What Should a Roboticist Do Next? A Progress Report from the Cornell Computer Science Robotics Laboratory*

Bruce Donald
Dinesh Pai**
Patrick Xavier

TR 89-1053
November 1989

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

*This paper describes research done in the Computer Science Robotics Laboratory at Cornell University. Support for our robotics research is provided in part by the National Science Foundation under Grant IRI-8802390, by and Presidential Young Investigators award, and in part by the Mathematical Sciences Institute.
**This work was supported by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research Contract N00014-88-K-0591, ONR Grant N00014-89-J-1946, and NSF Grant DMC-86-17355.

What Should a Roboticist Do Next? A Progress Report from the Cornell Computer Science Robotics Laboratory

Bruce R. Donald

Dinesh K. Pai

Patrick Xavier

Computer Science Department

Cornell University

What Should a Roboticist Do Next? A Progress Report from the Cornell Computer Science Robotics Laboratory

Bruce R. Donald

Dinesh K. Pai

Patrick Xavier

Cornell University

Abstract:

The most important intellectual capital of a research laboratory is its coherent vision of science, research and progress. In this progress report, we identify key areas for research, describe the progress we have made in attacking these areas, and discuss our plans for future work.

Our primary goal is to bring robotics science closer to its goal of task-level planning. We approach this goal through a blend of theory, implementation, and experimentation. A major impediment to developing truly task-level robotic systems has been the very hard algorithmic problems that arise in task-level robot planning. We have identified several key areas on which to concentrate in developing new algorithmic technologies to crack these problems. Roughly speaking, these areas are:

1. *Basic research in compliant motion planning under uncertainty.* This research will result in algorithms and systems that can assemble mechanical parts using compliant motion strategies, despite uncertainty and errors in sensing, control, and modeling.

2. *Basic research in planning with full dynamics.* It is vital that robots execute tasks quickly, and take dynamics into account. Our research on *kinodynamic* planning provides the first *provably good approximation algorithms* for planning nearly time-optimal collision-avoiding paths that respect dynamics bounds.

3. *Basic research in design for assembly.* The design of mechanical devices and the planning to assemble them should not be independent activities. We introduce a new, fully algorithmic, combinatorially precise approach to designing devices so that they are easy to assemble and (optionally) hard to disassemble. Our analysis can be used to validate good designs, and can be iterated to generate improved designs.

Contents

1 Technical Abstract

Our primary goal is to bring robotics science closer to its goal of task-level planning. We approach this goal through a blend of theory, implementation, and experimentation. A major impediment to developing task-level robotic systems has been the very hard algorithmic problems that arise in task-level robot planning. We have identified several key areas on which to concentrate in developing new algorithmic technologies to leverage research on these problems. Planned projects focus on the following areas:

1. We will design and implement algorithms for generating and executing assembly strategies from task-level descriptions. We will build a hand-eye system which interleaves planning and execution of sensor-based compliant motion strategies.

   Our research will focus on algorithmic synthesis of compliant motion programs using the geometric constraints imposed by the task, and the explicit modeling of uncertainty and error in sensing, control, and the shape of the manipulated parts. To test our theory of task-level assembly, we will build a force-controlled, experimental assembly system for the Cornell robot. This system represents one of the few attempts to verify the adequacy of an algorithmic theory of compliant motion planning by testing the plans on a force-controlled robot.

2. It is vital to develop planning algorithms so that robots can automatically synthesize control trajectories subject to kinematic and dynamic constraints. In a recent breakthrough (reported in [CDRX]), we developed a new algorithm for optimal kinodynamic planning. This algorithm may be applied in robotics to generate minimal time, collision-avoiding trajectories, that respect dynamics bounds. While exact solutions to this problem are unknown (except in trivial cases) we have discovered the first provably good approximation algorithm for optimal kinodynamic planning. Our bounds on solution accuracy and running time are the first that have been obtained for optimal kinodynamic planning, which has been an open problem in computational robotics for over ten years. We propose to continue research and development on "kinodynamic" planning algorithms, and to implement and test them.

3. We will pursue an algorithmic theory of *design for assembly*. To this end we will develop and implement algorithms that can analyze and generate designs for objects so that they will be easy to assemble. In particular, we observe that real objects that robots might assemble are typically not rigid. For example, significant advances were made in the design of the IBM ProPrinter, by replacing traditional fasteners such as screws with plastic parts that simply snap together. Since we wish to be able to design and have our robots assemble such objects given task-level descriptions, we propose a systematic program for reasoning about and predicting their motions in contact. A precise, algorithmic theory of design and assembly for such objects has not been developed before. We will model the physics of interaction of these "snap-fastener"-type parts, and proceed to develop combinatorially precise algorithms for predicting the motion of flexible objects near and in contact with their mating parts. We will also compute the forces required to mate the parts. Finally, we will explore the use of our analysis algorithm in an approach to " *design as search*", in which we modify and improve an existing design by changing its geometry incrementally and applying our analysis algorithm.

Basic research in these three areas can have a significant impact on our ability to develop and implement task-level robotic systems. We view robotics as the science of representing, reasoning about, and manipulating physical objects. Basic research in robotics science can infuse new techniques into mainstream robotics, and greatly enhance the usefulness and practicality of robots.

What Should a Roboticist Do Next? A Progress Report from the Cornell Computer Science Robotics Laboratory

Bruce R. Donald

Dinesh K. Pai

Patrick Xavier

Cornell University

2 Introduction

The most important intellectual capital of a research laboratory is its coherent vision of science, research and progress. In this progress report, we identify key areas for research, describe the progress we have made in attacking these areas, and discuss our plans for future work.

Our primary goal is to bring robotics science closer to its goal of task-level planning. We approach this goal through a blend of theory, implementation, and experimentation. A major impediment to developing truly task-level robotic systems has been the very hard algorithmic problems that arise in task-level robot planning. We have identified several key areas on which to concentrate in developing new algorithmic technologies to crack these problems. Roughly speaking, these areas are:

1. *Basic research in compliant motion planning under uncertainty.* This research will result in algorithms and systems that can assemble mechanical parts using compliant motion strategies, despite uncertainty and errors in sensing, control, and modeling.

2. *Basic research in planning with full dynamics.* It is vital that robots execute tasks quickly, and take dynamics into account. Our research on *kinodynamic* planning provides the first *provably good approximation algorithms* for planning nearly time-optimal collision-avoiding paths that respect dynamics bounds.

3. *Basic research in design for assembly.* The design of mechanical devices and the planning to assemble them should not be independent activities. We introduce a new, fully algorithmic, combinatorially precise approach to designing devices so that they are easy to assemble and (optionally) hard to disassemble. Our analysis can be used to validate good designs, and can be iterated to generate improved designs.

More specifically, our planned projects focus on the following areas:

1. We will design and implement algorithms for generating and executing assembly strategies from task-level descriptions. We will build a hand-eye system which interleaves planning and execution of sensor-based compliant motion strategies.

   Our research will focus on the following issues:

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

(b) The synthesis of compliant motion programs using the geometric constraints imposed by the task.

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

(d) The continued 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.

(e) To test our theory of task-level assembly, we will continue work on building an experimental force-control hand-eye system for the Cornell robot. This system represents one of the few attempts to verify the adequacy of an algorithmic theory of compliant motion planning by testing the plans on a force-controlled robot.

In task-level assembly, we propose several kinds of basic research on fundamental, theoretical issues in compliant motion planning with uncertainty. We also intend to implement these theories: Our laboratory setup permits us to simultaneously conduct research on incorporating our theories into task-level systems.

2. Imagine a space robot is performing an assembly or repair. A part is dropped; unless the part can be quickly reacquired, it will drift out of the robot's workspace. The robot must plan to move and grasp the part as quickly as possible; in addition, it must not spend too long computing what the motion plan will be. The former problem is one of optimal control; the latter, of computational complexity. Our work links control and complexity theory, and provides algorithms that quickly generate close-to-optimal control programs. We propose to extend our theory, develop new algorithms, and to implement and test them.

   It is vital to develop planning algorithms so that robots can automatically synthesize control trajectories subject to kinematic and dynamic constraints. In a recent breakthrough (reported in [CDRX]), we developed a new algorithm for optimal kinodynamic planning. This algorithm may be applied in robotics to generate minimal time, collision-avoiding trajectories, that respect dynamics bounds. While exact solutions to this problem are unknown (except in trivial cases) we have discovered the first provably good approximation algorithm for optimal kinodynamic planning. We propose to continue research and development on "kinodynamic" planning algorithms. These algorithms

   (a) Produce solution plans that respect dynamics constraints (velocity, torque, and acceleration bounds), and avoid obstacles by a speed-dependent "safety margin."

   (b) Produce solutions that are "fast" to execute (provably close to time-optimal) and "fast" to generate (polynomial time to plan);

   (c) Provide a link between control- and complexity- theory by explicitly and analytically trading off the computational complexity of planning time versus optimality in terms of: (i) execution time of the trajectory and (ii) strictness in observing the safety margin.

   In short, they represent the first algorithms for near-optimal kinodynamic planning in which it is possible to (i) bound the goodness of the approximate solution by an error term ϵ, (ii) polynomially bound the running time (complexity) of the algorithm, and (iii) express the computational complexity as a (polynomial) function of $(\frac{1}{\epsilon})$.

3. In collaboration with my post-doctoral associate Dinesh Pai, we will pursue an algorithmic theory of *design for assembly*. To this end we will develop and implement algorithms that can analyze and generate designs for objects so that they will be easy to assemble. In particular, we observe that real objects that robots might assemble are typically not rigid. For example, a Sony Walkman

is made of plastic parts that snap together. Significant advances were made in the design of the IBM ProPrinter, by replacing traditional fasteners such as screws with plastic parts that simply snap together. The reason these plastic parts snap together is that they are *flexible*: more precisely, they are *passively compliant.* This means that when the parts are brought together and an external force applied, the parts deform in a prescribed way. More interestingly, the force required to mate two parts may be much less than the force required to take them apart.

Since we wish to be able to design and have our robots assemble such objects given task-level descriptions, we must have a systematic program for reasoning about and predicting their motions in contact. To this end, we will first make precise a sufficiently powerful notion of flexibility to model the objects above, which encompasses several important and complicated mechanisms in mechanical design and automated assembly: snap-fasteners, latches, ratchet and pawl mechanisms, and escapements. We will model the physics of interaction between the flexible parts and the environment (including their mating parts). Then, using these tools, we will proceed to develop combinatorially precise algorithms for predicting the motion of a flexible object near and in contact with its mating part. We will also compute the forces required to mate the parts. Finally, we will explore the use of our analysis algorithm in an approach to *" design as search,"* in which we modify and improve an existing design by changing its geometry incrementally and applying our analysis algorithm.

Roadmap of this Report

Basic research in these three areas can have a significant impact on our ability to develop and implement task-level robotic systems. The remainder of this report is divided into three sections, corresponding to

3 *Basic research in compliant motion planning under uncertainty.*

4 *Basic research in kinodynamic motion planning.*

5 *Basic research in design for assembly.*

These three sections comprise the technical agenda of this report, and sketch the progress we have made so far in each area. We have split our proposals for future research into three parts, so that they could be tightly coupled to the technical content. At the end of each section is a subsection proposing future work in that area. These sections are, respectively, 3.10, 4.6, and 5.9. It is in these sections that we outline the next steps in our research agenda.

3 An Algorithmic Approach to Task-Level Assembly Planning

3.1 Introduction

For the past seven years, I have been interested in building robot planning systems that can function at the task-level. A task-level specification of a robot plan might have the form, *Put together this disk rotor assembly.* The planner is given geometric models of the parts, and geometric or analytic models of the robot dynamics. Beyond this, the specification, or input to the planner does not mention the specific kinematic and dynamic constraints that the robot must obey; these are determined by the planner using geometrical computation. Typically, there are additional constraints that the planner must also obey, for example, *Construct a robot plan that is robust in the face of uncertainty and error. Use compliant motion where appropriate to reduce uncertainty.* The goal of a task-level planner is to take a task-level specification and to produce a runnable robot program—one which is fully specified in terms of force-control, kinematics, and dynamics—that can accomplish the task.

