



**HAL**  
open science

# Computing good holonomic collision-free paths to steer nonholonomic mobile robots

Thierry Simeon, Stéphane Leroy, Jean-Paul Laumond

► **To cite this version:**

Thierry Simeon, Stéphane Leroy, Jean-Paul Laumond. Computing good holonomic collision-free paths to steer nonholonomic mobile robots. IEEE/FSR International Conference on Intelligent Robots and Systems (IROS), Sep 1997, Grenoble, France. hal-04295571

**HAL Id: hal-04295571**

**<https://laas.hal.science/hal-04295571>**

Submitted on 20 Nov 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Computing good holonomic collision-free paths to steer nonholonomic mobile robots

T. Siméon      S. Leroy      J.P. Laumond

LAAS-CNRS  
7, avenue du Colonel-Roche  
31077 Toulouse Cedex - France  
{nic,sleroy,jpl}@laas.fr

**Abstract:** *Several schemes have been proposed in the path planning-literature to plan collision-free and feasible trajectories for non-holonomic mobile robots. A classical scheme is the two-step approach proposed in [4, 8, 5] which consists in first computing a collision-free holonomic path, and then in transforming this path by a sequence of feasible ones. The quality of the solution and the computational cost of the second step depend on the shape of the holonomic path. In this paper, we introduce a nonholonomic cost of the geometric path to be approximated and we propose a configuration space structuring that allows us to compute an holonomic path minimizing at best the nonholonomic cost. The algorithms have been implemented and we present simulation results which illustrate the efficacy of the planner to produce good solutions with respect to the nonholonomic constraints of a mobile robot.*

## 1 Introduction

This paper deals with a two-step approach of motion planning for nonholonomic mobile robots consisting in computing a collision-free geometric (holonomic) path and then approximating this path by a sequence of collision-free feasible ones. Such planning scheme works for any small-time locally controllable systems; it has been exploited along several directions for mobile robots [4, 3, 6] as well as for mobile robots with trailers [8, 5].

The quality of the solution (in terms of the number of maneuvers) and the computational cost of the approximating step depends on the shape of the first geometric path. This dependance, already noticed in [6], is illustrated in Figure 1: a crab-wise path is more expensive to approximate than a path which follows

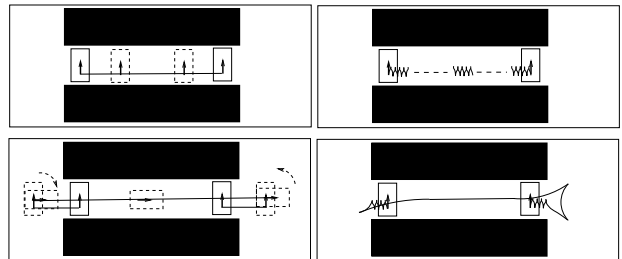


Figure 1: Two holonomic paths (left) and the associated feasible trajectories (right)

the direction of the vehicle. The geometric planner of [6] produces good paths by using a shortest feasible paths metric for building a maximal clearance skeleton. To our knowledge the implementation was however limited to a point robot.

This paper deals with polygonal robots. We first introduce the notion of nonholonomic cost of the geometric path to be approximated. We then propose a configuration space structuring that allows us to compute an holonomic path minimizing at best the non-holonomic cost.

The main idea consists in decomposing the configuration space into slices maintaining the orientation of the robot constant. We then apply a sweeping line algorithm to compute a convex cell decomposition of each slice into trapezoidal cells. For a given slice, *the direction of the sweeping line is the orientation of the robot in this slice*. The main axis of each trapezoidal cell is then supported by the orientation of the robot.

We assume that the robot and the obstacles are simple polygons. All the basic concepts used in this paper (generalized polygons, Minkowski sum, sweeping

lines, roadmaps, cell decomposition, ...) are described in [2].

## 2 Nonholonomic cost of holonomic paths

Let us consider a mobile robot corresponding to the following control system :

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos \theta \\ \sin \theta \\ 0 \end{pmatrix} u_1 + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} u_2 = Xu_1 + Yu_2$$

$u_1$  and  $u_2$  are the linear and angular velocities respectively. When  $u_2 = 0$  the robot moves along a straight line segment in the direction  $\theta$  of the vehicle. When  $u_1 = 0$ , the robot turns around its fixed position  $(x, y)$ .

