Planning Sequences of Squeeze-Grasps to Orient and Grasp Polygonal Objects

Matthew T. Mason Kenneth Y. Goldberg
Russell H. Taylor*

April 15, 1988
CMU–CS–88–127 (3)

*IBM T.J. Watson Research Center
Yorktown Heights, NY 10598, USA

Abstract

This paper describes the mechanics and automatic planning of a class of shape-orienting operations for a robotic manipulator. Using a parallel-jaw gripper, the robot repeatedly squeezes the object, rotating and moving the hand according to plan. Except for an irreducible 180 degree ambiguity, some polygonal shapes can be stably grasped in a completely determined orientation with just two squeezes, without sensing. The mechanics of the operation lead to a natural discretization of the continuum of action choices, so that straightforward graph-searching can be applied to generate the plan.

Submitted to Ro.man.sy '88
Seventh CISM-IFToMM Symposium on
Theory and Practice of Robots and Manipulators
September, 1988
Udine, Italy

Contents

List of Figures

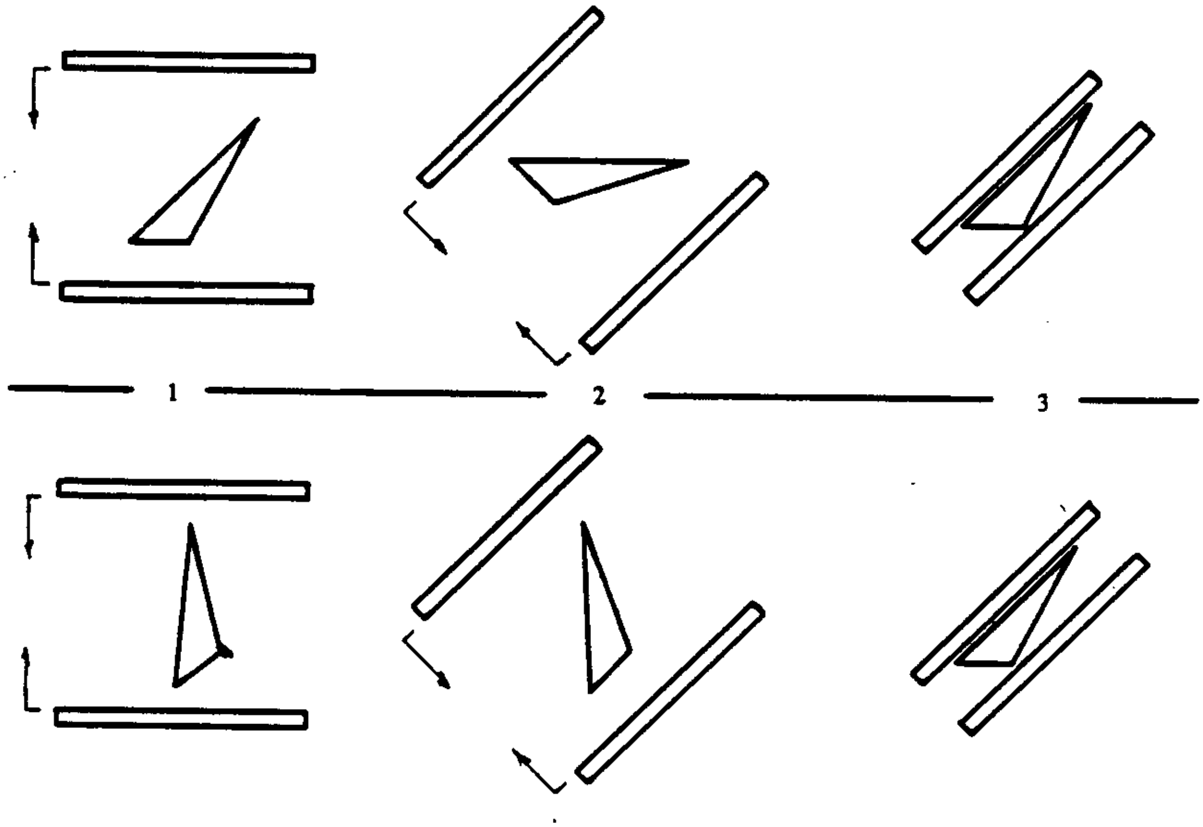

Figure 1: An automatically-generated squeeze plan, that orients a triangle in two squeezes.

