Robotica (1997) volume 15, pp 547-553. Printed in the United Kingdom © 1997 Cambridge University Press

Time-optimal smooth-path motion planning for a mobile robot with kinematic constraints

K. Jiang, L.D. Seneviratne, and S.W.E. Earles

Department of Mechanical Engineering, King's College London, Strand, London WC2R 2LS (UK)

(Received in Final Form: August 23, 1996)

SUMMARY

This paper presents a novel time-optimal motion planning strategy for a mobile robot with kinematic constraints. The method works in environments in presence of obstacles, without needing to generate the configuration space for the robot. Further, it derives a minimum time first derivative smooth path, as opposed to a minimum distance path which is commonly given by various present solution techniques. The problem is solved in three stages: (i) A reduced visibility graph for a point object is obtained. (ii) The reduced visibility graph is converted into a feasible reduced visibility graph accounting for the size and kinematic constraints of the robot. (iii) The A* algorithm is used to search the feasible reduced visibility graph with the cost function being the time of travel, to obtain a safe, time-optimal, smooth path. The algorithm runs in polynomial time. The method has been tested in computer simulations and test results are presented.

KEYWORDS: Time-optimal motion planning; Car-like robot; Nonholonomic constraints; Visibility graph; Feasible visibility graph.

1. INTRODUCTION

Automated motion planning strategies are essential in order to realize autonomous mobile robots, and minimum time path planning algorithms increase their efficiency. The objective of this study is to plan a collision-free path for a mobile rigid body robot through a workspace populated with obstacles.

The most popular approach to the path planning problem is the configuration space method,[1] where the robot is shrunk to a point while correspondingly growing the obstacles, in order to obtain the robot's free space. Techniques such as visibility graph,[2] Voronoi diagram[3] and cell decomposition[4] can be used to search the free space for a collision free path. The method works efficiently for free flying robots moving without changing its orientation, amongst fixed obstacles. For such a robot, working in Euclidean space $W \in R^n$ ($n = 2$ or 3), the configuration space can also be represented by an Euclidean space. However, if the robot changes its orientation in W, the configuration space becomes non-Euclidean; for $n = 2$ or 3 the configuration space becomes $\mathbf{R}^2 \times S^1$ or $\mathbf{R}^3 \times S^3/\sim$ respectively.[5] Thus the robot free space becomes computationally complex to represent and search. Further, the complexity of the problem is additionally increased for robots that are subject to kinematic constraints, such as car like robots.

The subject of car-like robot path planning has received much attention in recent years.[6–9] Such robots feature planar movements with nonholonomic constraints. An autonomous guided vehicles (AGV) is an example of such a robot. A robust algorithm for minimum time, smooth path planning would help to increase the transportation efficiency of AGV.

The smooth path planning problem was first addressed nearly thirty years ago by Dubins,[10] giving the form of the shortest bounded curvature path in the absence of obstacles. Recently, Laumond published work on the problem for the case where the workspace contains obstacles.[6] He later extended the work to non-polygonal obstacles represented by closed curves.[8] Fortune and Wilfong[11] gave a decision algorithm to determine the existence of a path under a set of given conditions. The algorithm is exact, runs in exponential time and space, but does not generate the path in question.

There has also been studies on planning time-optimal trajectories for robots. The idea of path planning with constraints on the robot's accelerations is presented in reference 12. The analysis is restricted to the case of a particle moving in one-dimension. The problem of a particle having spatial motion is addressed in reference 13 where a near-time-optimal safe trajectory for a particle moving in a plane is found; the particle is subjected to uniform L_∞ acceleration bounds on each axis. However, this model does not apply to a mobile robot system. Time-optimal trajectories for mobile robots with two independently driven wheels are presented in reference 14. Pontryagin's maximum principle is used, and the accelerations are considered as either maximum or minimum. This method does not consider collision avoidance, and the robots are not subject to kinematic constraints.

The problem of time-optimal and collision-free motion planning for robot manipulators has also been studied. The algorithm in reference 15 finds the time-optimal trajectory for robot manipulators by minimizing the time-derivative of the return (cost) function for this problem, satisfying the Hamilton-Jacobi-Bellman equation. For multiple obstacles, the trajectory is generated using a pseudo return function, which is an approximation of the return function. The obstacles considered in this paper are circular, as commonly used in trajectory planning for manipulators. This algorithm is difficult to

https://doi.org/10.1017/S0263574797000635 Published online by Cambridge University Press

extend to the trajectory planning for car-like robots, because of the dynamic model differences between a manipulator and a car-like robot.

