2007 IEEE International Conference on
Robotics and Automation
Roma, Italy, 10-14 April 2007



# Generation of Homotopic Paths for a Size-Changing Sphere

Enrique J. Bernabeu

***Abstract*—A new and fast technique for generating collision-free paths for a mobile sphere, whose size (radius) dynamically changes, is introduced in this paper. A collision between the mobile-sphere motion and an obstacle is predicted by computing its minimum translational distance. This distance parameterizes an intermediate configuration for the sphere, which is in contact with the obstacle or separated a configurable distance. Collision with a given obstacle is then avoided by forcing the mobile sphere to pass through a selected intermediate configuration. This paper describes a technique to reduce or to grow this configuration (sphere) in order to bring it into contact (or as close as desired) with a second obstacle. This fact provides the path planner some remarkable properties like it really generates a set of homotopic paths in narrow environments for not only a sphere but any other mobile object which always keeps inside the volume represented by the generated path.**

## I. Introduction

MOTION planning is an extensive studied area in robotics. Many techniques have been published and their application settings are continuously growing. A deep and wide review on planning algorithms is found in [1].

A new path-planning technique, centered on dealing with some encouraging points in motion planning: size-changing mobile objects, narrow corridors and enveloping the free space between obstacles, is introduced in this paper.

Although path planning is constrained to two degrees of freedom, obstacles are modeled by 3D spherically-extended polytopes. Mobile object is modeled by a sphere with the ability of changing dynamically its radius. Obstacle motions are also constrained to the same plane that the mobile sphere. It is assumed that obstacles' positions are measurable.

The proposed path-planning algorithm is based on the computation of the minimum translational distance [2] between a given mobile-sphere motion and an obstacle. Such a distance characterizes an intermediate configuration (sphere) for the mobile sphere which is in contact with the involved obstacle. When a collision is predicted, a new motion for avoiding such an obstacle is defined by its associated intermediate configuration. This avoidance strategy is based on the previous work in [3].

When an intermediate configuration defining a new motion has been selected, this configuration is immediately tested if it collides with a nearby obstacle. A fast technique is then applied to reduce the configuration size until such a collision disappears.

Collaterally, if this technique is applied to a configuration which does not collide with an obstacle, this configuration is grown, if desired, until it gets in contact with the obstacle.

After that, the modified intermediate configuration defines a new mobile-sphere motion. New collision tests between this new motion and the obstacle are required again.

A collision-free path for the size-changing mobile sphere is finally generated under real-time constraints. Path planner is so fast that can be run as frequently as new information from the sensor system is received.

The path generated envelopes the free-space volume swept by the mobile sphere. Given that the mobile sphere change its size, this volume really represents a set of homotopic paths for any sphere or generic mobile object that always fits inside the above-mentioned free space.

Generally, when this collision-free path is provided to a motion-planning algorithm dealing with complex objects, the collision-detection problem is then transformed into checking if the mobile object is inside the given free space.

## II. Spherically-extended Polytope

A spherically extended polytope (s-tope) is the convex hull of a finite set of spheres. A sphere is denoted $s=(c,r)$ where $c$ is the center and $r$ is its radius. Given the set of spheres $\mathrm{S}=\{s_0,s_1,\ldots,s_n\}$, the convex hull of such a set, $S_\mathrm{S}$, contains an infinite set of swept spheres expressed by

$$S_\mathrm{S}=\left\{s=(c,r):c=c_0+\sum_{\mathrm{i}=1}^{n}\lambda_\mathrm{i}(c_\mathrm{i}-c_0),\ r=r_0+\sum_{\mathrm{i}=1}^{n}\lambda_\mathrm{i}(r_\mathrm{i}-r_0),\right.$$
$$\left.s_\mathrm{i}=(c_\mathrm{i},r_\mathrm{i})\in \mathrm{S},\lambda_\mathrm{i}\geq 0,\sum_{\mathrm{i}=1}^{n}\lambda_\mathrm{i}\leq 1,\ \mathrm{i}=0,\cdots n\right\} \quad (1)$$

The convex hull of spheres does not include all possible spheres which can fit inside the s-tope, only those that are generated by (1). Spheres in S are called spherical vertices. S-tope order is the number of spherical vertices. Graphical examples of s-topes are shown along the paper.

An s-tope is said to be overspecified if one or more of its spherical vertices can be removed without changing the convex hull. These spherical vertices are called redundant. An s-tope which is not overspecified is called valid.

As a polytope with four or more points is a polyhedral object with triangular facets, tetra-spheres (or greater-order s-topes) are composed of tri-sphere facets [4]. A tri-sphere has three bi-spherical edges. The simplest s-tope is a sphere.

Manuscript received September 15, 2006. This work has been partially funded by the Generalitat Valenciana project with reference GV06/115, MASMICRO European Project number 500095-2 and FEDER projects with references DPI2005-08732-C02-02 and DPI2006-15320-C03-01.

Enrique J. Bernabeu is with the Instituto de Automática e Informática Industrial, Universidad Politécnica de Valencia, E-46022, Valencia, SPAIN (phone: +34-963877007 Ext. 88232, fax: +34-963879816; e-mail: ebernabe@isa.upv.es).

A bi-sphere is the s-tope characterized by a set of two spheres $\{(c_0,r_0), (c_1,r_1)\}$. The axis of a bi-sphere is characterized by the vector $c_1-c_0$. Degree η and angle of convergence α of a bi-sphere are respectively defined as follows

$$\eta = \frac{r_1 - r_0}{\| c_1 - c_0 \|}; \quad \alpha = \sin^{-1}(\eta) \tag{2}$$

where $\|\cdot\|$ is the Euclidean norm.

