韓國軍事運營分析學會誌
第18卷, 第2號, 1992. 12. 31.

SHORTEST PATH FOR ROBOT CAR

Chesoong Kim, Sangbok Ree*

Abstract

In this paper, we consider the shortest path problem of a Robot car moving in a workspace which consists of some obstacles. The motion of the Robot car is considered to have initial and final directions with some restrictions in the curvature of the path. At first we consider the problem in the case of having no obstacles and we give an analytical solution. Then we present an algorithm to find a feasible path in the case of having obstacles and a method to improve this feasible path into a minimal path. Some computational results using Graph theory and Linear programming have been included.

1. Introduction

The Robot Car in our problem (Fig. 1) consists of 4 wheels as normal automobile, 2 of them are fixed at the rear and the other one called the *steering wheel* in the front which steers and drives the vehicle. We have a restriction on the steering wheel, i.e., it can turn left or right upto a certain angle say θ (Fig. 1) which will cause a restriction on the curvature of the Robot car path. Also the vehicle does not move backwards. In the workspace there are several things which we call as obstacles (including the work stations) and we assume that they have polygonal shapes. Figure

* Seoul National University

2 provides an intuitive feeling about the problem i.e. finding a path from placement A to placement B with initial and final directions $\vec{U}.\vec{V}$, of robot car at A and B respectively. This curve should be *smooth* enough in order to be followed by our robot and wuth some restrictions on its curvature.

2. Shortest Path without Obstacles

Given are the two points A and B and the two unit vectors being $\vec{U}$ and $\vec{V}$ at A and B respectively. We are interested in a path, i.e. a curve in the plane, of minimal length from A to B with tangent vectors $\vec{U}$ and $\vec{V}$ and A and Brepectively. Thus, if we denote Γ as :

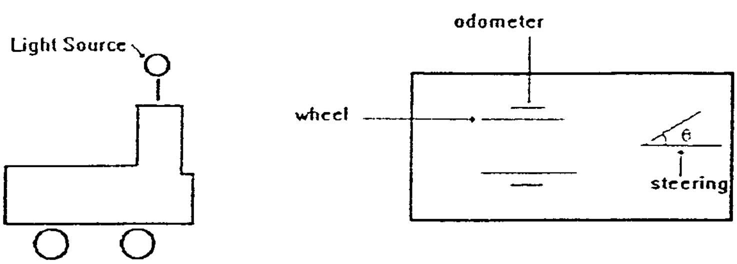

Figure 1 : Model of the Robot car

$\Gamma = \{X(t) \in R^2, a \le t \le b, |X| = 1, X(a) = A, X(b) = B, \dot{X}(a) = \vec{U}, \dot{X}(b) = \vec{V}\}$

We wish to find a curve in Γ with minimal length.

Non-existence Obstacles

It is easy to see that $\exists$ A, B $\in R^2$ and $\vec{U}, \vec{V} \in R^2$ for which no path of minimal length in Γ exists. So, we need to pose some further restriction on our curve. From the structure of the Robot car we know that the front wheel can turn in a certain range. i.e. the angle θ in the Figure 1 is restricted to range in an interval say $[-\theta_0,$

θ_0]. This will clearly force the curvature of our curve to be bounded above by maximal curvature which we denote it by K_{max} , i.e.,

$$|\ddot{X}(t)| \le K_{max} = \frac{1}{r_{min}} \qquad (1)$$

where

$$r_{min} := \frac{1}{K_{max}}$$

is the radius of curvature corresponding to K_{max} . But it can be shown that adding the restriction (1) to the set Γ, Will not remove non-existence difficulty, i.e. again there exists $A, B, \vec{U}, \vec{V} \in R^2$ such that no path of minimal length exists. The idea of *average curvature* provides a way to overcome the non-existence situation.

We first mention a classical lemma. The proof can found in [1].

