Robotica (2002) volume 20, pp. 49–58. © 2002 Cambridge University Press
DOI: 10.1017/S0263574701003630 Printed in the United Kingdom

Nonholonomic path planning among obstacles subject to curvature restrictions

Wilson D. Esquivel and Luciano E. Chiang

Mechanical Engineering Department, Pontificia Universidad Católica de Chile, Av. Vicuña Mackenna 4860, Santiago (Chile)
E-mail: lchiang@ing.puc.cl

(Received in Final Form: April 2, 2001)

SUMMARY

This paper addresses the problem of finding a non-holonomic path subject to a curvature restriction, to be tracked by a wheeled autonomous navigation vehicle. This robot is able to navigate in a structured environment, with obstacles modeled as polygons, thus constituting a model based system. The path planning methodology begins with the conditioning of the polygonal environment by offsetting each polygon in order to avoid the possibility of collision with the mobile. Next, the modified polygonal environment is used to compute a preliminary shortest path (PA) between the two extreme positions of the trajectory in the plane (x, y). This preliminary path (PA) does not yet consider the restrictions on the curvature and is formed only by straight line segments. A smoothing process follows in order to obtain a path (PS) that satisfies curvature restrictions which consist basically of joining the straight line segments by circular arcs of minimum radius R (filleting). Finally, the initial and final orientation of the vehicle are accounted for. This is done using a technique we have called the Star Algorithm, because of the geometric shape of the resulting maneuvers. A final complete path (PC) is thus obtained.

KEYWORDS: Robotics; Vehicles; Trajectory generation; Indoors navigation; Curvature restrictions; Four wheeled steering.

1. INTRODUCTION

In this paper a collision-free path planning method for the navigation of a wheeled mobile robot (WMR) is presented. The type of vehicle considered is a Reeds & Shepp's car-like robot whose main characteristics are: (a) a restriction on the minimum turning radius, and (b) a travelling direction defined in the Cartesian plane. The navigation occurs in a structured environment, with the presence of obstacles, which are modeled as polygons. The latter are preferred because of the computational efficiency in their handling.[1]

In most of the literature about autonomous navigation, vehicles are considered having a full turning capacity on the tracking point, and this point is located inside the mobile. Hence most of the existing path planners do not contemplate restrictions in the turning radius of the vehicle. However for AGV-PUC (see Figure 1), an autonomous vehicle developed in the Mechanical Engineering Department of the Catholic University of Chile, a path planner that considers a restriction on the minimum turning radius was essential, given the mechanical configuration of the mobile (Reeds & Shepp's configuration). This has been the motivation of the work described here.

As a first step, an algorithm of controlled amplification of the polygons has been devised in order to prevent collision of the mobile to the polygonal obstacles for any path that requires the vehicle to turn around a vertex. As a second step in order to obtain a preliminary optimal path (PA), the authors have developed an algorithm based on the theories of shortest route by Taha.[2] This algorithm is based on a set of basic matrices, which contain the information of the polygons representing the obstacles in the environment. With this information a preliminary path (PA) of minimum length formed by straight lines segments is generated from the initial to the final position. It may connect with vertices belonging to polygons that cross into the trajectory. This path is of minimum length but neither does it satisfy the curvature restrictions nor does it consider the required initial and final orientations of the mobile. This means that the preliminary optimal path (PA) only connects two positions (x, y), but not the overall configurations (x, y, θ). The preliminary path is smoothed by a filleting operation along all vertices of the path to obtain path PS. The filleting algorithm will always work well because of the provision in the polygon offsetting stage. As a fourth and final step, to satisfy the requirements of initial and final orientation of the

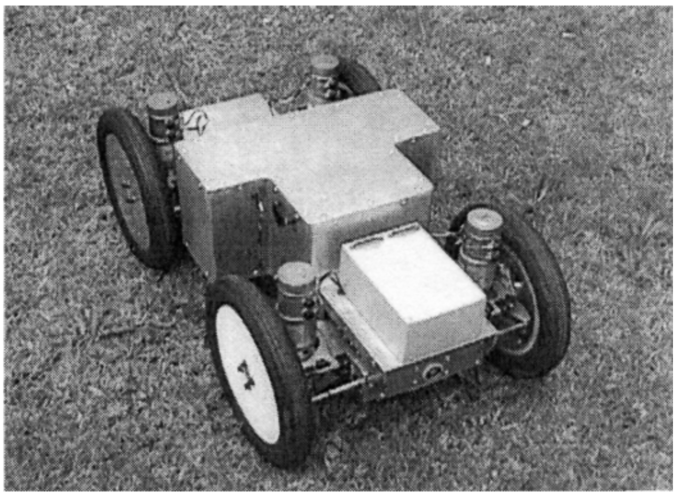

Fig. 1. Mobile robot AGV-PUC.

mobile, we have used what we call the Star Algorithm, which fits the necessary movements into the available space with minimum maneuvering.

With the four previous steps it is possible to obtain a smooth and continuous path, which is optimal or at least nearly optimal, taking the mobile from an initial to a final configuration.

2. BACKGROUND

In order to understand the context of our work, we will first review basic concepts related to autonomous navigation and analyze the main advances in the study of this subject so far.

2.1. Sensor-based or model-based systems?

In autonomous navigation of mobile robots, two different forms to accounting for the environment are distinguished. The first is to generate the information through sensors as the robot navigates, and the second is to use a mathematical model of the environment previously generated and use a reduced set of sensors. In the systems where environment information is based on sensors, the environment is perceived using such sensors as CCD cameras, laser systems, ultrasonic sensors, infrared sensors, tactile sensors, radar systems, as well as other less common such as light sensors. On the other hand, in systems where environment information is based on mathematical models, the environment must be previously known, and represented through mathematical entities, such as polygons.

The speed of current computers allows processing the information coming from the sensors in a short period of time, enabling real time autonomous navigation in unknown or partially known environment. However, the main disadvantage of these systems is the absence of any preliminary idea of obstacle location, therefore, paths towards the final configuration are not optimal. Nevertheless, sensor-based systems have been preferred, due to the flexibility offered by sensors, which allow to have almost real time feedback and also to cope with dynamic variations of the environment. However, a type of intermediate methodology can be used in which an initial mathematical model of the environment is supplied, thus freeing the computer of a substantial amount of work and reducing the number and complexity of sensors required. Sensors are then devoted only to handle the dynamic variations of the environment and to collaborate in the estimation of position to complement typically a system of encoders mounted on traction motors (dead-reckoning navigation). In this way an optimal path could be achieved if the dynamics of the environment allows it. This intermediate methodology is appropriate in indoor environments, where a great part of the surroundings behave in a static manner, i.e. a museum[3] or an industrial production plant.

2.2. Car-like robot or omnidirectional robot?

According to the mechanical characteristics of a mobile robot, it can be classified as an omnidirectional or a car-like robot. Omnidirectional robots are those that have the capability to follow any trajectory, without caring about first order continuity, that is to say, they do not have a minimum turning radius. A large number of well known robots have this capability, as is the case of the commercial robots NOMAD 200, RHINO, SAGE, EDINBURGH R2, GRASMOOR and the platforms PYGMALION, LiAS and HILARE. On the other hand, car-like mobile robots are those with a minimum turning radius due to restrictions in their traction and direction mechanisms.

