REAL TIME NAVEGATION OF AUTONOMOUS ROBOT WITH FUZZY CONTROLLER AND ULTRASONIC SENSORS

Oscar Calvo
LEICI, Facultad de Ingeniería,
Universidad Nacional de la
Plata, CICpBA, Argentina
Universitat de les Illes Balears
oscar@hp1.uib.es

Gonzalo Rodríguez
LEICI, Facultad de Ingeniería,
Universidad Nacional de la
Plata, CICpBA, Argentina

Rodrigo Picos
Universitat de les Illes Balears,
Departament de Física
Illes Balears, España
rodrigo@ribeiro.uib.es

Abstract

This paper shows a strategy suitable for navigating autonomous robots in a completely unknown environment. The method proposed combines optimum path planning techniques with fuzzy logic to avoid obstacles and to determine the shortest path towards its goal. The technique computes the potential surface using Dijkstra’s algorithm in a moving window, updating the cost map as it moves with the information obtained by the ultrasonic sensors. A Fuzzy Logic Controller (FLC) controls the wheels of a differential drive robot to the angle of minimum potential. This ensures a smooth trajectory towards the objective. A second FLC controls the average speed of the platform.

Keywords: Fuzzy Control, Autonomous Guided Vehicles, Autonomous Navigation; Robotics.

1 INTRODUCTION

Autonomous operation of mobile robots in real environments presents serious difficulties to classical control methods. Usually, the environment is poorly known, sensor readings are noisy and vehicle dynamics is fairly complex and non-linear. Robots reacts with the environment making decisions on the fly to avoid obstacles and to follow planned trajectories. AI techniques give the robot a certain degree of reasoning. Fuzzy logic has contributed to enhance the perception of the environment, providing tools that handle situations with some amount of vagueness. Sugeno already used this techniques back in the eighties to control a model car [10]. The vehicle used ultrasonic sensors to measure the distance to obstacles and, with fuzzy rules, it was able to navigate and park itself. His method wasn't goal oriented but merely follows obstacle avoidance rules. Navigation techniques nowadays essentially follow two main strategies depending on the information available about the environment. When the map is known, a safe path is calculated beforehand minimizing some cost. Dijkstra’s algorithm uses cost functions to build a potential map and the shortest path is obtained [5], [7], [4]. When the information about the environment is scarce, reactive techniques are used for real time navigation. Problems such as imprecision due to multiple sensors is solved by using uncertainty grids [6] [8]. Information from different sensors is fused by statistical methods such as extended Kalman filters.

Sometimes, position and characteristics of the objects are known beforehand and maps of the environment are obtained off-line and used by the robot to navigate safely. But, when the robot navigates in a completely unknown environment, these maps are not available and they have to be built on real time as the robot moves along. The vehicle must be equipped with sensors to recognize the obstacles. Among the many techniques available today to recognize the environment, the most popular ones are based on video cameras, infrared sensors, laser finders, tactile sensors and ultrasonic detectors [3]. Once the map is obtained, a trajectory can be built to reach the objective minimizing some cost variable (i.e. fuel, time, risk, etc) avoiding obstacles in the path. The vehicle follows the prescribed path if it is capable of position itself. Again, numerous techniques exist to determine the position of the mobile at any time: visual or ultrasonic correlation of perceived world with stored features or “Landmarks” (map matching), triangulation by RF beacons, GPS navigation or dead reckoning.

This paper present a mobile robot built entirely in our Laboratory, which navigates in an unknown environment. It uses ultrasonic sensors to build a map of obstacles, finds the best trajectory towards the goal, estimating its position by dead reckoning. The technique presented here determines the shortest path building the cost map and hence potential surface, based on the information obtained with ultrasonic sensors. Navigation is performed within a local map, continuously updating the cost matrix and the potential surface.

Direction of the mobile is based on minimum potential ahead.

2 DIJKSTRA’S ALGORITHM AND THE OPTIMUM PATH

Uniform cost algorithm proposed by Dijkstra is a sequential method that can efficiently compute the optimum path from a reference node to any arbitrary node for a given cost map.