Major advances in task-level planning can enable robotics to achieve its full potential in the assembly domain. Today, even existing robots cannot be exploited to their full capacity. For example, assembly tasks require compliant motion; however, compliance requires force-control, and such force-control motion strategies are quite difficult for humans to specify. Furthermore, robot assembly programs are very sensitive to the details of geometry. Finally, reprogramming a general purpose robot for a new assembly task can take time on the order of man-months. For these reasons, we have been working on the automatic synthesis of motion strategies for robots. Much work is required to reduce such current task-level technologies to practice.

Research in task-level planning is often characterized as *theoretical robotics.* There are several reasons for this; the first is that much of the work has been concerned with constructing a *theory* of planning. In other words, the computational problem "task-level planning" is not well-specified. Much of our work lies in specifying the computation precisely. Second, given some sort of decomposition of task-level planning into "planning problems", one is immediately driven to ask, *What are the algorithms for these problems? Can plans, in general, be computed? How efficiently can planning algorithms run?* Historically, the nature of these questions has led researchers to apply tools from theoretical computer science, computational geometry, and algebra. Much of this work, while foundational, must seem rather distant from the practitioner and the physical robot.

Recently, a great deal of attention has been focused on a particular robotics problem, called the find-path, or generalized movers' problem. In this problem, we ask the purely kinematic question, can a robot system be moved from one configuration to another, without colliding with obstacles? (See, eg., [LoP, BLP, Don87a, Can]). This is a nicely-defined mathematical problem, and, after much research, at this point its computational complexity is precisely known.

In fact, the neatness of this problem is deceptive, so much so that this formal problem has even been called "the" motion planning problem. From a task-level viewpoint, there is much hidden in the statement "Can the robot system be moved...?" Specifically, the find-path problem assumes that the robot has a perfect control system that can exactly execute the plan, and that the geometric and analytic models of the robot and obstacles are exact.

In reality, of course, robot control systems are subject to significant uncertainty and error. Typical robots are also equipped with sensors—force sensors, kinesthetic position sensors, tactile sensors, vision, and so forth. However, these sensors are also subject to significant uncertainty. Finally, the geometrical models of the robot and the environment (parts, obstacles, etc.) cannot be exact—they are accurate only to manufacturing tolerances, or to the accuracy of the sensors used to acquire the models. Uncertainty is not a mere engineering detail; in particular, it is characteristically impossible to "patch" these perfect plans in such a way that they will function once uncertainty comes into play. Uncertainty is an absolutely fundamental problem in robotics, and plans produced under the assumption of no uncertainty are meaningless. What is needed is a principled theory of planning in the presence of uncertainty. Such a theory must not only be computational, but must also take uncertainty into account *a priori*. The overlap with exact motion planning algorithms can be stated roughly as follows: exact kinematic planning algorithms provide a computational-geometric theory of holonomic constraints. In motion planning with uncertainty, we exploit compliant motion—sliding on surfaces—in order to effect a "structural" reduction in uncertainty. Such compliant motion plans can be synthesized from a computational analysis of the geometry of the holonomic constraints.

3.2 Uncertainty and Compliant Motion

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 has 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?

This is perhaps the most fundamental problem in robotics today. We call it the problem of *motion planning with uncertainty.*

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

Compliant motion occurs when a robot is commanded to move into an obstacle, but rather than stubbornly obeying its motion command, it complies to the surface of the obstacle. Work on compliant motion[1] attempts to utilize the task geometry to plan motions that reduce the uncertainty in position by maintaining sliding contact with a surface. Plans consisting of such motions can be designed to exploit the geometry of surfaces around the goal to guide the robot. By computing "preimages"[2] of a geometrical goal in configuration space, guaranteed strategies can be synthesized geometrically: We call this a *geometrical theory of planning.* The first results in this theory begin with Lozano-Pérez, Mason, and Taylor (or [LMT]), with subsequent contributions by Mason [Ma2], Erdmann [Erdmann] and Donald [D]. This research has led to a theoretical computational framework for motion planning with uncertainty, which we denote [LMT,E,D]. See [Buc, EM, Bro, CR] for other allied work.

The [LMT,E,D] framework begins by observing that the use of active compliance enables robots to carry out tasks in the presence of significant sensing and control errors. Compliant motion meets external constraints by specifying how the robot's motion should be modified in response to the forces generated when the constraints are violated. For example, contact with a surface can be guaranteed by maintaining a small force normal to the surface. The remaining degrees of freedom (DOF)—the orthogonal complement of the normal-space—can then be position-controlled. Using this technique, the

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

[2] The *preimage* of a goal [LMT] is the set of configurations from which a particular commanded compliant motion is guaranteed to succeed.

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 (see fig. 1) may be "topologically" very different, in that different surfaces of the peg and hole are in contact. Under these circumstances, the programmer must employ a combined strategy of force and position control that guarantees reaching the desired final configuration from all the likely initial configurations. We call such a strategy a *motion strategy.*

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

Note that compliant motion planning with uncertainty is significantly different from motion planning with perfect sensing and control along completely-known configuration space obstacle boundaries [Kou, HW, BK]. The first difference is physical:

- From a practical point of view, the motion-in-contact plans generated under the assumption of perfect control cannot ever be executed by a physical robot using position control alone.

The second difference is combinatorial:

- The planning of motions in contact with perfect control has the same time-complexity as planning free-space motions; that is, it can be done in time $O(n^r \log n)$ for r degrees of freedom and n faces or surfaces in the environment [C1]; the exponent is worst-case optimal. However, prior to [Donald 87b,88a], there are no upper bounds for planning compliant motions *with* uncertainty. However, for r fixed at 3, the problem is hard for non-deterministic exponential time [CR]. We showed in [Donald 87b,88a] that the planar case, somewhat surprisingly, turns out to be tractable.

The physical difficulty in fact motivates our work. This experimental viewpoint is manifest in [Donald 87b,88b; Jennings, Donald and Campbell]. It is also possible to adopt a more theoretical perspective; in [Donald 88a; Briggs 89] we concentrate on the geometric and combinatorial aspects of the problem.

3.3 Example: Synthesizing Guaranteed Plans

Error Detection and Recovery (EDR) strategies arose as a response to certain inadequacies in the "guaranteed success" planning model. Hence, in order to study EDR, it is first necessary to understand the structure of guaranteed compliant motion strategies, and how they may be synthesized. To this end, in this report, we wish to give a flavor for the kinds of issues that arise in planning guaranteed motion strategies under uncertainty. We examine a very special case—the planar polygonal case. (That is, the workspace environment ("parts" and "obstacles") are polygonal). First, we will briefly develop a simple dynamic model that is adequate for this situation. Next, we carefully define the computational problem of synthesizing planar guaranteed strategies under uncertainty in sensing and control, and generalize our definition to include uncertainty in the shape of the parts and obstacles. Finally, we hint at the types of computational techniques employed.

For this very simple planning problem, we find that motion plans with a simple structure suffice. This permits us to illustrate by a specific example the situation that we investigated in [Donald 87b, 88a, 88b, 89] for fairly arbitrary plans. Together with the problem of constructing guaranteed compliant motion strategies under uncertainty, we will consider a rather restricted class of strategies, namely those that terminate by sticking on a surface.

3.4 Dynamic Model

Compliant motion is only possible with certain dynamic models. We will employ the generalized damper model [W, Ma]. We assume that the environment is polyhedral, and that it describes the configuration space of the robot, so that the robot is always a point. The planned path consists of r successive motions in directions $v_1, \ldots, v_r$. Each motion terminates when it sticks, due to coulomb friction, on some surface in the environment. Because of control uncertainty, however, the robot cannot move with precisely velocity v_i on the i^{th} motion. Instead, it moves with velocity v_i^{free}, which lies in a cone of velocities $B_{ec}(v_i)$ about v_i. The boundaries of the cone form an angle of ϵ_c with v_i. ϵ_c is called the *control uncertainty*, and $B_{ec}(v_i)$ the *control uncertainty cone about* v_i. It is, in fact, equivalent to regard ϵ_c as specifying that v_i^{free} lies within a ball about v_i in velocity space.

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

The i^{th} motion terminates by sticking on a surface when the velocity v_i^{free} in $B_{ec}(v_i)$ points into the negative coulomb friction cone on a surface (see fig. 2). Thus sticking on a surface can be non-deterministic. We will assume that motion i can terminate on any reachable surface for which some velocity $v_i^{free} \in B_{ec}(v_i)$ is inside the negative friction cone. Sticking termination is motivated by the fact that a robot with a force-sensing wrist can easily recognize sticking and robustly terminate the motion.

To test whether sticking is possible on some set of (say, goal) edges, we simply perform a geometric cone intersection on each edge. Sticking is possible when the intersection of the cone of velocity uncertainty and the negative friction cone have a non-trivial intersection. Since determining the possibility (or necessity) of sticking reduces to a simple cone intersection, which may be done in constant time per edge, in this preface we will focus on the more difficult issue of computing reachability. Representing friction in our planar polygonal configuration space is easy; see fig. 2. However, the more general question of representing friction in configuration spaces with rotations is subtle; see [Erdmann, BRS].

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

3.5 Definitions

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

We now pose three problems (see fig. 4):

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

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

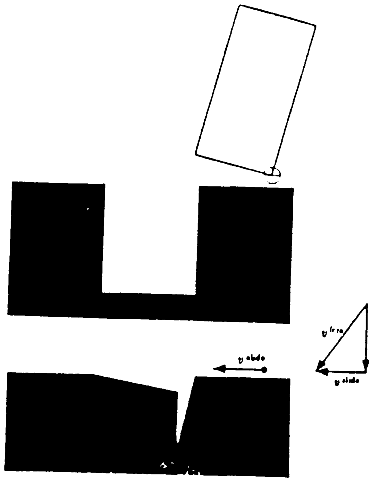

Figure 1: (a) Peg in hole environment. (b) Configuration space, showing the motion of the reference point during compliant motion.

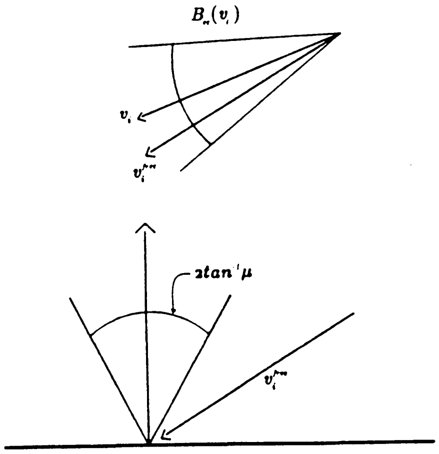

Figure 2: The control uncertainty cone B_{ec} about a nominal commanded velocity v_i. The coulomb friction cone edge makes an angle of $\tan^{-1}\mu$ with the normal. The velocity v_i^{free} will result in sliding on the surface.

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

3.6 Model Error

We now introduce model error into the picture. 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 can now state our fourth problem. We parameterize our family of geometries by the parameter space $\alpha_1, \ldots, \alpha_k$, so that $\mathcal{P}(\alpha_1, \ldots, \alpha_k)$ denotes a particular geometry, and each α_i lies within some given interval. Note that this parameterization need not be "continuous". We wish to find a plan which will succeed for all geometries $\mathcal{P}(\alpha_1, \ldots, \alpha_k)$.

