Efficient Distance Computation Using Best Ellipsoid Fit

Technical Report
Stanford University, Department of Electrical Engineering

Elon Rimon

Robotics Laboratory
Stanford University, Department of Computer Science

Stephen P. Boyd

Information Systems Laboratory
Stanford University, Department of Electrical Engineering

February 10, 1992 (revised May 31, 1992)

Abstract

Knowledge of the distance between a robot and its surrounding environment is vital for any robotic system. The robot must obtain this information rapidly in order to plan and react in real-time. Our technique first surrounds the robot links and the obstacles by optimal ellipsoids, and then computes the clearance of the links from the obstacles with a generalized distance function. This approach offers an attractive alternative to the widely used technique of computing the distance via polyhedral representation of the robot and the obstacles. In particular, our approach offers a drastic reduction in the complexity of the data structures: each polyhedron, typically represented by a list of its features and their adjacency graph; is replaced by a minimum-volume ellipsoid represented by its center and a symmetric matrix whose dimension is either two or three (the workspace dimension). Moreover, while the computation time of the distance between polyhedra is often a function of their geometrical complexity, computation time in the ellipsoidal case is essentially *constant*; and becomes even more rapid when it is computed repeatedly along the robot's trajectory.

Our method consists of the following two algorithms: The first computes the optimal ellipsoid surrounding a convex polyhedron. The second is an analytic formula for the *free margin* about one ellipsoid with respect to another, that is computed as a standard eigenvalue problem. An efficient incremental version of the latter algorithm is then proposed. This system has been implemented and preliminary simulation results are provided throughout the paper.

1 Introduction

The technique of bounding sets with minimum-volume ellipsoids seems to be applicable in other areas, such as pattern recognition and machine vision. The main concern of this paper, however, is to demonstrate the effectiveness of the ellipsoid representation for geometrical reasoning in the context of robotics. Specifically, this paper is concerned with the robot collision-detection problem, that consists of computing a quantity that reflects, as a function of the geometrical data, the amount of clearance between the robot and its environment. Knowledge of this distance is of central importance for planning collision-free paths [9], and its rapid computation is essential in the low-level control, where the gradient vector-field of the distance is used to guard the robot from collision [13]. A similar need also arises in many computer graphics applications, especially in physical simulations [25].

Unfortunately, the distance depends on the various geometrical features of the robot and the obstacles; and this introduces a major computational bottleneck. In many practical applications, however, the robot links tend to be elongated objects that can be effectively surrounded by ellipsoids. Supposing that the obstacles are described by union of convex polyhedra — there are efficient algorithms to decompose a polyhedron into union of convex polyhedra — we surround each convex polyhedron by an ellipsoid as well. The problem of computing the distance between polyhedral links and obstacles is thus replaced by the problem of computing the distance between pairs of ellipsoids. We shall see that a function related to the distance, the *free margin* about an ellipsoid, can be rapidly computed in constant time, independent of the geometrical complexity of the original polyhedra.

More generally, this work is part of a larger program of research, whose purpose is to develop a geometric modeling system based on a catalogue of shapes expressed as Boolean combinations of linear and quadratic inequalities. Such a catalogue would be able to approximate all possible shapes and, unlike the purely polyhedral representation, seems to include shapes whose boundary is smooth (continuous normal). The catalogue must come with a set of operations that render it useful for robotic applications. To mention some of the most important ones: automatic construction of the shapes from sensory data, rapid collision detection, and the computation of a measure of the distance between pairs of shapes.

In the context of the latter problem, it can be easily verified that computing the Euclidean distance between two shapes described by intersection of linear and quadratic polynomials, such that each quadratic polynomial describes a convex region, is a *convex optimization problem* (see e.g., [3]). Being such, the Euclidean distance can be effectively computed up to desired accuracy in time linear in the number of geometrical features . In robotics, however, efficient iterative methods are not good enough. Most of the reactive controllers use the gradient vector-field (or subgradient when it is not differentiable) of the distance to guide the robot. It is therefore advantageous to obtain a closed-form expression for the gradient. Moreover, there is a basic need to explicitly parametrize the location of the configuration-space "obstacles" — the forbidden regions in the robot's configuration space — in terms of the geometrical data i.e., the inside-outside relation between shapes parametrized by the shapes' geometrical data. To the best of our knowledge the free margin function constitutes, for the first time in the context of robotics, an analytic formula for ellipsoids. The formula has the form of an eigenvalue problem, and we also give a closed-form formula for the gradient vector-field.

1.1 Organization of the Paper

This section continues with a brief account of the related literature. In Section 2 we describe an efficient algorithm for the minimal-volume ellipsoid surrounding a convex polyhedron. This also turns out to be a convex optimization problem, for which efficient ϵ-accurate algorithms that require time proportional to $\log(1/\epsilon)$ and linear in the number of vertices are known. We shall use the standard *ellipsoid algorithm*[1], whose features, as well as the class of convex optimization problems, are briefly described in Appendix A. The ellipsoid algorithm, although simple and efficient, is not the only known algorithm for solving convex optimization problems. In fact, interior-point algorithms recently developed by Nesterov and Nemirovsky [20] are much faster and may be a better choice for high-dimensional versions of this problem.

A closed-form formula for the free margin about one ellipsoid with respect to another is presented in Section 3. Specifically, let N be the dimension of the ambient space ($N = 2$ or 3 in our case), and let $\mathcal{E}(x_i, X)$ be the ellipsoid with center x_i and shape described by a positive-definite symmetric matrix X (a condition written as $X > 0$),

$$\mathcal{E}(x_i, X) = \{x \in I\!R^N : (x - x_i)^T X (x - x_i) \leq 1\}.$$

[1]The fact that the topic of this paper is ellipsoids and that convex optimization problems are solved by an algorithm based on n-ellipsoids is coincidental.

Let the two ellipsoids be $\mathcal{E}_1$ and $\mathcal{E}_2$. First, a formula for the point x^* in $\mathcal{E}_2$ at which the ellipsoidal level-surfaces surrounding $\mathcal{E}_1$ touch $\mathcal{E}_2$ for the first time is computed. Then the *free margin* about $\mathcal{E}_1$ with respect to $\mathcal{E}_2$ is computed in terms of x^* such that

$$\mathsf{margin}(\mathcal{E}_1, \mathcal{E}_2) \begin{cases} < 0 & \text{iff } \mathcal{E}_1 \text{ overlaps } \mathcal{E}_2; \\ = 0 & \text{iff } \mathcal{E}_1 \text{ touches } \mathcal{E}_2; \\ > 0 & \text{otherwise.} \end{cases}$$

Geometrically, $\mathsf{margin}(\mathcal{E}_1, \mathcal{E}_2)$ is the (signed) distance between $\mathcal{E}_1$ and $\mathcal{E}_2$ as determined by the metric associated with the matrix of $\mathcal{E}_1$. When $\mathcal{E}_1$ is moving along a trajectory it becomes a function of its current configuration — $\mathcal{E}_1$'s position and orientation. Under this interpretation the condition $\mathsf{margin}(\mathcal{E}_1, \mathcal{E}_2) \leq 0$ characterizes the configuration-space "obstacle" due to $\mathcal{E}_2$ — those configurations of $\mathcal{E}_1$ that involve intersection with $\mathcal{E}_2$.

The formula for x^* involves the minimal eigenvalue of a $2N \times 2N$ matrix whose entries are expressed in terms of the geometrical data. The standard *QR method* is then used to find the minimal eigenvalue. In Section 4 we describe how to accelerate the computation along the robot trajectory by exploiting the previous computation. As long as the trajectory points are sufficiently close by, the minimal eigenvalue is computed with the faster *inverse iteration method.* We will make precise the notion of "sufficiently close by", and will show in the process that the free-margin function is an *analytic function* of the geometrical data. This last result makes $\mathsf{margin}(\mathcal{E}_1, \mathcal{E}_2)$ very attractive in actual implementations. In fact, we shall present a closed-form expression for its gradient vector-field, which is readily computable in terms of the geometrical data.

1.2 Related Literature

The topic of surrounding complicated shapes by simpler ones is considered in the computational geometry literature. For example, [19] discusses the problem of surrounding a polyhedron by minimum-volume box. Although such a box is an attractive alternative for the minimum-volume ellipsoid, it is currently not clear which approach is more effective. In fact, both approaches require the same number of parameters to represent their shape, and they seem to complement each other as effective geometrical approximation. The selection of the most effective approximating shape is the topic of research now in progress.

