Motion Planning for Car-like Robots Using Hierarchical Genetic Algorithms

N. Achour, A. Lakhdari and F. Ferguene

Abstract— **The problem of optimal motion generation is still a major focus of research in mobile robotics. Although many solutions have been proposed, it has always had to take into account new factors and constraints that may be related either to the environment (obstacles) or the robot itself. For a car-like vehicle, nonholonomic kinematic constraints force the robot to follow a trajectory imposed by the angle of its steering wheels. In this paper, we present an algorithm for motion planning in order to generate optimal movements for a nonholonomic mobile robot using a probabilistic network method (PRM) associated with two artificial intelligence techniques: the A * algorithm to search for the shortest path and genetic algorithms for optimization. We used the Hierarchical Genetic Algorithms (AGH) that played on two criteria: collision and path length to calculate the optimal path without collision. One of the key points of our approach lies in the evaluation of the individuals; we used the principle of artificial potential field that has yielded interesting results in terms of quality and optimization of trajectories.**

***Index Terms*—path Motion planning; PRM; Hierarchical Genetic algorithms; robotics.**

I. Introduction

Since a few years an increasing interest is carried within the robotics community on planning optimal motion. The interest is born of the need to optimize the energy consumed by nonholonomic vehicles (e.g car-like vehicles), developing algorithms in this sense has a very important economic significance.

Manuscript received February 21, 2013; revised April 02, 2013.
Nouara Achour is with the Laboratory LRPE at USTHB University of Sciences and Technology, BP 32, El Alia, Algiers, Algeria, phone: 21321247912, fax: 21321247187, e-mail: nachour@usthb.dz
Abdelhalim Lakhdari is with the Laboratory LRPE at USTHB University of Sciences and Technology, BP 32, El Alia, Algiers, Algeria.
Farid Ferguene is with the Laboratory LRPE at USTHB University of Sciences and Technology, BP 32, El Alia, Algiers, Algeria.

The problem of motion generation for non holonomic mobile robots can be divided into two sub-problems: the first is to find optimal paths, the second is to adapt these paths to be achievable by the robot taking into account the constraints related to the latter, these paths must satisfy certain conditions:

- They should be collision-free and minimum length
- They must be smooth with minimum of bends
- They must be achievable by the robot in monotonic time.

Recently, random sampling has emerged as a powerful technique for planning in large configuration spaces [1][2].
Random-sampling planners are classified into two categories: PRM (Probabilistic RoadMap) and RRT (Rapidly-Exploring Random Tree).
In this article we will use the Hierarchical Genetic Algorithms (AGH) that will play on two criteria: collision and path length to calculate the optimal path without collision. One of the key points of our approach lies in the evaluation of the individuals; we used the principle of artificial potential field that has yielded interesting results in terms of quality and optimization of trajectories.
Genetic algorithms (GA's) are search strategies based on models of evolution [3]. They have been shown to be able to solve hard problems in tractable time. Here, we need a solution space composed of a set of nodes randomly generated in C_{free} (free Configuration space).
Section II describes briefly the PRM-based path planning. Section III lays the mathematical description of the kinematic model of the non holonomic robot. Section IV details our approach in using genetic algorithms to plan optimized paths. In section V, we report a series of actual runs.

II. The PRM approach

The principle of probabilistic networks PRM is simple; we construct a graph G = (N, V) and capture the connectedness of $\mathcal{CS}_{free}$. [4].The N nodes are free configurations generated randomly according to a uniform distribution. Arcs of G collected in the set V, are free collision paths connecting the N nodes, two by two. Most PRM methods have the same general structure; they are usually divided into two phases: a learning

ISBN: 978-988-19252-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

phase and a research phase [1]. The learning phase consists of the graph building with enough nodes to cover uniformly the free configuration space $\mathcal{CS}_{free.}$ and this even in the most difficult places of this area. It usually consists of two stages: a stage of construction, and an another for expansion.

A. The Construction stage

In this step, we use a sampling strategy to randomly generate new nodes in the free configuration space and insert them in the graph **G**. Then we use a neighbor selection strategy to select a set of neighboring nodes for each new generated node. A collision detector can determine whether a node belongs or not to the free space. It is also used to test whether a given path is included or not in the free space.

B. The Expansion stage

This step improves the connectivity graph generated in the construction stage. It will intervene especially in places where G is disconnected while $\mathcal{CS}_{libre.}$ is not, which would correspond in most times to narrow regions so it is useful in exploring regions with narrow passages.

