2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

The Speed Graph Method: Time Optimal Navigation Among Obstacles Subject to Safe Braking Constraint

Gil Manor and Elon Rimon
Dept. of Mechanical Engineering, Technion, Israel

Abstract. *This paper describes a method for computing the global time optimal path of a mobile robot navigating among obstacles subject to safe braking constraints. The paper first generalizes the classical Brachistochrone problem into a time optimal navigation problem, where the mobile robot navigates under a braking safety constraint near a point obstacle or a wall segment. The time optimal navigation problem is then formulated for general polygonal environments. Based on this formulation, the paper constructs a* speed graph *for the environment which consists of time optimal arcs that connect critical via points. The speed graph is then used to identify the path homotopy class which most likely contains the global time optimal path. Once a candidate homotopy class is selected, the exact time optimal path subject to safe braking constraints is computed within the homotopy class based on convexity properties of these paths. The results are illustrated with examples, described as readily implementable procedures, and demonstrated with experiments.*

1. Introduction

High speed mobile robots strive to complete their tasks while minimizing overall travel time. This requirement is achieved by maximizing the robot's speed while maintaining velocity dependent safety constraints throughout the navigation process. High speed mobile robots already operate in sentry duty and surveillance [1], where high speed is required. Fast autonomous mobile robots are also expected to replace human drivers in urban environments [14]. Such autonomous systems are currently tested to ensure braking safety under all possible circumstances. Other high speed systems include mobile robots delivering cargo in large warehouses [16], as well as first responder robots we are currently developing.

In all of these applications, traveling *safely* along high speed paths imposes velocity dependant safety constraints. Therefore, pure position based navigation methods such as classical roadmap planning cannot be implemented directly. Rather, a safe high speed path must be planned in the mobile robot's position-velocity configuration space.

This paper describes a method for computing time optimal paths of a mobile robot in known polygonal environments subject to uniform braking safety constraints. Braking safety is encoded as a state (position and velocity) dependant constraint, which allows formulation of the time optimal navigation problem using calculus of variations. The time optimal path minimizes a *time functional,* denoted $T(\alpha)$, defined over all safe paths α connecting the start and target. The paper analyzes the properties of these paths and derives a scheme for computing the time optimal path within each homotopy class of the environment.

As a preliminary step, the classical *Brachistochrone* problem [2] is generalized to the problem of computing safe time optimal paths near a point obstacle or a wall segment. The time optimal navigation problem is then formulated for general polygonal environments. Based on this formulation, the paper introduces a two-stage method for computing safe time optimal paths. In the first stage, critical via points are connected by time optimal arcs into a search graph called the *speed graph.* The speed graph is then used to efficiently select the most promising path homotopy class in terms of total travel time between the start and target. In the second stage, convexity properties of $T(\alpha)$ are used to compute the time optimal path within each homotopy class suggested by the speed graph. Importantly, the two-stage process takes place in the robot's state space, thus treating the geometric path planning and the speed profiling in a single framework.

After discussing the literature in Section 2, the time optimal navigation problem is formulated in Section 3. Section 4 describes analytic solutions for safe time optimal paths near a single obstacle. Section 5 describes important properties of the global time optimal path. Section 6 introduces the speed graph. Section 7 describes a scheme for computing the time optimal path within each homotopy class suggested by the speed graph. Section 8 discusses simulations and experimental results. Finally, the concluding section mentions future extensions of the current research.