The appeal of ellipsoids as effective means for shape representation has been recognized in the machine vision literature for quiet some time (see e.g., [21]). In the context of computing the ellipsoidal approximation, Post [22] has proposed an exact algorithm that computes the minimum-volume ellipsoid in time proportional to m^2 independent of the ambient dimension N, where m is the number of vertices. Our algorithm uses standard convex programming techniques and solves the problem within ϵ accuracy in time $mp(N)\log(1/\epsilon)$, where $p(N)$ is a polynomial function which is a constant in our case.

The topic of closed-form formulas for the forbidden regions in configuration-space is relatively unexplored. It seems that general algebraic decision-methods can compute such formulas for polynomially bounded shapes (using, for instance, the multivariate resultant [5]), but we are not aware of any practical implementation of them. Specific closed-form formulas are known for the following cases: a polyhedron moving in the presence of polyhedral obstacles [15, Chap 3]; a convex rigid-body moving with fixed orientation amidst convex obstacles [1]; and specific planar articulated chains [6]. This paper presents such a formula for the ellipsoidal case, where an ellipsoidal object is moving in the presence of ellipsoidal obstacles. Characterization of the intersection between general quadratic shapes is also discussed in the computer graphics literature (see e.g., [17]).

Computation of the distance between polyhedral shapes has long history in robotics (see e.g., [3, 9, 10, 18]), and the technique presented here complements these results for ellipsoidal shapes. In particular, an algorithm recently proposed by Lin and Canny for polyhedra [18], computes the distance between two moving convex polyhedra by tracking the closest two features. The computational effort of their algorithm is essentially constant, except at instances where the identity of the closest two features changes. At these singular events the new closest two features are found in time roughly linear in the number of geometrical features. In contrast, the computational effort of our ellipsoid approach is always constant and does not require the substantial bookkeeping required to manage the polyhedral features. Further, the free-margin function for ellipsoids is an *analytic function* of the geometrical data (refer to Corollary 4.1), while the polyhedral distance is not even differentiable. Of course, these gains come on the expense of using approximate shapes and a "distance" function that is not the Euclidean distance.

In relation to rapid distance computation, some researchers have suggested to trade computation with memory, by computing the distance beforehand for all possible configurations on a suitably discretized configuration space (see e.g., [16]). Note that this computation must use the aforementioned inside-outside functions to be effective. Unfortunately, the required memory grows exponentially with the dimension of the configuration space, and this approach becomes impractical for more than few degrees of freedom. For low-dimensional stationary environments, however, it offers very rapid (discretized) distance computation that can be made to be independent of the geometrical complexity of the environment.

2 Computation of the Optimal Ellipsoid

It was shown quite some time ago that for any compact set[2] with non-empty interior $\mathcal{P}$ there exists a unique ellipsoid $\mathcal{E}$ of minimal volume containing it [12]. This ellipsoid is called the Löwner-John ellipsoid of $\mathcal{P}$, or simply the **L-J ellipsoid**. Of course, the L-J ellipsoid contains the convex hall of $\mathcal{P}$ and for this reason we shall restrict our attention to convex sets $\mathcal{P}$. The L-J ellipsoid has a remarkable property that $\mathcal{P}$ contains the ellipsoid obtained from $\mathcal{E}$ by shrinking it from its center by a factor equal to the dimension of the ambient space. This establishes an upper bound on the distance of the surface of $\mathcal{E}$ from $\mathcal{P}$, and consequently indicates that the L-J ellipsoid is always an intuitively acceptable approximation.

We describe now an efficient ϵ-accuracy algorithm for what we shall call the **L-J problem** — to compute the L-J ellipsoid containing a given convex polyhedron $\mathcal{P}$. We do this in two

[2] i.e., closed and bounded.

Figure 1: The N-dimensional L-J ellipsoid is obtained by intersecting the $(N+1)$-dimensional L-J ellipsoid centered at the origin with the plane at height $x_{N+1} = 1$.

steps. First, following [20], the L-J problem is shown to be a *convex optimization* problem. Then the standard *ellipsoid algorithm* used to solve such problems is described in the context of our problem. Appendix A contains a short account of the convex optimization problems and the ellipsoid algorithm.

Let N be the dimension of the ambient space ($N = 2$ or 3 in our case), and denote its coordinates by $(x_1, \ldots x_N)$. Also, let $\mathcal{E}_0^{N+1}$ be an $(N+1)$-dimensional ellipsoid centered at the origin of $\mathbb{R}^{N+1}$. The idea is to embed $\mathcal{P}$ in a space of dimension $N+1$, in the plane at height $x_{N+1} = 1$, and then to compute the minimum-volume $\mathcal{E}_0^{N+1}$ containing the embedded $\mathcal{P}$. It turns out that the resulting $\mathcal{E}_0^{N+1}$ determines the N-dimensional L-J ellipsoid by simply intersecting $\mathcal{E}_0^{N+1}$ with the plane $x_{N+1} = 1$, as shown in Figure 1. This fact is mentioned in [20, pp 229], and we prove it in Appendix B for the reader's convenience.

The latter problem — of computing the minimum-volume $\mathcal{E}_0^{N+1}$ containing $\mathcal{P}$ — is a convex optimization problem. To see this, consider the volume of an $(N+1)$-ellipsoid, $\mathcal{E}(x, X)$, given by

$$\mathsf{volume}\,(\mathcal{E}) = \frac{\beta_{N+1}}{\sqrt{\det X}}, \tag{1}$$

where X is $\mathcal{E}$'s $(N+1) \times (N+1)$ symmetric positive-definite matrix, and β_{N+1} is the volume of the unit ball in $\mathbb{R}^{N+1}$, but we will not need β_{N+1}. Since (1) is equivalent to the equation

$$\log\left(\mathsf{volume}\,(\mathcal{E})\right) = \log \beta_{N+1} - \tfrac{1}{2}\log(\det X),$$

the L-J problem is equivalent to minimizing $-\log(\det X)$ subject to the constraints that X be symmetric positive-definite, and that $\mathcal{E}_0^{N+1}$ contain the polyhedron $\mathcal{P}$ embedded at height

$x_{N+1} = 1$. In general, an ellipsoid contains a convex polyhedron if and only if it contains its vertices, $v_1, ..v_m$. Thus the L-J problem becomes

$$\min\{-\log(\det X)\} \tag{2}$$

subject to

$$X > 0 \quad \text{and} \quad \begin{pmatrix} v_i^T & 1 \end{pmatrix} X \begin{pmatrix} v_i \\ 1 \end{pmatrix} \leq 1 \quad \text{for } i = 1, ..m. \tag{3}$$

Let the optimization variables be the distinct entries of the symmetric matrix X, and let M be the number of these entries, $M = \frac{1}{2}(N+1)(N+2)$. The problem (2)-(3) is convex if the objective function (2) and the constraints in (3) are *convex functions* in terms of the entries of X. Indeed, it is shown in Appendix B that $-\log(\det X)$ is convex in the region $X > 0$. The convexity of the constraints in (3) can be seen as follows. The constraint $X > 0$ can be written as

$$\lambda_{\max}(-X) < 0 \tag{4}$$

($\lambda_{\max}(-X)$ is the largest eigenvalue of the matrix $-X$). The function $\lambda_{\max}(-X)$ can be written as the maximum attained by a family of functions:

$$\lambda_{\max}(-X) = \max_{\|v\|=1} \{v^T[-X]v\},$$

where each $v^T[-X]v$ is linear in the entries of X and is therefore convex. It is known from convex analysis (see e.g., [24]) that the maximum of an arbitrary family of convex functions is itself convex. Hence $\lambda_{\max}(-X)$ is convex. Similarly, the constraint $(v_i^T, 1)X \begin{pmatrix} v_i \\ 1 \end{pmatrix} \leq 1$ is a linear inequality in the entries of X and is also convex. Having shown that (2)-(3) is a convex optimization problem, we can now apply the ellipsoid algorithm.

Given a convex polyhedron $\mathcal{P} \subset I\!\!R^N$ described by its vertices $v_1, \ldots v_m$, the ellipsoid algorithm computes a matrix X that minimizes (2) up to ϵ accuracy,

$$0 \leq (-\log(\det X)) - (-\log(\det X^*)) \leq \epsilon,$$

where X^* is the true minimum. It is shown in Appendix B that the N-ellipsoid obtained by intersecting the resulting ϵ-optimal $(N+1)$-ellipsoid with the plane $x_{N+1} = 1$ is also ϵ-optimal. Hence the original L-J problem is solved within a specified relative error of $e^{-\epsilon}$. Let us describe now the algorithm and some of its details.