Bi-sphere will degenerate into a single sphere [5] (equal to its bigger spherical vertex) when the degree of convergence does not verify the condition $\eta \in [-1,1]$.

## III. Generation of a Collision-Free Path for a Mobile Size-Changing Sphere

Given a mobile sphere, whose size dynamically changes and a set of obstacles modeled by s-topes, a technique for generating a collision-free path for the mentioned sphere is introduced in this section.

### A. Problem Formulation

This technique also requires as inputs the start $s_s=(c_s,r_s)$ and goal $s_g=(c_g,r_g)$ configurations. These two configurations define a bi-sphere. This bi-sphere $S_{sg}$ represents the motion of the mobile sphere from its initial position to its final one. Each one of the infinite intermediate positions of the mobile sphere from its start to its goal position is parameterized as

$$S_{sg} = \{s = (c,r) : c = c_s + \lambda(c_g - c_s), \\ r = r_s + \lambda(r_g - r_s), \lambda \in [0,1]\} \tag{3}$$

Note that (3) is the formal definition of a bi-sphere.

If $r_s \neq r_g$, radius of the mobile sphere is progressively changing from its initial to its final value.

A collision-free path from the start to the goal configuration is obtained by computing the distance between the mobile-sphere motion and each one of the obstacles. When a collision is predicted, a new collision-free intermediate configuration is generated in order to avoid such a collision.

### B. Distance Computation.

The presented technique is based on a version of the GJK algorithm [4], called GJK* algorithm [2]. This algorithm computes the minimum translational distance (MTD) between two s-topes.

Each one of the obstacles is modeled by an s-tope. A separation or penetration distance between the bi-sphere, which states the mobile-sphere motion, and an obstacle is computed as the distance between the origin point and the Minkowski difference set of the involved s-topes.

Minkowski difference s-tope of two s-topes $S_A$, $S_B$, defined respectively by the sets of spheres $A$ and $B$, is given by the following spherical vertices [6],

$$S = S_A - S_B = \{s = (c,r) : c = c_A - c_B, r = r_A + r_B, \\ (c_A, r_A) \in A, (c_B, r_B) \in B\} \tag{4}$$

As MTD is a translational distance, MTD between two s-topes $S_A$, $S_B$ is defined as follows: if one of the s-topes is translated MTD, distance between $S_A$, $S_B$ is then zero.

MTD predicts a collision between the bi-sphere representing the mobile-sphere motion with an obstacle.

Anyway, two previous and important aspects have to be considered before running GJK* algorithm. First, given that motions are constrained to a plane, the 3D s-topes used for modeling obstacles are composed of just bi-spherical facets. Second, note that MTD has to be computed normal to the vector $c_g-c_s$, otherwise when a collision is predicted such a collision will not be properly avoided [2].

For this application, GJK* algorithm requires as an additional input a set of as maximum three spherical vertices of the involved Minkowski difference s-tope. Algorithm ends returning the following information

$$S' = \{c'_0,...,c'_l\};\ l \leq 2;\ O^{\perp} = c'_0 + \lambda(c'_l - c'_0) \\ \lambda \in [0,1];\ d_O = \| O^{\perp} \|\ ;\ \hat{v}_{MTD} = O^{\perp}/d_O \tag{5}$$

where $S' = \{c'_0, c'_l\}$ with $l \leq 1$ states the centers of the spherical vertices $\{(c'_0,r'_0),(c'_l,r'_l)\}$ that define the closest feature (sphere or bi-sphere) of the Minkowski difference s-tope to the origin point O. If $l$=1, directions hold by vectors $c_g-c_s$ and $c'_1 - c'_0$ are equal. $O^{\perp}$ is the projection of the origin point onto the structure defined by the centers in $S'$. Parameter λ states the $O^{\perp}$ relative position with respect to the spheres' centers in $S'$. Note that if $l$=0, no parameter λ is returned. In accordance with (3), λ characterizes the sphere in the mobile-sphere motion which is used to determine the minimum translational distance. When GJK* algorithm ends with $l$=0, it means that such a sphere in the mobile-sphere motion is the start or the goal configuration. Case $l$=0 is considered as a particular case of $l$=1 with λ=0 or λ=1. $d_O$ is the distance from O to the structure defined by the centers of the spherical vertices, which define the Minkowski difference s-tope. Finally, $\hat{v}_{MTD}$, with $\|\hat{v}_{MTD}\|=1$, states the translational vector of the MTD which is normal to $c_g-c_s$.

If $l \leq 1$, the minimum translational distance MTD between the origin point and the Minkowski difference s-tope, i.e., the MTD between mobile-sphere motion and a given obstacle is obtained as follows

$$\mathrm{MTD} = d_O - \left(r'_0 + \lambda \cdot (r'_1 - r'_0)\right) \quad \text{with } \lambda = -\frac{c'_0 \cdot (c'_1 - c'_0)}{\| c'_1 - c'_0 \|^2} \tag{6}$$

Anyway, the MTD computed by (6) is only true if $r'_0 = r'_1$, i.e., if spheres' radii returned by the GJK* algorithm are equal. In accordance with the given approach, it is trivial to proof that case $r'_0 \neq r'_1$ is just presented when $r_s \neq r_g$.

If $r'_0 \neq r'_1$, parameters λ, $d_O$, MTD and $\hat{v}_{MTD}$ has to be updated and respectively referred to as λ', $d'_O$, MTD' and $\hat{v}'_{MTD}$. Computation of these new parameters is shown in fig. 1.