C. The parameters selection

To calculate the number N_{max}, we used a method based on calculation of the ratio of the free surface on the total area of the environment, given a sampling distance d_{min}.
This process is repeated K times then we calculate the average of the ratios obtained for a better approximation, the number N_{max} will be the full value of the ration of the free surface on the disk surface which includes a square of side d_{min}.

$$N_{max} = integer\left(Q \times \frac{surface}{d_{min}^2 \times \frac{\pi}{2}}\right) \quad (1)$$

Figure 1 shows an example with d_{min}=5 and N_{max}=250.

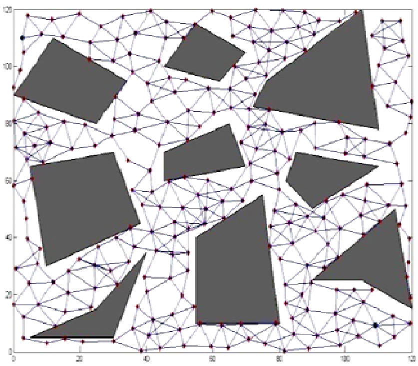

Fig 1.The environment discrétisation
$d_{min} = 5, N_{max} = 250$

To determine if a path between two nodes is a free path or not, we used a binary process of collision detection.

D. Search of the shortest path

To find the shortest path from the initial configuration to the target configuration we used a three-step algorithm. In the first step the A* algorithm [5] is applied to find the shortest path in the network built previously; this path is a concatenation of arcs included in the PRM network. Because of the probabilistic nature of the PRM, the calculated path may contain irregular parts. A shortening process is performed in a second step to eliminate uneven parts of the computed path (Fig 2).

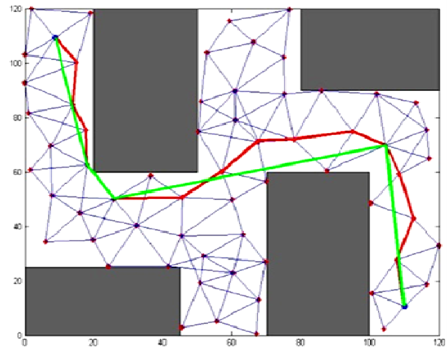

Fig 2. Paths before (red) and after (green) shortening.

E. Path smoothing

For car-like vehicles the problem of planning feasible motion can't be addressed the aspects considered so far. This is due to the existence of a link between some special configuration parameters and velocity of the vehicle making the generation of a feasible movement even more difficult [6]. The obtained paths during the searching stage can be long and irregular because of the probabilistic character of the PRM, therefore to be executed by the robot, paths must be smoothed and optimized.
A shortcutting process is used for path optimization by eliminating uneven parts [8].

III. THE NON HOLONOMIC ROBOT

For a car-like vehicle we can write:

$$G\ (q, \dot{q}, t) = 0 \quad (2)$$

$$G = -\dot{x}(t)\sin\theta\ (t) + \dot{y}(t)\cos\theta\ (t) \ = 0 \quad (3)$$

A configuration q of the robot is defined by the 4-uple $q = [x, y, \theta, \phi]$ de $\mathcal{R}^2 \times [0,2\pi\,[\times [-\boldsymbol{\phi}_{max}, \boldsymbol{\phi}_{max}]$ where x and y define the position of the reference point $\boldsymbol{M}$, θ is the orientation of the vehicle in the frame $\mathcal{F}(\boldsymbol{O}, \vec{\imath}, \vec{\jmath})$ and $\boldsymbol{\phi}$ the steering angle .
The discretization of the kinematic model can be written as follows [6][7]:

$$\begin{cases} x_{i+1} = x_i + \Delta t\ \frac{v}{2l}\ cos(\theta_i + tan\phi\ \Delta t) \\ y_{i+1} = y_i + \ \Delta t\ \frac{v}{2l}\ sin(\theta_i + tan\phi\ \Delta t\ ) \\ \theta_{i+1} = \theta_i + \ \Delta t \frac{v}{\ell}\ tan\ \phi. \end{cases} \quad (4)$$

ISBN: 978-988-19252-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

A path tracking function allows to extract the different steering angles along the latter. First, this function will calculate two points of the steering (the beginning and end of the steering) for every three successive points of the path (Fig 3) [7][8][12]. Then it will split the path that connects these three points in two subpaths, the first is a straight line from the first point to the start point and the second is the minimal radius arc ($\pm\phi_{max}$)) which connects the two steering points. Figure 3 shows the operation of path following.