First, we will need to compute subgradients of $\lambda_{\max}(-X)$. Let $\mathcal{E}(0, X)$ be an $(N+1)$-ellipsoid. Using the definition of subgradient given in Appendix A, the subgradient of $\lambda_{\max}(-X)$ is a matrix G satisfying the inequality

$$\lambda_{\max}(-Z) \geq \lambda_{\max}(-X) + \mathsf{tr}\left(G^T(Z-X)\right) \quad \text{for all symmetric matrices } Z$$

(tr denotes the trace). Let v be a unit-magnitude eigenvector of X corresponding to its maximal eigenvalue. It can be easily verified that the inequality

$$\lambda_{\max}(-Z) \geq v^T[-Z]v = v^T[-X]v - v^T(Z-X)v \quad \text{for all symmetric matrices } Z,$$

implies that the desired G is simply the symmetric matrix

$$G = -vv^T.$$

In the following, $(X_k, A_k) \in (I\!R^M, I\!R^{M\times M})$ is the center and matrix of the k^{th} ellipsoid in the ellipsoid algorithm and, for simplicity, X_k also represents the corresponding matrix of the $(N+1)$-ellipsoid $\mathcal{E}(0, X_k)$. For simplicity, as well, we replace $\lambda_{\max}(-X_k)$ by the equivalent expression $\lambda_{\min}(X_k)$. Last, we will need to use the stack notation: if $A \in I\!R^{n\times n}$, then A^s denotes the $n^2 \times 1$ vector obtained by stacking the columns of A over each other. More details about the algorithm can be found in Appendix A and in [4].

$X_1, A_1 \leftarrow$ an initial M-ellipsoid that contains the minimum;
$k \leftarrow 0$;
repeat {
$k \leftarrow k+1$;
compute $\lambda_{\min}(X_k)$ and $(v_i^T, 1)X_k \begin{pmatrix} v_i \\ 1 \end{pmatrix}$ for $i = 1, \ldots m$;
if $(\lambda_{\min}(X_k) \leq 0)$ { /* X_k is not positive definite */
compute eigenvector v for $\lambda_{\min}(X_k)$;
$h_k = (-vv^T)^s$; /* compute a subgradient */
$\tilde{g} \leftarrow h_k/\sqrt{h_k^T A_k h_k}$;
} else {
if ($(v_i^T, 1)X_k \begin{pmatrix} v_i \\ 1 \end{pmatrix} > 1$ for some $i = i_0$) { /* X_k is infeasible */
$h_k = \left(\begin{pmatrix} v_{i_0} \\ 1 \end{pmatrix}(v_{i_0}^T, 1)\right)^s$; /* compute a subgradient */
$\tilde{g} \leftarrow h_k/\sqrt{h_k^T A_k h_k}$;
}
} else { /* X_k is feasible */
$g_k = \nabla(-\log(\det X_k)) = \left(-X_k^{-1}\right)^s$; /* compute a gradient */
$\tilde{g} \leftarrow g_k/\sqrt{g_k^T A_k g_k}$;
}
$X_{k+1} \leftarrow X_k - \frac{1}{n+1}A_k\tilde{g}$;
$A_{k+1} \leftarrow \frac{n^2}{n^2-1}\left(A_k - \frac{2}{n+1}A_k\tilde{g}\tilde{g}^T A_k\right)$;
} until ($(\lambda_{\min}(X_k) > 0)$ and (no i_0 exists) and $(\sqrt{g_k^T A_k g_k} \leq \epsilon)$).

The initial X_1, A_1 can be conveniently chosen as follows. Let $0 < r < R$ be the radii of two balls in $I\!R^N$ such that the r-ball is contained in the polyhedron $\mathcal{P}$ and the R-ball contains it. Then the matrix X_1 can be initialized to be

$$X_1 = \mathsf{diag}(\frac{1}{1+R^2}).$$

This yields a feasible $(N+1)$-ellipsoid containing the embedded $\mathcal{P}$ and centered at the origin of $I\!R^{N+1}$. As for A_1, it can be easily visualized in the planar case (and rigorously proved in

Figure 2: The L-J ellipsoid of two convex polygons

general) that the cross-section of the optimal $(N+1)$-ellipsoid for various values of x_{N+1} in the interval $[-1, 1]$ must contain the r-ball. It follows that the eigenvalues of the matrices X_k are bounded from above by $\max\{1, 1/r^2\}$ and from below by $1/(1+R^2)$, so that the variation of the eigenvalues of X_k around their initial value should not exceed $\max\{1, 1/r^2\} - 1/(1+R^2)$. Since each X_k can be diagonalized by an appropriate rotation matrix, the variation of its entries should not exceed $(N+1)(\max\{1, 1/r^2\} - 1/(1+R^2)) \leq (N+1)\max\{1, 1/r^2\}$. Thus a good initial choice for the $M \times M$ matrix A_1 is

$$A_1 = \mathsf{diag}\left((N+1)^2 \mathsf{max}^2\{1, \frac{1}{r^2}\}\right).$$

The ellipsoid algorithm computes the L-J ellipsoid to a relative accuracy of $e^{-\epsilon}$ in $mp(M)\log(\frac{1}{\epsilon})$ steps, where $p(M)$ is a constant polynomial (see Appendix A). We have implemented a two-dimensional ($N = 2$ hence $M = 6$) version of this algorithm on a DEC5000 machine. A typical computation takes 600 iterations and runs for about 2 seconds[3]. Two numerical examples are shown in Figure 2.

[3] As we have already said, Nesterov and Nemirovsky's recent interior-point algorithms promise to be much faster [20].

3 The Free Margin Function

Given two ellipsoids in $\mathbb{R}^N$, $\mathcal{E}_1 = \mathcal{E}(x_1, P)$ and $\mathcal{E}_2 = \mathcal{E}(x_2, Q)$, we would like to find the point x^* in $\mathcal{E}_2$ such that

$$(x^* - x_1)^T P (x^* - x_1) \leq (x - x_1)^T P (x - x_1) \quad \text{for all } x \in \mathcal{E}_2. \tag{5}$$

Geometrically, x^* is the point in $\mathcal{E}_2$ that is the closest to $\mathcal{E}_1$ with respect to a distance function whose equidistance level-sets are the ellipsoidal surfaces surrounding $\mathcal{E}_1$. We consequently define the *free-margin* of $\mathcal{E}_1$ about $\mathcal{E}_2$ is defined to be,

$$\mathsf{margin}(\mathcal{E}_1, \mathcal{E}_2) \triangleq (x^* - x_1)^T P (x^* - x_1) - 1$$

($\triangleq$ denotes a definition).

Clearly, $\mathsf{margin}(\mathcal{E}_1, \mathcal{E}_2)$ is positive when $\mathcal{E}_1$ and $\mathcal{E}_2$ are disjoint, zero when they touch, and negative when their interiors overlap. Note, however, that $\mathsf{margin}(\mathcal{E}_1, \mathcal{E}_2)$ is not symmetric i.e., $\mathsf{margin}(\mathcal{E}_1, \mathcal{E}_2) \neq \mathsf{margin}(\mathcal{E}_2, \mathcal{E}_1)$, and that it resembles the actual Euclidean distance only when $\mathcal{E}_1$ and $\mathcal{E}_2$ are close to each other. Ideally, one would like to compute the Euclidean distance, but we are not aware of any closed-form formula for it[4]. In contrast, the computation of the free-margin function turns out to be equivalent to an eigenvalue problem, that can be solved by traditional methods.

First let us apply a coordinate transformation that will make $\mathcal{E}_2$ look like a unit ball,

$$\bar{x} \triangleq Q^{\frac{1}{2}}(x - x_2) \quad \text{or} \quad x = Q^{-\frac{1}{2}}\bar{x} + x_2. \tag{6}$$

In the new coordinates our problem becomes:

$$\min\{(\bar{x} - c)^T C (\bar{x} - c))\} \quad \text{such that} \quad \|\bar{x}\|^2 \leq 1, \tag{7}$$

where

$$C \triangleq Q^{-\frac{1}{2}} P Q^{-\frac{1}{2}} \quad \text{and} \quad c \triangleq Q^{\frac{1}{2}}(x_1 - x_2)$$

(C is positive definite). The ellipsoid $\mathcal{E}_2$ has thus become a unit ball. For our purposes we may assume that the center c is always outside the unit ball. This implies that the quadratic polynomial in (7) must attain its minimum on the boundary of the unit ball, where $\|\bar{x}\|^2 = 1$. For simplicity, we shall hereafter replace $\bar{x}$ by x.