Problem 4: Compliant Motion Planning with Uncertainty and Geometric Model Error Given $(R, \mathcal{P}(\alpha_1, \ldots, \alpha_k), \epsilon_c, \mu, G)$, and an integer r, find a sequence of r motions such that, for all possible values of $\alpha_1, \ldots, \alpha_k$, each motion terminates in sticking, and the final motion terminates in the goal. Or, if no such r-step strategy exists, then say so.

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.

3.7 Backprojections

We now sketch, for a specific example, a computational approach to the compliant motion planning problem with uncertainty. In [Donald 87b, 88a;b, 89] we have mounted a more systematic attack using similar, albeit considerably generalized methods.

Erdmann [Erdmann] has shown that in the plane, when G is a single edge of the polygonal environment $\mathcal{P}$, then the one-step verification problem (2) can be done in time $O((n+c)\log n)$, where c is the number of intersections encountered by a planar arrangement algorithm.

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

The backprojection $B_\theta(G)$ of a goal G (with respect to a commanded velocity v_θ^*) consists of those configurations guaranteed to enter the goal (under v_θ^*).[3] That is, the backprojection is the set of all positions from which all possible trajectories consistent with the control uncertainty are guaranteed to reach G. See fig. 3. The terms "preimage" and "backprojection" come from viewing motions as "mappings" between subsets of configuration space. Hence the backprojection of a goal is the set of configurations from which a particular commanded compliant motion is guaranteed to succeed. [LMT] envisioned a back-chaining planner that recursively computes preimages of a goal region. Successive subgoals are attained by motion strategies. Each motion terminates when all sensor interpretations indicate that the robot must be within the subgoal.

See fig. 4. Here is the key point about backprojections: Given $(R, \mathcal{P}, \epsilon_c, \mu, G, v_\theta^*)$, the one-step verification problem (2) reduces to testing set containment, i.e., that

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

Erdmann showed that when G is a single edge of a planar environment $\mathcal{P}$, then $B_\theta(G)$ has size $O(n)$ and can be computed as follows (see fig. 5):

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

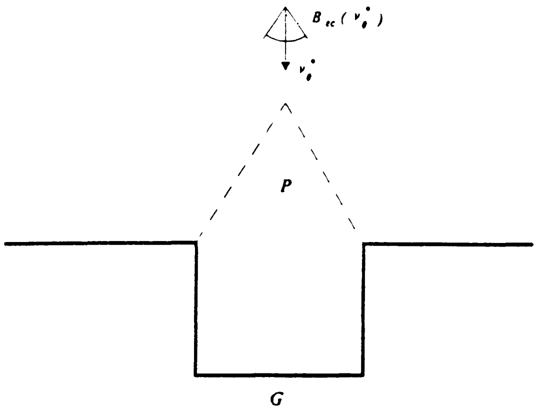

Figure 3: The goal is the region G. Sliding occurs on vertical surfaces, and sticking on horizontal ones. The commanded velocity is v_θ^*, and the control uncertainty is $B_{ec}(v_\theta^*)$. The *backprojection* of G with respect to θ is the region P.

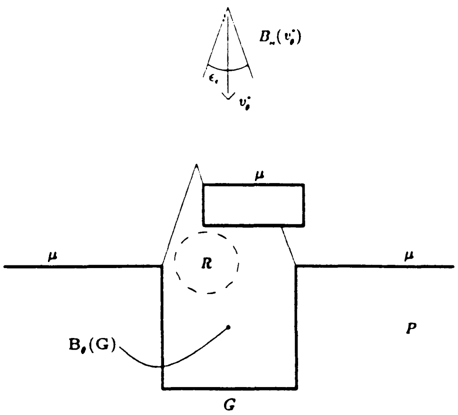

Figure 4: A compliant motion planning problem with goal G, start region R, control uncertainty ϵ_c, obstacles $\mathcal{P}$, and coefficient of friction μ. v_θ^* is a solution which is guaranteed to succeed.

(a) Find all vertices in the environment where sticking is *possible* under v_θ^*.

(b) At each of these vertices, erect two rays, parallel to the two edges of the *inverted* velocity cone $-B_{ec}(v_\theta^*)$.

(c) Compute the arrangement from the environment plus these additional $O(n)$ constraints.

(d) Starting at the goal edge, trace out the backprojection region.

An excellent exposition of Erdmann's algorithm can be found in [E]. With John Canny, we implemented a plane-sweep algorithm for backprojections from general polygonal goals. The idea is similar, and interested readers may find details in chapter V. While in general the complexity of these algorithms is $O((n + c)\log n)$, both methods take time $O(n \log n)$ and space $O(n)$ when the goal has $c = O(n)$ intersections with $\mathcal{P}$.

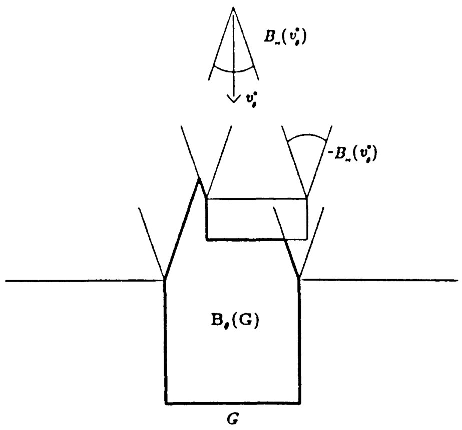

Figure 5: An illustration of Erdmann's algorithm for computing the backprojection $B_\theta(G)$ of a goal G in the plane.

This sketch of the algorithm for verifying a compliant motion strategy gives some flavor for the kind of solution we desire. Consider the remaining problems. The trick to obtaining a computational solution to problem (1) lies in considering all possible backprojections (for all possible θ) simultaneously, and choosing θ that are suitable. In [Donald 87b,88a, Briggs 89] we show that it is possible to do this efficiently, and also provide exact algorithms for problems (3) and (4) as well.

Finally, we must note that there are many variants and facets of the guaranteed compliant motion planning problem; these we address in [Donald 87b, 88a;b, 89, 90]. For these problems, we also gave computational approaches.

3.8 Error Detection and Recovery

A task-level planning system requires 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.* I believe that the chief contribution of our work has been to define and develop a theory of EDR. Perhaps the simplest way to view EDR is as a generalization, or extension, of guaranteed geometrical planning theories. Of course, much work is required to make this notion precise, and useful; that is the bulk of the work in our monograph [Donald 89], in which in effect, we proposed a "new" theory of planning.

Our approach to EDR observes that there are certain inadequacies with the "guaranteed success" planning model. It is simply 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 turn to 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.

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, our 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, called LIMITED, synthesizes robot control programs with built-in sensor-based EDR.

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 give a constructive, geometric definition for EDR strategies and show 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.

3.9 Experimental Results

Since 1987, my group at Cornell has implemented an experimental force-control system on our PUMA robot. In particular, we implemented an approximate generalized damper. Our PUMA 560 robot arm has 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. We closed the servo loop outside of VAL-II to obtain a servo rate we estimate as approximately 20-30 hz. Probably due to the inherent passive compliance in the joints and wrist, the robot can maintain compliant contact between a (gripped) steel bolt and an aluminum surface at speeds of 1-3 inches per second.

This experimental system was sufficiently stable to perform experiments executing the plans generated by LIMITED. In particular, we executed the plans and we performed all the assemblies shown in [Donald 87b, 89], chapter I, figs. 2-13. Again, the interesting thing about these plans is that they were generated by a machine, from task-level descriptions. Thus we were able to build a small slice of a task-level system, right down to the physical robot [Donald 88c, 90; Jennings, Donald and Campbell].

The peg-in-hole plans worked very well, even with tight (1 mm) clearances. The gear-meshing plans also worked, although we found it somewhat tricky to implement the failure mode analysis "oracle" that determines when sticking or breaking contact occurs using force- and position-sensing ([Donald 87b, 88c, 89, 90; Jennings, Donald and Campbell]). For both types of plans, we found that there were unmodelled effects (impact, bouncing) that could have an effect on plan execution, and unmodelled constants (the damping constants, the amount of force to maintain for sliding on a surface, the force termination thresholds) that must be chosen by the runtime executive. A more advanced planner would take these effects and constants into account. A more sophisticated runtime force-control plan executor might adaptively adjust these parameters to a local optimum. In addition, from an experimental viewpoint, much research is needed on the implementation and choice of termination predicates.

3.10 Research Issues and Proposed Research

We intend to pursue research in task-level assembly planning along the lines described above. In particular, we intend to follow the following directions:

1. To test our theory of task-level assembly, we will continue work on building an experimental force-control system for the Cornell robot. This system represents one of the few attempts to verify the adequacy of an algorithmic theory of compliant motion planning by testing the plans on a force-controlled robot.

2. The planner LIMITED was constrained to a world with two degrees of motion freedom, and could cope with several degrees of freedom of model error. It is vital that our assembly theory be

implemented and tested in 3D, and extended to handle rotations. To attain these goals:

(a) We have developed the first combinatorially precise backprojection algorithms [Donald 88a, Briggs 89] that can choose the control parameters. We will attempt to extend these techniques to configuration space with rotations. This will require solving a myriad of mathematical and computational problems: for example, it it not known in general whether the backprojection in $\Re^2 \times S^1$ (with holonomic constraints) is algebraic. Work in this area could provide a computational alternative to "slice-approximation" techniques (the current state of the art; eg, [Erdmann, Donald 89]), and may have an impact on the general problem of computing the trajectory envelope (forward projection) of a dynamical system specified by a parametric family of vector fields. Our domain is considerably simpler than this general problem, but we feel there are links between our approach and dynamical systems theory. (See also section 4 of this report!) The first stage would be to develop a precise representation for the backprojection and to decide containment of a start region in it. We also hope to extend our work to multi-step strategy generation, and other types of control.

(b) We intend to develop a new planning and execution system which explores the roles of history, goal recognizability, rich sensor domains, and planning/execution interleaving in task-level assembly. We will conduct both theoretical and experimental research on these fundamental issues. Our motivation is as follows:

LIMITED is a forward-chaining, multi-resolution planner. In order to forward chain, it computes forward projections. (One may think of the forward projection as the outer envelope of all possible trajectories resulting from the controls in a plan step). Let us view the termination predicate as a function on sensors, that recognizes when a goal has been entered, and returns *which* goal has been attained at that time. Naturally, the termination predicate may also know the start region, the geometry of the environment, and the controls. Given the forward projection and a termination predicate, LIMITED then computes the "push-forward" of the motion—all places where the motion can terminate.

In LIMITED's domain, computation of the forward projection was conceptually complicated but computationally efficient ($O(n \log n)$ time and $O(n)$ space by [Donald 87b,88a]). Computation of general push-forwards is a research issue (see [Donald 87b, 88b, 89, 90]) but for simple types of termination such as "move until sticking occurs", both push-forwards and the analogous real-time termination predicates in the robot control system are easy to implement, and quite robust in practice. In fact, for this class of simple termination predicates, the push-forward can be computed from the forward projection in linear time; furthermore, the push-forward is always a subset of the obstacle edges. Finally, note that these computations are also fast in practice [Donald 87b, 88b, 89, 90].

However, in the configuration space $\Re^2 \times S^1$, no better algorithms than slice-approximations are yet known for forward projections (although we hope to develop one; see above). In $\Re^3$, forward projections can have exponential size, and deciding containment in one is NP-hard [CR].

