# Autonomous Robot Navigation using Genetic Algorithms

F. ARAMBULA COSIO, M. A. PADILLA CASTAÑEDA
Lab. de Imágenes y Visión
Centro de Instrumentos, UNAM
México, D.F., 04510
MEXICO

*Abstract:* - In this paper is presented a navigation scheme, based on a genetic algorithm, for autonomous robot navigation. Potential fields are used to attract the robot by the goal position and reject it by the obstacles. In the scheme presented here obstacles are automatically detected by simulated sonar sensors. The configuration of the optimum potential field is determined by the genetic algorithm. Intermediate goal points are used for global path planning. Simulation results show that the scheme reported has a good performance in unknown environments with high obstacle densities.

*Key-Words:* - Automatic guided vehicles, Robot navigation, genetic algorithms, potential fields

## 1 Introduction

The two main approaches for the implementation of robot navigation algorithms are artificial potential fields, and artificial intelligence methods. Khatib [1] introduced the use of potential fields for autonomous navigation. The main idea is to generate attraction and repulsion forces to guide the robot to its goal. The goal point has an attractive influence on the robot and each obstacle tends to push away the robot, in order to avoid collisions. Potential field methods provide an elegant solution to the path finding problem. Since the path is the result of the interaction of appropriate force fields, the path finding problem becomes a search for optimum field configurations instead of the direct construction (e.g. using rules) of an optimum path. Different approaches have been taken to calculate appropriate field configurations. Vadakkepat et al.[2] report the development of a GA for autonomous robot navigation based on artificial potential fields. Repulsion forces are assigned to obstacles in the environment and attraction forces are assigned to the goal point. The GA adjusts the constants in the force functions. Multiobjective optimisation is performed on 3 functions which measure each: error to the goal point, number of collisions along a candidate path, and total path length. This scheme requires *a priory* knowledge of the obstacle positions in order to evaluate the number of collisions through each candidate path. Kun Hsiang et. al [3] report the development of an autonomous robot navigation scheme based on potential fields and the chamfer distance transform for global path planning in a known environment, and a local fuzzy logic controller to avoid trap situations. Simulation and experimental results on a real AGV are reported for a simple (4 obstacles) and known environment. McFetridge and Ibrahim [4] report the development of a robot navigation scheme based on artificial potential fields and fuzzy rules. The main contribution of the work consists in the use a variable for the evaluation of the importance of each obstacle in the path of the robot. Simulation results on a very simple environment (one obstacle) show that use of the importance variable produces smoother and shorter trajectories.

In this work is presented an adaptive navigation scheme based on artificial potential fields which are automatically adjusted to avoid obstacles, using a genetic algorithm. Auxiliary attraction points have been used in order to allow the robot to navigate around large obstacles. Each chromosome in the population of the GA represents the set of constants used to calculate the attractive and repulsive forces. Simulated sonar sensors are used to detect obstacles and the room walls. Intermediate goal points (the door positions of each room in a building) constitute the only *a priory* knowledge used for global path planning. The problem of finding a feasible robot path is approached as an iterative search for an optimum configuration of the potential field forces.

In the following section we describe the artificial potential (scalar) field and the corresponding (vector) force field functions used in this work. In section 3 we present the construction of the GA for optimisation of the force field during autonomous robot navigation. In section 4 we present the results of 30 simulation experiments performed on 3 different obstacle configurations. In section 5 are presented the conclusions drawn from the work reported.

## 2 Potential Field and Force Field Functions

The robot is represented as a particle under the influence of an artificial potential field $U$, defined as:

$$U = U_{att} + U_{rep} \qquad (1)$$

where: $U_{att}$ and $U_{rep}$ are the attractive and repulsive potentials respectively.

