The Complexity of Planar Compliant Motion Planning Under Uncertainty

Bruce Donald

87-889
December 1987

Department of Computer Science
Cornell University
Ithaca, NY 14853-7501

The Complexity of Planar Compliant Motion Planning Under Uncertainty

Bruce R. Donald

Computer Science Department
Cornell University
Ithaca, NY 14853

Abstract: We consider the computational complexity of planning compliant motions in the plane, given geometric bounds on the uncertainty in sensing and control. We can give efficient algorithms for generating and verifying compliant motion strategies that are guaranteed to succeed as long as the sensing and control uncertainties lie within the specified bounds. We also consider the case where a compliant motion plan is required to succeed over some parametric family of geometries. While these problems are known to be intractable in 3D, we identify tractable subclasses in the plane.

Acknowledgements. This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the Laboratory's Artificial Intelligence research is provided in part by the Office of Naval Research under Office of Naval Research contract N00014-81-K-0494 and in part by the Advanced Research Projects Agency under Office of Naval Research contracts N00014-85-K-0124 and N00014-82-K-0334. The author was funded in part by a NASA fellowship administered by the Jet Propulsion Laboratory.

Contents

1 Introduction

In motion planning with uncertainty, the objective is to find a plan which is guaranteed to succeed even when the robot cannot execute it perfectly due to control uncertainty. With control uncertainty, it is impossible to perform assembly tasks which involve sliding motions using position control alone. To successfully perform assembly tasks, uncertainty must be taken into account, and other types of control must be employed which allow *compliant motion.*

Compliant motion occurs when a robot is commanded to move into an obstacle, but rather than stubbornly obeying its motion command, it complies to the surface of the obstacle. Work on compliant motion[1] attempts to utilize the task geometry to plan motions that reduce the uncertainty in position by maintaining sliding contact with a surface. Plans consisting of such motions can be designed to exploit the geometry of surfaces around the goal to guide the robot. By computing "preimages" of a geometrical goal in configuration space, guaranteed strategies can be synthesized geometrically: We call this a *geometrical theory of planning.* The first results in this theory begin with Lozano-Pérez, Mason, and Taylor (or [LMT]), with subsequent contributions by Mason [Ma2], Erdmann [E] and Donald [D,D2,D3,D4]. This research has led to a theoretical computational framework for motion planning with uncertainty, which we denote [LMT,E,D]. See [Buc, EM, Bro, CR] for other allied work.

The [LMT,E,D] framework begins by observing that the use of active compliance enables robots to carry out tasks in the presence of significant sensing and control errors. Compliant motion meets external constraints by specifying how the robot's motion should be modified in response to the forces generated when the constraints are violated. For example, contact with a surface can be guaranteed by maintaining a small force normal to the surface. The remaining degrees of freedom (DOF) can then be position-controlled. Using this technique, the robot can achieve and retain contact with a surface that may vary significantly in shape and orientation from the programmer's expectations. Generalizations of this principle can be used to accomplish a wide variety of tasks involving constrained motion, e.g., inserting a peg in a hole, or following a weld seam. The specification of particular compliant motions to achieve a task requires knowledge of the geometric constraints imposed by the task. Given a description of the constraints, choices can be made for the compliant motion parameters, e.g., the motion freedoms to be force controlled and those to be position controlled. It is common, however, for position uncertainty to be large enough so that the programmer cannot unambiguously determine which geometric constraints hold at any instant in time. For example, the possible initial configurations for a peg in hole strategy may be "topologically" very different, in that different surfaces of the peg and hole are in contact. Under these circumstances, the programmer must employ a combined strategy of force and position control that guarantees reaching the desired final configuration from all the likely initial configurations. We call such a strategy a *motion strategy.*

Motion strategies are quite difficult for humans to specify. Furthermore, robot programs are very sensitive to the details of geometry. For this reason, we have been working on the automatic synthesis of motion strategies for robots.

Note that compliant motion planning with uncertainty is significantly different from

[1] See [Ma] for an introduction and survey.

motion planning with perfect sensing and control along completely-known configuration space obstacle boundaries [Kou, HW, BK]. The two chief differences are:

- The planning of motions in contact with perfect control has the same time-complexity as planning free-space motions; that is, it can be done in time $O(n^r \log n)$ for r degrees of freedom and n faces or surfaces in the environment [Ca]; the exponent is worst-case optimal. However, prior to this paper, there are no upper bounds for planning compliant motions *with* uncertainty. However, for r fixed at 3, the problem is hard for non-deterministic exponential time [CR].
- From a practical point of view, the motion-in-contact plans generated under the assumption of perfect control cannot ever be executed by a physical robot using position control alone. While this difficulty in fact motivates our work, in this paper we concentrate on the geometrical and combinatorial aspects of the problem; for further details on issues in compliant motion we recommend the reference [LMT].

1.1 Dynamic Model

Compliant motion is only possible with certain dynamic models. We will employ the generalized damper model [Ma]. We assume that the environment is polyhedral, and that it describes the configuration space of the robot, so that the robot is always a point. The planned path consists of r successive motions in directions $v_1, \ldots, v_r$. Each motion terminates when it sticks, due to coulomb friction, on some surface in the environment.

Because of control uncertainty, however, the robot cannot move with precisely velocity v_i on the i^{th} motion. Instead, it moves with velocity v_i^{free}, which lies in a cone of velocities $B_{ec}(v_i)$ about v_i. The boundaries of the cone form an angle of ϵ_c with v_i. ϵ_c is called the *control uncertainty*, and $B_{ec}(v_i)$ the *control uncertainty cone about* v_i. It is, in fact, equivalent to regard ϵ_c as specifying that v_i^{free} lies within a ball about v_i in velocity space.

For a compliant motion, the robot moves along an obstacle surface with a sliding velocity v_i^{slide} which is the projection onto the surface of the obstacle of some v_i^{free} in $B_{ec}(v_i)$. Under generalized damper dynamics, the motion of a polyhedral robot without rotations is completely specified by the motion of its reference point in configuration space. See fig 1.

The i^{th} motion terminates by sticking on a surface when the velocity v_i^{free} in $B_{ec}(v_i)$ points into the negative coulomb friction cone on a surface. Thus sticking on a surface can be non-deterministic. We will assume that motion i can terminate on any reachable surface for which some velocity $v_i^{free} \in B_{ec}(v_i)$ is inside the negative friction cone. Sticking termination is motivated by the fact that a robot with a force-sensing wrist can easily recognize sticking and robustly terminate the motion.

To test whether sticking is possible on some set of (say, goal) edges, we simply perform a geometric cone intersection on each edge. Sticking is possible when the intersection of the cone of velocity uncertainty and the negative friction cone have a non-trivial intersection. Since determining the possibility (or necessity) of sticking reduces to a simple cone intersection, which may be done in constant time per edge, in this paper we will focus on the more difficult issue of computing reachability. However, the more general question of representing friction in configuration space is subtle; see [E, BRS].

Figure 1: (a) Peg in hole environment. (b) Configuration space, showing the motion of the reference point during compliant motion.

We define the predicate $stick_{v_i}(x)$ to be true at a configuration x when sticking is possible at x under commanded velocity v_i.

While robust implementation of generalized damper dynamics is still a research issue, in our robotics laboratory we have recently implemented an experimental force-control system with this dynamic model to test our geometrical planning theories [D6].

1.2 Definitions

We will regard the goal region G as a polyhedral region in configuration space. Since in general we cannot precisely know the initial configuration of the robot, we will also assume that the start region R is some polyhedral region in configuration space.

We now pose three problems:

Problem 1: One-Step Compliant Motion Planning with Uncertainty. Given a polyhedral start region R of constant size, a polyhedral environment $\mathcal{P}$ of n vertices, control uncertainty ϵ_c, coefficient of friction μ, and a polyhedral goal G of constant size, find one commanded motion direction v such that under v, all possible motions from R terminate by sticking in G.

Problem 2: One-Step Compliant Motion Verification. Given $(R, \mathcal{P}, \epsilon_c, \mu, G)$ and v, verify that under v, all possible motions from R terminate by sticking in G.

Problem 3: Compliant Motion Planning with Uncertainty Given $(R, \mathcal{P}, \epsilon_c, \mu, G)$, and an integer r, find a sequence of r motions such that each motion terminates in sticking, and the final motion terminates in the goal. Or, if no such r-step strategy exists, then say so.

In the sequel we will in fact assume that $\mathcal{P}$ is an *arrangement* of size n; that is, $\mathcal{P}$ is a set of configuration-space obstacle polyhedra whose interiors do not intersect. We assume in the exposition that R and G are convex. We believe this restriction may, in fact, be relaxed; see sec. 7.

2 Related and Previous Work

See [Bra, LP2, Ma, LMT, Y2, SHS] for backround on robotics, compliant motion, and algorithmic motion planning.

2.1 Related Work

2.1.1 Voronoi Diagrams and Retraction Methods

One might ask whether exact algorithms for motion plannning can ever be utilized after uncertainty in sensing and control are introduced. The answer is a qualified "yes." In particular, the Voronoi diagram has proved to be useful for motion planning among a set of obstacles in configuration space (see [OSY1, OSY2, OY, Y], and the textbook of Schwartz and Yap [SY] for an introduction and review of the use of Voronoi diagrams in motion planning). The Voronoi diagram, as usually defined, is a *strong deformation retract* of free space so that free space can be continuously deformed onto the diagram. This means that the diagram is complete for path planning, i.e. Searching the original space for paths can be reduced to a search on the diagram. Reducing the dimension of the set to be searched usually reduces the time complexity of the search. Secondly, the diagram leads to robust paths, i.e. paths that are maximally clear of obstacles. Hence Voronoi-based motion planning algorithms are relevant to motion planning with uncertainty. [CD, CD2] define a "Simplified Voronoi Diagram" which is still complete for motion planning, yet has lower algebraic complexity than the usual Voronoi diagram, which is a considerable advantage in motion planning problems with many degrees of freedom. Furthermore, the Simplified diagram is defined for the 6D configuration space of the "classical" movers' problem. For the 6DOF "classical" polyhedral case, [CD, CD2] show that motion planning using the Simplified diagram can be done it time $O(n^7 \log n)$. Of course, these methods do not address the compliant motion planning problem.