We observe that for suitably restricted termination predicates, in 3D the effect of applying the forward projection mapping is to obtain an exponential-sized combinatorial-geometric object. The subsequent application of the push-forward operator returns a subset of the forward projection; however, this subset is contained in a linear number of faces. Hence, in higher dimensions, it appears that computing reachability is expensive, while computing *recognizability* (where a motion can terminate) appears to remain computationally feasible. An obvious approach is suggested: implement a planner that deemphasizes the role of reachability, and emphasizes the role of recognizability in both planning and execution. ***Designing and implementing such a planning algorithm is a key step in our basic research agenda.***

For example, we propose to approximate the forward projection in higher dimension by simply using "octants" of the workspace. More rigorously, one could investigate polynomial-size approximations to the forward projections, and imagine an output-sensitive algorithm which switched to a bounding, conservative approximation when the output size (of the forward projection) became too large. Using "weak" approximations to reachability is conceptually easy. However, the issue of making recognizability a "strong" partner in the planning process is considerably more complicated. More specifically, we observe the following point (3):

3. One may envision LMT and EDR strategies as a "tree" of pairs; each pair is a commanded damper motion and a termination predicate. At a node, the damper motion is commanded, and the successor node is chosen depending on the subgoal returned as "attained" by the termination predicate. To generate such a tree ahead of time, the planner must, in effect, consider a large portion of the tree, together with all its branchings—this is essentially the result of the lower bounds of [CR], who showed that LMT planning is hard for NEXPTIME.

   We hope to avoid the worst-case complexity of planning by combining *interleaving* and the use of *history*. Interleaving of planning and execution permits the robot to try a branch of the tree and see where the motion takes it, instead of exploring all subtrees of plans. If sensing is accurate enough to disambiguate which subgoal has been achieved, then this is a great advantage. We believe that for our robot, sensing is accurate enough to provide a lot of pruning; of course this hypothesis must be experimentally verified. In the assembly domain, it is also necessary to avoid catastrophic failures as branches are explored; we hope that our theory of EDR can facilitate the computational instantiation of this principle.

   A major tool for recognizability is the *history* of sensor values, and the history of controls and initial conditions encountered in planning (see, for example, [Erdmann]). A natural consequence of augmenting the role of recognizability in planning is to place a greater burden on the maintenance and interpretation of history. This is probably one of the most important research problems in fine-motion planning. Roughly speaking, the environment can be partitioned into regions that are statically-distinguishable (using current sensor values alone). When history is employed, this partition (of regions that are distinguishable using history as well) becomes much finer, although it is harder to compute. We wish to explore this computation, and how history can be used to recognize subgoals, and to select controls for real-time (interleaved) executions of plan subtrees. For example, it is vital that when one of these distinguishable regions is reached, that we record what controls we execute next. If the plan "loops" and the region is reached and recognized again, then different controls can be selected. Many practical and theoretical issues arise at this point:

   (a) Can such a non-deterministic strategy be declared "complete" in any sense?
   (b) How will it work in practice?
   (c) Are there probabilistic arguments about eventual convergence?
   (d) How should history be computed, represented, and interpreted?
   (e) What computational burden does this (history) place on the real-time executive?
   (f) While the planner is inherently a "global" entity–in that it generates control strategies based on global constraints—the question remains how exactly to choose control strategies based on "local" information. That is, the planner is in some sense permitted only a bounded horizon, and so the interleaved execution has a "local" flavor. How can history alleviate this locality? How can cascaded local control strategies be made to converge by interleaving planning?

(g) What is the realm of competence of such an interleaved planner using "rich" history? This requires a careful experimental analysis. Can some sort of speedup be proved, theoretically?

(h) Suppose an assembly is successfully accomplished (and recognized) after an interleaved planning/execution session. If we want to replicate the assembly, what plan can be "deduced" from the "trace" of the previous assembly, given bounds on sensing and control uncertainty?

These problems are very challenging; many lie at the heart of task-level fine-motion planning. We propose to theoretically extend our EDR framework as described above, build the interleaved task-level assembly system, and address the fundamental issues raised in this section.

4. Finally, in our research efforts to emphasize the role of recognizability and history in fine-motion planning, we wish to investigate richer sensor domains. In particular, we wish to incorporate visual sensing into our task-level force-controlled assembly system, to build a hand-eye system for assembly. To this end, we will collaborate with Prof. Dan Huttenlocher at Cornell. In particular, we will investigate the integration of visual information into the EDR framework, and how comparatively "rich" sensing such as vision can, together with history, provide powerful tools for recognizability. We will contrast these vision-plus-sensor-based strategies with the relatively sensorless strategies of LIMITED, and the totally sensorless strategies of [Erdmann and Mason]. We will characterize planners for "rich" sensor-based fine-motion planning. We will investigate "task-directed" sensing, where visual sensing is guided by expectations of the planner and run-time executive, and "active" vision, where information from the planner's forward-projections and push-forwards can be used to electronically or mechanically "focus" the attention of the vision system on a particular small portion of the image. Given expectations of "what to look for" (in terms of the the recognizability criterion's termination predicate) and "where to look for it" (in terms of push-forwards), combinatorial local feature-based, model-based matching algorithms can be run on comparatively small image segments to achieve high performance, and (hopefully) good recognizability. Finally, there are many difficult and challenging issues involved in integrating visual sensing into the EDR framework; we hope to develop a coherent, algorithmic theory of visual recognizability termination predicates in terms of task-level assembly.

Please Note: this section describes only our proposed research in motion planning with uncertainty. At the end of each section is a subsection proposing future work in that area. These sections are, respectively, 3.10, 4.6, and 5.9. It is in these sections that we outline the next steps in our research agenda.

4 Algorithms for Kinodynamic Planning in Robotics

4.1 Brief Technical Statement of Problem under Study

We consider the following problem: given a robot system, find a minimal-time trajectory from a start position and velocity to a goal position and velocity, while safely avoiding obstacles and respecting dynamics bounds. In [CDRX] we developed a theoretical, provably good approximation algorithm for the minimum-time trajectory problem for a robot system with decoupled dynamics bounds. This algorithm differs from previous work in three ways. It is possible (1) to bound the goodness of the approximation by an error term ϵ; (2) to polynomially bound the running time (complexity) of our algorithm; and (3) to express the complexity as a polynomial function of the error term. Hence, one supplies the algorithm with the geometric obstacles, dynamics bounds, and the error term ϵ. The algorithm returns a solution that is ϵ-close to optimal and promises to spend only a polynomial (in $(\frac{1}{\epsilon})$) amount of time computing the answer.

In [DX1,DX2,DX3], we present results for robots from two categories: (1) robots that obey decoupled dynamics bounds and (2) robots that obey coupled dynamics bounds.

For robots in the first category, we reanalyzed the [CDRX] algorithm. We halved the exponent in the polynomial bounds, and at the same time showed that the algorithm yields a better approximation than previously reported. The new analysis also includes extensions to the earlier algorithm needed in practice. These new results indicate that an implementation of the theoretical algorithm is reasonable. In [DX1,DX2,DX3], we report on a preliminary implementation of the extended algorithm and experiments.

For robot with coupled dynamics bounds we first proved a generalized trajectory-tracking lemma. Using this result we described polynomial-time approximation algorithms for Cartesian robots obeying L_2 dynamics bounds and open kinematic chain manipulators with revolute and prismatic joints; the latter class includes most industrial manipulators.

4.2 Introduction

The *kinodynamic planning problem* [CDRX] is to synthesize a robot motion subject to simultaneous kinematic constraints (such as avoiding obstacles) and dynamics constraints (such as modulus bounds on velocity, acceleration, and force). A kinodynamic solution is a mapping from time to generalized forces or accelerations. The resulting motion is governed by a dynamics equation. A long-standing robotics problem has been to synthesize *time-optimal* kinodynamic solutions, by which we mean solutions that require minimal time and respect the kinodynamic constraints.

While there has been a great deal of work on this problem, there are no algorithms to generate exact solutions, with the exception of the one-dimensional case. Furthermore, lower bounds on the complexity of the exact problem indicate that it is computationally intensive. It is thus reasonable to pursue *approximation algorithms* — algorithms that compute kinodynamic solutions that are "close" to optimal (in some appropriate sense). It is also desirable to find efficient algorithms, since these problems are currently computationally very intensive. In this vein, it is natural to ask the following three questions about such an algorithm:

- How close to optimal is the solution?
- What is the running time (computational complexity) of the algorithm?
- Can the running time be related to the desired solution accuracy?

Hence, ideally, we would like to find an algorithm for which one could guarantee the solution accuracy to some tolerance ϵ. For a fixed ϵ, one hopes such an algorithm would be efficient (i.e., polynomial-time). While one might expect such an algorithm to run longer when a more accurate solution was required, the algorithm would promise to spend only a polynomial amount of time longer as ϵ is made smaller. Hence, it is natural to approach kinodynamic planning from a combinatorial optimization or algorithmic analysis viewpoint, and we call such an algorithm a ***provably good polynomial-time approximation algorithm.***

Despite a great amount of previous work, until [CDRX] no such algorithm existed. For the many proposed approximate or heuristic techniques, there were no bounds on the goodness of the resulting solutions or on the time-complexity of the algorithms, and all the questions above remained completely open.

As a real robot motion can only be considered safe it it avoids obstacles by some margin, it is natural to incorporate a safety measure into the meaning of "optimal". We do this by including a safety criterion as a problem parameter, along with start and goal states, dynamics bounds, and an obstacle set. An optimal trajectory is thus a minimum-time trajectory that respects the safety criterion. An "approximately optimal" trajectory approximately respects the safety criterion and approximately takes the same amount of time as an optimal trajectory.

[CDRX] considered the restricted situation of particle dynamics, and provided a provably good approximation algorithm for 2- and 3-dimensional optimal kinodynamic planning. The running time of our algorithm is polynomial in the both in the closeness of the approximation $\frac{1}{\epsilon}$ and in the geometric complexity. (This is true even in three dimensions, where computation of an exact solution can be shown to be $\mathcal{NP}$-hard). These bounds on solution accuracy and running time were the first obtained for 2D and 3D optimal kinodynamic planning, which had been an open problem in computational robotics for over ten years. These results generalize to the class of category of robot systems with dynamics bounds.

We have obtained results for robots that obey decoupled dynamics bounds and robots that obey coupled dynamics bounds. For robots in the first category, we reanalyzed of the [CDRX] algorithm. We halved the exponent in the polynomial bounds, and at the same time showed that the algorithm yields a better approximation than previously reported. We showed that if there exists a "safe" optimal-time kinodynamic solution requiring time T, then the algorithm finds a "nearly-as-safe" solution that requires time T, whereas the results in [CDRX] only show that this motion requires at most $(1+\epsilon)T$. While this analysis was only outlined in [DX1], we give complete proofs in [DX3]. These new results indicate that an implementation of the theoretical algorithm is reasonable, and we reported in [DX1] on a preliminary implementation of the extended algorithm and experiments.

For robots with coupled dynamics bounds we first proved a generalized trajectory-tracking lemma. We applied the Coupled Tracking Lemma to extend the general [CDRX] algorithm. We obtained provably good polynomial-time approximation schemes for a class of robot systems with coupled dynamics that includes Cartesian robots obeying L_2 dynamics bounds and open-chain manipulators with rotational and prismatic joints. For these cases, we showed that if there exists a "safe" optimal-time kinodynamic solution requiring time T, then the motion the algorithm finds will be "also-safe" and "ϵ"-close and take at most time $(1+\epsilon)T$. Our treatment of state-dependent dynamics shares its theoretical basis with [JHCP].

4.3 Kinodynamic Motion Planning

Kinodynamic planning attempts to solve a motion problem subject to simultaneous kinematic and dynamic constraints. A robot with d degrees of freedom must move from a start state $\phi_s = (\mathbf{s}, \dot{\mathbf{s}})$ to a goal state $\phi_g = (\mathbf{g}, \dot{\mathbf{g}})$. In the course of the motion, it must avoid a set of obstacles and obey joint limits: these are the *kinematic* constraints. For Cartesian robot, we use the natural coordinate system in $\Re^d$; for a open-kinematic-chain robot we use joint coordinates.