2.3. Research on sensor-based systems

In this case the robot obtains the information of the environment in real time through sensors. In this type of robots there has been an important effort by researchers. For example, Oriolo et al.[4–7] and Khatib[8] work in obtaining a map of the environment in real time using ultrasonic sensors. Murray and Jennings[9] obtain a map using stereo vision. On the other hand, Song and Tang[10] combine the ultrasonic perception with a CCD camera. In project LiAS[11] the information coming from three types of sensors is used to obtain a precise model of an unknown environment. This model is subsequently used by a real time navigation algorithm. Stentz[12] carried out important work in partial or completely unknown environments. He performed a scanning of the environment with a range-finder laser to detect obstacles and then form a matrix that contains the environment model.

Stentz has worked in the development of the D* algorithm, which generates optimal free-collisions trajectories among obstacles, Yahja et al.[13] use this algorithm for the navigation of a vehicle operating in a vast dimension outdoor environment. Olson[14] presents a probabilistic study of self-localization comparing aspects of a map generated in real time with a previously known map. Fox et al. use a Markov localization algorithm to obtain the robot's probabilistic localization. With the help of special filters they can introduce the dynamics of environment. This technique was implemented in the RHINO robot,[15] which gave interactive tours to visitors of the "Deutsches Museum Bonn". Another previous work that uses successfully a Markovian localization technique to navigate in a partially observable environment is the one developed by Simmons and Koenig.[16] Similar approach is followed by Thrun et al.,[17,18] Vlassis and Tsanakas[19] and Yamauchi et al.[20] who face the problem of robot localization in an unknown environment. Nehmzow et al.[21,22] have used the robot NOMAD 200 to perform different tasks, while obtaining the robot localization through perception. For this purpose they have used a sensor configuration that combines sixteen ultrasonic sensors, sixteen infrared sensors, twenty tactile sensors, a flux gate compass and a monochrome CCD camera to obtain a map of multiple resolution. In addition to the previous works, Nehmzow[23] has experimented in robot navigation using a differential light compass. This compass is implemented with four to six light sensors. This technique has been proved in robots GRASMOOR and EDINBURGH R2.

2.4. Research on model-based systems

These systems use a reduced set of sensors to complete the environment information in the form of a previously generated mathematical model. To this category belongs the work carried out by Barraquand and Latombe,[24] who propose an approach to robot path planning that consists in the building and searching of a graph connecting the local minimal of a potential function defined over the robot's configuration space, that is to say, the minimal is found in the points farther away from the borders of the obstacles and of the limits of the work space. This planner is fast and can solve problems for robots with many more degrees of freedom. In another pioneering work, Singh and Wagh[25] presented a trajectory planning algorithm that uses the intersecting nodes between rectangular free convex areas in a specified environment with obstacles. This algorithm generates routes of approximately minimum length formed by the union of these nodes through straight lines.

2.5. Research on car-like robots

The study of the shortest paths for car-like robots has already a long history. It was first addressed without the presence of obstacles. The pioneering results were achieved by Dubins,[26] who characterized the shortest path as a set of at least three contiguous geometric entities. Two types of entities are used, either a straight line segment or a circular arc. The robot was considered to move always forwardly. Reeds and Shepp[27] have suggested that the feasible manifold of shortest paths is formed by 48 different cases when moving both forward and backward is included. Sussmann and Tang[28] reduced this family to 46 canonical paths. Souères and Laumond[29] synthesized the number of minimal paths by partitioning the manifold into cells reachable by only one type of minimal path among the 46 paths. They also computed the exact shape of the shortest path metric.[30] Souères et al.[31] characterize the shape of the shortest path from an initial configuration to a point in the plane for the Reeds and Shepp's car. On the other hand, Bui et al.[32] used a similar approach for the synthesis of the shortest path from a fixed configuration to a point in the plane for the Dubins's car. Mountarlier et al[33] explored general analytic tools to compute shortest paths to some sub-manifolds of configurations in the plane.

The problem of computing the shortest path for a car-like robot in the presence of obstacles is very difficult. The existence of a shorter collision-free path for a Reeds and Shepp's car is not guaranteed (Desaulniers).[34] In their work, Reif and Wang[35] discuss the complexity of the problem of obtaining the shortest path with constrained curvature in two dimensions. Recent results solve the problem of computing shortest path among moderate or well-behaved obstacles, which are distributed in the environment in ordinate form.[36,37]

A large number of research efforts have addressed the simpler problem of finding collision-free path without consideration of minimum length. Instead they focus on obtaining a smooth continuous path in the presence of curvature restrictions for either Reeds & Shepp's or Dubins's car.[38–42]

Among the most recent works that address the case of car-like robots is that of Vendittelli et al.[43] Their main contribution has been a methodology to compute the shortest nonholonomic path for a point on a mobile that is moving amidst polygonal obstacles. Geometric algorithms are used to compute the instantaneous shortest path from a given configuration to a set of polygonal obstacles for both Reeds & Shepp's and Dubins's car. A similar approach is followed by Laumond, Jacobs, Taïx and Murray.[44] They obtain a fast and exact planner, based upon recursive subdivision of a collision-free path generated by a lower-level geometric planner that ignores the motion constraints thus generating a holonomic path. Then, the trajectory is optimized to obtain a nonholonomic path that is of near-minimal length. This study was done in the context of the HILARE mobile robot project.[45]

Wang, Linnett and Roberts[46,47] investigate the feasible movements of a three wheel vehicle with a steering angle limitation, and point out the types of kinematic restrictions that should be taken into account in path planning. Their work concludes with a generic formulation of the problem of path planning, and the presentation of a generic algorithm, which is used to find an approximate optimal global path and also to evaluate potential collision with polygonal obstacles.

On the other hand, Wang and Cartmell[48] have developed algorithms to perform parallel transfer maneuvers, such as reverse parking, moving off, negotiating a stationary obstacle, overtaking a moving vehicle, and changing lanes. These maneuvers are designed using function fitting approaches, which generate curves that satisfy the boundary conditions. The type of curves used are: a quintic polynomial, a cubic polynomial and a triangular function. Murray and Sastry[49] proposed nonholonomic motion planning using sinusoids for chained systems. Esquivel and Chiang[50] analyze three types of curves for trajectory planning: a parametric curve, an improved cubic polynomial and a lineal-quadratic curve. More recently Esquivel and Chiang[51] present a trajectory planner amidst polygonal obstacles using Dubins's curves.

3. NEW METHODOLOGY TO OBTAIN A NEAR OPTIMAL PATH SUBJECT TO RESTRICTION IN THE RADIUS OF CURVATURE

The problem of finding a path subject to a limitation in the radius of curvature can be solved in three stages. The first stage consists in obtaining an environment with obstacles adapted to the mobile. The second stage consists in obtaining a smooth path that connects the initial and final position while taking into account the curvature restriction, and the third part consists in taking into account the initial and final orientation of the mobile.