Let us consider the vector field  $Z = (\sin \theta, \cos \theta, 0)$ . A path along  $Z$  is not feasible; it corresponds to a crab-wise path. Nevertheless, in some situations (e.g., a parking task), the only solution to reach the goal is to maneuver in order to steer the vehicle along this direction (such maneuvers exist because  $Z$  is the Lie bracket of the vector fields  $X$  and  $Y$ <sup>1</sup>). Moreover the number of maneuvers varies as  $\frac{1}{\epsilon^2}$  with  $\epsilon$  being the size of the free-space [4]; this situation corresponds to the worst case.

In the sequel we consider polygonal path in the configuration space  $\mathbf{R}^2 \times S^1$ , such that each straight line segment is either a ‘‘vertical’’ segment (i.e., maintaining a fixed position) or an ‘‘horizontal’’ one (i.e., maintaining the orientation constant). A vertical segment  $V$  is feasible by the mobile robot; it corresponds to a pure rotation. Let  $H_\theta^{\theta'}$  be an horizontal segment of orientation  $\theta'$  lying in the horizontal plane defined by the configurations with a fixed orientation  $\theta$ . The segment  $H_\theta^{\theta'}$  is feasible by the robot only if its orientation  $\theta'$  verifies  $\theta' = \theta$ .

**Definition:** The nonholonomic cost of a vertical segment is null; the nonholonomic cost of an horizontal segment  $H_\theta^{\theta'}$  of length  $\ell$  is  $\ell |\sin(\theta - \theta')|$ . The nonholonomic cost  $Cnh$  of a polygonal path consisting of an alternate sequence of vertical and horizontal segments is the sum of the nonholonomic costs of all the elementary segments.

This definition gives an account of the difficulty to approximate a path by a sequence of feasible ones. Indeed, the cost of a polygonal feasible path is null;

<sup>1</sup>See [4] for details on the Lie brackets and their geometric interpretation in nonholonomic motion planning.

the cost of straight line segments (with fixed length) increases when the segments tend to crab-wise paths.

The idea of the algorithm below is to generate polygonal paths of the type above with small non-holonomic costs.

## 3 A roadmap in the configuration space

The goal of this section is to build a roadmap of the collision-free configuration space consisting of horizontal and vertical straight line segments. The principle of the construction is based on the computation of roadmaps within a fixed number of horizontal slices (Section 3.1); then we compute elementary vertical paths to link two consecutive roadmaps iteratively (Section 3.2).

### 3.1 Roadmap in $\mathcal{P}_\theta$

Let us consider a slice  $\mathcal{P}_\theta$  (i.e., the plane gathering all the configurations with a fixed orientation  $\theta$ ).

The obstacles in  $\mathcal{P}_\theta$  are first computed from the Minkowski sum between the robot at the orientation  $\theta$  and the obstacles of the environment (Figure 2). Then we decompose the free-space of  $\mathcal{P}_\theta$  into trapezoidal cells. The key idea of the method is here: the trapezoidal cells are provided by a sweeping line algorithm, *the direction of the line being  $\theta$*  (Figure 4). The boundaries of all the cells have at least one straight line segment with orientation  $\theta$ . An effect of the sweeping line algorithm is to provide cells elongated in the direction of the sweeping line. Because this direction is the main direction of the robot, moving in the cell along this direction has a null nonholonomic cost.

Two cells  $C_i^\theta$  and  $C_j^\theta$  in  $\mathcal{P}_\theta$  are adjacent if they have a common straight line segment (with direction  $\theta$ ). Let  $p_{i,j}^\theta$  be the middle point of the segment common to the two adjacent cells. The cells being convex any straight line segment  $[p_{i,j}^\theta, p_{i,k}^\theta]$  lies in the cell  $C_i^\theta$ ; it then corresponds to a collision-free path. Computing all these straight line segments provides a roadmap of  $\mathcal{P}_\theta$ . This roadmap is exact in the sense that it solves completely the path planning problem for the robot-polygon moving in translation within  $\mathcal{P}_\theta$ .