2. Relationship to Prior Work

The literature on time optimal path planning subject to safety constraints can be classified into four approaches. The first approach computes a geometric collision free path from start to target, then generates a velocity profile minimizing total travel time along the chosen path subject to velocity and acceleration safety constraints [12, 17]. A second approach plans high speed paths in discrete time steps. In each step along its path, the robot selects a local maneuver that guarantees a fast and safe trajectory in its local environment. Examples of this approach are the *velocity obstacles* method [6], and the *dynamic window* approach [3, 7]. A third approach, usually called *kinodynamics,* considers a discretized state space. For example, [4, 10] discretize the robot's state space into a regular grid, then search the obstacle-free portion of the grid for a minimum travel time path subject to velocity dependent safety constraints. The RRT and RRT* methods [9, 11] construct search trees whose edges represent high-speed feasible maneuvers connecting randomly chosen points to the current tree. A forth approach taken by Wein et al. [18] suggests planning high quality paths (of which time optimal paths is one example) in known environments while considering the geometric and dynamic constraints in a single planning phase. In their work, the Voronoi diagram induced by the obstacles is modified with arcs that minimize a cost function which trades obstacles' clearance with path length. A minimum cost search on the graph yields an approximate optimal path. However, the chosen edge weights have no clear physical meaning. The speed graph yields approximate time optimal paths using physically meaningful weights. Moreover, it suggests a candidate path homotopy class in which the exact time optimal path can be efficiently computed using path convexity properties.

3. Problem Description and Preliminaries

This paper considers a disc robot navigating with high speed in planar environments populated by known stationary polygonal obstacles. The robot is assumed to move freely in the (x,y) plane. The robot's state, $(p,v) \in \mathbb{R}^2 \times \mathbb{R}$, consists of its *position* p and *speed* $v = \|\dot{p}\|$.

When traveling with high speed, the main velocity-dependent safety constraint is the need to maintain safe braking distance with respect to the surrounding obstacles. The robot must be able to brake and reach a *full stop* without hitting any of the surrounding obstacles. We start by defining the braking distance required for safe navigation.

Definition 1. *The robot's* **safe braking distance,** *denoted d, is the length of the shortest kinematically feasible braking path from the robot's current state* (p,v) *to a full stop at* $v=0$.

The safe braking distance, d, can be computed as a function of the robot's speed using energy balance as described in [8, 13]:

$$d(v) = \frac{1}{2\mu g} v^2, \tag{1}$$

where μ is the coefficient of friction at the ground contacts, and g is the gravitational acceleration. Although $d(v)$ does not depend on the robot's mass, m, the friction coefficient μ usually strongly depends on m due to local deformations at the wheels [15][p. 15]. Braking on the verge of sliding determines the robot's maximal deceleration, denoted a, as follows. The net tangential friction force acting on the robot is μmg. The robot's equation of motion during maximal deceleration, $ma = \mu mg$, gives the maximal deceleration $a = \mu g$.

Uniform braking safety is ensured when the robot is kept at least $d(v)$ away from the nearest obstacle at every state (p,v). Note that $d(v)$ is monotonically increasing in v. Hence as the robot travels with higher speeds its safe braking distance *increases*, resulting in a larger circular "safety zone" surrounding the robot. This circular zone must be kept free of any obstacles in order to allow safe deceleration to a full stop. Note, too, that a uniform braking distance is highly conservative. The extension of the method described here to non-uniform braking distance is discussed in the concluding section.

4. Time Optimal Path Near Single Obstacle

This section describes analytic solutions for the safe time optimal path of a point robot near a wall segment or a point obstacle (analytic solutions for a disc are summarized in [13]). We start with a formulation of the travel time variational problem. Let $\alpha(s) : [0,1] \to \mathbb{R}^2$ be the robot's path between the start $p_S = \alpha(0)$ and target $p_T = \alpha(1)$. When the geometric parameter s is parameterized by time, $s(t)$, the robot's velocity along α is given by the chain rule, $\frac{d}{dt}\alpha(s(t)) = \frac{d\alpha(s)}{ds} \cdot \frac{ds(t)}{dt}$. The robot's *speed,* $v(s) = \|\frac{d}{dt}\alpha(s(t))\|$, is given by

$$v(s) = ||\frac{\alpha(s)}{ds}|| \cdot \frac{ds(t)}{dt}.$$

Solving for dt gives:

$$dt = \frac{\|\alpha'(s)\|}{v(s)} ds, \tag{2}$$

where $\alpha'(s) = d\alpha(s)/ds$. Integrating both sides of (2) for $s \in [0,1]$ gives the *travel time functional* along α:

$$T(\alpha) = \int_0^1 \frac{\|\alpha'(s)\|}{v(s)} ds. \tag{3}$$

The integrand, denoted $G\big(\alpha'(s), \alpha(s), s\big)$ is given by

$$G\big(\alpha'(s),\alpha(s),s\big) = \frac{\|\alpha'(s)\|}{v(s)}.$$

Based on calculus of variations, any extremal path of $T(\alpha)$ over all piecewise smooth paths α connecting p_S and p_T must satisfy the Euler-Lagrange equation:

$$\frac{\partial G}{\partial \alpha} - \frac{\partial}{\partial s}\Big(\frac{\partial G}{\partial \alpha'}\Big) = 0, \tag{4}$$

where α and α' are to be treated as formal variables.

The safe extremal path near a wall segment or a point obstacle is derived as follows [13]. Let O_i denote a wall segment or a point obstacle, and let $\mathsf{dst}(\alpha(s),O_i)$ denote the minimal distance between the robot positioned at $\alpha(s)$ and the obstacle O_i. Braking safety requires that $\mathsf{dst}\big(\alpha(s),O_i\big) \geq d(v(s))$ for $s \in [0,1]$. Since $d(v) = \frac{1}{2a}v^2$ where a is the robot's maximal deceleration, braking safety becomes the inequality $2a \cdot \mathsf{dst}\big(\alpha(s),O_i\big) \geq v^2(s)$ for $s \in [0,1]$.

First consider a wall segment. When the wall segment is aligned with the x axis, $\mathsf{dst}\big(\alpha(s),O_i\big) = y(s)$ along a path $\alpha(s) = (x(s),y(s))$. The safety constraint thus becomes $v^2 \leq 2ay$. Travel time minimization requires maximal velocity. Substituting $v = \sqrt{2ay}$ in G, then solving the Euler-Lagrange equation gives a cycloid as the extremal path. The cycloid is parameterized by $\alpha(\varphi) = \big(x(\varphi),y(\varphi)\big)$ for $\varphi \in [\varphi_S,\varphi_T]$, where φ is the angle of a fixed point on a rolling circle that generates the cycloid:

$$\begin{pmatrix} x(\varphi) \\ \\ y(\varphi) \end{pmatrix} = \begin{pmatrix} A \\ \\ 0 \end{pmatrix} + \begin{pmatrix} \frac{1}{4ac_w^2}(\varphi - \sin\varphi) \\ \\ \frac{1}{4ac_w^2}(1-\cos\varphi) \end{pmatrix}.$$

The parameters A and c_w are uniquely determined by the endpoints p_S and p_T.

The safe extremal path near a point obstacle is parametrized in polar coordinates, $\alpha(\theta) = (\theta, r(\theta))$ for $\theta \in [\theta_S,\theta_T]$. When the point obstacle O_i is located at the origin, $\mathsf{dst}\big(\alpha(\theta),O_i\big) = r(\theta)$, and the safety constraint becomes $v^2 \leq 2ar$. Substituting the maximal allowed velocity $v = \sqrt{2ar}$ in G, then solving the Euler-Lagrange equation, gives the extremal path:

$$r(\theta) = 2ac_p^2\Big(\tan^2\Big(\tfrac{1}{2}\theta - \tfrac{1}{2}\theta_S + \tan^{-1}\sqrt{\frac{r_S}{2ac_p^2} - 1}\Big) + 1\Big)$$

where $r(\theta_S) = r_S$. The parameter c_p is uniquely determined by the end condition $r(\theta_T) = r_T$. As next discussed, the two time optimal paths are *local minima* of the travel time functional $T(\alpha)$.