3.1. Offset of the polygonal environment

We begin with an initial mathematical model of the environment, in which obstacles are modeled as polygons. As any given vehicle may have different dimensions, the polygonal environment must be adapted to include an offset that will ensure that no point on the vehicle will touch any present obstacle. The required offset is in general different for every obstacle because it is dependent on the geometry of the mobile as well as the geometric shape of the obstacle. To compute the offset for a given polygon it is first required

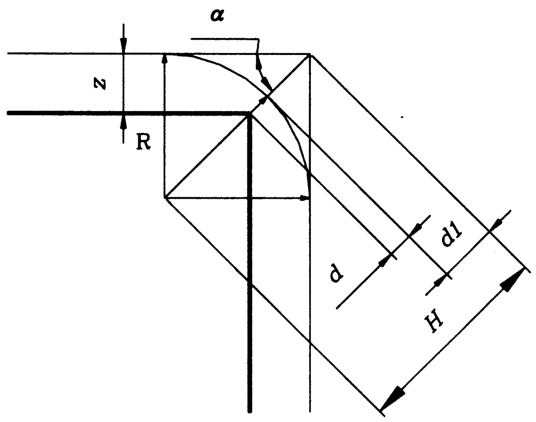

Fig. 2. Vertex with offset.

to know the angle of the sharpest convex vertex in the polygon. As shown in Figure 2 the angle in this vertex can be acute, straight or obtuse. The arc of minimal radius that can be followed by the mobile to skirt this vertex is also shown. Now, to maintain the mobile sufficiently far away from the vertex and its sides, the arc should pass at a certain distance *d* of the vertex. Distance *d* is related to the magnitude of the offset (*z*). Distance *d* depends on the geometry of the mobile and is computed using the width and length of the vehicle. Thus,

$$d = \frac{1}{2}\sqrt{Length^2 + Width^2} \qquad (1)$$

From the previous figure we have that:

$$H = \frac{R}{\sin(\alpha)} \qquad (2)$$

$$d_1 = H - R \qquad (3)$$

$$z = (d + d_1)\sin(\alpha) \qquad (4)$$

So replacing equation 2 in 3 and the result to equation 4 we obtain.

$$z = R - R\sin(\alpha) + d\sin(\alpha) \qquad (5)$$

Therefore, the offset z depends on the minimum turning radius of the mobile (R), on the geometric constant d and on the angle α, of the sharpest vertex of the polygon. With this value of z it is assured that any vertex of the polygon can be turned around with a minimum curvature radius R. The initial polygonal environment is then modified by offsetting each polygon by the corresponding z value. Figure 3 shows an example for an environment of three polygons adapted with an offset for a test vehicle.

3.2. *Generation of a smooth path*

We now wish to obtain a smooth path that takes into account the turning restriction in a vehicle, that is to say its minimum turning radius *R*. A vehicle with such a restriction is known as a car-like robot. To solve this problem a preliminary path (PA) is needed which is obtained by modification of an algorithm developed by Taha for construing critical routes. This algorithm and its adaptation to the case in hand is described in Appendix I.

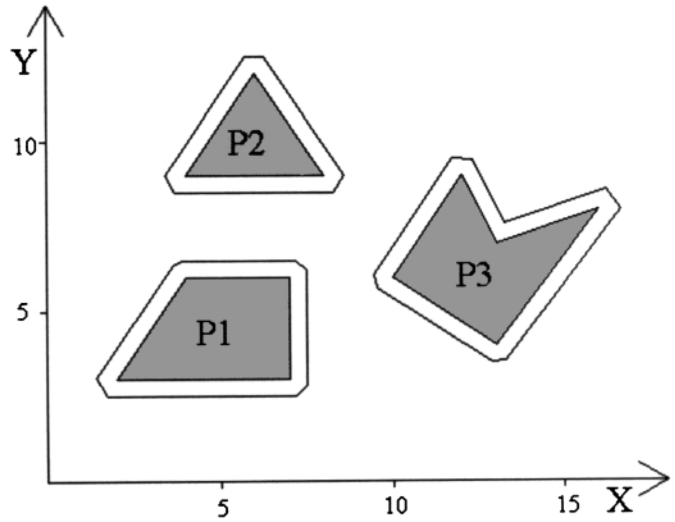

Fig. 3. Environment of three polygons with offset.

Figure 4 shows the preliminary path (PA) for a particular task obtained by Taha's modified algorithm. Note that the resulting PA path is formed by piecewise straight lines segments.

A smooth path can be obtained next by replacing each vertex in the preliminary path PA by a fillet of radius *R*, which is the minimum turning radius of the vehicle. The computation carried to offset the polygons assures that the fillet inserted fits in place of the replaced vertex. The smooth path obtained is called PS.

3.3. *Initial and final orientations*

3.3.1. R-Geodesic approach. Now in the third step of our methodology we must solve the problem of the initial and final orientations. This problem arises when the vehicle is not able to follow path PS from its initial position, because the vehicle is in a different starting orientation. Therefore the mobile must be reoriented without colliding against any neighboring obstacles.

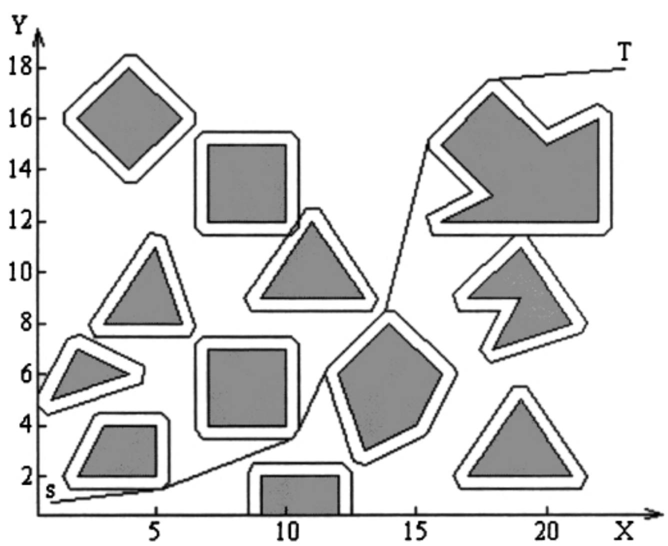

Fig. 4. Preliminary shortest path (PA) obtained by means of the extension of the method of Taha.

https://doi.org/10.1017/S0263574701003630 Published online by Cambridge University Press

If enough space exists, the reorientation maneuver can be performed through an R-geodesic, which is the curve of smallest length that connects two poses in the plane subject to curvature restrictions (Dubins). As shown in Figure 5, there are two possible R-geodesics to connect the initial configuration (x_i, y_i, θ_i) with path PS, whose starting orientation is θ_{path}. These R-geodesics are composed of an arc segment and a straight line segment. It is clear from Figure 5 that it is preferable to select the R-geodesic 1 because it is the shortest. When this is not possible because a collision occurs, then R-geodesic 2 can be selected.

3.3.2. Star method. If there is no space available to follow any of the R-geodesic curves described above, we can resort to the so called Star algorithm developed by the authors. This algorithm consists in finding the appropriate angle (α) and the necessary number of circular arc steps (n) to go from the actual initial orientation θ_i to the initial orientation (θ_{path}) of path PS. A sequence of forward and backward movements is defined, resembling a star with cusps. The angle α should be of such magnitude to allow completion of the reorientation maneuver in an integer number of steps.

The Star Method begins by computing distance d_{min} which is the smallest distance between the tracking point of the mobile and either the nearest obstacle or the physical limits of the work space.

Distance d_{min} is the minimum distance to maneuver, obtained through a simple scanning of the environment, using mathematical expressions like the minimum distance between a point and a straight line.