1. Introduction

This paper describes part of an ongoing inquiry into the problems of autonomous robotic manipulation. We seek a systematic understanding of perception, planning, and plan execution, in tasks that include uncertainty. We have focussed on the mechanics of manipulator operations, and how knowledge of the mechanics is applied in automatic planning. Our approach is to study very simple, but well-defined, task domains, and to design a robot planner that is competent in the task domain. The task domain itself may be of some interest, but our main hope is that the principles of autonomous manipulation that apply in this narrow domain will also be important in a more general class of tasks.

The task domain considered here is the problem of orienting planar, polygonal, objects, by squeezing them with a parallel-jaw gripper. Each squeeze has the potential of aligning an edge of the object with the face of one of the fingers. A carefully designed sequence of squeezes can completely orient an object, except for an irreducible 180 degree ambiguity. The method is an extension of the results of Brost (1988), who developed the approach in the context of a single squeezing operation. In this paper we will consider sequences of one or more squeezes. An earlier paper (Taylor, Mason, and Goldberg 1987) describes the overall approach, focussing on the integration of sensory information during execution of the plan. This paper is also closely related to work by Erdmann and Mason (1986), and Peshkin and Sanderson (1987).

2. Overview of system

Figure 1 shows a typical squeezing sequence. A small triangle is to be oriented and grasped in the fingers. After the first squeeze, the triangle is either aligned with one of the fingers, or it is cocked. After the second squeeze, the triangle will be aligned with one of the fingers. The remaining 180 degree ambiguity cannot be removed, due to symmetry in the operation. The system described in this paper is capable of constructing plans, such as the plan shown in Figure 1, automatically. The basic elements of the approach are:

- We assume planar motion of rigid polygonal objects, subject to Coulomb friction, with negligible inertial force. Contacts occur with the finger faces, not the corners.
- Each finger performs a uniform translation until a large resisting force is encountered. The fingers' motions are symmetric with respect to a line centered between the two fingers. These assumptions are consistent with a parallel-jaw pneumatic gripper, where the robot arm translates the hand in a direction tangential to the finger faces, while the fingers close.
- The planner is given the following information:
  - a description of the object's shape;
  - the object's center of gravity;
  - the coefficient of friction;
  - the *a priori* possible object orientations; and
  - the goal.
- If the planner converges, it returns a sequence of squeeze operations that will achieve the goal, subject to the assumptions above.

So far, the planner has produced sensible plans for a few, very simple, objects. The planner's search is (almost) exhaustive, so that in principle, if a correct plan exists, the planner should (almost) never fail to converge. However, our present implementation has a number of difficulties with more complicated objects, discussed at the end of the paper. The next sections describe the mechanics of squeezing, and the structure of the planner.

3. How squeezing works.

This section recapitulates Brost's (1988) analysis of squeezing and extends it to the problem of sequential squeezing. We begin by analyzing *pushing* and *pure squeezing*, before turning to more complex squeeze operations.

First, consider the problem of a single finger pushing an object. We can use a result from (Mason 1986) to predict the object's rotation. At the contact point we construct three rays (Figure 2): two rays are the edges of the friction cone—showing the limits of the contact force allowed by Coulomb's law. The third ray shows the direction of the finger's motion. We also plot the object's center of gravity. These three rays vote to determine the object's rotation—if the majority pass the center of gravity on the left, the object rotates clockwise. If the majority pass on the right, the object rotates counter-clockwise.

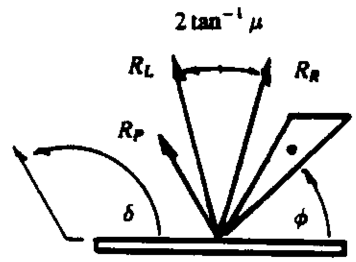

Figure 2: Given the contact geometry, coefficient of friction μ, and center of gravity, the object's rotation direction is determined by the votes of three rays: R_P, R_L, and R_R. (Mason 1986)

If the vote is a tie—one ray on the left, one on the right, and one straight through—the object makes a pure translation.

For a fixed object shape, these predictions can be economically expressed in a *pushing diagram* (Figure 3a). Let ϕ be the initial object orientation, let δ be the direction of motion of the finger, with both angles measured relative to the finger face (Figure 2). Let $|d\phi|$ be the direction of object rotation, i.e. either plus, minus, or zero. The pushing diagram plots rotation direction as a function $|d\phi|(\phi, \delta)$ of object orientation and finger motion. The details of the construction are contained in (Brost 1988), but note that the rotation direction $|d\phi|$ is piecewise constant. Hence, the diagram is divided into regions of constant rotation direction. Each boundary represents a discontinuous change in rotation direction, where one of the three rays of Figure 2 changes its vote. Since there are two fingers, we will need two different pushing diagrams, representing two different functions: $|d\phi|_1(\phi, \delta)$ for finger 1 (Figure 3a), and $|d\phi|_2(\phi, \delta)$ for finger 2 (Figure 3b). The symmetry in the finger's motions gives the following relation:

$$|d\phi|_2(\phi, \delta) = |d\phi|_1(\phi + 180, 180 - \delta).$$

The symmetry can be observed by comparing Figures 3a and 3b.