5. Basic Properties of the Time Optimal Path in Multiple Obstacle Environments

We start by formulating the time optimal variational problem in a multiple obstacles environment. When a mobile robot is subjected to a uniform braking constraint, its maximal allowed speed is governed by its distance to the nearest obstacle. Suppose the environment contains the obstacles $O_1,\ldots,O_k$. The robot's safe braking distance is given by $d(v) = \frac{1}{2a}v^2$, where $a = \mu g$ is the robot's maximal deceleration. For braking safety, the robot must keep at least $d(v)$ away from the nearest obstacle. Thus, when the robot moves along a collision free path $\alpha(s)$, its braking distance $d(v)$ must satisfy the safety constraint:

$$\min_{i=1\ldots k}\{\mathsf{dst}(\alpha(s),O_i)\} \geq d\big(v(s)\big) \qquad s \in [0,1], \tag{5}$$

where $\mathsf{dst}(\alpha(s),O_i)$ is the minimal distance between the robot positioned at $\alpha(s)$ and the obstacle O_i. Travel time minimization requires maximal speed, and according to (5) the maximal *safe* speed is given by

$$v(s) = \sqrt{2a \cdot \min_{i=1\ldots k}\{\mathsf{dst}(\alpha(s),O_i)\}},$$

The travel time functional to be minimized is given by

$$T(\alpha) = \int_0^1 G\big(\alpha'(s),\alpha(s)\big)ds,$$

where

$$G\left(\alpha(s),\alpha'(s)\right) = \frac{\|\alpha'(s)\|}{\sqrt{2a \cdot \min_{i=1\ldots k}\{\mathsf{dst}(\alpha(s),O_i)\}}}.$$

Next three basic properties of the time optimal path are described. Consider a planar environment, denoted $\mathfrak{F}$, containing multiple internal obstacles and surrounded by an outer boundary. The set of piecewise differentiable paths in $\mathfrak{F}$, is denoted M. Let us start with the notion of a path homotopy class.

Definition 2. *Two continuous paths with the same endpoints, $\alpha,\beta : [a,b] \to \mathfrak{F}$, belong to the same* **homotopy class** *if there is a continuous mapping, $F : [a,b]\times[0,1] \to \mathfrak{F}$, such that $F(t,0) = \alpha(t), F(t,1) = \beta(t)$ for $t \in [a,b]$.*

Every path homotopy class forms a connected component in M, since two paths α and β in the same homotopy class can be connected by the path homotopy. The first basic property concerns the smoothness of any extremal path of $T(\alpha)$.

Proposition 5.1 ([13]). *Every extremal path of $T(\alpha)$ in $\mathfrak{F}$ is a $C^{(1)}$* **curve** *having a continuously varying tangent.*

The second property concerns the time minimality of any extremal path of $T(\alpha)$.

Proposition 5.2 ([13]). *Let α_{opt} be an extremal path of $T(\alpha)$. Then α_{opt} is a* ***local minimum*** *of $T(\alpha)$ in a local neighborhood of α_{opt} in the metric space M.*

The third property ensures the existence of a unique time minimal path.

Proposition 5.3 ([13]). *The travel time functional, $T(\alpha)$, possesses a* **unique** *time minimal path in every homotopy class of paths connecting p_S and p_T in $\mathfrak{F}$.*

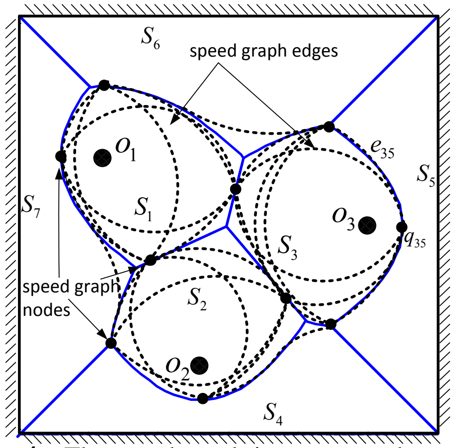


Figure 1. The speed graph in a rectangular environment populated by three point obstacles.