The robot motion is controlled by applying generalized forces according to classical mechanics. Forces are applied at the center of mass of a Cartesian robot; torques and forces are applied at the joints of the open-chain manipulator. Generalized velocities and applied generalized forces must obey certain bounds. In particular, an open-chain manipulator obeys dynamics bounds $(\mathbf{v}_{max}, \mathbf{f}_{max})$ if for all times t the joint velocities $\dot{\mathbf{p}}(t)$ and the applied generalized forces $\mathbf{f}(t)$ obey the the following *dynamics* constraints at each joint i:

$$\begin{aligned} |\dot{p}_i(t)| &\leq v_{max\ i}, \\ |f_i(t)| &\leq f_{max\ i}. \end{aligned} \tag{1}$$

A Cartesian robot obeys L_p dynamics bounds (v_{max}, a_{max}) if for all times t, the acceleration $\mathbf{a}(t)$ and the velocity $\dot{\mathbf{p}}$ obey the following constraints:

$$\begin{aligned} |\dot{\mathbf{p}}(t)|_p &\leq v_{max}, \\ |\mathbf{a}(t)|_p &\leq a_{max}. \end{aligned} \tag{2}$$

We will denote our configuration space by C, and its phase space (the robot state space) by TC. A point in TC is a (position, velocity) pair such as ϕ_s or ϕ_g.

The *commanded acceleration* is a map $\mathbf{a} : [0,T] \rightarrow \Re^d$ for a closed interval $[0,T]$. The velocity $\dot{\mathbf{p}}$ corresponding to $\mathbf{a}$ is its time-integral with respect to the initial state ϕ_s, and the *path* $\dot{\mathbf{p}}$ is simply the time-integral of $\dot{\mathbf{p}}$ w.r.t. ϕ_s. The trajectory Γ for $\mathbf{a}$ and ϕ_s is the mapping $\Gamma : [0,T] \rightarrow TC$ that takes time t to $(\mathbf{p}(t), \dot{\mathbf{p}}(t))$.

We assume that obstacles are polyhedral and input as an arrangement $\mathcal{E}$ with n vertices. We map these obstacles to c-space obstacles in the configuration space [LoP] [C]. *Free space* is the complement of the c-space obstacles in C. Note that the free space of a *rigid, non-rotating* polyhedral robot in a d-dimensional world with polyhedral obstacles is the free space of some point robot in a world d-dimensional with polyhedral obstacles, so that kinodynamic planning for a polyhedral Cartesian robot reduces to kinodynamic planning for a point robot.

For an open-chain manipulator, the problem parameters must include an encoding $\mathcal{M}$ of the physical quantities that determine the robot's dynamics equations, which relate state and applied forces to accelerations. For a given link, we include the mass, the centroid, and the distance and relative orientation between joints. A *general kinodynamic planning problem*, then, is a tuple $(\mathcal{E}, \phi_s, \phi_g, \mathbf{f}_{max}, v_{max}, \mathcal{M})$. We assume that the set of free configurations is bounded by a d-cube of side length l combined with joint limits.

A *solution* to the kinodynamic planning problem is a suitable encoding of the acceleration map $\mathbf{a}$ such that $\Gamma(0) = \phi_s$, $\Gamma(T) = \phi_g$, and Γ obeys the kinematic and dynamic constraints. That is, $\mathbf{p}$ avoids all obstacles, and $\dot{\mathbf{p}}$ and $\mathbf{a}$ respect the appropriate (2) or (1). The *time* for a solution $\mathbf{a}$ is simply T. The *optimal kinodynamic planning problem* is to find a kinodynamic solution requiring minimal execution time.

The theoretically optimal solution may require unrealizable precision in control or sensing. We would like to take such constraints into account in our analysis, in order to supply a result that is not only theoretically interesting, but also perhaps of practical value. Thus, we define the notion of a δ_v-*safe kinodynamic solution*. The intuition behind such solutions is that they avoid obstacles by a safety margin δ_v. We define this safety margin to be an affine function of the trajectory speed; the precise modeling of control and sensing error is not the focus of our present research. We choose the safety margin *a priori* using two positive scalars c_1 and c_0. One may think of this choice as corresponding to how accurately the dynamical system can control its energy consumption. This variation specifies a tube about the safe path that must remain obstacle-free. We call this tube a $\delta_v(c_1, c_0)$-*tube*; its diameter grows linearly with speed.

Hence, an optimal solution should observe a safety margin; in practice, this margin should be speed-dependent. Note that at the same time, the safety margin ensures the existence of a "tube" or family of "nearby" solutions in time, and in phase-space. Hence, these nearby solutions "approximate"

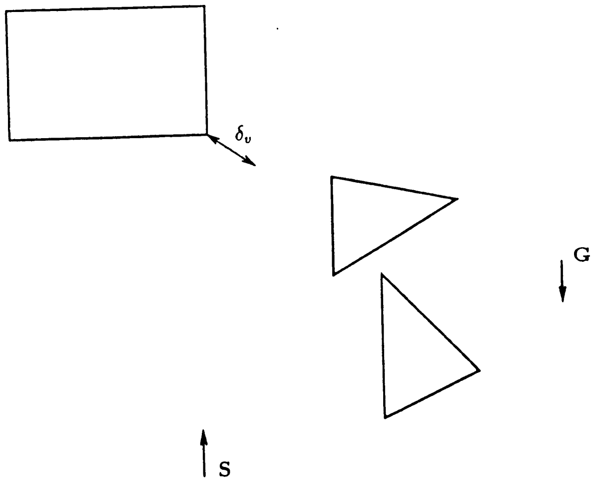

Figure 6: A kinodynamic planning problem, showing the obstacles, the start s, the goal g.

the optimal safe solution both in time and in phase space. The existence of such a "tube" of approximating solutions is essential for our approach; hence safety is not only practically motivated, but also mathematically necessary.

Formally, a *δ_v-safe kinodynamic solution* has the property that for all times t in $[0, T]$, there exists a ball about $\mathbf{p}(t)$ in free space of radius $\delta_v(t) = c_1|\dot{\mathbf{p}}(t)| + c_0$. (Here "ball" is used in a topological sense: i.e., its shape depends on the metric being used.)

Now, for fixed c_1, c_0, consider the class of all δ_v-safe kinodynamic solutions. We define an *optimal δ_v-safe kinodynamic solution* to be a δ_v-safe solution whose time is minimal in the class of δ_v-safe solutions. We will henceforth abbreviate this to "*optimal safe kinodynamic solution*" since δ_v-safety is the only type we consider here.

We must specify what it means for one kinodynamic solution to "approximate" another. First, we say that a trajectory $\Gamma_q : [0, T_q] \to TC$ is *ϵ-close* at the start ϕ_s (resp. goal ϕ_g) if, at time 0 (resp. time T_q), it achieves the start (resp. goal) position to within an additive error that is polynomial in ϵ, and the start (resp. goal) velocity to within a multiplicative error of ϵ, plus or minus another additive error, also polynomial in ϵ. That is, if $\Gamma_q = (\mathbf{p}_q, \dot{\mathbf{p}}_q)$, then both $|\mathbf{p}_q(0) - \mathbf{s}|_\infty$ and $|(1+\epsilon)\dot{\mathbf{p}}_q(0) - \dot{s}|_\infty$ are $\epsilon^{O(1)}$ (and similar conditions hold at the goal). By making ϵ smaller, Γ_q can be made to achieve the start and goal as accurately as desired.

Finally, we define an *approximate optimal safe solution* Γ_q to be a kinodynamic solution which is "ϵ-close" at the start and the goal, "near-optimal" in time, and "also safe". By "near-optimal", we mean that if the optimal safe solution Γ takes time T, the time T_q required by Γ_q is bounded from above by $(1+\epsilon)T$. By "also safe", we mean nearly δ_v-safe: specifically, that the position component $\mathbf{p}_q$ of Γ_q lies in an obstacle-free safety tube $\delta_v(c_1', c_0')$ for constants c_1', c_0', where $c_0' = (1-\epsilon)c_0$ and $c_1' = (1-\epsilon)c_1$. Note that δ_v-safety is actually a *mixed* kinematic and dynamic constraint; it is an example of a kinodynamic constraint that is neither purely kinematic nor purely dynamic.

4.4 Statement of Results for a Cartesian Robot Obeying L_∞ Dynamics Bounds

We assume the workspace has diameter l. We have developed provably good approximation algorithms for the optimal safe kinodynamic planning problem $\mathcal{K}_{L\infty} = (\mathcal{E}, \phi_s, \phi_g, a_{max}, v_{max}, c_1, c_0, \epsilon)$. The algorithms produce an approximate optimal "also-safe" safe solution. c_1, c_0 specify the class of trajectories to be considered "safe." ϵ specifies how close in safety the desired solution should be to

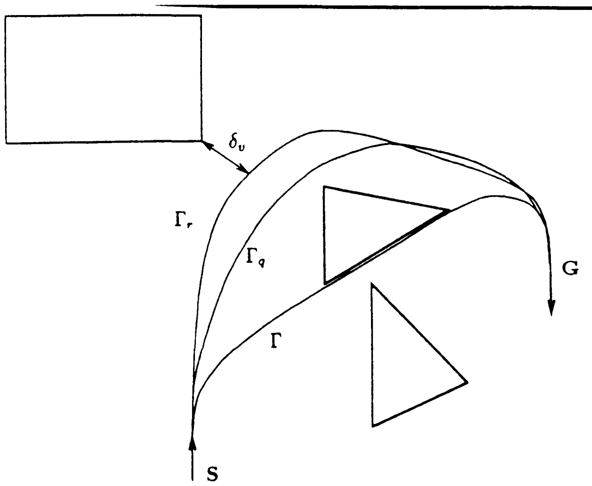

Figure 7: Γ is the optimal kinodynamic solution. However, it is not δ_v-safe. Γ_r is the optimal δ_v-safe solution. Γ_q illustrates a solution that is almost as fast as Γ_r, and almost as safe. Our algorithm can find solutions such as Γ_q, and provides guarantees as to how fast, and how safe they are, relative to Γ_r.

the optimal, safe solution. Concisely stated, in [CDRX], we showed the following. Suppose a point mass in $\Re^d$ $(d = 2, 3)$ must be moved from a start position and velocity $\mathbf{S} = (\mathbf{s}, \dot{\mathbf{s}})$ to a goal position and velocity $\mathbf{G} = (\mathbf{g}, \dot{\mathbf{g}})$. Then:

Theorem 4.1 *Let $0 < \epsilon < 1$. Suppose there is a $\delta_v(c_0, c_1)$-safe trajectory from* $\mathbf{S}$ *to* $\mathbf{G}$ *taking time T_{opt}. Then the algorithm finds a $\delta_v((1-\epsilon)c_0, (1-\epsilon)c_1)$-safe trajectory taking time at most $(1+\epsilon)T_{opt}$ from some $\mathbf{S}^* = (\mathbf{s}^*, \dot{\mathbf{s}}^*)$ to some $\mathbf{G}^* = (\mathbf{g}^*, \dot{\mathbf{g}}^*)$ such that $\mathbf{S}^*$ and $(\mathbf{s}, \frac{\dot{\mathbf{s}}}{1+\epsilon})$ are ϵ-close and $\mathbf{G}^*$ and $(\mathbf{g}, \frac{\dot{\mathbf{g}}}{1+\epsilon})$ are ϵ-close.*

The running time of the algorithm is $O\left(n^2(\log n)\left[\frac{a_{max}v_{max}lc_1^3}{c_0^3\epsilon^6}\right]^d\right)$, $d = 2, 3$, where n is the geometric complexity of the problem.

Our algorithm runs in time polynomial in the geometric complexity n, and in the resolution $(\frac{1}{\epsilon})$. Thus we can bound both the goodness of our approximation, and the running time of the algorithm. Furthermore, we can relate the running time to the error term, and show that this relationship is polynomial.