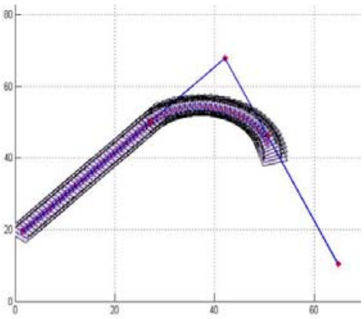

Fig 3. The path following between three points

During the following process, we will extract the different steering angles in each interval of time Δt (Fig 4) [13].

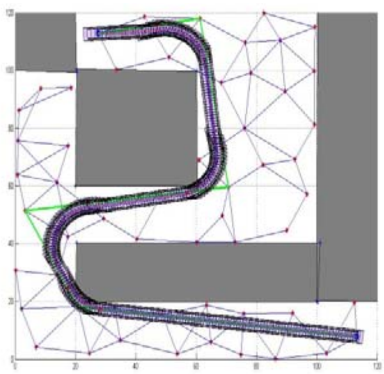

Fig 4. execution of the path by a nonholonomic robot

IV. Optimizing by Genetic Algorithms

During the execution of the trajectory, the robot may be in collision with the obstacles (fig 4) even if it is not the case, the calculated path is not necessarily the best, we must seek a sequence of steering angles corresponding to a collision free shortest path. To do this, we will use the Hierarchical Genetic Algorithms (HGA) [9] that will perform on two criteria; collision and path length to calculate optimal path without collision.

To generate the initial population we must repeat all previous steps from the construction of the PRM to the following path and that in order to have more possible paths connecting the two positions start and goal [10]. Fig 5 shows ten paths obtained by the PRM method executed ten times.

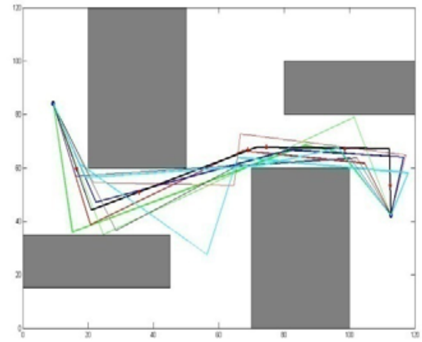

Fig 5. Ten Paths obtained by PRM executed ten times

Each individual has two chromosomes, the first contains genes carrying angle values ϕ_i extracted during the follow up of each path and the second contains the angle θstart of the startup configuration.

The size of the first chromosome differs from one individual to another depending on the way it extracts its genes (steering angle).

The size of the first chromosome differs from one individual to another depending on the path in which it extracts its genes (steering angle). To overcome this problem we added to the first chromosome of each individual a certain number of genes (initialized to zero) until its size is equal to the size of the largest chromosome then we assigned to each parameters gene a control gene g_i which will disable the parameters gene if it is an added gene ($g_i = 0$) or activate it if not ($g_i = 1$).

The structure of an individual can be represented by figure 6.

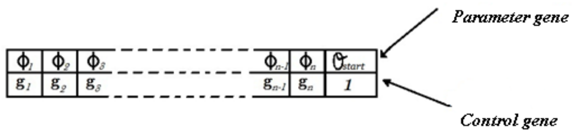

Fig 6. An individual structure

A. The fitness function

The evaluation of each individual is performed as follows:

We calculate a reference path by applying to one of the paths calculated previously the procedure of shortening and discretization (with a very small stepsize) enough times to get the shortest path that connects start and goal configurations. Then we define a potential v_i of a configuration q_i which is proportional to the distance to obstacles and the reference path when the robot is at the configuration q_i thus the reference path will exert attractive field on the solution while the obstacles will exert a repulsive field [11] and the evaluation of each gene should be proportional to the resultant of these two fields and also to the number of active parameters genes ($g_i = 1$) which is proportional to the length of the associated path. The potential v_i of a configuration q_i can be expressed by:

$$v_i = \begin{cases} \frac{1}{n}\left(\frac{K_a}{(d_i+1)^2} - \frac{K_r}{(dr_i)^2}\right) & if \quad dr_i \leq d_0 \\ \frac{1}{n}\frac{K_a}{(d_a+1)^2} & if \quad dr_i > d_0 \end{cases} \qquad (5)$$

