Towards Task-Level Robot Programming

Bruce Donald

87-878

November 1987

Department of Computer Science
Cornell University
Ithaca, New York 14853-7501

TOWARDS TASK-LEVEL ROBOT PROGRAMMING

Bruce R. Donald
Computer Science Department
Cornell University

Towards Task-Level Robot Programming

Bruce R. Donald
Cornell University

Abstract: This paper summarizes our research efforts in robotics. The primary goal is to bring robotics science closer to its goal of task-level planning. We approach this goal through blend of theory, implementation, and experimentation. We have identified several key problem areas on which to concentrate, and are undertaking research on generating assembly strategies from task-level descriptions. Our projects focus on:

- Algorithmic techniques for modelling geometric constraints, and the planning of robot motions using these constraints.
- The synthesis of compliant motion programs using the geometric constraints imposed by the task.
- The explicit modelling of uncertainty and error in sensing, control, and the shape of the manipulated parts.
- The development of a precise theory of Error Diagnosis and Recovery, and a method for generating sensor-based plans with built-in error detection and recovery.

The heart of this research lies in a geometrical planning theory, for reasoning about and manipulating physical systems. We will develop this underlying theory, implement and test the theory in simulation, and conduct robotics laboratory experiments.

Contents

Towards Task-Level Robot Programming

Bruce R. Donald
Cornell University

1 Introduction

The possibilities engendered by the concept of "robot" are almost limitless. However, unless major theoretical advances are achieved to make robots more useful, flexible, and practical these visions will never become realities. If the present course is maintained, robots will continue to be utilized far below their capacity, limited to such tasks as spray painting, spot welding, and palletizing.

Our research in robotics focuses on more than the mechanical aspects of the field. We view robotics as the science of representing, reasoning about, and manipulating physical objects. We believe that basic research in robotics science can infuse new techniques into mainstream robotics, and greatly enhance the usefulness and practicality of robots.

The primary goal of our research is task-level planning. We approach this goal by utilizing a blend of theory, implementation, and experimentation. In our research on generating assembly strategies from task-level descriptions, we have identified several problem areas on which to concentrate: these include:

1. Algorithmic techniques for modelling geometric constraints, and the planning of robot motions using these constraints.

2. The synthesis of compliant motion programs using the geometric constraints imposed by the task.

3. The explicit modelling of uncertainty and error in sensing, control, and the shape of the manipulated parts.

4. The development of a precise theory of Error Diagnosis and Recovery, and a method for generating sensor-based plans with built-in error detection and recovery. We call this the *Error Detection and Recovery (EDR) Theory.*

In fact, we regard these goals as hierarchical, in that (1–3) are essential for the solution of (4).

2 Research Projects

2.1 Modelling and Planning with Geometric Constraints

In [Donald 84,85,87a] we designed and implemented a motion planning algorithm for a six degree of freedom (DOF) robot. While at the time there were theoretical algorithms for this problem, this was the first program that could solve planning problems with more than two or three degees of freedom in environments of complexity comparable to real-world situations. The algorithm required developing a number of new theoretical tools in computational robotics; these tools have had an impact in motion-planning [Lozano-Pérez 85, 86], qualitative reasoning [Faltings 86, 87], and have been used for path-planning in a hand-eye system developed at MIT [Lozano-Pérez *et al.*, 87]. The implemented planner was also of practical interest: for example, it provided a provably good approximate algorithm directly applicable to navigation planning for a spacecraft with six degrees of freedom.

This research led to joint theoretical work [CD] with John Canny of MIT and Berkeley: Using tools from differential topology and homotopy theory, we demonstrated theoretical results on "simplified" Voronoi diagrams (of low algebraic degree) for motion planning, and the first implementation of a Voronoi-based motion-planning algorithm. After obtaining combinatorial bounds on the size of the diagram, we applied Canny's results on stratifications [C1] to show that in addition to having lower algebraic complexity, the geometric complexity of the algorithm exhibited an improvement of two orders of magnitude in the exponent [CD]. For example, the worst-case performance in the six-degree of freedom case is $O(n^7 \log n)$ (within a quadratic factor of optimal) whereas the best previous bound was $O(n^{4096})$. The new algorithm has actually been implemented[1] and appears quite fast in practice. The motivation behind this work is that, as observed by [Ó'Dúnlaing and Yap 82; Brooks 83; Ó'Dúnlaing, Sharir, and Yap 84; Yap 84][2] the paths produced by the algorithm lie along the diagram, and hence are "robust" in the sense that they are maximally clear of obstacles, and therefore could be executed reliably even given some uncertainty in sensing and control.

2.2 Uncertainty and Compliant Motion Planning

The Voronoi research was concomitant with our growing interest in robot motion planning with uncertainty. Robots are subject to the following kinds of uncertainty:

1. Inaccuracy and errors in sensing,

2. Inaccuracy and errors in control,

3. Uncertainty about the geometry of the environment.

The last (3) is called "model error", and had received little previous attention. Model error arises because, in general, a robot can have only approximate knowledge of the shape and position of objects in the environment.

We now ask the question:

- How can robots plan and execute tasks (for example, a mechanical assembly using compliant motion) in the presence of these three kinds of uncertainty?

[1] Although only for two- and three-degree of freedom systems.

[2] See the textbook of [Schwartz and Yap 86] for an introduction and review of the use of Voronoi diagrams in motion planning.

This is perhaps the most fundamental problem in robotics today.

Work on compliant motion[3] 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], [Erdmann], and [Donald 86a;b,87b,88]. This research has led to a theoretical computational framework for motion planning with uncertainty, which we denote [LMT,E,D]

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 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. This research requires exploration of the *contact problem*, that is, the study and modelling of the motion of objects in contact. The contact problem is a rich area for research, and its solution will undoubtedly draw on techniques from mathematics, computational geometry, and engineering physics. At this point, we are developing a planning system, called LIMITED, for the domain of planar assemblies. The planner models contact using an algorithmic technique called *forward projection* [Erdmann]. This technique relies on a restricted dynamic model. In particular, it employs *quasi-static* assumptions, in which it is assumed that frictional forces dominate inertial forces. In parallel work by John Hopcroft, the simulation of kinematic chains under full dynamics, but without contact, has been explored. We hope to take advantage of the close alignment between our research interests and Professor Hopcroft's to explore contact

[3] See [Mason 81] for an introduction and survey; see also our review in sec. 7.2.

modelling and compliant motion planning issues under more sophisticated dynamic models.

2.3 Model Error

How can compliant motion strategies be synthesized in the presence of sensing, control, and geometric model error, such that the strategies are guaranteed to succeed so long as the errors lie within the specified bounds? As an example, consider a peg-in-hole assembly with sensing and control uncertainty, with toleranced parts. We wish to synthesize a compliant motion strategy that is guaranteed to succeed so long as the parts lie within the specified tolerances, and the sensing and control errors lie within the specified bounds.

We have begun an attack on this problem by introducing additional dimensions to the configuration space; each dimension represents a way in which the parts could parametrically vary. We termed the product space of the motion degrees of freedom and the geometric model variational dimensions "generalized configuration space" and showed how to compute "preimages" [LMT,E] of a geometrical goal in this generalized configuration space. The preimage of a goal is the set of (generalized) configurations from which a particular commanded compliant motion is guaranteed to succeed. Using this technique, we have developed a framework for computing motion strategies that are guaranteed to succeed in the presence of sensing, control, and geometric model uncertainty. The motion strategies comprise sensor-based gross motions, compliant motions, and simple pushing motions.

2.4 Error Detection and Recovery

It is not always possible to find plans that are guaranteed to succeed. For example, if tolerancing errors render an assembly infeasible, the plan executor should stop and signal failure. In such cases the insistence on guaranteed success is too restrictive. For this reason we wish to investigate *Error Detection and Recovery (EDR)* strategies. EDR plans will succeed or fail recognizably: in these more general strategies, there is no possibility that the plan will fail without the executor realizing it. The EDR framework fills a gap when guaranteed plans cannot be found or do not exist: it provides a technology for constructing plans that might work, but fail in a "reasonable" way when they cannot. We are developing both approximate algorithms and precise, combinatorial algorithms for EDR planning, implementing a planner, called LIMITED, and performing experiments to have LIMITED automatically synthesize EDR strategies in the domain of planar assemblies.

While EDR is largely motivated by the problems of uncertainty and model error, its applicability may be quite broad. EDR has been a persistent but ill-defined theme in both AI and robotics research. We are developing a constructive, geometric definition for EDR strategies and demonstrating how they can be computed. This theory represents what is perhaps the first systematic attack on the problem of error detection and recovery based on geometric and physical reasoning. Initial progress on the EDR theory has been reported in [Donald 86a;b,87b,88].

3 Examples of EDR Strategies

3.1 A Simple Example

Consider fig. 1, which depicts a peg in hole insertion task. One could imagine a manipulation strategy derived as follows: The initial plan is to move the peg straight down towards the bottom of the hole. However, due to uncertainty in the initial position of the peg, the insertion may fail because the peg contacts to the left or right of the hole. Either event might be regarded as an "error." The "recovery" action is to move to the right (if the peg contacted to the left) and to move to the left (if the peg contacted to the right). Thus a plan can be obtained by introducing sensing steps and conditional branches.

Suppose that this conditional plan can be guaranteed—that is, it is a complete manipulation strategy for this simple task. In this case, it seems strange to view the contact conditions as "errors." We do not regard these events as "errors." Our reasoning is that if they can be detected and planned for, then they are simply events in a guaranteed plan.

We are interested in a different class of "errors." Now suppose that there is uncertainty in the width of the hole. If the hole is too small, we will consider this an error, since it causes all plans to fail. Similarly, if some object blocks the hole, and cannot be pushed aside, this is also an error, since it makes the goal unreachable. If either error is possible, there exists no guaranteed plan, for there is no assurance that the task can be accomplished. Since no guaranteed plan can be found, we are left with the choice of giving up, or of considering a broader class of manipulation strategies: plans that might work, but fail in an "reasonable" way when they cannot. Specifically, we propose that EDR strategies should achieve the goal when it is recognizably reachable, and should signal failure when it is not. For example, an EDR strategy for the peg-in-hole problem with model error might attempt to achieve the insertion using compliant motion, but be prepared to recognize failure in case the hole is too small. We are implementing a planner, called LIMITED, which synthesizes such strategies. LIMITED is an implementation of the EDR theory in the restricted domain of planar assemblies.

3.2 Application: Planning Gear Meshing

We must stress that EDR is not limited to problems with model error. There are many applications in which the geometry of the environment is precisely known, but in which guaranteed plans cannot be found, or are very difficult to generate. We now describe such a situation.