The attraction influence tends to pull the robot towards the target position, while repulsion tends to push the robot away from the obstacles. In a two dimensional map, the vector field of artificial forces $\mathbf{F}(\mathbf{q})$ is given by the gradient of $U$:

$$\mathbf{F}(\mathbf{q}) = -\nabla U \qquad (2)$$

where: $\nabla U$ is the gradient vector of $U$ at $\mathbf{q}$(x,y) robot position.

In this manner, $\mathbf{F}$ is defined as the sum of two vectors $\mathbf{F}_{att}(\mathbf{q}) = -\nabla U_{att}$ and $\mathbf{F}_{rep}(\mathbf{q}) = \nabla U_{rep}$, as shown in eq. 3.

$$\mathbf{F}(\mathbf{q}) = \nabla U_{rep} - \nabla U_{att} \qquad (3)$$

### 2.1. Attractive and Repulsive Forces

For stabilisation purposes[5], the potential field $U$ is defined as a parabolic function, both, $U_{att}$ and $U_{rep}$ are also defined in this way. The artificial force $\mathbf{F}$ is obtained from the derivation of $U$, both $\mathbf{F}_{att}$ and $\mathbf{F}_{rep}$ are derivatives of parabolic functions. The following artificial attraction force $\mathbf{F}_{att}$ is used in this work for the goal point, and for each of the 4 auxiliary attraction points:

$$\mathbf{F}_{att}(\mathbf{q}) = -\xi(\mathbf{q} - \mathbf{q}_a)\frac{1}{|\mathbf{q} - \mathbf{q}_a|} \qquad (4)$$

where: $\mathbf{q}$ is the current position of the robot;
$\mathbf{q}_a$ is the position of an attraction point;
$\xi$ is a positive weight adjusted by the genetic algorithm.

Eq. 4 is normalised to produce an attraction force independent of the distance between the robot and the goal point. The artificial repulsion force $F_{rep}$ is defined as:

$$\mathbf{F}_{rep}(\mathbf{q}) = \frac{1}{2}\eta * sqrt\left(\frac{1}{d} - \frac{1}{d_0}\right) * \frac{(\mathbf{q} - \mathbf{q}_o)}{(d - o.o4)^2}$$

if $d < d_0$ (5)

$$\mathbf{F}_{rep}(\mathbf{q}) = 0 \ : \text{if} \ \ d > d_0$$

where: $\mathbf{q}$ is the robot position;
$\mathbf{q}_o$ is the obstacle position;
$d = |\mathbf{q} - \mathbf{q}_o|$
$d_0$ is the influence distance;
$\eta$ is a positive weight adjusted by the genetic algorithm.

As the robot gets closer to an obstacle, the repulsion force of the closest obstacle cells grows in the opposite direction of the robot trajectory. In the event of a collision with an obstacle, the value of $\boldsymbol{F}_{rep}$ is bounded by setting the minimum value of $d$ to 0.05. On the other hand if the robot distance to an obstacle cell is higher than $d_0$, that obstacle cell has no effect on the robot.

## 3 Robot Navigation Approach

The robot is represented as a particle $R$ that moves in the configuration space $C$, modelled as a two dimensional grid, where each cell $U$ inside $C$ can be occupied by the robot, the goal or the obstacles. There is also an associated obstacle map $M$ of the same size of $C$. The obstacle map is initially empty, and it is filled at the positions of the obstacles detected by the robot, as it moves inside $C$. The goal cell, and 4 auxiliary attraction cells exert an attractive force on $R$ given by Eq. 4, while each of the detected obstacle cells exerts repulsion forces given by Eq. 5 . For obstacle detection, a $5x5$ grid simulates the robot sensors. When $R$ moves, the positions of the sensors in the mask are updated and used to calculate the distance $d_{\min}$ to the closest detected obstacle (Fig.1). A predefined distance of 5 is assigned to obstacles outside of the detection mask.