This paper presents a novel time-optimal motion planning strategy for a car-like mobile robot with kinematic constraints, operating in an environment cluttered with obstacles. The method works with the original polygonal obstacles. The main idea behind the method is in modifying the visibility graph for a point robot into a feasible visibility graph for the given dimensioned mobile robot with kinematic constraints, and converting the feasible visibility graph into time a graph, from which the time-optimal path is chosen. The strategy finds the time-optimal path considering of only straight line and circular arc segments. The problem is solved in three stages: (i) A reduced visibility graph for a point object is obtained. (ii) The reduced visibility graph is converted into a feasible visibility graph by ruling out the paths which are not feasible for a robot with dimensions and nonholonomic constraints. Here all paths for the robot are constructed by straight lines and circular arcs of the minimum turning radius of the robot. (iii) The A* algorithm is used to search the feasible reduced visibility graph to obtain a safe, time-optimal, first time derivative smooth path. The algorithm is computationally efficient and runs in polynomial time.

2. REDUCED VISIBILITY GRAPH

The first step of the proposed algorithm is to construct the reduced visibility graph for a point robot. The concept of a reduced visibility graph arises in the context of shortest path planning for a point robot, for which the basic problem can be stated as follows:

In a workspace $W \in R^2$, there is a set of polygonal obstacles $O = \{o_i : i = 1, 2, \ldots n\}$. Given an initial point $q_{\text{init}}(x_s, y_s)$ and a goal point $q_{\text{goal}}(x_g, y_g)$ for a point robot, find the shortest collision free path.

The solution to this problem is well established.[3] It consists of constructing a visibility graph, containing the shortest path, which is found by searching using an optimization algorithm.

For polygonal obstables, the visibility graph is constructed from the set of nodes consisting of q_{init}, q_{goal} and all the obstacle vertices. A link is a straight line that connects any two nodes and a visible link connects two nodes that are visible to each other. Thus a visible link will not intersect any of the obstacles. The visibility graph consists of all the visible links, and a reduced visibility graph is constructed from the visible tangent links. It can be shown[3] that a non-tangent visible link will not be a part of the shortest path.

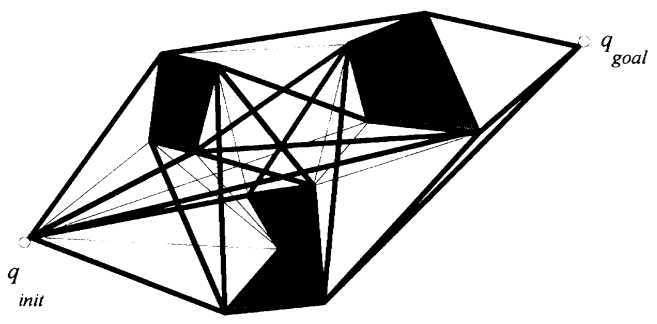

Fig. 1. A visibility graph (all lines) and a reduced visibility graph (thick lines) for a scene with polygonal obstacles.

Figure 1 shows the visibility graph and the reduced visibility graph for a particular example consisting of three polygonal obstacles.

The concept of visible tangent links can be extended to arbitrary shaped obstacles. Here the edges of the arbitrary shaped obstacles are represented by cubic splines. The visible tangent links can then be determined by a single variable optimization scheme.

Finding the shortest path for a point robot is computationally efficient. Rohnert[3] presents an algorithm for computing the tangents between two convex polygons in $O(\log n_1 + \log n_2)$ time, where n_1 and n_2 are the numbers of vertices of the two polygons, and the shortest path is computed in $O(nk + n \log n)$ time, where k is the number of convex parts of the obstacles and n is the number of obstacle vertices. Non-polygonal obstacles require iterative computations, and the computational time depends on the precision of determining the tangential points.

3. FEASIBLE VISIBILITY GRAPH

In this study the path for the dimensioned robot is found by locally modifying the reduced visibility graph for the point robot into a feasible visibility graph. The modification involves moving the point path away from the vertices and edges of the obstables, in accordance with the algorithm presented. The path modification algorithm depends on (i) the relative position between the point path and the obstacles, and (ii) the characteristics of the dimensioned robot. The latter includes features such as configuration, turning radius, manner of steering and kinematic constraints. The modified reduced visibility graph, termed the feasible visibility graph, is then searched to yield the time minimum path for the dimensioned robot. The path modification algorithm employs a steering model of the robot and minimum distance computations, and also satisfies the kinematic constraints on the robot.