An interesting application domain for EDR is gear meshing. It is an example where EDR is applicable even though the shape of the manipulated parts is precisely known. Let us consider a simplified instance of this problem. In fig. 2 there are two planar gear-like objects, A and B. The task is to plan a manipulation strategy which will mesh the gears. The state in which the gears are meshed is called the *goal.*

We will consider two variants of this problem. In the first, we assume that the manipulator has grasped A, and that neither A nor B can rotate. However, A can slide along the surfaces of B. In the second, B is free to rotate about its center, but this rotation can only be effected by pushing it with A. In both cases, the initial orientation of B is unknown. We regard A as the moving object and B as the environment; hence *even though the shape*

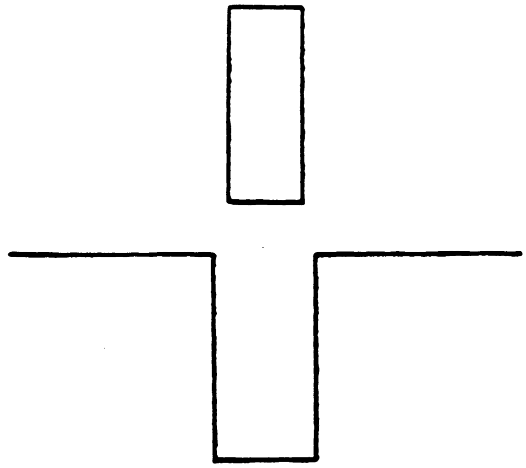

Figure 1: The goal is to insert the peg in the hole. No rotation of the peg is allowed. One can imagine a strategy which attempts to move straight down, but detects contact on the top surfaces of the hole if they occur. If the peg sticks on the top surfaces, the manipulator tries to move to the left or right to achieve the hole. Are these contact conditions "errors"? We maintain that they are not, since they can be planned for and verified.

of B is precisely known, we choose to view the uncertainty in B's orientation as a form of model error. In the first case, the system has only two degrees of motion freedom. In the second, there are three degrees of motion freedom, one of which is rotational, since B can be pushed. We distinguish between the rotation and non-rotation variants of the problem in order to highlight the additional techniques our planner employs when rotations are introduced.

In both variations, there is uncertainty in control, so when a motion direction is commanded, the actual trajectory followed is only approximately in that direction. There is also uncertainty in position sensing and force sensing, so that the true position and reaction forces are only known approximately. The magnitude of these uncertainties are represented by error balls.

In general, a commanded motion of A may cause A to move through free space, and contact B, possibly causing B to rotate. Our EDR theory is a technique for analyzing these outcomes geometrically to generate strategies that achieve the goal when it is recognizably reachable, and signal failure when it is not.

In an experiment, the EDR theory in the gear domain was applied using the planner,

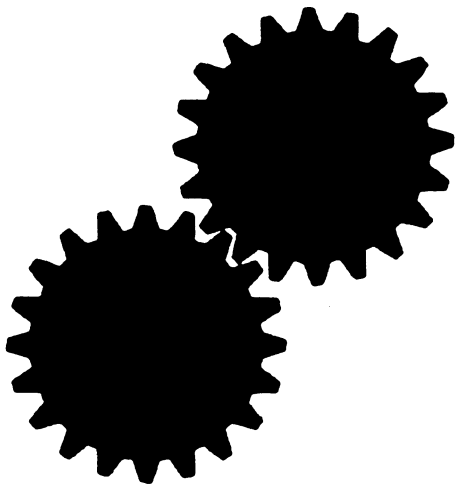

Figure 2: Geometric models of two gear-like planar objects A and B. A is grasped and can translate but not rotate. B can rotate about its center if pushed. The orientation of B is unknown. The task is to generate a motion strategy to mesh the gears.

LIMITED, as follows. Consider the problem of meshing two planar gears, under uncertainty as above. Suppose that gear B can rotate passively but has unknown initial orientation, as above. Suppose that A has been gripped by a robot. The initial position of A is uncertain. The robot can impart either pure forces (translations), or pure torques (rotations) to A. The planner can choose the direction of translation or rotation. Can a multi-step strategy of commanded translations and rotations be found to mesh the gears?

LIMITED was able to generate an EDR strategy for this problem. The characteristics of the experiment are:

- There are three degrees of motion freedom (two translational and one rotational) for A.
- There is one degree of rotational model error freedom (the orientation of B).
- It is possible to push B to change its orientation.
- There is sensing and control uncertainty.
- The geometry of the gears is complicated—they have many edges.

- Quasi-static analysis [Mason] is used to model the physics of interaction between the gears.

Thus we have a kind of four-degree of freedom planning problem with uncertainty and pushing. To generate multi-step EDR strategies under pushing, LIMITED employed the EDR theory together with a technique called *failure mode analysis.*

Now, there may exist a guaranteed strategy to mesh the gears. For example, experimental evidence suggests that for involute teeth gears, almost any meshing strategy will succeed. For other gear shapes perhaps some complicated translation while spinning A will always succeed. I don't know if there is such a guaranteed strategy for this case. It seems difficult for a planner to synthesize such guaranteed strategies, or even to verify them, if they exist at all.

A person might try to solve this problem with the following motion strategy:

- *Ram the gears together. See if they mesh.*

Or, somewhat more precisely,

- *Ram A into B. If they mesh, stop. If they jam, signal failure and try again.*

Probabilistically, this is a rather good strategy. It is certainly very simple, and probably easier to generate than a guaranteed strategy. If vision can be used to sense whether A and B are meshed, then it is an EDR strategy with just one step.

Suppose, however, that vision is poor, or that the gears are accessible to the robot gripper, but not to the camera. This means that position sensing will be very inaccurate, and hence may be of no use in determining whether the gears are successfully meshed. This will often be the case in practice. In this case, force sensing must be used to disambiguate the success of the motion (meshing) from failure (jamming in an unmeshed state). If the robot has force sensing, then it might use the following two-step EDR strategy:

- *Ram the gears together. Spin them to see whether they meshed.*

Or, again more precisely,

- *Ram A into B. Next, spin A. If A and B break contact, or if the gears* stick *(don't rotate), then signal failure. Otherwise, signal success.*

This strategy is essentially the one that LIMITED generates. The plan is

Motion 1: *Command a pure translation of A into B.*[4]

Terminate the motion based on force-sensing when sticking occurs (when there is no motion).

[4] LIMITED generates the actual force vector.

Motion 2: *Command a pure rotation of A.*

If breaking contact or sticking occurs, signal failure. Otherwise, signal success.

In this plan, motion (1) does not terminate distinguishably in success (meshed) or failure (jammed). That is, after motion (1) terminates, the plan executive cannot necessarily recognize whether or not the gears are meshed. LIMITED predicts this, and generates motion (2), which disambiguates the result of motion (1). The generation of the second, disambiguating motion involves the use of *failure mode analysis. Breaking contact* and *sticking* are examples of failure modes. The second motion is generated so that from any unmeshed state resulting from motion (1), all possible paths will terminate distinguishably in a failure mode. Failure mode analysis is a robust subtheory of EDR by which LIMITED generates multi-step strategies under pushing.

3.3 Experiment: Peg-in-Hole with Model Error

This section describes a plan that was generated by LIMITED for a peg-in-hole problem with model error. It gives the flavor of how EDR strategies work. Since pushing motions are not involved here, LIMITED does not use failure mode analysis to solve this problem.

Another peg-in-hole problem is depicted in fig. 3. There is uncertainty in the width of the hole; that is, the width is known to lie within some given interval. In addition, there are chamfers on the sides of the hole. The depth of the chamfers is also unknown, but we are given bounds on the depth. Finally, the exact orientation of the hole is uncertain. The geometry of the hole is input to the planner as a set of parametrically defined polygons. They are defined by a three parameter family, for *width* of the hole, *depth* of the chamfers, and *orientation* of the hole. An associated bounding interval is also input for each parameter. The geometry of the peg is input as a polygon.

In this problem, the width of the hole may be smaller than the width of the peg. Thus there can exist no strategy that is guaranteed to succeed for all geometric uncertainty values. However, assume that the assembly—the hole geometry—is inaccessible to robust vision or position-measuring devices. In particular, the measurement error will typically determine the model error bounds, which in this example are large for the purpose of illustration. *Thus it is not a priori possible to measure the dimensions ahead of time to determine whether or not the assembly is feasible.* Instead, the best we can hope for is an EDR strategy: a strategy that takes some action in the world to attempt the assembly, but whose outcome can be recognizably diagnosed as success or failure by the run-time robot executor.

The peg is allowed to translate in the plane. Its motion is modeled using generalized damper dynamics. This permits sliding on surfaces about the hole. Friction is modeled using Coulomb's law. With these dynamics and *perfect* control, the peg would exhibit straight-line motions in free space, followed by sliding motions in contact, where friction permits. Here, however, there is control uncertainty, which is represented by a cone of velocities. Motions in free space fan out in a kind of "spray." Again, sliding is possible on surfaces, but so is sticking, depending on the effective commanded velocity at a given instant. (In this case, we say sliding is *non-deterministic*). The size of the control uncertainty cone of velocities is an input to the planner. Whether sticking may occur on an edge may be computed by intersecting the friction cone with the *negative* control uncertainty cone.

It is possible to sense the position of the peg and the forces acting on it. This information is only approximate. The error bound on the position sensor readings is input to the planner as the radius of a disc.

LIMITED generates plans using a configuration space representation of the constraints [Lozano-Pérez]. In the plane, one imagines shrinking the moving object to a point, and correspondingly "growing" the obstacles. The point must be navigated through free-space, sliding on surfaces, and so forth, into the hole. Fig. 4 shows configuration spaces for different parametric variations of model error. Notice that when the "real" hole is too small for the peg to fit, then there is simply no hole at all in the corresponding configuration space. Each frame in fig. 4 is called a "slice;" a slice represents a cross-section where the model error parameters are constant. To synthesize an EDR strategy, LIMITED must in some sense consider all such slices. In practice LIMITED works by constructing a finite, although typically large number of slices. [Donald, 87b] shows that in many cases, only a low polynomial number need be considered. LIMITED begins by considering a small number of slices, and generates a tentative motion strategy. This strategy must pass a test—which we call the EDR test—to be recognized as an EDR strategy. One of the chief goals of this work is to derive this test, and to make it formal and algorithmic. Next, LIMITED attempts to "generalize" the strategy by considering successively more slices. The strategy is modified so that it passes the EDR test in all slices. The number of slices considered is the *resolution* of the planning. This approach is called *multi-resolution* planning.

Let's consider an EDR plan that LIMITED computed for this problem. Figs. 5-13 show the plan graphically. Qualitatively, the plan may be described as follows:

- *(1) First, move left and slightly down. The motion will terminate on the left side of the hole, on the left chamfers, or overshoot the hole entirely. Where the motion terminates depends both on the trajectory evolution within the control uncertainty, and on the actual geometry of the hole. The motion may, however, slide down the left edge of the hole all the way into the goal. However, this sliding is non-deterministic, and the motion may stick anywhere along that edge. Since the first motion may terminate arbitrarily close to the goal region,* LIMITED *predicts that the run-time executive system cannot necessarily distinguish whether or not the first motion failed to achieve the goal.*

- *(2) The termination regions from motion (1) are taken as the start regions for a new motion. Next, try to recover by commanding a motion straight down and slightly to the right. This motion may achieve the goal, or may undershoot it, or may overshoot it. The second motion terminates when the peg sticks on a surface. If such a termination surface is outside the goal, it is called a* failure region. LIMITED *calculates that after the second motion, the* failure regions *are distinguishable from the goal regions. Hence after the second motion, the run-time executive can recognize whether or not the plan has failed.*

Finally, since LIMITED is a forward-chaining planner, it is possible to take the failure regions from motion (2) and plan a third recovery motion. Thus, roughly speaking, in the EDR framework, *recovery* actions are planned by forward-chaining from the failure regions of the previous motion. When the failure regions are potentially indistinguishable from the goal (using sensors), then the recovery action must satisfy the formal EDR test when executed from the *union* of the goal and the previous failure regions. For example, when we view motion strategies as "mappings" between subsets of configuration space, then typical "robust" recovery actions are EDR plans in which the goal is a "fixed point."[5] Motion (2) is an example of such a one-step EDR plan.

Figs. 5–13 show the plan in just four different slices, to give a flavor for the plan. The rest of the slices may be found later in the thesis. Fig. 5 shows the configuration spaces of the four slices. The goal region here is shaded black. Note that in one slice, the goal disappears. The initial uncertainty in the position of the peg is represented by constraining the reference point (the point to which the peg has been shrunk) to lie in one of the start regions in fig. 6.

