, Tlalpan 14380, México D.F., México

Abstract

In this study a fuzzy logic controller for mobile robot navigation has been designed. The designed controller deals with the uncertainty and ambiguity of the information the system receives. The technique has been used on an experimental mobile robot which uses a set of seven ultrasonic sensors to perceive the environment. The designed fuzzy controller maps the input space (information coming from ultrasonic sensors) to a safe collision-avoidance trajectory (output space) in real time. This is accomplished by an inference process based on rules (a list of IF–THEN statements) taken from a knowledge base. The technique generates satisfactory direction and velocity maneuvers of the autonomous vehicle which are used by the robot to reach its goal safely. Simulation and experimental results show the method can be used satisfactorily by wheeled mobile robots moving on unknown static terrains.

1. INTRODUCTION

Robots have been used for specialized tasks such as couriers in offices, hospitals, security, undersea and underground exploration, space exploration, and many other kinds of tasks too risky for human beings. The design of a autonomous robot is a complex task and the criteria of success are evaluated in terms of its capabilities to make decisions and to act by itself in a reliable and satisfactory manner. Robot navigation has been studied extensively in the past, employing different approaches (Lozano-Perez & Wesley, 1979; Bouchon & Despres, 1987; Doshi et al., 1988; Lumelsky et al., 1990; Barshan & Durrant-Whyte, 1995; Beaufrere & Zeghloul, 1995; Kundur & Raviv, 1995); however, the navigational problem becomes extremely complex when the robot has to deal with unknown environments and ambiguous imprecise environment information in real time (Surman et al., 1995; Freund & Hayer, 1988). For many years, researchers have tried to develop navigation algorithms to deal with the inherently imprecise information received from sensors. The majority of such algorithms include maps of the terrain and consider exact information from sensors. Only few of them are suitable to be used in unknown terrains (without maps). One main disadvantage of such algorithms, developed so far, is that they typically relay on exact information obtained by a large number of sensors to reduce the ambiguity of the data, increasing the cost of the machine. Recently, Beaufrere & Zeghloul (1995) developed a fuzzy logic controller using information from a few ultrasonic range proximity sensors to control the navigation of mobile robots in dynamic environments. Their approach reduces the large quantity and variety of sensors usually used in autonomous vehicles and at the same time handles uncertain and noisy data. The problem with their approach is that they do not control the robot's velocity, which has not been addressed by many researchers, and consequently it does not optimize the travel time of the robot from one point to another. To fulfill this need it will be useful to reduce the travel time as well as the distance traveled by developing a device to control the velocity as well as addressing the problem of imprecise and uncertain information. In unknown environments optimal navigation distance is unknown, and therefore, cannot be formally optimized. However, travel time can be reduced by controlling the robot's velocity.

This study presents the steps for designing a fuzzy logic controller for a mobile robot called MOBIROB. The controller will guide MOBIROB from an initial position to a final position (goal), without any prior knowledge of the environment and in the shortest

* Author for correspondence.

0957-4174/98/$19.00
PII S0957-4174(97)00059-6

possible travel time. To simplify the explanation, we suppose that the environment consists of static objects. The controller will have to build a plan for a collision-free path at the run time. The procedure or the planning algorithm has to make real-time decisions such as speed up, turn left, slow down, halt, etc., to reliably reach the goal.

This paper is organized in five sections. In Section 2, we present the motivation for choosing fuzzy knowledge-based control of the movements of our robot. Section 3 describes the fuzzy knowledge-based architecture. Simulation and experimental results are discussed in Section 4. Finally, Section 5 gives a summary of our technique and presents future directions to extend and improve our results.

2. MOTIVATION FOR USING FUZZY-BASED KNOWLEDGE

MOBIROB can be defined as a programmable wheeled device whose decisions for reaching a target goal are guided by ultrasonic sensors. To be able to navigate, the mobile robot has to obtain information about its surroundings. A first step in the navigation control process for mobile robots is the detection of obstacles that may damage the robot. In this case, our mobile robot only obtains information that is in front of the vehicle (we only consider forward velocities). However, the system can be easily extended to include backward velocities, as explained later in the paper. Environment information is obtained by a set of seven ultrasonic sensors mounted on top of the robot (see Fig. 1). The information obtained via the sensors is the relative position of the obstacle expressed in the two robot polar coordinates Pos $=(D,\theta)$, where D is the distance and θ is the angular direction of the obstacle. The seven parameters Pos (one for each sensor) are used to determine the future velocity and direction of the robot, so avoidance maneuvers can be taken. The environment data obtained by the sensors (distance and orientation of the obstacles) is analyzed by the robot, whose controller will make simple and fast decisions at run time.