3.1 Robot steering model

In this section a generalized steering model for a rectangular car like robot is presented. A four wheel rectangular robot R, which is $2a$ long and $2b$ wide, is considered. A moving frame, F_m, is attached at the centroid of R, Figure 2. The x axis of F_m is along the main axis of the robot. When the robot's front and back wheel steering angles are α and β respectively, the coordinates of the robot's instantaneous centre O_i, with respect to F_m, (x_o and y_o), are given by

$$\left.\begin{aligned} x_o &= -\frac{l}{2}\left(\frac{\tan\alpha + \tan\beta}{\tan\alpha - \tan\beta}\right) \\ y_o &= \left(\frac{l}{\tan\alpha - \tan\beta}\right) \end{aligned}\right\} \tan\alpha \neq \tan\beta \qquad (1)$$

where $l/2$ is the distance between F_m and each the two

https://doi.org/10.1017/S0263574797000635 Published online by Cambridge University Press

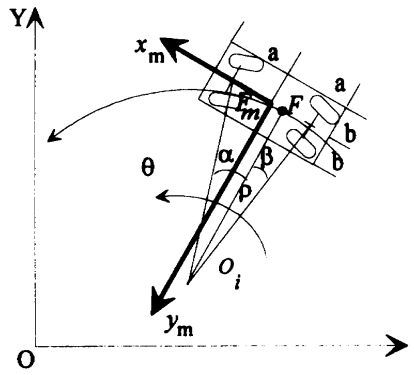

Fig. 2. The model of a moving robot.

wheel axles. α and β are measured positive counter-clockwise from the robot's main axis, and in practice will usually have opposite signs.

Let F be a reference point on the robot's main axis such that the velocity of F is along the main axis of the robot. In the following sections, the motion of point F will be used to specify the velocity of the robot. For a robot steered by front wheels only, the reference point F will be at the mid-point of the rear wheel axle. The coordinates of F in frame F_m are $(-x_0, 0)$.

The turning radius, ρ, of the robot is given by y_o, and it varies with α and β, (1). If there were no constraints on α and β, the robot could achieve any turning radius employing an appropriate choice of α and β. However, in practice, mechanical stops in the steering gear limit the range of α and β. Let α and β be contained within the range $\pm\gamma$. Then the minimum turn radius, ρ, is given by $\rho_{\min} = l/2 \tan \gamma$, when $\alpha = -\beta = \gamma$.

Equation (1) represents a robot steered at both wheels. It contains front wheel only steered robots ($\beta = 0$) and back wheel only steered robots ($\alpha = 0$) as special cases. For both these cases, $\rho_{\min} = l/\tan \gamma$.

3.2. Nonholonomic kinematic constraints

Consider a rectangular shaped mobile robot moving on flat ground, Figure 2. When the wheels are in pure rolling contact with the ground, the reference point F describes a curve that is tangential to the main axis of the robot. Hence the robot's motion is constrained by:

$$-\dot{x} \sin \theta + \dot{y} \cos \theta = 0 \qquad (2)$$

where (x, y) are the coordinates of F relative to a globally fixed frame X-Y and θ is the angle made by the robot's main axis relative to the global X axis, Figure 2.

Equation (2) is a kinematic constraint which must be satisfied by the robot's motion. Further equation (2) is non-integrable and hence is a non-holonomic equality constraint.

As mentioned previously, car-like robots in general have limited steering angles which imposes a lower bound, $\rho_{\min}$, on the turning radius. Since $\mathbf{v}$, the global velocity of point F, is given by $\mathbf{v} = \rho \times \dot{\theta}$, where $\dot{\theta}$ is the instantaneous angular velocity of the robot, it follows that

$$\mathbf{v} \geq \rho_{\min} \times \dot{\theta}, \text{ i.e. } \dot{x}^2 + \dot{y}^2 - \rho^2 \min \dot{\theta}^2 \geq 0 \qquad (3)$$

Condition (3) must also be satisfied by all configurations of the robot. It is a non-holonomic inequality constraint.

3.3. Minimum distance computations

The algorithm presented finds a path for a point robot and then locally modifies this path to account for kinematic constraints and obstacle avoidance. It is assumed that the geometry and the configuration of the obstacles are known. Minimum distance computations between the obstacles, and between the point path and the obstacles are central to the path modification algorithm. Such computations provide a measure of the robot's free space relative to the selected point path.