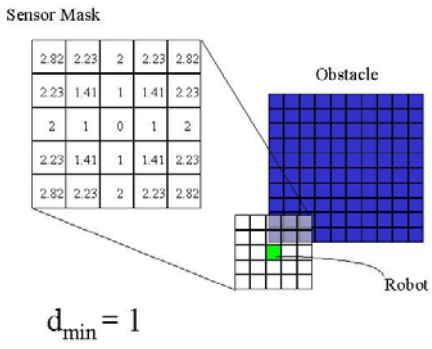



Fig.1, Obstacle detection

In order to avoid trap situations or oscillations in the presence of large or closely spaced obstacles [5], 4 auxiliary attractive cells have been placed around the goal cell (Fig. 2). Each attractive force $\mathbf{F}_{att}^{i}$ of the cell $c_i$, depends on the corresponding value of $\xi_i$, which is automatically adjusted by the genetic algorithm described in section 3.3.

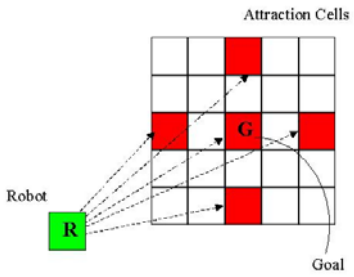



Fig. 2, Attraction field composed of 5 attraction cells with adjustable force intensity.

### 3.2.Objective function for robot navigation

An *adhoc* objective function was constructed to evaluate force field configurations which correspond to an optimum robot position (i.e. positions closer to the goal cell which also avoid obstacles). The objective function value of each candidate force field configuration is evaluated with three criteria: minimisation of the error distance $E$ between the robot and the goal cell; maximisation of the distance $d_{min}$ to the closest obstacle cell; and minimisation of the magnitude of the resultant repulsion force vector $\mathbf{F}_r$. Eq. 6 shows a function which produces optimum (minimum) values for minimum $E$, maximum $d_{min}$ and minimum magnitude of $\mathbf{F}_r$.

$$f(\mathbf{q}) = \sqrt{E/2} \cdot e^{-d_{\min}} \cdot \left|\mathbf{F}_r\right|^2 \quad \text{:if } d_{min}>0 \qquad (6)$$
$$f(\mathbf{q}) = 2000 \qquad \text{:if } d_{min}=0$$

where:

$$E = \left|q_{rx} - q_{gx}\right| + \left|q_{ry} - q_{gy}\right|$$

$\mathbf{q}_r$ is a candidate cell for the new robot position;
$\mathbf{q}_g$ is the goal cell;
$\mathbf{F}_r$ is the resultant repulsion force vector.

The construction of the objective function ($f$) favours robot paths that run away from the obstacles and result in decreasing distance to the goal cell. The case where $d_{min}$= 0 (which corresponds to a collision) is severely penalised in order to avoid the selection of the corresponding chromosome in the next generation. In Fig. 3 is shown the plot of Eq.6 for: 0<=E<=44, 0.1<= $d_{min}$<=5, and $\left|\mathbf{F}_r\right| = 1$.

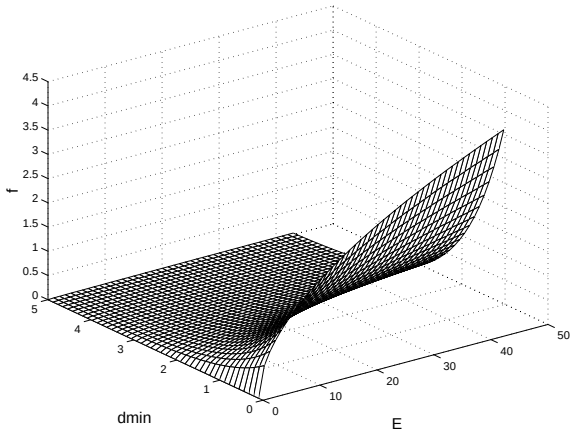



Fig. 3, Plot of Eq.6 for: 0<=E<=44, 0.1<= $d_{min}$<=5, and $\left|\mathbf{F}_r\right| = 1$;