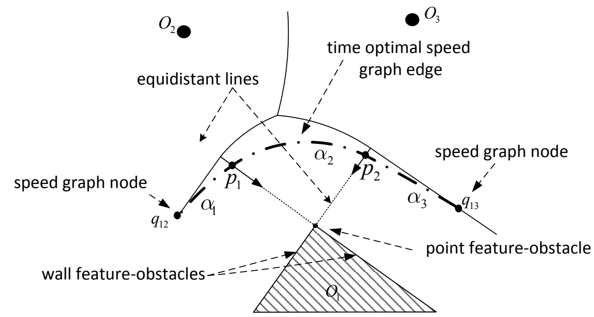


Figure 2. A speed graph edge in the vicinity of a polygonal obstacle.

Every extremal path of $T(\alpha)$ is a local minimum according to Proposition 5.2. Since $T(\alpha)$ possesses a unique local minimum in each path homotopy class according to Proposition 5.3, every extremal path of $T(\alpha)$ must be the unique minimum in the path's homotopy class.

6. The Speed Graph

The speed graph is first constructed for environments populated by point obstacles, then extended to general polygonal environments.

Consider an environment containing multiple point obstacles $O_1,\ldots,O_k$. Let e_{ij} denote the Voronoi edge passing between adjacent obstacles O_i and O_j (e.g. the edge e_{35} in Figure 1). Note that e_{ij} is equidistant from O_i and O_j. Denote by q_{ij} the point along e_{ij} where the distance to the obstacles is minimal.

The speed graph nodes: The speed graph nodes consist of all *local minimum* points q_{ij}. These nodes lie on the Voronoi diagram, such that each Voronoi edge contains a single node (Figure 1).

The speed graph edges: Given an obstacle O_i, the *Voronoi cell* of O_i, denoted S_i, is the region surrounding O_i and bounded by Voronoi edges (Figure 1). Let $\mathbf{V}(O_i)$ be the set of speed graph nodes located on the boundary of S_i. The speed graph edges are the *safe time optimal arcs* connecting every pair of nodes in $\mathbf{V}(O_i)$. The *cost* of each edge is the value of the safe travel time functional $T(\alpha)$ along this edge[13].

Note that q_{ij} lies either in the interior or at an endpoint of e_{ij} (q_{35} in Figure 1). Also note that the number of Voronoi edges is linear in the number of obstacles, and consequently there are $O(k^2)$ speed graph edges.

Every start and target, p_S and p_T, can be connected by time optimal arcs to the speed graph. The time optimal path from p_S to p_T along the augmented speed graph can then be computed using any standard graph search algorithm. This path, denoted α_G, is necessarily piecewise smooth. Hence α_G *cannot* be an extremal path of $T(\alpha)$. However, α_G suggests the *most promising* path homotopy class in terms of total travel time.

The Voronoi saturation problem: In environments populated by multiple point obstacles, the time optimal path near a particular point obstacle may emerge out of its Voronoi cell and penetrate a neighboring cell. In order to find the true time optimal path between two speed graph nodes, a *saturation method* is proposed and explained in the technical report [13].

The speed graph in next extended to polygonal environments. Each polygonal obstacle perimeter consists of line segments joined by vertices. These can be treated as individual point or wall segment obstacles called *obstacle-features*. When the robot travels between polygonal obstacles, it is in the vicinity of a single obstacle-feature (a vertex or edge) at each instant along its path.