There are a number of possible algorithms for computing the minimum distance between two objects.[16,17] These algorithms are based on either linear or nonlinear programming, and involve recurrent searches which can be time consuming. The computational time depends on the total number of vertices between the two obstacles. Further, the minimum distance between each pair of obstacles needs to be computed.

The minimum distance between two polygonal obstacles will be either between two vertices or between an edge and a vertex, and hence can be obtained by minimum distance computations between two points and between a straight line and a point.

The algorithm for minimum distance computations employed does not involve a recurring search. In addition, the number of minimum distance computations is minimized, by restricting the regions where such information is computed.[16] This is achieved by dividing the obstacle set into various subsets relative to the point path in question. The point path itself divides the obstacles into a left set O^l and right set O^r, Figure 3. Further, the linear segments l_i of the point path can be used to divide the two obstacle sets into radial sets O_i^l and O_i^r ($i = 1, 2, \ldots m$; where m is the total number of linear path segments), Figure 3; overlapping obstables belonging to two radial sets. The path modification

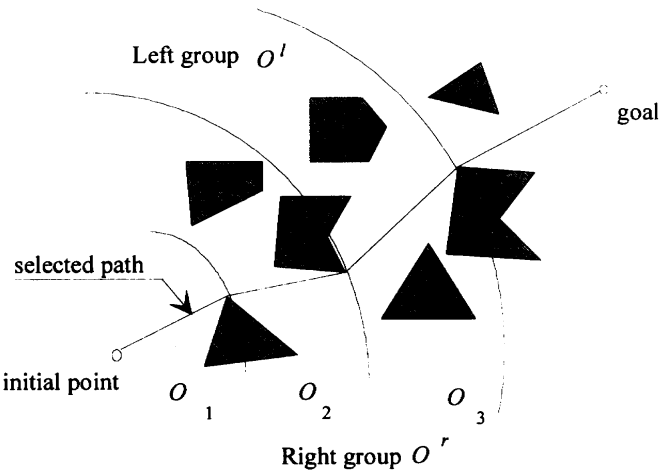

Fig. 3. Obstalce groups divided by the path and the turn points of the path.

https://doi.org/10.1017/S0263574797000635 Published online by Cambridge University Press

algorithm requires only the following minimum distance computations:
(i) between l_i and $O_i^l \cup O_i^r$ and (ii) between O_i^l and O_i^r. Let the minimum distance between two obstacles be $D_{\min}$ and the minimum distance between l_i and an obstacle be $\bar{D}_{\min}$. The minimum distances are computed by using,

$$D_{\min}(v_i, v_j) = \min \|v_i v_j\| \quad \text{and} \quad \bar{D}_{\min}(l_i, v_j) = \min \|l_i v_j\|,$$

where v_i and v_j are vertices of the relevant obstacles. Figure 4 indicates the minimum distances computed for a particular example.

3.4 Feasible visibility graph

A feasible visibility graph (FVG) is defined as a reduced visibility graph which allows the dimensioned robot to travel while acoiding collisions. Since the reduced visibility graph for a point robot does not provide a collision free path for a dimensioned robot, it has to be converted into a FVG, which can be searched for the time minimum path.

In order to detect possible collision regions, the dimensions of the swept volume of the traveling robot need to be known. Let a rectangle shaped robot R, $2a$ long and $2b$ wide, move in a workspace $W \in R^2$. The robot R will sweep a certain width along its path. When it moves along a straight line, its swept width is,

$$w_s = 2b \tag{4}$$

When it turns with steering angles α and β, at the front and back wheels respectively, its swept width $w_s(\alpha, \beta)$ has the form, Figure 5:

$$w_s(\alpha, \beta) = \rho_{\text{out}}(\alpha, \beta) - \rho_{\text{in}}(\alpha, \beta), \tag{5}$$

where

$$\rho_{\text{out}}(\alpha, \beta) = \sqrt{\left(a + l\frac{\tan\alpha + \tan\beta}{\tan\alpha - \tan\beta}\right)^2 + \left(b + \frac{2l}{\tan\alpha - \tan\beta}\right)^2}$$

$$\rho_{\text{in}}(\alpha, \beta) = \frac{2l}{\tan\alpha - \tan\beta} - b$$

In the visibility graph the robot travels along straight lines and changes direction only at obstable vertices. Thus the visibility graph needs to be modified at obstacle vertices. In addition, it also needs to be modified in regions where $\bar{D}_{\min}$, the minimum distance between a straight line segment of the visibility graph and the obstacles, is less than $2b$; $\bar{D}_{\min} < 2b$.

