# A Proximity Metric For Continuum Path Planning

Charles E. Buckley and Larry J. Leifer

Stanford University and Palo Alto VA Hospital

ABSTRACT

The problem of planning motions of robot manipulators and similar mechanical devices in the presence of obstacles is one of keen interest to the artificial intelligence community. Most of the algorithms previously reported for solving such problems have been *combinatorial algorithms,* which work by partitioning the problem domain continuum into a finite set of equivalence classes, and applying combinatorial search algorithms to plan transitions among them. However, the few *continuum algorithms that* have been reported, which do not rely on such a partitioning, have shown greater promise when applied to problems of complexity equivalent to that of planning a true manipulator motion. This is true even though the heuristics employed in these continuum algorithms have been extremely simple in nature. A significant barrier to the development of more refined heuristics for use in continuum algorithms is the uncertainty over how to characterise the proximal relationship between rigid bodies. In this paper, a new measurement function is reported which permits such characterisation. An introduction is made to a new type of path planning algorithm which this function makes possible, which promises to significantly increase the capabilities of continuum path planning software.

## 1 Prior Work

Two early research efforts in this area may be seen as cornerstones of the two basic methods employed.

Whitney introduced the first combinatorial algorithm for manipulator path planning [Whitney 1969]. Although this algorithm was only concerned with the planning of gripper motions in the plane, and only a few evenly spaced positions and orientations were considered, the algorithm suffered from problems of combinatorial explosion. The algoritlim was formally verified to be correct.

Peiper's algorithm was not combinatorial, and was applied to the planning of motions for a full six degree of freedom robot manipulator [Peiper 1968]. Although he reported qualitative success, he was unable to verify correctness.

This basic tradeoff between vcrifiability and the size of tractable problems remains a fundamental issue among those who study path-planning today.

Perhaps the most successful combinatorial results have been built on the work of [Lozano Perez 1981], who defined a scheme for exactly partitioning admissible points from inadmissible ones for polyhedral bodies in fixed relative orientation. .Jarvis has reported a similar scheme which works for arbitrary classes of objects, and is based on constructing adjacency graphs among uniform sized quanta in 3 space[1] [Jarvis 1984]. Brooks later extended Lozano Perez' work to apply to planar objects with variable orientation through the use of a successive approximation technique (Brooks 1982], but was subsequently unable to extend it to three dimensions. [Schwartz 1982] reported a generalization of this concept to apply to objects of arbitrary dimension bounded by algebraic surfaces (e. g. planes, cylinders, etc.). Although these results confirmed earlier results showing the path planning problem to be P-space complete [Reif 1970], they were of little practical interest since time complexity for discretization of the problem domain exceeded that of simple subdivision into hyperparallelopipeds of the smallest mesh size reprcsentable by standard computing hardware[2] [Buckley 1985].

Two non combinatorial algorithms stand out as significant. Loeff and Soni reported an algorithm for planning the motions of a planar, line segment manipulator among circular forbidden zones [Loeff 1975]. Khatib has implemented an algorithm in which simulated repulsive forces generated from a restricted set of object models were used to generate commanded positions for certain distinguished points on a manipulator [Khatib 1980].

Another simpler algorithms was reported by Myers [Myers 1981]. Although this algorithm was essentially one of hypothesis and test, its computation times for solving a general purpose path planning problem for a PUMA manipulator were on the same order of magnitude as those reported in [Brooks 1982] for the planar, free body case.

These results suggest that practical algorithms for path planning will be heuristic in nature, and not formally verifiable. However, prior research has only begun to explore the question of what sort of heuristics may be used to best advantage. Notably, excepting the algorithms of Loeff and Soni and that of Khatib, heuristics employed for collision *avoidance* in prior research have really been based only on whether or not a collision was *detected* in following an hypothesized trajectory. In most cases the direction in which liypothetical impact occurred was not even taken into account. Often the occurrence of a collision at one point along an hypothesized trajectory was considered grounds for rejection of that entire trajectory. Local perturbation of offending portions of trajectories has hardly been explored.