Figs. 7–8 represents the *forward projection* of the first motion. This region is the outer envelope of all possible trajectories evolving from the start regions. It is the set of all configurations that are reachable from the start regions, given the commanded velocity and control uncertainty cone.

Fig. 9 shows the termination regions for motion (1). The termination regions outside the goal are not necessarily distinguishable from the goal.

Figs. 10–11 show the forward projection of the second motion.

Fig. 12 shows the termination regions for the second motion.

Fig. 13 shows the size of the position sensing uncertainty ball. The goal and the failure regions in fig. 12 are distinguishable using sensors.

4 Research Issues

The gross motion planning problem with no uncertainty has received a great deal of attention recently; see [Yap] for a good survey article. In this problem, the state of the robot may be represented as a point in a configuration space. Thus moving from a start to a goal point may be viewed as finding an arc in free space connecting the two points. Since the robot is assumed to have perfect control and sensing, any such arc may be reliably executed once it is found. In particular, given a candidate arc, it may be tested. That is, motion along the arc may be simulated to see whether it is collision free. For example, an algebraic curve may be intersected with semi-algebraic sets defining the configuration space obstacles. In the presence of uncertainty, however, we cannot simply simulate a motion strategy to verify it. Instead, we need some technique for simulating *all* possible orbits, or evolutions of the robot system, under any possible choice of the uncertain parameters. With sensing and control uncertainty, the state of the robot must be viewed as a subset of the configuration space. Motions, then, can be viewed as mappings between these subsets. Of course there are many such subsets! From this perspective, it is clear that a chief contribution of [LMT] has been to identify and give a constructive definition for a privileged class of subsets, called *preimages*, and show that it is necessary and sufficient to search among this class. This framework appears very promising for planning guaranteed motion strategies under sensing and control uncertainty. The [LMT] framework assumes no model error. In [Donald, 86a;b], we reduced the problem of planning guaranteed strategies with

[5] That is, when motion (2) originates in the goal, it also terminates recognizably in the goal.

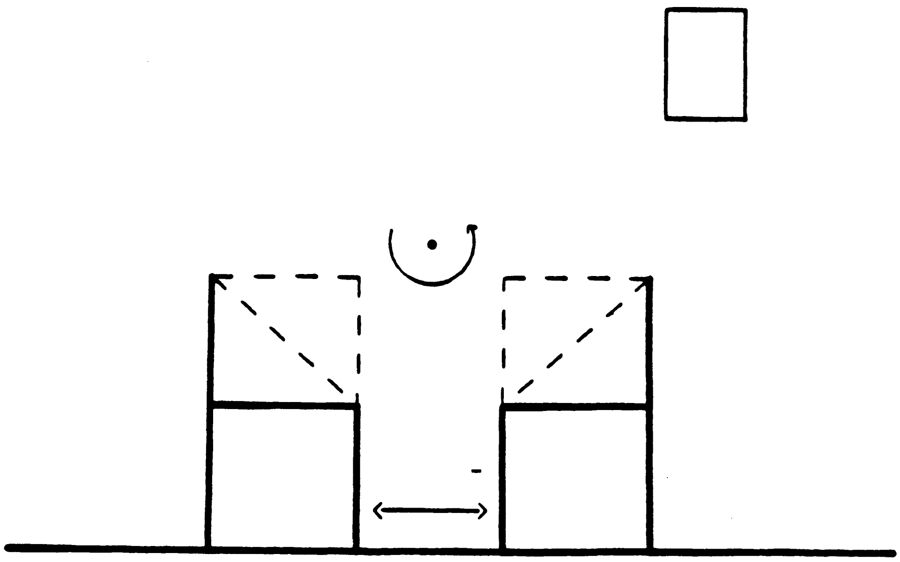

Figure 3: A peg-in-hole environment with model error. The width of the hole (α_1), depth of chamfer (α_2), and orientation of the hole (α_3) are the model parameters. The hole is allowed to close up.

sensing, control, *and geometric model* uncertainty to the problem of computing preimages in a (higher dimensional) generalized configuration space. [6]

This result is useful in as much as there was really no prior theory of planning in an environment whose geometry is not precisely known. However, I do not think that it is the main point the EDR theory. This is because there are certain inadequacies with the "guaranteed success" planning model. The insistence that strategies be guaranteed to succeed is too restrictive in practice. To see this, observe that guaranteed strategies do not always exist. In the peg-in-hole problem with model error (figs. 3-13) there is no guaranteed strategy for achieving the goal, since the hole may be too small for some model error values. For these values the goal in configuration space does not exist. Because tolerances may cause gross topological changes in configuration space, this problem is particularly prevalent in the presence of model error. More generally, there may be model error values for which the goal may still exist, but it may not be reachable. For example, in a variant of the problem in fig. 3, an obstacle could block the channel to the goal. Then the goal is non-empty, but also not reachable. Finally, and most generally, there may be model error values for which the goal is reachable but not *recognizably* reachable. In this case we still cannot guarantee plans, since a planner cannot know when they have succeeded.

These problems may occur even in the absence of model error. However, without model

[6] We use the terms model error and model uncertainty interchangeably.

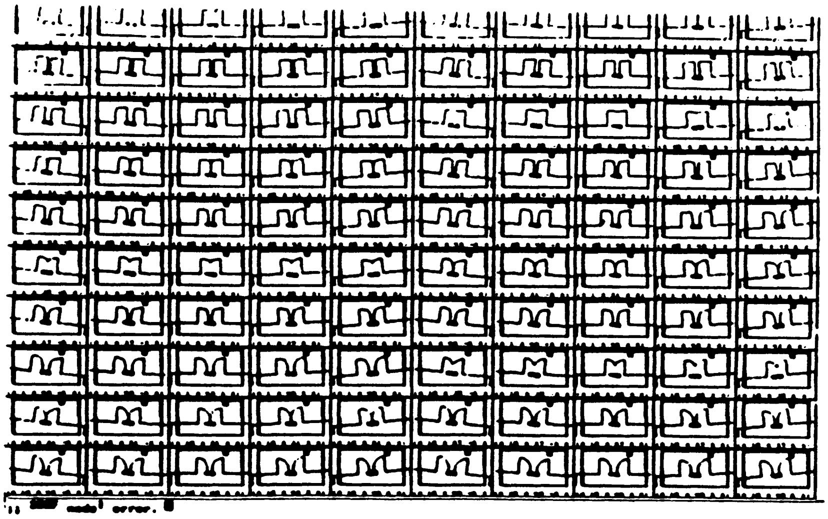

Figure 4: The configuration space slices for many different parametric model error values. These configuration spaces were generated for the peg-in-hole problem with model error depicted in fig. 3.

error a guaranteed plan is often obtainable by back-chaining and adding more steps to the plan. In the presence of model error this technique frequently fails: in the peg-in-hole problem with model error, this technique will not work since no plan of any length can succeed when the hole closes up.

This is why we investigate EDR strategies, and, in particular, attempt to formalize EDR planning. The key theoretical issue is: How can we relax the restriction that plans must be guaranteed to succeed, and still retain a theory of planning that is not completely *ad hoc*? We attempt to answer this by giving a constructive definition of EDR strategies. In particular, this approach provides a formal test for verifying whether a given strategy is an EDR strategy. The test is formulated as a decision problem about projection sets in a generalized configuration space which also encodes model error. Roughly speaking, the projection sets represent all possible outcomes of a motion (the *forward projection*), and weakest preconditions for attaining a subgoal (the *preimage*).

Given the formal test for "recognizing" an EDR strategy, we can then test the definition by building a generate-and-test planner. The generator is trivial; the recognizer is an algorithmic embodiment of the formal test. It lies at the heart of this research. A second key component of the planner is a set of techniques for chaining together motions to synthesize multi-step strategies. The planner is a forward-chaining, multi-resolution planner, called LIMITED. LIMITED operates in a restricted domain. Plans found by LIMITED in experiments

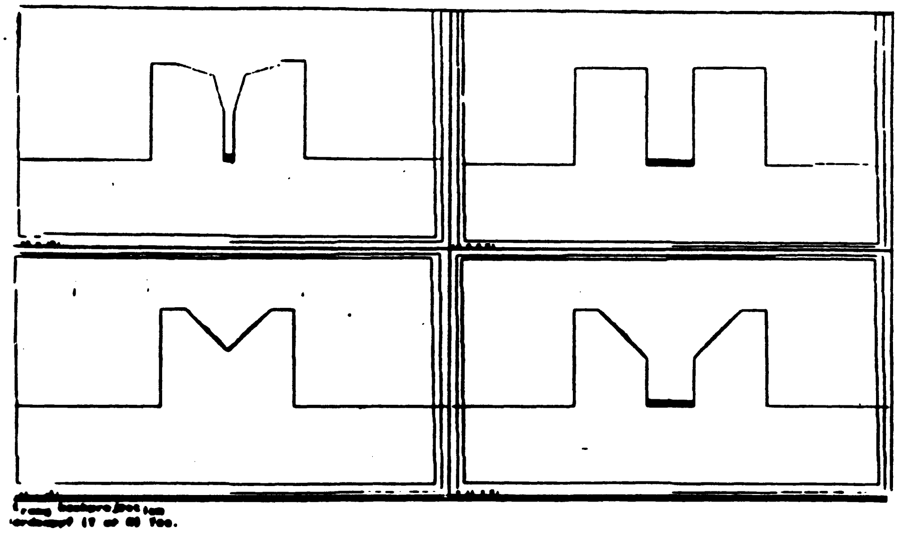

Figure 5: 4 configuration space slices for the peg-in-hole with model error problem. The goal region is shaded black. In one slice, the goal vanishes.

are described above (figs. 2-13).

Finally, let me suggest that a new framework—the EDR framework— for planning with uncertainty may be justified not only by the restrictiveness of the guaranteed-success model, but also by the hardness of the problem. The gross motion planning problem without uncertainty may be viewed, under some very general assumptions, as a decision problem within the theory of real closed fields. This gives a theoretical decision procedure with polynomial running time once the degrees of freedom of the robot system are fixed [SS]. However, no such theoretical algorithm is known for the general compliant motion planning problem with uncertainty. Furthermore, the lower bounds for computing guaranteed strategies even in 3D are dismal: the problem is known to be hard for exponential time [CR]. At this point it is unknown whether EDR planning is more efficient than guaranteed planning. However, there is some experimental evidence leading one to conjecture that certain problems requiring very complicated, exponential-sized guaranteed plans may admit very short EDR plans.