Using Lagrange multiplier, a necessary condition for x^* to be a solution of (7) is

$$\lambda x^* = C(x^* - c) \quad \text{for some scalar } \lambda; \tag{8}$$

or, equivalently,

$$x^* = [C - \lambda I]^{-1} b,$$

---

[4] In the case of planar ellipses, for instance, this problem is equivalent to the minimization of a quadratic form over a two-dimensional torus embedded in four-dimensional Euclidean space $\mathbb{R}^4$. (This is, however, still a convex programming problem).

where

$$b \triangleq Cc,$$

and I is the $N \times N$ identity matrix. Substituting x^* into the constraint $\|x\|^2 = 1$,

$$b^T[C - \lambda I]^{-2}b = 1, \tag{9}$$

yields a $2N$-degree polynomial in λ.

It turns out that the minimal (real) root of the polynomial (9) solves the problem. This fact is evident from the following identity:

Theorem 1 ([7]) *If (x_1, λ_1) and (x_2, λ_2) are solutions of (8)-(9), then*

$$(x_2 - c)^T C(x_2 - c) - (x_1 - c)^T C(x_1 - c) = \frac{\lambda_2 - \lambda_1}{2}\|x_1 - x_2\|^2.$$

Thus the problem is solved if we can compute the minimal (real) root of the polynomial (9).

Using a method developed by Gander, Golub, and Matt [8], the problem is transformed into an eigenvalue problem via the introduction of two new variables, $y \in I\!R^N$ and $z \in I\!R^N$, as follows

$$y \triangleq [C - \lambda I]^{-2}b \quad \text{and} \quad z \triangleq [C - \lambda I]^{-1}b. \tag{10}$$

Expressing equations (9) and (10) in terms of (y, z) and λ, yields the following system of equations

$$\begin{aligned} b^T y &= 1 \\ [C - \lambda I]\, z &= b \\ [C - \lambda I]\, y - z &= 0. \end{aligned} \tag{11}$$

Substituting $b^T y = 1$ into the right side of the second equation yields,

$$[C - \lambda I]z = [bb^T]y.$$

Combining the third equation in (11) with the last equation yields,

$$\begin{bmatrix} C & -I \\ -bb^T & C \end{bmatrix} \begin{pmatrix} y \\ z \end{pmatrix} = \lambda \begin{pmatrix} y \\ z \end{pmatrix}, \tag{12}$$

a standard eigenvalue problem. Let λ^* be the minimal (real) eigenvalue of (12). It is shown in Appendix C that λ^* is exactly the minimal root of the $2N$-degree polynomial (9). It follows that the point x^*, and consequently $\mathsf{margin}(\mathcal{E}_1, \mathcal{E}_2)$, can be computed in terms of λ^* as follows:

Theorem 2 *Given two ellipsoids $\mathcal{E}_1 = \mathcal{E}(x_1, P)$ and $\mathcal{E}_2 = \mathcal{E}(x_2, Q)$, let λ^* be the the minimal eigenvalue of (12). Then the point $x^* \in \mathcal{E}_2$ is given by*

$$x^* = [C - \lambda^* I]^{-1}b \quad \text{where } C = Q^{-\frac{1}{2}}PQ^{-\frac{1}{2}} \text{ and } b = Q^{-\frac{1}{2}}P(x_1 - x_2), \tag{13}$$

and the free-margin function, $\mathsf{margin}(\mathcal{E}_1, \mathcal{E}_2) = (x^* - x_1)^T P(x^* - x_1) - 1$, satisfies

$$\mathsf{margin}(\mathcal{E}_1, \mathcal{E}_2) \begin{cases} < 0 & \text{iff } \mathcal{E}_1 \text{ overlaps } \mathcal{E}_2; \\ = 0 & \text{iff } \mathcal{E}_1 \text{ touches } \mathcal{E}_2; \\ > 0 & \text{otherwise.} \end{cases} \tag{14}$$

Figure 3: The closest point, x^*, according to the generalized distance determined by the ellipse surrounding the robot link

The correctness of (14) follows from the fact that the ellipsoid determined by the inequality

$$\begin{aligned} \mathcal{E} &= \{x : (x - x_1)^T P(x - x_1) - 1 \leq \mathsf{margin}(\mathcal{E}_1, \mathcal{E}_2)\} \\ &= \{x : (x - x_1)^T P(x - x_1) \leq (x^* - x_1)^T P(x^* - x_1)\}, \end{aligned}$$

is the smallest ellipsoid with center x_1 and matrix P that touches $\mathcal{E}_2$.

Two planar ($N = 2$) examples are shown in Figure 3. We have computed the minimal eigenvalue using the *QR algorithm*. This algorithm requires a preprocessing step that converts the matrix in (12) into Hessenberg form in about $(2N)^3$ steps [23, pp 386]. Then the QR-algorithm computes all the eigenvalues in roughly $3(2N)^3$ operations [23, pp 392]. The QR algorithm was used without exploiting the specifics of our matrix, and the average time for one distance computation was 2.5 `msec` (on a DEC5000 machine). In the next section we describe how to track only the minimal eigenvalue, and consequently achieve a considerable efficiency gain.

4 Incremental Computation of the Generalized Distance

We have seen that the computation of the generalized distance with the QR method is efficient. It computes, however, all the eigenvalues of the matrix

$$M \triangleq \begin{bmatrix} C & -I \\ -bb^T & C \end{bmatrix},$$

while only the *minimal* eigenvalue is needed.

In robotics, as well as in computer-graphics animation, the distance is typically computed along a trajectory i.e., the matrix M becomes $M(x(k))$, where $x(k)$ is the robot's k^{th} configuration. The computation time can be substantially reduced by tracking only the minimal eigenvalue along the trajectory. The *Inverse Iteration method* [23, pp 394] is suitable for this task. It is initialized with an estimate for the minimal eigenvalue, $\hat{\lambda}$, and for the corresponding eigenvector, $\hat{v}$; and works as follows:

$x(0) \leftarrow \hat{v};$
$k \leftarrow 0;$
$A \leftarrow [M - \hat{\lambda} I]^{-1};$
repeat {
$k \leftarrow k + 1;$
$x(k) \leftarrow Ax(k-1);$
normalize $x(k)$;
} until ($\|x(k) - x(k-1)\| \leq \epsilon$)
$\lambda^* = \hat{\lambda} + \|x(k-1)\|^2 / (x(k) \cdot x(k-1)).$

The idea behind this method is simple. Let λ^* and v^* be the minimal eigenvalue and the corresponding eigenvector of M. For any value of $\hat{\lambda}$, v^* is also an eigenvector of $M - \hat{\lambda} I$, with eigenvalue $\lambda^* - \hat{\lambda}$. Hence if $\hat{\lambda}$ is closer to λ^* than to the other eigenvalues of M, then the error $\|x(k) - x(k-1)\|$ converges exponentially to zero. The number of steps K required for the error to become less than ϵ satisfies

$$K \leq c \log(\frac{1}{\epsilon}),$$

where c is a constant that depends on the initial estimate $\hat{\lambda}$. Note that K grows slowly with the accuracy ϵ. The constant c depends on $\hat{\lambda}$ via the ratio $\left|\hat{\lambda} - \lambda^*\right| / \left|\hat{\lambda} - \lambda(M)\right|$ for $\lambda \neq \lambda^*$, and we shall hereafter make the rather gross simplification that c is approximately unity.

For this method to be of practical use, one must characterize the distance between λ^* and the other eigenvalues of M. The following theorem asserts that λ^* is the only eigenvalue in the left-hand side of the complex plane. We shall hereafter call λ^* the *minimal* eigenvalue of M. To the best of our knowledge this fact was previously unknown.

Theorem 3 *The minimal eigenvalue of M, λ^*, is* **negative real** *whenever the center c of the ellipsoid $\mathcal{E}(c, C)$ is outside the unit ball. Moreover, all the other eigenvalues of M satisfy*

$$\mathrm{Re}\{\lambda(M)\} \geq \lambda_1 > 0,$$

where λ_1 is the minimal eigenvalue of C ($C > 0$).

The theorem, whose proof is given Appendix C, asserts that λ^* is always isolated in the complex plane. This, in turn, affords a conclusion, stated in the following corollary, that λ^* is a real analytic function of the geometrical data. We will use in the corollary the following notation.