2.1.2 Tray-Tilting and Parts-Orienting

Erdmann and Mason [EM] have described an implemented tray-tilting planner for orienting planar polygonal parts. While they give no explicit time bounds for their algorithm, it is clear that it runs in exponential time. [Nat] showed that after introducing a variety of simplifying assumptions, the Erdmann-Mason planner can run in polynomial time. In addition, [Nat] demonstrated polynomial algorithms for other parts-orienting algorithms under uncertainty. [Bro] has developed algorithms for planning compliant grasps ("squeeze-grasps") under uncertainty.

2.1.3 Lumelsky's Approach

Lumelsky [Lum] considers the following related problem: suppose that a robot has a 2D configuration space, perfect control and sensing, the obstacles are finite in number, and each obstacle boundary is a homeomorphic image of the circle. Then a collision free-path may be found by tracing around the boundary of any obstacles encountered when moving in a straight line from the start to the goal. At each obstacle boundary encountered, there is a binary choice of which way to go, and the move may be executed with perfect accuracy. Lumelsky also demonstrates complexity bounds under these assumptions, and has considered configuration spaces such as the plane, the sphere, the cylinder, and the 2-torus. While it is not clear how this technique can extend to higher-dimensional configuration spaces, it is useful to compare Lumelsky's approach as an example of how to exploit a useful geometric primitive (wall-following). See also [Kod] for extensions to this approach using potential fields.

2.2 Previous Work on Compliant Motion Planning with Uncertainty

2.2.1 Lower Bounds

[CR] have shown that in 3D, the one-step verification problem (2) (and hence one-step planning (1)) is $\mathcal{NP}$-hard, and the multi-step planning problem (3) is $\mathcal{NEXPTIME}$-hard. Previously, [Nat] had shown that (3) is $\mathcal{PSPACE}$-hard.

2.2.2 Upper Bounds

Erdmann [E] has shown that in the plane, when G is a single edge of the polygonal environment $\mathcal{P}$, then the one-step verification problem (2) can be done in time $O((n+c)\log n)$, where c is the number of intersections encountered by a planar arrangement algorithm. Using plane-sweep techniques, Canny and Donald [D3] implemented an $O((n+c)\log n)$ algorithm for the case where G is polygon of size n, and c is the number of intersections of G with $\mathcal{P}$.

Buckley [Buc] implemented an interesting multi-step compliant-motion planner in 3D that uses sticking termination. While his algorithm is "heuristic" (in that it is not guaranteed to find a plan), it appears to generate a useful class of strategies in practice. [Buc] gives upper bounds of time $\left(2^{2^{O(n)}}\right)$.

2.2.3 Backprojections

Erdmann's algorithm makes use of *backprojections*, which he defined as a simplified case of the [LMT] notion of geometrical preimages. The question of goal reachability from a start region can be reduced to deciding the containment of the start region within the backprojection of the goal.

The backprojection of a goal G (with respect to a commanded velocity v_θ^*) consists of those configurations guaranteed to enter the goal (under v_θ^*).[2] That is, the backprojection is the set of all positions from which all possible trajectories consistent with the control uncertainty are guaranteed to reach G. See fig. 2. The terms "preimage" and "backprojection" come from viewing motions as "mappings" between subsets of configuration space. Hence the backprojection of a goal is the set of configurations from which a particular commanded compliant motion is guaranteed to succeed. [LMT] envisioned a back-chaining planner that recursively computes preimages of a goal region. Successive subgoals are attained by motion strategies. Each motion terminates when all sensor interpretations indicate that the robot must be within the subgoal.

Here is the key point about backprojections: Given $(R, \mathcal{P}, \epsilon_c, \mu, G, v_\theta^*)$, the one-step verification problem (2) reduces to testing set containment, i.e., that

$$R \subset B_\theta(G).$$

[2] The star * denotes the ideal, or perfect control velocity. Henceforth, we will typically identify a commanded motion v_θ^* with its angular direction θ.

Figure 2: The goal is the region G. Sliding occurs on vertical surfaces, and sticking on horizontal ones. The commanded velocity is v_θ^*, and the control uncertainty is $B_c(v_\theta^*)$. The *backprojection* of G with respect to θ is the region P.

Erdmann showed that when G is a single edge of the environment $\mathcal{P}$, then $B_\theta(G)$ has size $O(n)$ and can be computed as follows:

1. Find all vertices in the environment where sticking is *possible* under v_θ^*.
2. At each of these vertices, erect two rays, parallel to the two edges of the *inverted* velocity cone $-B_{ec}(v_\theta^*)$.
3. Compute the arrangement from the environment plus these additional $O(n)$ constraints.
4. Starting at the goal edge, trace out the backprojection region.

An excellent exposition of Erdmann's algorithm can be found in [E]. Canny and Donald implemented a plane-sweep algorithm for backprojections from general polygonal goals. The idea is similar, but interested readers may find details in [D3]. Both methods take time $O(n \log n)$ and space $O(n)$ when the goal has $O(n)$ intersections with $\mathcal{P}$.

3 Statement of Results

3.1 Restriction to *Planar* Compliant Motion Planning with Uncertainty

Here are the main results of this paper. We consider problems (1-3) in the plane, and call these problems the *Planar* Compliant Motion Planning with Uncertainty Problems. That is, we restrict G, R, $\mathcal{P}$ to be planar polygonal. Note that when we say *planar* we also mean that no rotations are allowed. So we may speak of the Planar One-Step Compliant Motion Planning Problem, the Planar One-Step Verification Problem, and so forth, so we have the following definition of the multi-step planar motion strategy generation problem. While the compliant motion strategies we consider employ sticking termination, for conciseness, we will not write this out in each definition.

Definition: *The* planar compliant motion planning problem with uncertainty *is defined as follows. Given a polygonal start region R of constant size, an integer r, a polygonal environment $\mathcal{P}$ of size n, control uncertainty ϵ_c, coefficient of friction μ, and a polygonal goal G of constant size, find a sequence of r motions $\theta_1, \ldots \theta_r$ such that each motion terminates in sticking, and the final motion θ_r terminates in the goal. Or, if no such r-step strategy exists, then say so.*

Definition: *The* one-step planar compliant motion planning problem with uncertainty *is defined as above, with $r = 1$.*

Figure 3: Geometric models of two gear-like planar objects A and B. A is grasped and can translate but not rotate. The orientation of B is unknown. The task is to generate a motion strategy to mesh the gears.

Figure 4: The configuration space for the gear example (fig. 3) at one α-slice ($\alpha = 0$) of $C \times J$. The goal region is the "valleys" of the cspace obstacle. The start region is the diamond to the lower left. B is not allowed to rotate, so no motion across J is possible.

Theorem 1. *The one-step planar compliant motion planning problem with uncertainty can be decided in time $O(n^4 \log n)$.*

Theorem 2. *The planar compliant motion planning problem with uncertainty is decidable in time $n^{r^{O(1)}}$.*

Theorems (1) and (2) represent the first upper bounds for compliant motion strategy generation under uncertainty. Comparison with lower bounds [CR] therefore indicates that the planar case is a tractable subclass. (In 3D one-step planning is $\mathcal{NP}$-hard and multi-step planning is worst-case doubly-exponential).

3.2 One-Step Compliant Motion Planning with Model Error

Finally, we consider the following problem as well. Suppose that we regard $\mathcal{P}$, the configuration space environment, to be generated by two sets of real-space polygons, A and B. A and B are each sets of convex polygons, which may overlap. The union of B represents the real-space obstacles. The union of A represents the robot (or the manipulated object). In general, if A and B both contain m edges, then the complexity of computing the resulting configuration space at a fixed orientation will be $O(m^2 \log m)$. We may, in fact, regard n as $O(m^2)$; see [LP, D3].

Now suppose that the orientation of B is fixed, but unknown ahead of time. Assume B will remain at the same orientation throughout any compliant motion strategy. What we want is a motion strategy that will succeecd for *any* initial orientation of B. For example, in [D3], an application where A and B are planar gears (fig. 3) was considered. A strategy was synthesized to mesh the gears despite initial uncertainty in their relative orientation.

We model this problem in a three-dimensional "generalized" configuration space $\Re^2 \times S^1$. Thus the initial state of the planar system is some triple (x, y, α), where (x, y) lies in $R \subset \Re^2$, and $\alpha \in S^1$ can be any orientation of the environment B. During the course of a compliant motion, α remains fixed; that is, we remain within some fixed, albeit unknown "slice" of the environment. Consider the one-step compliant motion verification problem. Suppose θ represents a direction in $\Re^2$. Hence the control uncertainty cone about v_θ is two-dimensional, and lies in the x-y plane at (x, y, α). This can be regarded as the problem of computing a 3D backprojection $B_\theta(G)$ in the "generalized" configuration space $\Re^2 \times S^1$. Thus a point in $B_\theta(G)$ represents an initial position (x, y) and an orientation α of the environment, from which we are guaranteed to achieve the goal G under commanded motion

Figure 5: The backprojection in slice $\alpha = 0$ of the goals in fig. 4, assuming that B cannot rotate. The coefficient of friction is taken to be .25.

in direction θ. The strategy θ may be considered verified when all such points (x, y, α) for $(x, y) \in R$ lie within $B_\theta(G)$. The problem is to decide containment of the region $R \times S^1$ within the 3D backprojection $B_\theta(G)$ of G in the generalized configuration space $\Re^2 \times S^1$.

We may view this construction as follows. We regard our generalized configuration space as having two degrees of motion freedom, called C, and one degree of variational model freedom, called J. Here C is $\Re^2$ and J is S^1. Motions can be commanded only in C, and the J-value never changes. The motion strategy must succeed for all J-values. We think of J as representing a parametric family of environments, or a set of toleranced parts for which the strategy must succeed. J is called the space of *model error*. This technique for planning with model error was introduced in [D]: essentially, we considered compliant motion planning problems with n degrees of motion freedom, and k dimensions of variational geometric model uncertainty. We reduced this planning problem to the problem of computing preimages in an $(n + k)$-dimensional generalized configuration space, which encompasses both the motion and the model degrees of freedom, and encodes the control uncertainty as a kind of non-holonomic constraint. Further details of the planning model are not required for this paper, but the interested reader is referred to [D,D2,D3,D4].

Our last result is:

Theorem 3. *Let R be a polygon of constant size, configuration space C be $\Re^2$, model error J be S^1, $B_\theta(G)$ be the backprojection of G in $C \times J$ as above. Suppose G is of constant size. Then there exists an algorithm deciding the containment of $R \times J$ in $B_\theta(G)$ in time $O(n^4 \log n)$.*

This means that one-step planar compliant motion verification with control uncertainty and 1 DOF rotational model error can be decided in the same time bound.