Suppose we define a bi-dimensional grid with PxQ nodes. Node (i,j) has connections with its M closest neighbors, represented by the set Π_{ij} (in our case M=8) (Figure 1). The cost map defines the transitional cost between neighbors and is given by:

$$C_{[(i,j),(k,l)]} = c_{kl} \quad \text{for } (k, l)\ \varepsilon\ \Pi_{ij}$$

where $c_{kl} > 0$ is the local transitional cost between node(k,l) and node (i,j). Next, a Potential surface is built based on the previous cost map. The procedure to build it is the following: Initially, the potential of the reference node (i’,j’) is fixed to zero and the potential of the rest of the nodes is set to a large value V_{max}. Later, beginning with the reference node, all the (i,j) nodes are updated with the following formula:

$$V_{(i,j)} = \min_{(k,l)\hat{\imath}\,P\ ij} [V_{(i,j)} , V_{(k,l)} + c_{kl}]$$

This process is repeated until the potential surfaces converges to a stable value. Each point obtained $V_{(i,j)}$ in (i,j) represents a potential measured w.r.t a ground node (i’,j’), and is a measure of the minimum cost to travel from the ground node (i’,j’) to node (i,j).
For the case of a reference node A and a goal node B, the algorithm can be extended to obtain the optimum path between those nodes with the following: First, node A is assigned as the ground node (i’, j’) and the previous algorithm is applied yielding the potential surface $V_{A.}$ This process is repeated using B as the ground node. Then, a combined potential surface $V = V_A+V_B$ is obtained. An interesting property of V is that the trajectory with minimum cost connecting node A with node B is obtained by the nodes with minimum potential $V_{min.}$

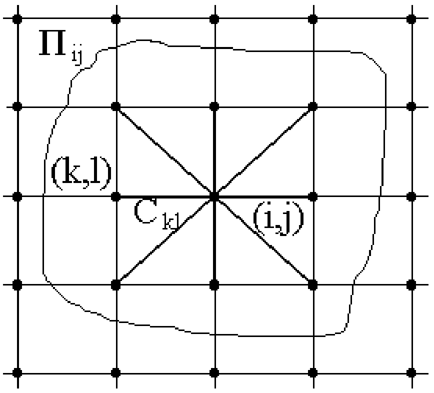

Figure 1: Connections with eight adjacent neighbors.

3 NAVIGATING WITH OBSTACLES

We are going to apply Dijkstra’s algorithm to compute the shortest path from the actual position of an AGV to a target point. If we know where the objects are, we can obtain this path by applying Dijkstra’s algorithm to all the points in the map. We build a cost map with low cost values for transitions in the free space and high cost values in the places occupied by an object. We assign a constant value c_k for horizontal or vertical transitions, and $\sqrt{2}\,c_k$ for transition in diagonal. Simulations were performed on a 30x40 grid. To lower the risk of collision, the size of the objects can be artificially augmented, setting a high cost in the neighboring nodes of the objects. To get a smoother trajectory and to model the noisy nature of the sensors, Gaussian bells can substitute the abrupt cost functions. First, a cost map representing the transition between nodes in the free space is set and on the cells occupied by objects a bell, centered in that node is drawn. This modifies the cost in that cell and also in its neighbors. Applying a MAX function to the costs of all the cells, the objects will be represented by plateau surrounded with smooth walls. Applying this concept to our previous case, the resulting cost map obtained is shown in figure 2 and the potential surface is shown in fig. 3. The same surface is represented by contour lines in figure 4. The resulting trajectory is also shown in the same figure. In this case the traveled distance is 233 units.

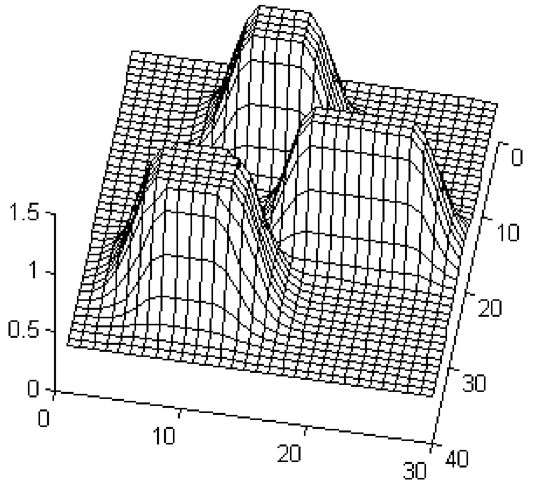

Figure 2: Cost map with Gauss bells centered on the obstacles.