However, the motivation for this work is not complexity-theoretic. Instead, the chief thrust is to show how to compute motion strategies under model error (and sensing and control uncertainty), using a formal and constructive definition of EDR strategies. The first goal is a precise geometric characterization of EDR planning—when one thinks about it, it is in fact somewhat surprising that such a thing should exist at all! But in fact it does, as

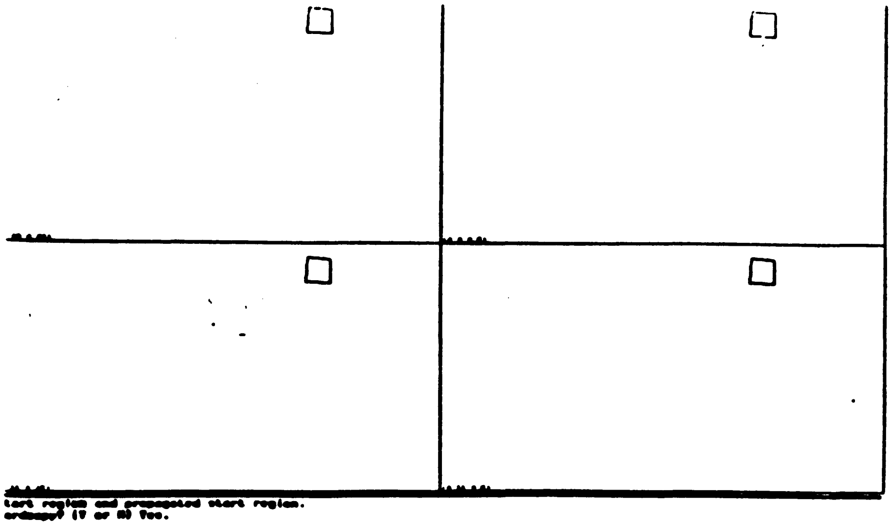

Figure 6: The start region in the four slices. The reference point of the peg is known to start within this region.

we shall sketch in sec. 5 and appendix A. The second goal is to test this characterization by building a planner. Thus it is necessary to devise implementable algorithms to construct the geometric projection sets and decide questions about them. Therefore, this theory and LIMITED both contain a mixture of precise combinatorial algorithms and of approximation algorithms. We will indicate which algorithms are exact and give combinatorial bounds. We will also identify the approximation algorithms, and indicate the goodness of the approximation and whether it is conservative. For initial progress on algorithmic issues, see [Donald, 87b]. Much work, or course, remains in developing better algorithms for EDR planning, and in testing out the plans using real robots.

5 The Basic Idea

In this section we give an overview of the EDR theory. This review is necessarily somewhat abbreviated. The reader is cautioned that the account below is somewhat intuitive and informal; we attempt to describe the key points of the EDR theory in the plain style, and omit proofs and mention of certain subtle complications. For a detailed development, please refer to [Donald 86a;b,87b,88].

To represent model error, we proceed as follows. Suppose we have a configuration space C for the degrees of freedom of the moving object. Let J be an arbitrary index set which

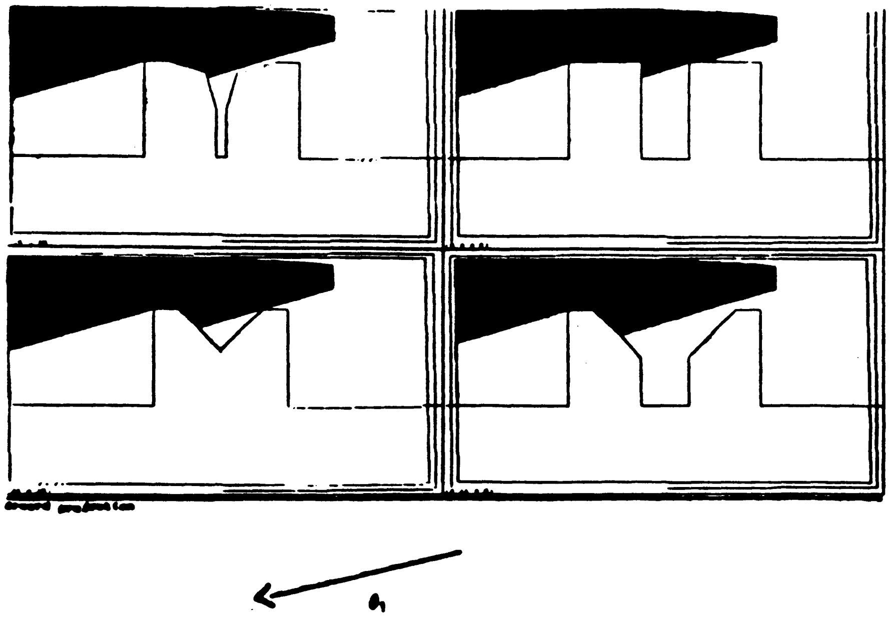

Figure 7: The forward projection of the first motion. This region is the outer envelope of all possible trajectories evolving from the start regions. It is the set of all configurations that are reachable from the start regions, given the commanded velocity and control uncertainty cone.

---

parameterizes the model error. Then the *generalized configuration space* with model error is $C \times J$. One way to think of this construction is to imagine a collection of possible "universes", $\{C_\alpha\}$ for α in J. Each C_α is a configuration space, containing configuration space obstacles. The ambient space for each C_α is some canonical C. $C \times J$ is simply the natural product representing the ambient space of their disjoint union. There is no constraint that J be finite or even countable. In fig. 3, C is the cartesian plane, and J is a three-dimensional product space. One of the J dimensions is circular, to parameterize the angular variation represented by α_3.

Next, we constrain all motions to be differentially tangent to C and have 0 component across J. In essence, this may be done by making the control velocity uncertainty cone have the dimension of C. This construction may be viewed as an axiom for "physics" in $C \times J$— specifically, it is a non-holonomic constraint. Similarly, position sensing in J is bounded by the accuracy to which the robot can perceive the shape of the environment. We envision a back-chaining planner which recursively computes pre-images [LMT,E] of a goal region G. Successive subgoals are attained by motion strategies. Each motion terminates when all sensor interpretations indicate that the robot must be within the subgoal. Preimages provide a way to construct guaranteed plans for the situation with no model error. Can preimages be generalized to situations with model error? The answer is yes. The generalized

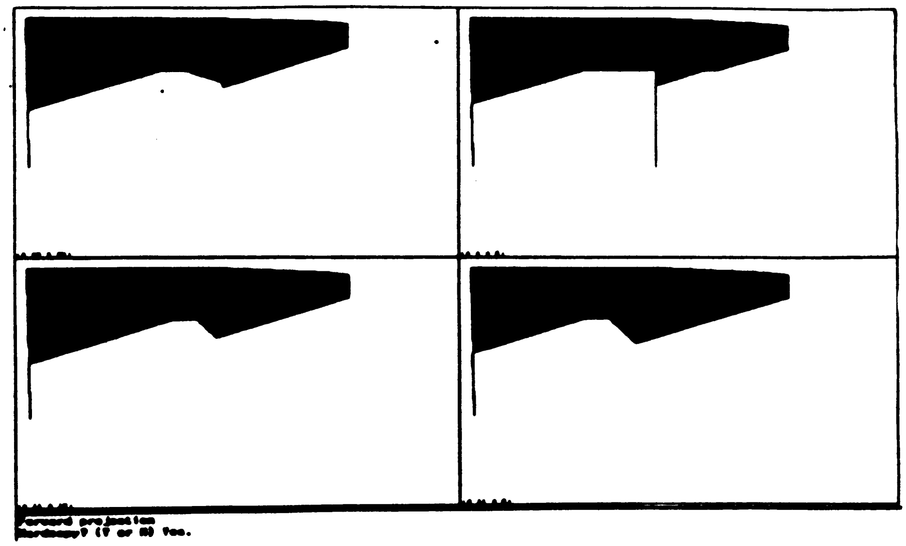

Figure 8: The forward projection of the first motion, shown without the obstacles.

control and sensing uncertainties in $C \times J$ are given by the physics axioms we developed for characterizing motion planning with model error. These uncertainties completely determine how motions in generalized configuration space must behave. We form the preimage of G under these uncertainties. The trick here is to view the motion planning problem with n degrees of motion freedom and k degrees of model error freedom as a planning problem in an $(n+k)$-dimensional generalized configuration space, endowed with the special physics described in [Donald 86a;b,87b,88].

We now review the EDR construction. These criteria are called the "EDR Axioms:"

- An EDR strategy should attain the goal when it is recognizably reachable, and signal failure when it is not.

- It should also permit serendipitous achievement of the goal.

- Furthermore, no motion guaranteed to terminate recognizably in the goal should ever be prematurely terminated as a failure.

- Finally, no motion should be terminated as a failure while there is any chance that it might serendipitously achieve the goal due to fortuitous sensing and control events.

We introduce an "additional" goal-like set H, which is disjoint and distinguishable from G, such that when H is recognizably achieved, then failure of the motion may be signalled.

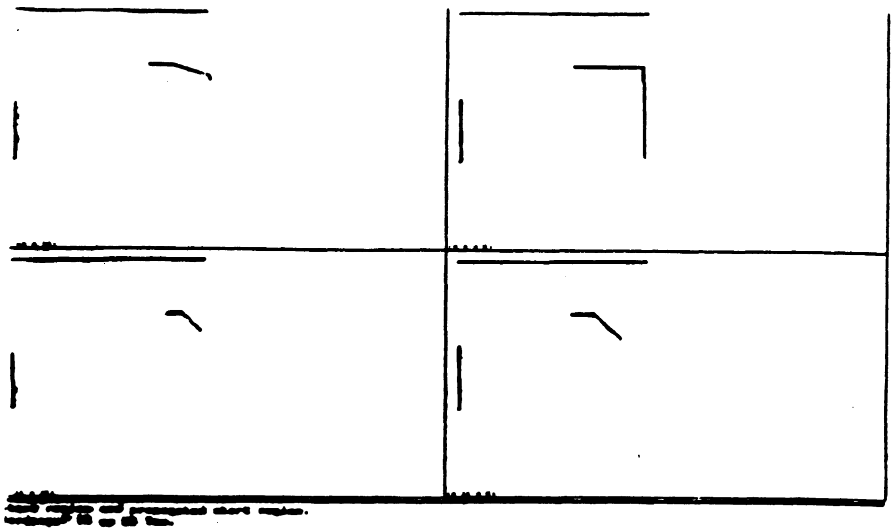

Figure 9: The termination regions from the first motion. These regions are configurations where the motion finishes.

That is, we construct an H such that recognizably achieving H is equivalent to "failure." H is called the *EDR region.* Remarkably, H may be constructed such that the EDR axioms are satisfied. Intuitively, H has two components: the first is the set-difference of the forward projection minus the "weak" preimage. The second is regions where sticking may occur within the "weak" preimage. (The "weak" preimage [LMT] is all points from which the goal may *possibly* (but not necessarily) be achieved). H may be computed given G, the motion direction θ, and the start region R; see sec. A. Next, we must test to see whether G and H are distinguishable using sensors; this is the "EDR test" of sec. 3.3. If so, then a one-step EDR strategy is in hand. If not, then a multi-step strategy must be constructed. These algorithms have been implemented in LIMITED. At a high level, the one-step algorithm is:

Algorithm 1EDR

1. *Generate a commanded velocity* v_θ^*.
2. *Compute the EDR region* H *for* v_θ^*,
3. *Determine whether the EDR region* H *and the goal* G *are distinguishable using sensors. If so, then* v_θ^* *yields a one-step EDR strategy which recognizably terminates in* G *or* H *by monitoring position and force sensors.*

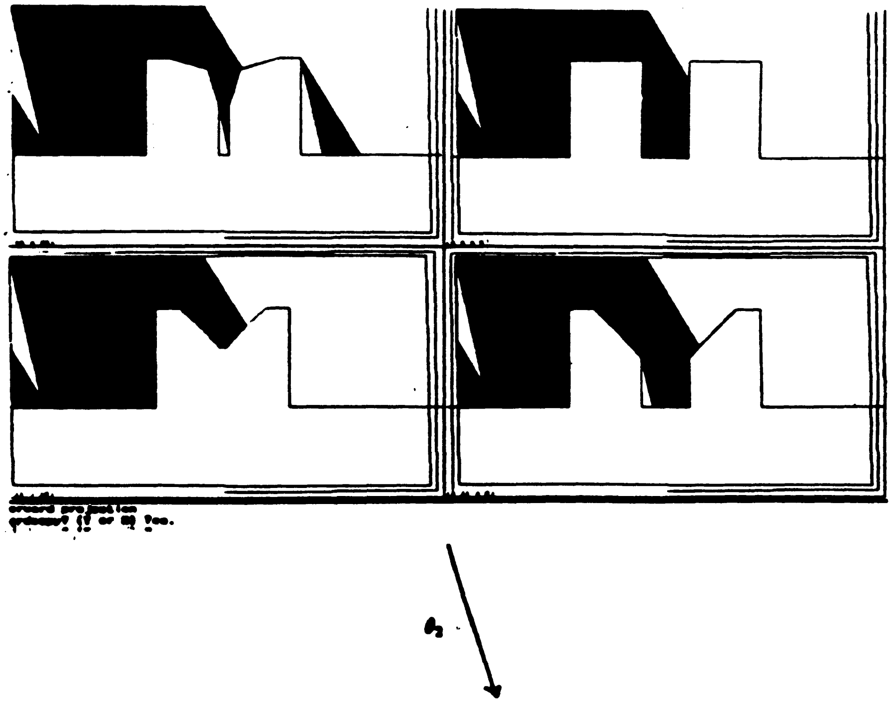

Figure 10: The forward projection of the second motion.