## 2 The Proximal Relationship of Objects in Space

To a large extent, the simplicity of the heuristics employed in previous algorithms is due the fact that there doesn't really exist

[1] This adjacency graph was searched using dynamic programming, which that computations be performed for *each* quantum in the spare.

[2] c. g. a single precision floating point number on something like a VAX

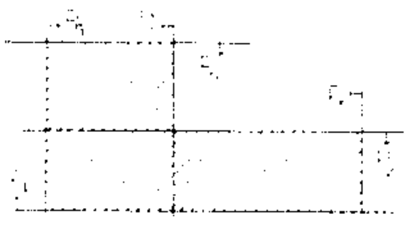

Figure 1: A Point Set Defined by Boundary Constraints

a good way of characterizing the proximal relationship between two rigid bodies or models. Conceptually, it is easy to determine whether or not an intersection between two such bodies occurs, given their relative position. However, if this information is insufficient, then what? It is often proposed that the actual point set corresponding to the intersection of rigid bodies be computed, but there have been no serious proposals made as to what might be done with all of this information if were it available.

Practically, the only alternatives to hypothesize and test methods which have enjoyed any success at all are the *relaxation methods.* These methods were originally conceived to solve constrained non linear optimization problems, and have been quite successfully applied. Loeff and Soni and Khatib took advantage of the fact that these algorithms work by continuously perturbing a state vector from some initial guess to a final optimum state to generate trajectories between known endpoints.

The main inconvenience with relaxation methods as they are almost always formulated is that all constraints must apply *simultaneously.* This has to do partially with the formulation of the Kuhn-Tucker stationarity equations for an extremum, which are used as error equations to drive state perturbations, and partially with long—standing conventions of the discipline. This requirement is fundamentally incompatible with standard solid modelling practice, in which solid objects are represented as point sets defined by *arbitrary* Boolean functions of boundary predicates.

For example, the set of points contained in the L-shaped planar region shown in Figure 1 may be described as those points which satisfy the following Boolean expression:

$$(P_1 \wedge P_4 \wedge P_5 \wedge P_6) \vee (P_5 \wedge P_2 \wedge P_3 \wedge P_4) \qquad \text{to}$$

where each Boolean expression $P_i$ corresponds to a point lying on the proper side of the associated line[8], as indicated by the direction of the normal arrows drawn in the figure. If $g_i$,(*) = 0 were the equation of boundary line P,, then the Boolean expression $P_i$ might be #,(x) < 0. Only the constraints in one or the other of the two disjuncts are necessary to qualify a point as being part of the shaded region. Further, for points lying in one of the arms of the "cell" it is *not possible* for all of the constraints to apply simultaneously. A similar argument may be made with respect to points outside of the shaded region. The developers of prior algorithms for path-planning in which relaxation methods were used were very aware of this problem, and in fact the limitations which they placed on their algorithms stemmed directly from it.

[8] an instance of a boundary manifold

In this paper is described a new metric for characterizing the proximal relationship between two convex rigid bodies which addresses this issue and others. This metric may be related to an algorithm intended exclusively for *detecting intersections* among convex bodies published in [Comba 1968].

The use of the Comba method actually involves the numerical solution of an unconstrained minimization problem itself, which can be time-consuming. The quantity to be minimized is a "pseudo-constraint function" defined as follows:

Let the $n_c$ boundary predicates (those which define *all* of the objects under consideration) be defined by

$$g_i(x) \leq 0, \; i = 1, n_c$$

where the $g_i$ are convex functions of their arguments[4]. Then, the Comba constraint function $G(x)$ is defined by the three equations:

$$v_i(x) = (g_i(x)^2 + t^2)^{\frac{1}{2}} + g_i(x) \qquad (2)$$

$$V = \sum_{i=1}^{n_c} v_i \qquad (3)$$

$$G = \underbrace{\frac{1}{2}\left(V - \frac{t^2}{V}\right)}_{a} + c \qquad (4)$$

where $t, c$ are small, positive constants.

