Fuzzy Obstacle Avoidance and Navigation for Omnidirectional Mobile Robots

Panagiotis G. Zavlangas, Prof. Spyros G. Tzafestas, Dr. K. Althoefer*
Intelligent Robotics and Automation Laboratory
Department of Electrical and Computer Engineering
National Technical University of Athens
Zografou, 15773 Athens, Greece
Phone: +30-1-7721527, Fax: +30-1-7722490
email: zavlang@central.ntua.gr
e-mail: tzafesta@softlab.ece.ntua.gr
* Department of Mechanical Engineering
King's College London
Strand Campus, Strand, London WC2R 2LS, United Kingdom
Phone: +44-(0)20-78482431, Fax: +44-(0)20-78482932
email: k.althoefer@kcl.ac.uk

ABSTRACT: The motion planning and control problem is a well-known problem in the field of robotics. The objective is to find collision-free trajectories for a robot, in static or dynamic environments containing some obstacles, between a start and a goal configuration. It has attracted much research in recent years. In this context the term control has a broad meaning that includes many different controls, such as low-level motor control, and behaviour control, where behaviour represents many complicated tasks, like obstacle avoidance and goal seeking. This paper describes an intelligent navigation system for omnidirectional mobile robots based on fuzzy logic. Owing to its simplicity and hence its short response time, the fuzzy navigator is especially suitable for on-line applications with strong real-time requirements. On-line planning is an on-going activity. The planner receives a continuous flow of information about occurring events and generates new commands in response to the incoming events, while previously planned motions are being executed. The fuzzy rule-base of the proposed system combines the repelling influence, which is related to the distance and the angle between the robot and nearby obstacles, with the attracting influence produced by the angular difference between the actual direction and position of the robot and the final configuration, to generate a new actuating command for the mobile platform. It can be considered as an on-line local navigation method for omnidirectional mobile robots for the generation of instantaneous collision-free motions. This reactive system is especially suitable for real-time applications. The use of fuzzy logic leads to a transparent system which can be tuned by hand or by a set of learning rules. Furthermore, this approach allows obstacle avoidance in dynamic environments. The navigation system is being currently extended into an adaptive system using a neural network training algorithm for the rule-base. The presented approach to the obstacle avoidance problem has been influenced mainly by Khatib's artificial potential fields method and the subsumption architecture developed by Brooks. The functioning of the fuzzy navigator with respect to omnidirectional mobile robots and results of simulated experiments are presented.

KEYWORDS: On-line motion planning, obstacle avoidance, local navigation, fuzzy logic, omnidirectional mobile robots.

INTRODUCTION

The goal of autonomous mobile robotics is to build physical systems that can move purposefully and without human intervention in unmodified environments - that is, in real-world environments that have not been specifically engineered for the robot. The development of techniques for autonomous robot navigation constitutes one of the major trends in the current research in robotics (Tzafestas (1999)). This trend is motivated by the current gap between the available technology and the new application demands. On the one hand, current industrial robots have low flexibility and autonomy: typically, these robots perform pre-programmed sequences of operations in highly constrained environments, and are not able to operate in new environments or to face unexpected situations. On the other hand, there is a clear emerging market for truly autonomous robots. Possible applications include intelligent service robots for offices, hospitals, and factory floors; maintenance robots operating in hazardous or hardly accessible areas; domestic robots for cleaning or entertainment and semi-autonomous vehicles for help to disabled people.

Despite the impressive advances in the field of autonomous robotics in recent years, a number of problems remain. Most of the difficulties originate in the nature of real-world, unstructured environments, and in the large uncertainties that are inherent to these environments. First, prior knowledge about the environment is, in general, incomplete, uncertain, and approximate. For example, maps typically omit some details and temporary features, spatial relations between objects may have changed since the map was built, and the metric information may be imprecise and inaccurate. Second, perceptually acquired information is usually unreliable. The limited range, combined with the effect of environmental features (e.g., occlusion) and of adverse observation conditions (e.g., poor lighting), leads to noisy and imprecise data; and errors in the measurement interpretation process may lead to incorrect beliefs. Third, real-world environments typically have complex and unpredictable dynamics: objects can move, other agents can modify the environment, and relatively stable features may change with time. Finally, the effect of control actions is not completely reliable, e.g. the wheels of a mobile robot may slip resulting in accumulated odometric errors.

ROBOT MOTION PLANNING : A BRIEF SURVEY

To be useful in the real world, robots need to move safely in unstructured environments and achieve their given goals despite unexpected changes in their surroundings. The environments of real robots are rarely predictable or perfectly known so it does not make sense to make precise plans before moving. The robot navigation problem can be decomposed into the following two problems (Ratering (1995)):

- *Getting to the goal*. This is a global problem because short paths to the goal generally cannot be found using only local information. The topology of the space is important in finding good routes to the goal.
- *Avoiding obstacles*. This can often be solved using only local information, but for an unpredictable environment it cannot be solved in advance because the robot needs to sense the obstacles before it can be expected to avoid them.