4. *Let* $\text{push}_\theta(G)$ *and* $\text{push}_\theta(H)$ *denote the* sticking push-forwards. *They are the set of obstacle edges within G and H, resp., on which sticking can occur under* v_θ^*. *Determine whether these regions are distinguishable using sensors. If so, then* v_θ^* *yields a one-step EDR strategy which recognizably terminates when sticking is detected.*

LIMITED decided whether G and H are distinguishable using sensors using error bounds on position- and force-sensing accuracy. H and G are distinguishable using position sensing alone if their convolutions (Minkowski sums) by the position sensing error ball B_{ep} do not intersect. Each obstacle edge of H and G has an associated configuration space friction cone. Two edges are distinguishable using force sensing if the convolutions of their friction cones by the force sensing uncertainty B_{ev} have a trivial intersection. This procedure is correct for linear edges, where position- and force-sensing are separable because the set of possible reaction forces is constant along an edge. For the general case, see [Donald, 87b].

However, for both the gear-meshing and peg-in-hole problem, one-step strategies cannot be generated for the sensing and control error bounds we input. More specifically: in these cases, H can be computed for any hypothesized one-step motion θ, but LIMITED cannot find a θ for which H is distinguishable from the goal. Intuitively, the reason that H and G are not distinguishable is that (a) since the control uncertainty is "large", therefore H is "large", and (b) since the error bounds on position and force sensing are large, therefore it is not always possible to distinguish between H and G.

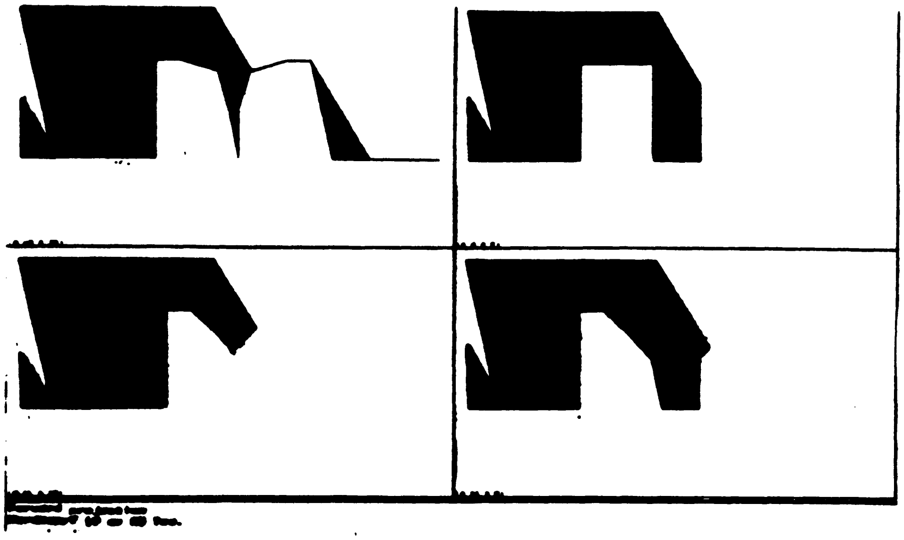

Figure 11: The forward projection of the second motion, shown without the obstacles.

What is to be done in this case? In short: multi-step strategies are required. This raises the central issue: how can multi-step EDR strategies be generated? We are interested both in an implementable (and, in LIMITED, implemented) practical approach, and also in developing unifying theory for multi-step EDR planning. See [Donald, 87b] for initial progress in this direction. We summarize one aproach to multi-step strategy generation below:

5.1 Multi-step EDR Plan for Peg-in-Hole with Model Error

The advantage of the push-forward technique is that it can be made computational. We now give LIMITED's algorithm for generating multi-step strategies using push-forwards. Recall figs. 3-13, which described a two-step EDR plan for a peg-in-hole plan with 3 DOF model error. Here is how this multi-step strategy was generated:

Algorithm Multi

1. *First, try to generate a one-step EDR strategy using the algorithm 1EDR in sec. 5.*
   Suppose this fails. Then:
2. *Generate a commanded velocity v_θ^*, such that the forward projection of the start region intersects the goal in some slice.*

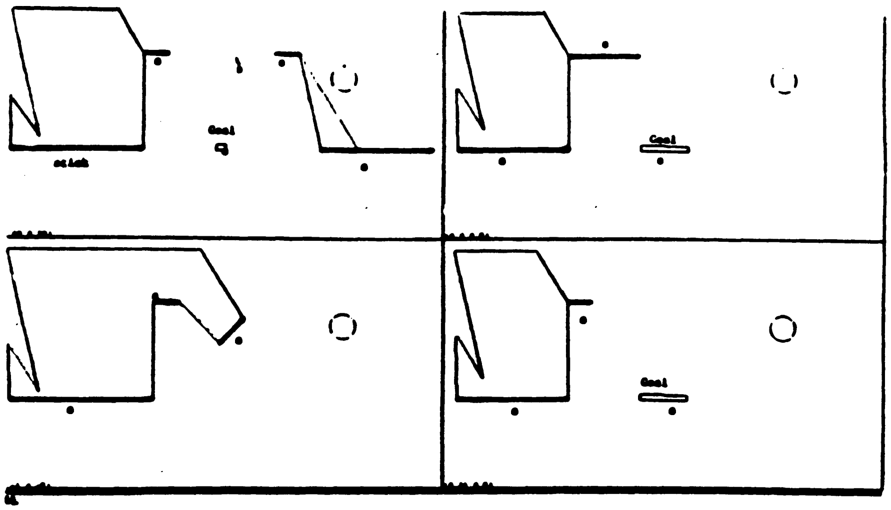

Figure 12: The termination regions for the second motion. These are edges in configuration space where sticking can occur.

3. *Compute the EDR region H for v_θ^*.*

4. *Compute the sticking push-forward of the motion, $R_1 = \text{push}_\theta(G \cup H)$.*

5. *Using R_1 as the start region, generate a one-step EDR strategy using algorithm 1EDR.*

Of course, in LIMITED the computation is memoized so that the projection and EDR regions computed in step (1) are not recalculated in steps (2) and (3). Obviously, we can extend this algorithm to generate longer strategies which push-forward several times and finally terminate in a single-step EDR strategy.

Now, LIMITED is a multi-resolution planner. The algorithm outlined above generates a multi-step strategy at a single resolution. The *resolution* of planning is simply the set of α values in which slices are taken. A resolution S_1 is *finer* than S_2 if it contains more slices. The multi-resolution outer loop works like this:

M1. *At a coarse resolution, generate a multi-step EDR strategy $\theta_1, \ldots, \theta_n$ using the forward-chaining single-resolution algorithm above.*

M2. *Select a finer resolution. Use the directions $\theta_1, \ldots, \theta_n$ as a suggested strategy and attempt to verify that it is an EDR strategy at the finer resolution.*

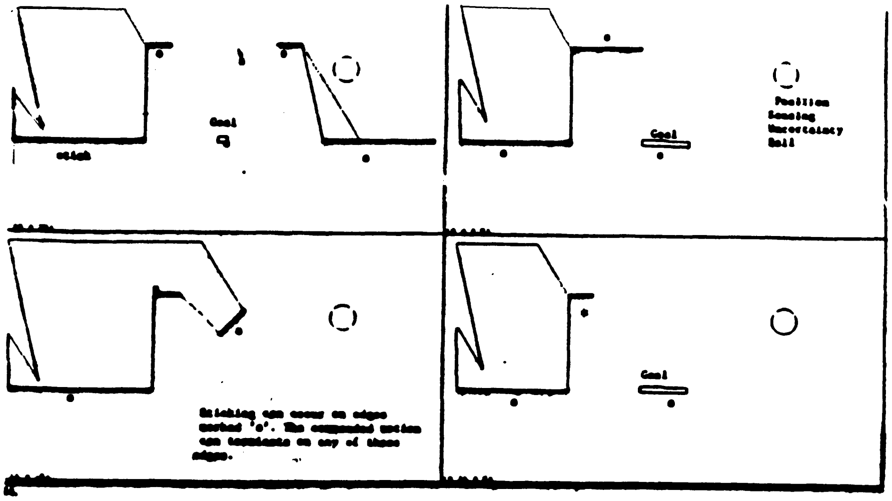

Figure 13: The failure regions and the goal are distinguishable, even given the sensing uncertainty. The disc indicates the magnitude of the position sensing uncertainty.

M3. *If $\theta_1, \ldots, \theta_n$ is not an EDR strategy at the finer resolution, try to modify it so that it is, by using $\theta_1, \ldots, \theta_n$ as suggested directions, and searching nearby directions at all levels.*

The process terminates when the resolution is finer than some predetermined level.[7] The critical slice method described in [Donald, 87b] may be one way to obtain such an *a priori* bound and know that it is sufficient. In LIMITED, however, the bound is a user input because otherwise the number of slices required would be prohibitive.

In the peg-in-hole example there were 3DOF of model error: the ***width*** of the hole, the ***depth*** of the chamfers, and the ***orientation*** of the hole. The resolutions used in planning the two-step strategy were as follows:

R1. Holding orientation fixed, 4 slices of the depth $\times$ width axes.

R2. Holding orientation fixed, 16 slices of the depth $\times$ width axes.

R3. Holding orientation fixed, 72 slices of the depth $\times$ width axes.

R4. 100 slices of the depth $\times$ width $\times$ orientation axes.

[7] Or, when at some level, no EDR strategy can be found.

For detailed output traces of LIMITED, please see [Donald, 87b]. The theory of multi-step EDR planning is complex and somewhat subtle. See appendix A for an introduction. This is a rich and fruitful area for future research.

6 Planned Research

Our primary goal is to extend the science base for task-level planning by exploring and refining our theory of Error Detection and Recovery. We feel this theory, if reduced to practice, could have an enormous influence on both theoretical and experimental robotics. For example, the EDR theory has proved useful in the research of Randy Brost, at Carnegie-Mellon.[8] Brost is synthesizing EDR strategies for manipulator pushing and grasping operations in Matt Mason's group. What is perhaps most encouraging about Brost's results is that the EDR paradigm has found applicability in a new domain. Brost's domain is significantly different from EDR's original application areas.

A key issue in task-level planning is how to plan multi-step EDR strategies. We have recently begun developing an exciting new generalization of the EDR theory, which admits a very clean technique for synthesizing multi-step strategies; this generalization, which we call the "weak" EDR theory, is described briefly in an appendix. The weak EDR theory provides a new mathematical and algorithmic framework for motion planning with uncertainty.

While the exploration of the EDR theory requires further advances in the theory of manipulation and geometry-based strategy synthesis, we also intend to continue developing and testing algorithms for geometrical EDR planning. The formal analysis of such algorithms is still in its infancy, and we intend to continue studying their correctness, completeness, and complexity. We also intend to continue implementing our planning and geometric reasoning algorithms to test them. We are also eager to explore using more sophisticated dynamic models in geometrical planning, and their effect on the contact problem.