K_a and K_r are positive constants

ISBN: 978-988-19252-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

d_i : is the distance between the configuration q_i and the reference path.
dr_i : is the distance between the configuration q_i and the nearest obstacle.
d_0 : is the influence distance of the obstacles.
n is the number of active parameters genes.
The evaluation of an individual is therefore the sum of the potentials v_i of its active parameters genes.
The fitness function can be expressed as follows:

$$f = \sum_{i=1}^{n} g_i \times v_i \qquad (6)$$

B. Selection

The selection of individuals is performed using the elitist method where the individuals are sorted in descending order according to the evaluation of each individual. The most successful individuals (who have a high valuation) are selected to participate to the next generation.

C. Reproduction of individuals

The best solutions are combined to give rise to new individuals who are added to the previous set of individuals, this step can be carried out by two processes: crossover and mutation.

D. Crossover

We made a draw with replacement of two individuals who will undergo two types of crosses: the first is a barycentric cross carried out on the genes of parameters with a weight taken randomly in the interval [0 1]. The second is a classical crossover (exchange of genes) performed on control genes, Figure 7 describes the principle of the used crossover.

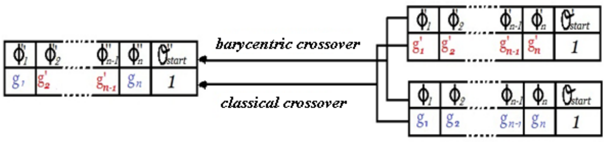

figure 7. The used crossover principle

Where :

$$\begin{cases} \emptyset'' = \alpha\, \emptyset' + (1-\alpha)\emptyset \\ \theta''_{start} = \alpha\, \theta'_{start} + (1-\alpha)\theta_{start} \end{cases} \qquad (7)$$

We repeat the crossing until a sufficient number of individuals who will create the new population.

E. Mutation

In the process of mutation, we used a uniform mutation by adding a gaussian noise with a vey small variance (of the order of $\frac{\emptyset_{max}}{25}$) to the genes of the first chromosome with a probability of about 20%.
For a steering angle $\emptyset$ the mutation process can be expressed as follows:

$$\emptyset' = \emptyset + N\left(0, \frac{\emptyset_{max}}{25}\right) \qquad (8)$$

The angle $\emptyset$ is obtained after mutation is accepted $\emptyset' \in [-\emptyset_{max} \quad \emptyset_{max}]$. We repeat the selection operators crossover and mutation to obtain the shortest feasible path.

V. RESULTS

In this section we present some results obtained under Matlab environment (v.R2009a). Like all heuristic algorithms, our algorithm can take a long time to converge to the optimal solution, as it can quickly converge, this depends on several parameters, the first is the initial population which in turn depends on the probabilistic roadmap planner PRM (N_{max}, d_{min} ...), also the discretization l and M the number of times the shortening and discretization procedure is applied (Figure 5.16):
The solution can converge quickly if the initial population contains a path very close to optimal.
The choice of the evaluation function parameters (K_a K_r, d_0) plays a very important role in the convergence of the solution, in fact more Ka, Kr and d0 increase, the solution approaches the reference path and moves away from obstacles.(Figure 5.16). We present two tests obtained on two different environments with two different start and goal configurations (Figures 7.a , 7.b, 8.a and 8.b).

Environment 1

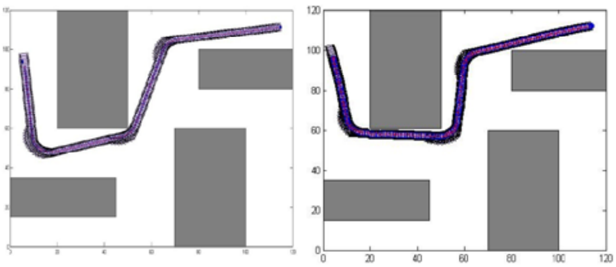

Fig 7.a. Trajectory after 10 generations Fig 7.b. Trajectory after 200 generations

Environment 2

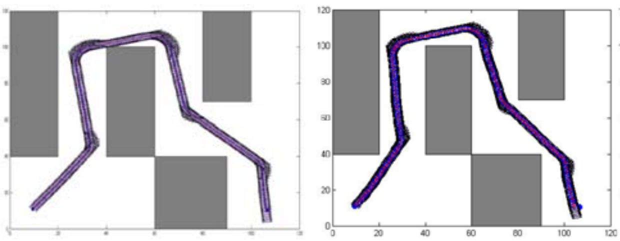

Fig 8.a. Trajectory after 10 generations Fig 8.b. Trajectory after 300 generations

The evolution of the trajectories obtained for the two environments is given in Tables 1 and 2.