In [DX1], we halved the exponent of $(\frac{1}{\epsilon})$ in the theorem, and improved the solution accuracy bound from $(1+\epsilon)T_{opt}$ to T_{opt} (thus, the solution achieves the optimal, safe time, but is slightly less safe).

4.5 Results In Cases of Coupled Dynamics Bounds

A robot system obeys *non-independent dynamics bounds* if for all coordinate systems there is an axial direction in which a velocity (acceleration) bound depends on the velocity (acceleration) along another axis. L_2 velocity and acceleration bounds are examples of non-independent dynamics bounds. A robot system is said to obey *coupled dynamics* if there is no global coordinate system in which the dynamics equation and bounds for each coordinate are independent of the state and acceleration of the other coordinates. We say that a robot obeys *coupled dynamics bounds* if it obeys coupled dynamics or if its dynamics bounds are non-independent.

In [DX2,DX3], we described provably good approximation algorithms for the optimal safe kinodynamic planning problems for Cartesian Robots with L_2 dynamics bounds $(\mathcal{K}_{L2} =$

$(\mathcal{E}, \phi_s, \phi_g, a_{max}, v_{max}, c_1, c_0, \epsilon))$ and for open-chain manipulators with rotational and prismatic joints $(\mathcal{K}_{oc} = (\mathcal{E}, \phi_s, \phi_g, \mathbf{f}_{max}, \mathbf{v}_{max}, \mathcal{M}, c_1, c_0, \epsilon))$. The algorithms produce an approximate optimal safe solution. ϵ, which is an input parameter, specifies how close in time the desired solution should be to the optimal, safe solution. c_1, c_0 specify the class of trajectories to be considered "safe." Thus the optimal safe solution Γ takes time T_{opt}, then the algorithm finds an "ϵ-close", "also-safe" solution Γ_q such that $T_q \leq (1+\epsilon)T_{opt}$.

These algorithms run in time polynomial in the geometric complexity n and in the resolution $(\frac{1}{\epsilon})$. The coefficient of the polynomial for an open chain robot depends on certain constants determined from its physical properties. Our algorithms are ϵ-approximation schemes *fully polynomial* in the combinatorial and algebraic complexity of the geometry, and *pseudo-polynomial* in the kinodynamic bounds. Roughly speaking, our algorithm is $O((\frac{1}{\epsilon})^{6d-1})$.

For more details on our kinodynamic results, see [DX1,DX2,DX3].

4.6 Proposed Research

We propose to continue to apply our new kinodynamic techniques and algorithms to robotics, and to implement and test the extensions. We also propose to use the techniques we developed to investigate new algorithms for kinodynamic planning and related problems. In this report, we described our research and our technical approach, and give details on our progress to date. In section 4.6.1 we give several directions for future work that we intend to explore. In particular, we wish to explore further the mathematical and algorithmic implications of our results, implement our algorithms and perform experiments, improve our approximation and complexity bounds, and investigate related open problems. An important related problem is: how to generate provably-good kinodynamic solutions with bounded resources (eg, in a fixed amount of time, what is the best solution one can obtain?) We hope our techniques will extend to this problem. Finally, we hope to find algorithmic solutions to the problems of integrating kinodynamic planning techniques into a task-level robotic assembly system.

Uncertainty in sensing, control, and modeling is perhaps the most fundamental problem in robotics today. To develop and explore new approaches to planning and sensing under uncertainty, we are building new experimental robotic devices. With graduate students Michael O'Donnell, Jim Jennings, and Bob Blondell, we are designing, constructing, controlling, and programming a mobile autonomous robotic platform to investigate planning, sensing, and execution in unstructured environments populated by humans and other robots. We view our mobile robot as an experimental apparatus, or testbed for trying out kinodynamic planning experiments. We would like to generalize our kinodynamic planning algorithms to "non-holonomic" robots such as our vehicle, implement the planning algorithms, and run experiments to have the robot execute the trajectories. We hope that this research could lead to robots that can perform operations at high speeds.

4.6.1 Future Work

In [CDRX] and related work, we described the first polynomial-time, provably good approximation algorithm for kinodynamic planning. We feel that kinodynamic planning represents a new direction in algorithmic motion planning, and expect to see much progress in this area.

There are many directions for future research:

1. The complexity of our algorithm can probably be improved. For work in this direction, see [DX1,DX2,DX3].

2. Other search algorithms, such as A*, may be employed in place of a breadth-first search.

3. Precise lower bounds for kinodynamic planning should be established (especially in the 2D case).

4. Exact algorithms should be explored.
5. We conjecture that if contact is allowed (rather than δ_v-safety) then the complexity of the problem increases considerably. More specifically, one can imagine three related kinodynamic planning problems:
   (a) The first is explored in this report, where the robot must avoid obstacles by a speed-dependent safety margin.
   (b) A second problem might be likened to figure skating: forbidden regions are marked out in the plane (the "ice"), and a path with velocity-dependent non-holonomic constraints must be synthesized. The "obstacles" may be grazed but not crossed. However, the forbidden regions exert no reaction forces on the robot, even when in contact. This second problem corresponds to theoretical "true" optimality.
   (c) One can also imagine a third problem in which the reaction forces (impact, constraint forces, and friction) of the obstacle surfaces are taken into account.

   Finally, one may consider the optimization version of each of these problems. Note that while the theoretical formulation of the "figure skating" problem is quite clean, it may be rather far from practical interest.
6. It would be interesting to extend our approach to 2-norm velocity and acceleration bounds. For work in this direction, see [DX2, DX3, RT].
7. It would be of value to extend our approach to to manipulator systems with full rotational dynamics. For example, one might consider the rigid body dynamics of open kinematic chains with revolute and prismatic joints. Finding near-optimal kinodynamic solutions in these cases would be of great interest. For work in this direction, see [JHCP, DX2, DX3].

In addition, there is a great deal of interesting heuristic and experimental work to be done, in reducing these algorithms to practice. For work on implementation of our approach, and experiments, see [DX1]. Computational kinodynamics seems a particularly fruitful area in which to pursue fast, provably good approximation algorithms, since while the problems are of considerable intrinsic interest, exact solutions may well be intractable. Finally, since the problem has an optimization flavor, the algorithms and proof techniques draw on several branches of computer science and robotics.

Please Note: this section describes only our proposed research in kinodynamic planning. At the end of each section is a subsection proposing future work in that area. These sections are, respectively, 3.10, 4.6, and 5.9. It is in these sections that we outline the next steps in our research agenda.

5 Towards An Algorithmic Theory of Design for Assembly

Working with post-doctoral associate Dinesh Pai, we have been taking steps towards an algorithmic approach to design for assembly. The technical content of this report mirrors a paper we are preparing for the IEEE ICRA [PD89,DP89].

5.1 A Brief Statement of Our Technical Approach

We consider the problem of planning and predicting the motion of a flexible object amidst obstacles in the plane. (Thus, one views the mating part as an "obstacle.") We model the flexible object as a rigid "root" body, attached to compliant members by torsional springs. The root's position may be controlled, but the compliant members move in response to forces from contact with the environment. Such a model encompasses several important and complicated mechanisms in mechanical design and automated assembly: snap-fasteners, latches, ratchet and pawl mechanisms, and escapements. The problem is to predict the motion of such a flexible mechanism amidst fixed obstacles. For example, our algorithm could be used to determine whether a snap-fastener design can be assembled with a certain plan.

We will analyze the physics of these flexible devices, and develop the first combinatorially precise algorithm for predicting their movement under a motion plan. Our algorithm determines when and where the motion will terminate, and also computes the time-history of contacts. In addition to providing the first known exact algorithm that addresses flexibility in motion planning, we also note that our approach to compliance permits an exact algorithm for predicting motions under rotational compliance, which was not possible in earlier work.

5.2 Introduction

Previous work on algorithmic motion planning has largely concentrated on the movement problem for rigid bodies [Lozano83,Brady82,Yap87]. Recently, work in compliant motion planning under uncertainty [LoMaTa84,Erdman84,Donald87b,88a,89] has focused on the problem of moving rigid objects (e.g. pegs) in contact (i.e., compliantly) with obstacles (such as holes) under force-control, subject to bounded uncertainty and error. However, real objects that robots might assemble are typically not rigid. For example, a Sony Walkman is made of plastic parts that snap together. Significant advances were made in the design of the IBM ProPrinter, by replacing traditional fasteners such as screws with plastic parts that simply snap together. The reason these plastic parts snap together is that they are *flexible*: more precisely, they are *passively compliant*. This means that when the parts are brought together and an external force applied, the parts deform in a prescribed way. More interestingly, the force required to mate two parts may be much less than the force required to take them apart.

Since we wish to be able to design and have our robots assemble such objects given task-level descriptions, we must have a systematic program for reasoning about and predicting their motions in contact. To this end, we will first make precise a sufficiently powerful notion of flexibility to model the objects above. Our model is based on the well known *generalized spring* in the robotics and controls literature [Mason81]. We will model the physics of interaction of the flexible parts and the obstacles using quasi-static analysis and Coulomb friction. Then, using these tools, we will proceed to develop a combinatorially precise algorithm for predicting the motion of a flexible object amidst obstacles. The algorithm is roughly $O(N^2 \log N)$ in the geometric complexity, N.

From a complexity-theoretic standpoint, our result may also be viewed as follows. Earlier work on compliant motion planning [LoMaTa84,Erdman84,Donald88a] has employed a dynamic model called the *generalized damper* [Whitne76]. While this model has led to exact algorithms in the case of pure translations [Donald88a,Canny89a], so far we have not been able to provide combinatorially precise algorithms once rotations are permitted. This is true even if the commanded motion is a

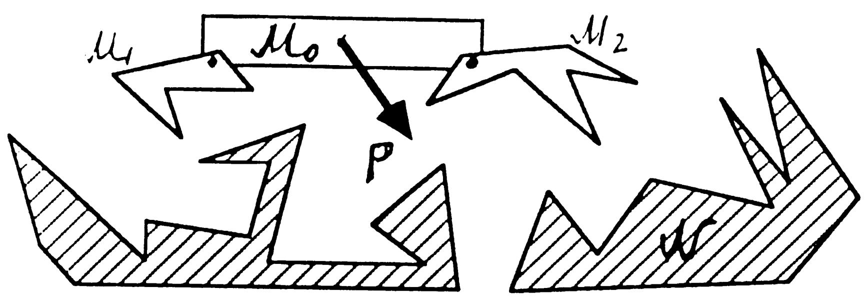

Figure 8: Linked body $\mathcal{M}$ moving among $\mathcal{N}$.

pure translation and the moving object (eg., the peg) is merely rotationally compliant. One reason for this difficulty seems to be that under the damper model, the resulting motions are not obviously algebraic; even more disturbing, the outer envelope (or *forward projection*) of these motions cannot yet be algebraically described. (This is also a problem in many more complicated dynamical systems). However, under the model in this report, rotationally compliant subparts are permitted, and yet all resultant trajectories of the system are algebraic. Finally, we believe that our algorithm is sufficiently simple that it could be implemented, and used to plan and verify the design of flexible mechanisms for ease of assembly.

5.3 Problem Statement

We consider the problem of moving a flexible, linked body $\mathcal{M}$ in the plane, in a polygonal environment $\mathcal{N}$. The flexible body $\mathcal{M}$ is constructed as follows: polygons $\mathcal{M}_h$, $h = 1, \ldots, l$, called "pawls", are attached to a root polygon $\mathcal{M}_0$, at hinge points P_h. Each hinge is coupled with a spring of stiffness k_h. (See Figure 8). The motion of the body $\mathcal{M}$ consists of rigid translation of the root polygon $\mathcal{M}_0$. The pawls $\mathcal{M}_h$ are free to move compliantly as dictated by interactions with the environment and the spring.