Now, from Figure 6 we can deduce the following expression

$$\alpha_{min} = 2\tan^{-1}\left(\frac{R}{d_{min}}\right) \tag{6}$$

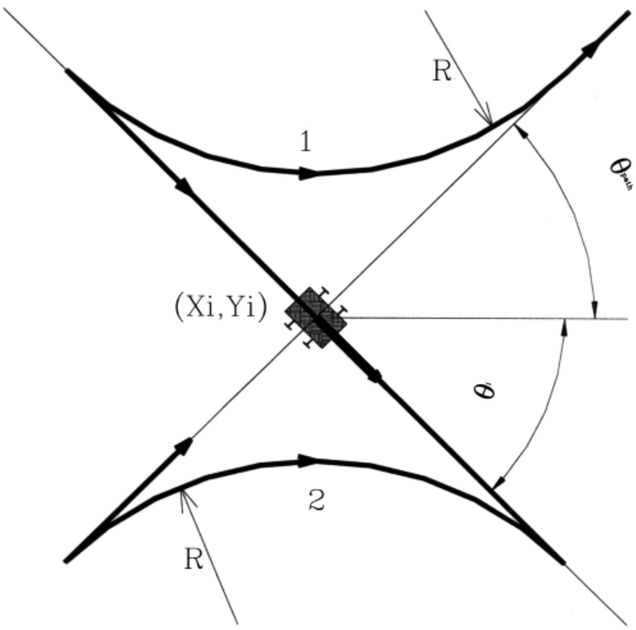

Fig. 5. Two R-geodesic to match initial configuration with path PS.

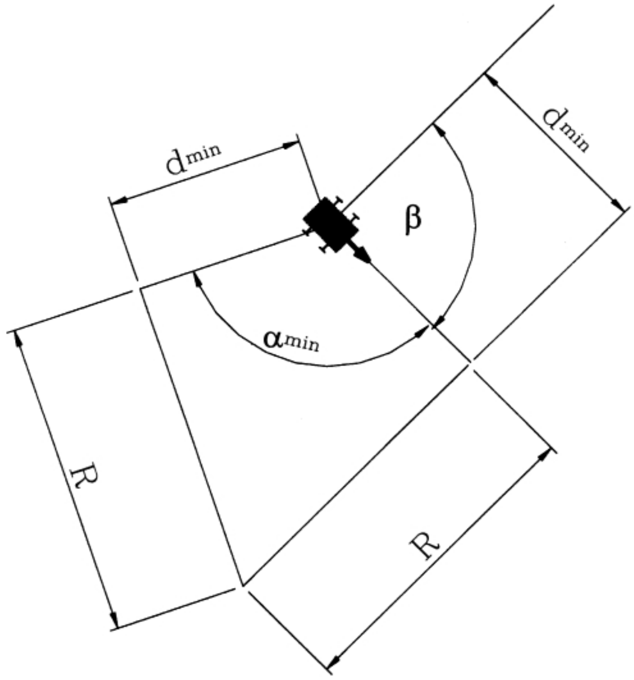

Fig. 6. Angle α_{min} necessary to obtain a distance d_{min}.

On the other hand, in appendix II we prove that the value of the angle α is equivalent to

$$\alpha = \alpha_i + \frac{2^n - 1}{2^n}\beta \tag{7}$$

In the above expression n is the number of circular arc pairs of the type shown in Figure 7, that are necessary to take the mobile from one orientation to another. β is the absolute angle between the actual initial orientation of the mobile

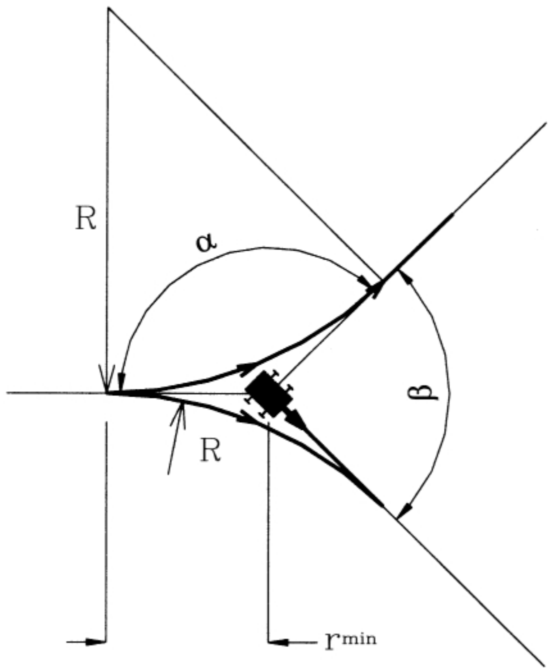

Fig. 7. Connection maneuver between the vehicle's position and the path for $n = 1$.

https://doi.org/10.1017/S0263574701003630 Published online by Cambridge University Press

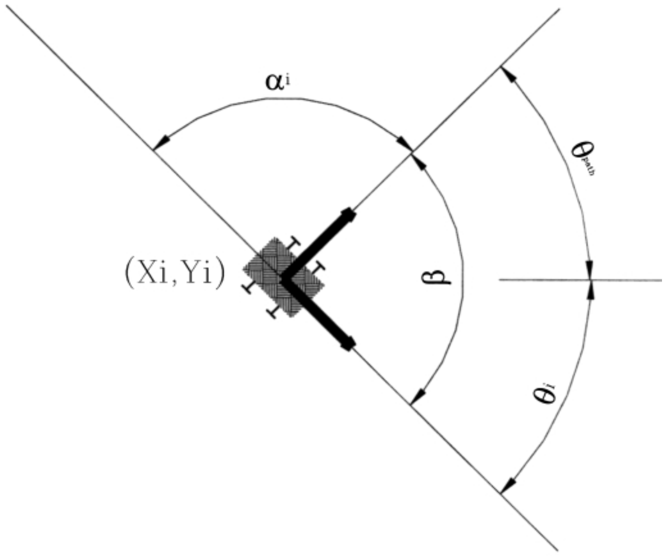

Fig. 8. Absolute angles α_i and β.

and the initial orientation of path PS, and α_i is the complement of β, as is shown in Figure 8. Then replacing α for $\alpha_{\min}$ and isolating n, we have:

$$n=\frac{\ln\left[\dfrac{1}{1-\dfrac{\alpha_{\min}-\alpha_i}{\beta}}\right]}{\ln[2]} \tag{8}$$

The value of n is not necessarily an integer, therefore it is rounded to the next highest integer. Once the new value of n is obtained, angle α is updated from equation (7). With new angle α and by virtue of equation (9) a new and definitive minimum distance $r_{\min}$ is computed, which is always smaller or equal than the critical distance $d_{\min}$. In Figure 7 we can see graphically the resulting maneuvers for n=1.

$$r_{\min}=\frac{R}{\tan\left(\dfrac{\alpha}{2}\right)} \tag{9}$$

In general, to reach the final orientation, n pairs of circular arcs around the starting point are followed. Each pair of arcs will reduce in the n-th part the angle β (angle between the orientation of the mobile and the initial orientation of path PS).

In Figure 9 the methodology described is used to generate a complete path (PC) for one obstacle and initial configuration (2, 2, $-\pi/2$) to final configuration (4, 4, 0°).

In Figure 10 the details of the initial orientation maneuvers for the path of Figure 9 is shown. In this case for $\theta_{path}=109.544°$, $\theta_i=-90°$, $\beta=160.456°$, $\alpha_i=19.544°$, $d_{\min}=0.82$ m, $R=1$ m, $\alpha_{\min}=101.296°$, $n=2$, $\alpha=138.75°$, $r_{\min}=0.39$ m.

In Figure 11 the path obtained for a test vehicle in the environment of Figure 4 is shown. Now in this case the shortest path PA has been smoothed to take into account the curvature restriction. The configurations to connect are (1, l, 0°) and (23, 18, 0°).