In previous research, robot collision avoidance has been a component of high level controls in hierarchical robot systems. Collision avoidance has been treated as a planning problem, and research in this area was focused on the development of collision-free path planning algorithms. These algorithms aim at providing the low-level control with a path that will enable the robot to accomplish its assigned task free from any risk of collision. However, this places limits on the robot’s real-time capabilities for precise, fast, and highly interactive operations in a cluttered and evolving environment. Collision avoidance at the low-level control is not intended to replace high-level functions or to solve planning problems. The purpose is to make better use of low-level control capabilities in performing real-time operations.

A number of different architectures for autonomous robot navigation have been proposed in the last twenty years (Tzafestas (1999)). These include hierarchical architectures that partition the robot's functionalities into high-level (model and plan) and low-level (sense and execute) layers; behaviour-based architectures that achieve complex behaviour by combining several simple behaviour-producing units; and hybrid architectures that combine a layered organisation with a behaviour-based decomposition of the execution layer (see e.g. Topalov and Tzafestas (2000)). While the use of hybrid architectures is gaining increasing consensus in the field, a number of technological gaps still remain.

Some researchers have solved the navigation problem by solving these two sub-problems one after the other. A path is first found from the robot's initial position to the goal and then the robot approximates this path as it avoids obstacles. This method is restrictive in that the robot is required to stay fairly close to or perhaps on a given path. This would not work well if the path goes through a passageway which turns out to be blocked by an unforeseen obstacle. Solutions that are only local or reactive (Brooks (1986)) can lead the robot into local minima traps. Solutions that assume a priori knowledge of the position of the obstacles (e.g. Erdmann (1987), Fugimura (1991)), or select a path using only information on stationary obstacles, and determine the speed of the robot while following the path (e.g. Griswold (1990)), or solutions that require the robot to stay within some distance from its assigned path while avoiding unknown moving obstacles (e.g. Gil de Lamadrid (1990)), are not always sufficiently flexible to deal with situations in which an obstacle blocks a path to the goal.

In the general case, knowledge of the environment is partial and approximate; sensing is noisy; the dynamics of the environment can only be partially predicted; and robot's hardware execution is not completely reliable. Though, the robot needs to take decisions and execute actions at the time-scale of the environment. Classical planning approaches have been criticised for not being able to adequately cope with this situation, and a number of reactive approaches to robot control have been proposed (e.g. Firby (1987), Kaufman (1987), Gat (1991)), including the use of fuzzy control techniques (e.g., Sugeno (1985b), Yen (1992)). Reactivity provides immediate response to unpredicted environmental situations by utilizing the idea of reasoning about future consequences of actions. Reasoning about future consequences (sometimes called "strategic planning"), however, is still needed in order to intelligently solve complex tasks.

OBSTACLE AVOIDANCE AND NAVIGATION STRATEGY

In developing our strategy we were influenced by Khatib's obstacle avoidance method, which is based on an *artificial potential field* (Khatib (1986)). Khatib computes an artificial potential field that has a strong repelling force in the vicinity of obstacles and an attracting force produced by the target location. The superposition of the two forces creates a potential field, which incorporates information about the environment. Following the steepest gradient from a start position, a path can be found that guides the robot to the target position avoiding obstacles (Khatib (1986)). In our approach the amount of computation, that is required, is reduced by using only the nearest obstacles to determine the direction of motion. The system is also related to the subsumption architecture developed by Brooks (Brooks (1986)). The main idea of his work is a collection of modules, which are interconnected on different layers with different hierarchies. These modules are for example *wall following*, *obstacle avoidance*, *goal reaching*, etc. Depending on sensory input, a module becomes active and generates a command for the robot (Brooks (1986)). While Brooks' system resembles an expert system where for any input signal one specific reaction module or a specific combination of modules is active, the fuzzy approach is a parallel processing approach where each input contributes to the final decision.

The technique is based on a fuzzy-based obstacle avoidance unit which controls the mobile robot. This unit has three principal inputs:

1) the distance between the robot and the nearest obstacle d_j,
2) the angle between the robot and the nearest obstacle γ_j , and
3) the angle between the robot's direction and the straight line connecting the current position of the robot and the goal configuration $\theta_j = \alpha_j - \beta_j$, where β_j is the angular difference between the straight line connecting the robot's current position and the goal configuration, and α_j is the current direction of the robot (see Figure 1).

The output variable of the unit is the motor command τ_j. All these variables can be positive or negative, i.e. they do not only inform about the magnitude, but also about the sign of displacement relative to the robot - left or right. The motor command which can be interpreted as an actuation for the robot's direction motors is fed to the mobile platform at each iteration. It is assumed that the robot is moving with constant velocity, and no attempt is being made to control it.