Let R be an $N \times N$ rotation matrix that diagonalizes C, and let Λ be the resulting diagonal matrix,

$$\Lambda = \mathsf{diag}(\lambda_1, \ldots \lambda_n) \triangleq R^T C R.$$

and let

$$\bar{b} \triangleq R^T b.$$

Corollary 4.1 *λ^*, and consequently* $\mathsf{margin}(\mathcal{E}_1, \mathcal{E}_2)$*, are* **real analytic functions** *of the geometrical data.*

Moreover, a formula for the gradient of λ^* is given by

$$\frac{\partial \lambda^*(\bar{b}, \Lambda)}{\partial \bar{b}_i} = -\frac{1}{\alpha} \frac{\bar{b}_i}{(\lambda_i - \lambda^*)^2} \quad \text{and} \quad \frac{\partial \lambda^*(\bar{b}, \Lambda)}{\partial \lambda_i} = \frac{1}{\alpha} \frac{\bar{b}_i^2}{(\lambda_i - \lambda^*)^3},$$

where

$$\alpha \triangleq \sum_{i=1}^{N} \frac{\bar{b}_i^2}{(\lambda_i - \lambda^*)^3}.$$

Proof: According to Theorem 3, λ^* is always an isolated root of the characteristic polynomial of M. It is well-known from function theory that an isolated root of a polynomial is an analytic function of its coefficients [14, pp 125].

The formula for the gradient is easily derived by implicit differentiation of the equation:

$$\sum_{i=1}^{N} \frac{\bar{b}_i^2}{(\lambda_i - \lambda^*(\bar{b}, \Lambda))^2} = 1,$$

which holds true according to Lemma C.1 in Appendix C.

□

Theorem 3 guarantees that there is a fixed-size disc of radius larger than λ_1 in the complex plane from which all initial guesses $\hat{\lambda}$ will converge to λ^*. In fact, only when $|\lambda^*| - \left|\hat{\lambda}\right| > \lambda_1$ correct convergence is not guaranteed. A practical criterion to detect correct convergence is the attainment of ϵ-convergence to some negative real number in less than K steps, where $K = \log(\frac{1}{\epsilon})$.

Let us count the number of operations required. The inverse-iteration method requires one matrix inversion — a $(2N)^3$-step operation, and then a repetitive multiplication by a vector — a $(2N)^2$-step operation. We have used in our implementation a closed-form formula for $[M - \hat{\lambda} I]^{-1}$ that takes only $5N^3$ steps to compute. Thus the total number of operations is about $5N^3 + (2N)^2 \log(\frac{1}{\epsilon})$. Comparing this with the QR method yields the ratio

$$\frac{\texttt{QR method}}{\texttt{inverse iteration}} = \frac{4(2N)^3}{5N^3 + (2N)^2 \log(\frac{1}{\epsilon})} = \frac{32}{5 + \frac{4}{N}\log(\frac{1}{\epsilon})}.$$

Figure 4: The closest point, marked by $*$, is traced as the ellipse surrounding the robot link moves around an obstacle (the ellipse surrounding the obstacle is not shown)

Substituting K for $\log(\frac{1}{\epsilon})$, it follows that the number of steps K required for an efficiency gain of two is

$$K = 3N \quad \text{where } N = 2, 3;$$

and the accuracy of the solution obtained after K such steps is

$$\epsilon = 10^{-3N} = \begin{cases} 10^{-6} & \text{if } N = 2 \\ 10^{-9} & \text{if } N = 3. \end{cases}$$

We have experimented with the incremental algorithm on a planar scene, in which an ellipsoidal link navigates in the presence of one stationary ellipsoidal obstacle. The link executes a biased random-walk, and at each step the free margin about the ellipsoidal link is computed with the incremental method. The random-walk was repeated for several randomly chosen polygonal link and obstacle pairs. A numerical example is shown in Figure 4, in which the rotational increment is one degree and the translational increment is one cell in a 128×128 grid. The average number of iterations required to attain $\epsilon = 1e - 06$ accuracy was 7, and the average time for one distance computation was 1 `msec` (on a DEC5000 machine)— shorter by a factor of about 2.5 than the QR-method discussed in the previous section. The incremental method ceased to converge correctly for rotational increment of about five degrees and translational increment of about five cells.

4.1 Conclusion

We have proposed in this paper a "complete" system: First a polyhedral robot and environment are approximated by ellipsoids. Then the free margin about each of the ellipsoidal links is computed in constant time per ellipsoidal obstacle. Hence, in terms of the ellipsoidal representation, the free margin about an n-link robot in an environment described by union of m convex polyhedra takes $O(n \cdot m)$ to compute. We have also shown how to accelerate the computation by exploiting the previous one along the robot's trajectory. Our ellipsoidal approach compares favorably with the polyhedral one, since there is no need to deal with the geometrical features of the underlying polyhedra the free-margin function takes essentially constant time to compute.

We have also presented in this paper an analytic formula for the free-margin function, and for its gradient vector-field. This, in turn, expands the catalogue of shapes for which a closed-form formula for the forbidden regions in configuration-space is known. This new formula also advances our larger program of research, concerned with setting up a system that will admit arbitrary Boolean combinations of linear and quadratic inequalities.

Last, it is worthwhile to note in the context of bounding shapes by ellipsoids that similar convex programming techniques can be used to efficiently compute the ellipsoid of maximal volume contained in a given convex polyhedron. This could be used to model the outer boundary of a robot work-cell. Moreover, by maintaining a pair of such ellipsoids, one surrounding the polyhedron and one contained in it, an estimate for the tightness of the ellipsoidal approximation can be effectively computed. The intersection between two polyhedra could then be first checked for the enclosing ellipsoids, for if they do not intersect than the underlying polyhedra are disjoint. Otherwise their interior ellipsoids are checked for intersection, and their intersection would imply that the underlying polyhedra intersect.

A Convex Programming

This appendix contains a short account of the class of convex optimization problems and the ellipsoid algorithm.

A real-valued function $\phi : \mathbb{R}^n \to \mathbb{R}$ is a **convex function** if

$$\phi(\alpha x_1 + (1-\alpha)x_2) \leq \alpha\phi(x_1) + (1-\alpha)\phi(x_2) \quad \text{for all } x_1, x_2 \in \mathbb{R}^n \text{ and } 0 \leq \alpha \leq 1.$$

Geometrically, ϕ is convex if and only if its *epigraph* — the set above its graph described by

$$\{(x,t) \in \mathbb{R}^{n+1} : t \geq \phi(x)\}$$

is a convex set. Moreover, convexity of ϕ implies that for any constant c the region in $\mathbb{R}^n$

$$\{x \in \mathbb{R}^n : \phi(x) \leq c\}$$

is a convex set in $\mathbb{R}^n$.

A **convex optimization** (or convex programming) problem is to compute x^* that minimizes $\phi(x)$, subject to the constraint that x be in $\mathcal{K}$, where ϕ is a convex function and $\mathcal{K} \subset \mathbb{R}^n$ is a convex region. One standard algorithm used to solve this problem is the **ellipsoid algorithm** developed in the 1970's by Shor,Yudin, and Nemirovsky [4, chap. 14]. It requires that $\mathcal{K}$ be described by a convex function $\psi : \mathbb{R}^n \to \mathbb{R}$ in the form

$$\mathcal{K} = \{x \in \mathbb{R}^n : \psi(x) \leq 0\}.$$

The algorithm produces a sequence of points $x_k \in \mathbb{R}^n$ that converge to x^*. It needs to compute at the k^{th} step a separating plane passing through x_k for one of the two convex regions

$$\{x : \phi(x) \leq \phi(x_k)\} \quad \text{or} \quad \{x : \psi(x) \leq \psi(x_k)\}.$$

The separating plane is not necessarily unique, since the boundary of the region may have a sharp corner at x_k. In such situations the separating-plane's normal becomes a subgradient. More precisely, if $\phi : \mathbb{R}^n \to \mathbb{R}$ is convex, but not necessarily differentiable, $g \in \mathbb{R}^n$ is a **subgradient** of ϕ at x if

$$\phi(z) \geq \phi(x) + g^T(z - x) \quad \text{for all } z \in \mathbb{R}^n.$$

The ellipsoid algorithm is initialized with an n-ellipsoid containing the minimizer x^* (e.g. a large n-ball containing $\mathcal{K}$). At the k^{th} step the current center of the ellipsoid, x_k, is compared against the constraint function ψ. If the constraint is violated ($\psi(x_k) > 0$), a separating plane passing through x_k for the region $\{x : \psi(x) \leq \psi(x_k)\}$ is computed. Otherwise, a separating plane passing through x_k for the region $\{x : \phi(x) \leq \phi(x_k)\}$ is computed. Clearly, one side of the resulting plane contains the entirety of $\mathcal{K}$ in the first case, and contains the minimizer x^* in the other. In both cases the $(k+1)^{th}$ ellipsoid is computed as the *minimum-volume* ellipsoid that contain the intersection of the k^{th} ellipsoid with the half space determined by the separating plane. A closed-form formula for such an ellipsoid is known:

$$\mathcal{E}_{k+1} = \{x : (x - x_{k+1})^T X_{k+1}^{-1} (x - x_{k+1}) \leq 1\},$$

where

$$x_{k+1} = x_k - \frac{1}{n+1} X_k \tilde{g} \quad \text{and} \quad X_{k+1} = \frac{n^2}{n^2 - 1}\left(X_k - \frac{2}{n+1} X_k \tilde{g} \tilde{g}^T X_k\right),$$

and $\tilde{g}$ is the normal to the separating plane. The algorithm stops when $|\phi(x_k) - \phi(x^*)| \leq \epsilon$ and $\psi(x_k) \leq 0$. An upper bound on $|\phi(x_k) - \phi(x^*)|$ is easily obtained from knowledge of a subgradient g_k to ϕ at x_k as follows,

$$\phi(x^*) \geq \phi(x_k) + g_k^T(x^* - x_k),$$

therefore

$$\phi(x_k) - \phi(x^*) \leq -g_k^T(x - x_k) \leq \max_{x \in \mathcal{E}_k} \{-g_k^T(x - x_k)\} = \sqrt{g_k^T X_k g_k},$$

using the formula for the extrema attained by a linear functional over an ellipsoid.