Notice that this roadmap method is computationally more expensive than the corresponding cell decomposition method that could have been defined directly from the trapezoidal cells; its interest is to provide a graph of paths (and not a graph of cells) for which the evaluation of the nonholonomic costs is immediate (a path in the graph is also a path in the workspace).

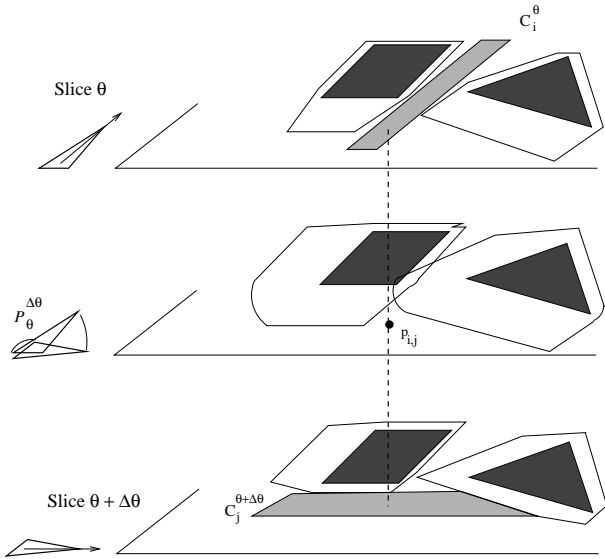


Figure 2: The roadmap's slices

### 3.2 Vertical paths

Let us consider two adjacent slices  $\mathcal{P}_\theta$  and  $\mathcal{P}_{\theta+\Delta\theta}$ . Let  $R_{\Delta\theta}^\theta$  be the generalized polygon<sup>2</sup> swept by the robot when it turns of an angle  $\Delta\theta$  (from a configuration  $(\cdot, \cdot, \theta)$  and by maintaining its position fixed). The collision-free space for the generalized polygon  $R_{\Delta\theta}^\theta$  moving in translation is obtained by computing the Minkowski sum of this polygon with the workspace obstacles (Figure 2).

If  $R_{\Delta\theta}^\theta$  does not overlap the obstacles for some position  $(x, y)$ , then the straight line segment  $[(x, y, \theta), (x, y, \theta + \Delta\theta)]$  is a collision-free vertical path for the robot. Therefore two cells  $C_i^\theta$  and  $C_j^{\theta+\Delta\theta}$  are adjacent if both of them contain a common point belonging to the collision-free space of  $R_{\Delta\theta}^\theta$ .

### 3.3 Roadmap in the configuration space

The roadmap in  $\mathbf{R}^2 \times S^1$  is built by first computing the roadmaps of  $s$  slices  $\mathcal{P}_{\theta_k}$  for  $\theta_k = \frac{2k\pi}{s}$  ( $0 \leq k < s$ ). Two roadmaps belonging to two consecutive slices  $\mathcal{P}_{\theta_k}$  and  $\mathcal{P}_{\theta_{k+1}}$  are linked by vertical paths according to the following method (Figure 2):

- For each cell  $C_i^{\theta_k}$  and each cell  $C_j^{\theta_{k+1}}$ , compute their intersection cell  $C_{i,j}^{\theta_k}$
- For each not empty  $C_{i,j}^{\theta_k}$  cell compute its intersection set with the collision-free space of  $R_{\Delta\theta}^{\theta_k}$ . Select a point  $p_{i,j}$  in each connected component

<sup>2</sup>A generalized polygon is constituted of straight line segments and arcs of a circle.

of the intersection set. Each point  $p_{i,j}$  gives rise to two new configurations  $p_{i,j}^{\theta_k} \in C_i^{\theta_k}$  and  $p_{i,j}^{\theta_{k+1}} \in C_j^{\theta_{k+1}}$ .

- Link each new configuration  $p_{i,j}^{\theta_k}$  (resp.  $p_{i,j}^{\theta_{k+1}}$ ) by an horizontal path to the vertices of the roadmap belonging to the boundary of  $C_i^{\theta_k}$  (resp.  $C_j^{\theta_{k+1}}$ ).
- Link each pair of configurations  $p_{i,j}^{\theta_k}$  and  $p_{i,j}^{\theta_{k+1}}$  by a vertical path.