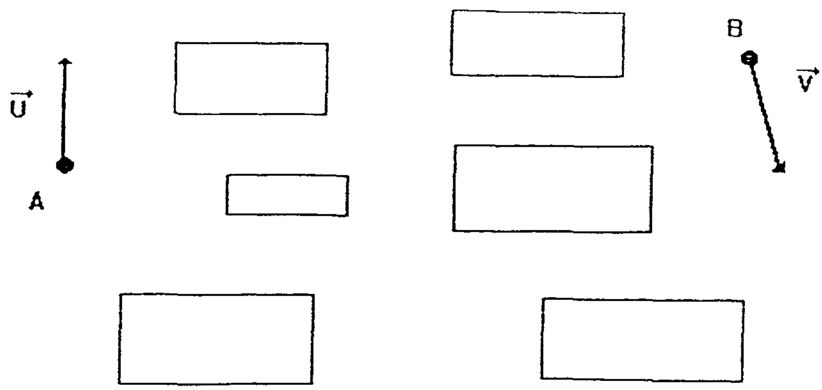

Figure 2 : Workspace with Obstacles

Lemma : Let

$$\Gamma = \{X(t) : a \le t \le b, X(t) \in R^2, \ |\dot{X}(t)| = 1\}$$

If $\ddot{X}(t)$ exists $\forall t \in [a, b]$ then

$$\|\ddot{X}(t)\| \le \frac{1}{r} \iff \|\dot{X}(t_1) - \dot{X}(t_2)\| \le r^{-1}|t_1 - t_2| \ \forall t_1, t_2 \in [a, b] \ (L)$$

So, define

$$\overline{K}(x) := \frac{\| X(S_1) - X(S_2) \|}{S_1 - S_2}$$

to be the *average curuature* of the curve $X(t)$ in the interval $[S_1, S_2]$ and

$$\overline{K}(X) \le r^{-1} \text{ on } [a,b] \iff \dot{X}(t) \textit{ exists on } [a,b] \cap \| \dot{X}(t) \| = 1 \forall t \cap$$
$$X(t) \textit{ satisfies the (L) Condition } \forall t_1, t_2 \in [a,b].$$

We denote

$$\begin{aligned} c(n, A, u, B, v, r) = \{ X \mid & X : [0, L] \to R^n, \\ & \| \dot{X}(t) \| = 1 \quad \forall t \in [0, L], \\ & \overline{K}(X) \le r^{-1} \textit{ on } [0, L] \textit{ and} \\ & X(0) = A, X(L) = B \\ & \dot{X}(0) = u, \dot{X}(L) = v \} \end{aligned}$$

Now it can be proved that in $c(n, A, u, B, v, r)$ paths of minimal length necessarily exist. Such a path is called r-geodesic. L. E. Dubins[1] has investigated analytically the types of r-gedesics for $n=2$. His main result can be summarized in the following theorem.

Theorem : Every planar r-geodesic is necessarily a continously differentiable curve which is either

1. an arc of a circle of radius r, followed by a line segment, followed by an arc of a circle of radius r; (CLC)
2. a sequence of 3 arcs of circles of radii r; (CCC)
3. a subpath of a path of type (1) or (2).

3. Shortest Path with Obstacles

This section is split into 2 parts namely 3. 1) Finding the collision-free paths of the vehicle with obstacles from one placement tp another placement and 3. 2) Improving the collision-free path into a minimal one between 2 placements. In 3. 1) we provide an algorithm for generating collision-free path by constructing a directed graph. In 3. 2) We take this path and use LP tp make it a path of minimal length.

3. 1 Collision-free path of the Robot car

We restrict motion of the vehicle by constructing lanes (line segments) in the workspace. This restriction can be justified 1) for safety reasons in practice and 2) that our analytical solution in the case of obstacle free environment consists of paths along line segments and circular arcs. Due to these lanes it is obvious that whenever the vehicle is moving on some lane then it is not colliding with any obstacle.

Following are some definitions

1. $Sweep_t(r)$: This denotes the region of the plane swept out by the vehicle as it takes a circular arc denoted by $Arc_t(r)$ (Figure 3). t is the turn from lane c_1 to lane c_2 and z_1 and z_2 are start and end positions.
2. Minimum free radius : It is the minimal radius of a turn of circular arc such that the vehicle is not colliding with any obstacle and also as close to the intersection of theh lanes as possible so as to *overshoot* other intersections.
3. Critical radius : A radius r is said to be a critical radius for a turn t if $r_{min} \le r \le r_{max}(t)$ and at least one of the following holds :
   (a) $r = r_{min}$
   (b) $r = r_{max}(t)$
   (c) $\alpha_t(r)$ is tangent to an obstacle wall i.e. $\exists\ O : \alpha_t(r)$ is tangent to $W(O)$
   (d) $\exists\ O : \beta_t(r)$ is tangent to $W(O)$
   (e) $cor(O) \in \alpha_t(r)$
   (f) $cor(O) \in \beta_t(r)$
   (g) $cor(O) \in \gamma_t(r)$
   (h) $cor(O) \in B_t(r) \quad \forall\ O$
   (i) $B_t(r)$ is collinear with obstacle wall (W)
   (j) $cor(O) \in E_t(r) \quad \forall\ O$
   (k) $E_t(r)$ is collinear with obstacle wall (W)