Now, consider pure squeezing, where the object is definitely in contact with both fingers. Determining the rotation of the object, or whether the object is stable, is purely a geometrical problem. Again, the direction of object rotation can be expressed as a function $|d\phi|_s(\phi, \delta)$ (Figure 3c) although there is no real dependence on finger motion direction δ. Here we have added a subscript s to indicate that this is the function for squeezing. Again, see (Brost 1988) for details of the construction.

Now we are prepared to analyze a complex squeeze operation. At any point in time, any of three sub-operations may be active: finger 1 pushing, finger 2 pushing, or pure squeezing. Without knowing which sub-operation is active at any particular point in time, the robot must include the predictions of all three operations. Hence we define

$$|d\Phi|(\phi, \delta) = \{|d\phi|_1(\phi, \delta),\ |d\phi|_2(\phi, \delta),\ |d\phi|_s(\phi, \delta)\}$$

Figure 3d is a typical diagram, showing all possible instantaneous motions of the object, as a function of (ϕ, δ).

Given the feasible instantaneous motions, we can determine the feasible *finite* motions $\Delta\Phi(\phi, \delta)$ by taking the *transitive closure* of the instantaneous motions (Figure 3e). The resulting diagram shows, for a given initial (ϕ, δ), the set of all possible changes in ϕ. To complete analysis of squeezing, we want to identify those ϕ that can be obtained when the operation is completed. This imposes a restriction that