In addition, working with two students in computer science (Doug Campbell and Jim Jennings) and one in mechanical engineering (Dinesh Pai), we are currently implementing an approximate generalized damper dynamics force-control compliant-motion system for our six degree-of-freedom PUMA robot. This will allow us to test our EDR plans in realistic assembly situations that are very close to the complexity of real-world assembly problems.

7 Review of Previous Work

The initial progress on the EDR theory has been reported in [Donald, 87b]. Broadly speaking, previous work falls in the following categories: algorithmic motion planning, compliant motion planning with uncertainty, model error, and error detection and recovery.

7.1 Algorithmic Motion Planning

In algorithmic motion planning, (also called the piano movers' problem, or the find-path problem) the problem is to find a continuous, collision-free path for a moving object (the robot) amidst completely known polyhedral or semi-algebraic obstacles. It is assumed

[8]See Brost's forthcoming Ph.D. thesis.

than once such a path is found, it can be reliably executed by a robot with perfect control and sensing. Many algorithms employ configuration space, [Lozano-Pérez, Arnold, Abraham and Marsden, Udupa]. [Lozano-Pérez and Wesley] proposed the first algorithms for polygonal and polyhedral robots and obstacles without rotations. These results were later extended by [Lozano-Pérez 81, 83] to polyhedral robots which could translate and rotate. [Brooks 83] designed a find-path algorithm based on a generalized-cone representation of free-space. Brooks later extended this method for a revolute-joint robot. [Donald 84,85,87a] developed a motion-planning algorithm for a rigid body that could translate and rotate with six degrees of freedom amidst polyhedral obstacles (the so-called "classical" movers' problem). [Lozano-Pérez 85] reported another 6DOF algorithm for 6-link revolute manipulators. [Canny 85] developed an algebraic formulation of the configuration-space constraints, which led to a very clean collision-detection algorithm. All of these algorithms have been implemented.

There are many theoretical results on upper and lower bounds for the find-path problem, see [Yap] for a good survey article. These results begin with [Lozano-Pérez and Wesley], who give the first upper bounds: they give efficient algorithms for planning in 2D and 3D in the absence of rotations. [Reif 79] obtained the first lower bounds, demonstrating the problem to be $\mathcal{PSPACE}$-hard when the number of degrees of freedom are encoded in the input specification of the problem. [Hopcroft, Joseph, and Whitesides] and [Hopcroft, Schwartz, and Sharir] have also given intersesting lower bounds for motion planning. [Schwartz and Sharir] gave a very general theoretical algorithm for motion planning via a reduction to the theory of real closed fields. The algorithm is doubly-exponential in the degrees of freedom, but polynomial in the algebraic and geometric complexity of the input. Over the next five years, there were many papers reporting more efficient special-purpose motion planning algorithms for certain specific cases; see [Yap] for a survey. To date the fastest general algorithm is due to [Canny, 87], who gives a generic motion planning algorithm which is merely singly-exponential in the degrees of freedom. For a motion planning problem of algebraic complexity d, geometric complexity n, and with r degrees of freedom, Canny's algorithm runs in time $(d^{O(r^2)} n^r \log n)$ which is within a log factor of optimal. While none of these theoretical algorithms have been implemented, Canny's is conjectured to be efficient in practice as well.

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 [Ó'Dúnlaing and Yap 82; Ó'Dúnlaing, Sharir, and Yap 84; Yap 84], and the textbook of [Schwartz and Yap 86] 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. [Canny and Donald] 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, [Canny and Donald] show that motion planning using the Simplified diagram can be done it time $O(n^7 \log n)$.

Many additional robotics issues are discussed in [Paul; Brady *et al.*].

7.2 Compliant Motion Planning with Uncertainty

This section reviews previous work on planning compliant motions which are guaranteed to succeed even when the robot system is subject to sensing and control uncertainty. All of this work assumes perfect geometric models of the robot and obstacles.

Work on compliant motion can be traced to [Inoue, Whitney, Raibert and Craig, Salisbury]. This work in force control attempted to use the geometric constraints to guide the motion. By cleverly exploiting the task geometry, placements far exceeding the accuracy of pure position control can be achieved. [Mason 83] develops spring and damper compliance models, and gives an extensive review of research in compliant motion. [Simunovic, Whitney, Ohwovoriole and Roth, Ohwovoriole, Hill and Roth] have all considered frictional constraints, as well as jamming and wedging conditions. [Erdmann], [Burridge, Rajan and Schwartz] have considered algorithmic techniques for predicting reaction forces in the presence of friction. [Caine] has considered manual techniques for synthesizing compliant motion strategies, generalizing the methods of [Simunovic, Whitney]. [Mason, 82] has developed a way to model pushing and grasping operations in the presence of frictional contact. [Peshkin] has extended this work. [Brost] has further developed techniques for predicting pushing and sliding of manipulated objects to plan squeeze-grasp operations. In addition, Brost is currently investigating the application of EDR techniques to the squeeze-grasp domain.

Early work on planning in the presence of uncertainty investigated using skeleton strategies. [Lozano-Pérez 76] proposed a task-level planner called LAMA which used geometric simulation to predict the outcomes of plans, and is one of the earliest systems to address EDR planning. [Taylor] used symbolic reasoning to restrict the values of variables in skeleton plans to guarantee success. [Brooks 82] later extended this technique using a symbolic algebra system. [Dufay and Latombe] implemented a system which addresses learning in the domain of robot motion planning with uncertainty.

[LMT] proposed a formal framework for automatically synthesizing fine-motion strategies in the presence of sensing and control uncertainty. Their method is called the *preimage* framework. [Mason, 83] further developed the preimage termination predicates, addressing completeness and correctness of the resulting plans. [Erdmann] continued work on the preimage framework, and demonstrated how to separate the problem into questions of *reachability* and *recognizability*. He also showed how to compute preimages using backprojections, which address reachability alone, and designed and implemented the first algorithms for computing backprojections. [Erdmann and Mason] developed a planner which could perform sensorless manipulation of polygonal objects in a tray. Their planner makes extensive use of a representation of friction in configuration space [Erdmann]. [Buckley] implemented a multi-step planner for planning compliant motions with uncertainty in 3D without rotations. He also developed a variety of new theoretical tools, including a combined spring-damper dynamic model, 3D backprojection and forward projection algorithms, and a finitization technique which makes searching the space of commanded motions more tractable.

[Hopcroft and Wilfong] addressed the problem of planning motions in contact, and proved important structural theorems about the connectivity of the 1-edges of configuration space obstacle manifolds. [Koutsou] has suggested a planning algorithm which plans along 1-edges. Other planning systems for compliant motion have been developed by [Turk], who used backprojections, [Laugier and Theveneau], who use an expert system for geometric reasoning about compliant motion, and [Valade].

Recently, there has been some theoretical work on the complexity of robot motion planning with uncertainty. [Erdmann] showed the problem to be undecidable when the obstacles are encoded as a recursive function on the plane. [Natarajan] has shown the problem to be $\mathcal{PSPACE}$-hard in 3D for finite polyhedral obstacles. [Canny and Reif] have demonstrated that in 3D the problem of synthesizing a multi-step strategy is hard for non-deterministic exponential time; in addition, they proved that verifying a 1-step strategy is $\mathcal{NP}$-hard.

7.3 Model Error

There is relatively little previous work on planning in the presence of model uncertainty. [Requicha] and [Shapiro] address representational questions of how to model part tolerances, and mathematical models for variational families of parts. [Buckley] considers some extensions of his planner to domains with model uncertainty. [Brooks 82] developed a symbolic algebra system which can constrain the variable values in skeleton plans, and introduce sensing and motion steps to reduce these values until the error ranges are small enough for the plan to be guaranteed. Some of the variables in these plans can represent model error—particularly, the position of objects in the workspace—and hence his planner can reason about motion planning in the presence of model uncertainty.

Work on manipulator pushing and sliding [Mason, Peshkin] and squeeze-grasping [Brost] may be viewed as addressing model error where the error parameters are the position and orientation of the manipulated part. The *operation space* of [Brost] is a clever example how to model actions with uncertain effects, and objects with uncertain orientation, in the same space. [Durrant-Whyte] considers how to model geometric uncertainty probabilistically, and how to propagate such information in applications related to motion planning.

[Lumelsky] considers the following 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 [Koditschek] for extensions to this approach using potential fields. The *potential-field* approach to collision avoidance, as formulated by [Khatib], also can deal with uncertain obstacles, and gross motions around these obstacles can often be synthesized in real time. [Brooks 85] has described a map-making approach for a mobile robot in a highly unstructured environment – i.e., amidst unknown obstacles. His approach allows the robot to aquire information about the position and shape of these obstacles as the robot explores the environment. [Davis] has addressed the mobile robot navigation problem amidst partially unknown obstacles using an approximate map.

There is almost no work on planning compliant motions or assemblies in the presence of model error.

7.4 Error Detection and Recovery

There has been almost no formal analysis of the EDR problem. STRIPS [Fikes and Nilsson] has a run-time executive (PLANEX) which embodied one of the first systems addressing EDR. STRIPS' triangle tables may be viewed as a kind of forward projection. [Ward and McCalla; Hayes] have presented research agendas for error diagnosis and recovery in domain-independent planning. [McDermott] has stressed the importance of EDR in plan execution and sketched an approach based on possible worlds. [Srinivas] described a robot planning system for a Mars rover which could detect certain manipulation errors and recover. [Gini and Gini] have described a view of EDR based on a predetermined list of high-level error types. The domain-independent planning literature [Chapman] is relevant to the history of EDR; for example, the planner of [Wilkins] has an error recovery module in which the executor can detect inconsistencies in the set of logical propositions representing the world state. At this point, an operator can intervene and type in new propositions to disambiguate the state and aid recovery. The robots described by [Brooks 85] have an EDR flavor—they are not required to achieve a particular goal, but merely to attempt it until some other goal takes a higher priority.

Preliminary reports on the EDR theory have been presented in [Donald 86a,b; 87b]. In [Donald, 88], we describe details of the geometrical characterization of EDR, and of one-step EDR planning. [Brost] is employing some of these EDR techniques in his research on planning squeeze-grasp operations.

8 Available Resources

During the last four years, the Cornell Computer Science Department has built up a major facility with which to pursue research in automation, solid modelling, and robotics. We can draw on skills and equipment from within the robotics group itself, from the department, and from the University. The robotics effort has received both direct industrial gifts, from AT&T Bell labs and from Xerox, and gifts through the Cornell Manufacturing Engineering and Productivity Program (COMEPP). The department's coordinated experimental research (CER) grant, while providing no additional personnel support for robotics, will provide additional computing resources.

8.1 Facilities

Dr. Donald is director of the Cornell Computer Science Group Robotics Laboratory. The laboratory houses a PUMA 560 robot arm with a VAL-II controller, a Multibus-based 68000 computer system running NRTX (a real-time variant of UNIX) for real-time control, an Astek 6-axis force-sensing wrist, and a Monforte tool-changing gripper with finger positioning and sensing. Computer equipment in the laboratory includes a VAX 11/750, a SUN workstation and a Xerox dandelion workstation. In addition, the robotics group has Symbolics 3640 lisp machines and two vectrix color display systems. Our robotics group also has access to a six degree-of-freedom cartesian robot with a force-sensing wrist, housed in the department of mechanical engineering. This robot is an IBM RS/2 running AML/X.

This fall, the robotics laboratory will purchase LUCID common lisp for its SUN workstation and for a general departmental SUN workstation. In addition, with a grant from AT&T Bell labs, the laboratory will purchase vision equipment, probably including stereo cameras, Datacube frame-grabber, frame-buffers, and DSP/IP boards, all of which will plug into the SUN VME bus.