In certain cases, before implementing the Star Method, a controlled movement along a straight line can be followed

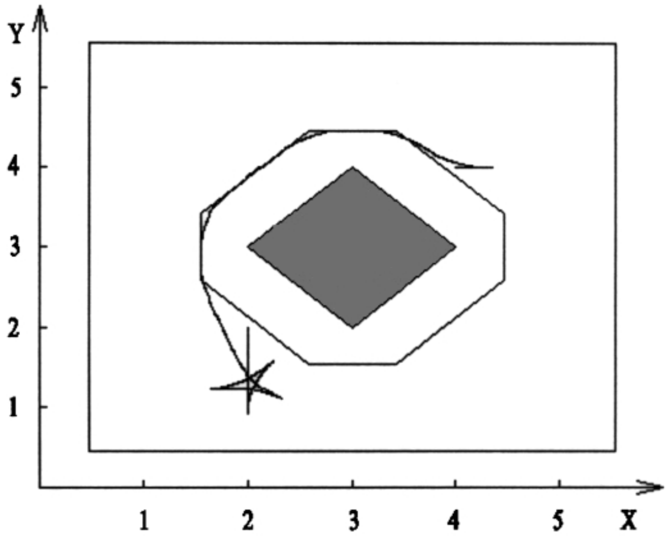

Fig. 9. Example of a path obtained with the proposed path planner.

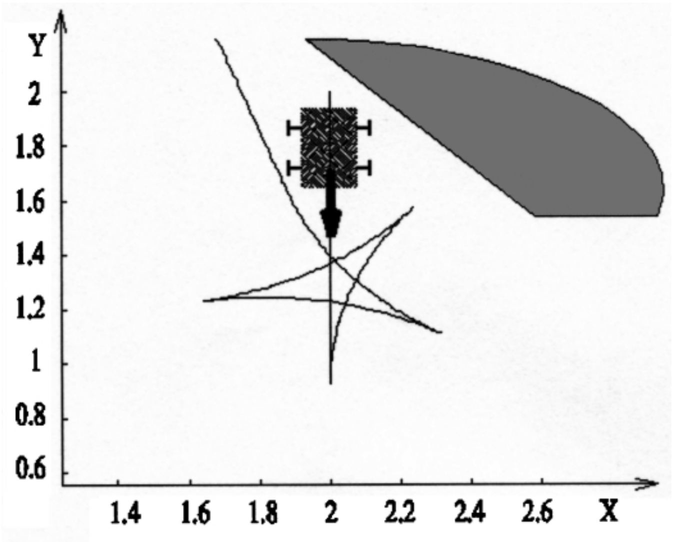

Fig. 10. Detail of initial orientation maneuver for $n=2$.

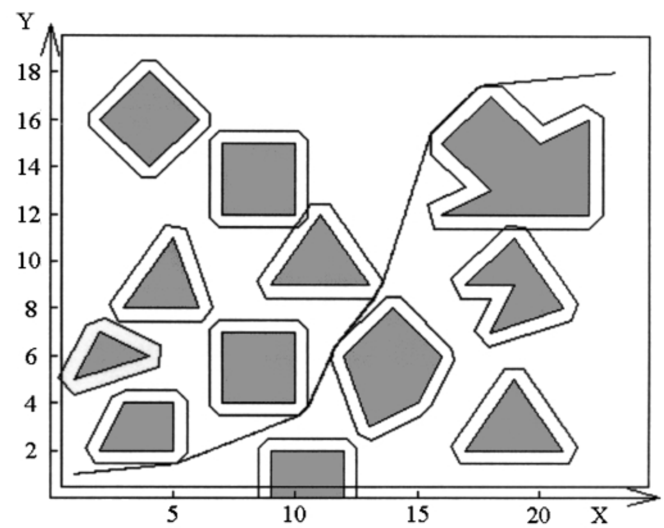

Fig. 11. Shortest path that satisfies curvature restriction.

https://doi.org/10.1017/S0263574701003630 Published online by Cambridge University Press

in order to center the mobile inside the available maneuvering space. In this way a larger value of d_{min} is obtained which facilitates the initial orientation maneuvers. However this could be done at the expense of increasing significantly the resulting path length. Whenever the movement described above is performed, the mobile changes its initial position, and this new position becomes the starting point to perform the reorientation maneuver.

4. CONCLUSIONS

One of the most difficult problems in autonomous navigation path planning is that of obtaining a path subject to curvature restrictions. Indeed, it has been demonstrated that the problem of finding the shortest path in presence of obstacles does not always have an optimal solution. In such cases it is desirable to have a planner that is able to generate a feasible path of acceptable length. On the other hand, the problem of unlocking the mobile from a position has not been studied thoroughly since most of the investigators assume the presence of well distributed obstacles, with enough space to maneuver without colliding.

In this paper we have presented a method to obtain paths subject to curvature restrictions to serve as reference in the navigation of a WMR in a modeled-based environment, with obstacles modeled as polygons. In this method an initial algorithm is used, which perform a controlled amplification of the obstacles of environment, using for this purpose the characteristic geometric dimensions and the turning radius of the mobile, and the angle of the sharpest vertex in every polygon. A second algorithm presented in this article gives a path of minimal length, which we call PA. This second algorithm is based on the minimal route theories developed by Taha. Now the path PA must be smoothed to account for the curvature restrictions on the vehicle path which are a consequence of the vehicle limitation in the turning radius. For this purpose we present a third algorithm in which PA is successively modified by filleting so that the vehicle can skirt an obstacle while maintaining itself at a desirable minimal distance. Hence a smoothed polyline is obtained, which we call PS.

To account for the initial and final orientations of the mobile, a novel algorithm is presented called the Star Algorithm because of the shape of the maneuvers involved. Hence collision-free maneuvers can be specified cases where there is an initial blockage of the mobile or the final orientation is not directly reachable. For this purpose a minimal maneuvering distance is determined which is the basis to compute the necessary number of movements to achieve the desired initial or final orientation of the vehicle.

All four previously mentioned algorithms form part of a path planning methodology of great speed and relatively low computational complexity. A path called PC is obtained which is a path of minimum length if a solution exists and of approximately minimum length otherwise.

This planner has been developed within the frame of project AGV_PUC (Autonomously Guided Vehicle Pontificia Universidad Catolica). This project consists in the development of a WMR (see Figure 1), that uses a navigation system based on a mathematical model. For its localization a dead reckoning is used. Information is communicated through wireless modems between a host computer and the on-board computer of the mobile. Among future tasks in this project is the implementation of a ring of 16 ultrasonic sensors to help in the tasks of localization of the mobile, through triangulation.

ACKNOWLEDGEMENTS

The financial support of the National Fund of Scientific and Technological Development of project FONDECYT 2990071 (Trajectory Generation for the Autonomous Navigation of a Mobile Robot) is gratefully acknowledged.