Theorem (3) represents a case where $n = 2, k = 1$ and containment in the backprojection can be computed in polynomial time (note for $n = 3$, $k = 0$, this is false [CR]).

Finally, we will describe how our algorithms can be generalized for R and G of size n, and how the bounds in theorems (1) and (3) might be tightened to $O(n^2 \log n)$ and $O(n^3 \log n)$, respectively.

4 Proof of Theorem 3

Verification is at least conceptually easier than generation. The development will be smoothest if we prove theorem (3) first.

4.1 Critical Slices: An Introduction

An *α-slice* of the generalized configuration space $\Re^2 \times S^1$ is the subspace $\Re^2 \times \{\alpha\}$. We now ask the question: what is the complexity of one-step planar compliant motion verification with control uncertainty and one DOF rotational model error?

The key to answering this question may be addressed using *critical slices.*[3] The idea is as follows. Consider the gear example, where gear A can translate and B has unknown orientation. *Assume that the orientation of B is fixed, so it cannot rotate when pushed by A.* Let α denote the orientation of B. Then consider the three-dimensional backprojection of G in $C \times J$. By taking x-y slices of the backprojection at different values of α, it is clear that generically, as α varies, the topology of the backprojection remains unchanged. Similarly for the forward projection (see below): The topology of two backprojection slices are the same if no edges or vertices appear or disappear at α values between them. At singular values of α, however, a small change in α will result in a change in the topology of the backprojection slice. Such a change is called a "catastrophe." These singular values are called *critical* α, and the generic values of α are called non-critical. Two critical values are called *adjacent* if there is no critical value between them.

The idea is that a planning algorithm can compute a backprojection slice at each critical value of α. In addition, between each adjacent pair of critical values, the algorithm computes a slice at a non-critical α. This slice of the backprojection at that value is representative of a continuum of intermediate non-critical slices. Between critical slices, in addition, it is clear how the surfaces of the backprojection change. The obstacle vertices of the backprojection, for example, move along curved edges that are algebraic helicoids. The obstacle edges are developable algebraic surfaces. The equations of the surfaces are found in [BLP]. The equations of the edges, as parameterized by orientation, are found in [Don]. No additional vertices may be introduced except at critical values. The free-space edges of the backprojection remain fixed across α between critical values. What we obtain is a complete combinatorial characterization of the 3D backprojection in $C \times J$. It can be used to derive precise, combinatorial algorithms for decision problems about the backprojection.

Suppose we wish to decide whether a start region R is contained in the 3D backprojection. (That is, to decide whether the goal is guaranteed reachable from the start region). By deciding the containment question, guaranteed strategies can be planned. Thus by deriving upper bounds on the containment problem in the backprojection, we obtain bounds for the planning of guaranteed strategies.

Suppose R has the form $U \times J$ for U a polygon in the plane. Then U must be tested for containment in each critical and non-critical slice as defined above. In addition, we must ensure that U lies inside the backprojection as the boundaries of the backprojection move with α. Since the equations of these surfaces are algebraically defined, we simply test them for intersection with the boundary edges of U.

The next question is: how many critical values of α are there? In the following lemma, when we speak of edges of the backprojection, or convex configuration space obstacle (CO) vertices, we mean edges of the backprojection in a slice, or a vertex which is convex in a slice. Of course these edges and vertices sweep out surfaces and curves (resp.) as α changes.

Lemma: *Let C be $\Re^2$, J be the circle S^1. Suppose m is size of the input in real-space edges so that $n = O(m^2)$ is the number of generalized configuration space constraints. Let G have constant size, and $B_\theta(G)$ be the backprojection of G in $C \times J$ as above. Then there are $O(n^3)$ critical values of $\alpha \in J$ for $B_\theta(G)$.*
Proof: We enumerate the various types of critical values:

[3] Note that slice methods have been studied in other domains. See, for example, [LP, SY, E].

A. First, an α value is (potentially) critical when a new edge or vertex is introduced into, or disappears from, the union of the configuration space obstacles. This can introduce a topological change in the obstacle boundary of the backprojection. If A and B are convex, then as α varies, there are potentially m^2 topological changes in the configuration space obstacles. These generate $O(m^2)$ critical values of α, which we call *obstacle*-critical. However, when A and B are non-convex, there can be $O(m^6)$ obstacle-critical values. This bound arises as the number of critical values for an arrangement of m^2 surfaces in dimension $d = 3$.

B. In addition, an α value can be critical if the determination of sliding vs. sticking on an edge can change there. A change in sliding can result in the introduction or deletion of a free-space constraint, and hence change the free-space boundary of the backprojection. This occurs when an edge of the friction cone on some edge becomes parallel to an edge of the velocity cone of control uncertainty. Now, as a configuration space edge rotates with α, its friction cone rotates with it. Thus as α changes, a friction cone edge can be parallel to a velocity cone edge at most 4 times. Hence there can be at most $4n$ values of α at which the sliding determination changes. These values are called *sliding*-critical.

C. Next, the topology of a slice of $B_\theta(G)$ can change when a convex vertex of a rotating configuration space obstacle edge touches a free-space edge of the backprojection. These α-values are called *vertex*-critical. Now, each free-space edge of a backprojection slice is anchored at a convex configuration-space obstacle (CO) vertex. Vertex-criticality occurs when a free-space edge of a backprojection slice joins two CO vertices in that slice. The edge then lies in the visibility graph of the generalized configuration space obstacles in that slice. Now, we can obtain a bound of $O(m^6)$ on the number of vertex-critical values as follows. Introduce an additional $O(m^2)$ constraints, each anchored at a convex CO vertex and parallel to the left or right edge of the velocity cone. These, together with the $O(m^2)$ obstacle surface constraints form an arrangement of $O(m^2)$ surfaces in 3 dimensions, yielding a total of $O(m^6)$ critical values. This bound may be improved to $O(m^4)$, by observing that each vertex-critical value is generated by a pair of convex CO vertices, and that there exist $O(m^2)$ such vertices.

D. Finally, an *edge*-critical value occurs when a configuration space edge, rotating with α, touches a free-space backprojection vertex. Free-space backprojection vertices are formed by the intersection of two free-space edges of the backprojection. Each free-space edge of the backprojection is anchored at a convex CO vertex. The number of edge-critical values is $O(m^6)$, because each is generated by a CO edge, and two convex CO vertices (one per free-space backprojection edge).

Finally, we observe that these bounds are additive, and that n is $O(m^2)$. □

Comments: We conjecture that the bounds on edge-critical values (D) can be improved to $O(m^4)$. One approach to proving the improved bound is to identify each free-space vertex v of the backprojection, with the right generating CO vertex. Follow the locus of v as α varies. It remains to show that the locus is piecewise-smooth, and touches each CO edge at most a fixed number of times.

We can now address the complexity of deciding containment in the backprojection.

Theorem 3: *Let U be a polygon of constant size, C be $\Re^2$, J be S^1, $B_\theta(G)$ be the backprojection of G in $C \times J$ as above. Suppose G is of constant size. Then there exists an algorithm deciding the containment of $R = U \times J$ in $B_\theta(G)$ in time*

$$O(n^4 \log n).$$

Proof: $O(n^3)$ slices of the backprojection can be computed in time $O(n^4 \log n)$. Now, to test for containment of U in the 3D backprojection region between two adjacent critical slices will take time $O(n)$, since the backprojection has size $O(n)$. The cost of deciding the containment of U between successive adjacent pairs of n^3 slices, each of size n, is $O(n^4)$. Since the time for computation of the slices dominates, this yields total complexity $O(n^4 \log n)$. □

4.2 Comments: Generic Singularities

Some comments are in order. First, our algorithm is naive, in that each backprojection slice is recomputed from scratch. In fact, this extra work is unnecessary. At a critical value of α, very few aspects of the topology of the backprojection will change. That is, typically, only one or two edges will be introduced or disappear at any critical value. We can make this notion precise as follows. If α is a generic singularity, then exactly one edge or vertex will appear or disappear there. Hence, for example, we can ensure that all critical values are generic singularities with probability one by subjecting the input to small rational perturbations.

Suppose that a backprojection has been computed in a critical slice at α. Then to compute a backprojection in a nearby non-critical slice at $\alpha + \epsilon$, we merely need to update the portion of the backprojection boundary that was critical at α. This requires only constant work: only one edge or vertex must be changed to derive a backprojection in the new slice! The new slice, furthermore, need not be copied in entirety. Instead, the representation for the new slice can simply indicate how it has changed from the old slice. It seems reasonable to conjecture that this technique would yield an algorithm of complexity $O(n^3 \log n)$ for deciding containment in a backprojection.

Finally, it appears that there are many problems in which the number of critical values fails to achieve the theoretically possible n^3 bound. This is because characteristically, there are orientation restrictions; typically, even with model error, B is not allowed to rotate freely. In other cases, there are symmetries. For example, in the gear case, even though B is allowed to rotate freely, it is unnecessary to consider n^3 slices since due to symmetry the configuration spaces "repeat" periodically.

4.3 Comparison with Lower Bounds

Consider the one-step compliant motion planning problem in 3D amidst precisely known polyhedral obstacles. This problem may be addressed via 3D backprojections in $\Re^3$. [CR] have shown that deciding containment in such a 3D backprojection is NP-hard. In particular, such backprojections can have an exponential number of faces. However, in the previous theorem we demonstrated a special class of 3D backprojections that have only $O(n^4)$ faces, along with an efficient algorithm for deciding containment. This special class of backprojections arises in the presence of model error. Specifically, they arise when C is $\Re^2$, J is one-dimensional, and no motion is permitted across J. In this case, the non-holonomic constraints that keep the robot within one slice essentially disallow the kind of fanning out and branching that [CR] discovered in $\Re^3$. Thus, our polynomial-time algorithm identifies a tractable subclass of the 3D motion planning problem with uncertainty. This subclass is also interesting in that it arises naturally in planning with model uncertainty.

5 Non-Directional Backprojections

5.1 Intuition

Let us now return to the assumption of no model error. We now address problems (1) and (3) in the plane, that is, the problem of planar compliant motion planning with uncertainty for one-step (1) and for multi-step (3) strategies. To this end we define a combinatorial object called the *non-directional backprojection*, and give a critical slice algorithm for constructing it. The non-directional backprojection may be used to represent, in a sense, "all possible backprojections" of a fixed goal. We intend to use it to generate motion strategies.