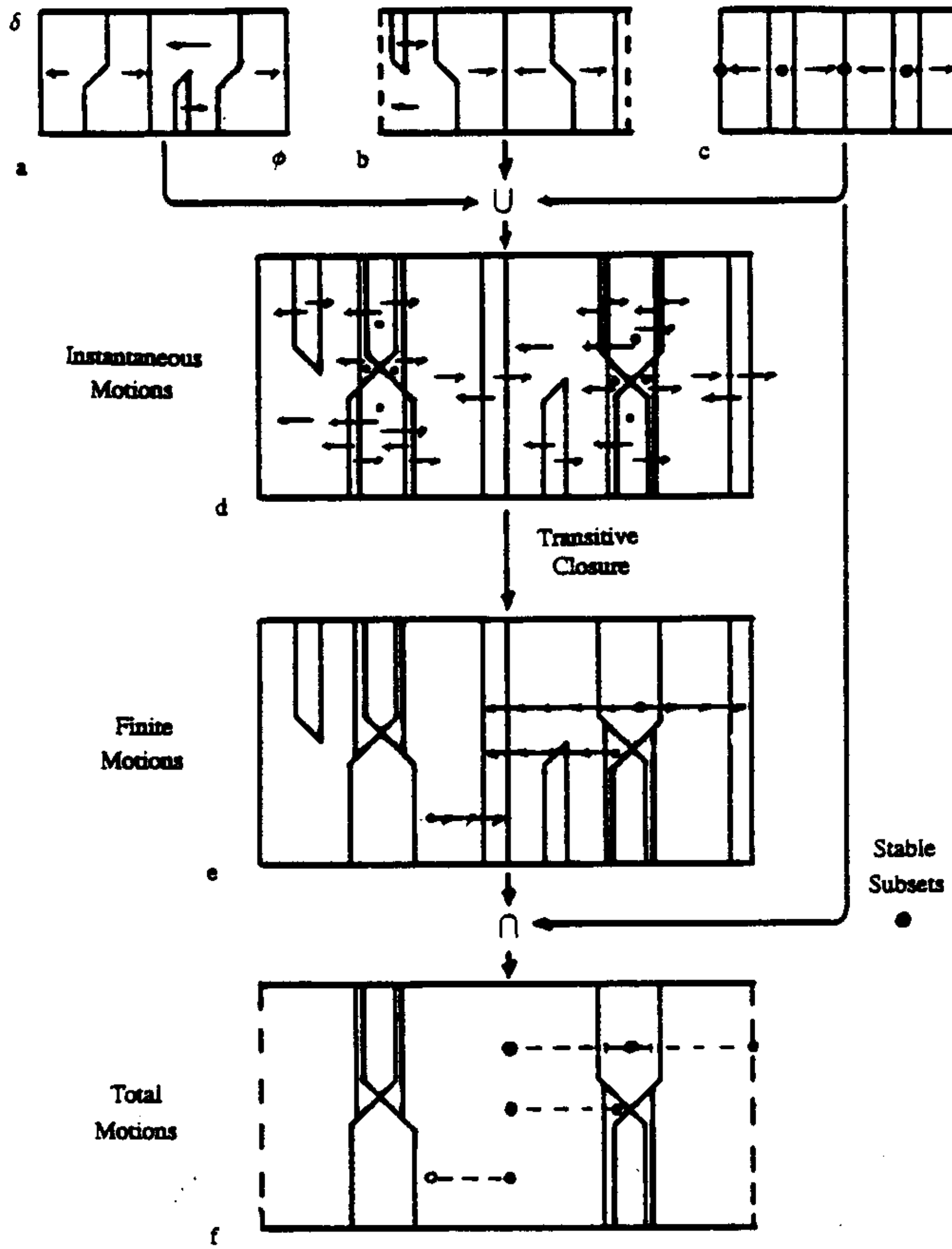

Figure 3: Automatic planning relies on analysis of the squeeze operation, shown here. Each diagram plots some motion feature against object orientation ϕ and finger motion direction δ. (a,b) Each pushing diagram shows the instantaneous rotation direction $|d\phi|_i(\phi,\delta)$, for finger i. (c) The squeezing diagram $|d\phi|_s(\phi,\delta)$ gives the direction of rotation for pure squeezing. (d) The set of possible instantaneous motions $|d\Phi|(\phi,\delta)$. (e) The set of feasible *finite* motions $\Delta\Phi(\phi,\delta)$, shows the orientations that might occur during the motion, for a few typical points. (f) The set of *total* motions $\hat{\Phi}(\phi,\delta)$, is the set of possible object orientations when the operation ends, shown for a few typical points.

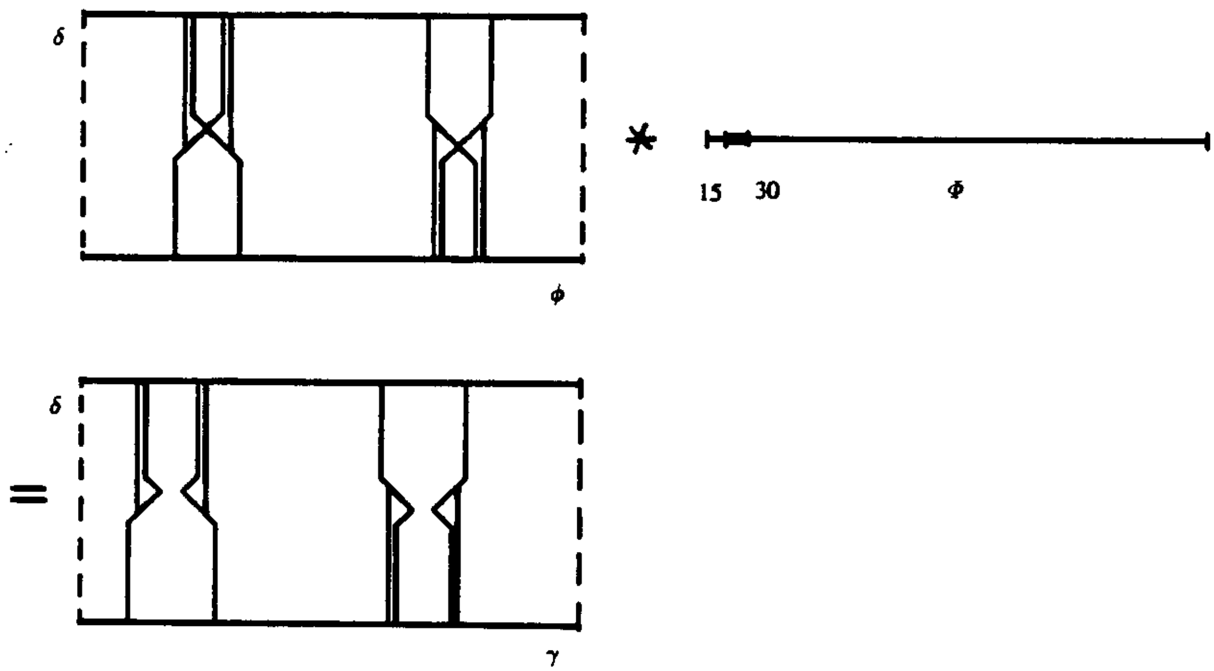

Figure 4: The total motion diagram, starting from an initial Φ of $[15, 30]$. This gives the set of *a posteriori* orientations, given the specified *a priori* orientations, as a function of hand orientation and finger motion direction.