Each Pos $=(D,\theta)$ parameter obtained by each sensor is

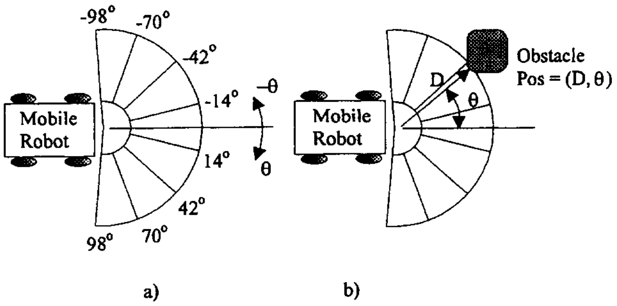


FIGURE 1. (a) Area of perception for the seven ultrasonic sensors. (b) Obstacle detected at distance D and at angle θ (both relative to the robot) by one of the ultrasonic sensors.

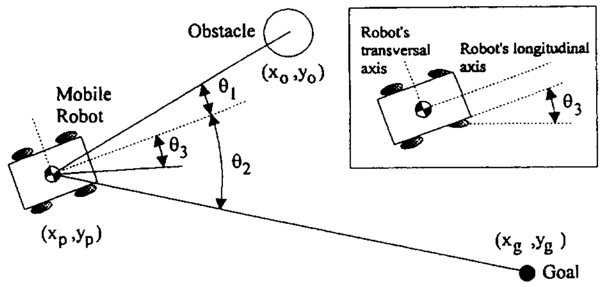


FIGURE 2. Geometric configuration of the navigation problem of a mobile robot.

independently observed and analyzed by the controller. For each sensor a direction and velocity vector is obtained by the fuzzy controller. Seven direction and seven velocity vectors are obtained. These vectors are expressed with respect to the robot's longitudinal (X-axis) and transversal (Y-axis) axes (see Fig. 2). The final direction carried out by the robot is obtained by a simple summation over the previously seven direction vectors obtained. In narrow corridors, for example, the controller, based on the left-sensor information, will command the robot to go to the right (avoiding the left wall). At the same time, based on the right sensor environment information, the controller will command the vehicle to go to the left (avoiding the right wall). Similar commands (direction vectors) will be obtained for each of the remaining sensors. Adding all (seven) direction vectors obtained by the controller, a final direction command is sent to the robot. In the narrow corridor, for example, the robot will try to move away from the walls, and at the same time it will try to move toward the goal. In this sense, direction vectors can be seen as rejection (from obstacles) and/or attraction (to the goal) vectors.

For the velocity case, the robot's velocity is based on how far obstacles are detected from the robot. If objects are detected near (far from) the robot, no matter which sensor detected the obstacle(s), a small (large) velocity command will be send by the fuzzy controller to the robot. For safety reasons the smallest of the seven velocity vectors is used as the final velocity. The magnitudes of all direction and velocity vectors are proportional to the distance of the obstacle detected: direction-vector$_i = \alpha d_i$, velocity-vector$_i = \beta d_i$, where $i = 1,2,\ldots$; the number of sensors, d_i is obtained from the Pos $=(D,\theta)$ parameter, and α and β are proportional 'safety' constants with a value of unity.

Using this approach to direct the robot in a safe way to its goal point, the system can be easily expanded by adding sensors on other parts of the robot (i.e. rear) and extending the knowledge base incorporating rules for such new sensors. Sensors can be added until an omnidirectional perception of the environment is achieved. However, more sensors imply more computing time will be needed to obtain the final direction and velocity commands. Therefore, a trade-off between environment perception and computing time must be made for the robot to properly react in real time as perception of the environment constantly changes.

The sensors used on MOBIROB are Polaroid electrostatic transducers controlled by a 68HC11 microcontroller. The sensors can detect objects between the range of 0.15 and 10.5 m. The microcontroller determines the distance of each detected obstacle according to the signal received by each electrostatic transducer. All seven sensors work and behave in the same manner regardless of their position. This simplifies the analysis of the environment and the construction of the system. To avoid crosstalk between the sensors, transducers are fired (send a ping signal) in a sequence such that the next sensor to be fired is as much as possible in a perpendicular line from the previous fired sensor. The following is the firing sequence used in the study: left-sensor ($-98°:-70°$), front-sensor ($-14°:14°$), right-sensor ($70°:98°$), left-front-sensor ($-70°:-42°$), front-right-sensor ($14°:42°$), front-left-sensor ($-42°:-14°$), right-front-sensor ($42°:70°$).