### 3.4 Analysis and complexity

The final roadmap is then composed of  $s$  layers (i.e., the roadmaps in  $\mathcal{P}_{\theta_k}$ ) linked by vertical paths (corresponding to pure rotations of the robot).

**Completeness:** Each collision-free (and contact-free) path in the configuration space can be approximated by a polygonal line consisting of a finite sequence of horizontal and vertical straight line segments. For  $s$  sufficiently high, there exists an approximation such that the horizontal segments are constrained to lie in the slices  $\mathcal{P}_{\theta_k}$ . Moreover there exists an homotopic path in the roadmap traversing the same cells of  $\mathcal{P}_{\theta_k}$ . The roadmap is then resolution complete, in the sense that the existence of a collision-free path in the configuration space is characterized by the existence of a path in the roadmap when  $s$  tends to infinity.

**Complexity:** Let us evaluate the computational complexity of the construction. We denote by  $n$  the number of vertices of the workspace obstacles;  $s$  is the number of slices; the number of vertices of the robot is sufficiently small to not be taken into account.

- *Cell decomposition of a slice:* The complexity of a sweeping line algorithm in the plane is in  $O(n \log n)$  [1]. Then the  $O(n)$  cells in each  $\mathcal{P}_{\theta_k}$  can be computed in  $O(n \log n)$ . The complexity of the corresponding roadmap is in  $O(n^2)$ .
- *Linking two consecutive slices:* A brute force algorithm computes the intersections  $C_{i,j}^{\theta_k}$  between all the cells  $C_i^{\theta_k}$  and  $C_j^{\theta_{k+1}}$  in  $O(n^2)$ . Their number can be in  $O(n^2)$ . Checking if  $C_{i,j}^{\theta_k}$  intersects the collision-free space of  $R_{\Delta\theta}^{\theta_k}$  can be done in  $O(n^3)$ . Then the total cost of the vertical paths construction is in  $O(n^3)$ .
- Finally the total cost of the roadmap construction is in  $O(sn^3)$ .

## 4 Heuristic search in the roadmap

Once the roadmap  $\mathcal{R}$  is constructed, holonomic paths between two arbitrary start and final configurations  $\mathbf{q}_i$  and  $\mathbf{q}_f$  are computed by first considering an augmented roadmap  $\mathcal{R}'$  obtained by connecting both configurations to the roadmap  $\mathcal{R}$ , and then by searching  $\mathcal{R}'$  for a minimum-cost path.

### 4.1 Connecting the roadmap

Let  $\mathbf{q} = (x_q, y_q, \theta_q)$  be the start (or final) configuration and let  $\mathcal{P}_\theta$  and  $\mathcal{P}_{\theta+\Delta\theta}$  denote the consecutive slices such that the orientation  $\theta_q$  belongs to the interval  $[\theta, \theta + \Delta\theta]$ . The procedure used to connect  $\mathbf{q}$  to the roadmap  $\mathcal{R}$  is the following:

- Check that the generalized polygon  $R_{\Delta\theta}^\theta$  placed at position  $(x_q, y_q)$  does not intersect the obstacles (otherwise return failure).
- Let  $\mathbf{q}_\theta = (x_q, y_q, \theta)$  and  $\mathbf{q}_{\theta+\Delta\theta} = (x_q, y_q, \theta + \Delta\theta)$  denote the projections of  $\mathbf{q}$  onto  $\mathcal{P}_\theta$  and  $\mathcal{P}_{\theta+\Delta\theta}$ . Find the cell  $C_i^\theta$  (resp.  $C_{i'}^{\theta+\Delta\theta}$ ) of the slice  $\mathcal{P}_\theta$  (resp.  $\mathcal{P}_{\theta+\Delta\theta}$ ), containing the configuration  $\mathbf{q}_\theta$  (resp.  $\mathbf{q}_{\theta+\Delta\theta}$ ).
- Link  $\mathbf{q}_\theta$  (resp.  $\mathbf{q}_{\theta+\Delta\theta}$ ) by an horizontal path to all vertices  $p_{i,j}^\theta$  (resp.  $p_{i',j'}^{\theta+\Delta\theta}$ ) of  $\mathcal{R}$  belonging to the boundary of the cell  $C_i^\theta$  (resp.  $C_{i'}^{\theta+\Delta\theta}$ ).
- Link  $\mathbf{q}$  by a vertical path to both intermediate configurations  $\mathbf{q}_\theta$  and  $\mathbf{q}_{\theta+\Delta\theta}$ .