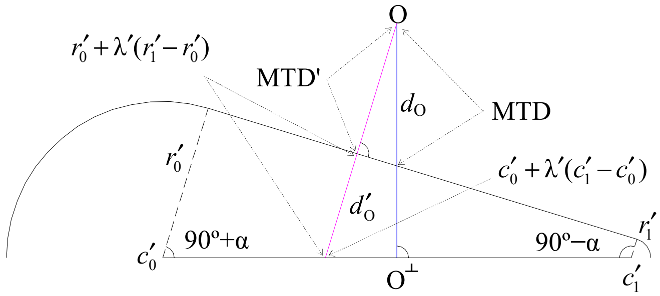


Fig. 1. Minimum translational distance computation when GJK* algorithm finishes returning two spherical vertices (bi-sphere) with angle of convergence α≠0. For clarity, bi-sphere has been schematically depicted. Real bi-sphere is obtaining by rotating the image upon its axis. α is negative

Attending to fig. 1, such parameters are obtained as follows

$$\lambda' = \lambda + \frac{d_O \cdot \tan\alpha}{\| c_1' - c_0' \|} \; ; \; d_O' = \frac{d_O}{\cos\alpha}$$
$$v'_{MTD} = O^\perp + \frac{d_O \cdot \tan\alpha}{\| c_1' - c_0' \|}\cdot(c_1'-c_0') = c_0' + \lambda'\cdot(c_1'-c_0') \tag{7}$$
$$\hat{v}'_{MTD} = v'_{MTD}/d_O' \; ; \quad \| \hat{v}'_{MTD} \| = 1$$

where α is the corresponding angle of convergence.

The right MTD' is then computed as

$$\mathrm{MTD}' = d_O' - \left(r_0' + \lambda'\cdot(r_1'-r_0')\right) \tag{8}$$

Note that the sign of the computed minimum translational distance codifies the relationship between the mobile-sphere motion and the involved obstacle. In this way, if MTD' is negative a collision is predicted, and consequently a penetration distance has been computed. If MTD' is zero, the mobile sphere, along its motion, will be in contact with the obstacle, whereas if MTD' is positive, its value state the maximum approach of the mobile sphere along its motion to the obstacle.

Nevertheless, GJK* algorithm can end with $l$=2. In this case, only a set of three spherical vertices is returned. Therefore, it means that O is inside the area defined by the centers of the spherical vertices, which define the mentioned Minkowski difference s-tope. In order words, obstacle is divided by the motion axis. Consequently, a collision is presented between the mobile-sphere motion and the obstacle. In order to compute the corresponding penetration distance, a version of the GJK* algorithm is run in triplicate [2]. Each one receives as an initial set, one of the three different bi-spheres obtained from the spherical vertices previously returned.

Note that this GJK*-algorithm version returns the distance from the origin point to the axis of the external bi-spherical facet which is the closest to the initially provided bi-sphere.

After finishing this GJK*-algorithm version, the minimum translational distance is computed by applying the corresponding equations (6), or (7) and (8). However, two important differences have to be considered. First, MTD expression associated with the radius in (6) or (8) is added instead of subtracted and afterwards its sign is set to be negative. Second, the sign of the vector that states the translational distance has to be changed, since such a distance has been computed in the opposite direction (from inside to outside).

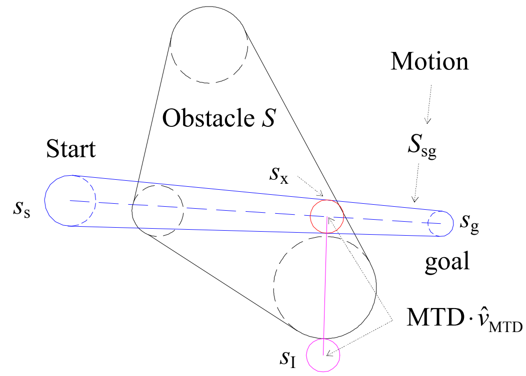


Fig. 2. Computation of the minimum translational distance between a mobile-sphere motion and an obstacle modeled by a tri-sphere.

As this version of the GJK* algorithm is run more than once. Several translational distances are obtained. As all these translational distances are normal to the motion axis, those ones that are maximal for each direction are selected. The lowest one of these two states the minimum penetration distance between the mobile-sphere motion and the obstacle.

It is important to remark that in the same way that in [4], GJK* algorithm does not previously require to compute the Minkowski difference s-tope. Consequently, its complexity is linear with the total number of the spherical vertices

### C. Collision-Free Path Generation

From now and for clarity, with independence of cases $r_s$=$r_g$ or $r_s$≠$r_g$, parameters obtained after computing the minimum translation distance are going to be referred to as λ, $d_O$, MTD and $\hat{v}_{MTD}$.

This distance-computation technique is fast enough to be used as a path-planner mechanism [2].

An important advantage is that the parameters returned by the distance-computation algorithm characterize the sphere position in the motion whose translation is minimal to bring it into contact with the involved obstacle.

Let $S_{sg}$ be the bi-sphere representing the motion of a mobile sphere from a start position to a goal one, which are respectively given by $s_s$=($c_s$,$r_s$) and $s_g$=($c_g$,$r_g$). Let $S$ be a given obstacle modeled by an s-tope. Let $s_x$=($c_x$,$r_x$) be the sphere in the motion which is translated MTD to bring it into contact with $S$. Let $s_I$=($c_I$,$r_I$) be such a sphere position touching $S$.