### 3.3. The genetic algorithm

The principles of operation of a GA are presented in [6]. In this work a GA is used to optimise the values of the weights ($\xi_i$) of the 5 attraction cells and the values of the weights ($\eta_j$) of up to 155 obstacle cells. Each variable has a range of {0, 1000} and is binary coded with 20 bits of resolution in order to maintain a large number of values for the repulsion and attraction forces. A chromosome is formed by concatenation of the 160 binary coded variables. The GA searches for optimum values of $\xi_i$ and $\eta_j$ in a given binary string (chromosome) which move the robot to a position such that $f$ (Eq 6) has a minimum value. Only those $\eta_j$ which have been detected by the robot are used to calculate the force fields given by Eq. 5, the rest of the repulsion weights in the string is ignored. At each generation of the GA, every chromosome in the current population is evaluated using Eqs. 4, 5, and 6. The selection probability ($P_s$) of a given chromosome is determined with a *ranking* method [7]. Each chromosome in the current population is assigned a number of copies with probability $P_s$ using stochastic universal sampling [8]. Single point crossover is applied with a probability of 0.6, mutation is applied to each string with a probability of 0.01 per bit. Finally, the next generation of the

GA is formed using fitness based reinsertion with a generation gap of 0.8. This process continues until the robot reaches the goal cell or 200 generations are completed.

## 4 Experiments and results

The algorithm was implemented in Matlab using the GA toolbox developed at the University of Sheffield [7]. For evaluation we used a cell map of 40x40 cells simulating a 5-room floor. Shown in Fig. 4a counter clockwise from the top-left corner: the first room simulates an storage room, the next room simulates a hospital bed, the next room simulates a meeting room, the next room is empty with random obstacles, and the last room is a rotated hospital bed. Three different random obstacle distributions of different obstacle densities were used ( Figs. 4a, 4b, 4c).

Ten experiments were performed for each obstacle distribution, the start and goal positions for each experiment are shown in table 1, the origin is placed at the top-left corner of the cell map. Two intermediate goal points have been used to guide the robot through the corridor corner as well as through the door of the appropriate room. The positions of the intermediate goal points are also shown in table 1. The robot travels from the start position to each successive intermediate goal point and to the final goal point.

| Exp. No | (start)-(goal) | intermediate goal 1 | intermediate goal 2 |
|---|---|---|---|
| 1 | (34, 9)-(11, 3) | (20,10) | (15,10) |
| 2 | (34, 9)-(13, 14) | (20,10) | (15,21) |
| 3 | (34, 9)-(3, 26) | (20,10) | (15,38) |
| 4 | (34, 9)-(36, 27) | (20,10) | (25,38) |
| 5 | (34, 9)-(37, 14) | (20,10) | (25,22) |
| 6 | (34, 4)-(3, 6) | (20,10) | (15,10) |
| 7 | (34, 4)-(3, 14) | (20,10) | (15,21) |
| 8 | (34, 4)-(12, 26) | (20,10) | (15,38) |
| 9 | (34, 4)-(30, 30) | (20,10) | (25,38) |
| 10 | (34, 4)-(38, 22) | (20,10) | (25,22) |

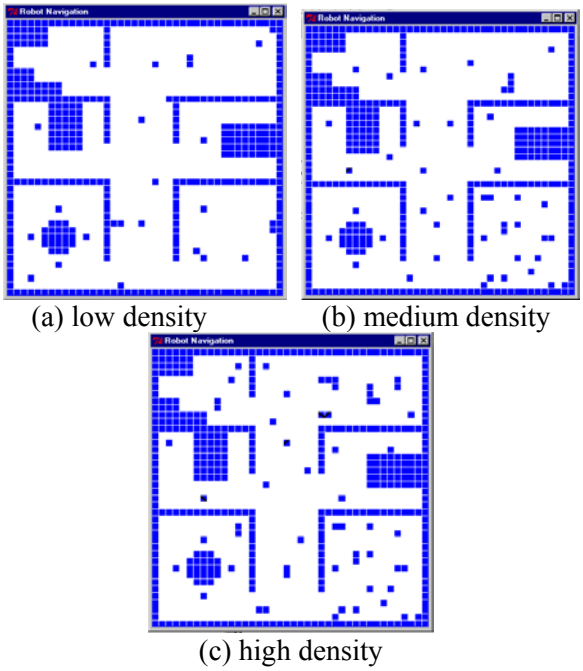
Table 1. Start-goal and intermediate goal positions of each experiment