[LMT] first defined non-directional preimages. Erdmann [E] defined the non-directional backprojection as the union of all backprojections in the plane:

$$\bigcup_{\theta} B_\theta(G).$$

We will use a different definition. However, it is in the same spirit as [LMT,E], and so we will employ the same name. We must point out, however, that both M. Erdmann and R. Brost have considered[4] a similar construction for generating commanded velocities, and also thought about a critical slice approach to computing it.

Our definition exploits generalized configuration space. Consider the following intuitive argument.

a. Suppose we have a planar polygonal environment with no model error. In generating motion strategies, we do not know which way to point the robot—that is, we do not know which way to command the motion. Thus in some sense, there is "uncertainty" in "which way to go." This "uncertainty" is the variable θ. Thus we have a kind of three-dimensional planning problem, with degrees of freedom x, y, θ. We intend to map this uncertainty in "which way to go" into generalized configuration space.

b. Now, consider a problem which is in some sense dual to generating motion strategies. In this problem, we only consider *one* commanded motion in a fixed direction v_0^*. However, there is total uncertainty in the orientation of the entire environment. We may represent this uncertainty by a variable θ also.

[4] [Personal communication]. I am grateful to M. Erdmann for pointing out the similarity of the construction.

Clearly, both problems (a) and (b) can be represented in an generalized configuration space where x and y are the degrees of motion freedom, and θ is "model error." Here is the difference, however. In (b), θ is universally quantified: that is, we are required to ensure that a motion strategy succeeds for *all* θ. In (a), however, θ is existentially quantified. We merely need one θ to find a commanded motion.

The precise analogue of (a) is a problem like (b) in which we get to choose the orientation of the environment such that the v_0^*, the fixed commanded motion under consideration, will guarantee reaching G.

5.2 Computing the Non-Directional Backprojection

We now make the intuitive argument more precise. Let J be the space of all commanded motions, so that J is exactly the circle, S^1. We write $\theta \in J$ for a commanded motion direction.

Definition: *Let G be a goal amidst polygonal obstacles in the plane. The* Non-Directional Backprojection $B(G)$ *of G is a set in* $\Re^2 \times J$,

$$B(G) = \bigcup_{\theta} \Big(B_\theta(G) \times \{\,\theta\,\} \Big). \tag{1}$$

Now, recall the critical slice algorithm of sec. 4.1. This algorithm computes 3D directional backprojections in a three dimensional generalized configuration space, $\Re^2 \times S^1$. It operates by determining critical orientations at which the topology of backprojection slices change.

$B(G)$ is also a 3D backprojection-like region. We can develop critical slice algorithms for computing $B(G)$ also. They will work by finding all values of θ at which the topology of $B_\theta(G)$ can change. Then the algorithm takes slices at these critical θ's and at an intermediate non-critical θ's between each pair of adjacent critical values.

Now, $B(G)$ is bounded by developable algebraic surfaces. These surfaces are of two types, *obstacle* surfaces, and *free-space* surfaces. The obstacle surfaces are liftings into $\Re^2 \times J$ of the obstacle edges in $\Re^2$. The free-space surfaces are swept out by free-space edges of $B_\theta(G)$ as they rotate with θ. The manner in which the bounding algebraic surfaces of $B(G)$ sweep between slices is completely known—the obstacle edges stay fixed, while the free-space edges rotate with θ, remaining parallel with edges of the velocity cone. Now, each free-space edge is anchored at an obstacle vertex cobounding a possible sticking edge. As θ varies, the free-space edge rotates about that vertex. Clearly, as θ varies, the topology of $B_\theta(G)$ can change if the free space edge contacts an obstacle vertex. When this happens, there is an edge connecting two obstacle vertices which is parallel to an edge of the commanded velocity cone. Next, we note that any such edge lies in the *visibility graph* of the planar input environment. The visibility graph may be computed in time $O(n^2)$. This gives us the following lemma, which gives an upper bound on the number of critical values of θ. Here is the intuition behind the lemma:

Consider a free-space edge $e_i(\theta)$ of $B_\theta(G)$. $e_i(\theta)$ lies in the infinite half-ray $r_i(\theta)$ which extends from $e_i(\theta)$'s anchor vertex. We call $r_i(\theta)$ a *constraint ray*; it is parallel to an edge of $B_{ec}(v_\theta^*)$. There are $O(n)$ constraint rays in each backprojection slice $B_\theta(G)$. $r_i(\theta)$ rotates with θ, and it can intersect $O(n)$ obstacle edges as θ sweeps along. Now, how many other constraint rays of the form $r_j(\theta)$ can $r_i(\theta)$ intersect as it rotates? Note that all constraint rays $\{ e_j(\theta) \}$ move "with" $r_i(\theta)$, and are either parallel to it, or else intersect it always. Therefore how $r_i(\theta)$ can intersect these other constraint rays as θ sweeps is also $O(n)$.

We assume that the input polygons represent configuration space obstacles.[5] We use the boundary operator ∂ to denote the topological boundary.

Lemma: *Given a goal G of constant size and an arrangement of input polygons $\mathcal{P}$ of size $O(n)$, there are $O(n^2)$ critical values of θ in the non-directional backprojection $B(G)$.*
Proof: Let $B_{ec}(v_\theta^*)$ denote the control velocity uncertainty cone about a commanded velocity v_θ^*. We think of $B_{ec}(v_\theta^*)$ as rotating with θ. The topology of $B_\theta(G)$ can change when any of the following occur:

A. An edge of $B_{ec}(v_\theta^*)$ becomes parallel to an edge in the visibility graph of $\mathcal{P}$. Such values of θ are called *vgraph*-critical.

B. θ is a *sliding*-critical value (see sec. 4.1), where the determination of sliding vs. sticking behavior on an edge can change. Sliding-critical values occur when an edge of $B_{ec}(v_\theta^*)$ becomes parallel to the edge of a friction cone on some configuration space edge.

C. Let $e_i(\theta)$ and $e_j(\theta)$ be free-space edges of $B_\theta(G)$. They rotate with θ about their anchor vertices. Let $p_{ij}(\theta)$ denote their intersection; it is a free-space vertex of the backprojection. Then θ is *vertex*-critical when $p_{ij}(\theta) \in \partial B_\theta(G)$ and $p_{ij}(\theta)$ intersects some obstacle edge.

Now, there are $O(n^2)$ edges in the visibility graph of $\mathcal{P}$. In sec. 4.1 we showed that there are $O(n)$ sliding-critical values. Only sliding-critical values can introduce additional constraint rays.

Now, since there are $O(n)$ constraint rays in each slice, it would appear *a priori* that there could be potentially $O(n^2)$ $p_{ij}(\theta)$'s. Note, however, that each free-space vertex $p_{ij}(\theta)$ of the backprojection can be identified with exactly one constraint ray, say the "left" one, $r_i(\theta)$. Hence we see that there are merely $O(n)$ $p_{ij}(\theta)$'s. Each moves in a circle. Observe that in effect, each free-space vertex of the backprojection moves with θ in a piecewise-circular, possibly disconnected locus. Consider the discontinuities in the locus caused by type (A) or (B) critical values. In between discontinuities, each circular arc in the locus can intersect only a fixed number of obstacle edges. In particular, the arc cannot intersect n obstacle edges without "using up" more type (A) or (B) critical values. Hence, there are $O(n^2)$ vertex-critical values of θ.

Next we observe that the bounds for (A) (B) and (C) are additive. In particular: the bounds on vertex-critical and vgraph-critical values apply to all *possible* free-space edges; hence the vgraph-critical and vertex-critical values do not interact and their complexities do

[5] See sec. 3.2 for the complexity where the input is given in real space obstacles.

not multiply. Similarly, the sliding-critical bounds cover all possible ways that a constraint ray can be added or deleted from the backprojection boundary as θ changes. Hence this bound is also additive. Thus we obtain the $O(n^2)$ upper bound. □

Corollary: *There exists a representation of size $O(n^3)$ for the non-directional backprojection $B(G)$.*
Proof: Take $O(n^2)$ slices at critical values. Compute a backprojection slice $B_\theta(G)$ of size $O(n)$ at each of the critical values of θ. □

Comments: This upper bound means that $O(n^2)$ slices are required for a critical slice representation of $B(G)$. However, as in sec. critical, it seems that this upper bound will almost never be attained in practice. In practice we will consider only small ranges of θ. For example, for a peg-in-hole strategy, we would probably only consider directions in the lower (downward) half-plane. While these arguments do not affect the worst-case complexity, they do suggest that in practice the number of critical θ values may be smaller than $O(n^2)$.

We can now address the complexity of computing $B(G)$. By this we mean, what is the complexity of computing a precise, combinatorial description of $B(G)$. The output representation is a finite ordered set of alternating critical and non-critical slices $\{ B_{\theta_{c_1}}(G), B_{\theta_{nc_1}}(G), \ldots \}$, along with an algebraic description of how the free-space edges of the backprojection change between slices. (For a free-space edge, this is completely specified by the anchor vertex and an interval of θ for which the surface bounds $B(G)$).

As above, let $\mathcal{P}$ be an arrangement of input polygons representing configuration space obstacles.

Theorem 4: *Given a goal G of constant size and an arrangement of input polygons $\mathcal{P}$ of size $O(n)$, a representation of size $O(n^4)$ for the non-directional backprojection $B(G)$ can be computed in time $O(n^4 \log n)$.*
Proof: First, we compute the critical values of θ. Sliding-critical values can be computed in linear time. Vgraph-critical values can be computed in time $O(n^2 \log n)$. While it may be possible to compute the vertex-critical values in quadratic time, we give the following simple $O(n^3)$ algorithm: Intersect all constraint rays to obtain $O(n^2)$ points $p_{ij}(\theta)$. Each of these points is a possible free-space vertex of the backprojection, and each moves in a circle with θ. Intersect these circles with the obstacle edges to obtain $O(n^3)$ possible critical values of θ. The actual vertex-critical values will be contained in this set.