2.1. Collision Avoidance Strategy

Once the distance and direction of the obstacles are known, a collision avoidance maneuver can be planned. Our objective is to develop a control system that will enable a mobile robot to navigate from a start point to a goal point, without collisions, in the shortest possible travel time. The geometrical configuration of the navigation problem is illustrated in Fig. 2. To increase the efficiency of safe navigation, travel time and distance traveled must be minimized. To accomplish this, the robot must steer toward the goal, reducing the error angle θ_2, and modifying its velocity depending on how far the robot is from the goal and if an obstacle is detected. On one hand, high velocities are achieved when the robot is far from the goal point (x_g,y_g) and there are no obstacles near the vehicle. On the other hand, low velocities are established when the robot is near an obstacle or near the goal point. If an obstacle is detected, angle θ_1 (obstacle angle) must be increased, changing the steering angle θ_3. After the obstacle has been avoided angle θ_2 must again be reduced.

2.2. Fuzzy Logic Approach

Our objective is to design a controller to guide the robot safely in an unknown environment from an initial point to a final point. The robot will have to take actions such as to speed up, turn left, turn right, slow down, etc. These actions are taken by determining or controlling the values of variables such as velocity, steering angle, obstacle angle, etc. These variables are called output variables. To compute the values of output variables $(y_1,\ldots,y_m)$ it is possible to determine the change of input variables $(x_1,\ldots,x_n)$ such as the distance of the robot from an obstacle, the velocity of the robot, the error angle, etc. The solution of the control problem is to define for each output variable y_i, $i=1,\ldots,m$, a control function χ: $X_1x\ldots xX_n \rightarrow Y_i$ that assigns to each input tuple $(x_1,\ldots,x_n) \in X_1x\ldots xX_n$ its adequate output $y_i=\chi(X_1,\ldots,x_n)$.

To model how the robot will behave a controller can be designed using classical methods such as differential equations. Although mathematical models are well suited for obtaining an exact solution to problems where all necessary information is available, it is extremely difficult to specify an accurate model of the robot behavior for responding efficiently to hazards without intensive and complex computational processes.

Alternative approaches have been explored, based on human knowledge, to make decisions more efficiently. Human knowledge can satisfactorily deal with such information in an efficient manner. For instance, humans can drive a small car without knowing the exact mathematical models of the system. To solve the navigation problem a model based on human knowledge can be built. An expert is asked to formalize his knowledge in the form of linguistic rules for navigating in an unknown environment. An example of a linguistic rule is the following:

IF obstacle distance relative to the robot is short and obstacle angle is Right Small and error angle is Right Big THEN steering angle is Right Big.

Linguistic rules have an antecedent and a consequence. The antecedent describes a fuzzy situation and the consequence specifies a fuzzy output. Fuzzy logic has been successfully used in various knowledge-based systems to control real-time decision-making in the area of command and control, in environments where no mathematical model can be applied with efficiency (information is imprecise, models are too expensive, etc.) (Bouchon & Despres, 1987; Ramirez-Serrano & Boumedine, 1996). Robot navigation in unknown terrains is a very complex task, difficult to control, basically because of the great amount of imprecise and ambiguous information that has to be considered. However, the common sense knowledge acquired by humans (linguistic rules and reasoning process) can be easily represented as a knowledge base. In this context, in addition to the imprecision and uncertainty of the information perceived

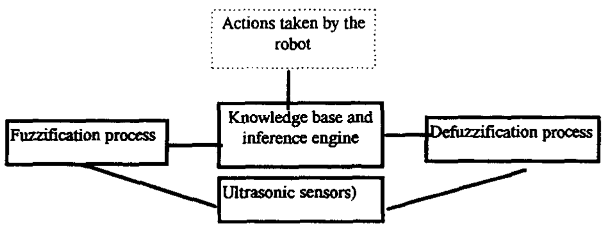
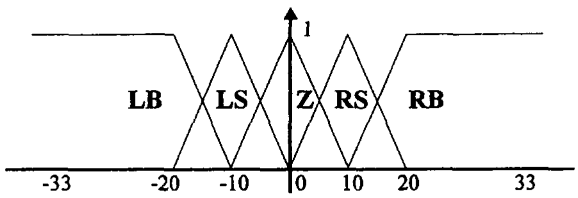


FIGURE 3. Architecture of the knowledge-based fuzzy controller.