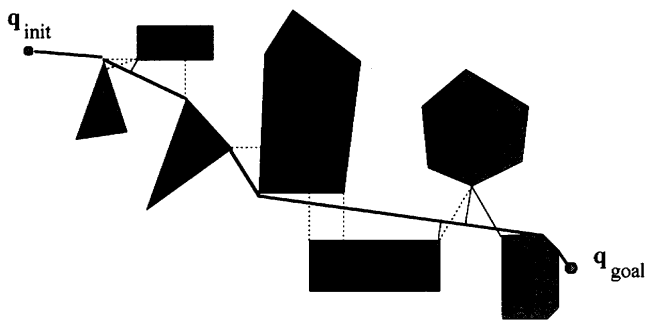

Fig. 4. Minimum distances between obstacles in O_i^l and O_i^r.

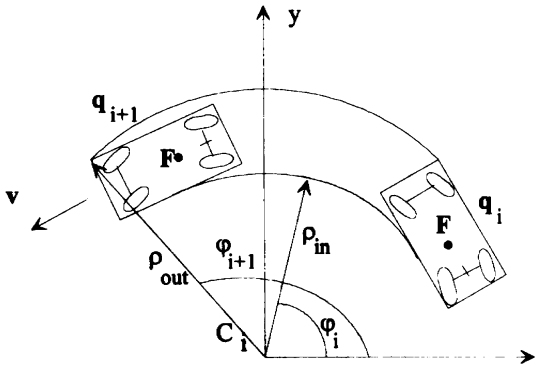

Fig. 5. Swept area of a robot R.

Once the positions where modifications are necessary have been identified, a check is made to see if modifications are possible. Using the minimum distance between two obstacles, a collision free condition for the robot to pass between the two obstacles is,

$$D_{\min} - w_s(\alpha, \beta) > 0.$$

Thus for straight line segments,

$$D_{\min} > 2b \qquad (\alpha = \beta = 0) \tag{6}$$

and at obstacle vertices

$$D_{\min} + \frac{2l}{\tan\alpha - \tan\beta} - b - \sqrt{\left(a + l\frac{\tan\alpha + \tan\beta}{\tan\alpha - \tan\beta}\right)^2 + \left(b + \frac{2l}{\tan\alpha - \tan\beta}\right)^2} > 0 \quad (\text{for } \alpha \neq \beta) \tag{7}$$

Conditions (6) and (7) are algebraic expressions which are environmental holonomic constraints to be satisfied by the robot's motion.

The application of (6) is straight forward. (7) contains two variables, α and β. Assuming that in practice, either $\beta = 0$ (front wheel steering) or $\alpha = 0$ (back wheel steering) or $\alpha = -\beta = 0$ (both front and back wheel steering), then (7) can be used to solve for α (or β) and hence it can be determined whether passage for the robot is possible.

If (6) or (7) is not satisfied in any region of the reduced visibility graph, then all path segments in that region are eliminated. In regions where (6) and (7) are satisfied, the reduced visibility graph is modified to yield the feasible visibility graph, FVG.

The approach adopted is to modify the reduced visibility graph using circular arc segments. That is, it is assumed that when the robot changes direction it does so with a constant steering angle, as determined by (7).

Once the steering angle is determined, the turning radius, ρ, is found from (1). The radii of the inner and outer circles of the robot's swept volume, ρ_{in} and ρ_{out}, can then be found as outlined in this section. The centre of the circular arc segment, C_i, is located a distance ρ_{in} away form the vertex, along the bisector of the angle between the two adjacent edges at the vertex. The exact position of the centre depends on the free space used by

https://doi.org/10.1017/S0263574797000635 Published online by Cambridge University Press

the swept volume. The circular arc segment is drawn, with radius ρ and centre C_i, Figure 5. The circular arc path thus constructed will lead the robot to sweep the area between ρ_{in} and ρ_{out}, and if the area between ρ_{in} and ρ_{out} is free of obstacles, the robot motion is collision-free in this area.

The FVG is completed by connecting consecutive circular segments with tangential lines. The straight line segments of the FVG are checked for collisions, and are modified or rejected, as for the circular paths outlined above, until a collision free FVG is obtained.

4. ROBOT TRAVEL TIME

The feasible visibility graph consists of a series of connected path segments which are either straight lines or circular arcs. The lengths of all the path segments of the FVG are known.