### 4.2 Arc costs in the roadmap

Let  $\mathbf{q}_i = (\mathbf{p}_i, \theta_i)_{i=1,2}$  be two configurations and let  $\theta_{12}$  denote the orientation of segment  $\mathbf{p}_1\mathbf{p}_2$  in  $\mathbf{R}^2$ . We define a distance in  $\mathbf{R}^2 \times S^1$  by:

$$Dist(\mathbf{q}_1, \mathbf{q}_2) = |\theta_1 - \theta_{12}| + d_{euc}(\mathbf{p}_1, \mathbf{p}_2) + |\theta_{12} - \theta_2|$$

with  $d_{euc}$  for the euclidian distance.

Let us consider the path between  $\mathbf{q}_1$  and  $\mathbf{q}_2$  consisting of a pure rotation from  $\mathbf{q}_1$  to the intermediate configuration  $(\mathbf{p}_1, \theta_{12})$ , followed by the horizontal path  $[\mathbf{p}_1\mathbf{p}_2]$  in the slice  $\mathcal{P}_{\theta_{12}}$  and a pure rotation from  $(\mathbf{p}_2, \theta_{12})$  to  $\mathbf{q}_2$ . According to the distance defined above, this path is a shortest feasible path.

The cost of an (horizontal/vertical) segment of the roadmap is defined by  $Dist(\mathbf{q}_1, \mathbf{q}_2)$  and an additional term reflecting (as defined in section 2) the nonholonomic cost  $Cnh(\mathbf{q}_1, \mathbf{q}_2)$  of the segment:

$$Cost(\mathbf{q}_1, \mathbf{q}_2) = Dist(\mathbf{q}_1, \mathbf{q}_2) + \alpha Cnh(\mathbf{q}_1, \mathbf{q}_2)$$

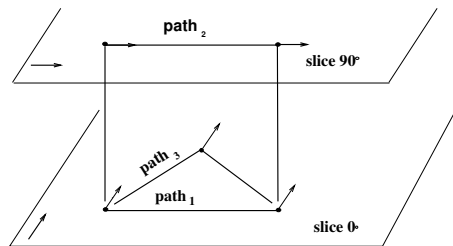


Figure 3: Arc costs in the roadmap

A minimum-cost path between two given configurations is searched in the roadmap with an  $A^*$  algorithm using  $Dist(\mathbf{q}, \mathbf{q}_f)$  as heuristic function.

Let us illustrate the cost on the example depicted in Figure 3. The first arc ( $path_1$ ) corresponds to a crab-wise motion of length  $\ell$  in a direction perpendicular to the orientation of the slice. Its cost is  $\pi + \ell \cdot (1 + \alpha)$  while the total cost of the three arcs ( $path_2$ ) passing through the slice aligned with the direction of the segment would only be  $\pi + \ell$  (this is a shortest path). Also, note that a less direct holonomic path ( $path_3$ ) composed by two horizontal segments in the same slice can be better than  $path_1$  depending on  $\alpha$ . This parameter allows the length increase to be tuned with respect to the improvement of the nonholonomic cost of the holonomic path.

## 5 Results

The algorithms described above for constructing and searching the roadmap have been implemented in C++ on a Sparc 20 workstation.

The example shown in Figure 5 is aimed at illustrating the interest of the proposed algorithms for improving the global efficacy of the two-step motion planning approach for nonholonomic mobile robots.

Figure 4 shows some of the roadmap's slices that were computed to solve this example. For the indicated robot's orientation, the free-space is represented in grey on the left figures, and the result of the trapezoidal decomposition along the main direction of the robot is displayed on the right.