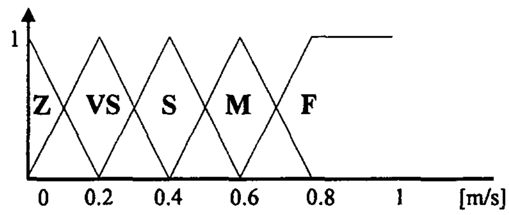
TABLE 1
Fuzzy sets with their respective adjectives

Obstacle angle (θ_1)	Error angle (θ_2)	Steering angle (θ_3)	Robot's velocity	Distance from the robot to the goal	Obstacle distance relative to the robot
Left Big (LB)	Left Big (LB)	Left Big (LB)	Zero (Z)	Zero (Z)	Near (N)
Left Small (LS)	Left Small (LS)	Left Small (LS)	Very Slow (VS)	Near (N)	Medium (M)
Zero (Z)	Zero (Z)	Zero (Z)	Slow (S)	Regular (RE)	Far (FA)
Right Small (RS)	Right Small (RS)	Right Small (RS)	Medium (M)	Medium (M)	-
Right Big (RB)	Right Big (RB)	Right Big (RB)	Fast (F)	Far (FA)	-



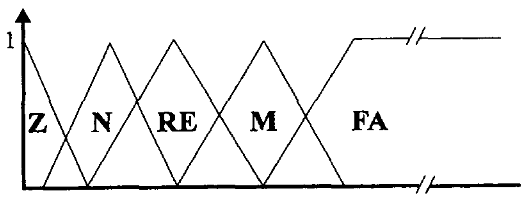
LB x	LB μ(x)	LS x	LS μ(x)	Z x	Z μ(x)	RS x	RS μ(x)	RB x	RB μ(x)
-33	1	-20	0	-10	0	0	0	10	0
-20	1	-10	1	0	1	10	1	20	1
-10	0	0	0	10	0	20	0	33	1

FIGURE 4. Fuzzy sets for the linguistic variable '*Steering angle* (θ_3)'.



Z x	Z μ(x)	VS x	VS μ(x)	S x	S μ(x)	M x	M μ(x)	F x	F μ(x)
0.0	1	0	0	0.2	0	0.4	0	0.6	0
0.2	0	0.2	1	0.4	1	0.6	1	0.8	1
		0.4	0	0.6	0	0.8	0	1.0	1

FIGURE 5. Fuzzy sets for the linguistic variable '*Robot's velocity*'.



Z x	Z μ(x)	N x	N μ(x)	RE x	RE μ(x)	M x	M μ(x)	FA x	FA μ(x)
0	1	5	0	10	0	25	0	40	0
10	0	15	1	25	1	40	1	55	1
		25	0	40	0	55	0	1000	1

FIGURE 6. Fuzzy sets for the linguistic variable '*Distance from the robot to the goal*'.

from the environment, other sources of imprecision and uncertainty have to be considered, such as rule description, reasoning process, etc. In these circumstances, the quality of the decision is strongly dependent on the fuzzy models that will be implemented to control the movements of the machine.

3. ARCHITECTURE FOR THE FUZZY KNOWLEDGE-BASED CONTROLLER

To design a fuzzy knowledge-based controller, several steps have to be stressed. Input values perceived by the sensors are crisp values and the controller should be able to produce the appropriate output value. For this, we have to define a computational model for processing linguistic terms used in the rules and to determine to which degree a crisp value corresponds to a linguistic expression. Also, it is necessary to define how linguistic terms will be represented by a fuzzy set in our model. Then, a reasoning mechanism to process the rule has to be implemented to produce fuzzy outputs or conclusions. Then, knowing the fuzzy outputs which have been derived from our rules, we need to transform these linguistic terms to the form of crisp output values. The following section describes in detail the steps previously mentioned.

3.1. Environment Perception

This navigation scheme is achieved in real time by the fuzzy controller that takes information from the ultrasonic sensors and according to the general specifications previously described.

The fuzzy controller used in our study was designed using five basic steps, as outlined in the following paragraphs (see Fig. 3). These steps are also described in the book by Kruse et al. (1995). After identifying the variables (inputs and outputs) with their corresponding ranges for the controller, meaningful linguistic states and fuzzy sets for each variable were established. As the second step, to express the uncertainty, membership functions (μ(x)) were developed for each linguistic variable previously defined. The membership functions developed and some of their corresponding linguistic states are represented in Table 1 and in Figs 4–6. The third step was devoted to formulating fuzzy inference rules using human knowledge (regarding the control

TABLE 2
Rules necessary to change and avoid obstacles; obstacle distance = Medium (M)