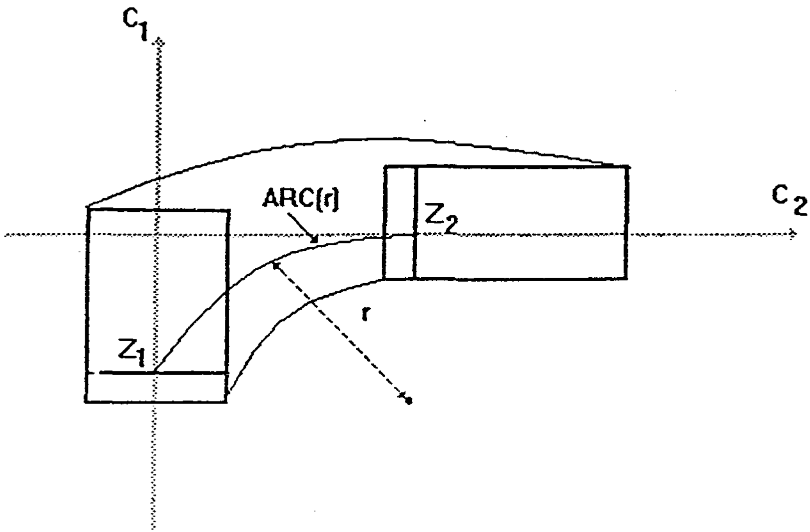

Figure 3 : Sweep and arc

Our goal now is to find the minimum free radius for each turn. In order to do this we compute a set of "critical radii" for a turn t such that every radius r in the open interval between two successive critical radii for t the statement "$Sweep_t(r)$ intersects the interior of no obstacle".

Testing and finding free radii In order to compute the minimum free radius for a turn t it is sufficient to find the minimum free critical radius for t. To decide if a radius r is free for t, first test to see if some obstacle is completely contained in $Sweep_t(r)$ by choosing an interior point from each obstacle and checking if any of these points are in the interior of $Sweep_t(r)$. If at least one of these points is in $Sweep_t(r)$ then r is not in not free for t. Otherwise no obstacle is completely contained in $Sweep_t(r)$ and so test if some obstacle partially overlaps $Sweep_t(r)$.

Building the graph G=(V, E)

$V = \{t \mid \exists r_t : r \text{ is a free radius}\}$

$E = \{e_{ij} : e\text{'s are lanes}\}$

Main Algorithm

1. Compute

$$T_0 = \{t = (c_0, c, d_0, d)\}$$

where c_0, d_0 are the initial lane and initial direction.

2. Compute

$$T_f = \{t = (c, c_f, d, d_f)\}$$

wher c_f, d_f are the final lane and final direction.

3. Search in the graph G for a directed path from any vertex in T_0 to any vertex in T_f. Some aspects of this idea have bveen already considered by G. T. Wilfong [2, 3].

3.2 Deformation of a Path into a Shortest Path

Until now there are no practical solution and more detail example to solving this problem, specially, Wilfong[2, 3] had studied theoretical solution existence and complexity. Here we try to find practical solution method by using given Wilfong idea and LP form.

Our idea to find the shortest path is as follows :

The algorithm in the above section will give us a collision-free path, say $\hat{p}$. This path $\hat{p}$ is then deformed into a shortest path. Again we begin with some definitions as below.

Definitions :

Feasible path : A path p is said to be feasible if all the minimum free radii $r_i(p) \geq r_{min}$ $i = 1, 2, \cdots, n$.

Lengh $\lambda_i(p)$: $\lambda_i(p)$ are the lengths of the portion on the lanes between 2 intersection points of the lanes.

Equivalent paths : If p and q are 2 feasible paths with same initial and final placements respectively and $T_p = T_q$ (T_p and T_q ard sequence of turns $t_1, t_2, \cdots, t_{k-1}$) then p and q are said tp be *equivalent.*

We deform the path $\hat{p}$ by performing the following.

1. Make q equivalent $\hat{p}$. i.e. $T_p = T_q$.
2. p should be feasible.
3. Make the minimum free radii for p, $r_i(p) \in I_{ij} = [a_{ij}, b_{ij}]$, I_{ij} is the family of free intervals corresponding to turn t_i.

Then we will evaluate the length of p as follows :

$$length(p) = l(p) = \sum_{i=1}^{k} \lambda_i - (2 - \frac{\pi}{2}) \sum_{i=1}^{k-1} r_i(p)$$

As we have restructions on our radii, the problem now becomes a minimization problem formulated below.