References

1. M.E. Mortenson, *Geometric Modeling* (John Wiley & Sons, 1985).
2. H.A. Taha, *Operations Research, An Introduction* (Macmillan Publishing Co., 5th Ed. (1992).
3. I. Nourbakhsh, J. Bobenage, S. Grange, R. Lutz, R. Meyer and A. Soto, "An Affective Mobile Educator With a Full-Time Job", *Artificial Intelligence* **114**(1–2), 95–124 (October, 1999).
4. G. Oriolo, G. Ulivi and M. Vendittelli, "Real-Time Map Building and Navigation for Autonomous Robots in Unknown Environments", *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics* **28**(3), 316–333 (June, 1998).
5. G. Oriolo, G. Ulivi and M. Vendittelli, "On-Line Map Building and Navigation for Autonomous Mobile Robots", *Proc. 1995 IEEE International Conference on Robotics Automation*, Nagoya, Japan (May, 1995) pp. 2900–2906.
6. F. Gambino, G. Oriolo and G. Ulivi, "A Comparison of Three Uncertainty Calculus Techniques for Ultrasonic Map Building", *Proc. 1996 SPIE International Symposium on Aerospace/Defense Sensing Control*, Orlando, FL (1996) pp. 249–260.
7. G. Fortarezza, G. Oriolo, G. Ulivi and M. Vendittelli, "A Mobile Robot Localization Method for Incremental Map Building and Navigation", *Proc. 3rd International Symposium on Intelligent Robotic Systems*, Pisa, Italy (1995) pp. 57–65.
8. O. Khatib, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots", *Int. J. Robot. Res.* **5**(1), 90–98 (1996).
9. D. Murray and C. Jennings, "Stereo Vision Based Mapping and Navigation for Mobile Robots", *Proc. IEEE International Conference of Robotics and Automation*, Albuquerque, NM, USA (April, 1997) **Vol. 2**, pp. 1694–1699.
10. K.-T. Song and W.-H. Tang, "Environment Perception for a Mobile Robot Using Double Ultrasonic Sensors and a CCD Camera", *IEEE Transactions on Industrial Electronics* **43**(3), 372–379 (June, 1996).
11. J. Vandorpe, H. Van Brussel and H. Xu, "LiAS: A Reflexive Navigation Architecture for an Intelligent Mobile Robot System", *IEEE Transactions on Industrial Electronics* **43**(3), 432–440 (June, 1996).
12. A. Stentz, "Optimal and Efficient Path Planning for Partially-Known Environments", *Proc. IEEE International Conference on Robotics and Automation*, San Diego, CA, USA (May, 1994) **Part 4**, pp. 3310–3317.
13. A. Yahja, A. Stentz, S. Singh and B.L. Brumitt, "Framed-Quadtree Path Planning for Mobile Robots Operating in Sparse Environments", *Proc. IEEE Conference on Robotics and Automation* (*ICRA*), Leuven, Belgium (May, 1998) **Vol. I**, pp. 650–655.
14. C.F. Olson, "Probabilistic Self-Localization for Mobile Robots", *IEEE Transactions on Robotics and Automation* **16**(1), 55–66 (February, 2000).
15. D. Fox, W. Burgard, S. Thrun and A.B. Cremers, "Position Estimation for Mobile Robots in Dynamic Environments", *Proc. of the Fifteenth National Conference on Artificial*

Intelligence (*AAAI-98*), Madison, Wisconsin (1998) pp. 983–988.
16. R. Simmons and S. Koenig, "Probabilistic Robot Navigation in Partially Observable Environments", *Proc. of the Fourteenth International Joint Conference on Artificial Intelligence*, Montreal, Canada (1995) **Vol. 2**, pp. 1660–1667.
17. S. Thrun, "An Approach to Learning Mobile Robot Navigation", *Robotics and Autonomous Systems* **15**, 301–319 (1996).
18. S. Thrun, D. Fox and W. Burgard, "Probabilistic Mapping of an Environment by a Mobile Robot", *Proc. IEEE International Conference on Robotics and Automation*, Leuven, Belgium (May, 1998) **Vol. 2**, pp. 1546–1551.
19. N.A. Vlassis and Tsanakas, "A Sensory Uncertainty Field Model for Unknown and Non-stationary Mobile Robot Environments", *Proc. IEEE International Conference on Robotics and Automation*, Leuven, Belgium (May, 1998) **Vol. I**, pp. 363–368.
20. B. Yamauchi, A. Schulz and W. Adams, "Mobile Robot Exploration and Map-Building with Continuous Localization", *Proc. IEEE International Conference on Robotics and Automation*, Leuven, Belgium (May, 1998) **Vol. 4**, pp. 3715–3720.
21. U. Nehmzow and C. Owen, "Robot Navigation in the Real World: Experiments with Manchester FortyTwo in Unmodified, Large Environments", *Robotics and Autonomous Systems* **33** 223–242 (2000).
22. T. Duckett and U. Nehmzom, "Mobile Robot Self-Localisation and Measurement of Performance in Middle-Scale Environments", *Robotics and Autonomous Systems* **24**, 57–69 (1998).
23. U. Nehmzov, "Animal and Robot Navigation", *Robotics and Autonomous Systems* **15**(1–2), 71–81 (1995).
24. J. Barraquan and J.C. Latombe, "Robot Motion Planning: A Distributed Representation Approach", *Int. J. Robotics Research* **10**(6), 51–72 (1991).
25. J.S. Singh, M.D. Wagh, "Robot Path Planning Using Intersecting Convex Shapes: Analysis and Simulation", *IEEE Journal of Robotics and Automation* **RA-3**(2), 101–108 (April, 1987).
26. L.E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents", *American Journal Mathematics* **79**, 497–516 (1957).
27. J.A. Reeds and R.A. Shepp, "Optimal Paths for a Car that Goes Both Forward and Backward", *Pacific J. Mathematics* **145**(2), 367–393 (1990).
28. H.J. Sussmann and W. Tang, "Shortest Path for the Reeds-Shepp Car: A Worked Out Example of the Use of Geometric Techniques in Nonlinear Optimal Control", *Tech. Rep. SYCON-91-10* (Rutgers Univ., New Brunswick, NJ, 1991).
29. P. Souères and J.-P. Laumond, "Shortest Path Synthesis for a Car-Like Robot", *Proc. European Control Conference*, Groningen (June, 1993) pp. 570–577.
30. J.-P. Laumond and P. Souères, "Metric Induced by the Shortest Paths for a Car-Like Mobile Robot", *Proc. IEEE Int. Conf. Intell. Robots Systems* 1299–1303 (July, 1993).
31. P. Souères, J.-Y. Fourquet and J.P. Laumond, "Set of Reachable Position for a Car", *IEEE Transaction on Automation Control* **39**, 1626–1630 (August, 1994).
32. X.-N. Bui, P. Souères, J.-D. Boissonnat and J.-P. Laumond, "Shortest Path Synthesis for Dubins Nonholonomic Robots", *Proc. IEEE International Conference on Robotics and Automation*, San Diego, CA, USA (May, 1994) **pt. 1**, pp. 1–7.
33. P. Moutarlier, B. Mirtich and J. Canny, "Shortest Paths for a Car-Like Robot to Manifolds in Configuration Space", *Int. J. Robot Res*. **15**, 36–60 (1996).
34. G. Desaulniers, "On Shortest Paths for Car-Like Robot Maneuvering Around Obstacle", *Robot. Auton. Syst*. **17**, 139–148 (1996).
35. J. Reif and H. Wang, "The complexity of the Two Dimensional Curvature-Constrained Shortest-Path Problem", *Robotics: Proc. 3rd Workshop on the Algo. Found. Of Robotics* (P.K. Agarwal, L.E. Kavraki and M. Mason, eds) (A.K. Peters, Natick, MA, Wellesley, 1998) pp. 49–58.
36. P.K. Agarwal, P. Raghavan and H. Tamaki, "Motion Planning for a Steering-Constrained Robot Through Moderate Obstacles", *Proc. 27th Annu. ACM Symp. Theory Comput*. (1995) pp. 343–352.
37. J.D. Boissonnat and S. Lazard, "A Polynomial-Time Algorithm for Computing a Shortest Path of Bounded Curvature Amidst Moderate Obstacle", *Proc. 12th Annu. ACM Sympos. Comput. Geom*. (1996) pp. 242–251.
38. J. Sellen, "Planning Path of Minimal Curvature", *IEEE International Conference on Robotics and Automation*, Nagoya, Japan (May, 1995) **Vol. 2**, pp. 1976–1982.
39. H. Wang and P.K. Agarwal, "Approximation Algorithms for Curvature Constrained Shortest Paths", *Proc. 7th ACM-SIAM Sympos. Discrete Algorithms* (1996) pp. 409–418.
40. T. Fraichard, "Smooth Trajectory Planning for a Car in a Structured World", *IEEE Int. Conf. Robot. Automat*. (April, 1991) **Vol. 7**, pp. 318–323.
41. A. Scheuer and T. Fraichard, "Continuous-Curvature Path Planning for Car-Like Vehicle", *Proc. IEEE-RSJ International Conference on Intelligent Robots and Systems*, Osaka, Japan (November, 1997) **Vol. 3**, pp. 1304–1311.
42. A. Shkel and V. Lumelsky, " Curvature-Constrained Motion Within a Limited Workspace", *Proc. IEEE International Conference on Robotics and Automation*, Albuquerque, NM, USA (April, 1997) **Vol. 2**, pp. 1394–1399.
43. M. Vendittelli, J.-P. Laumond and C. Nissoux, "Obstacle Distance for Car-Like Robots", *IEEE Transactions on Robotics and Automation* **15**(4), 678–691 (1999).
44. J.P. Laumond, P.E. Jacobs, M. Taïx and R.M. Murray, "A Motion Planner for Nonholonomic Mobile Robots", *IEEE Transactions on Robotics and Automation* **10**(5), 577–593 (1994).
45. F. Lamiraux, S. Sekhavat and J.-P. Laumond, "Motion Planning and Control for Hilare Pulling a Trailer", *IEEE Transactions on Robotics and Automation* **15**(4), 640–652 (August, 1999).
46. Y. Wang, J.A. Linnett and J.W. Roberts, "Motion feasibility of a wheeled vehicle with a steering angle limit", *Robotica* **12, Part 3**, 217–226 (1994).
47. Y. Wang, J.A. linnett and J.W. Roberts, "Kinematics, kinematic constraints and path planning for wheeled mobile robots", *Robotica* **12**, **Part 5**, 391–400 (1994).
48. Y. Wang, and M.P. Cartmell, "Autonomous vehicle parallel parking design using function fitting approaches", *Robotica* **16**, **Part 2**, 159–170 (1998).
49. R.M. Murray and S.S. Sastry, "Nonholonomic Motion Planning: Steering Using Sinusoids", *IEEE Transactions on Automatic Control* **38**(5), 700–716 (1993).
50. W.D. Esquivel and L.E. Chiang, "Planeación de Rutas Para Robot Móviles", *Eighth Chilean Congress of Mechanical Engineering*, University of Concepción, Chile, **1**, 439–444 (1998).
51. W.D. Esquivel and L.E. Chiang, "Planificador de Trayectorias a Través de Obstáculos Poligonales", *Fourth Iberoamerican Congress of Mechanical Engineering CIDIM 99*, Santiago, Chile (November 23–26, 1999) **Vol. 2**, pp. ???–???.