<table>
<tr><th colspan="2">Steering Angle (θ_3)</th><th colspan="5">Error angle (θ_2)</th></tr>
<tr><th colspan="2"></th><th>Left Big (LB)</th><th>Left Small (LS)</th><th>Zero (Z)</th><th>Right Small (RS)</th><th>Right Big (RB)</th></tr>
<tr><td rowspan="5">Obstacle angle (θ_1)</td><td>Left Big (LB)</td><td colspan="2">Left Small (LS)</td><td colspan="2" rowspan="3">Right Small (RS)</td><td rowspan="4">Right Big (RB)</td></tr>
<tr><td>Left Small (LS)</td><td rowspan="4">Left Big (LB)</td><td>Zero (Z)</td></tr>
<tr><td>Zero (Z)</td><td rowspan="3" colspan="2">Left Small (LS)</td></tr>
<tr><td>Right Small (RS)</td><td>Zero (Z)</td></tr>
<tr><td>Right Big (RB)</td><td colspan="2">Right Small (RS)</td></tr>
</table>

problem) obtained from experience. In this case, knowledge was obtained via our common sense. The entire body of knowledge (set of rules), needed by the mobile robot to navigate, was expressed in four tables (such as Table 2) containing 47 rules needed to navigate autonomously from a start point A to a goal point B, avoiding obstacles. One example rule, taken from Table 2, is:

IF	Obstacle distance relative to the robot is MEDIUM
AND	Obstacle angle is RIGHT SMALL

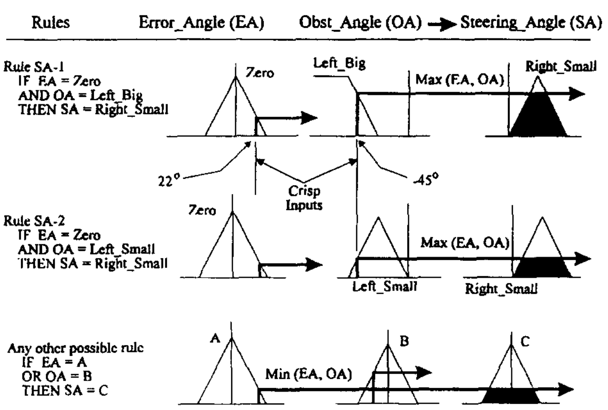


FIGURE 7. Fuzzification example corresponding to Table 2. An extra, hypothetical rule not present in Table 2 has been added to the example to observe all possible defuzzification techniques considered in the study (see Fig. 8).

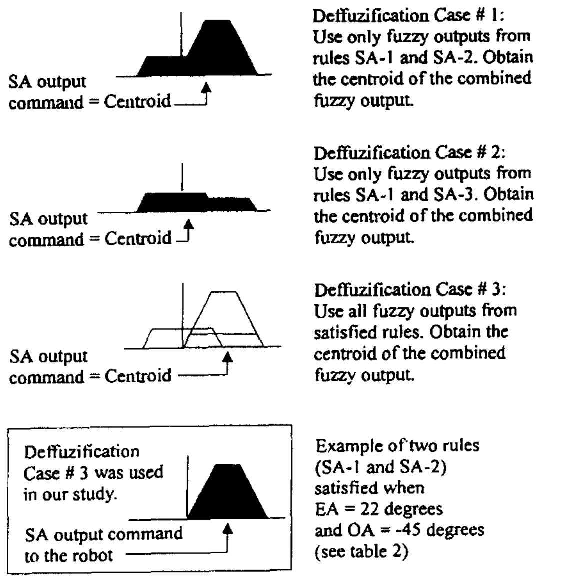


FIGURE 8. Output defuzzification techniques (our design used Case 3).

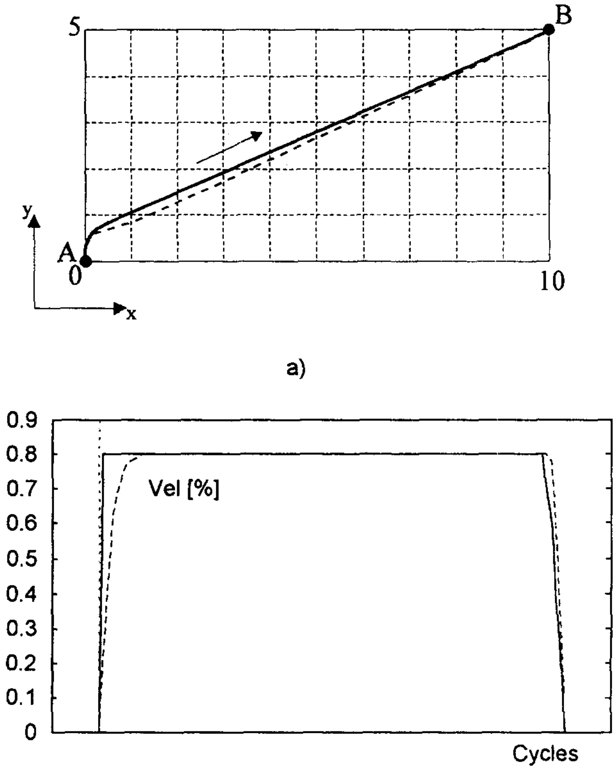


FIGURE 9. Controller behavior on terrain 1. Simulation (continuous line) and real navigation (dashed line). (a) Trajectory. (b) Velocity.