At each step the volume of the new ellipsoid is less than the volume of the previous one:

$$\mathsf{volume}\ (\mathcal{E}_{k+1}) \leq e^{-1/2n} \mathsf{volume}\ (\mathcal{E}_k),$$

by a factor that depends only on the dimension n of the ambient space. This affords in turn a conclusion that the number of steps K required to achieve ϵ-accuracy solution satisfies,

$$K \leq 2n^2 \log(\frac{c}{\epsilon}),$$

where c is constant. Note that this number grows slowly with both dimension n and accuracy ϵ ($n = 2$ or 3 and is constant in our case). Note, as well, that each step of the algorithm requires an evaluation of $\psi(x_k)$, an operation that is typically linear in the number of constraints used to describe $\mathcal{K}$. A more complete description can be found in [4, Chap. 14].

B Some Details Concerning the Optimal Ellipsoid

B.1 $-\log(\det X)$ is Convex In the Region $X > 0$

Let $X \in \mathbb{R}^{n\times n}$ be a symmetric matrix, and let M be the number of distinct entries in X ($M = \frac{1}{2}n(n+1)$). An alternative proof of the following theorem can be found in [2, proposition 11.8.9.5].

Theorem 4 *The function $-\log(\det X)$ is* **convex** *in the region of positive-definite matrices.*

Proof: A smooth function $f : \mathbb{R}^M \to \mathbb{R}$ is convex if and only if its Hessian matrix, $D^2 f$, is positive semidefinite [24]. In our case, let $V \in \mathbb{R}^M$ be a symmetric $n \times n$ matrix, and let $\alpha(t) : \mathbb{R} \to \mathbb{R}^M$ be a line passing through Z and having the direction V,

$$\alpha(t) = Z + tV.$$

Then we have to show that

$$\left.\frac{d^2}{dt^2}\right|_{t=0} (-\log\det(Z + tV)) \geq 0 \quad \text{for all } V.$$

To evaluate this derivative we will need the stack notation introduced in Section 2. Using this notation, the first derivative is

$$\frac{d}{dt}(-\log\det(Z+tV)) = -\frac{1}{\det(Z+tV)}(\cdots,[Z+tV]_{ij},\cdots)V^s,$$

where $(\cdots,[Z+tV]_{ij},\cdots)$ is a row vector and $[Z+tV]_{ij}$ is the ij^{th} cofactor of $Z+tV$. Using the identity $(A^s)^T B^s = \mathsf{tr}(A^T B)$ [11], the last equation can be written as

$$\frac{d}{dt}(-\log\det(Z+tV)) = -\frac{1}{\det(Z+tV)}\mathsf{tr}(([Z+tV]_{ij})^T V) = -\mathsf{tr}([Z+tV]^{-1}V),$$

where we have used the linearity of $\mathsf{tr}(\cdot)$, and the identity $([A_{ij}])^T = \det(A)A^{-1}$.

Taking the second derivative yields,

$$\frac{d^2}{dt^2}(-\log\det(Z+tV)) = \mathsf{tr}([Z+tV]^{-1}V[Z+tV]^{-1}V),$$

where we have used again the linearity of $\mathsf{tr}(\cdot)$, and the fact that $\dot{A}^{-1} = -A^{-1}\dot{A}A^{-1}$. Substituting $t=0$ in the last equation yields

$$\left.\frac{d^2}{dt^2}\right|_{t=0}(-\log\det(Z+tV)) = \mathsf{tr}((Z^{-1}V)^2) \geq 0,$$

since, using the identity $\mathsf{tr}(AB) = \mathsf{tr}(BA)$,

$$\mathsf{tr}((Z^{-1}V)^2) = \mathsf{tr}(Z^{-1}VZ^{-1}V) = \mathsf{tr}\left((Z^{-\frac{1}{2}}VZ^{-\frac{1}{2}})^T(Z^{-\frac{1}{2}}VZ^{-\frac{1}{2}})\right) \geq 0,$$

being the trace of a positive semi-definite matrix.

□

B.2 The Optimal n-Ellipsoid is Determined By an $(n+1)$-Ellipsoid

The following theorem asserts that the minimum-volume $(n+1)$-ellipsoid centered at the origin of $I\!R^{n+1}$ and containing $\mathcal{P}$ embedded at height $x_{n+1}=1$ determines the minimum-volume n-ellipsoid containing $\mathcal{P}$ by simply intersecting the $(n+1)$-ellipsoid with the plane $x_{n+1}=1$ [20, pp 229]. We will need the following lemma.

Let $\mathcal{E}_{n+1}(0,P)$ be an $(n+1)$-ellipsoid centered at the origin of $I\!R^{n+1}$. Its matrix P can be written as

$$P = \begin{bmatrix} P_{11} & P_{12} \\ P_{12}^T & P_{22} \end{bmatrix},$$

where P_{11} is $n \times n$, P_{12} is $n \times 1$, and P_{22} is a scalar. The following lemma gives a formula for the n-ellipsoid obtained by intersecting $\mathcal{E}_{n+1}(0,P)$ with the plane $x_{n+1}=1$.

Lemma B.1 *The n-ellipsoid obtained by intersecting $\mathcal{E}_{n+1}(0,P)$ with the plane $x_{n+1}=1$, denoted by $\mathcal{E}_n(y,Y)$, is given by*

$$y = -P_{11}^{-1}P_{12} \quad \textit{and} \quad Y = \frac{1}{1-\rho(P)}P_{11},$$

where $\rho(P)$ is a scalar in the interval $(0,1)$ defined by

$$\rho(P) = P_{22} - P_{12}^T P_{11}^{-1} P_{12}. \tag{15}$$

Thus, $\mathcal{E}_n(y,Y)$ is described by the inequality

$$\mathcal{E}_n(y,Y) = \{x \in I\!R^n : (x-y)^T Y (x-y) \leq 1\}.$$

Proof: All points $(x_1, \dots x_{n+1})$ inside $\mathcal{E}_{n+1}(0, P)$ at height $x_{n+1} = 1$ must satisfy,

$$\begin{pmatrix} x^T & 1 \end{pmatrix} \begin{bmatrix} P_{11} & P_{12} \\ P_{12}^T & P_{22} \end{bmatrix} \begin{pmatrix} x \\ 1 \end{pmatrix} \leq 1 \quad \text{where } x = (x_1, \dots x_n).$$

Expanding this inequality yields the inequality,

$$\begin{aligned} \begin{pmatrix} x^T & 1 \end{pmatrix} \begin{bmatrix} P_{11} & P_{12} \\ P_{12}^T & P_{22} \end{bmatrix} \begin{pmatrix} x \\ 1 \end{pmatrix} &= \|P_{11}^{1/2} x\|^2 + 2(P_{11}^{1/2} x)^T P_{11}^{-1/2} P_{12} + P_{22} \\ &= (x + P_{11}^{-1} P_{12})^T P_{11} (x + P_{11}^{-1} P_{12}) + P_{22} - P_{12}^T P_{11}^{-1} P_{12} \\ &\leq 1, \end{aligned}$$

or, equivalently,

$$(x + P_{11}^{-1} P_{12})^T \left[\frac{1}{1 - (P_{22} - P_{12}^T P_{11}^{-1} P_{12})} P_{11} \right] (x + P_{11}^{-1} P_{12}) \leq 1.$$

Let us verify that $0 < \rho(P) < 1$. We will need the fact that a symmetric matrix is positive definite if and only if all its principal minors are positive. In particular, this implies that P_{11}, all of whose principal minors being also principal minors of P, is positive definite.