For $t$ non-zero, each of the $v_i$ functions is always positive. However, when the corresponding $g_i$ is negative, then that $v_i$ tends toward zero. In particular, when a $g_i = 0$, then $v_i = t$. When $V = t$, then the term $a$ in equation 4 is zero. Therefore, when *all* of the $g_i$ are zero, then $V = O(t)$. When a $g_i$ is negative, then its corresponding $v_i = O(t^2)$. When $t$ is small, $t^2 < t$, and the contribution of a $v_i$ corresponding to a negative $g_i$ will be less. Therefore, except in instances where the bodies in question are just touching, if

$$G^* = \min_x G \qquad (5)$$

then $G^* < 0$ corresponds to a condition of intersection, $G^* > 0$ corresponds to non intersection, and the $x$ corresponding to a $G^*$ near zero are in a gray area. The constant $c$ in the equation for $G$ helps minimize the problems associated with this gray area. For example, it may be used to create a safety buffer around the obstacles being tested.

The Comba function is meaningful only at its minimum, and then only when compared against the threshold of zero. Trying to assign other meaning to the value of the function taken at its minimum is hampered by the influence of the $t$ and $c$ parameters, which were introduced to insure continuous differentiability of the function as an aid to finding its minimum.

The Comba function is undefined at any point $x$ satisfying all constraints if $t = 0$. However, for the non-intersecting case, if both $t$ and $c$ are set equal to zero, then the Comba function becomes:

$$G = \min_x \frac{1}{2} \sum_{i=1}^{n_c} \max(0, g_i(x)) \qquad (6)$$

If the number of convex bodies being tested is restricted to 2, and there is only one $g_i$ function per body, corresponding to $g_i(x) \equiv$

[4] The $g_i$ must be continuously differentiable in order to employ numerical methods for minimization, such as Davidon–Fletcher–Powell[Avriel 1976].

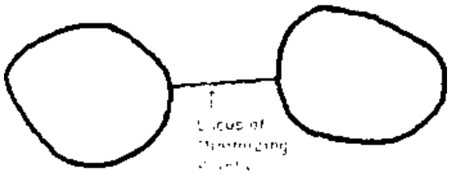

Figure 2: Locus of Minimizing Points for Comba Distance Function

$d(x, B_i)$, the convex distance function[5] [Rockafellar 1970], then the function $g$ will be minimized anywhere on the line segment connecting the two closest points of bodies $B_1$ and $B_2$. This segment is shown in Figure 2. More important, its value there will correspond to half the distance between these closest points, or in other words, half the minimum distance between bodies $B_1$ and $B_2$[6]

This specialized version of the Comba function has the advantage that the *generating segment* corresponding to the locus of minimizing points is a function of only a single point on each of the bodies. This makes it easier to generate a derivative of this specialized Comba function than it is for the general case. On the other hand, this specialization effectively reimposes the restrictions that:

1. Each of the convex bodies involved be represented by a single, differentiate manifold.

2. The bodies in question should not intersect.

These restrictions correspond almost exactly to those which Loeff and Soni, and subsequently Khatib found it necessary to impose in their algorithms. Loeff and Soni restricted their attentions to planar problems, in which the manipulator links were modelled as line segments, and the obstacles "[did] not have sharp corners or sides" (Loeff 1975], and in which the influence function used to repel a moving from a fixed body was a decreasing function of the minimum distance between them. Khatib restricted his attention to the interaction between obstacles whose surfaces could be modelled as single differentiate manifolds, and selected discrete points on a moving manipulator[7]. The minimum distance between these points and obstacles was subsequently used in an inverse square potential function to simulate forces between the two bodies generating it.