Consider two nodes that lie on the boundary of a common Voronoi cell (e.g. q_{12} and q_{13} in Figure 2). To construct the time optimal arc between these nodes, consider the proximal obstacle-features of the polygonal obstacle at the center of the Voronoi cell (e.g. O_1 in Figure 2). Assume the speed graph edge crosses l equidistant lines spanned between adjacent obstacle-features of the polygonal obstacle. The speed graph edge is consequently affected by $l+1$ feature-obstacles (e.g. $l+1=3$ obstacle-features in Figure 2). The speed graph edge can thus be parameterized by the points $p_1,\ldots,p_l$ where the edge crosses the equidistant lines. Each crossing point p_i can freely vary along the i'th equidistant line, but only in the segment bounded by the obstacle and the Voronoi edge. Since each speed graph edge is required to be time optimal, it must possess a continuous tangent at $p_1,\ldots,p_l$. Computing the optimizing values of $p_1,\ldots,p_n$ is done by solving a suitable convex optimization problem (see [13]).

7. Computation of the Time Optimal Path

This section describes a scheme for computing the time optimal path within a specific homotopy class suggested by the speed graph. Starting with the speed graph time optimal path α_G, let $p_1,\ldots,p_n$ be the crossing points of α_G with the Voronoi diagram. At these points α_G is *not* necessarily smooth. To find the global time optimal path within the homotopy class of α_G, the position of the crossing points along the Voronoi diagram which smooth the path should be found. The resulting $C^{(1)}$ path will consist of time optimal path segments in each Voronoi cell, and hence will satisfy the conditions for the global time optimal path [13]. The procedure accepts as input the initial crossing points $p_1,\ldots,p_n$ and consists of three stages.

Time optimal path computation:

(i) At each crossing point p_i, determine the local smoothing direction of α_G along the *acute angle* spanned by the tangents of the path segments of α_G meeting at p_i. Denote by v_{p_i} the Voronoi node towards which p_i should locally move. Move p_i to v_{p_i}, then compute the path's smoothing direction at v_{p_i}. Compare it with the original smoothing direction at p_i. If both directions point toward each other (Figure 3(a)), denote the Voronoi segment between p_i and v_{p_i} as e_i. Otherwise, move the crossing point p_i to v_{p_i}, then split p_i into two points p_{i_1} and p_{i_2} (Figure 3(b)).

(ii) Let $p_1,\ldots,p_{\bar{n}}$ be the crossing points obtained at the end of stage (i), where $\bar{n}\geq n$. At this stage, $p_1,\ldots,p_{\bar{n}}$, lie on their respective optimal Voronoi edges, $e_1,\ldots,e_{\bar{n}}$. Compute the *exact* position of the optimal crossing points, $(p_1^*,\ldots,p_{\bar{n}}^*)$, on the Voronoi segments $e_1,\ldots,e_{\bar{n}}$ as suggested below.

(iii) Obtain the globally time optimal path by connecting each pair of adjacent crossing points, p_i^* and p_{i+1}^*, with a time optimal arc within the Voronoi cell containing these points.

To complete stage (ii) of the procedure, recall that the time optimal path must have a continuously varying tangent. By requiring tangents continuity at the crossing points, the computation of the optimal crossing points is reduced into a system of $\bar{n}$ equations in $p_1,\ldots,p_{\bar{n}}$. This system can be readily solved by any numerical solver such as Newton's method (see [13]).

Example: Figure 4 depicts a rectangular environment populated by four point obstacles O_1, O_2, O_3, and O_4. The speed graph for the environment is constructed, then the time optimal path α_G is found along the speed graph. The optimizing positions of the crossing points are found using Newton's method (stopping after 5 consecutive iterations), and the time optimal path α_{opt} within the homotopy class of α_G is computed (Fig. 4).

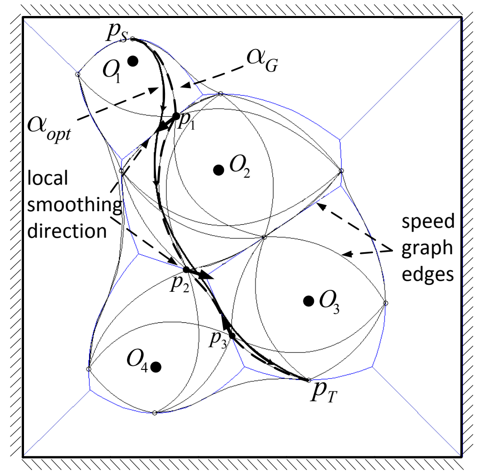


Figure 4. The time optimal path from p_S to p_T suggested by the speed graph.