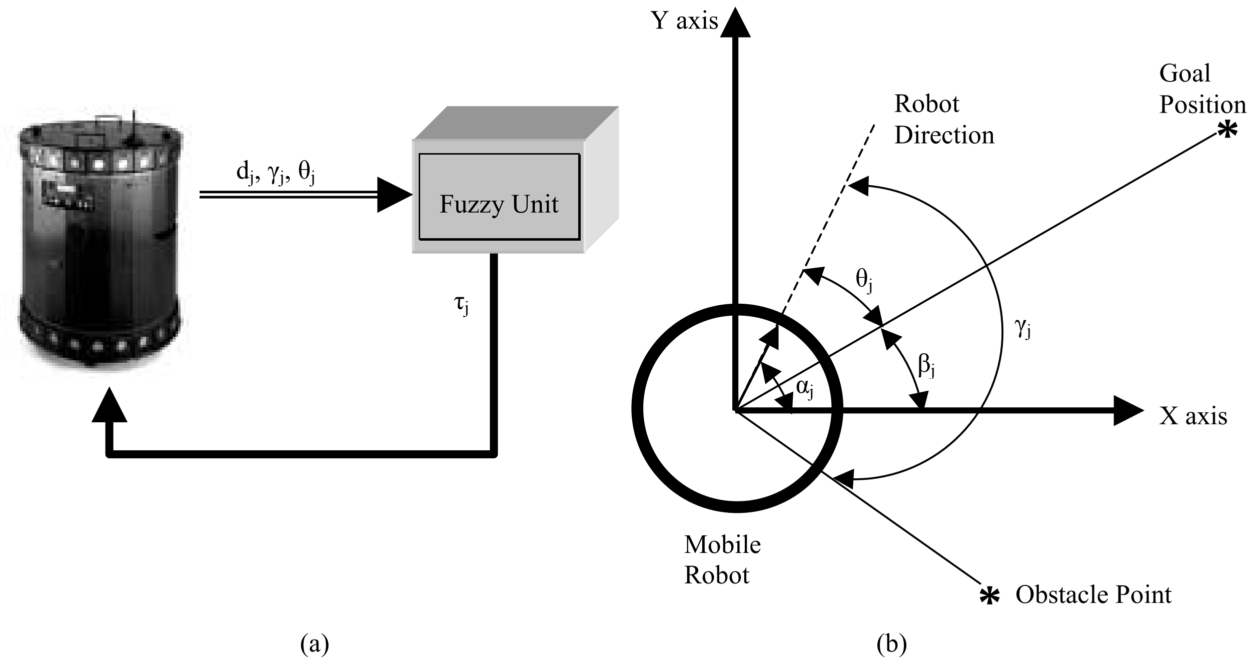

Figure 1 : (a) The Nomad XR4000 industrial mobile robot connected to the corresponding fuzzy-based obstacle avoidance unit. This unit receives via an input the angle $\theta_j = \alpha_j - \beta_j$ between the robot's direction and the straight line connecting the current position of the robot and the goal configuration, and the distance and the angle of the nearest obstacle (d_j, γ_j) (see (b)). If no obstacle is detected inside the scan area, the fuzzy unit is informed of an obstacle in the far distance The output variable of the unit is the motor command τ_j.

For the calculation of the distance, the only obstacles considered are those which fall into a bounded area surrounding the robot and moving along with it. In this implementation, this area is chosen to be a cylindrical volume around the mobile platform and reaches up to a predefined horizon. This area can be seen as a simplified model for the space

scanned by ranging sensors (for example ultra-sonic sensors) attached to the sides of the robot (Pedrycz (1995)). Besides an input from ultrasonic sensors, a camera can also be used to acquire the environment. Mobile robots are usually equipped with a pan/tilt platform where a camera is mounted. This camera can also be utilised (Jaitly (1996), Jaitly (1996b), Perdycz (1995)). If no obstacle is detected inside the scan area, the fuzzy unit is informed of an obstacle in the far distance.
The task of the fuzzy unit is to provide a control function, which produces an appropriate motor command from the given inputs. In broad lines, the control function can be described as follows: on the one hand the function has to lead the mobile robot to its attracting goal-position; on the other hand it has to force the robot to back up when approaching an obstacle which conveys a repelling influence. The fuzzy-rule-base (which represents the control function in the fuzzy unit) is built up by using common sense rules or by a neural network training algorithm (this is currently under investigation by the authors (Tzafestas, Zavlangas (1999))). An alternative fuzzy motion control scheme of mobile robots which employs the sliding-mode control principle can be found in Rigatos et. al (2000).