AND Error angle is RIGHT BIG
THEN Steering angle is RIGHT BIG

The fourth step corresponds to the definition of the inference engine. Fuzzy inference mechanisms can be represented in several ways (Mamdani, Rescher–Gaines, Kleene–Dienes, Brouwer–Godel, Goguen, Lukasiewsky, etc.) (Eshragh & Mamdani, 1981; Bouchon & Despres, 1987; Kruse et al., 1995), and their properties and effects have to be studied before implementation in the real system. Therefore, before incorporating the fuzzy control system for guiding the robot, in our approach, we chose to simulate first the behavior of the robot using the Mamdani implication technique (see Fig. 7). The main reason for choosing this technique was its simplicity and high reliability. After simulation, the controller was tested on our experimental mobile robot. The results obtained are reported in Section 4. Finally, conclusions in fuzzy form obtained by the inference engine were converted to real numbers. Each number obtained by the defuzzification process (fifth step) represents the action to be taken by the controller. In this work the centroid method or the center of gravity (COG) method has been preferred over methods such as the Max criterion or the mean of the Maxima method. The main advantage of COG is that the behavior of the robot is always fairly smooth, which is beneficial for robot navigation (see Fig. 8).

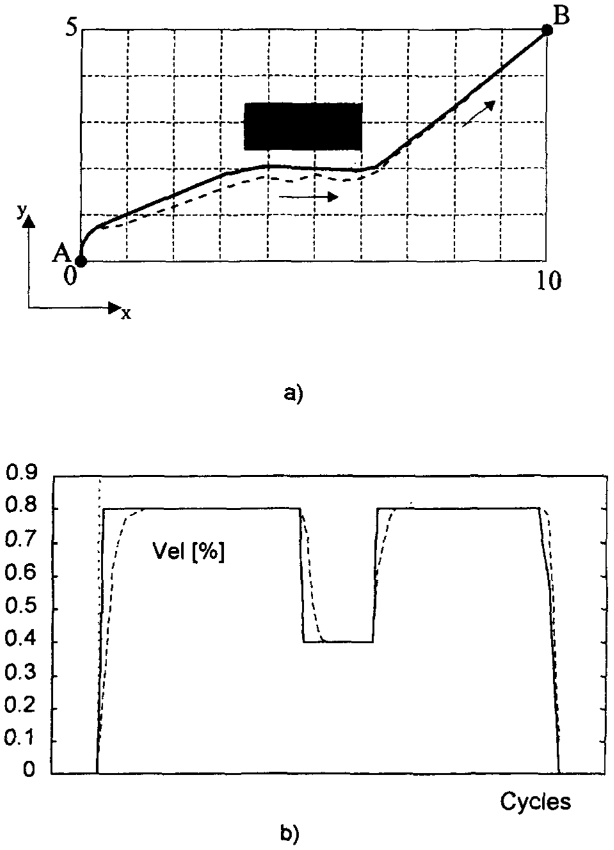


FIGURE 10. Controller behavior on terrain 2. Simulation (continuous line) and real navigation (dashed line). (a) Trajectory. (b) Velocity.

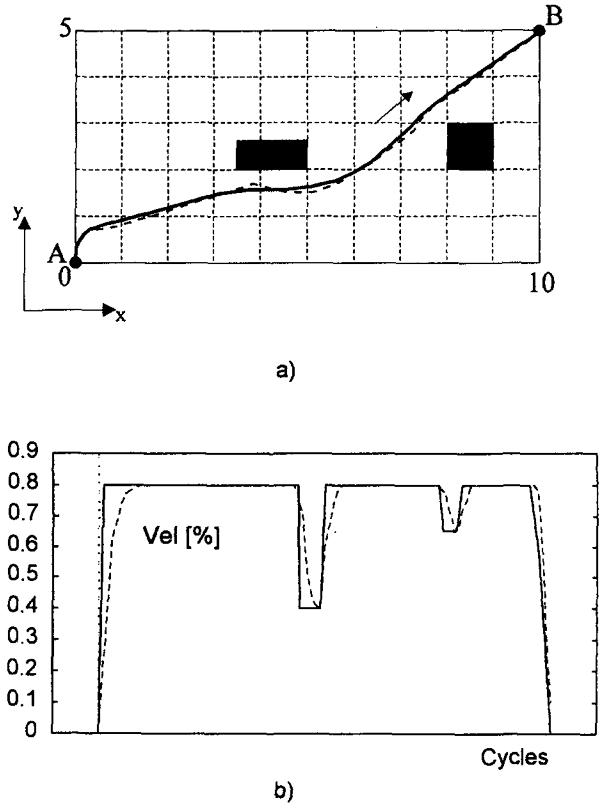
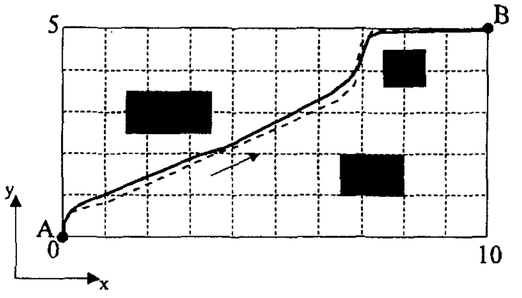