The planar backprojection algorithms referenced in sec. 2.2.3 compute a 2D backprojection slice in time $O(n \log n)$, and the output has size $O(n)$. Using this algorithm, compute $O(n^3)$ slices $B_\theta(G)$, at each the possibly-critical value θ. □

Some comments are in order. Again our algorithm is naive, in that each backprojection slice is recomputed from scratch. If θ is a generic singularity, then exactly one edge or vertex of $B_\theta(G)$ will appear or disappear there. It is reasonable to speculate that application of the techniques suggested in sec. 4.2 would yield an algorithm of time and space complexity $O(n^2 \log n)$ for computing $B(G)$. (The log factor arises from the apparant necessity of sorting the critical values).

Figure 6: The (directional) forward projection $F_\theta(R)$ in slice $\alpha = 0$ of the diamond-shaped start region R in fig. 4, assuming that B cannot rotate.

5.3 Generating One-Step Strategies using the Non-Directional Backprojection

Let C be the configuration space $\Re^2$, and J the space of commanded motions S^1 as above. Define the projection map

$$\begin{aligned} \pi_J : C \times J &\rightarrow J \\ (x, y, \theta) &\mapsto \theta. \end{aligned}$$

Consider algorithm *One-Step* which computes the set T of all motions guaranteed to reach G from a start region R:

Algorithm One-Step

```
1. $R_1 \leftarrow R \times J$.
2. $T \leftarrow J - \pi_J\Big(R_1 - B(G)\Big)$.
3. *Return any* $\theta \in T$.
```

We must now argue that given our representation of $B(G)$, the set difference required above can be done efficiently. R is convex and of constant size. Assume *wlog* that the closure of R does not intersect any obstacle. $B(G)$ is bounded by $O(n)$ surfaces in $C \times J$. We throw R_1 into the arrangement of these surfaces, and introduce a new type of critical value, called *R-critical.* An *R-critical* value θ_i arises when an edge of R intersects an edge of $B_{\theta_i}(G)$. (Equivalently, an R-critical value arises when ∂R_1 intersects $\partial B(G)$.) There are $O(n)$ R-critical slices. Now, suppose that in each R-critical slice θ_i, we label each vertex v of R as *in* or *out*, depending whether or not $v \in B_{\theta_i}(G)$. Then we merely need to find some θ_i with all vertices of R labeled *in.*

This shows that One-step Planar Compliant Motion Planning with Uncertainty can be decided in time $O(n^4 \log n)$. Thus theorem (1) is proved.

6 Planar Multi-Step Compliant Motion Planning with Uncertainty

6.1 The Directional Forward Projection

See figs. 3 and 6, which show the *forward projection* [E] of a commanded motion. This region is the outer envelope of all possible trajectories evolving from the start region in fig. 3, under a *particular* commanded motion. It is the set of all configurations that are reachable from the start region, given the particular commanded velocity and control uncertainty cone.

Formally, the *forward projection* of a set R under commanded motion θ is all configurations which are possibly reachable from R under v_θ^* (subject to control uncertainty). It is denoted $F_\theta(R)$. [Buc] described the first algorithms for computing forward projections. Canny and Donald [D3] showed how in the plane, the same plane-sweep algorithm for backprojections (see sec. 2.2.3) can also be used to compute forward projections. This algorithm works by sweeping from R in the direction of θ. When R intersects the environment $\mathcal{P}$ $O(n)$ times, the forward projection $F_\theta(R)$ can be computed in time $O(n \log n)$ and has size $O(n)$. The forward projection algorithm is quite similar to that of sec. 2.2.3; but interested readers may find details in [D3].

We note that the forward projection $F_\theta(R)$ is "directional", in that it depends on the direction of commanded motion, θ.

6.2 The Non-Directional Forward Projection

The analogue the non-directional backprojection is the non-directional forward projection:

Definition: *Let R be a start region amidst polygonal obstacles in the plane. The* Non-Directional Forward Projection $F(R)$ *of R is a set in $\Re^2 \times J$,*

$$F(R) = \bigcup_\theta \Big(F_\theta(R) \times \{\,\theta\,\} \Big). \tag{2}$$

As a corollary to our bounds on the complexity of the non-directional backprojection, we obtain the following theorem which may be derived *mutatis mutandis*:

Theorem 5: *Given a start region R of constant size and an arrangement of input obstacle polygons $\mathcal{P}$ of size $O(n)$, let $F(R)$ be the non-directional forward-projection of R. Then*

a. *there are $O(n^2)$ critical values of θ for $F(R)$;*

b. *there exists a representation of size $O(n^3)$ for $F(R)$;*

c. *a representation of size $O(n^4)$ for $F(R)$ can be computed in time $O(n^4 \log n)$.* □

We will need the following corollary later:

Corollary: *For a constant-sized start region R and goal region G, amidst an arrangement of input obstacle polygons $\mathcal{P}$ of size $O(n)$, the non-directional forward projection $F(R)$ and non-directional backprojection $B(G)$ have representations as polynomial-sized formulae in the language of semi-algebraic (s.a.) sets. Furthermore, these formulae are quantifier-free.*
Proof: We can represent the non-directional forward projection (resp., backprojection) at a polynomial (in n) number of critical values $\{\,\theta_1, \ldots, \theta_l\,\}$ via the formula

$$A = \bigwedge_{i=1}^{l} (\theta = \theta_i \quad \Longrightarrow \quad (x, y) \in F_{\theta_i}(R)). \tag{3}$$

Let two adjacent critical values be $\theta_i^{\min}$ and $\theta_i^{\max}$. In between adjacent critical values of θ, the non-directional projection is bounded by a fixed[6] set of $O(n)$ developable algebraic surfaces. That is, when θ is between $\theta_i^{\min}$ and $\theta_i^{\max}$, the non-directional projection is the intersection of some fixed set of $O(n)$ algebraic half-spaces. These half spaces are represented by algebraic inequalities, $\{\, g_{ij}(x,y,\theta) \leq 0 \,\}$ where each g_{ij} is a polynomial. The form of the g_{ij} is discussed in 5.2. We define the predicate

$$C_i = \bigwedge_{j=1}^{m_i} (g_{ij}(x,y,\theta) \leq 0), \tag{4}$$

where m_i is $O(n)$. (4) only addresses the convex case: in the non-convex case, C_i still has linear size since there are only $O(n)$ g_{ij}'s. Specifically, C_i becomes a linear-sized disjunction of conjunctions of the g_{ij}'s halfspaces. We construct the non-directional projection as a s.a. set in a case statement,

$$A \wedge \bigwedge_i (\theta \in (\theta_i^{\min}, \theta_i^{\max}) \implies C_i).$$

□

6.3 Proof of Theorem 2

Above, we described a polynomial time exact algorithm for generating one-step guaranteed compliant motion strategies amidst planar polygonal obstacles. We now address the general case of generating guaranteed r-step compliant motion strategies. Assume sticking termination. Recall sec. 1.1: for a configuration x, we defined the predicate $stick_\theta(x)$ to be true when sticking is possible at x under commanded velocity v_θ^*. Let us define $F_{*\theta}(R)$, the *push-forward of θ from R*, to be all configurations x in the forward projection $F_\theta(R)$ such that sticking is possible at x. That is,

$$F_{*\theta}(R) = \{\, x \in F_\theta(R) \mid stick_\theta(x) \,\}.$$

By analogy with the non-directional backprojection, we defined the non-directional forward projection. We also observed that all directional projection sets are semi-algebraic (s.a.). Then by the lemma on critical values of $B(G)$, so are the non-directional projection sets. Furthermore, when R has constant size, the lemma shows that the non-directional projection sets have descriptions (as s.a. sets) that are polynomial in the size of the input arrangement $\mathcal{P}$.

Theorem 2: *The planar compliant motion planning problem with sticking termination is decidable in time $n^{r^{O(1)}}$.*

Proof: Let $p_0, \ldots, p_m \in \Re^2$. We define the predicates

$$f_\theta(p_1, p_2) \iff p_2 \in F_\theta(p_1) \tag{5}$$

and

[6] i.e., fixed between $\theta_i^{\min}$ and $\theta_i^{\max}$.

$$f_\theta^*(p_1,p_2) \iff p_2 \in F_{*\theta}(p_1). \tag{6}$$

Clearly, definition (6) is equivalent to

$$f_\theta^*(p_1,p_2) \iff f_\theta(p_1,p_2) \wedge stick_\theta(p_2). \tag{7}$$

We have shown how in polynomial time to compute a quantifier-free polynomial-sized formula (in n) for the s.a. set $F(p_i)$—the non-directional forward projection of p_i. It remains to show that (5), and consequently (6) are polynomial-sized predicates. Now, $\theta \in S^1$, $p_1 \in \Re^2$, and $p_2 \in \Re^2$. Consider $f_\theta(\cdot,\cdot)$ as a predicate on a 5D space $S^1 \times \Re^2 \times \Re^2$, that is, as $f(\theta, p_1, p_2)$. We can obtain a bound on the complexity of f by enumerating all possible edges of $F_\theta(p_1)$ as θ and p_1 vary. These edges then sweep out developable algebraic surfaces in the domain of the predicate. There are four types of edges that can bound $F_\theta(p_1)$:

a. An edge e_i of a generalized configuration space obstacle. These edges sweep out n surfaces of the form $S^1 \times \Re^2 \times e_i$.

b. A free-space edge anchored at a vertex v_j of a generalized configuration space obstacle and parallel to the left or right edge of the velocity cone. Let $r(v_j,\theta)$ denote the infinite ray anchored at v_j at orientation θ. Then type (b) edges sweep out $2n$ surfaces of the form $\bigcup_\theta(\{\theta\} \times \Re^2 \times r(v_j, \theta \pm \epsilon_c))$.

c. A free-space edge anchored at p_1 and parallel to the left or right edge of the velocity cone. These edges sweep out 2 surfaces of the form

$$\bigcup_\theta \bigcup_{p_1} \Big(\{\theta\} \times \{p_1\} \times r(p_1, \theta \pm \epsilon_c) \Big).$$

d. A partial edge of a generalized configuration space obstacle. Let v_1, v_2 be the vertices of a generalized configuration space obstacle edge. A partial generalized configuration space edge can start at v_1 or v_2 and extend to v', where v' is a vertex of a type (b) or (c) free-space edge. Clearly v' simply arises as the intersection of a type (a) surface with a type (b) or (c) surface.

By enumeration, we clearly obtain a linear ($O(n)$) bound on the number of surfaces in the 5D domain of f. The arrangement of these surfaces has polynomial size; in particular, it has $O(n^5)$ critical values. Hence we may conclude that f is a predicate of polynomial size in n.