Departmental resources include about fifteen additional Symbolics lisp machines, a large number of Vaxes, SUNs, and Dandelions.

APPENDIX

A An Introduction to the Weak EDR Theory

Beginning with a simplified notation for preimages, we now give the construction of the EDR region H. It is a set in phase-space, that is, Position- $\times$ Velocity- space. Let (x, v) be a point in phase-space, and define the *projection* $\pi(x, v) = x$. Let the *zero-section* $Z(x)$ denote the set of phase-space points of the form $(x, 0)$. If R is the start region, let $F_\theta(R)$ denote the forward projection of R under commanded motion θ; it is all configurations achievable under θ. Let $P_\theta(G)$ denote the strong preimage of G under θ, and $\widehat{P}_\theta(G)$ denote the weak preimage. Define

$$\begin{aligned} H_0 &= F_\theta(R) - \widehat{P}_\theta(G) \\ H_s &= \{\, x \in \widehat{P}_\theta(G) - P_\theta(G) \mid \text{sticking is possible at } x \,\}. \end{aligned}$$

Then we define $H = \pi^{-1}(H_0) \cup Z(H_s)$. H is the *EDR region for θ, R, and G.*

We now introduce the full notation for preimages. The key point [LMT,E,Mason] is that knowing where the motion began (that is, knowing R) can be used by the termination predicate to predict what configurations are reachable, and therefore, to effect recognizable termination. This prediction may be accomplished via the forward projection; its use as such was termed ***history*** by Erdmann. Hence the preimage map P depends not only on θ and G, but also on R. Our notation for preimages must reflect this dependence. We observed above that if the termination predicate can distinguish between the goal G and the EDR region H, then H is a good EDR region and an EDR strategy was in hand. Formally, we write this recognizability constraint as

$$P_{\theta,R}(\{\, G, H \,\}) = R. \tag{$*$}$$

We say that the preimage $(*)$ is taken *with respect to R*. $(*)$ means that the (strong) preimage of the set of goals $\{\, G, H \,\}$, with respect to commanded velocity v_θ^*, is all of R. When we have a *set* of goals, the termination predicate must return *which* goal (G or H) has been achieved. This is different from $P_{\theta,R}(G \cup H)$, which means the termination predicate will halt saying "we've terminated in G or H, but I don't know which." The region R appears on both sides of $(*)$ because the preimage depends on knowing where the motion started. This is a subtle point, see [LMT,E]. Thus solving preimage equations like $(*)$ for R is like finding the fixed point of a recursive equation. Here, however, we know R, H, and G, so $(*)$ is a constraint which must be true, rather than an equation to solve. Presumably $(*)$ is easier to check than to solve for R; see [LMT,E,D].

The unifying theory behind the push-forward algorithm and failure-mode analysis is the "weak" EDR theory. The key idea behind the weak EDR theory is: given a collection of goals $\{\, G_\beta \,\}$ (possibly including H), we consider all unions of the subcollections to get some measure of weakest recognizability. In [Donald, 87b], we show that failure mode analysis and the push-forward algorithm are essentially special cases of the weak EDR theory. The motivation behind this theory is that when a motion terminates ambiguously, a subsequent motion may be synthesized which disambiguates the success or failure of the first. Oddly enough, it is not necessary for either motion individually to satisfy the EDR axioms.

However, when taken together, the two-motion plan can often be considered "equivalent" to a one-step EDR strategy. The weak EDR theory is developed using *reachability* and *recognizability diagrams*. For example, the fixed-point condition (F3) is denoted by

$$\begin{tikzcd} & \mathrm{push}_{\theta_1}(G) \arrow[r, Rightarrow, "\theta_2"] & G \\ R \arrow[ur, "\theta_1"] \arrow[dr] & & \\ & \mathrm{push}_{\theta_1}(H). & \end{tikzcd} \tag{1}$$

The single arrows denote reachability; the double arrows denote recognizability. Branching arrows indicate disjunction. Thus (1) says that from R, θ_1 is guaranteed to *reach* G or H; specifically, it will terminate somewhere in $\mathrm{push}_{\theta_1}(G \cup H)$ (although we may not know in which of G or H). However, when motion θ_2 known to have started in $\mathrm{push}_{\theta_1}(G)$, then it will *recognizably* terminate within G. See (*); formally, (1) says that $R = P_{\theta_1,R}(\mathrm{push}_{\theta_1}(G) \cup \mathrm{push}_{\theta_1}(H))$. If instead we had $R = P_{\theta_1,R}(\{\,\mathrm{push}_{\theta_1}(G), \mathrm{push}_{\theta_1}(H)\,\})$, then the single arrows in (1) could be replaced by double (recognizability) arrows.

The weak EDR theory effectively defines some laws of "composition" that permit two single-step plans to be concatenated into a two-step plan satisfying the EDR axioms. Hence it is often possible to construct multi-step plans that are EDR plans "globally" although not "locally". That is, considered as entire plans, they satisfy the EDR axioms; this is the "global" condition. However, "locally" they are not EDR plans, in that no single step is an EDR strategy; such steps are called "weak" EDR strategies. The key to pasting together non-EDR plans to make a global EDR strategy lies in defining certain local "niceness" conditions for how plans must mesh. These are called the *linking conditions*. For example, let "push_{θ_1}" denote $\mathrm{push}_{\theta_1}(G \cup H)$, and H' denote the EDR region for θ_2. For a start region R, let

$$\begin{aligned} R_1 &= R \cap P_{\theta_1,F_{\theta_1}(R)}(G) \\ F_{\theta_2} &= F_{\theta_2}(\mathrm{push}_{\theta_1}) \\ R_2^* &= G \cap \mathrm{push}_{\theta_1} \cap P_{\theta_2,F_{\theta_2}}(G). \end{aligned}$$

Consider

$$\underbrace{\begin{array}{ccccc} R_1 & \xrightarrow{\theta_1} & G & \supset \\ & \nearrow & & \\ R - R_1 & \longrightarrow & H & \supset \end{array}}_{\text{reachability } \theta_1} \overbrace{\begin{array}{ccc} G \cap \mathrm{push}_{\theta_1} & = & R_2^* \\ & & \\ H \cap \mathrm{push}_{\theta_1} & = & \mathrm{push}_{\theta_1} - R_2^* \end{array}}^{\text{linking conditions}} \underbrace{\begin{array}{cc} \xRightarrow[b]{a} & G \\ \Nearrow & \\ \xRightarrow{c} & H' \\ \Searrow & \\ d & H' \cup G \end{array}}_{\text{recognizability } \theta_2} \tag{2}$$

(2) is the full reachability and recognizability diagram for the weak EDR theory. The arrows (a)-(d) all correspond to motion θ_2. The branching recognizability arrows indicate that θ_2 is guaranteed to distinguish which set is achieved after termination. In the "strong" EDR theory, (d) is absent.

Failure Mode Analysis. The reachability and recognizability diagram for failure mode analysis is found by deleting arcs (b) and (d) from (2). In LIMITED, arc (a) is essentially a user input[9]. Failure mode analysis links a weak EDR strategy θ_1 followed by a strong EDR strategy θ_2. (a) warrants that G is a fixed-point under θ_2. (c) ensures that failure is preserved under θ_2: no serendipitous goal achievement from H is possible. Thus such plans are pure *disambiguation* strategies.

Push-Forward Algorithms. Plans found by push-forward algorithms such as *Multi* admit a diagram from (2) containing arcs (a), (b), and (c), but not containing (d). The arc (b) permits serendipitous goal achievement from H under θ_2. Push-forward plans differ from failure mode analysis plans in that the arc (b) is permissible, and (a) is not a user input. The peg-in-hole plan with model error (figs. 3–13) is an example of such a plan.

2-Step Weak EDR. A plan admitting the diagram (2) with all four arcs (a)–(d) demonstrates "weak" EDR. It is formed by linking together two weak EDR strategies into a 2-Step plan. Here is the meaning of such a plan: The key is that θ_2 does not corrupt the goal state; that is, G is a fixed point under θ_2. The desirability of outcomes G and H' are clear: regarding H' as the EDR region for the composed strategy $\theta_1 * \theta_2$, the EDR axioms are satisfied when (d) is absent. One might ask, what good is weak EDR? Why would anyone want outcome $H' \cup G$? The answer is: in strong EDR (arc (d) absent), the run-time executor can (i) disambiguate the result of motion θ_1, and (ii) in case of failure, know that the robot is not in the goal. In weak EDR, we have (i) but not (ii). That is, in outcome $H' \cup G$, we have completely diagnosed the result of motion θ_1, although in the process, we may have accidentally moved into the goal. That is, we may indicate failure when we have, in fact, succeeded. However, we will never indicate success unless it is certain. In short, when linked, the composed strategy $\theta_1 * \theta_2$ is "conservative" about declaring success.

In all cases, the linking conditions are required. Thus the linking conditions have somewhat surprisingly turned out to be the underlying characterization for multi-step EDR strategies. That is, since they are necessary and sufficient conditions for constructing multi-step EDR plans, the linking conditions may, in fact, be taken as the *definition* of multi-step EDR strategies.

[9] Although we have discussed methods for inferring (a) computationally, this is really a direction for future work rather than a focus of this research.

B References

Certain frequently cited references have been given shorter mnemonics, eg., [LMT].

Abraham, R. and Marsden, J. *Foundations of Mechanics*, Benjamin/Cummings, London (1978).

Arnold, V. I. *Mathematical Methods of Classical Mechanics*, Springer-Verlag, New York (1978).

Ben-Or M., Kozen D., and Reif J., "The Complexity of Elementary Algebra and Geometry", J. Comp. and Sys. Sciences, Vol. 32, (1986), pp. 251-264.

Boyse J. W., "Interference Detection Among Solids and Surfaces", Comm ACM, vol 22, No 1 (1979) pp 3-9.

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

Brooks, R. A. *Symbolic Error Analysis and Robot Planning*, International Journal of Robotics Research, Vol 1, no. 4, Dec., 29–68 (1982).

Brooks, R. A *A Robust Layered Control System for a Mobile Robot*, IEEE Journal of Robotics and Automation **RA-2** (1): 14–23. Also MIT A.I. Lab Memo 864 (1985).

Brooks, R., "Solving the Find-Path Problem by Good Representation of Free Space", *IEEE Transactions on Systems, Man, and Cyberbetics*, Vol. 13, 1983.

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.

Brost, R. C. *Planning Robot Grasping Motions in the Presence of Uncertainty*, Computer Science Department and the Robotics Institute, Carnegie-Mellon University, CMU-RI-TR-85-12 (1985).

Brost, R., "Automatic Grasp Planning in the Presence of Uncertainty", *IEEE International Conference on Robotics and Automation*, San Francisco, April, 1986.

Brost, R. *A State/Action Space Approach to Planning Robot Actions*, Forthcoming Ph.D. Thesis, Computer Science Dept., CMU (to appear).

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).

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).

Caine, M., "Chamferless Assembly of Rectangular Parts in Two and Three Dimensions", S.M. dissertation, MIT Department of Machanical Engineering, June 1985.

Cameron S., "A Study of the Clash Detection Problem in Robotics", proc. IEEE conf. on Robotics and Automation, 1985, pp 488-493.

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

[C] **Canny, J. F.** *Collision Detection for Moving Polyhedra*, PAMI-8(2) (1986).

[C1] **Canny, J.F.** *The Complexity of Robot Motion Planning*, Ph.D. Thesis, MIT Department of Electrical Engineering and Computer Science (1987).