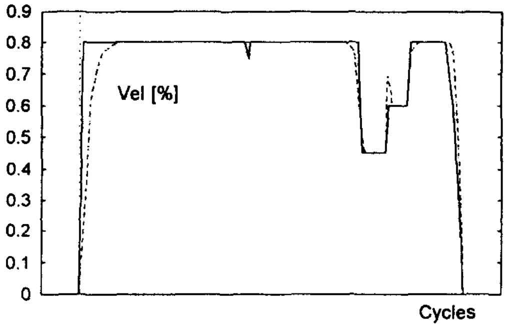
FIGURE 11. Robot behavior on terrain 3. Simulation (continuous line) and real navigation (dashed line). (a) Trajectory. (b) Velocity.

4. RESULTS

To test and analyze the controller, simulation and experimental studies were performed. Simulation was performed using the MS-DOS Matlab Fuzzy Logic Toolbox. On the other hand, experimental results were obtained using MOBIROB, a US$300.00 experimental mobile robot developed at ITESM-CCM. During the study, position, direction and velocity of the robot were studied. Figs 9(a), 10(a), 11(a), 12(a) and 13(a) show simulation and experimental results of the motion of the robot as it navigates from a start position A to a goal point B through various static environments. The environment information obtained by the robot (obstacle sensing) is the input to the fuzzy controller which, gives the direction and velocity of the robot as its output. The fuzzy controller continually reads information and provides an output until the robot reaches the goal point. To accomplish this, the system also maintains the current position of the robot within a coordinate system. Figures 9(b), 10(b, 11(b), 12(b) and 13(b) show how the robot accelerates when no obstacle is near, whereas lower velocities are reached when obstacles are present near the robot (just as desired). This allows a safer navigation and increases the robot's performance in terms of travel

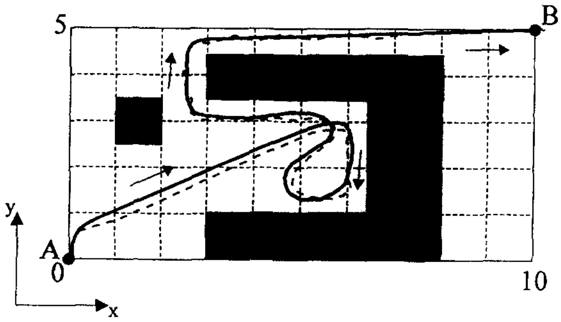


a)

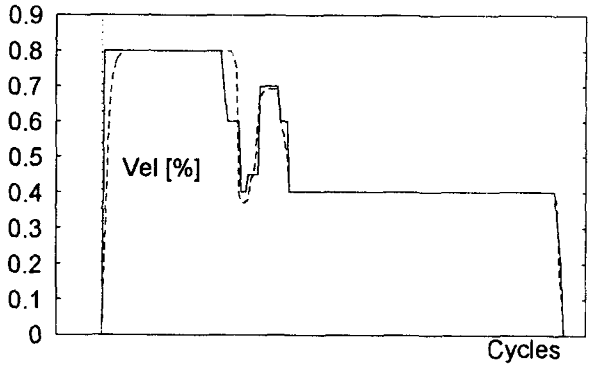


b)

FIGURE 12. Controller behavior on terrain 4. (a) Navigation trajectory. (b) Robot's velocity. Continuous and dashed lines represent simulation and experimental results, respectively.



a)



b)

FIGURE 13. Controller behavior on terrain 5. (a) Navigation trajectory. (b) Robot's velocity. Continuous and dashed lines represent simulation and experimental results, respectively.

time. To perceive and avoid several obstacles at the same time, the controller checks the entire knowledge base after the seven ultrasonic sensors transmit their corresponding information (obstacle distance) to the robot. The control signals (direction or velocity) transmitted to the robot are obtained using all of the fuzzy conclusions derived by all of the satisfied individual rules. After control signals (direction and velocity) are transmitted to the robot a new set of environment information is captured by the sensors and transmitted to the fuzzy controller. This strategy is applied until the robot reaches its goal.