the object attitude be stable in the true-squeezing instantaneous motion diagram (Figure 3c). So we take the finite-motion diagram, and intersect with the stable ϕ in the the true-squeezing instantaneous motion diagram, to obtain a *total motion* function $\Phi'(\phi, \delta)$. A typical total motion diagram is shown in Figure 3f, that shows the possible final ϕ resulting from an initial (ϕ, δ).

A squeeze operation consists of a rotation of the fingers by some angle $-\gamma$, and then a squeezing with finger motion δ. If an object is initially at orientation ϕ, then after the operation the orientation will be in the set $\Phi'(\phi + \gamma, \delta)$. If the orientation is initially ambiguous, i.e. if we know only that $\phi \in \Phi$, then after the operation the orientation is given by the function

$$\Phi'(\gamma, \delta) = \bigcup_{\phi \in \Phi} \Phi'(\phi + \gamma, \delta)$$

In effect, the total motion diagram is logically convolved with the set of initial orientations Φ. For example, suppose the object's initial orientation is in a small interval $\phi \in [15, 30]$. To compute $\Phi'(\gamma, \delta)$, we take the diagram of Figure 3f, shift it by fifteen degrees, and "smear" it another fifteen degrees, giving Figure 4.

4. How the planner works.

The planner works by searching a graph. Each node in the graph is a set of possible object orientations Φ. Each arc in the graph is a squeezing operation (γ, δ). The planner begins with a root node, for which the object orientation is typically unconstrained, $\Phi = [0, 360)$. The planner determines a finite set of

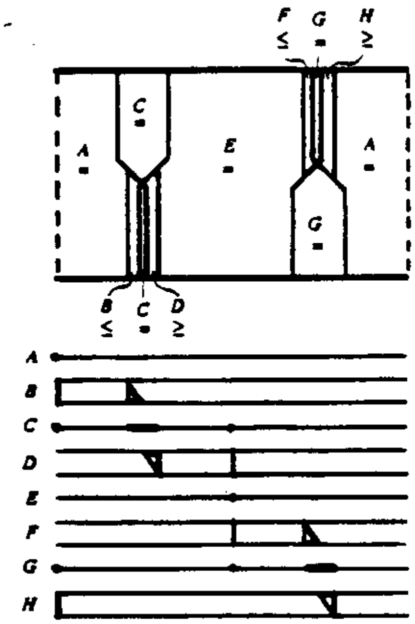

Figure 5: An actual plot of Φ', showing equivalent and ordered regions. Out of the entire two-dimensional continuum of possible actions, only two distinct effects can be obtained.

potentially useful squeeze motions, constructing an arc (γ, δ) and a successor node Φ' for each squeeze. By repeating this step for new nodes, the planner gradually explores the set of all possible squeeze sequences. During this construction, if a node is constructed satisfying the goal, the plan is complete.

The key step is the problem of "expanding" a node—identifying a finite number of arcs from a given node, and determining the corresponding sets Φ'. The planner constructs a total motion diagram $\Phi'(\phi, \delta)$, and constructs an arc for each different region in the diagram. The key problem is identifying a *finite* number of useful squeezes. First, consider the conditions under which a squeeze may be pruned from the search. Imagine a situation slightly simpler than squeeze operations—where there is a single control variable u and a single state variable x. The initial set of states X is given, and we are constructing the resultant set of states X', as a function of u. First, consider two different actions u_1 and u_2. Obviously, if the two actions are equivalent, i.e. if $X'(u_1) = X'(u_2)$, we need consider only one of the actions during our search. The arc corresponding to u_2 may be pruned. Less obviously, it is also okay to prune u_2 if u_1 *dominates* u_2, i.e. if $X'(u_1) \subset X'(u_2)$. The argument goes as follows: suppose we have a correct plan that employs u_2, we argue that the same plan will work, with u_1 substituted for u_2. Hence pruning u_2 will not prevent convergence of the search. (Some subtle problems with this argument are discussed in (Taylor, Mason, and Goldberg 1987).)