The geometric path obtained for this illustrative example is shown in Figure 5-a. It consists of an alternate sequence of segments (horizontal paths in  $\mathcal{R}$ ) and pure rotations (vertical paths). For example, the segments  $[start-2]$ ,  $[11-12]$ ,  $[12-goal]$  were found in the first slice of Figure 4 and  $[3-4]$ ,  $[5-6]$ ,  $[7-8]$  respectively in the three next displayed slices. Figure 5-a also shows the feasible path derived from this geometric path by the nonholonomic planner [4] of the two-step approach. We simply recall that the method consists

of successively substituting collision-free feasible sub-paths (Reeds and Shepp curves [7]) for portions of the geometric path until the entire path is feasible. The feasible path is then optimized by repeatedly selecting random configurations along the path and by joining them by collision-free feasible curves. For this example, the optimization required less than one second to produce the solution displayed in Figure 5-a which only contains four (necessary) maneuvers in the vicinity of the goal.

This path has to be compared with the solution obtained without considering the nonholonomic cost <sup>3</sup> (ie. for a null value of  $\alpha$ ). As shown in Figure 5-b, the shorter geometric path found in this case leads to an unrealistic feasible path containing a huge number of maneuvers.

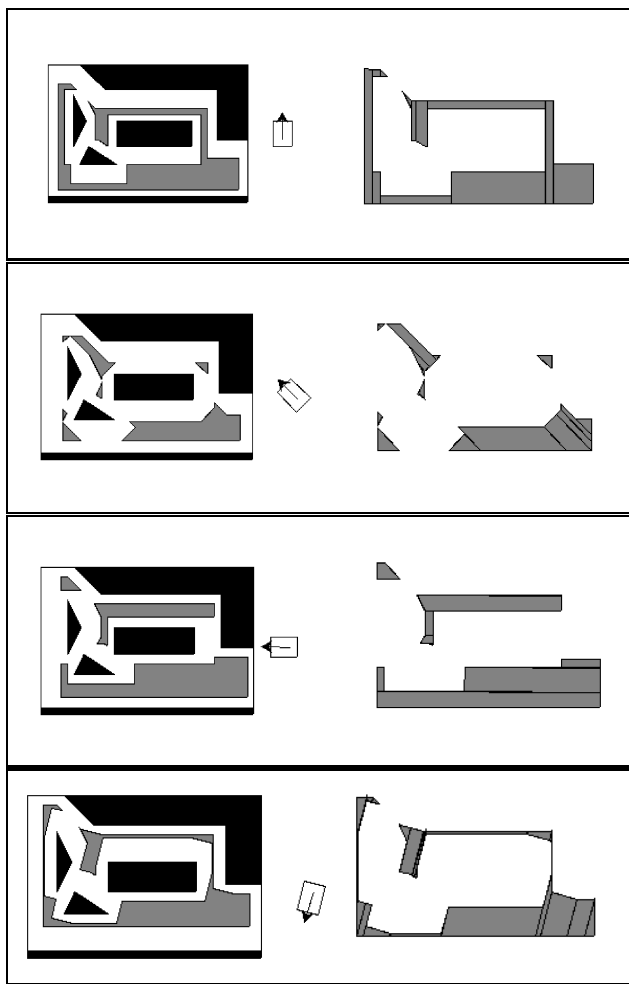


Figure 4: Trapezoidal cell decomposition of free-space's slices

<sup>3</sup>Note that a similar solution would have been produced by previous planners [4, 3] using this two-step approach.

## 6 Conclusion

The two-step approach has previously proven to be an effective scheme for planning with nonholonomic constraints. The main advantage of the method is to limit the combinatorial complexity of the problem to the one of the geometric problem. Its drawback is that