In both of these cases, a clear effort was made to avoid having the two bodies in question intersect. There is good reason for this — if the two bodies intersect, the minimum distance between them drops to 0 identically and abruptly, and remains that way for arbitrary intersections. This means that the gradient of the minimum distance function is identically zero, and can therefore provide no information which might be used to drive a relaxation algorithm.

An even more important limitation than that placed on the type of objects which can be modelled is the one which prevents the modified Comba algorithm from working if the objects between which constraints are computed intersect. Both the Loeff/Soni and Khatib algorithms were subject to this limitation, which is a severe one because it forces an evolutive algorithm for trajectory generation, in which one known state is sequentially perturbed towards another, as shown in Figure 3a, and no inadmissible states are ever entered. This approach will only work if the problem in question is free from false local minima. The limitations of this type of algorithm have been aptly demonstrated in mobile base path planning research, e. g. [Chatila 1981], [Cahn 1975].

Figure 3: Two Paradigms for Path Planning

An alternate paradigm for path planning is shown in Figure 3b, in which an entire trajectory is hypothesized between the two known endpoints, and *perturbed* into admissibility, if it is not already. Such an approach depends on being able to deal effectively with inadmissible states (in this case intersections between the moving body and its obstacles) should they arise.

## 3 A, the Minimum Directed Distance

The function A, called the minimum directed distance between two arbitrary convex bodies, was developed in order to address these two issues. That is, it was developed to:

- be valid for bodies bounded using more than one boundary predicate.
- be valid even when the bodies in question intersect.

A also stems from the distance form of the Comba intersection form described earlier, hence its restriction to convex bodies. This restriction does not constitute much of a problem, since convex decomposition can be accomplished "off-line*'. Algorithms exist for performing this decomposition automatically in the case of certain classes of objects, such as polyhedra [Chazelle 1980]. It can also be done by hand if necessary.

Although the merit of the A function stems in large part from the fact that it is valid between two arbitrary convex bodies, it is simplest to explain for the case in which one of the bodies is a point. Extension to the point body case follows from the same configuration space obstacle transformation described in [Lozano-Perez 1981].

### 3.1 Disjoint Case

Consider first the case in which the givens of the problem, a point $x_0$ and a set of points of a rigid, convex body $C$ are fixed in

corresponding to the minimum distance from a point $x$ to all points in body

The 1/2 factor is included so an to make the $a$ term correspond to the inverse of the transformation from g, to $V_i$. It's presence is not essential.

[7]which he called "points submitted to a potential" (Khatib 1980]

relative position so that $C \cap x_0 = \emptyset$. Let $C$ be defined symbolically by

$$C = \{\forall x \in R^n \ : \ g_i(x) \geq 0, i = 1, n_c\},$$

where $n_c$ is the number of constraints. The $g_i$ are concave functions, which means that $g_i(x) \geq 0$ is a convex set.

The minimum distance problem (DP) can be defined as the problem of finding:

$$d(x_0; C) = \min_x \sqrt{(x - x_0)^T (x - x_0)} \qquad (7)$$

subject to

$$g_i(x) \geq 0, \ i = 1, n_c.$$

The stationary conditions for this problem are:

$$\frac{(x - x_0)}{\sqrt{(x - x_0)^T (x - x_0)}} = \sum_{i=1}^{n_c} \lambda_i \nabla g_i(x) \qquad (8)$$

in which

$$\lambda_i \geq 0, \quad \lambda_i g_i(x) = 0.$$

This problem is a well-behaved, convex one.

Now, the following two-stage problem (SP) is introduced as an alternative to problem (DP) above.

$$s(x_0; C) = - \min_\xi \max_x \xi^T (x - x_0) \qquad (9)$$

subject to

$$g_i(x) \geq 0, \ i = 1, n_c$$

*and*

$$1 - \xi^T \xi \geq 0.$$

Considering the two extremizations as separate problems, the objective functions of each are linear, and may be considered as simultaneously convex and concave. The constraints are concave, so each extremization represents a convex problem.