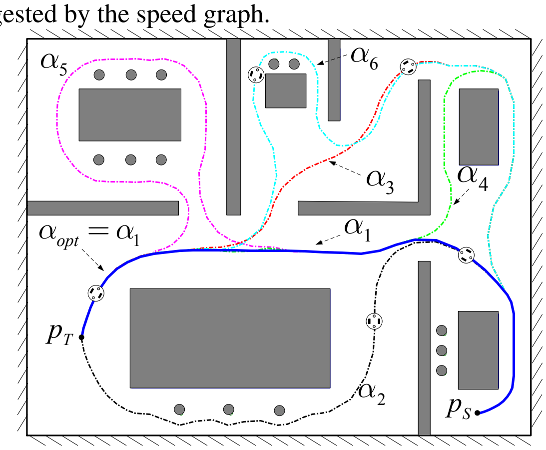
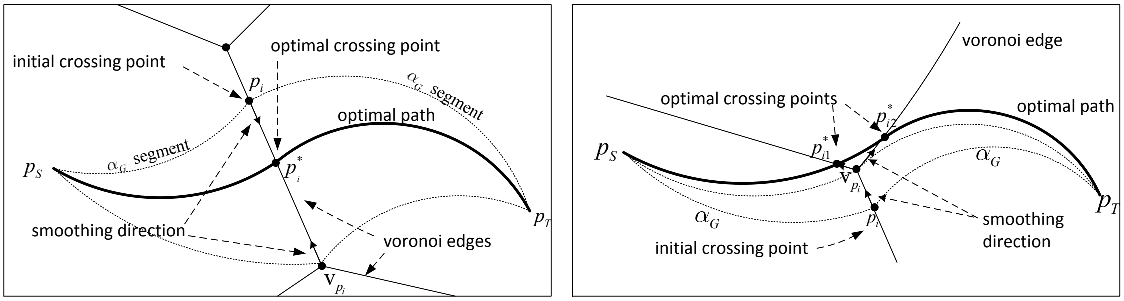


Figure 5. The six best safe high-speed paths in an office floor environment.

8. Simulations and Experiments

Figure 5 depicts a simulated office floor environment containing inner walls and pieces of furniture. The size of the environment is 3×4 meters, and the robot's maximal deceleration is $a=0.1\ \mathrm{m/sec^2}$. For a robot navigating in speeds of $0.5\ \mathrm{m/sec^2}$, the characteristic braking distance would be 1.25 meters. The start and target, p_S and p_T, can be connected by various homotopy classes in this environment. The six best homotopy classes along the speed graph in terms of total travel time were selected. The safe time optimal path in each homotopy class, denoted $\alpha_1,\ldots,\alpha_6$, was then computed using the computation scheme of Section 7 (see Figure 5). The travel time along these optimal paths are: $27.3, 35.7, 42.6, 46.7, 53.3, 56.4$sec. The global time optimal path in the environment is achieved by the bold path $\alpha_1=\alpha_{opt}$ depicted in Figure 5. Based on the travel time results, the $\mathbf{k}=1$ homotopy class provides the best total travel subject to safe braking constraints. For more examples and some preliminary ex-



(a) (b)

Figure 3. (a) Local shifting of p_i along a Voronoi edge. (b) Local splitting of p_i into two crossing points p_{i1} and p_{i2}.

periments with an Irobot platform see [13].

Complexity: In polygonal environments with a total of n obstacle-features (vertex or edge) there are at most $O(n)$ Voronoi edges. Consequently ,the speed graph's size is $O(n^2)$. A time minimal path along the graph can be found in $O(n^2 \log n)$. Next, the exact crossing points which bring to minimum the travel time are computed using a convex optimization algorithm. This computation is $O(n^3 \log \frac{1}{\varepsilon})$, where ε is the desired accuracy.