The real value of this observation is that it often allows us to prune away an entire continuum of actions. Consider Figure 5, showing Φ', for each region in the total motions diagram. In region A, an *equivalent* region, every choice of squeezing motion (ϕ, δ) gives $\Phi' = \{0\}$. In region B, an *ordered* region, Φ' varies continuously, but there is a single choice of (ϕ, δ) that dominates the other choices. Every region in the diagram is a two-dimensional continuum that can be represented as a single action. In some instances, though, an *unordered* region can occur, with no dominant action, and no justification for pruning away any of the motions.

It is now possible to describe in some detail how the planner works:

1. The planner is given the object shape, center of gravity, coefficient of friction, initial set of object orientations, and a goal predicate. Typically, the initial set is $[0, 360)$ and the goal is to orient the object completely, modulo the unavoidable 180 degree ambiguity.

2. For efficiency, the planner first computes a *kernel* motion diagram $\Phi'(\phi, \delta)$, for a completely oriented object. The diagram in Figure 3f is the kernel for the small triangle. This kernel is saved for use in constructing motion diagrams corresponding to different nodes in the search. Analysis of the kernel also allows the planner to construct a bit-vector representation of object orientation sets Φ. After this construction, all operations on sets of orientations can use discrete representations and bit-wise logical operations.

3. At this point, an exhaustive breadth-first search begins. For each node, we construct a motion diagram, by convolution of the kernel diagram with the node's set of orientations Φ. In the kernel diagram, equivalent regions are not distinguished from ordered regions. This means that in the convolved diagram, we cannot distinguish unordered regions. The result is that equivalent, ordered, and unordered regions are all treated as equivalent regions. The resulting branching factor is always finite, but the search is not truly exhaustive, as the unordered regions are, in effect, being sampled.

4. As each new set of orientations Φ' is constructed, it is compared with previously constructed Φ', and all supersets are pruned from the search. (We should also prune any set that can, by any rotation, be transformed into a superset of a previously generated set.)

5. As each new node is encountered, the goal predicate is checked. If it is satisfied, the search is complete, and the path from the root to the new node encodes the shortest sequence of squeezes that attains the goal.

5. Experimental Results

Constructing the total motion diagram for each node is time-consuming, and subject to a variety of numerical problems. Consequently, we have successfully planned sequences of squeezes for only three objects: a triangle, a skinny rectangle, and a fat rectangle. When we tried a simple five-sided shape, (Brost's *roly-pointy*), the program failed to produce a plan in 48 hours of elapsed time on a Symbolics 3600 Lisp Machine. The search fails due to a subtle problem in our treatment of unordered regions. During construction of a diagram, a region is defined by a collection of contiguous areas that have the same effect. Since the subsets of an unordered regions do not have the same effect, they are not identified as defining a single region. Hence, where one or two unordered regions should be created, our implementation constructs, in the case of the roly-pointy, 91 different regions. Each region is very small, and all 91 regions have nearly identical effects. With our next implementation we hope to correct this problem and explore the performance of the system on a wider variety of parts.

6. Conclusion

We have designed an automatic planning system for what seems, *a priori*, to be a relatively simple class of problems. What we've learned is that even for a simple class of problems, modelling the mechanics and constructing a planner is a moderately complex project. We have also learned that going the next step, reasoning about continuous sets of action primitives, is important. Nonetheless, our preliminary experiments are suggestive that the overall approach is sound.

Acknowledgements

Our results depend heavily on the work of Randy Brost, who generously allowed us to use his code. We are also grateful to Rod Brooks for a geometric modeling system. Mason and Goldberg were supported by a grant from the National Science Foundation., Grant Number DMC-8520475.

References

R.C. Brost. 1988. Automatic grasp planning in the presence of uncertainty. *Int J Robotics Research*, v7 n1, pp. 3–17.

M.A. Erdmann and M.T. Mason. 1986. An exploration of sensorless manipulation. *Proceedings, 1986 IEEE Int Conf on Robotics and Automation*, San Francisco.

M.T. Mason. 1986. Mechanics and planning of manipulator pushing operations. *Int J Robotics Research*, v5 n3, pp. 53–71.

M.A. Peshkin and A.C. Sanderson. 1987. Planning robotic manipulation strategies for sliding objects. *Proceedings, 1987 IEEE Int Conf on Robotics and Automation*, Raleigh, NC.

R.H. Taylor, M.T. Mason, and K.Y. Goldberg. 1987. Sensor-based manipulation planning as a game with nature. *Fourth International Symposium on Robotics Research*. Santa Cruz, CA.