APPENDIX I
COMPUTATION OF SHORTEST PATH WITHOUT TURNING RESTRICTIONS

(a) Mathematical modeling of the Environment

It is desired to obtain a geometric path of minimum length between two positions on the plane passing through polygonal obstacles, without considering curvature restrictions. For this purpose we have appealed to the acyclic algorithm of the shortest path (Taha). This algorithm has been adapted and extended for the specific goals of our problem.

In the first place we must have a geometric map of the environment in a tabulated form. This map consists of two vectors, one with the x values and the other with the y values of the vertices of all polygons, and also the initial and final position. The coordinates of the starting point will be the first elements in the vectors, and the target point gives the last elements in both vectors.

There are N polygons distributed in the environment, each one is labeled with a number, from 1 until N. Each polygon has a number of vertices, enumerated in clock-wise form, beginning with vertex 2 in polygon 1. All vertices are enumerated in this way successively, thus obtaining the following vectors:

$$X=\begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ x_K \end{bmatrix}; \quad Y=\begin{bmatrix} y_1 \\ y_2 \\ \cdot \\ \cdot \\ y_K \end{bmatrix} \tag{A1}$$

The dimension of these vectors is K. This value corresponds to the total number of vertices of the polygons, plus two. The elements x_i and y_i correspond to the ordinate and the abscissa respectively of vertex i. In Figure Al the enumeration of a typical environment characterized by $K=25$ points and with $N=3$ obstacles is shown. The obstacles must be amplified through an arbitrary offset before obtaining vectors X and Y. Such offset obeys to the necessity of generating space for the volume of the vehicle as analyzed in section 3.1 of this article.

Once the environment is characterized by vectors X and Y, a matrix of distances M is computed:

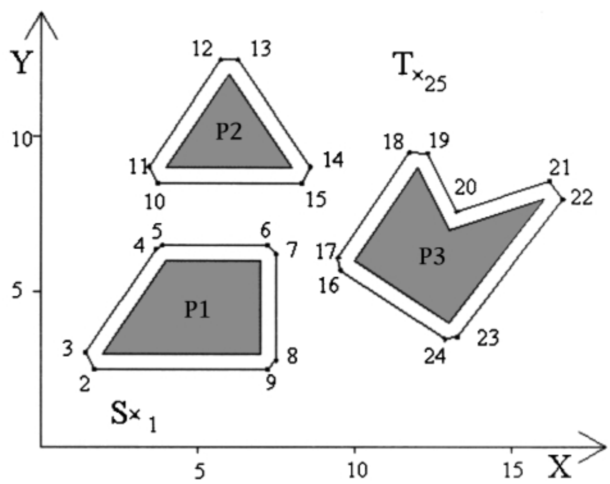

Fig. A1. Typical environment characterization.

$$M=\begin{bmatrix} d_{11} & d_{12} & \cdot & \cdot & d_{1K} \\ d_{21} & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ d_{K1} & \cdot & \cdot & \cdot & d_{KK} \end{bmatrix} \tag{A2}$$

The dimensions of this matrix are $K \times K$ and the element d_{ij} is the distance between vertex points i and j, therefore M is a symmetrical matrix with a null diagonal. When the union between two points are not possible, because there is an obstacle between them, a very high unreal distance is assigned, to avoid this itinerary in the shortest path.

(b) Algorithm of the shortest path

The steps of the algorithm will be explained using the schematized environment of the Figure A1. We have

u_i = shorter distance between the point 1 (start) and the point i

Where $u_i=0$, by definition. The u_i values, $i=2, \ldots, K$, are calculated in recursive form by means of the following formula:

$$u_i=\min\ \{\text{distance } u_j+\text{distance } d_{ij}\}$$
$$\text{for } i=2, \ldots, K;\ j=1, \ldots, i-1 \tag{A3}$$

Where d_{ij} is the distance between the vertex points i and the point j given by the matrix of distances M.