There are two sets of stationary conditions, one corresponding to each of the stages:

$$\frac{\partial}{\partial x}: \qquad \xi = \sum_{i=1}^{n_c} \beta_i \nabla g_i(x), \ \beta_i \geq 0 \qquad (10)$$

$$\frac{\partial}{\partial \xi}: \qquad (x - x_0) = -\beta_0 \xi, \ \beta_0 \geq 0. \qquad (11)$$

It is easy to eliminate $\xi$, which yields

$$(x - x_0) = \beta_0 \sum_{i=1}^{n_c} \beta_i \nabla g_i(x). \qquad (12)$$

Comparing these stationary conditions with those of problem (DP), it is seen that they are equivalent if

$$\lambda_i = \frac{\beta_0 \beta_i}{\sqrt{(x - x_0)^T (x - x_0)}}, \quad i = 1, n_c.$$

From the definition of the Kuhn–Tucker stationary conditions, when a multiplier $\beta_i$ is *equal* to 0, then that constraint is inactive. If $\beta_0 = 0$, then the stationary conditions 11 dictate that $x^* = x_0$, where $x^*$ is an $x$ which satisfies the stationary conditions. However, by definition, there is no $x \in C$ which can equal (be coincident with) $x_0$, since $x_0$ and $C$ are disjoint. Therefore, the unit magnitude constraint on $\xi$ must be active in this case.

The activity of the unit magnitude constraint on $\xi$ in turn implies that

$$\beta_0 = \sqrt{(x - x_0)^T (x - x_0)},$$

and therefore

$$\lambda_i = \beta_i, \ i = 1, n_c.$$

Further, if $\xi^*, x^*$ are extrema which satisfy the stationary conditions, then

$$\xi^* = - \frac{(x^* - x_0)}{\sqrt{(x^* - x_0)^T (x^* - x_0)}}$$

and

$$\begin{aligned} s(x_0; C) &= -(\xi^*)^T (x^* - x_0) \\ &= \frac{(x^* - x_0)^T (x^* - x_0)}{\sqrt{(x^* - x_0)^T (x^* - x_0)}} \\ &= \sqrt{(x^* - x_0)^T (x^* - x_0)} \\ &= d(x_0; C). \end{aligned} \qquad (13)$$

The equivalence of the two problems under conditions of non-intersection has been fully established.

The solution of problem (SP) has a simple interpretation in terms of support functions, which is useful in developing intuition for the development which follows.

The *indicator function* of a convex set $C$ is defined by

$$i(x, C) = \begin{cases} 0 & x \in C; \\ +\infty & x \notin C. \end{cases}$$

Its convex conjugate function is given by [Avriel 1976]

$$\begin{aligned} i^*(\xi, C) &= \max_x \{\xi^T x - i(x; C)\} \\ &= \sup_{x \in C} \{\xi^T x\}. \end{aligned} \qquad (14)$$

The $x$–extremum portion of the problem (SP) is given by

$$\max_x \{\xi^T (x - x_0)\}$$

subject to

$$g_i(x) \geq 0, \ i = 1, n_c$$

or in other words

$$\begin{aligned} & \sup_{x \in C} \{\xi^T (x - x_0)\} \\ = & \sup_{x \in C} \{\xi^T x\} - \xi^T x_0 \\ = & i^*(\xi; C) - \xi^T x_0 \\ = & i^*(\xi; C \ominus \{x_0\}) \end{aligned} \qquad (15)$$

where the operator $\ominus$ denotes set subtraction. Set subtraction will be formally defined later on, but in this case the operation simply involves subtracting $x_0$ from each point in the convex set $C$. Its easy to see that $x_0 \notin C \Leftrightarrow 0 \notin C \ominus x_0$.