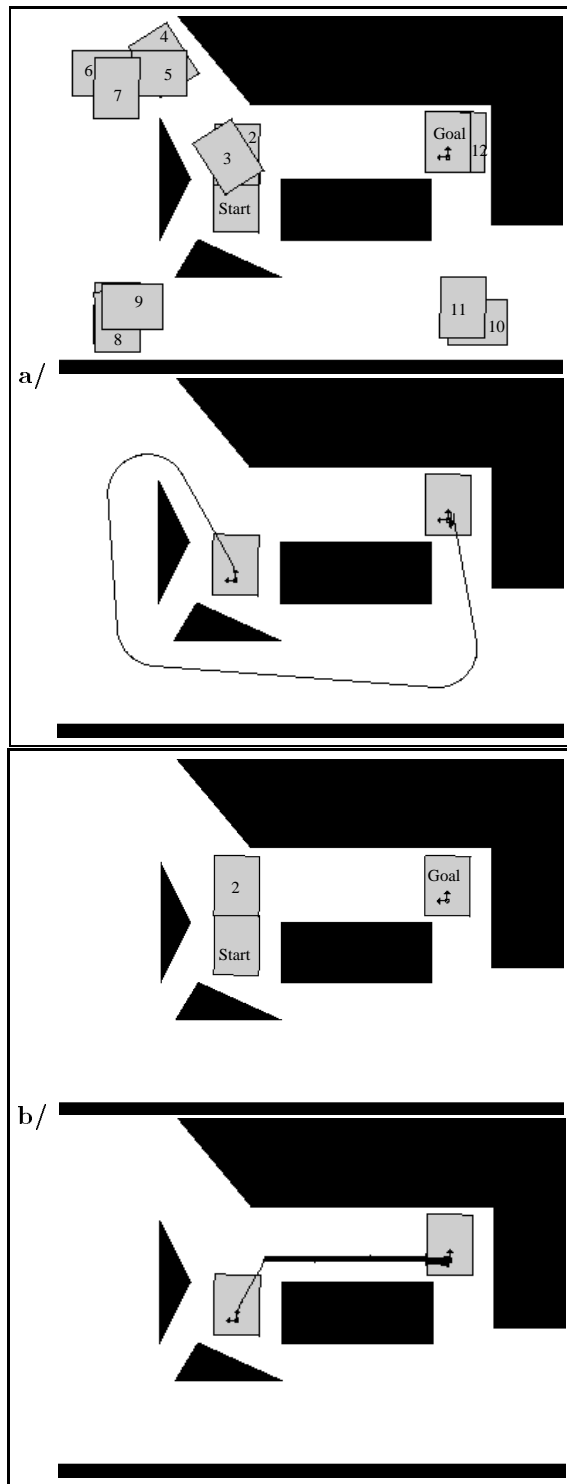


Figure 5: a/ An example of problem solved by the planner compared to b/ the solution obtained without considering the nonholonomic cost...

in some situations, the generated path may include a much larger number of maneuvers than is actually needed. To circumvent this problem, we have introduced in this paper the notion of nonholonomic cost associated to an holonomic path. This cost indicates the difficulty of transforming the path into a feasible one. We have also described a geometric planner based on a configuration space structuring which allows to produce holonomic paths minimizing the non-holonomic cost.

This work has been specifically developed for car-like robots. Possible extensions would be to formalize the nonholonomic cost for other systems (eg. tractor-trailer robots), and also to consider more complex metrics (eg. introducing the clearance to the obstacles) for the evaluation of the nonholonomic cost.

## References

- [1] B. Chazelle, "Approximation and decomposition of shapes", in *Algorithmic and Geometric Aspects of Robotics* J.-T. Schwartz and C.-K. Yap Eds, L.E.A. Pubs, 1987.
- [2] J.C. Latombe, *Robot Motion Planning*, Kluwer Academic Pub., 1991.
- [3] J.C. Latombe, "A Fast Path Planner for a Car-Like Indoor Mobile Robot," in *Ninth National Conference on Artificial Intelligence, AAAI*, pp. 659-665, Anaheim, CA, July 1991.
- [4] J.P. Laumond, P. Jacobs, M. Taix, and R. Murray, "A motion planner for nonholonomic mobile robot", *IEEE Trans. on Robotics and Automation*, 10 (5), 1994.
- [5] J.P. Laumond, S. Sekhavat, and M. Vaisset, "Collision-free motion planning for a nonholonomic mobile robot with a trailer", in *IFAC Symposium on Robot Control*, Capri, September 1994.
- [6] B. Mirtich, and J. Canny, "Using skeletons for nonholonomic motion planning among obstacles," in *IEEE Conf. on Robotics and Automation*, pp. 2533-2540, Nice, France, 1992.
- [7] J. A. Reeds and R. A. Shepp, "Optimal paths for a car that goes both forward and backwards," *Pacific Journal of Mathematics*, 145 (2), 1990.
- [8] S. Sekhavat and J.P. Laumond, "Topological property of trajectories computed from sinusoidal inputs for nonholonomic chained form systems," *IEEE Conf. on Robotics and Automation*, Minneapolis, 1996.