THE PROPOSED FUZZY-ALGORITHM

The fuzzy controller in our approach is based on the fuzzy control principles developed by Sugeno (for example see Sugeno (1984), Sugeno (1985)). Each input space is partitioned by fuzzy sets as shown in Figure 2. In the literature, a variety of functions can be found which are employed to represent fuzzy sets (for example see Kosko (1992)). Here, asymmetrical triangular and trapezoidal functions which allow a fast computation, essential under real-time conditions, are utilized to describe each fuzzy set (see Eqs. (1), (2) and (3)) below. The fuzzy sets of the three inputs d_j, γ_j and θ_j are depicted in Figure 2. To calculate the fuzzy intersection, the product operator is employed (see Eq. (5)). The final output of the unit is given by a weighted average over all rules (see Eq. (6) and Figure 2).

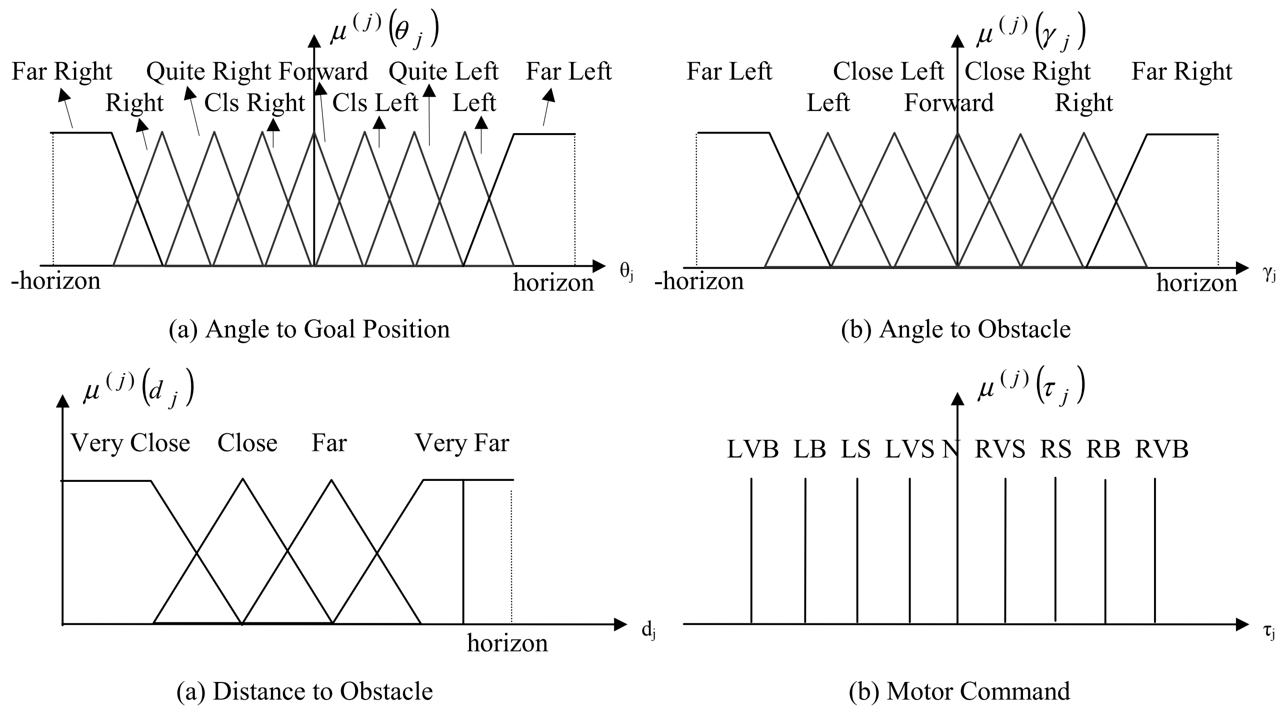

Figure 2 : Fuzzy sets for the mobile robot : (a) angle between robot and goal position, (b) angle between robot and obstacle, (c) distance to obstacle, and (d) motor command. Note that the output is not partitioned into fuzzy sets, but consists of crisp values.

Intuitively, the rules for obstacle-avoiding navigation can be written as sentences with three antecedents and one conclusion. This structure lends itself to a tabular representation such as the one shown in Table 1. This table represents the prior knowledge of the problem domain. The tools of fuzzy logic allow us to translate this intuitive knowledge into a control system.
In fuzzy logic terms, the range of a variable is called the universe of discourse. To translate Table 1 into fuzzy logic, the universe of discourse D_j which describes the distance $d_j \in D_j$ to the obstacle is partitioned by fuzzy sets $\mu_1^{(j)},\dots,\mu_{p_j}^{(j)}$, where p_j is the number of fuzzy sets. Each set $\mu_{\tilde{p}_j}^{(j)}$, $\tilde{p}_j = 1,\dots,p_j$, represents a mapping $\mu_{\tilde{p}_j}^{(j)}(d_j): D_j \to [0,1]$ by which