Now, define

$$\mathcal{F}(p_0, \ldots, p_m, \theta_1, \ldots, \theta_m) \iff f_{\theta_1}^*(p_0,p_1) \wedge f_{\theta_2}^*(p_1,p_2) \wedge \cdots \wedge f_{\theta_m}^*(p_{m-1},p_m). \tag{8}$$

Since (6) has polynomial size in n, clearly the predicate (8) has polynomial size in n as well. Furthermore, it is quantifier-free.

Now, we let the points p_i serve as via points (sometimes known as switch-points) for the strategy. We quantify over all possible via points achievable by the motion strategy $\theta_1, \ldots, \theta_r$. By letting m be r, this is sufficient.

We can formulate the question of the existence of an r-step strategy as a decision problem within the theory of real closed fields:

$$(\exists\theta_1, \ \ldots, \theta_r) \\ \Big(\ \forall p_0, \ldots, p_r \ \ \Big((p_0 \in R) \wedge \mathcal{F}(p_0, \ldots, p_r, \theta_1, \ldots, \theta_r)\Big) \Longrightarrow (p_r \in G)\Big). \tag{9}$$

Now, deciding sentences in the theory of real closed fields is known be doubly-exponential only in the number of quantifier alternations. More specifically, the truth of a Tarski sentence for k polynomials of degree $< d$ in r variables, where $a \leq r$ is the number of quantifier alternations in the prenex form of the formula, can be decided in time

$$(kd)^{O(r)^{4a-2}},$$

(see [Gri]). We have $a = 2$, and hence (9) can be decided in time $n^{O(r)^6}$. □

6.4 Discussion

This result is of interest for the following reasons. First of all, the *general* compliant motion planning problem with uncertainty (in 3D) is known to be hard for non-deterministic exponential time [CR]. This means that any algorithm for the problem takes at least doubly-exponential time in the worst case. In this section, we have introduced restrictions on the problem which make it more tractable. These restrictions are:

- The configuration space is the plane, where directional forward projections have linear size. (In 3D they can have exponential size). A key step in our construction was then to show that the non-directional backprojection $B(G)$ has polynomial size.
- Sticking termination is used.
- The maximum number of steps in the strategy is given as input to the algorithm.

With these restrictions, the problem becomes decidable in time roughly exponential in r. In fact, we conjecture that for a great number of planning problems, r is in fact a small constant. When r may be so regarded, we effectively obtain a polynomial-time algorithm for this restricted planar motion planning problem with uncertainty.

It might have been possible to devise these restrictions *a priori*, from a strictly complexity-theoretic viewpoint. However, I believe that in light of [LMT,E,EM,Bro,Buc,D,D2,D3,D4] it becomes clear that these restrictions are indeed *physically* meaningful, and in fact define a useful and interesting subclass of planning problems. In a way, this work has been an exploration of problems solvable within these restrictions. From this perspective, I believe it is reasonable to conjecture that a large class of planning problems do fall under this rubric.

7 Improving the Bounds

We are currently working on improving the bounds in theorems (1) and (3) to $O(n^2 \log n)$ and $O(n^3 \log n)$, respectively. We believe that both problems can be reduced to decision problems within the purely existential theory of real closed fields. This reduction would yield the improved bound, and generalize the results for G and R non-convex, having size n. See [D5] for details.

8 Conclusions

The chief goal of this paper was to analyze the complexity of compliant motion planning with uncertainty. While in general it is known that the problem is intractable, we were able to demonstrate a number of special cases where there exist efficient algorithms.

We introduced a combinatorial object call the *non-directional backprojection*, and analyzed its complexity. Our analysis led to efficient algorithms for certain subproblems in compliant planning with uncertainty. In particular, we gave an efficient algorithm for planning one-step strategies in the plane. By using results from computational algebra, we showed that planning a guaranteed planar multi-step strategy with sticking termination can be decided in time polynomial in the geometric complexity, and roughly singly-exponential in the number of steps in the plan.

We also considered compliant motion planning problems with n degrees of motion freedom, and k dimensions of variational geometric model uncertainty. We reduced this planning problem to the problem of computing preimages in an $(n+k)$-dimensional generalized configuration space, which encompasses both the motion and the model degrees of freedom, and encodes the control uncertainty as a kind of non-holonomic constraint. We demonstrated a case where $n = 2$, $k = 1$ and containment in the backprojection could be computed in polynomial time (note for $n = 3$, $k = 0$, this is false [CR]). In this case, the one DOF model error represented the uncertain orientation of the environment.

Of course, this is only a start. From the standpoint of developing theoretical, "exact" algorithms, we have only addressed the problem of planning a restricted class of guaranteed strategies in the plane. It remains to consider exact algorithms in higher-dimensional configuration spaces, model error, Error Detection and Recovery, and more sophisticated termination conditions. For more on work in this direction, see [D3].

9 References

[BK] Bajaj, C. and M. Kim, "Compliant Motion Planning with Geometric Models", *Proc. ACM Symposium on Computational Geometry*, Waterloo, 1987.

[BLP] Brooks, R., and T Lozano-Pérez, "A Subdivision Algorithm in Configuration Space for Findpath with Rotation", *Eighth International Joint Conference on Artificial Intelligence*, Karlsruhe, Germany, August, 1983.

[Bra] Brady, M. et. al. (eds). *Robot Motion: Planning and Control.*, Cambridge, Mass.: MIT Press. (1982).

[Bro] Brost, R., "Automatic Grasp Planning in the Presence of Uncertainty", *IEEE ICRA*, San Francisco, April, 1986.

[Buc] **Buckley, S. J.** *Planning and Teaching Compliant Motion Strategies,* Ph.D. Thesis. Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, 1987. Also MIT-AI-TR-936 (1987).

[BRS] **Burridge, R., Rajan, V. T., and Schwartz, J. T.** *The Peg-In-Hole Problem: Statics and Dynamics of Nearly Rigid Bodies in Frictional Contact,* IEEE ICRA, Raleigh, NC (1983).

[Ca] **Canny, J.F.** *A New Algebraic method for Robot Motion Planning and Real Geometry,* FOCS (1987).

[CD] **Canny, J.F. and Donald, B. R.** *Simplified Voronoi Diagrams,* Proc. ACM Symposium on Computational Geometry, Waterloo, June (1987).

[CD2] **Canny, J.F. and Donald, B. R.** *Simplified Voronoi Diagrams,* Discrete and Computational Geometry (In press).

[CR] **Canny, J., and J. Reif,** "New Lower Bound Techniques for Robot Motion Planning Problems", *FOCS* (1987).

[Don] **Donald, B. R.** *A Search Algorithm for Motion Planning with Six Degrees of Freedom,* Artificial Intelligence, 31 (**3**) (1987a).

[D] **Donald, B. R.** *Robot Motion Planning with Uncertainty in the Geometric Models of the Robot and Environment: A Formal Framework for Error Detection and Recovery,* IEEE International Conference on Robotics and Automation, San Francisco, April (1986a).

[D2] **Donald, B. R.** *A Theory of Error Detection and Recovery for Robot Motion Planning with Uncertainty,* Intl. Workshop on Geometric Reasoning, Oxford, England (1986b).

[D3] **Donald, B. R.** *Error Detection and Recovery for Robot Motion Planning with Uncertainty,* MIT Artificial Intelligence Laboratory, MIT-AI-TR 982 (1987b).

[D4] **Donald, B. R.** *A Geometric Approach to Error Detection and Recovery for Robot Motion Planning with Uncertainty,* To appear in *Artificial Intelligence* (1988).

[D5] **Donald, B. R.** *Forthcoming.,* (1988).

[D6] **Donald, B. R.** *Towards Task-Level Robot Programming,* Cornell Computer Science Department Technical Report (1987).

[E] **Erdmann, M.** *Using Backprojections for Fine Motion Planning with Uncertainty,* IJRR Vol. 5 no. 1 (1986).

[EM] **Erdmann, M., and M. Mason,** "An Exploration of Sensorless Manipulation", *IEEE International Conference on Robotics and Automation,* San Francisco, April, 1986.

[Gri] **Grigoryev D. Y.,** "Complexity of Deciding Tarski Algebra" Jour. Symbolic Computation, special issue on decision algorithms for the theory of real closed fields, to appear (1987).

[HW] **Hopcroft J., and Wilfong G.,** "Motion of Objects in Contact," Int. Jour. Robotics Res. vol 4, no. 4, (1986).

[Kod] **Koditschek, D.,** "Exact Robot Navigation by Means of Potential Functions: Some Topological Considerations", Proc. IEEE Intl. Conf. Robotics, Raleigh, March 1987.

[Kou] **Koutsou, A.,** "A Geometric Reasoning System for Moving an Object While Maintaining Contact with Others", *ACM Symposium on Computational Geometry,* Yorktown Heights, N.Y., 1985.

[LP] **Lozano–Pérez, T.** *Spatial Planning: A Configuration Space Approach,* IEEE Trans. on Computers (C–32):108–120 (1983a).

[LP2] **Lozano-Pérez, T.**, "Robot Programming", *IEEE Proceedings*, 1983b.

[LMT] **Lozano-Pérez, T., Mason, M. T., and Taylor, R. H.** *Automatic Synthesis of Fine-Motion Strategies for Robots,* Int. J. of Robotics Research, Vol 3, no. 1 (1984).

[Lum] **Lumelsky, V. J.** *Continuous Motion Planning in Unknown Environment for a 3D Cartesian Robot Arm. , Proceedings of the 1986 IEEE International Conference on Robotics and Automation*, pp. 1569–1574. (April 7–10, San Francisco, Calif.) (1986).

[Ma] **Mason, M.T.** *Compliance and force control for computer controlled manipulators,* IEEE Trans. on Systems, Man and Cybernetics (SMC-11):418-432 (1981).

[Ma2] **Mason, M. T.** *Automatic Planning of Fine Motions: Correctness and Completeness,* 1984 IEEE International Conference on Robotics, Atlanta Ga. (1984).

[Nat] **Natarajan, B. K.** 1986. On Moving and Orienting Objects. Ph.D. Thesis. Ithaca, N.Y.: Cornell University Department of Computer Science.