After computing the separation or penetration distance between the mention motion and the obstacle $S$, spheres $s_x$, $s_I$, are determined by means of the known parameters λ, MTD, $\hat{v}_{MTD}$ as follows

$$\begin{aligned} s_x &= (c_x, r_x):\; c_x = c_s + \lambda(c_g - c_s)\,;\; r_x = r_s + \lambda(r_g - r_s) \\ s_I &= (c_I, r_I):\; c_I = c_x - (\mathrm{MTD} - \delta)\cdot\hat{v}_{MTD}\,;\; r_I = r_x \end{aligned} \tag{9}$$

δ≥0 states a safety threshold. If δ=0, sphere $s_I$ is in contact with the involved obstacle. A graphical example is shown in fig. 2. For clarity, fig. 2 has been represented in 2D. In fig. 2,

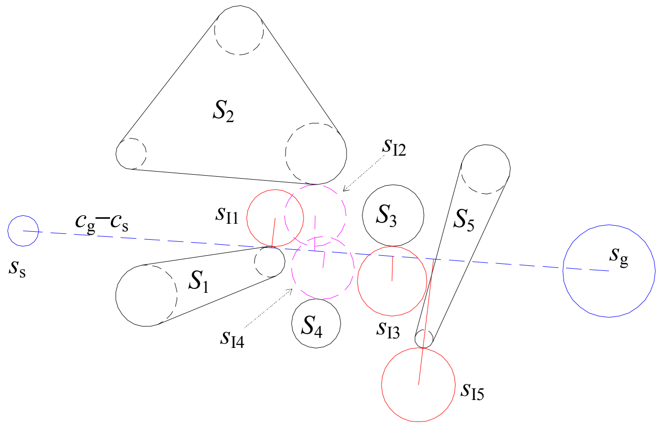


Fig. 3. Intermediate configurations of the mobile sphere in contact with the obstacles. $S_i$ with i=1,...,5, are obstacles modeled by s-topes and $s_{Ii}$ are their associated intermediate positions. Intermediate positions associated with an obstacle without collision have been depicted by a dashed line. Those ones associated with a collision situation have been represented by a solid line. Bi-sphere axis $c_g - c_s$ states the motion of the mobile-sphere center.

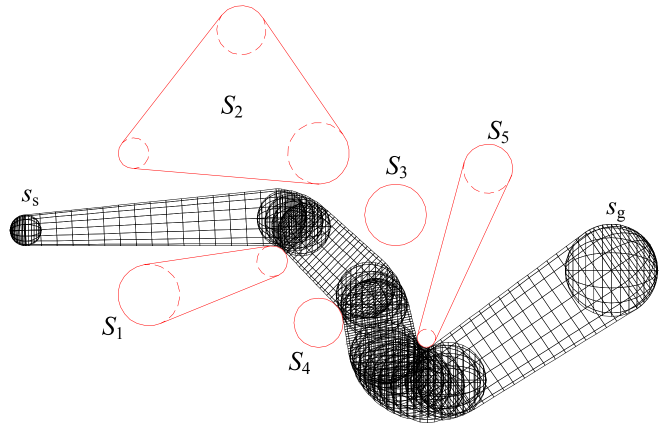


Fig. 4. Generation of a collision-free path. Path is the volume swept by the size-changing sphere from its start to its goal position. Every selected intermediate position has been depicted. Objects are represented in 2D.

if $S_{sg}$ is translated $\mathrm{MTD} \cdot \hat{v}_{\mathrm{MTD}}$, $S_{sg}$ is in contact with obstacle $S$ and $s_x$ is then the sphere in $S_{sg}$ which is contact with $S$.

In accordance with the situation depicted in fig. 2, after computing the corresponding distance, a collision is predicted (MTD is negative). Note that $s_I$ represents an intermediate position that avoids such a collision. This position $s_I$ is locally optimal. Nevertheless, the distance-computation algorithm can end computing more than one translational distance. All these distances are rejected except two. In fig. 2, the minimum one has been the selected. The other one characterizes an alternative way of avoiding the obstacle that can be used for a global motion planner [2].

From a given intermediate position, two new motions are obtained. One motion is represented by the bi-sphere defined by the start and the intermediate spheres. The second one is characterized by the bi-sphere defined by the intermediate and the goal spheres.

Now, the MTD between each motion and the obstacle is computed again. Note that a collision-free path is obtained by repeating recursively this process.

A collision-free path for a mobile sphere, whose radius dynamically changes, is obtained by computing the minimum translational distance between the mobile-sphere motion and each obstacle. After that, a set of parameters ($\lambda_i$, $\mathrm{MTD}_i, \hat{v}_{\mathrm{MTDi}}$) with i=1,…,$n$ and $n$ being the number of obstacles in sight, is computed. As a consequence, a set of intermediate configurations are determined by applying (9). A situation is presented in fig. 3.

Although, $\lambda_i$, $\mathrm{MTD}_i$, $\hat{v}_{\mathrm{MTDi}}$ have not been explicitly pointed out in fig. 3, it is easily to deduce the following properties:

a) Obstacles can be sorted in accordance with their parameter $\lambda_i \in [0,1]$. See (9).

b) Bi-sphere axis $c_g - c_s$, which represents the motion of the mobile-sphere center, divides the motion plane into two half planes. Vector $\mathrm{MTD}_i \cdot \hat{v}_{\mathrm{MTDi}}$ points to one half plane.

c) Obstacles are trivially classified into two types. A type 1 obstacle is completely located in a half plane. A type 2 obstacle is divided by the motion axis.