d_j is associated with e.g. a number in the interval [0,1] indicating to what degree d_j is a member of the fuzzy set. Since d_j is a measure of distance, "*very_close*", it may be considered as a particular fuzzy value of the variable distance and each d_j is assigned a number : $\mu_{very\_close}(d_j) \in [0,1]$ which indicates the extent to which this particular d_j is considered to be *"very_close"* (Mamdani (1981). Similarly, fuzzy sets $\nu_1^{(j)},\ldots,\nu_{q_j}^{(j)}$ can be defined over the universe of discourse Θ_j which represents the distance to the goal : $\theta_j \in \Theta_j$. Finally, fuzzy sets $u_1^{(j)},\ldots,u_{g_j}^{(j)}$ can be defined over the universe of discourse Γ_j that represents the angle to the nearest obstacle $\gamma_j \in \Gamma_j$. In contrast to the Mamdani controller, Sugeno's controller (see Sugeno (1984), Sugeno (1985)), of which ours is an example, has an output set which is not partitioned into fuzzy sets (see also Figure 2). Thus, the rule conclusions merely consist of scalars $\tau_{\tilde{r}_j}, \tilde{r}_j = 1,\ldots,r_j$.

Table 1: A rule-base for the mobile robot when γ_j is *"far_left"*. This rule-base is a translation of the common-sense knowledge of the problem domain into the language of fuzzy logic. Rows represent the fuzzy measures of the distance to an obstacle, while columns are fuzzy representations of the angle to the goal. Each element of the table can be interpreted as a particular motor actuation command.

$\nu_j \setminus \mu_j$	very close	close	far	very far
far right	right big	right small	left very big	left very big
right	right big	right small	left big	left big
quite right	right big	right small	left small	left small
close right	right big	right small	left very small	left very small
forward	null	null	null	null
close left	right very small	right very small	right very small	right very small
quite left	right small	right small	right small	right small
left	right big	right big	right big	right big
far left	right very big	right very big	right very big	right very big

The fuzzy sets $\mu_{\tilde{p}_j}^{(j)}, \tilde{p}_j = 1,\ldots,p_j$, are described by asymmetrical triangular and trapezoidal functions. Defining the parameters, $ml_{\tilde{p}_j}^{(j)}$ and $mr_{\tilde{p}_j}^{(j)}$ as the *x*-co-ordinates of the left and right zero crossing respectively, and $mcl_{\tilde{p}_j}^{(j)}$ and $mcr_{\tilde{p}_j}^{(j)}$ as the *x*-co-ordinates of the left and right side of the trapezoid's plateau, the trapezoidal functions can be written as:

$$\mu_{\tilde{p}_j}^{(j)}(d_j) = \begin{cases} \max\left(\left(d_j - ml_{\tilde{p}_j}^{(j)}\right) \Big/ \left(mcl_{\tilde{p}_j}^{(j)} - ml_{\tilde{p}_j}^{(j)}\right)\ ,\ 0\right) & \text{if } d_j < mcl_{\tilde{p}_j}^{(j)} \\ 1 & \text{if } mcl_{\tilde{p}_j}^{(j)} \le d_j \le mcr_{\tilde{p}_j}^{(j)} \\ \max\left(\left(d_j - mr_{\tilde{p}_j}^{(j)}\right) \Big/ \left(mcr_{\tilde{p}_j}^{(j)} - mr_{\tilde{p}_j}^{(j)}\right)\ ,\ 0\right) & \text{if } d_j > mcr_{\tilde{p}_j}^{(j)} \end{cases} , \tag{1}$$

with $\tilde{p}_j = 1,\ldots,p_j$. Triangular functions can be achieved by setting $mcl_{\tilde{p}_j}^{(j)} = mcr_{\tilde{p}_j}^{(j)}$.

At the left and right side of the interval the functions are continued as constant values of magnitude one:

$$\mu_1^{(j)}(d_j) = \begin{cases} 1 & \text{if } d_j \le mcr_1^{(j)} \\ \max\left(\left(d_j - mr_1^{(j)}\right) \Big/ \left(mcr_1^{(j)} - mr_1^{(j)}\right)\ ,\ 0\right) & \text{if } d_j > mcr_1^{(j)} \end{cases} \tag{2}$$

and

$$\mu_{p_j}^{(j)}(d_j) = \begin{cases} \max\left(\left(d_j - ml_{p_j}^{(j)}\right) \Big/ \left(mcl_{p_j}^{(j)} - ml_{p_j}^{(j)}\right), 0\right) & \text{if } d_j \le mcl_{p_j}^{(j)} \\ 1 & \text{if } d_j > mcl_{p_j}^{(j)} \end{cases} . \tag{3}$$

The fuzzy sets for θ_j and γ_j are defined analogously. Figure 2 shows the fuzzy sets $\mu_1^{(j)},\ldots,\mu_{p_j}^{(j)}$, $\nu_1^{(j)},\ldots,\nu_{q_j}^{(j)}$ and $u_1^{(j)},\ldots,u_{g_j}^{(j)}$.