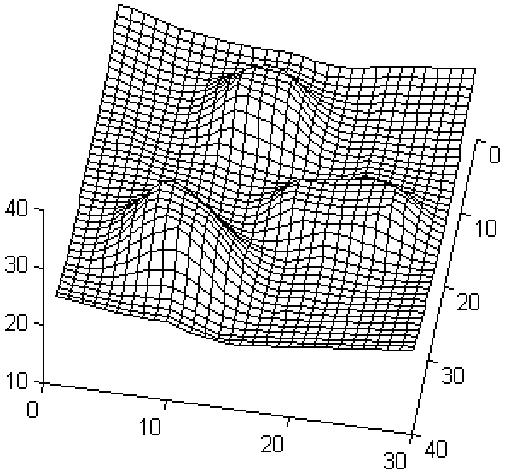

Figure 3: Potential map with Gauss bells centered on the obstacles

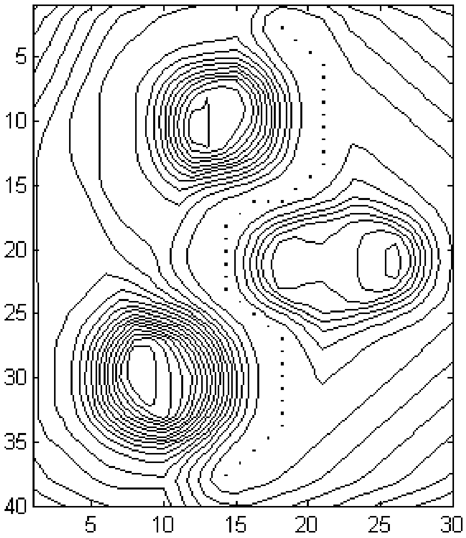

Figure 4: Contour lines showing the optimum trajectory

4 REAL TIME NAVIGATION IN AN UNKNOWN ENVIRONMENT

When we want to navigate in an unknown environment, the cost map and the potential surface are not available beforehand and they have to be built as the AGV moves towards the target. In this case, a partial goal is set and the algorithm is computed within a window using a local map. The information about the objects in the vicinity of the mobile platform is obtained with sensors. To minimize calculations, a 20 x 20 cm grid is defined and the cost and potential functions are to be computed only considering points on the grid. A window is defined covering a few cells on the grid in the neighborhood of the vehicle. To compute the Dijkstra's algorithm, the local goal is established in the intersection of the line drawn from the actual position to the global goal, with the border of the active window. The case for a 13x8 window and a 20x20 grid was analyzed with the mobile position is fixed with respect to the window. There may be a case where the objective falls in an obstacle but when the windows moves this local objective will change. If the final objective is really occupied by an obstacle, a trap situation will occur. Prevision in software shoul be taken to cope with those situations.

Later, cost map is placed on the active window based on the information obtained by the sensors using the Gaussian bells as we described before. Then, with this *partial* cost map Dijkstra’s algorithm is computed in the window. First, the actual position of the vehicle is taken as the ground node and later, the local destination is used as the reference node. Potentials are added to obtain single potential surfaces that hold the optimum path. A new iteration follows: the window is moved one step in the grid and the previous procedure is repeated until the local objective matches the global one.

5 APPLICATION TO REAL TIME NAVIGATION OF AN AGV

This paper presents the application of the method just described to the navigation of an autonomous vehicle built entirely in our Laboratory. The platform is shown in figure 5.

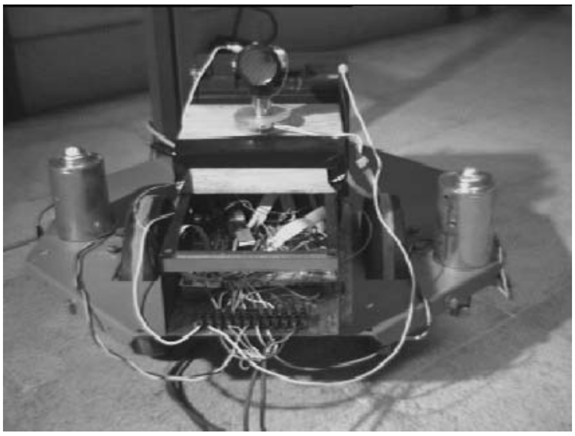

Figure 5: Photograph of the AGV.

Obstacle detection is done with an array of ultrasonic sensors. They determine the distance to the obstacle by measuring the time-of-fly of ultrasonic pulse. Periodically, information received by the sensors is used to detect obstacles updating the cost map by placing Gaussian bells where the obstacles are found and updating the potential surface. With this information, orders are sent to the motors to advance the wheels to the next position.

To smooth the trajectory followed by the robot a pair of fuzzy controllers is used to control the speed of the motors, governing heading and velocity of the platform. Potential ahead of the mobile is determined within an angle of 180 degrees, at a fixed distance from the vehicle.