The *Motion Prediction Problem* is to determine:

1. The time of termination of the motion, the cause of termination (such as sticking due to friction, sticking due to kinematic constraints, etc.), and the configuration of $\mathcal{M}$ at termination.

2. The time history of contacts between $\mathcal{M}$ and $\mathcal{N}$.

We make the following assumptions about the physics of object interactions and the motion of $\mathcal{M}$:

- Object interactions are restricted to those between $\mathcal{M}_h$ and $\mathcal{N}$. In other words, pawls do not collide with each other, but may collide with the environment $\mathcal{N}$. The effect of this assumption is to make the motion of each pawl independent of the motion of other pawls. Henceforth we shall consider the motion of a single pawl $\mathcal{M}_h$.

- Since the root $\mathcal{M}_0$ is undergoing a rigid translation, so does the hinge point P_h. We shall assume that this translation is a straight line motion given by $\boldsymbol{p} = \boldsymbol{p}_o + \dot{\boldsymbol{p}}t$ where t is the time, $\boldsymbol{p}_o$ is the initial position (at $t = 0$) of P_h, and $\dot{\boldsymbol{p}}$ is its velocity.

- Quasi-static motion: The motion is assumed to be slow enough that inertial effects are not significant. This corresponds to assuming that there is no acceleration of the pawl, and hence the forces on the pawl are balanced. The quasi-static assumption is reasonable at small speeds and is widely used (see, for example, [Whitne82,Mason82a,Donald89,Pai88]).

- If the pawl slides off $\mathcal{N}$ into free space, it may have a residual torque due to the spring being "cocked". We shall assume that the pawl rotates back towards its rest orientation at such great speed that $\boldsymbol{p}$ does not change significantly during the rotation and can be taken to be constant. This, incidentally, is the "snap" in the snap-fasteners that we wish to model. This assumption can be relaxed by assuming a linear relationship between the translation $\boldsymbol{p}$ and the rotation (e.g., see [Canny86]), but we do not deal with it here.

- The forces of friction arising from contact obey Coulomb's Law. We further assume that there is a single coefficient of friction. This assumption is also widely used.

Two types of contact are possible between the pawl $\mathcal{M}_h$ and a polygon in $\mathcal{N}$. Following the convention of Lozano-Pérez [Lozano83] and Donald [Donald 87a], we say that Type-A contact occurs when a vertex of $\mathcal{N}$ touches an edge of the pawl; Type-B contact occurs when a vertex of the pawl touches an edge of $\mathcal{N}$.

We can now write the contact constraint equations for the two types of contact, as in [Canny86]. We shall index features (vertices and edges) of the moving pawl by the subscript i and features of the polygonal environment by the subscript j. Let an edge of $\mathcal{M}_h$ be represented by its outward normal, $\boldsymbol{n}_i$, and its distance to the hinge point along the normal, d_i. Let $\boldsymbol{p}_j$ be a vector to the contact vertex of $\mathcal{N}$. Let R_θ be the linear transformation which rotates a vector by an angle θ. Then the type-A constraint can be written as

$$(\boldsymbol{p}_j - \boldsymbol{p})\cdot R_\theta \boldsymbol{n}_i - d_i = 0. \tag{3}$$

Similarly, the type-B constraint can be written as

$$(R_\theta \boldsymbol{p}_i + \boldsymbol{p})\cdot \boldsymbol{n}_j - d_j = 0. \tag{4}$$

Where $\boldsymbol{p}_i$ is the vector from the hinge point P_h to the contact vertex of $\mathcal{M}_h$.

5.4 Sticking due to Friction

A careful analysis of the physics of compliant sliding is necessary to formulate a precise, combinatorial version of this apparently continuous problem. When the pawl is in contact with the environment, it is possible for the motion to stop because the contact forces are adequate to balance the applied forces. This is called "sticking due to friction," to distinguish it from other kinds of sticking (see Section 5.7). The algorithm must determine if this can happen during motion. The works of Mason [Mason82a] and Erdmann [Erdman84] address this issue in considerable detail. In this section, we show that for our problem this determination can be made by a simple geometric test.

Coulomb's Law of friction states that the tangential force due to friction, $\boldsymbol{f}_t$, on a point which is sliding on an edge in a specified direction is given by $\boldsymbol{f}_t = \mu \boldsymbol{f}_n$, where $\boldsymbol{f}_n$ is the force normal to the edge and μ is the coefficient of friction. In addition, the direction of the friction force is opposite that of the motion.

Coulomb's Law has a geometric interpretation which is frequently more useful (see, for example, [Mason82a]). We provide here a slightly different interpretation, which is more applicable to our problem. Suppose point $\boldsymbol{p}_i$ is in contact with edge e_j. The outward pointing unit normal vector to the edge is denoted $\boldsymbol{n}_j$. We are also given the direction of sliding by specifying $\boldsymbol{v}$, the unit vector tangent to the edge in the direction of sliding (see Figure 9). Coulomb's Law defines a half-cone of forces acting on the point $\boldsymbol{p}_i$. The cone is the convex combination [4] of the vectors $\boldsymbol{n}_j$ and the vector $\boldsymbol{f}_e = \boldsymbol{n}_j - \mu\boldsymbol{v}$.

[4] The convex combination of two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ is the set of all vectors of the form $\zeta\boldsymbol{x} + \eta\boldsymbol{y}$, where $\zeta, \eta \geq 0$.

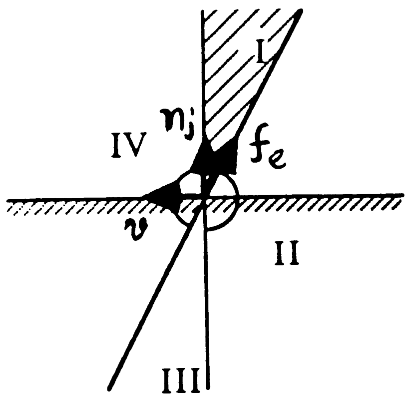

Figure 9: Geometric Interpretation of Coulomb's Law

Any force in the interior of this cone and along $\boldsymbol{n}_j$ can be resisted and the point will stick. A force along $\boldsymbol{f}_e$ will result in $\boldsymbol{p}_i$ sliding in equilibrium, in the direction of $\boldsymbol{v}$. A force outside the cone results in the point sliding along $\boldsymbol{v}$, out of equilibrium, which is precluded by our assumption of quasi-static motion.

We first consider the type-B contact: the point is the vertex $\boldsymbol{i}$ of the pawl, and the edge is the edge e_j of the environment. Let $\boldsymbol{r} = R_\theta \boldsymbol{p}_i$. The condition that contact is maintained between the point and the edge restricts the possible values of the force $\boldsymbol{f}$ and the torque $\boldsymbol{r} \times \boldsymbol{f}$ [5] due to the contact force. First, $\boldsymbol{f} \cdot \boldsymbol{n}_j$ must be positive; second, if the hinge is in the right half-plane, we require $\boldsymbol{r} \times \boldsymbol{f} < 0$, and if it is in the left half-plane, we require that $\boldsymbol{r} \times \boldsymbol{f} > 0$. For quasi-static sliding, the total force is given by $\boldsymbol{f} = \alpha \boldsymbol{f}_e$, $\alpha > 0$. Therefore $\boldsymbol{f}$ trivially satisfies the condition $\boldsymbol{f} \cdot \boldsymbol{n}_j > 0$. The condition on the torque $\boldsymbol{r} \times \boldsymbol{f}$, however, depends on the location of the hinge point. Let the possible locations of the hinge point with respect to the contact point, $-\boldsymbol{r}$, be divided into four sectors as follows: if $-\boldsymbol{r} = \beta \boldsymbol{n}_j + \gamma \boldsymbol{f}_e$, the hinge point is in Sector

I if $\beta > 0, \gamma > 0$,
II if $\beta < 0, \gamma > 0$,
III if $\beta < 0, \gamma < 0$,
IV if $\beta > 0, \gamma < 0$.

Figure 9 also indicates the sectors.

Proposition 5.1 *For type-B contact, the contact point will slide when the hinge is in sectors II and IV. The contact point will not slide (will stick due to friction) when the hinge is in sectors I and III.*

Proof. Suppose the pawl is sliding quasi-statically. Then,

$$\begin{aligned}
\boldsymbol{f} &= \alpha \boldsymbol{f}_e \\
&= \alpha(\boldsymbol{n}_j - \mu \boldsymbol{v}). \\
\boldsymbol{r} &= -\left[\beta \boldsymbol{n}_j + \gamma \boldsymbol{f}_e\right] \\
&= -\left[(\beta + \gamma)\boldsymbol{n}_j - \gamma\mu \boldsymbol{v}\right]. \\
\boldsymbol{r} \times \boldsymbol{f} &= -\alpha\left[0 - (\beta + \gamma)\mu + \gamma\mu + 0\right] \\
&= \alpha\beta\mu. \qquad (5)
\end{aligned}$$

[5]Here $\boldsymbol{r} \times \boldsymbol{f}$ is the component of the usual cross product along the axis orthogonal to the plane, i.e., in coordinates, $\boldsymbol{r} \times \boldsymbol{f} = r_x f_y - r_y f_x$.

Since $\alpha, \mu > 0$, the sign of $\boldsymbol{r} \times \boldsymbol{f}$ is just that of β. Now, in sectors I and II, the right-half plane, we require that $\boldsymbol{r} \times \boldsymbol{f} < 0$ to maintain contact. But from Equation 5, $\boldsymbol{r} \times \boldsymbol{f} > 0$ in sector I and $\boldsymbol{r} \times \boldsymbol{f} < 0$ in sector II. Hence the contact point will stick in sector I and slide in sector II. A similar argument shows that the contact point will stick in III and slide in IV. □

A similar argument applies to type-A contact if the sectors and related terms are defined appropriately . Let $\boldsymbol{f}$ be, as before, the force on the point of contact, which is now a vertex j of the environment, and let $\boldsymbol{r}$ be the vector from the hinge point to the contact point, which is now given by $\boldsymbol{r} = \boldsymbol{p}_j - \boldsymbol{p}$. Let the outer normal of the edge e_i be denoted by $\boldsymbol{m}_i = R_\theta \boldsymbol{n}_i$, $\boldsymbol{v}$ be the unit vector tangent to the edge in the direction in which the vertex slides with respect to the pawl's edge and $\boldsymbol{f}_e = \boldsymbol{m}_i - \mu \boldsymbol{v}$. To maintain contact, $\boldsymbol{f} \cdot \boldsymbol{m}_i$ must be positive and, if the hinge is in the right half-plane, we require $\boldsymbol{r} . \times -\boldsymbol{f} > 0$, i.e.$\boldsymbol{r} \times \boldsymbol{f} < 0$, and if it is in the left half-plane, we require that $\boldsymbol{r} \times \boldsymbol{f} > 0$.

With the terms defined as above, we can divide the possible locations of the hinge point with respect to the contact geometry into four sectors exactly as we did for Type-B contact, despite the fact that the interpretation of the sectors is different. Now it is easy to prove the corresponding proposition:

Proposition 5.2 *For type-A contact, the contact point will slide when the hinge is in sectors II and IV. The contact point will not slide (will stick due to friction) when the hinge is in sectors I and III.*

Proof. The proof is textually identical to the proof for Type-B contact, with $\boldsymbol{n}_j$ replaced by $\boldsymbol{m}_i$. □

5.5 The Sliding Direction

The analysis of Section 5.4 utilized the direction of motion to determine whether the pawl could stick due to friction. It turns out that the direction of the motion can be determined from the differential kinematics of the contact, given the motion of the hinge, $\boldsymbol{p}$.