In this investigation it is assumed that when the robot travels along any given path segment, it is either moving at its maximum speed, if this is attainable, or accelerating/decelerating at its maximum value. Thus the speed-time curves for any path segment are as shown in Figure 6a or 6b. The modulus of the non-zero slopes of Figure 6 will be equal to a_{max}, the maximum acceleration/deceleration limit of the robot. v_1 and v_3 are the speeds at the beginning and end of the segment respectively. v_2 is the maximum possible speed of the robot for the segment concerned, subject to the kinematic constraint (3).

Let the maximum speed of the robot along a straight line segment be v_{ms}, and the maximum speed along a circular segment be $v_{mc}(\rho)$, which is in general a function of ρ, the radius of the circular segment, and in general $v_{mc}(\rho) \leq v_{ms}$. It is assumed that $v_{mc}(\rho)$ is known from the specification of the robot. Hence $v_2 = v_{ms}$ for a straight line segment and $v_2 = v_{mc}(\rho)$ for a circular segment. Further, for circular arc segments it is supposed that $v_1 = v_3 = v_{mc}(\rho)$, i.e. the robot travels the entire circular arc at its maximum speed. For straight line segments, v_1 and v_3 will be equal to the maximum speeds of the adjacent circular segments. The exception to this is at the initial and goal points where v_1 and v_3 will be zero respectively. If the segment is at the start or end of the path, then v_1 or v_3 will respectively be zero. Thus v_1, v_2 and v_3 for any path segment are known.

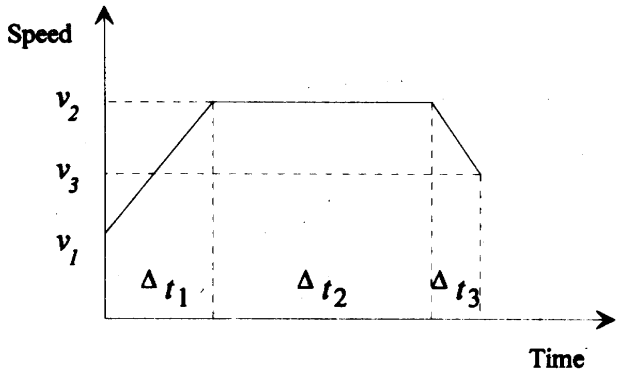

Fig. 6a. A case where the maximum speed v_2 is reached.

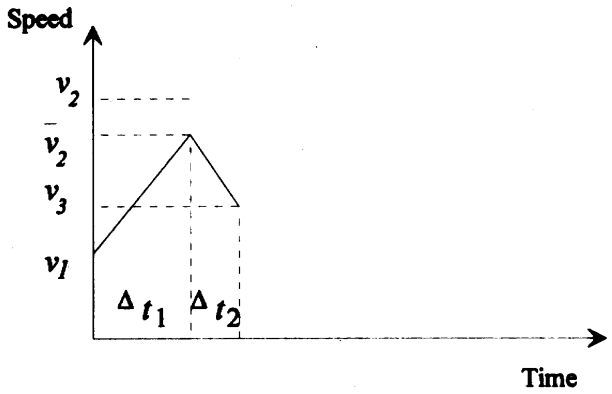

Fig. 6b. A case where the maximum speed v_2 is not reached.

It can be shown that if

$$L_i > \frac{1}{2a_{max}}(2v_2^2 - v_1^2 - v_3^2),$$

then Figure 6a applies and the total time of travel for the segment

$$T_i = (2v_2 - v_1 - v_3)/a_{max} + [L_i - (2v_2^2 - v_1^2 - v_3^2)/2a_{max}]/v_2 \quad (9)$$

and, if

$$L_i \leq \frac{1}{2a_{max}}(2v_2^2 - v_1^2 - v_3^2),$$

then Figure 6b applies and

$$T_i = [\sqrt{2(2L_i a_{max} + v_1^2 + v_3^2)} - v_1 - v_3]/a_{max} \quad (10)$$

Using (9) and (10), a travel time for each path segment is found. The travel time depends on the type of segment (straight line, circular or other), the assumed speed characteristics, the length of the segment and its adjacent connectivity. Features such as vehicle dynamics and the nature of the terrain covered are not considered in this study. However these aspects may be included in this approach.