Canny, J.F. *Computing Roadmaps of Compact Semi-Algebraic Sets*, Intl. Workshop on Geometric Reasoning, Oxford, England, June (1986).

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

[CD] **Canny, J.F. and Donald, B. R.** *Simplified Voronoi Diagrams*, Discrete and Computational Geometry (to appear).

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

Chapman, D. *Planning for Conjunctive Goals*, MIT AI-TR 802 (1985).

Chistov A. L. and Grigoryev D. Y., "Complexity of quantifier elimination in the theory of algebraically closed fields", Lect. Notes Comp. Sci. 176, Springer Verlag, (1984).

Collins G. E. "Quantifier Elimination for Real Closed Fields by Cylindrical Algebraic Decomposition" Lecture Notes in Computer Science, No. 33, Springer-Verlag, New York, (1975), pp. 135-183.

Cutkosky, M., "Grasping and Fine Manipulation for Automated Manufacturing", Ph.D. dissertation, Carnegie-Mellon University, January, 1985.

Davis, E. and McDermott, D. *Planning and Executing Routes through Uncertain Territory*, Yale University, Dept. of Computer Science (1982).

Donald, B. R. *Motion Planning with Six Degrees of Freedom*, MIT AI-TR 791, Artificial Intelligence Lab. (1984).

Donald, B. R. *On Motion Planning with Six Degrees of Freedom: Solving the Intersection Problems in Configuration Space*, IEEE International Conference on Robotics and Automation, St. Louis, MO (1985).

[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).

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

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

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

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

Draper Laboratories, Fourth Annual Seminar on Robotics and Advanced Assembly Systems, Cambridge, Massachusetts, November, 1983.

Dufay, B., and J. Latombe, "An Approach to Automatic Robot Programming Based on Inductive Learning", in Brady, M., and R. Paul, *Robotics Research: The First International Symposium*, MIT Press, 1984.

Durrant-Whyte, H. *Concerning Uncertain Geometry in Robotics*, Intl. Workshop on Geometric Reasoning, Oxford, England, June (1986).

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

Erdmann, M. *On Motion Planning With Uncertainty*, MIT AI Lab, MIT-AI-TR 810 (1984).

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

Faltings, B., "A Theory of Qualitative Kinematics in Mechanisms", Computer Science Department, University of Illinois at Urbana-Champaign, UIUCDCS-R-86-1274; May, 1986.

Faltings, B., "Qualitative Kinematics in Mechanisms", Proc. IJCAI 87, Milano, 1987

Faverjon, B. *Obstacle Avoidance Using an Octree in the Configuration Space of a Manipulator*, Proc. IEEE Intl. Conf. Robotics, Atlanta (March 1984).

[STRIPS] **Fikes, R. and Nilsson, N.** *STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving*, Artificial Intelligence vol. 2 (1971).

Fortune, S., Wilfong, G., and Yap, C. 1986 (April 7 10, San Francisco, California). Coordinated Motion of Two Robot Arms. *Proceedings of the 1986 IEEE International Conference on Robotics and Automation*, pp. 1216 1223.

Gini, M. and Gini, G. *Towards Automatic Error Recovery in Robot Programs*, IJCAI-83 (1983).

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).

Grossman, D., and R. Taylor, "Interactive Generation of Object Models with a Manipulator", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 8, No. 9, September, 1978.

Hayes, P. *A Representation for Robot Plans*, 4th IJCAI (1976).

Hopcroft, J. E., Schwartz, J. T., and Sharir, M. 1984 On the Complexity of Motion Planning for Multiple Independent Objects; *PSPACE*-Hardness of the "Warehouseman's Problem." *International Journal of Robotics Research.* **3**(4):76–88.

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

Hungerford, T. W. *Algebra*, Springer-Verlag, New York GTM 73 (1974).

Hogan, N., "Impedance Control of Industrial Robots", *Robotics and Computer-Integrated Manufacturing*, Vol. 1, No. 1, 1984.

Inoue, H., "Force Feedback in Precise Assembly Tasks", MIT Artificial Intelligence Laboratory, AIM-308, August, 1974.

Khatib, O. *Real-Time Obstacle Avoidance for Manipulators and Mobile Robots*, Int. Jour. Rob. Res. vol. 5, No. 1, pp. 90-99 (1986).

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

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.

Kozen D., and Yap C. "Algebraic Cell Decomposition in NC", Proc IEEE symp. FOCS, (1985), pp. 515-521.

Laugier, C., "A Program for Automatic Grasping of Objects with a Robot Arm", *Eleventh Symposium of Industrial Robots*, Japan Society of Biomechanisms and Japan Industrial Robot Association, 1981.

Lee, D. T., and Drysdale, R. L., "Generalization of Voronoi diagrams in the plane," SIAM J. Comp. (10) (1981) pp. 73-87.

Lieberman, L., and M. Wesley, "AUTOPASS: An Automatic Programming System for Computer Controlled Mechanical Assembly", *IBM Journal of Research Development*, Vol. 21, No. 4, 1977, pp. 321-333.

Lozano-Pérez, T., "The Design of a Mechanical Assembly System", S.M. dissertation, MIT Department of Electrical Engineering and Computer Science, also AI-TR-397, MIT Artificial Intelligence Laboratory, 1976.

Lozano-Pérez, T. *Automatic Planning of Manipulator Transfer Movements*, IEEE Trans. on Systems, Man and Cybernetics (SMC-11):681-698 (1981).

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

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

Lozano-Pérez, T., "Motion Planning For Simple Robot Manipulators", *Third International Symposium on Robotics Research*, Paris, October, 1985.

Lozano-Pérez, "A Simple Motion Planning Algorithm for General Robot Manipulators," in Proceedings of Fifth National Conference for the American Association of Artificial Intelligence, Philadelphia, 1986, pp. 626-631.

Lozano-Pérez T., Jones J., Mazer, E., O'Donnell, P., Grimson, E., "Handey: A Robot System that Recognizes, Plans, and Manipulates", Proc. IEEE Int. Conf. on Robotics and Automation, 1987.

[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).

Lozano-Pérez, T., and Wesley, M. A. *An algorithm for planning collision-free paths among polyhedral obstacles*, Communications of the ACM (22):560-570 (1979).

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).

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

Mason, M.T. *Manipulator Grasping and Pushing Operations*, MIT AI Lab, MIT AI-TR-690 (1982).

Mason, M. T. 1986. Mechanics and Planning of Manipulator Pushing Operations. *International Journal of Robotics Research* **5**(3).

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

Mason, M. T. 1985 (March 25-28, St. Louis, Missouri). The Mechanics of Manipulation. *Proceedings of the 1985 IEEE Int. Conf. on Robotics and Automation*, pp. 544-548.

McDermott, D. *A Temporal Logic for Reasoning about Processes and Plans,* Cog. Sci. 6, pp. 101-55 (1982).

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.

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

Neivergelt, J., and Preparata, F. P. *Plane-Sweep Algorithms for Intersecting Geometric Figures,* CACM Vol. 25, no. 10 (1982).

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)

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)

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

Ohwovoriole, M., and B. Roth, "A Theory of Parts Mating For Assembly Automation", *Proceedings of the Robot and Man Symposium 81*, Warsaw, Poland, September 1981.

Paul, R., *Robot Manipulators*, MIT Press, Cambridge, Massachusetts, 1981.

Peshkin, M., "Planning Robotic Manipulation Strategies for Sliding Objects", Ph.D. dissertation, Department of Physics, Carnegie-Mellon University, 1986.

Peshkin, M., and A. Sanderson, "Reachable Grasps on a Polygon: The Convex Rope Algorithm", *IEEE Journal of Robotics and Automation*, Volume 2, Number 1, March, 1986.

Raibert, M., and J. Craig, "Hybrid Position/Force Control of Manipulators", *Journal of Dynamic Systems, Measurement, and Control*, No. 102, June, 1981, pp. 126-133.

Reif J., "Complexity of the Mover's Problem and Generalizations," Proc. 20th IEEE Symp. FOCS, (1979). Also in "Planning, Geometry and Complexity of Robot Motion", ed. by J. Schwartz, J. Hopcroft and M. Sharir, , Ablex publishing corp. New Jersey, (1987), Ch. 11, pp. 267-281.

Requicha, A. A. *Representation of Tolerances in Solid Modeling: Issues and Alternative Approaches*, Solid Modeling by Computers: From Theory to Applications; Plenum, N. Y. (1984).

Salisbury, J.K., "Active Stiffness Control of a Manipulator in Cartesian Coordinates", *IEEE Conference on Decision and Control*, Albuquerque, New Mexico, November, 1980.

Salisbury, J.K., "Kinematic and Force Analysis of Articulated Hands", Ph.D. dissertation, Stanford University, Department of Mechanical Engineering, 1982.

Segre, A. M., and G. DeJong, "Explanation-Based Manipulator Learning: Acquisition of Planning Ability Through Observation", *IEEE International Conference on Robotics and Automation*, St. Louis, March, 1985.

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

[SS] **Schwartz J. and Sharir M.**, "On the 'Piano Movers' Problem, II. General Techniques for Computing Topological Properties of Real Algebraic Manifolds," Comp. Sci. Dept., New York University report 41, (1982). Also in "Planning, Geometry and Complexity of Robot Motion", ed. by J. Schwartz, J. Hopcroft and M. Sharir, Ablex publishing corp. New Jersey, (1987), Ch. 5, pp. 154-186.

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

Simunovic, S. N., "An Information Approach to Parts Mating", Ph.D. dissertation, Department of Electrical Engineering, Massachusetts Institute of Technology, 1979.

Simunovic, S. N. 1975 (Sept. 22–24, Chicago, Illinois). Force Information in Assembly Processes. *Proceedings 5th International Symposium on Industrial Robots.* Bedford, U.K.: IFS Publications, pp. 415–431.

Shapiro, V. *Parametric Modeling and Analysis of Tolerances*, GM Research Lab. Rept. CS-460 (1985).

Srinivas, Sankaran *Error Recovery in Robot Systems*, Cal. Tech. Ph.D. Thesis, Computer Science (1977).

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

Tarski A., "A Decision Method for Elementary Algebra and Geometry" Univ. of Calif. Press, Berkeley, (1948), second ed. 1951.

Turk, M., "A Fine-Motion Planning Algorithm", *SPIE Conference on Intelligent Robots and Computer Vision*, Cambridge, Massachusetts, September, 1985.

Udupa, S., "Collision Detection and Avoidance in Computer Controlled Manipulators", Ph.D. dissertation, Department of Electrical Engineering, California Institute of Technology, 1977.

Udupa S., "Collision Detection and Avoidance in Computer Controlled Manipulators", Proc. 5th Int. Joint. Conf. on Art. Intell., Mass. Inst. Tech. (1977) pp 737-748.

Valade, J., "Automatic Generation of Trajectories for Assembly Tasks", *Sixth European Conference on Artificial Intelligence*, Pisa, Italy, September, 1984.

Ward, B. and McCalla, G. *Error Detection and Recovery in a Dynamic Planning Environment*, AAAI (1983).

Whitney, D., "Force Feedback Control of Manipulator Fine Motions", *Journal of Dynamic Systems, Measurement, and Control*, June, 1977, pp. 91-97.

Whitney, D., "Quasi-Static Assembly of Compliantly Supported Rigid Parts", *Journal of Dynamic Systems, Measurement, and Control*, Vol. 104, March 1982.

Whitney, D., "Historical Perspective and State of the Art in Robot Force Control", *IEEE International Conference on Robotics and Automation*, St. Louis, March, 1985.

Wilkins, D. E. *Domain-Independent Planning: Representation and Plan Generation*, Artificial Intelligence, Vol. 22 No. 3 (1984).

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

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