A collision-free path from the start to the goal position is obtained by considering a set of new submotions. Attending to fig. 3, these new submotions can be respectively defined by the bi-spheres $\{s_s,s_{I1}\}$, $\{s_{I1},s_{I5}\}$ and $\{s_{I5},s_g\}$. These submotions are defined from a subset of the intermediate configurations associated with the obstacles collided by the mobile-sphere motion [3].

For each new submotion, minimum translational distances are computed again, but only closer obstacles to this submotion are considered. Note that these obstacles are easily selected by its current $\lambda_i$. Each one of these new distances is computed by providing to the GJK* algorithm as initial set of spherical vertices, the same returned by the last distance computation in which such a obstacle was involved. This fact makes that the distance-computation algorithm complexity tends to 1.

Repeating recursively this process, a collision-free path is found. From situation in fig. 3 and with the above-mentioned submotions, the collision-free path shown in fig. 4 is obtained. This collision-free path has been computed in 139 µsec. on a Pentium® 4 CPU 3.00 GHz. 20 intermediate positions have been totally considered. Note that the path in fig. 4 represents a set of homotopic paths for a mobile sphere whose size always fits inside such a volume.

This path-planner technique always finds a collision-free path from the start to the goal configuration except when a generated intermediate position collides with any other obstacle. This situation is given when the mobile sphere is intended to pass through a narrow region of the free space.

## IV. Path Planning in Narrow Regions

When a selected intermediate configuration collides with another obstacle, a narrow area of the workspace is presented. Two options can be then adopted. First, both obstacles (s-topes) are joined by defining a new obstacle. For instance, obstacles $S_3$ and $S_5$ in fig. 4 degenerate into a tri-

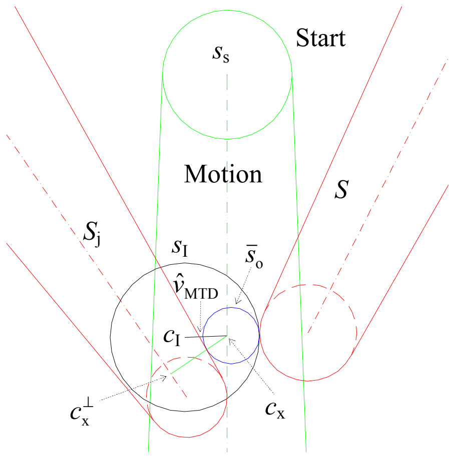


Fig. 5. Reduction of an intermediate position (sphere) $s_I=(c_I,r_I)$ which collides with obstacle $S_j$. $\bar{s}_o$ is the reduced sphere from $s_I$ which does not collide with $S_j$. For clarity, every object is depicted in 2D.

sphere when they are joined. Then, distance between the current motion and the new obstacle is computed and, consequently, a new intermediate position is determined [2]. This process is repeated until a collision-free intermediate position is found. After that, the joined obstacle is substituted by the original ones. The second option consists of reducing the size (radius) of the intermediate-position until the new intermediate does not collide with the two involved obstacles. See fig. 5. This option is introduced in this paper.

Let $S$ be an obstacle that collides with a given motion. This obstacle is then characterized by the parameters $\lambda_i \in [0,1]$, MTD<0 and $\hat{v}_{MTD}$. Its associated intermediate configuration $s_I=(c_I,r_I)$ has been selected to define a new submotion. Such an intermediate position collides with another obstacle $S_j$. See fig. 5.

From $s_I$, a set of infinite spheres is defined by means of a parameter $\mu \in \Re$. Spheres $\bar{s}(\mu)=(\bar{c}(\mu),\bar{r}(\mu))$ do not collide with obstacle $S$. Centers $\bar{c}(\mu)$ of spheres $\bar{s}(\mu)$ are defined as

$$\bar{c}(\mu)=c_x-\mu\cdot\hat{v}_{MTD} \tag{10}$$

$c_x$ is described in (9). Note that centers $\bar{c}(\mu)$ are constrained to one degree of freedom which is hold by the vector $\hat{v}_{MTD}$. As sphere $\bar{s}(\mu)$ does not collide with $S$, if $\mu$=MTD−$\delta$, then radius verifies $\bar{r}(\mu)=r_I$. $\delta\geq 0$ is the safety threshold cited on (9). On the other hand, if $\bar{r}(\mu)=0$, then it is trivial that $\mu=r_I+\mathrm{MTD}-\delta$ is verified. Consequently, $\bar{r}(\mu)$ is defined

$$\bar{r}(\mu)=r_I+\mathrm{MTD}-\delta-\mu \quad ; \quad \mu\in(\mathrm{MTD}-\delta, r_I+\mathrm{MTD}-\delta] \tag{11}$$

Equation (11) limits $\mu$. In this way, as $\bar{r}(\mu)\geq 0$, $\mu$ verifies $\mu\leq r_I+\mathrm{MTD}-\delta$. On the other hand, as $s_I$ has to be reduced to avoid a collision with $S_j$, $\bar{r}(\mu)<r_I$, and then, $\mu>\mathrm{MTD}-\delta$.

Problem is now stated as finding the value of $\mu$ which defines the sphere $\bar{s}_o$ in $\bar{s}(\mu)$ whose separation distance from $S_j$ is $\delta$. If $S_j$ is a simple sphere, problem is then trivial.

Let $S_j$ be a valid s-tope. Let $s_c^j$ be the closest spherical vertex, defining $S_j$, to $s_I$. Let $\{s_c^{j-1}, s_c^j\}$ and $\{s_c^j, s_c^{j+1}\}$ be the bi-spherical facets of $S_j$ which are the closest to $s_I$. For clarity, such bi-spherical facets are generally referred to as $\{s_{j0},s_{j1}\}$ with $s_{ji}=(c_{ji},r_{j1})$ with i=0,1.