Let us first verify that $\rho(P) > 0$. Using the identity

$$\det P = \det P_{11} \det \overbrace{(P_{22} - P_{12}^T P_{11}^{-1} P_{12})}^{\rho(P)}, \tag{16}$$

it must be that $\rho(P) > 0$, since P and P_{11} are positive definite. To see that $\rho(P) < 1$, note that all points $(x_1, \dots x_n, 1)$ inside $\mathcal{E}_{n+1}(0, P)$ satisfy

$$\begin{pmatrix} x^T & 1 \end{pmatrix} \begin{bmatrix} P_{11} & P_{12} \\ P_{12}^T & P_{22} \end{bmatrix} \begin{pmatrix} x \\ 1 \end{pmatrix} = \underbrace{(x + P_{11}^{-1} P_{12})^T P_{11} (x + P_{11}^{-1} P_{12})}_{*} + \rho(P) \leq 1.$$

But the term $(*)$ is positive since P_{11} is positive definite. Thus $1 - \rho(P) > 0$.

□

We are now ready to prove that if $\mathcal{E}_{n+1}(0, P^*)$ is the minimum-volume $(n+1)$-ellipsoid containing the embedded polyhedron $\mathcal{P}$, then the n-ellipsoid obtained by intersecting $\mathcal{E}_{n+1}(0, P^*)$ with the plane $x_{n+1} = 1$ is the minimum-volume n-ellipsoid containing $\mathcal{P}$. We shall also prove the following related fact. Let $\mathcal{E}(0, P)$ be an ϵ-optimal $(n+1)$-ellipsoid, that is,

$$0 \leq (-\log \det P) - (-\log \det P^*) \leq \epsilon,$$

or, equivalently,

$$1 \geq \frac{\det P}{\det P^*} \geq e^{-\epsilon}.$$

Then the n-ellipsoid obtained by intersecting $\mathcal{E}_{n+1}(0, P)$ with the plane $x_{n+1} = 1$ is also ϵ-optimal.

Theorem 5 *Let $\mathcal{E}_{n+1}(0, P^*)$ be the minimum-volume $(n+1)$-ellipsoid centered at the origin of $\mathbb{R}^{n+1}$ and containing the embedded $\mathcal{P}$. Then $\mathcal{E}_n(y^*, Y^*)$, obtained by intersecting $\mathcal{E}_{n+1}(0, P^*)$ with the plane $x_{n+1} = 1$, is the minimum-volume n-ellipsoid containing $\mathcal{P}$.*

Moreover, if $\mathcal{E}(0, P)$ is an ϵ-optimal $(n+1)$-ellipsoid containing the embedded $\mathcal{P}$, then $\mathcal{E}_n(y, Y)$, obtained by intersecting $\mathcal{E}_{n+1}(0, P)$ with the plane $x_{n+1} = 1$, is also ϵ-optimal.

Proof: According to Lemma B.1 each $(n+1)$-ellipsoid $\mathcal{E}(0, P)$ determines an n-ellipsoid $\mathcal{E}(y(P), Y(P))$ at height $x_{n+1} = 1$, as well as a real number $0 < \rho(P) < 1$. It turns out that every $\mathcal{E}(y, Y)$ and $0 < \rho < 1$ determine an $(n+1)$-ellipsoid $\mathcal{E}(0, P)$ that coincides with $\mathcal{E}(y, Y)$ in the plane at height $x_{n+1} = 1$. In fact, the resulting $P(y, Y, \rho)$ is given by the formula

$$P(y, Y, \rho) = \begin{bmatrix} (1-\rho)Y & -(1-\rho)Yy \\ -(1-\rho)(Yy)^T & \rho + (1-\rho)y^TYy \end{bmatrix}. \tag{17}$$

Let us compute the determinant of $P(y, Y, \rho)$,

$$\begin{aligned} \det P(y, Y, \rho) &= (1-\rho)^n \cdot \det Y \cdot \left\{\rho + (1-\rho)y^TYy - (1-\rho)y^TYy\right\} \\ &= \rho(1-\rho)^n \det Y, \end{aligned} \tag{18}$$

where we have used the identity $\det(\begin{bmatrix} A & B \\ C & D \end{bmatrix}) = \det(A)\det(D - CA^{-1}B)$. Since

$$\max_{\rho, Y}\{\rho(1-\rho)^n \det Y\} = \max_{\rho}\{\rho(1-\rho)^n\} \max_{Y}\{\det Y\},$$

it follows that $\det P(y, Y, \rho)$ attains its maximum exactly when $\det Y$ attains its maximum. Equivalently, the $(n+1)$-ellipsoid $\mathcal{E}(0, P)$ attains its minimal volume exactly when the volume of the corresponding n-ellipsoid $\mathcal{E}(y, Y)$ attains its minimum. In fact, $\rho(1-\rho)^n$ for $0 < \rho < 1$ attains its maximum at $\rho = \frac{1}{n+1}$ and therefore

$$\det P^* = \frac{1}{n+1}(1 + \frac{1}{n})^{-n} \det Y^*. \tag{19}$$

Let us show that ϵ-optimality of $\mathcal{E}(0, P)$ implies ϵ-optimality of $\mathcal{E}(y(P), Y(P))$. Given that

$$1 \geq \frac{\det P}{\det P^*} \geq e^{-\epsilon},$$

we have to show that

$$1 \geq \frac{\det Y}{\det Y^*} \geq e^{-\epsilon},$$

where $\mathcal{E}(y^*, Y^*)$ is determined by $\mathcal{E}(0, P^*)$. Using (18) and (19),

$$1 \geq \frac{\det Y}{\det Y^*} = \frac{\frac{1}{n+1}(1 + \frac{1}{n})^{-n}}{\rho(P)(1-\rho(P))^n} \frac{\det P}{\det P^*} \geq \frac{\frac{1}{n+1}(1 + \frac{1}{n})^{-n}}{\rho(P)(1-\rho(P))^n} e^{-\epsilon},$$

according to the theorem's hypothesis. But $0 < \rho(P) < 1$ according to Lemma B.1 , hence the denominator attains its maximum at $\rho(P) = \frac{1}{n+1}$, which implies in turn that

$$\frac{\frac{1}{n+1}(1 + \frac{1}{n})^{-n}}{\rho(P)(1-\rho(P))^n} \geq 1,$$

and the desired result is obtained.

□

C Some Details Concerning the Distance Computation

This section contains a proof that the minimal eigenvalue of the matrix

$$M = \begin{bmatrix} C & -I \\ -bb^T & C \end{bmatrix},$$

is isolated. We will need the following Lemma.

Lemma C.1 *Let $R \in \mathbb{R}^{N \times N}$ be the rotation matrix that diagonalizes the block C in M,*

$$\Lambda = \mathsf{diag}(\lambda_1, \ldots \lambda_n) = R^T C R.$$

Then the characteristic polynomial of M is

$$p(\lambda) = p_1(\lambda) p_2(\lambda),$$

where

$$p_1(\lambda) = \prod_{i=1}^{N} (\lambda_i - \lambda)^2 \quad \textit{and} \quad p_2(\lambda) = \sum_{i=1}^{N} \frac{\bar{b}_i^2}{(\lambda_i - \lambda)^2} - 1,$$

where $\bar{b} = R^T b$.

Proof: First note that

$$\begin{bmatrix} R^T & 0 \\ 0 & R^T \end{bmatrix} \begin{bmatrix} C - \lambda I & -I \\ -bb^T & C - \lambda I \end{bmatrix} \begin{bmatrix} R & 0 \\ 0 & R \end{bmatrix} = \begin{bmatrix} \Lambda - \lambda I & -I \\ -\bar{b}\bar{b}^T & \Lambda - \lambda I \end{bmatrix},$$

hence

$$\det(M - \lambda I) = (-1)^N \det \begin{bmatrix} -\bar{b}\bar{b}^T & \Lambda - \lambda I \\ \Lambda - \lambda I & -I \end{bmatrix}.$$

Using the identity $\det(\begin{bmatrix} A & B \\ C & D \end{bmatrix}) = \det(D)\det(A - BD^{-1}C)$,

$$\begin{aligned} \det(M - \lambda I) &= \det(-\bar{b}\bar{b}^T + [\Lambda - \lambda I]^2) \\ &= \det([\Lambda - \lambda I]^2)\det(-[\Lambda - \lambda I]^{-1}\bar{b}\bar{b}^T[\Lambda - \lambda I]^{-1} + I). \end{aligned}$$

But, in general, $\det(uv^T + I) = u \cdot v + 1$. Thus,

$$\det(M - \lambda I) = \det([\Lambda - \lambda I]^2)(\bar{b}^T[\Lambda - \lambda I]^{-2}\bar{b} - 1).$$

□

We are now ready to prove that λ^* is isolated. Recall that after the coordinate transformation of equation (6) is applied, the objective function to be minimized becomes the quadratic polynomial associated with the ellipsoid $\mathcal{E}(c, C)$, and the constraint becomes the unit ball.