During the experimental phase of the study our autonomous mobile robot (MOBIROB) was directly linked (hard-wired) to a 486 personal computer (PC) used to store the real inputs (environment data) and output (direction and velocity) variables to and from the robot. The experimental position of the robot within the given coordinate system (see Figs 9(a), 10(a), 11(a), 12(a) and 13(a) was determined using the experimental output variables stored in the linked PC. Here the system (robot) was considered slip-free.

5. SUMMARY AND CONCLUSIONS

We have presented an approach for designing a fuzzy controller for autonomous mobile robot navigation to reduce the travel time from an initial position to a final point (goal). The navigation trajectory is obtained after mapping the information obtained from the real world via ultrasonic sensors. In our approach, we designed a first knowledge base which modeled the behavior of an expert human and the inference implication was derived using the Mamdani approach (Eshragh & Mamdani, 1981). According to our simulation and experimental results, the designed fuzzy controller has demonstrated satisfactory behavior. As the information needed to navigate is obtained via ultrasonic sensors the method can be easily extended to include an analysis of possible moving obstacles. Thus, the method can be greatly improved by minimizing both the distance traveled and the travel time, taking into account the velocity (and possible acceleration) of the obstacles. This can be done by integrating a fuzzy set representing the velocity (acceleration) of the obstacles, and integrating the corresponding rules (knowledge) into the system.

The method presented can be easily extended to include backward velocities (not considered in this study), increasing the maneuverability of the robot. For this to be possible, sensors have to be incorporated in other parts of the robot and corresponding rules added into the knowledge-base. Obviously, new membership functions have also to be added to the existing ones.

Our next step is to experiment with other fuzzy inference techniques and to present a comparative study of the results. Also, the controller will be modified to navigate within dynamic environments. At the first stage, continuing with our assumptions, some problems, such as the effect of a wheel rolling over small obstacles not detected on dead reckoning, will not be considered. The use of the fuzzy controller is not limited to common static environments such as factories and warehouses, but can also be used in dynamic environments, where unknown moving objects are part of the surroundings.

REFERENCES

Barshan, B., & Durrant-Whyte, H. F. (1995). Inertial navigation system for mobile robots. *IEEE Transactions on Robotics and Manufacturing*, **11**(3), 328–342.

Beaufrere, B., & Zeghloul, S. (1995). Avoidance of moving obstacles by a mobile robot. In *Proceedings of the 3rd IASTED International Conference on Robotics and Manufacturing*, pp. 237–240.

Bouchon, B., & Despres, S. (1987). *Propagation of Uncertainty and Inaccuracies in an Expert System. Lecture Notes in Computer Science, 286*. Berlin: Springer.

Doshi, R. S., Lam, R., & White, J. E. (1988). Region-based route planning: multi-abstraction route planning based on intermediate level vision processing. In *Proceedings of SPIE—The International Society for Optical Engineering, Cambridge, MA, Sensor Fusion: Spatial Reasoning and Scene Interpretation, (1003)*, pp. 470–489.

Eshragh, F., & Mamdani, E. H. (1981). In E. H. Mamdani & B. R. Gaines (Eds) *A General Approach to Linguistic Approximation in Fuzzy Reasoning and its Applications*. New York: Academic Press.

Freund, E., & Hoyer, H. (1988). Real-time pathfinding in multirobot systems including obstacle avoidance. *International Journal of Robotics Research*, **7**(1), 42–70.

Kruse, R., Gebhart, J., & Klawonn, F. (1995). *Foundations of Fuzzy Systems*. Chichester, UK: Wiley.

Kundur, S. R., & Raviv, D. (1995). Visual motion cue for autonomous navigation. In *Proceedings of the 3rd IASTED International Conference on Robotics and Manufacturing*, pp. 67–69.

Lozano-Perez, T., & Wesley, M. A. (1979). An algorithm for planning collision-free paths among polyhedral obstacles. *Communications of the ACM*, **22**(10), 560–570.

Lumelsky, V. J., Mukhopadhyay, S., & Sun, K. (1990). Dynamic path planning in sensor-based terrain acquisition. *IEEE Transactions on Robotics and Automation*, **6**(4), 462–472.

Ramirez-Serrano, A., & Boumedine, M. (1996). Real-time navigation in unknown environments using fuzzy logic and ultrasonic sensing. In *Proceedings of the 11th International Symposium on Intelligent Control, Dearborn, MI*, pp. 26–31.

Surman, H., Huser, J., & Peters, L. (1995). Fuzzy system for indoor robot navigation. In *Proceedings of the Fourth IEEE International Conference on Fuzzy Systems, Yokohoma*, pp. 83–86.