[Natl] **Natarajan, B. K.** 1986 (Oct. 27–29, Toronto, Ontario). An Algorithmic Approach to the Automated Design of Parts Orienters. *Proceedings of the 27th Annual IEEE Symposium on Foundations of Computer Science*, pp. 132–142.

[OSY1] **Ó'Dúnlaing, C., Sharir, M., and Yap C.**, "Generalized Voronoi diagrams for moving a ladder: I Topological Analysis," NYU-Courant Institute, Robotics Lab. Tech. report No. 32 (1984)

[OSY2] **Ó'Dúnlaing, C., Sharir, M., and Yap C.**, "Generalized Voronoi diagrams for moving a ladder: II Efficient construction of the diagram," NYU-Courant Institute, Robotics Lab. Tech. report No. 33 (1984)

[OY] **Ó'Dúnlaing C., and Yap C.**, "A retraction method for planning the motion of a disc," J. Algorithms (6) (1985) pp. 104-111

[SHS] **Schwartz J., Hopcroft J., and Sharir M.**, "Planning, Geometry and Complexity of Robot Motion Planning", Albex Publishing Co., New Jersey, (1987).

[SY] **Schwartz J. and Yap C. K.**, "Advances in Robotics," Lawrence Erlbaum associates, Hillside New Jersey, (1986).

[T] **Taylor, R. H.** *The Synthesis of Manipulator Control Programs from Task-level Specifications,* Stanford Artificial Intelligence Laboratory, AIM-282, July (1976).

[Y] **Yap, C.**, "Coordinating the motion of several discs," NYU-Courant Institute, Robotics Lab. No. 16 (1984)

[Y2] **Yap, C.**, "Algorithmic Motion Planning", in *Advances in Robotics: Volume 1*, edited by J. Schwartz and C. Yap, Lawrence Erlbaum Associates, 1986.

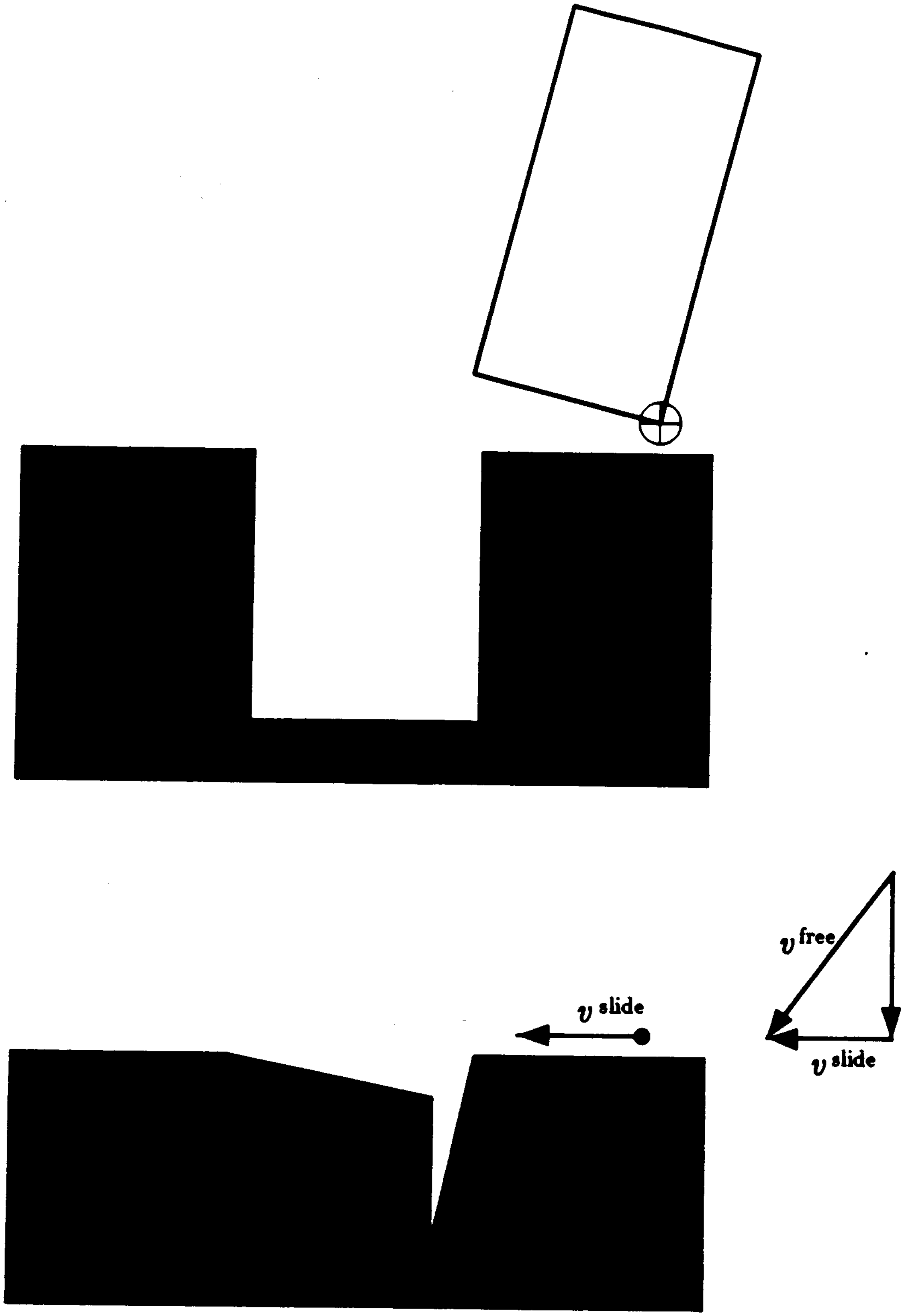

figure 1.

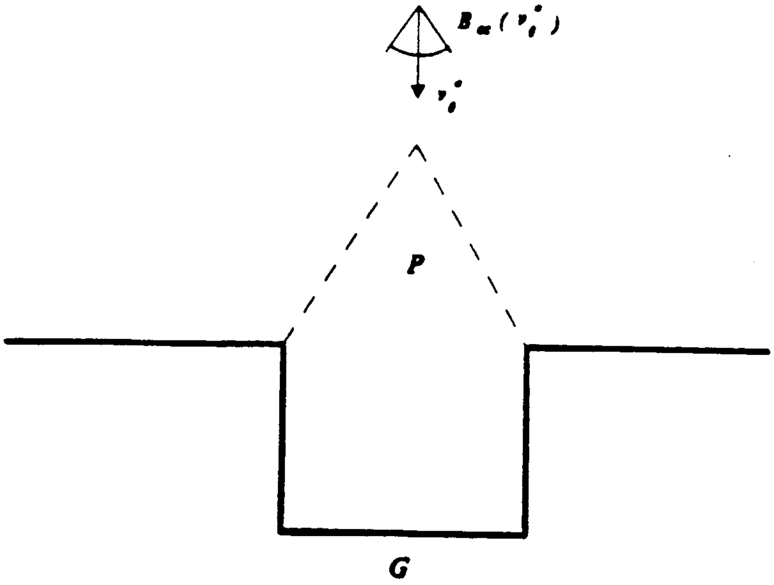

Fig. . The goal is the region G. Sliding occurs on vertical surfaces, and sticking on horizontal ones. The commanded velocity is v_θ^*, and the control uncertainty is $B_{ec}(v_\theta^*)$. The *preimage* of the G with respect to θ is the region P.

Fig 2

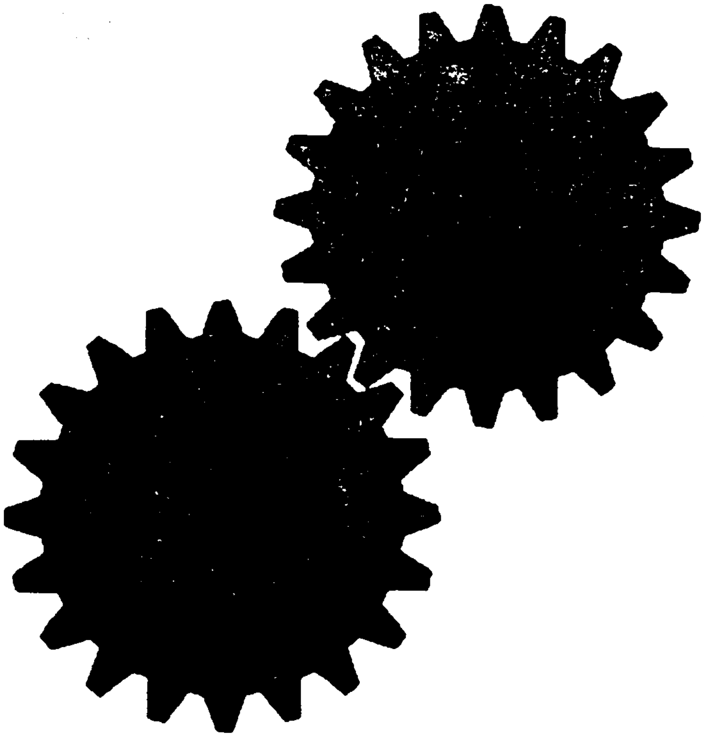

Fig. . Geometric models of two gear-like planar objects A and B. A is grasped and can translate but not rotate. The orientation of B is unknown. The task is to generate a motion strategy to mesh the gears.