If the distance u_i is smaller than u_{i-1}, this means that the most direct path toward the initial point from the point i, is not through the previous point $i-1$, therefore in this moment a new calculation must be performed in inverse form to compute again the shortest distance to the points below j, in the following way.

$$u_h=\min\ \{\mathrm{u_h},\ \mathrm{u_l}+\mathrm{d_{hl}}\}$$
$$\text{for } h=i-1, i-2, \ldots, 2;\ 1=i, i-1, \ldots, i \tag{A4}$$

Thus, it is most probable that the minimum distance of the previous points to point i has changed. This means that the path through point i is shorter than the previous one. When arriving at a point that doesn't change the current shortest distance, the inverse loop is finished, since the following points will not change the minimum distance.

In the final solution for the shortest path, it is not enough to determine only the value of the shortest distance u_K from the target point to the starting point. In concurrent form, we should also identify the points along the path. To achieve this, a labeled procedure is used, which associates the shortest distance from the start point to each point (u_i) to the point predecessor's number.

$$\text{Label of the point } i=[u_i, j] \tag{AS}$$

Where j is the point that precedes i immediately in the shortest path toward the initial point, that is to say,

$$u_i=u_j+d_{ij} \tag{A6}$$

By definition, the label in the point 1 is [0, 0], what indicates that the point 1 is the source or departure.

In Table I the labels for the first 8 points of the studied model of the Figure A1 are shown.

https://doi.org/10.1017/S0263574701003630 Published online by Cambridge University Press

Table I. Examples of labels.

Vertex i	u_i	Label
1	$u_1 = 0$	[0, 0]
2	$u_2 = d_{21}$	[u_2, 1]
3	$u_3 = d_{31}$	[u_3, 1]
4	$u_4 = u_3 + d_{43}$	[u_4, 3]
5	$u_5 = u_4 + d_{54}$	[u_5, 4]
6	$u_6 = u_5 + d_{65}$	[u_6, 5]
7	$u_7 = u_6 + d_{76}$	[u_7, 6]
8	$u_8 = u_7 + d_{87}$	[u_8, 7]

Table II. Example of label's change.

Vertex i	U_i	Label
1	$u_1 = 0$	[0, 0]
2	$u_2 = d_{21}$	[u_2, 1]
3	$u_3 = d_{31}$	[u_3, 1]
4	$u_4 = u_3 + d_{43}$	[u_4, 3]
5	$u_5 = u_4 + d_{54}$	[u_5, 4]
6	$u_6 = u_7 + d_{67}$	[u_6, 7]
7	$u_7 = u_8 + d_{78}$	[u_7, 8]
8	$u_8 = u_9 + d_{89}$	[u_8, 9]
9	$u_9 = d_{91}$	[u_9, 1]

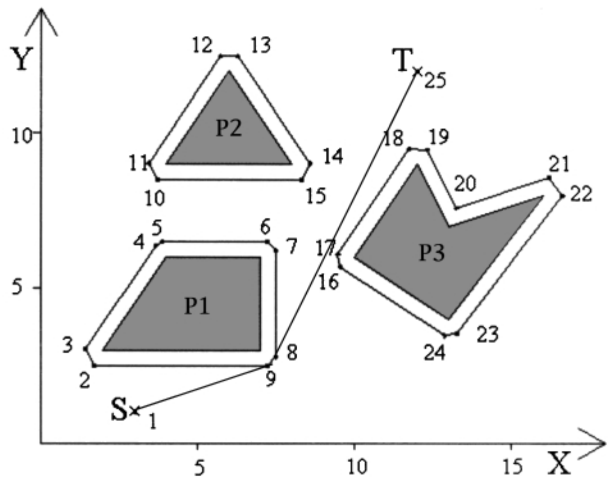

Fig. A2. Shortest path between S and T.

The labels are partial and they can change. This happens when a point presents a different shorter path toward the starting point than through the previous point and its label doesn't correspond to this previous point. For example in Figure A1, the point 9 has direct connection with the starting point, therefore its distance is $u_9 = d_{91}$ and its label number is 1, and not 8. Therefore both the path and label are computed once again to the previous points in inverse sense until arriving to a point that doesn't change its shorter path. The result of this new computation is shown in Table II. Notice the change in the shortest path for points 8, 7 and 6, which was expected. However points 5, 4, 3 and 2 remain with their already calculated path.

This recursive method implies that the definitive shorter distance from the point i to the initial point only can be obtained when arriving to the final or target point K.

In Figure A2 the minimal path obtained to connect the configurations given in Figure A1 is shown.

APPENDIX II
COMPUTATION OF THE ANGLE α_n REQUIRED FOR A REORIENTATION MANEUVER IN A LIMITED SPACE

In Figure A3 we show the reorientation maneuver to move a mobile robot from the initial orientation $\theta_i = -90°$ to $\theta_{path} = 45°$ for both $n = 1$ and $n = 2$ cases. Here n is the number of circular arc pairs (α_n) that are necessary to move the mobile from one orientation to another, β is the absolute angle between the current orientation of the mobile and the desired orientation. β is always smaller than π (rad), and α_i is the complement of β. In the previously mentioned figure the circular area for any given value of n is the free space for performing the maneuver. Note that for a small available area the value of n is forced to increase.

The angle α_n is the angle of the corresponding circular arc. In the above figure α_1 is shown. In Table III we show how this value changes for the first four cases.

Thus in general, we can deduce the following expression

$$\alpha_n = \alpha_i + \frac{2^n - 1}{2^n}\beta \tag{A7}$$

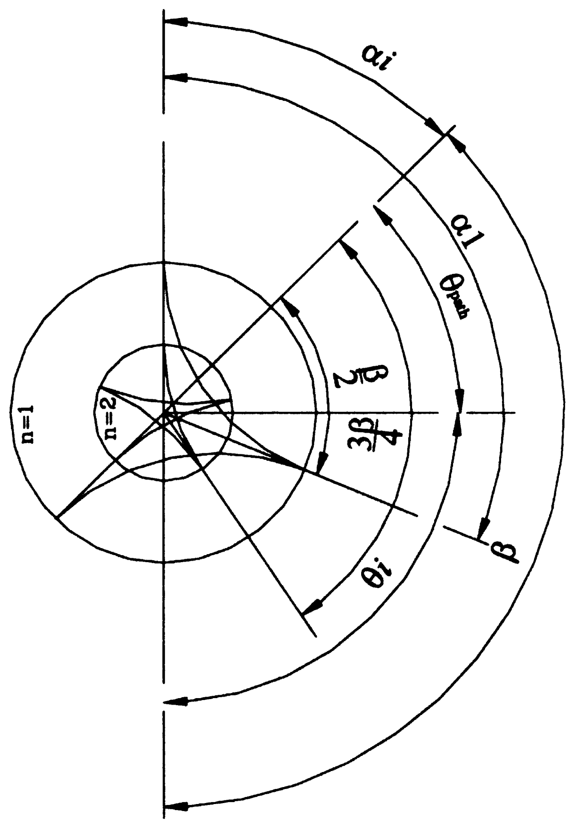

Fig. A3. Reorientation maneuver for $n = 1$ and $n = 2$.

Table III. Values of α_n.

N	α_n
0	$\alpha_0 = \alpha_i$
1	$\alpha_1 = \alpha_i + \beta/2$
2	$\alpha_2 = \alpha_i + 3\beta/4$
3	$\alpha_3 = \alpha_i + 7\beta/8$

https://doi.org/10.1017/S0263574701003630 Published online by Cambridge University Press