5. SEARCHING FOR THE MINIMUM TIME PATH

The FVG consists of feasible, collision free, first derivative smooth paths for the dimensioned car like robot. The links of the FVG are either straight lines or circular arcs. Since any connected path in the FVG is first derivative continuous, thus satisfying the non-holonomic constraint (4).

Past investigations have been mainly concerned with obtaining the minimum distance paths, where the cost function is the total distance traveled. In this study the cost function is taken to be the total time of travel, from the start to the goal, along a path of the FVG. By introducing (9) and (10), the FVG is converted into a time graph, which is searched for the time minimum path. The time spent in traveling path P_j is,

$$T_j = \sum_{k=1}^{n} T_{jk} \quad (11)$$

https://doi.org/10.1017/S0263574797000635 Published online by Cambridge University Press

where T_{jk} is the time of travel for segment P_{jk}, and n is the total number of segments of path P_j.

The A* algorithm is employed to find the time minimum path with (11) as the cost function and h, the heuristic function still being the distance to the goal.

6. COMPUTER SIMULATIONS

The motion planning strategy presented is implemented and tested for a variety of operating conditions, using computer simulations. A wide range of obstacle shapes, numbers, locations and robot models have been employed. The algorithm generated collision-free motion for all cases tested.

The results from a particular example consisting of 8 polygonal obstacles is given in Figure 8. The objective is to move the robot from the start location (30, 30, 90°) to the goal location (850, 550, 90°). The robot model simulated has the following characteristics:
length: $2a = 1.5$;
width: $2b = 0.8$;
the distance between the two axes: $l = 1.3$;
the distance between the two wheels: $d = 0.8$;
the steering angles for front and back wheels: $-45^0 < \alpha < 45^0$; $\beta = 0^0$;
the maximum velocity and acceleration along a straight line path: $v_{ms} = 12$/sec and $a_{mc} = 50/\text{sec}^2$;
the maximum velocity and acceleration along a circular path: $v_{mc}(\rho) = v_{mc}(\rho_{\max}) = 8$/sec;
$a_{mc}(\rho_{\max}) = 50$/sec;

The FVG generated by the algorithm is shown in Figure 7a. The FVG is searched using the A* algorithm, and the minimum time, first derivative smooth, collision-free motion is shown in Figure 7b. The total travel time is 74.6 seconds.

The algorithm is tested on a number of cluttered environments, giving collision-free motions, and demonstrating its robust nature.

7. DISCUSSION AND CONCLUSIONS

An algorithm for minimum time first derivative smooth path motion planning for a rectangular car like robot subject to kinematic constraints, is presented. The algorithm operates on the original obstacles without needing to generate the robot's configuration space. The algorithm has three stages: (i) The reduced visibility graph for a point robot is generated. (ii) The reduced visibility graph is modified to a feasible visibility graph, FVG, for the care like robot. (iii) The FVG is searched for the minimum time path.

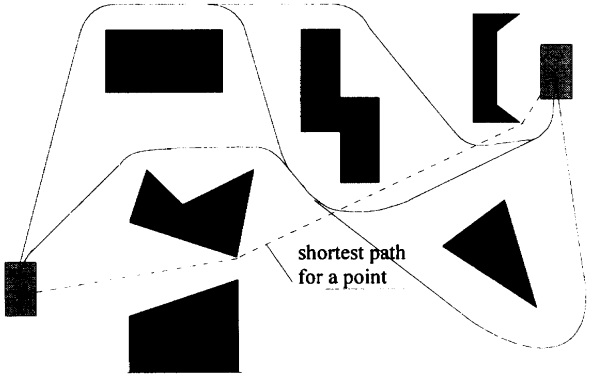

Fig. 7a. Feasible visibility graph of a simulation for a car-like robot with limited 45° steering angle.

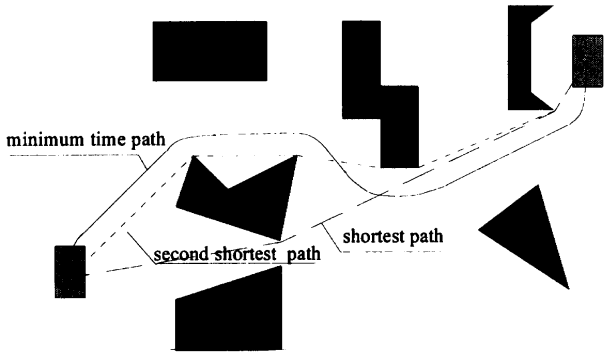

Fig. 7b. Final motion path of a simulation for a car-like robottraveling from q_{init} to $_{q\text{goal}}$ in minimum time.