Theorem 3 *The minimal eigenvalue of M, λ^*, is* **negative real** *whenever the center c of the ellipsoid $\mathcal{E}(c, C)$ is outside the unit ball. Moreover, all the other eigenvalues of M satisfy*

$$\mathsf{Re}\{\lambda(M)\} \geq \lambda_1 > 0,$$

where λ_1 is the minimal eigenvalue of C ($C > 0$).

Proof: First let us establish that $\det(M) < 0$. This would imply that M has at least one negative real eigenvalue. Using the identities $\det(\begin{bmatrix} A & B \\ C & D \end{bmatrix}) = \det(D)\det(A - BD^{-1}C)$ and $\det(uv^T + I) = u \cdot v + 1$, we have that

$$\begin{aligned} \det M &= (-1)^N \det \begin{bmatrix} -bb^T & C \\ C & -I \end{bmatrix} \\ &= \det(-bb^T + C^2) \\ &= \det{}^2(C)\det(1 - b^T C^{-2} b). \end{aligned}$$

Substituting $b = Cc$ obtains,

$$\det M = \mathsf{det}^2(C)\det(1 - \|c\|^2).$$

It follows that $\det M < 0$ as long as $\|c\|^2 > 1$ i.e., when the center c is outside the unit ball.

Next let us show that λ^* is isolated in the complex plane. The characteristic polynomial of M, given in Lemma C.1, is a function of a complex variable z,

$$p(z) = p_1(z)p_2(z) = \prod_{i=1}^{N}(\lambda_i - z)^2 \left(\sum_{i=1}^{N} \frac{\bar{b}_i^2}{(\lambda_i - z)^2} - 1 \right).$$

Consider the region of the complex plane defined by

$$\mathsf{Re}\{\mathsf{z}\} < \lambda_1$$

($\lambda_1 > 0$ is the minimal eigenvalue of C). Then, since $\lambda_i - \mathsf{Re}\{\mathsf{z}\} > 0$ for $i = 1, \ldots N$, the polynomial $p_1(z)$ cannot vanish, and only $p_2(z)$ may become zero in this region.

The i^{th} summand in $p_2(z)$ can be written as

$$\frac{\bar{b}_i^2}{(\lambda_i - z)^2} = \bar{b}_i^2 \frac{(\lambda_i - \bar{z})^2}{(\lambda_i^2 + |z|^2)^2} = \bar{b}_i^2 \frac{(\lambda_i - \mathsf{Re}\{\mathsf{z}\})^2 - \mathsf{Im}\{\mathsf{z}\}^2 + 2j(\lambda_i - \mathsf{Re}\{\mathsf{z}\})\mathsf{Im}\{\mathsf{z}\}}{(\lambda_i^2 + |z|^2)^2}.$$

It follows that the imaginary part of $p_2(z)$ is

$$\mathsf{Im}\{p_2(z)\} = 2\mathsf{Im}\{\mathsf{z}\} \sum_{i=1}^{N} \bar{b}_i^2 \frac{\lambda_i - \mathsf{Re}\{\mathsf{z}\}}{(\lambda_i^2 + |z|^2)^2}.$$

But $\lambda_i - \mathsf{Re}\{\mathsf{z}\} > 0$ for $i = 1, \ldots N$. So the roots of $p_2(z)$ must occur when $\mathsf{Im}\{\mathsf{z}\} = 0$.

The polynomial $p_2(\lambda)$ for λ real has exactly one root in the interval $(-\infty, \lambda_1)$. This observation is made in [8] and is a consequence of the following two facts. The first,

$$\lim_{\lambda \to -\infty} p_2(\lambda) = -1 \quad \text{and} \quad \lim_{\lambda \to \lambda_1^-} p_2(\lambda) = +\infty;$$

implies that $p_2(\lambda)$ has at least one root in $(-\infty, \lambda_1)$. The second,

$$\frac{d}{d\lambda} p_2(\lambda) = 2 \sum_{i=1}^{N} \frac{\bar{b}_i^2}{(\lambda_i - \lambda)^3} > 0$$

(since $\lambda_i - \lambda > 0$ for all $\lambda \in (-\infty, \lambda_1)$ and $i = 1, \ldots N$); implies that $p_2(\lambda)$ is strictly monotonic. This, together with the fact that $\det M < 0$ imply that $p_2(\lambda)$ has exactly one root, $\lambda^* < 0$, in $(-\infty, \lambda_1)$.

□

References

[1] C. Bajaj and M. Kim. Generation of configuration space obstacles: Moving algebraic surfaces. *The International Journal of Robotics Research*, 9(1):92–112, February 1990.

[2] M. Berger. *Geometry I,II.* Springer-Verlag, N.Y., 1980.

[3] J. E. Bobrow. A direct minimization approach for obtaining the distance between convex polyhedra. *The International Journal of Robotics Research*, 8(3):65–76, June 1989.

[4] S.P. Boyd and C.H. Barratt. *Linear Controller Design.* Prentice Hall, NJ, 1991.

[5] J. F. Canny. *The complexity of robot motion planning.* MIT Press, Cambridge, MA, 1988.

[6] J.R. Dooley and J.M. McCarthy. Parametrized descriptions of the joint space obstacles for a 2r closed chain robot. In *IEEE International Conference on Robotics and Automation*, pages 1536–1541, Cincinnati, OH, May 1990.

[7] W. Gander. Least squares with a quadratic constraint. *Numerische Mathematik*, 36:291–307, 1981.

[8] W. Gander, G.H. Golub, and U. Matt. A constrained eigenvalue problem. *Linear Algebra and its Applications*, pages 815–839, 1989.

[9] E. Gilbert and D. W. Johnson. Distance functions and their application to robot path planning in the presence of obstacles. *IEEE Journal of Robotics and Automation*, RA-1(1):21–30, March 1985.

[10] E.G. Gilbert and C-P Foo. Computing the distance between general convex objects in three-dimensional space. *IEEE Transactions on Robotics and Automation*, 6(1):53–61, Feb 1990.

[11] A. Graham. *Kronecker Products and Matrix Calculus: with Applications.* John Wiley and Sons, NY, 1981.

[12] F. John. Extremum problems with inequalities as subsidiary conditions (1948). In J. Moser, editor, *Fritz John Collected Papers*, chapter Vol 2, pages 543–560. Birkhauser, Boston, 1985.

[13] O. Khatib. Real time obstacle avoidance for manipulators and mobile robots. *The International Journal of Robotics Research*, 5(1):90–99, Spring 1986.

[14] K. Knopp. *Theory of Functions II.* Dover, New York, 1947.

[15] J.-C. Latombe. *Robot Motion Planning.* Kluwer Academic Publishers, Boston, 1990.

[16] J. Lengyel, M. Reichert, B.R. Donald, and D.P. Greenberg. Real-time robot motion planning using rasterizing computer graphics hardware. In *ACM SIGGRAPH*, pages 327–335, Las-Vegas, July 1990.

[17] J. Levin. A parametric algorithm for drawing pictures of solid objects composed of quadratic surfaces. *Communications of the ACM*, 19(10):555–563, Oct 1976.

[18] M. C. Lin and J.F. Canny. A fast algorithm for incremental distance calculation. In *IEEE International Conference on Robotics and Automation*, pages 1008–1014, Sacramento, CA, April 1991.

[19] R. R. Martin and P.C. Stephenson. Containment algorithm for objects in rectangular boxes. In *Theory and Practice of Geometric Modeling*, pages 307–325. Springer Verlag, New York, 1989.

[20] Y.E. Nesterov and A.S. Nemirovsky. *Interior Point Polynomial Methods in Convex Programming: Theory and Applications.* Springer Verlag, New York, 1992.

[21] A. P. Pentland. Perceptual organization and the representation of natural form. In *Readings in Computer Vision*, pages 680–699. M. Kaufmann Publishers, Los Altos, CA, 1987.

[22] M.J. Post. Minimum spanning ellipsoids. In *ACM Symposium on Theory of Computing*, pages 108–116, Washington, D.C., April 1984.

[23] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipes in C.* Cambridge University Press, NY, 1988.

[24] R. T. Rockafellar. *Convex Analysis.* Princeton University Press, Princeton, New Jersey, 1972.

[25] S. Sclaroff and A. Pentland. Generalized implicit functions for computer graphics. In *ACM SIGGRAPH*, pages 247–250, Las Vegas, July 1991.