LP formulation

We formulated the LP problems and used GAMS for solving them. The program listings and the results are attached.

$$\text{Min } l(p) = \sum_{i=1}^{k} \lambda_i - (2 - \frac{\pi}{2}) \sum_{i=1}^{k-1} r_i(p)$$

$$\text{s.t.} \qquad r_i + r_{i+1} \le \lambda_{i+1}$$

$$r_i \in \bigcup_j I_{ij}$$

4. EXAMPLE

We give in this section an example (Figure 4 : Workspace including initial and final position) which consists of several parts containing the main ideas discussed above.

Computing critical intervals for each turn :

$$t = (S_1, S_2, d_1, d_2)$$

Notaions and Conventions :

1. M_{rs} := intersection point of the lanes r and s.
2. $\pm i$, $\pm j$ for directions.
3. It can be shown that if

$t = (S_1, S_2, d_1, d_2)$ and $t' = (S_2, S_1, -d_2, -d_1)$ then

$\forall^r\ Arc_t(r)$ is collision free $\Longleftrightarrow$ $Arc_t(r)$ is collision free, thus, critical intervals at t and t' are equivalent.

At each intersection points M_{rs} $\exists$ 8 turns. For simplicity we write critical intervals only for 4 turns.

t_j $\rightarrow$ turn in clockwise direction

t_j' $\rightarrow$ turn in anticlockwise direction

$t_j = (S_1, S_2, d_1, d_2) \Rightarrow t_j' = (S_2, S_1, -d_2, -d_1)$

Only the turns with free radii have been numbered below and drawn in the figure.

At M_{1a} : $t_1 = (1, a, j, i)$, $t_1' = (a, 1, -i, -j)$ $I_1 = (1, 3)$

At M_{1b} : $t_2 = (1, b, i, -j)$, $t_3 = (b, 1, j, i)$ $I_2 = (1, 6)$, $I_3 (1, 9)$

At M_{1c} : $t_4 = (1, c, i, -j)$, $t_5 = (c, 1, j, i)$ $I_4 = (5, 10)$, $I_5 = (1, 9)$

At M_{1d} : $t_6 = (1, d, i, -j)$ $I_6 = (3, 5) \cup (7, 9)$

At M_{2a} : $t = (a, 2, j, i)$, $t_7 = (2, a, -i, j)$ No free radius, $I_7 = (1, 9)$

At M_{2b} : $t_8 = (b, 2, -j, -i)$, $t_9 = (2, b, i, -j)$, $t_{10} = (2, b, -i, j)$, $t_{11} = (b, 2, j, i)$

$I_8 = (1, 10)$, $I_9 = (4, 8)$, $I_{10} = (1, 10)$, $I_{11} = (5, 8)$

At M_{2c} : $t_{12} = (c, 2, -j, -i)$, $t_{13} = (2, c, i, -j)$, $t_{14} = (2, c, -i, j)$, $t_{15} = (c, 2, j, i,)$

$I_{12} = (1, 9)$, $I_{13} = (6, 9)$, $I_{14} = (1, 3) \cup (6, 9)$, $I_{15} = (1, 10)$

At M_{2d} : $t_{16} = (d, 2, -j, -i)$, $t_{17} = (2, d, i, -j)$ $I_{16} = (1, 9)$, $I_{17} = (1, 10)$

At M_{3a} : $t = (3, a, i, j)$, $t_{18} = (3, a, -i, j)$ No free radius, $I_{18} = (1, 7)$

At M_{3b} : $t_{19} = (b, 3, -j, -i)$, $t_{20} = (3, b, -i, j)$, $I_{19} = (1, 9)$, $I_{20} = (6, 9)$

At M_{3c} : $t_{21} = (c, 3, -j, -i)$, $t_{22} = (3, c, -i, j)$, $I_{21} = (6, 9)$, $I_{22} = (1, 10)$

At M_{3d} : $t_{23} = (d, 3, -j, -i)$, $I_{23} = (1, 10)$

In the example z_0 and d_0 are the initial point and direction and z_f and d_f are the final point direction of the robot car.

The following are the sequences of turns for the above example :

1. (t_1, t_2)
2. $(t_1, t_4, t_{12}, t'_{11})$
3. $(t_1, t_2, t'_{10}, t'_{12}, t'_3)$
4. $(t_1, t_2, t'_{10}, t'_{12}, t'_4, t'_1, t'_{18}, t'_{19})$

5. $(t_1, t_6, t_{16}, t'_{10}, t'_2, t'_1, t'_{18}, t'_{19})$

5. Conclusion

In this paper, We present solution method and an example to find a *feasible* path in the case of having obstacles. Until now there are no practical solution and more detail example to solving above problem, therefore we have difficult to apply real robot car operation. Here we formulate example problem LP form. LP formulation is solved easily by usual package i.e. LINDO, GAMS etc. But we need to consider more constraints real robot car, to sake application real fields.