In table 2 are shown the results of the 30 experiments performed, the first column shows the obstacle distribution used (Fig.4), and the (start)-(goal) position number as previously shown in table 1. Columns two and three show respectively, the total distance travelled by the robot (measured in cells), and the deviation (as a percentage) from the optimum shortest path.

| Exp.No. | Total distance (cells) | Deviation from optimum 1 (%) |
|---|---|---|
| a-1 | 39 | 34.5 |
| a-2 | 33 | 10.0 |
| a-3 | collision | collision |
| a-4 | 62 | 14.8 |
| a-5 | 48 | 41.2 |
| a-6 | 36 | 16.1 |
| a-7 | 47 | 11.9 |
| a-8 | 67 | 24.1 |
| a-9 | 71 | 26.8 |
| a-10 | 43 | 7.5 |
| b-1 | 30 | 15.4 |
| b-2 | 41 | 36.7 |
| b-3 | collision | collision |
| b-4 | 66 | 17.9 |
| b-5 | 44 | 4.8 |
| b-6 | 43 | 26.5 |
| b-7 | collision | collision |
| b-8 | collision | collision |
| b-9 | 55 | 12.2 |
| b-10 | 44 | 15.8 |
| c-1 | 34 | 17.2 |
| c-2 | 44 | 29.4 |
| c-3 | collision | collision |
| c-4 | 68 | 21.4 |
| c-5 | 46 | 15.0 |
| c-6 | 40 | 21.2 |
| c-7 | 49 | 11.4 |
| c-8 | 70 | 27.2 |
| c-9 | 75 | 41.5 |
| c-10 | 48 | 17.0 |
| | | Average:20.7% |

Table 2. Experiment results: Total distance 1, and Deviation from optimum 1 obtained with auxiliary cells placed at fixed distance (15 cells) from the goal

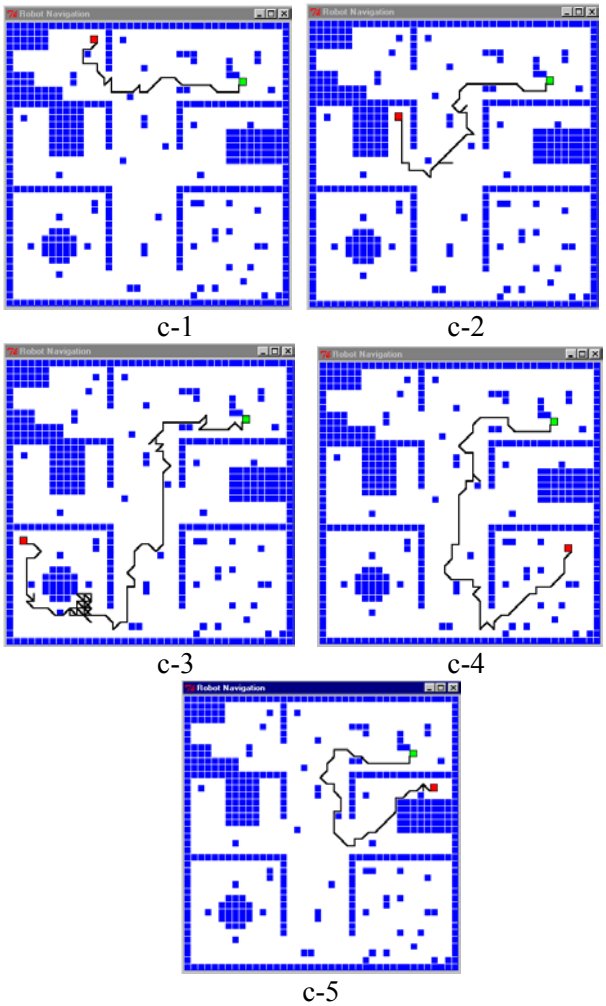
From the results shown in table 2 the success rate (i.e. the percentage of paths completed without collision) is 83.3%. The average deviation from the optimum path length is 21%. In figure 5 are shown 5 paths produced by the navigation approach for obstacle configuration (c) (as shown in Fig.4), which has the highest obstacle density. The average time for path completion on a Pentium III PC at 750MHz is 115s with an average path length of 56 cells (i.e.2.05 s/step).