Distance between $\bar{s}(\mu)$ and bi-sphere $\{s_{j0},s_{j1}\}$ is computed such as it has been previously indicated. Minkowski difference s-tope between $\bar{s}(\mu)$ and $\{s_{j0},s_{j1}\}$ is a bi-sphere defined by $\{M_0,M_1\}$

$$\begin{aligned} M_0&=(\bar{c}(\mu)-c_{j0},\bar{r}(\mu)+r_{j0}) \\ M_1&=(\bar{c}(\mu)-c_{j1},\bar{r}(\mu)+r_{j1}) \end{aligned} ;\ \mu\in(\mathrm{MTD}-\delta, r_I+\mathrm{MTD}-\delta] \tag{12}$$

$M_0$, $M_1$ centers and radii verify the following properties

$$\begin{aligned} (\bar{c}(\mu)-c_{j1})-(\bar{c}(\mu)-c_{j0})&=c_{j0}-c_{j1} \\ (\bar{r}(\mu)+r_{j1})-(\bar{r}(\mu)+r_{j0})&=r_{j1}-r_{j0} \end{aligned} \tag{13}$$

Consequently, although bi-sphere $\{M_0,M_1\}$ depends on an unknown parameter $\mu$, its angle of convergence is computed

$$\bar{\alpha}=\sin^{-1}\left((r_{j1}-r_{j0})/\|c_{j0}-c_{j1}\|\right) \tag{14}$$

Next step in computing such a translational distance requires the corresponding parameter $\lambda$, called now $\bar{\lambda}$.

$$\bar{\lambda}=-\frac{(c_{j0}-c_{j1})\cdot(\bar{c}(\mu)-c_{j0})}{\|c_{j0}-c_{j1}\|^2} \tag{15}$$

In accordance with (10), (15) is now rewritten

$$\bar{\lambda}=-\frac{(c_{j0}-c_{j1})\cdot(c_x-\mu\cdot\hat{v}_{MTD}-c_{j0})}{\|c_{j0}-c_{j1}\|^2}=\lambda_x+\lambda_\mu\mu \tag{16}$$

where $\lambda_x$ and $\lambda_\mu$ are finally computed as

$$\lambda_x=-\frac{(c_{j0}-c_{j1})\cdot(c_x-c_{j0})}{\|c_{j0}-c_{j1}\|^2};\quad \lambda_\mu=\frac{(c_{j0}-c_{j1})\cdot\hat{v}_{MTD}}{\|c_{j0}-c_{j1}\|^2} \tag{17}$$

Note that $\lambda_x$ characterizes $c_x^{\perp}$ i.e., the projection of $c_x$ onto axis $c_{j0}-c_{j1}$, with $c_x^{\perp}=c_{j1}+\lambda_x(c_{j0}-c_{j1})$. See fig. 5. If angle of convergence verifies $\bar{\alpha}\neq 0$, $\lambda'_x$ is determined by

$$\lambda'_x=\lambda_x+\frac{\|c_x^{\perp}-c_x\|\cdot\tan\bar{\alpha}}{\|c_{j0}-c_{j1}\|} \tag{18}$$

The minimum translational distance $\overline{\mathrm{MTD}}$ between $\bar{s}(\mu)$ and bi-sphere $\{s_{j0},s_{j1}\}$ is the distance between origin point and the Minkowski difference bi-sphere $\{M_0,M_1\}$ [2].

$$\overline{\mathrm{MTD}}=\|(\bar{c}(\mu)-c_{j0})+\bar{\lambda}(c_{j0}-c_{j1})\|-\left(\bar{r}(\mu)+\bar{\lambda}(r_{j1}-r_{j0})\right) \tag{19}$$

Forcing $\overline{\mathrm{MTD}}$ to be equal to $\delta\geq 0$, a second-degree equation is generated with $\mu$ as the unknown parameter. But previ

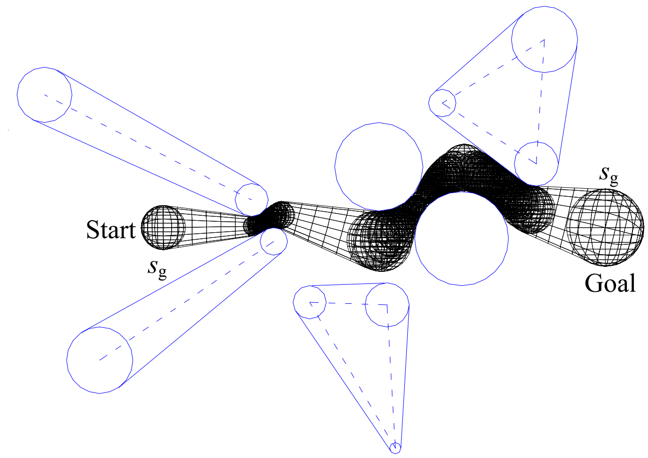


Fig. 6. Path generation for a size-changing sphere in narrow regions. Objects are modeled in 2D. Path represents the volume swept by such a sphere.

ously, the following aspect is considered. If $\lambda_x \le 0$, or $\lambda'_x \le 0$, if computed, then $\bar{\lambda}$ is set to be $\bar{\lambda}=0$. The reason is because the $s_I$ size reduction to avoid a collision with bi-sphere $\{s_{j0}, s_{j1}\}$ only requires considering $s_{j0}$. If $\lambda_x \ge 1$, or $\lambda'_x \ge 1$, if computed, then $\bar{\lambda}$ is set to be $\bar{\lambda}=1$. Analogously, in this case, only spherical vertex $s_{j1}$ has to be taken into account.