9. Conclusion

This paper described the speed graph method for computing time optimal paths of a mobile robot traveling among obstacles subject to uniform braking safety constraints. The speed graph method first constructs a *speed graph* for the environment. The time optimal path along the speed graph, α_G, is then computed and subsequently used to suggest the most promising path homotopy class in terms of total travel time. Basic properties of the travel time functional $T(\alpha)$ guarantee that the resulting path is the global time optimal path within the homotopy class of α_G. Simulations and experiments support the speed graph's usefulness in suggesting the path homotopy class that gives the best travel time in the environment.

The following are two extensions of the current work. The first extension concerns the computation of the safe time optimal path for a disc robot rather than a point robot. The time optimal path of a disc robot near a point obstacle or a wall segment is given in terms of an *elliptic integral.* While considered to be a "closed form" formula, a simpler analytic expression for the time optimal path is desired. A second extension concerns the addition of the robot's *heading*, into the time optimal navigation scheme. Under this extension the braking safety constraint becomes a function of the robot's speed as well as the robot's heading. This extension seems to require optimal control tools rather than calculus of variations tools. While extremely challenging, its solution will yield much more practical *safe* high speed paths in environments containing narrow passages and corridors.

References

[1] GUARDIUM-UGV. *G-NIUS Unmanned Ground Systems*, accesssed Jan. 2014. http://g-nius.co.il.

[2] C.B. Boyer and U.C. Merzbach. *A History of Mathematics*. New York, Wiley, 1991.

[3] O. Brock and O. Khatib. High-speed navigation using the global dynamic window approach. In *ICRA*, 1999.

[4] B. Donald, P. Xavier, J. Canny, and J. Reif. Kinodynamic motion planning. *Journal of the ACM*, 1993.

[5] D. Eppstein. Finding the k shortest paths. *SIAM Journal on Computing*, 1998.

[6] P. Fiorini and Z. Shiller. Motion planning in dynamic environments using velocity obstacles. *IJRR*, 1998.

[7] D. Fox, W. Burgard, and S. Thrun. The dynamic window approach to collision avoidance. *Robotics Automation Magazine, IEEE*, 1997.

[8] T. Fraichard. A short paper about motion safety. In *ICRA*, 2007.

[9] S. Karaman and E. Frazzoli. Sampling-based algorithms for optimal motion planning. *IJRR*, 2011.

[10] R. Kindel et Al. Kinodynamic motion planning amidst moving obstacles. In *ICRA*, 2000.

[11] S.M. Lavalle. Rapidly-exploring random trees: A new tool for path planning. *Technical report*, Dept. of Computer Science, Iowa State University, 1998.

[12] M. Lepetic et Al. Time optimal path planning considering acceleration limits. *Robotics and Autonomous Systems*, 2003.

[13] G. Manor and E. Rimon. Time optimal navigation among obstacles subject to safe braking constraint. *Technical Report*, accesssed Jan. 2014. http://robots.technion.ac.il/publications.htm.

[14] J. Markoff. Google cars drive themselves, in traffic. *The New York Times*, October 9th 2010.

[15] R.H. Smith. *Analyzing Friction in the Design of Rubber Products and Their Paired Surfaces*. CRC Press, 2008.

[16] K.C. Teixeira, M. Becker, and G.A.P. Caurin. Automatic routing of forklift robots in warehouse applications. In *ABCM Symposium in Mechatronics*, 2010.

[17] E. Velenis and P. Tsiotras. Optimal velocity profile generation for given acceleration limits: theoretical analysis. In *American Control Conference*, 2005.

[18] R. Wein, J. van den Berg, and D. Halperin. Planning high-quality paths and corridors amidst obstacles. *IJRR*, 2008.