Expressed in this form, the $x$-extremization may be given a simple graphical interpretation, as shown by the (necessarily planar) example in Figure 4. A support function with an argument of $\xi$ (considered as a free vector) may be computed graphically by taking a hyperplane[a] with normal $\xi$ and moving it so that it

[a] a line in the planar case

Figure 4: A Graphical Interpretation of $x$-Extremization

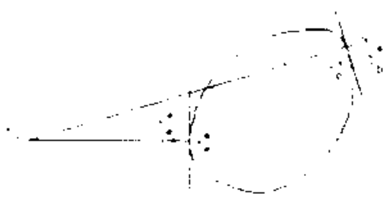

Figure 5: A Solution and a Non-Solution to Problem (SP')

osculates against the set in question, with the normal pointing away as shown. Then the value of the support function is equal to the dot product between a vector from the origin of the coordinate system to any point of the osculation. Of course, this value depends on the choice of the origin, but the transformation

$$C \rightarrow C \ominus x_0$$

turns the point $x_0$ into the origin, and makes this method appropriate for describing proximal relationships between $x_0$ and $C$.

The problem (SP') can be restated in light of the above as

$$s(x_0; C) = -\min_{\xi}\{i^*(\xi; C \ominus x_0)\}$$

subject to

$$|\xi| \leq 1.$$

For $x_0, C$ disjoint, solving $s$ simply involves finding the unit magnitude direction vector which minimizes the support function of the transformed set. This corresponds to the identification of a point such as $x_a^*$ in figure 5, and its corresponding $\xi$. It is expressly pointed out that this is *not* equivalent to solving

$$\max_{\xi}\{i^*(\xi; C \ominus \{x_0\})\}.$$

Such a solution would correspond to the point $x_b^*$ in the same figure. The unit vector $\xi$ points along the *generating segment* from $x_a^*$ to $x_0$.

### 3.2 Non–Disjoint Case

When $x_0 \in C$[9], the solution to the problem (DP) is uniformly zero, and therefore uninteresting.

[9]corresponding to intersection of two bodies in the original problem

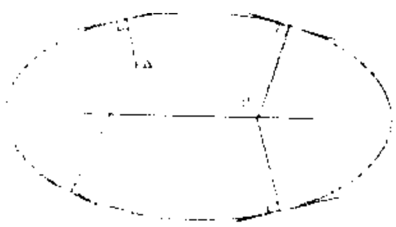

Figure 6: Multiple Stationary Conditions for $\Delta$

For problem (SP), the origin of the transformed problem will lie inside of $C \ominus \{x_0\}$. Therefore, $i^*(\xi, C \ominus \{x_0\}) \geq 0$ for all $\xi \in R^n$. The $\xi$-minimization of problem (SP') is therefore solved by $\xi = 0$, which means $s(x_0; C) = 0$ and $x^* = x_0$, just as with (DP). Therefore, problem (SP) is again equivalent to (DP), and equally uninteresting.

However, the formulation of (SP) suggests a small modification which is not apparent in the problem (DP). If a problem (SP=) is written, which consists of finding

$$\Delta(x_0; C) = -\min_{\xi}\max_{x}\xi^T(x - x_0) \qquad (16)$$

subject to

$$g_i(x) \geq 0, \; i = 1, n_c$$

and

$$1 - \xi^T\xi \underbrace{=}_{!} 0.$$

is defined, in which the only difference between it and problem (SP) is that the unit magnitude constraint on $\xi$ is an *equality* constraint, its solutions assume interesting properties. To begin with, when $x_0 \notin C$, the behavior of problems (SP) and (SP=) are the same.

However, the guaranteed activity of this constraint at the optimum means that $\xi$ will always be of unit length, and parallel to the vector from $x^*$ to $x_0$. Therefore $\Delta(x_0; C)$ will always have units of length. Because $\xi \neq 0$, any $x^*$ will always lie on the boundary of $C$ or

$$x^* \in \partial C.$$

Therefore, the magnitude of $\Delta$ will corresponds to the shortest normal distance to the boundary of $C$. As was the case with the Comba function, the sign of $\Delta$ may be used as an intersection predicate:

$$\Delta(x_0; C)\begin{cases} > 0 & x_0 \notin C, \\ = 0 & x_0 \in \partial C, \\ < 0 & x_0 \in C - \partial C. \end{cases}$$