The FVG consists of straight line segments and tangential circular arcs; thus any connected path in the FVG will satisfy the nonholonomic constraints acting on the robot. Also all path in the FVG are first derivative continuous.

The circular arc assumption also implies that the robot employs a constant steering angle when changing direction. For a robot whose steering angle is controlled by a limit switch, this assumption is always valid, and the algorithm will generate the minimum time path among all smooth paths formed by straight lines and circular arcs. The significance of this assumption for robots that can vary their steering angles continuously, needs to be investigated. The algorithm can be made more general by using spline functions instead of circular arcs.

The FVG is generated by employing minimum distance computations and the robot's steering model. The FVG is then converted to an equivalent time graph based on the robot's speed model; it is assumed that the robot travels at its maximum possible speed when it is not accelerating/decelerating at its maximum value. The speed model does not include features such as robot dynamics and the nature of terrain traveled.

The algorithm has been tested by computer simulations, and is shown to be computationally efficient with wide applicability. The algorithm can be used directly in AGV motion planning.

References

1. T. Lozano-Pérez, "Spatial planning: A Configuration Space Approach" *IEEE Transaction on Computers* **IC-32**(2), 108–120 (1983).
2. H. Rohnert, "Shortest paths in the plane with convex-polygonal obstables" *Information Processing Letters* **23,** 71–76 (1986).
3. O. Takahashi and R.J. Schilling, "Motion planning in a plane using generalized Voronoi Diagram" *IEEE Transactions on Robotics and Automation* **5,** No. 2, 143–150 (1989).

https://doi.org/10.1017/S0263574797000635 Published online by Cambridge University Press

4. R.A. Brooks, T. Lozano-Pérez, ''A subdivision algorithm in configuration space for findpath with rotation'' *IEEE Transactions on Systems, man and Cybernetics* **15,** No. 2, 224–233 (1985).
5. J-C. Latombe, *Robot Motion Planning* (Kluwer Academic Publisher, 1991).
6. J.P. Laumond. ''Feasible trajectories for mobile robots with kinematic and environment constraints,'' *Intelligent Autonomous Systems* 346–354 (Dec., 1986).
7. J.P. Laumond, P. Jacobs, M. Taïx and R.M. Murray, ''A motion planner for nonholonomic mobile robots'', *IEEE Transactions on Robotics and Automation* **10,** No. 5, 577–593 (1994).
8. J.P. Laumond. ''Finding collision free smooth trajectories for a non holonomic mobile robot'' *Proc. Int. Joint Conf. on Artificial Intelligence,* Milan, Italy (Aug., 1987) pp. 1120–1123.
9. Th. Fraichard, ''Smooth trajectory planning for a car in a structed world'' *Proc. 1991 IEEE Int. Conf. on Robotics and Automation,* Sacramento, California (April, 1991) pp. 318–323.
10. L.E. Dubins. ''On curves of minimum length with a constraint on average curvature and with prescribed initial and terminal position and tangents,'' *Amer. J. Mathematics* **79,** 497–516 (1957).
11. S.J. Fortune and G.T. Wilfong, ''Planning constrained motions'' *ACM Symp. on Theory of Computation* (1988) (1988) Ch. 53, pp. 445–459.
12. Colm Ó'Dúnlaing, ''Motion planning with inertial constraints'' *Robotics Report* 73 (New York University, July 1986).
13. J. Canny, B. Donald, J. Reif and P. Xavier, ''On the complexity of kinodynamic planning'' *29th Symposium on the Foundations of Computer Science,* White Plains, NY October 1988) pp. 306–316.
14. D.B. Reister and F.G. Pin, ''Time-optimal trajectories for mobile robots with 2 independently driven wheels'' *Int. J. Robotics Research* **13,** No 1, 38–54 (1994).
15. S. Sundar and Z. Shiller, ''Time-optimal obstacle avoidance'', *Proceedings of IEEE International Conference on Robotics and Automation,* Los Angles, CA (1995) Ch. 492, pp. 3075–3080.
16. L.D. Seneviratne, K. Jiang and S.W.E. Earles, ''A fast collision avoidance algorithm for a rectangular object'' *Eight World Congress on the Theory of Machines and Mechanisms,* Prague (Aug., 1991) pp. 127–130.
17. J.T. Schwartz, ''Finding the minimum distance between two convex polygons'' *Inform. Process Lett.* **13,** 168–170 (1981).