(a) low density (b) medium density

(c) high density

Fig.4, Obstacle distributions tested



c-1 c-2

c-3 c-4

c-5

Fig. 5, Paths produced by the navigation algorithm, for maximum obstacle density.

## 5 Conclusions

An autonomous robot navigation algorithm has been developed. The scheme would enable a mobile robot, equipped with sonar sensors, to navigate through unknown obstacle distributions. Intermediate goal points have been used to guide the robot through corridor corners and through the door of the appropriate room in a simulated floor plan. This knowledge is available from the floor plan of any building. Given an start and end position, intermediate goal positions, can be easily calculated (e.g. using rules). The potential fields approach has been used and modified to allow avoidance of large or closely spaced obstacles, through the use of auxiliary attraction cells with adjustable force strength, 4 auxiliary cells have been used in this work providing good simulation results.

A genetic algorithm has been used for on-line optimisation of the force intensity parameters of the repulsion and attraction cells, as well as the position parameter of the auxiliary attraction cells. The scheme has been able to complete ten different paths in three different unknown obstacle configurations with a success rate of 83.3%. The total estimated processing time (i.e. ultrasound scanning plus force field optimisation) of an implementation in C of the scheme reported is 0.36 s/step, this value would allow for a maximum robot speed of 0.27 m/s (with a cell size of 0.1 x 0.1m). This is within acceptable robot speed values for real applications.

*References:*


[1] Khatib O. Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. In: Autonomous Robot Vehicles, I.J. Cox and G.T. Wilfong, Eds. Springer-Verlag 1990; 396-404.

[2] Vadakkepat, P., Kay Chen Tan, Wang Ming-Liang. Evolutionary artificial potential fields and their application in real time robot path planning. In: Proceedings of the 2000 Congress on Evolutionary Computation 2000; July: 256-263.

[3] Kun-Hsiang Wu, Chin-Hsing Chen, and Juing-Ming Ko. Path planning and prototype design of an AGV. Mathematical and Computer Modelling 1999; 30: 147-167.

[4] McFetridge L. and Yousef Ibrahim M. New technique of mobile robot navigation using a hybrid adaptive fuzzy-potential field approach. Computers ind. Engng 1998; 35 (3-4):471-474.

[5] Koren Y., Borenstein J. Potential field methods and their inherent limitations for mobile robot navigation. In: Proceedings of the IEEE Int. Conf. on Robotics and Automation 1991;1398-1404.

[6] Goldberg D.E.: Genetic Algorithms in Search, Optimisation, and Machine Learning. Addison-Wesley, MA, (1989).
[7] Chipperfield, P. Fleming, H. Pohlheim, C. Fonseca.: Genetic Algorithm Toolbox for use with Matlab. User's Guide. Automatic Control and Systems Engineering, University of Sheffield, UK (1995).
[8] Baker, J.E.: Reducing bias and inefficiency in the selection algorithm. Proc. 2nd Int. Conf. on Genetic Algorithms. Hillsdale, N.J., USA, 1987; 14-21.