Table 1

Generation	10	20	50	100	200
Evaluation	-180.42	-192.41	-220.71	-250.07	-330.35
Trajectory length	170	170	170	169	163
Collision	yes	no	no	no	no

Table 2

Generation	10	20	50	100	300
Evaluation	-531.24	-664.82	-776.07	-963.33	6997.46
Trajectory length	242	240	240	240	240
Collision	yes	no	no	no	no

ISBN: 978-988-19252-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

In the first environment, we note that the algorithm has converged to the optimal solution after 200 generations, and the evolution of the solution is sensitive from one generation to another, while the evolution of the solution in the second environment is very slow from the 50th generation to generation 300. This is usually due to the initial population or the distribution of obstacles that may contain narrow passages. It is difficult to define a stop condition of the algorithm because the solution can stagnate until the mutation operator improves the solution and this can be after any number of generations, so we repeat the genetic operators enough time to have the best solution.

VI. Conclusion

In this article, we focused on the problem of generating optimal motion for a nonholonomic mobile robot which operates in a 2D environment, where obstacles are assumed to be fixed and polygonal. For this type of robot, a free path is not necessarily a feasible path because of the non holonomic constraints that impose a velocity tangent to the trajectory.

We proceeded in three steps, we first calculated free paths without taking into account the constraints of non-holonomy of the robot, using the method of probabilistic roadmap planner PRM associated with a search algorithm for the shortest path (A star) , these paths are then improved by using shortening and discretization algorithms.

In the second step we calculated the commands required to follow these paths, we finally used the HGA algorithms associated to the artificial potential field principle to generate the adequate controls for optimized motion between a start and a goal configuration. The results showed the advantages of genetic algorithms in motion optimization.

References

[1] L. Kavraki, P. Svestka, J. C. Latombe, et M. Overmars, "Probabilistic roadmaps for fast path planning in high dimensional configuration spaces", IEEE Transactions on Robotics and Automation,vol.12(4),pages 566-580, 1996.

[2] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge University Press, 2006.

[3] John Holland, ''Outline for a logical theory of adaptive systems'', *Journal of the Association of Computing Machinery*, 3, 1962.

[4] T. Lozano-Pérez, "Spatial Planning: A Configuration Space Approach", In A.I. Memo No. 605. Massachusetts Institute of Technology, 1980.

[5] P. E. Hart, N.J Nilsson, B.Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths ", in *IEEE Transactions on Systems Science and Cybernetics SSC4*, vol. 4, nº 2, 1968, p. 100–107.

[6] J.C. Latombe, "Robot motion planning", Kluwer Academic Publishers 1991.

[7] J .P. Laumond. "Finding collision-free smooth trajectories for a non holonomic mobile robot", The International Joint Conference on Artificial Intelligence , p.1120- 1123 , Milan , Italy, 1987.

[8] R. Geraerts and M. H. Overmars, "Creating high-quality paths for motion planning. International Journal of Robotics Research", 26(8): 845–863, 2007.

[9] G. Garai, B.B. Chaudhuri. '' A distributed hierarchical genetic algorithm for efficient optimization and pattern matching'', Pattern Recognition 40 (2007) 212 – 228.

[10] Nouara Achour, Mohamed Chaalal " Mobile Robots Path Planning using Genetic Algorithms" Seventh International Conference on Autonomic and Autonomous Systems, pp. 111- 115, Italy, 2011.

[11] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *Int. Journal of Robotics Research*, 5(1):90–98, 1986.

[12] P. Svestka and M. H. Overmars. Motion planning for car-like robots using a probabilistic learning approach. Int. Journal of Robotics Research, 16(2):119–143, 1997

[13] D. G. Macharet, A. A. Neto, and M. F. M. Campos, "Feasible UAV Path Planning Using Genetic Algorithms and Bezier Curves," in *Brazilian Symposium on Artificial Intelligence*, 2010.

ISBN: 978-988-19252-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)