Then the angle ϕ, measured from the main axis of the robot to the orientation where the minimum potential detected, is used as the input of two fuzzy controllers. One controller uses ϕ and the previous control action to act on the speed of both motors and hence the velocity of the mobile. The other controller, with the same inputs, adds a Δω to the velocity on one wheel and subtracts the same amount from the velocity of the other to controlling the direction of the platform.

The speed of each wheel is governed by:

$$left\_wheel\_speed = average\_speed - \Delta\omega$$
$$right\_wheel\_speed = average\_speed + \Delta\omega$$

The combination of both controllers provides an average positive speed to the center of the platform though one of the individual wheels could be spinning backwards. This allows for quick heading corrections without significant displacements.

The rulebases utilized are shown in tables I and II. Each controller has only 25 rules.

Table I - Rulebase controlling heading.

Ø **Δ Vel. ant.**	**NB**	**NM**	**Z**	**PM**	**PB**
NB	NB	NM	NM	Z	Z
NM	NB	NM	Z	Z	PM
Z	NM	MN	Z	PM	PM
PM	NM	Z	Z	PM	PB
PB	Z	Z	PM	PM	PB

Table II - Rulebase controlling average speed

Ø **Prev. speed.**	**NB**	**NM**	**Z**	**PM**	**PB**
VB	B	VB	VB	VB	B
B	M	B	VB	B	M
M	M	B	B	B	M
S	S	M	M	M	S
Z	Z	S	S	S	Z

Fig. 6 shows the trajectories obtained applying the method described previously. The black dots represent the reflections detected by the sonar. Also, the cost is drawn by contour lines.

6 CONCLUSIONS

A method has been developed and tested in real time from navigating a robot in the presence of obstacles. The technique consists on the application of Dijkstra algorithm to a spatial window determining local objectives based on the cost maps and potential surfaces built with the information obtained by the ultrasonic sensors. A fuzzy technique is used to smooth the trajectory. The configuration of the objects was changed with good results in most of the cases. A comparison was made with the Virtual Force Field technique (VFF) developed by Borestein giving shortest and smoothest trajectories but with longer computing times.

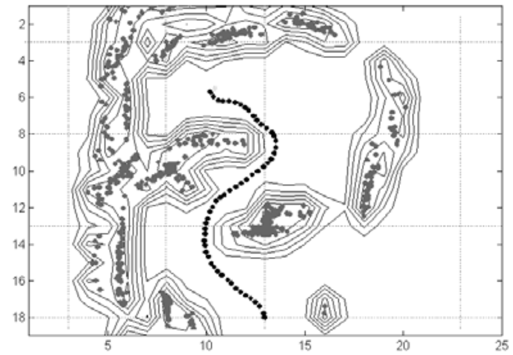

Figure 6: Experimental results.

REFERENCES

[1] Abril, J., Calvo, O. and Salom, J.(1995). *Lógica Difusa a bordo de un velero*, V Congreso Español sobre Tecnologías y Lógica Fuzzy, Murcia , España, 20,21 y 22 de Setiembre de 1995.

[2] Borestein, J. and Koren, Y. (1989). *Real-time obstacle avoidance for Fast Mobile Robots*, Transactions on Systems, Man and Cybernetics, vol 19, No 5:1179-1187, Sep/Oct

[3] Borestein, J., Everett, H.R. and Feng, L. (1996) “Where am I ?” Sensors and Methods for Mobile Robots Positioning. University of Michigan.

[4] Cancelo, G., Mayosky M. and Vignoni, R. (1995) *Gaussian Networks for Autonomous Path Planning*, International Symposium on Artificial Neural Networks

[5] Dijkstra, E. (1959) *A note on two problems in connexion graphs.* Numerishe Mathematik,v1, 269-272

[6] Elfes, A. (1990). Occupancy Grids: *A stochastic spatial representation for active robot perception*, Sixth Conference on Urcentainty in AI.

[7] Hassoun, M. and A. Sanghvi (1990) *Fast Computation of Optimal Paths in Two- and Higher-Dimension Maps*, Neural Networks, V3, 355-363

[8] Moravec, H.P.and A.E. Elfes (1985) *High resolution maps from wide angle sonars*, Proceedings of the 1985 IEEE International Conference on Robotics and Automation, pages 116-121, Washington, D.C.

[9] Rodriguez Mir, G., Calvo, O. (1996). *Navegación Autónoma de Vehículos usando Lógica Difusa*, 7° Congreso Latinoamericano de Control automático (AADECA’96, Buenos Aires, 9 1l 13 de septiembre de 1996.

[10] Sugeno, M. and M. Nishida (1985) *Fuzzy control of a model car*, Fuzzy sets and Systems,16:103-113.