When $\Delta < 0$ the problem (SP=) is no longer a convex program, and is therefore much harder to solve. It is not hard to think of situations in which there are several $x^* \in \partial C$ which satisfy the stationary conditions of (SP=); Figure 6 shows a couple of such cases. In the case where the point $A$ is taken as $x_0$, although there are multiple stationary conditions, there is a clear minimum distance to the boundary. In the case where $B$ is taken as $x_0$, both stationary conditions may be used, since their corresponding distances are equal. $B$ is called a *conjugate point*.

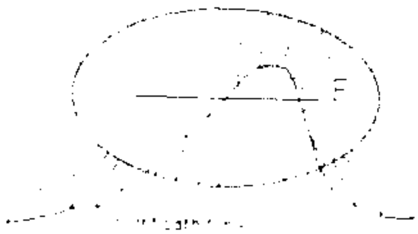

Figure 7: Δ Between a Point and a Body

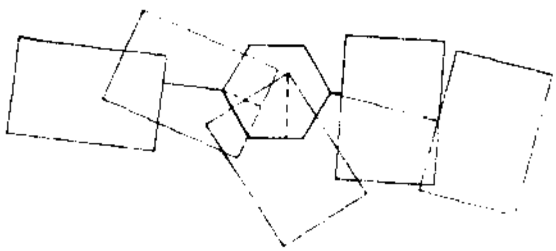

Figure 8: Δ Between Two Non-Point Bodies

### 3.3 Properties of $\Delta(x_0; C)$

These properties of $\Delta(x_0; C)$ are established in [Buckley 1985]:

- $\Delta(x_0; C)$, considered as a function of $x_0$ is convex.
- $\Delta(x_0; C)$ is a continuous function.
- $\xi$, the unit vector parallel to the generating segment, is always contained in the subdifferential of $\Delta(x_0; C)$, and may be used as a gradient.

An intuitive feeling for the behavior of Δ as a point $x_0$ passes near a body $E$ may be seen in Figure 7. Dashed generating segments correspond to negative values of Δ, and the magnitude of Δ is proportional to the length of the generating segment. Note that the subgradient of Δ will be discontinuous.

### 3.4 Extension to the Body-Body Case

The derivation for the body body case proceeds almost exactly as with the point body case, except that both ends of the generating segment are now free to move about in the minimizations. Details are given in [Buckley 1985]. Equivalence to the configuration space obstacle method described in [Lozano-Pérez 1981] is also established there. Δ between two convex bodies is also a continuous function, but because the relative position and orientation of two bodes must be expressed in terms of at least one rotation group, it cannot be a convex function of its arguments. An analytic expression which may always be used as the gradient of Δ has also been derived.

An intuitive feeling for the behavior of Δ as a rectangle moves near a hexagon may be seen in Figure 8. The encoding on the generating segment is the same as with Figure 7.

## 4 Computational Issues

While A may be attractive theoretically, it is not straightforward to compute for arbitrary convex bodies. However, if the bodies between which it is to be computed are *polyhedral sets,* then its computation becomes a combinatorial problem. Efficient ways of computing this function have been explored extensively [Buckley 1985]. Briefly, the complexity bounds O(nlogn) in the planar case, where *n* i is the combined number of vert ices of the two polyhedral bodies for which A is computed. The corresponding figure for spatial polyhedral bodies is $0(n^2 logn)$.

Implementational issues of the A function are currently being studied, and it is being incorporated into a free body path-planning system. The performance of this system is being assessed relative to a combinatorial system, that of [Brooks 1982].

## 5 Conclusion

A new function for characterizing the proximal relationship between two convex bodies has been developed, and its properties studied. This function makes it possible to implement path planning algorithms significantly different in capability from those heretofore reported.