After solving such an equation, two solutions are returned, one of these solutions is always rejected because is strictly greater than $r_I$+MTD−δ. The other one, called $\bar{\mu}$ has to be updated to $\bar{\mu}'$, if $\bar{\alpha} \neq 0$, by means of the process pointed out by (7). In this way, $\bar{\lambda}$ is modified and it is now called $\bar{\lambda}'$

$$\bar{\lambda}' = (\lambda_x + \lambda_x \bar{\mu}) + \frac{\bar{d}_O \cdot \tan\bar{\alpha}}{\| c_{j0} - c_{j0} \|} \qquad \text{with } \bar{d}_O = \left(\bar{r}(\bar{\mu}) + \bar{\lambda}(r_{j1} - r_{j0})\right) + \delta \tag{20}$$

$\bar{d}_O$ expression comes from (19) with the constraint $\overline{\mathrm{MTD}}=\delta$. The desired solution $\bar{\mu}'$ is finally obtained by solving the simple equation

$$\bar{d}'_O = \left(\bar{r}(\bar{\mu}') + \bar{\lambda}'(r_{j1} - r_{j0})\right) + \delta; \quad \text{with } \bar{d}'_O = \bar{d}_O / \cos\bar{\alpha} \tag{21}$$

If $\bar{\mu}'$ verifies $\bar{\mu}'$>MTD−δ, sphere $\bar{s}_o$ is obtained by substituting μ for $\bar{\mu}'$ in (10) and (11). See fig. 5. It is important to remark that if $\bar{\mu}'$<MTD−δ is verified, it implies that size reduction of $s_I$ is not required because $s_I$ does not collide with $S_j$.

This process is applied to both bi-spherical facets $\{s_c^{j-1}, s_c^{j}\}$, $\{s_c^{j}, s_c^{j+1}\}$. Smallest result (sphere) returned is selected to be the new intermediate position, instead of $s_I$, for the path planner.

Now, the selection process of obstacle $S_j$ is formally introduced. Let $s_I=(c_I,r_I)$ be a selected intermediate configuration in order to define a new submotion. Let $S_k$ be any other obstacle where corresponding parameters $(\lambda_k, \mathrm{MTD}_k, \hat{v}_{\mathrm{MTDk}})$ are related with the current motion. Let $s_{Ik} = (c_{Ik}, r_{Ik})$ be the intermediate configuration sphere associated with $S_k$. Let $s_{Xk} = (c_{Xk}, r_{Xk})$ be the sphere in the current motion whose translation is minimal or maximal to be in contact with $S_k$, i.e., to obtain $s_{Ik}$. Let *f* be the following function

$$f(S, S_k) = \begin{cases} \| c_I - c_{Ik} \| - r_I - r_{Ik} & ;\text{if } S_k \text{ is a type-1 obstacle} \\ \| c_I - c_{Xk} \| - r_I - r_{Xk} & ;\text{if } S_k \text{ is a type-2 obstacle} \end{cases} \tag{22}$$

Note that *f* is not applied to all the type-1 obstacles. It is only applied to those type-1 obstacles whose $c_{Ik}$ is in the same half plane that $c_I$. Finally, $S_j$, if exists, is the obstacle with the minimum *f*.

This planning algorithm has been implemented in C and extensively run on a Pentium® 4 CPU 3.00 GHz. An example is shown in fig. 6. Path in fig. 6 has been obtained in 428 μsec. 61 intermediate configurations have been considered and 42 of these configurations have been reduced. Note that path in fig. 6 would be different if others intermediate configurations should have been chosen. An important property of this path-planning technique is based on the fact that the determined path represents a set of homotopic paths for a general object which always keeps inside of such a free-space volume.

A direct consequence of the proposed method is concluded when the final obtained parameter $\bar{\mu}'$ verifies $\bar{\mu}'$<MTD−δ. In this case, no reduction of $s_I$ is required, because $s_I$ does not collide with the involved obstacle. Anyway, note that $\bar{\mu}'$<MTD−δ represents a sphere, strictly bigger than $s_I$, which is separated δ≥0 from obstacle *S*. In other words, $s_I$ would have been grown.

## V. Finding the Free Space between Two Obstacles

A narrow region among obstacles is an unavoidable problem in many motion-planning algorithms [1]. Nevertheless, when free space between two obstacles is wider than the mobile-object size, most of these path planners do not get worried about computing such a free space.

Now, after selecting an intermediate configuration $s_I$, it is reduced or grown, with the same computational effort, by using the same technique presented in the previous section. Size of $s_I$ is modified to be separated δ≥0 from obstacle $S_j$. $S_j$ is selected as it has been indicated in the previous section.

After that, $s_I$ is substituted for the modified one, and a new submotion is defined.

This planning algorithm has been implemented in C and extensively tested. An example of this path planner is shown in fig. 7. Path has been obtained by considering 30 intermediate configurations, 17 have been grown and 13 reduced. Such a path has been obtained in 240 μsec. Note that this path also represents a set of homotopic paths for a general object which is always inside of such a free-space volume.

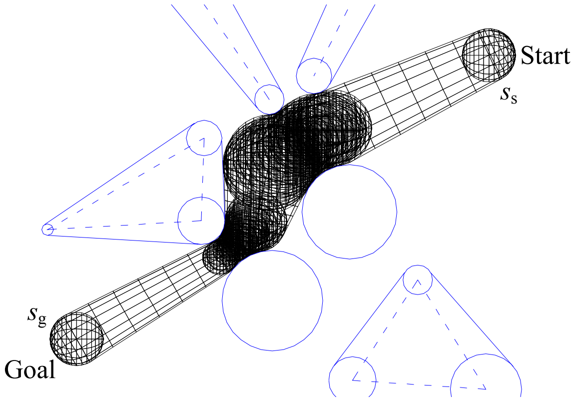


Fig. 7. Path enveloping free space. Path represents the volume swept by the mobile sphere. For clarity, objects have modeled in 2D.