Every fuzzy set, $\mu_{\tilde{q}_j}^{(j)}$, $\nu_{\tilde{q}_j}^{(j)}$ and $u_{\tilde{g}_j}^{(j)}$, is associated with linguistic terms $A_{\tilde{q}_j}^{(j)}$, $B_{\tilde{q}_j}^{(j)}$ and $C_{\tilde{g}_j}^{(j)}$ respectively. Thus, for link l_j the linguistic control rules $R_1^{(j)},\ldots,R_{r_j}^{(j)}$, which constitute the rule base, can be defined as:

$$R_{\tilde{r}_j}^{(j)}: \textbf{IF } d_j \textbf{ is } A_{\tilde{p}_j}^{(j)} \textbf{ AND } \theta_j \textbf{ is } B_{\tilde{q}_j}^{(j)} \textbf{ AND } \gamma_j \textbf{ is } C_{\tilde{g}_j}^{(j)} \textbf{ THEN } f(\tau_{\tilde{r}_j}) \quad (\tilde{r}_j = 1,\ldots,r_j) \tag{4}$$

where the **AND** operations use the *t*-norm product operator:

$$\sigma_{\tilde{r}_j} = \mu_{\tilde{p}_j,\tilde{r}_j}^{(j)}(d_j) \cap \nu_{\tilde{q}_j,\tilde{r}_j}^{(j)}(\theta_j) \cap u_{\tilde{g}_j,\tilde{r}_j}^{(j)}(\gamma_j) = \mu_{\tilde{p}_j,\tilde{r}_j}^{(j)}(d_j) * \nu_{\tilde{q}_j,\tilde{r}_j}^{(j)}(\theta_j) * u_{\tilde{g}_j,\tilde{r}_j}^{(j)}(\gamma_j). \tag{5}$$

Finally, the output of the unit is given by a weighted average over all rules (see Figure 2 and Nauck (1994)):

$$\tau_j = \frac{\sum_{\tilde{r}_j=1}^{r_j} \sigma_{\tilde{r}_j} \cdot \tau_{\tilde{r}_j}}{\sum_{\tilde{r}_j=1}^{r_j} \sigma_{\tilde{r}_j}} . \tag{6}$$

Eq. (4) together with Eqs. (5) and (6) define how to translate the intuitive knowledge reflected in Table 1 into a fuzzy rule-base. The details of this translation can be modified by changing the number of fuzzy sets, the shape of the sets (by choosing the parameters, $ml_{\tilde{p}_j}^{(j)}$ and $mr_{\tilde{p}_j}^{(j)}$, $mcl_{\tilde{p}_j}^{(j)}$, $mcr_{\tilde{p}_j}^{(j)}$) as well as the value $\tau_{\tilde{r}_j}$ of each of the rules in Eq. (6). As an example, the control rules for the particular mobile robot are shown in Table 1. In this application, the number of fuzzy sets which fuzzify the obstacle distance d_j, the angle to the goal θ_j and the angle to the nearest obstacle γ_j are chosen to be four, nine and seven, respectively. All the other parameters were refined by trial and error.

EXPERIMENTAL RESULTS

To test the functionality of the basic algorithm, the fuzzy navigator was applied to a simulated omnidirectional mobile robot (Nomad XR4000). All experiments were carried out on a Personal Computer (Pentium II, 350MHz). For the development of the fuzzy navigator and for the visualisation and animation of the robot's path MatLab 5.2 software package was used.
The performance of the fuzzy navigator for the Nomad XR4000 industrial mobile robot was tested in a variety of environments, with different obstacle constellations and working scenarios, even in dynamic environments with moving obstacles. In all cases, the fuzzy navigator provided the system with a collision free motion. Simulation results obtained in three different working scenarios (with different obstacle constellations) are presented in Figure 3.

CONCLUSIONS

In this paper, a fuzzy-based navigator has been proposed for the obstacle avoidance and navigation problem in omnidirectional mobile robots. It has been tested in a variety of working scenarios with different obstacle constellations, both static and dynamic, providing each time a collision-free trajectory for the robotic platform. This local planning method has shown a robust and stable performance and the experimental results were very satisfactory.
The difference between our approach and the existing obstacle avoidance techniques is that the proposed navigator considers only the nearest obstacle to decide upon the manipulator's next move. Clearly, this clearly leads to a large reduction in the required remote sensing and computations. A drawback is that, this reduction in information about the robot's environment, leads to an increased possibility of getting trapped to local minima. There may be routes leading to the goal that avoid the obstacles, which our navigator will not be able to find.
Global methods consider the whole workspace at each iteration and compute an optimal obstacle avoiding path to the goal. These techniques work well in simple workspaces; however, the cost of calculating global trajectories in densely populated or dynamic environments, is enormous.
The proposed fuzzy navigator is a fast local method that can perform in real time, providing the system with a collision-free trajectory. However, the problem of local minima cannot be ignored. Future work will aim to the design of a hierarchical structure with higher planning levels (e.g. global path planning modules), which will be able to detect these obstacle constellations that lead to local minima and provide the system with an escaping manoeuvre. Also future research should consider the application of neural learning mechanisms to the fuzzy navigator providing an adaptive (neurofuzzy) planner (Althoefer (1996), Wang (1994)).