REFERENCES

[1] Dubins, L. "On curves of minimal length with a constraint on Average Curvature and with prescribed initial and final positions and tangents", *American Journal of Mathematics* 79, (1957)

[2] Wilfong, G. "Motion Planning for an Autonomous Vehicle," *IEEE Int. Conf. Robotics and Automation*(1988).

[3] Wilfong, G. "Shortest paths for Autonomous Vehicles", *IEEE Int. Conf. Robotics and Automation*(1989).

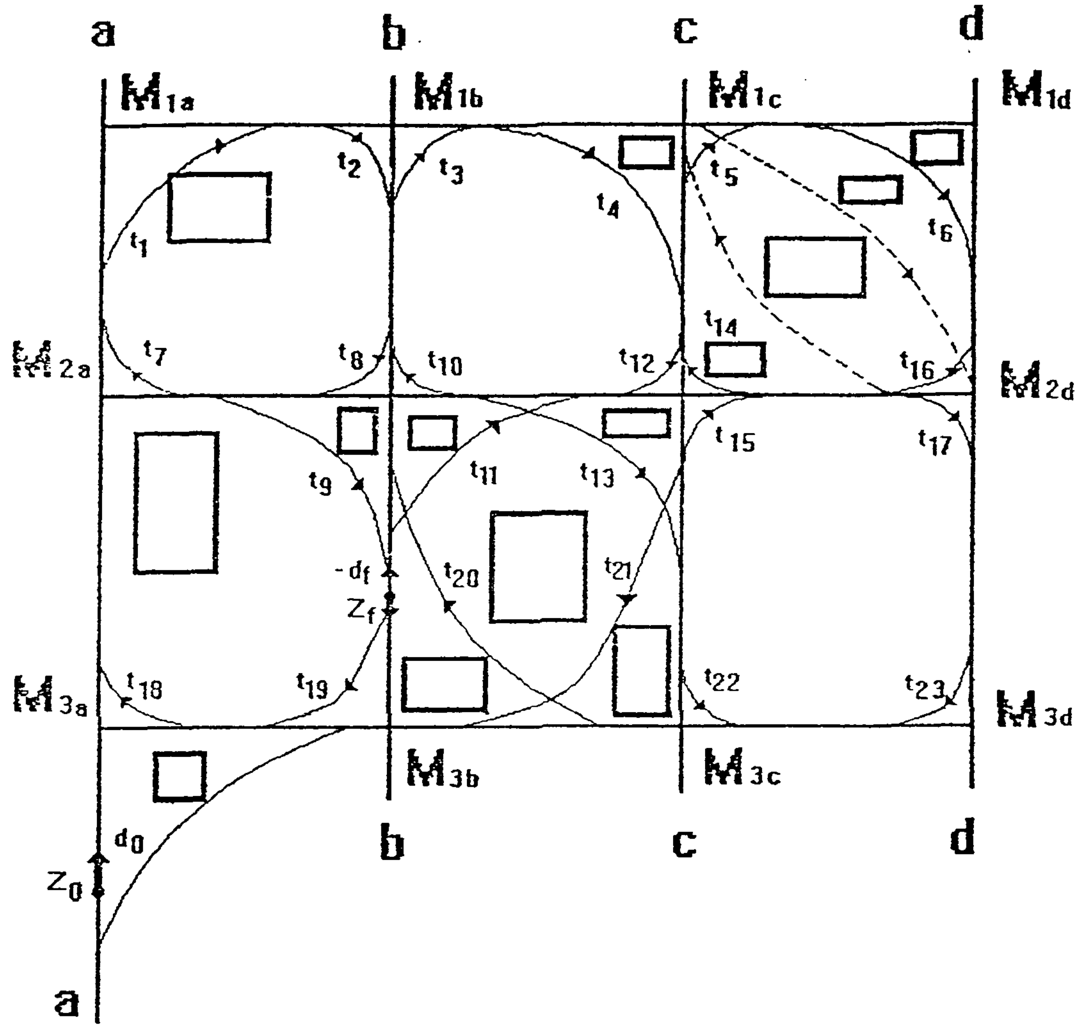

Figure 4 : Workspace including initial and final position