## VI. Additional Properties

The path planning introduced in this paper is so fast that it can be run each time new information from the sensor system is received. Consequently, this path planner can work together with a robot navigator system.

The process for reducing or growing a sphere to put it in contact (or separated a given distance) with an obstacle acquires properties of the growth-distance function [7].

The free-space volume returned by this path planner can be used as a safety or collision-free area to constrain the solution of a motion planner for deformable linear objects like the one introduced by [8]. Reducing, in this way, the collision-detection computational cost which is generally required by this complex objects.

Sequence of intermediate configurations (spheres) between obstacles returned by the path planner characterizes an axis by joining the centers of such spheres. This axis can be used as a medial axis and, consequently, as an input for sample-based planning algorithms [9], [10].

In the same way, the above-mentioned axis states the maximum clearance roadmap between two obstacles.

A sensor system provides a set of points from the surface of each obstacle in sight. Each set of points is enveloped by a spherically-extended polytope [11]. As a consequence, this roadmap is conceptually connected to the hierarchical generalized Voronoi graph [12], where centers of the generated intermediate configuration are interpreted as meet points.

The path generated for the motion planner represents a volume that can be used as elastic strips [13]. Axis of such a volume would be the proposed as a candidate path. See, for instance, path in fig. 7.

## VII. Conclusion

A new technique for generating a collision-free path for a mobile sphere in environments with narrow regions has been presented in this paper. Mobile sphere has the ability of changing its size (radius) while it moves. Collisions between mobile sphere and obstacles are predicted by computing their minimum translational distance. It is assumed that obstacle positions are measurable.

In this way, when a collision is predicted, an intermediate configuration for the mobile sphere is determined in order to avoid such a collision. If this configuration (sphere) collides with a second obstacle (narrow region), a method for reducing its radius is then applied. This reduction is carried out by constraining sphere center to one degree freedom and separation from this obstacle is set equal to a configurable distance. If sphere and obstacle do not collide, this sphere is similarly grown by this method.

Any path generated characterizes a free-space region. Therefore, this path planner really generates a set of homotopic paths for any type of mobile object which always keeps inside of such a free space. Indeed, if collision detection is computationally prohibitive for a given mobile object, this volume can be used as a free workspace for finding a path for such a mobile object.

This motion-planning technique is so fast that can be run as frequent as new information from the world is received.

## Acknowledgment

Author would like thank to J. Minguez for discussions during the early stages of the work.

## References

[1] S. M. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.

[2] E. J. Bernabeu and J. Tornero, “Hough transform for distance computation and collision avoidance,” *IEEE Trans. on Robotics and Automation*, vol. 18, nº. 3, pp. 393-398, 2002.

[3] E. J. Bernabeu, and J. Tornero, "Real-Time generation of collision-free paths for a mobile sphere," in *Proc. IEEE Int. Conf. on Robotics and Automation*, San Francisco, 2000, pp. 2278-2283.

[4] E. G. Gilbert, D. W. Johnson, and S. S. Keerthi, "A fast procedure for computing the distance between complex objects in 3D space," *IEEE Journal of Robotics & Automation*, vol. 4, nº 2, pp. 193-203, 1988.

[5] J. Tornero, G. J. Hamlin, and R. B. Kelley, “Spherical-object representation and fast distance computation for robotics applications,” in *Proc. IEEE Int. Conf. on Robotics & Autom.*, 1991, pp. 1602-1608.

[6] G. J. Hamlin, R. B. Kelley, and J. Tornero, "Efficient distance calculation using spherically-extended polytope model," in *Proc. IEEE Int. Conf. on Robotics and Automation,* 1992, pp. 2502-2507.

[7] C. J. Ong and E. G. Gilbert, “Growth distances: new measures for Object Separation and Penetration,” *IEEE Trans. on Robotics and Automation*, vol. 12, nº. 6, pp. 885-903, 1996.

[8] M. Moll and L. E. Kavraki, “Path planning for deformable linear objects,” *IEEE Trans. on Robotics and Automation*, vol. 22, nº. 4, pp. 625-636, 2006.

[9] J.-M. Lien, S. L. Thomas and N. M. Amato, “A general framework for sampling on the medial axis of the free space,” in *Proc. IEEE Int. Conf. on Robotics and Automation*, Taipei, 2003, pp. 4439-4444.

[10] C. Holleman, and L. E. Kavraki, “A framework for using the workspace medial axis in PRM planners,” in *Proc. IEEE Int. Conf. on Robotics and Automation*, San Francisco, 2000, pp. 1408-1412.

[11] E. J. Bernabeu, and J. Tornero, "Optimal geometric modeler for robot motion planning,” Journal of Robotic Systems, vol. 17, nº 11, pp. 593-608, 2000.

[12] H. Choset, and J. Burdick, “Sensor-based exploration: The hierarchical Generalized Voronoi Graph,” The International Journal of Robotic Research, vol. 19, nº 2, pp. 96-125, 2000.

[13] O. Brock, and O. Khatib, “Real-time replanning in high-dimensional configuration spaces using sets of homotopics paths,” in *Proc. IEEE Int. Conf. on Robotics and Autom.*, San Francisco, 2000, pp 550-555.