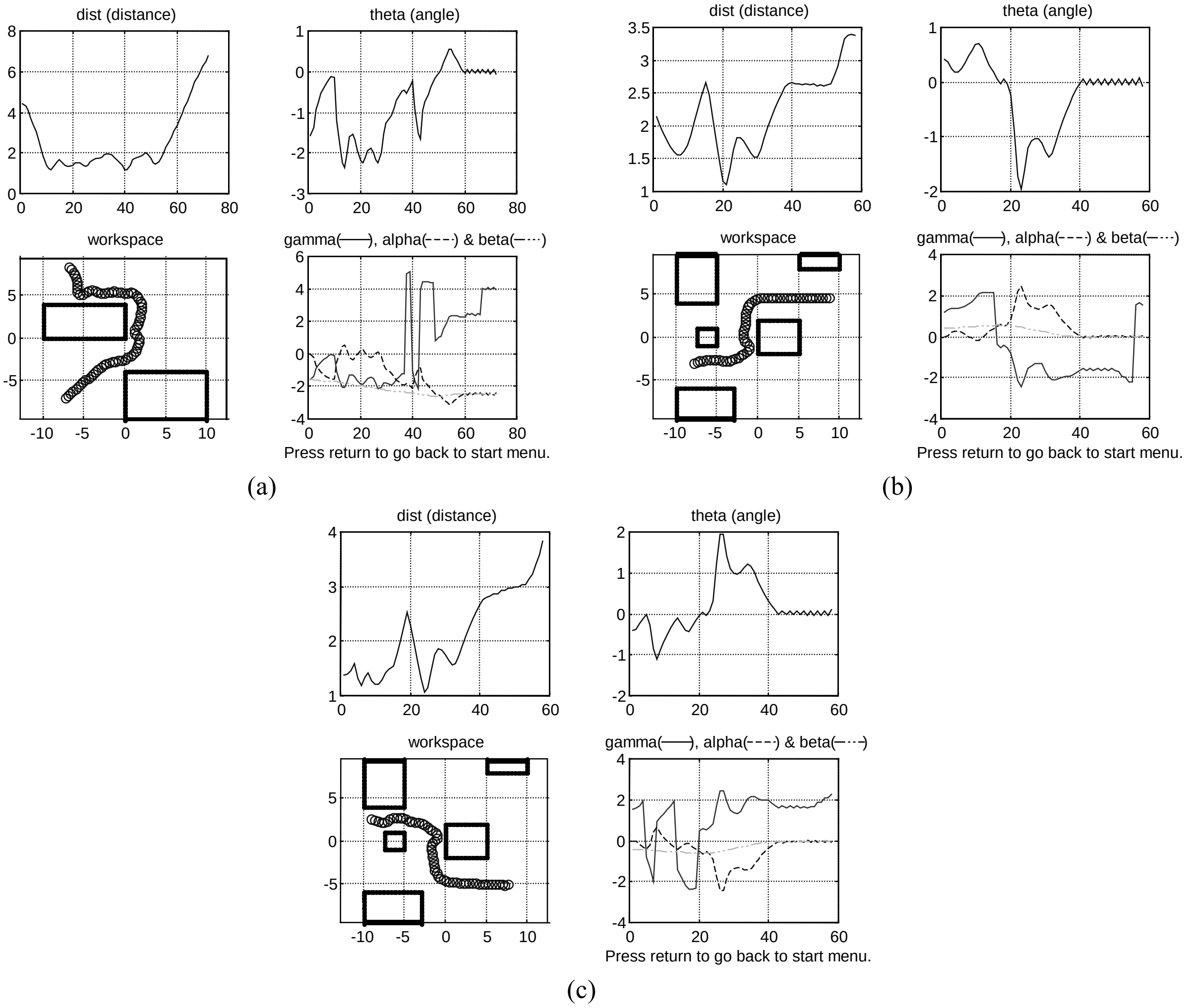

Figure 3 : Simulation results using the Nomad XR4000 industrial mobile robot. The proposed fuzzy navigator was successfully tested for the navigation/obstacle avoidance problem in three working scenarios with different obstacle constellations (a), (b) and (c), each time providing the mobile robot with a collision-free trajectory.

ACKNOWLEDGEMENT

The research carried out by P. G. Zavlangas and S. G. Tzafestas was supported by the European Union and the Greek Secretariat for Research and Technology (Project IGIOROBOT – PENED – 99ED623).

REFERENCES

Althoefer, K., 1996, „Neuro-Fuzzy Path Planning for Robotic Manipulators", PhD Thesis, King's College, London, U.K.