## 6 Acknowledgement

The support of the Palo Alto VA Hospital in this research is gratefully acknowledged. Computational facilities and a super working environment were made available by the Fairehild Laboratory for Artificial Intelligence Research (now a part of Schlumberger). Many thanks to Rod Brooks for providing the code for his combinatorial configuration space planning algorithm to serve as a benchmark.

[Author's address: Design Division. Stanford. Cal. 94305]

## Bibliography

(Avriel 1976] Avriel, M., *Nonlinear Programming: Analysts and Methods,* Prentice-Hall, Inc., Englewood Cliffs, 1976.

[Brooks 1982] Brooks, R. A. and Lozano-Perez, T., "A Subdivision Algorithm in Configuration Space for Findpath with Rotation", M. I. T. A. I. Memo. No. 684, December, 1982.

[Buckley !98f>] Buckley, C. E., "The Application of Continuum Methods to Path Planning", Ph. D. Dissertation, Department of Mechanical Engineering, Stanford University, (to appear) 1985.

[Cahn 1975] Cahn, D. F. and Phillips, S. R., "ROBNAV: A Range-Based Robot Navigation and Obstacle Avoidance Algorithm", IEEE Transactions on Systems, Man, and Cybernetics, pp. 544-551, September 1975.

[Chatila 1981] Chatila, R., "Systeme de Nevigation pour un Robot Mobile Autonome: Modelisation et Processus Decisionnels", Doct.-Ing. Thesis, Paul Sabatier University, Toulouse, France, No. 7G2, 1981.

[Chazelle 1980] Chazelle, B. M., "Computational Geometry and Convexity", Ph. D. Dissertation, Computer Science Department, Carnegie-Mellon University, Report No. CMU-CS 80-150, 1980.

[Gomba 19C8] Comba, P. G., "A Procedure for Detecting Intersections of Three-Dimensional Objects", Journal of the Association of Computing Machinery, Vol. 15, No. 3., pp. 354-366, July 1968.

[Khatib 1980] Khatib, O., "Commande Dynamique dans l'Espace Operationnel des Robots Manipulateurs en Presence d'Obstacles", Doc.-Ing. thesis, French National Superior School of Aeronautics and Astronautics, 1980.

[Jarvis 1984] Jarvis, R. A., "Projection Derived Space Cube Scene Models for Robotic Vision and Collision-Free Trajectory Planning", 2nd International Symposium of Robotics Research, Kyoto 1984, pp. 294 301.

[Loeff 1975] LoefT, L. A. and Soni, A. H., "An Algorithm for Computer Guidance of a Manipulator in Between Obstacles", ASME Journal of Engineering for Industry, pp. 836-842, August 1975.

[Lozano Perez 1981] Lozano-Perez, To mas, "Automatic Planning of Manipulator Transfer Movements", IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-11, No. 10, pp. 681-698, October 1981.

(Myers 1981] Myers, J. K., "A Supervisory Collision-Avoidance System for Robot Controllers", M. S. Thesis, Carnegie Mellon University, December 1981.

[Peiper 1968] Peiper, D. L., "The Kinematics of Manipulators under Computer Control", Ph. D. dissertation, Mechanical Engineering Department, Stanford University, October 1968.

[Reif 1979] Rjeif, J. H., "Complexity of the Mover's Problem and Generalizations (Extended Abstract)", Proceedings of the 20th Symposium on the Foundations of Computer Science, pp. 421-427, 1979.

[Rockafellar 1970] Rockafellar, R. T., *Convex Analysis,* Princeton University Press, Princeton, N. J. 1970.

[Schwartz 1982] Schwartz, J. T. and Sharir, M., "On the Piano Movers' Problem II. General Techniques for Computing Topological Properties of Real Algebraic Manifolds", Courant Institute of Mathematical Sciences Report No. 41, New York University, February 1982.

[Whitney 1969] Whitney, D. E., "State Space Models of Remote Manipulation Tasks", IEEE Transactions on Automatic Control, Vol. AC-14, No. 6, pp. 617-623, December, 1969.