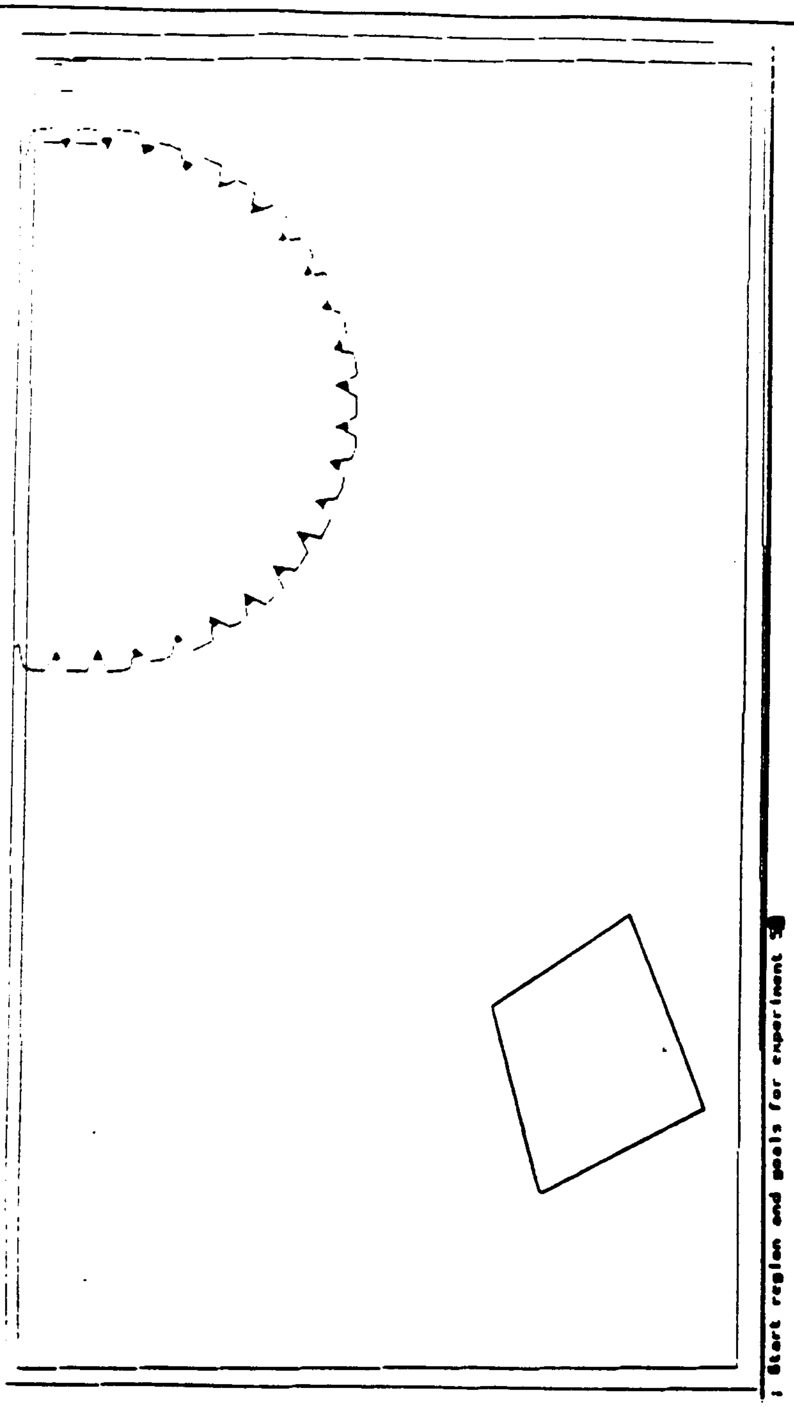

Fig. . The configuration space for the gear example (fig. 2) at one α-slice ($\alpha = 0$) of $\mathcal{G}$. The goal region is the "valleys" of the cspace obstacle. The start region is the diamond to the lower left. B is not allowed to rotate, so no motion across J is possible.

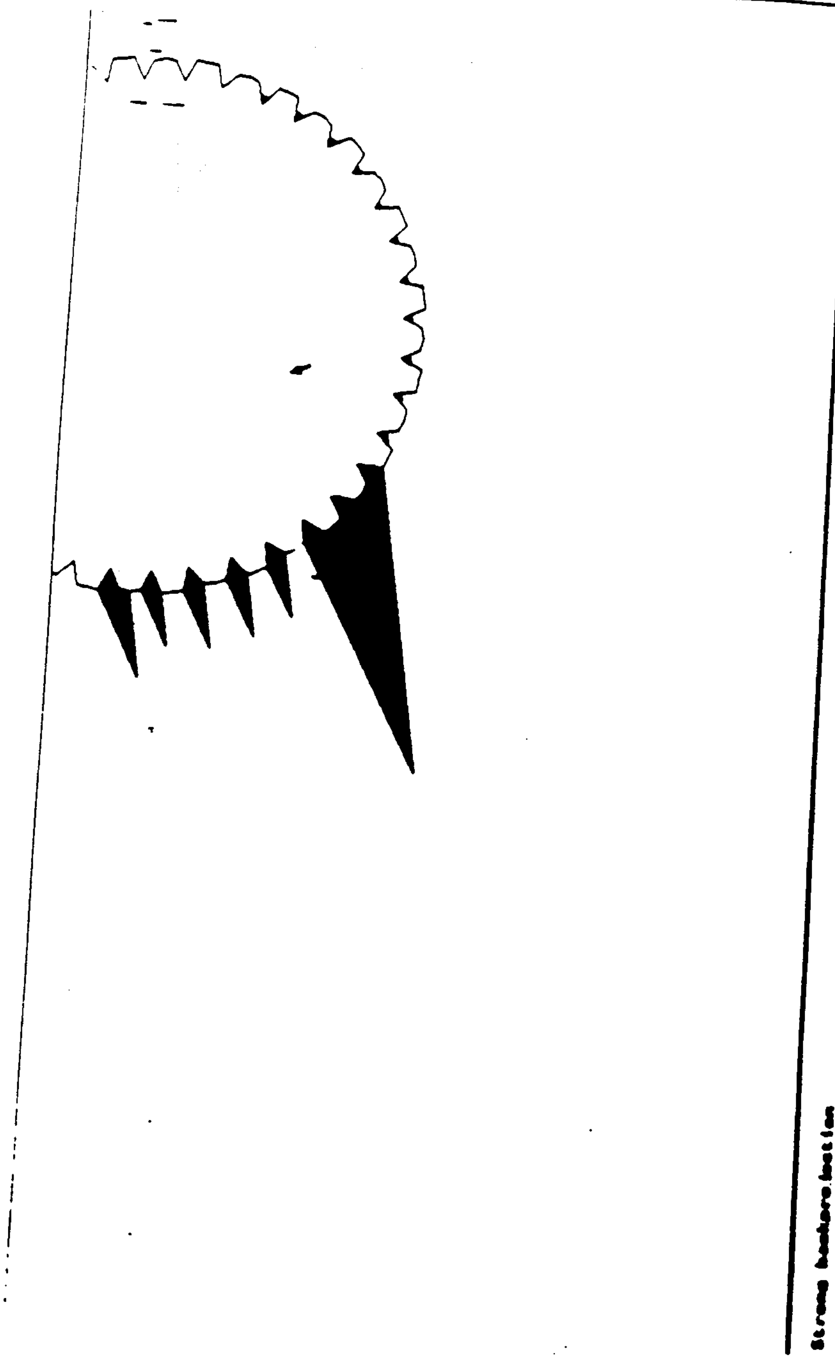

Fig. . The backprojection in slice $\alpha = 0$ of the goals in fig. , assuming that B cannot rotate. In all these experiments, the coefficient of friction is taken to be .25.

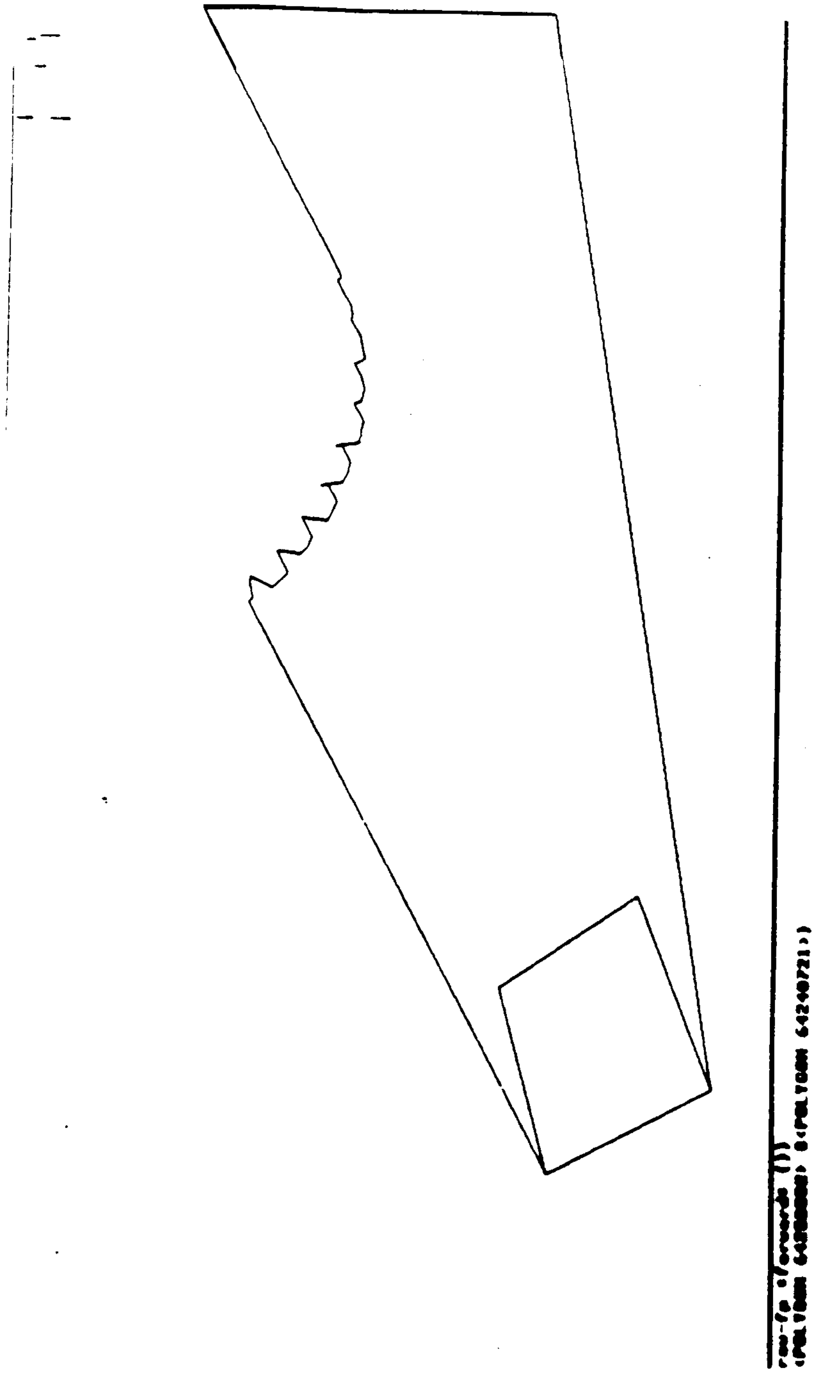

Fig. The forward projection of the start region in slice $\alpha = 0$. Note the degenerate edges due to sliding.