Bison, P.; Chenello, G.; Sossai, C.; Trainito, G., 1997, „A Syntactical Approach to Data Fusion", European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty, pp. 58 – 70.

Brooks, R. A., 1986, „A robust Layered Control System for a Mobile Robot", IEEE Journal of Robotics and Automation, RA-2(1), pp. 14 – 23.

Erdmann, M.; Lozano-Perez, T., 1987, „On Multiple Moving Obstacles", Algorithmica 2(4), pp. 477 – 521.

Firby, J. R., 1987, „An Investigation into Reactive Planning in Complex Domains", AAAI Conference.

Fugimura, Kikuo, 1991, „Motion Planning in Dynamic Environments", Springer Verlang, Tokyo, Japan.

Gat, E., 1991, „Reliable Goal-Directed Reactive Control for Real-World Autonomous Mobile Robots", PhD Dissertation, Virginia Polytechnic Institute and state University.

Gil de Lamadrid, J.; Gini, Maria, 1990, „Path Tracking through Uncharted Moving Obstacles", IEEE Transactions on System, Man, and Cybernetics, 20(6), pp. 1408 – 1422.

Griswold, N. C.; Elan, J., 1990, „Control of Mobile Robots in the Presence of Moving Obstacles", IEEE Transactions on Robotics and Automation, 6(2).

Hohle, U., 1997, „A Survey on the Fundamentals of Fuzzy Set Theory", Handbook of Mechanical Engineering, CRC Press, Boca Raton/FL, USA.

Jaitly, R.; Althoefer, K.; Fraser, D., 1996, „From Vision to Path Planning : A Neural-based Implementation", 2nd International Conference on Engineering Applications of Neural Networks, pp. 209 – 212, London, U.K.

Jaitly, R.; Fraser, D., 1996b, „Automated 3D Object Recognition and Library Entry System", Neural Network World 6(2), pp. 173 – 183.

Kaufman, Morgen, 1987, „An Architecture for Intelligent Reactive Systems", In Georgeff, M. P., 1987, „Reasoning about Actions and Plans".

Khatib, O., 1986, „Real-Time Obstacle Avoidance for Manipulators and Mobile Robots", The International Journal of Robotics Research, 5(1), pp. 90.

Kosko, B., 1992, „Neural Networks and Fuzzy Systems", Prentice-Hall Int. Editions.

Mamdani, E. H.; Assilian, S., 1981, „An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller", Fuzzy Reasoning and Applications, Academic Press, pp. 311.

Nauck, D.; Klawonn, F.; Kruse, R., 1994, „Neuronale Netze und Fuzzy Systeme", Vieweg, Braunschweig/Wiesbaden, Germany.

Pedrycz, W., 1995, „Fuzzy Sets Engineering", CRC Press, Boca Raton/FL, USA.

Ratering, Steven; Gini, Maria, 1995, „Robot Navigation in a Known Environment with Unknown Obstacles", Autonomous Robots, pp. 149 – 165.

Rigatos, G. G.; Tzafestas, C. S.; Tzafestas, S. G., 2000, „Mobile Robot Motion Control in Partially Unknown Environments Using a Sliding-Mode Fuzzy Logic Controller", Robotics and Autonomous Systems, In Press.

Sugeno, M.; Murakami, K., 1984, „Fuzzy Parking Control of Model Car", 23rd IEEE Conference on Decision and Control, pp. 902, Las Vegas, USA.

Sugeno, M, 1985, „An Introductory Survey of Fuzzy Logic", Information Science, 36, 59.

Sugeno, M.; Nishida, M., 1985b, „Fuzzy Control of a Model Car", Fuzzy Sets and Systems 16, pp. 103 – 113.

Topalov, A. V.; Tzafestas, S. G., 2000, „Layered Multi-Agent Reactive Behaviour Learning in a Robotic Soccer", Proc. SYROCO'2000 : 6th IFAC Symp. on Robot Control, Vienna Univ. of Technology, Vienna, Austria.

Tzafestas, S. G. (ed.), 1999, „Advances in Intelligent Autonomous Systems", Kluwer, Dordrecht/Boston.

Tzafestas, S. G.; Zavlangas, P. G., 1999, „Adaptive Neuro-Fuzzy Navigation for Industrial Manipulators", Proc. 1999, ICIMS-NOE Advanced Summer Institute : Production Management, Control and Supervision (ASI '99), Leuven, Belgium.

Yen, J.; Pflunger, N., 1992, „A Fuzzy Logic Based Robot Navigation System", AAAI Fall Symposium on Mobile Robot Navigation, pp. 195 – 199, Boston/MA, USA.

Wang, Li Xin, 1994, „Adaptive Fuzzy Sets and Control : Design and Stability Analysis", Prentice Hall Int.

Zadeh, L. A., 1994, „Why the Success of Fuzzy Logic is not Paradoxical", IEEE Expert, pp. 43 – 46.