Consider the motion for a type-B contact. For convenience, assume that the Y-axis of the reference coordinate frame is along the outward normal of the edge, i.e., $\boldsymbol{y} = \boldsymbol{n}_j$, and the X-axis is along the edge, oriented to form a right-handed coordinate system. Recall that $\boldsymbol{r} = R_\theta \boldsymbol{p}_i$. Then, differentiating the type-B kinematic constraint equation, Equation 4, we get

$$(\dot{\boldsymbol{r}} + \dot{\boldsymbol{p}}) \cdot \boldsymbol{n}_j = 0. \tag{6}$$

Now $\dot{\boldsymbol{r}} = \omega(-r_y, r_x)^T$, where ω is the angular velocity of the pawl. Hence, the above equation can be written as

$$\omega r_x + \dot{p_y} = 0. \tag{7}$$

This equation can be derived in another way, which will be useful in Section 5.7. Define the *departure speed*, u, as the component of the velocity of the contact point along the edge normal $\boldsymbol{n}_j$. The velocity of the contact point is given by $\dot{\boldsymbol{r}} + \dot{\boldsymbol{p}}$. Hence

$$u = (\dot{\boldsymbol{r}} + \dot{\boldsymbol{p}}) \cdot \boldsymbol{n}_j = \omega r_x + \dot{p_y}. \tag{8}$$

The differential kinematic constraint for sliding corresponds to the requirement that the departure speed is zero. It is easy to see that the two derivations are equivalent.

The *sliding speed*, s, is defined as the component of the velocity of the contact point in the $\boldsymbol{x}$ direction (the vector $\boldsymbol{v}$ of Section 5.4 is therefore $sgn(s)\boldsymbol{x}$). Hence,

$$\begin{aligned} s &= (\dot{\boldsymbol{r}} + \dot{\boldsymbol{p}}) \cdot \boldsymbol{x} && (9) \\ &= -\omega r_y + \dot{p_x}. && (10) \end{aligned}$$

Combining with Equation 7,

$$s = \dot{p_x} + \frac{\dot{p_y}}{r_x} r_y. \tag{11}$$

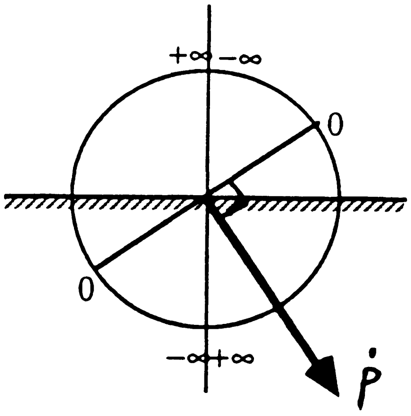

Figure 10: The sliding direction for Type-B contact.

Equation 11 has a simple geometric interpretation, shown in Figure 10. The circle represents the possible positions of the hinge point with respect to the contact point. The numbers on the circle indicate the sliding speed, when $\dot{p_y} < 0$ (if $\dot{p_y} > 0$, the speeds shown have opposite sign; if $\dot{p_y} = 0$, s is constant). Notice that the sliding speed is 0 when $\boldsymbol{r} \cdot \dot{\boldsymbol{p}} = 0$. Also, from Equation 7, $\omega < 0$ when the hinge point ($-\boldsymbol{r}$) is in the right half-plane of the figure, and $\omega > 0$ in the left half-plane.

We can now combine the above results with those of section 5.4. Suppose, without loss of generality, that the hinge point is in the right half-plane. If the hinge is in the region corresponding to $s < 0$, the possible direction of slide is to the left. If, in addition, the hinge is in sector I, it will stick. If it is outside sector I, the location of the hinge will move in the direction of ω, at some point (corresponding to $\boldsymbol{r} \cdot \dot{\boldsymbol{p}} = 0$), the direction of slide will reverse to $s > 0$. This also brings sectors III and IV to the right half-plane. Eventually, the hinge will enter sector III for $s > 0$, and stick. Figure 10 also depicts the evolution of the hinge point.

The time at which the contact will stick in sector III is easily computed by letting $R_\theta \boldsymbol{p}_i = \frac{|\boldsymbol{p}_i|}{1+\mu^2}(\boldsymbol{n}_j + \mu \boldsymbol{v})$ in the constraint equation 4 and solving the resulting linear equation for t.

A similar but slightly more complicated analysis applies to Type-A contact. We shall first derive an expression for the velocity $\boldsymbol{w}$ of the contact point with respect to a frame fixed on the moving pawl, and expressed in the world coordinate frame. We shall assume for simplicity of derivation, that the coordinate frame attached to the pawl has its Y-axis along the edge normal $\boldsymbol{m}_i = R_\theta \boldsymbol{n}_i$, and the X-axis is along the edge so as to form a right-handed coordinate system. As in Section 5.4, for Type-A contact, $\boldsymbol{r} = \boldsymbol{p}_j - \boldsymbol{p}$, and let ${}^P\boldsymbol{r} = R_\theta{}^T \boldsymbol{r}$ be the corresponding vector in the coordinate frame fixed to the pawl. Let Ω be the skew-symmetric angular velocity matrix, given by

$$\Omega = \begin{pmatrix} 0 & -\omega \\ \omega & 0 \end{pmatrix}. \tag{12}$$

where ω is the angular velocity of the pawl. From elementary kinematics (see, for example, [BotRot79]), $\Omega = \dot{R}_\theta R_\theta{}^T$.

The velocity in the frame attached to the pawl, $R_\theta{}^T \boldsymbol{w}$, can now be calculated by differentiation:

$$\begin{aligned} R_\theta{}^T \boldsymbol{w} &= {}^P\dot{\boldsymbol{r}} \\ &= R_\theta{}^T \dot{\boldsymbol{r}} + \dot{R_\theta}{}^T \boldsymbol{r} \\ &= R_\theta{}^T \dot{\boldsymbol{r}} + R_\theta{}^T \Omega^T \boldsymbol{r}. \end{aligned}$$

Therefore, (13)

$$\boldsymbol{w} = \dot{\boldsymbol{r}} + \Omega^T \boldsymbol{r}. \tag{14}$$

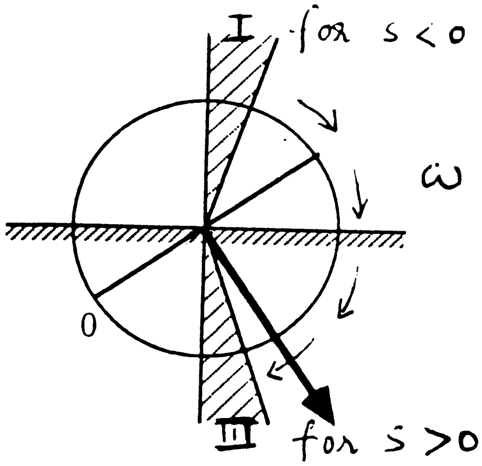

Figure 11: The sliding direction for Type-A contact.

The differential constraint corresponds to the departure speed, u, being zero. Therefore,

$$\begin{aligned} u &= \boldsymbol{w}\cdot\boldsymbol{m}_i \\ &= \dot{r_y} - \omega r_x \\ &= 0. \end{aligned} \tag{15}$$

Solving for ω,

$$\begin{aligned} \omega &= \frac{\dot{r_y}}{r_x} \\ &= \frac{-\dot{p_y}}{r_x}. \end{aligned} \tag{16}$$

Hence, the sliding speed, s, is given by

$$\begin{aligned} s &= \boldsymbol{w}\cdot\boldsymbol{x} \\ &= \dot{r_x} + \omega r_y \\ &= \dot{r_x} + \frac{-\dot{p_y}}{r_x} r_y \\ &= -\dot{p_x} - \dot{p_y}\frac{r_x}{r_y} \end{aligned} \tag{17}$$

Figure 11 depicts the sliding speed for a Type-A contact and the orbit of the hinge point during sliding. The time at which the hinge point will enter Sector III and stick can be computed as follows. At the time of entry into Sector III, the $\boldsymbol{r}$ is given by

$$\begin{aligned} \boldsymbol{r} &= \frac{|\boldsymbol{r}|}{\sqrt{1+\mu^2}}(\boldsymbol{m}_i + \mu \boldsymbol{v}) \\ &= \frac{|\boldsymbol{r}|}{\sqrt{1+\mu^2}}\left(\boldsymbol{m}_i + \mu \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \boldsymbol{m}_i\right) \\ &= \frac{|\boldsymbol{r}|}{\sqrt{1+\mu^2}} \begin{pmatrix} 1 & -\mu \\ \mu & 1 \end{pmatrix} \boldsymbol{m}_i \end{aligned} \tag{18}$$

Hence,

$$\boldsymbol{m}_i = \frac{1}{|\boldsymbol{r}|\sqrt{1+\mu^2}} \begin{pmatrix} 1 & \mu \\ -\mu & 1 \end{pmatrix} \boldsymbol{r} \tag{19}$$

Plugging this back in Equation 3, we get

$$\begin{aligned} \boldsymbol{r}\cdot\boldsymbol{m}_i - d_i &= \frac{1}{|\boldsymbol{r}|\sqrt{1+\mu^2}}\boldsymbol{r}^T \begin{pmatrix} 1 & \mu \\ -\mu & 1 \end{pmatrix} \boldsymbol{r} - d_i \\ &= \frac{|\boldsymbol{r}|}{\sqrt{1+\mu^2}} - d_i = 0. \end{aligned} \quad (20)$$

$$(21)$$

Since $\boldsymbol{r}$ is a linear function of t, this is a quadratic equation in time and can be solved for t.

5.6 Collision Detection

As the linked body $\mathcal{M}$ is moved in the environment $\mathcal{N}$, the pawl $\mathcal{M}_h$ may collide with an obstacle. Three types of motion are possible for the pawl in our problem: pure translation without rotation; pure rotation without translation; and most importantly, the motion of the pawl sliding while maintaining contact with a given feature of the obstacle. The collision detection problem is to find the times at which the pawl collides with $\mathcal{N}$, and report which features are in contact during collision.

In each of the cases, the obstacle constraint equations (3 and 4) and can be reduced to algebraic equations. This is possible since the trigonometric functions in the rotation matrix R_θ can be replaced by new indeterminate quantities (eg. $\sin\theta = s$, $\cos\theta = c$), with the addition of a new constraint ($s^2 + c^2 - 1 = 0$), as in [Donald 87a]. More practically, the substitution $u = \tan\frac{\theta}{2}$ yields constraint equations that are quadratic in u, with coefficients that are affine in x and y. Since x and y are affinely parametric in time t, hence the coefficients are also affine in t. Intersecting two of these constraints requires intersecting two quadratics. Pure rotational intersection detection (the "snap") requries solving a quadratic in u. Pure translational intersections require solving an affine equation. Hence there exists a closed-form, purely algebraic solution to these intersection problems. The collision detection for each obstacle constraint can be done in constant time, since the degree, size, and number of variables in the constraint polynomials is fixed. If m is the geometric complexity of the pawl and n is that of the size of the environment, there are mn such constraints, and the total time complexity for collision detection is $O(mn)$.

5.7 Resolving Multiple Contacts

When the pawl is sliding on the environment, it is possible for a second contact to occur due to collision. Generically, the pawl can maintain two contacts for only an instant in time. The algorithm has to determine if subsequent motion is possible, and if so, which of the two contacts remain. In this section we show that the second contact either causes the pawl to stick due to incompatible constraints, or the second contact replaces the first. We also provide the means for determining which of the two cases occurs.

We shall first consider the case when both contacts are of type-B. As in section 5.5, we define the *departure speed*, u, by

$$u = (\dot{\boldsymbol{r}} + \dot{\boldsymbol{p}})\cdot\boldsymbol{n}_j = \omega r_x + \dot{p}_y. \quad (22)$$

Let us subscript the quantities corresponding to the original contact by 1, and the the quantities for the new contact by 2. Since both the contacts are on with the same pawl, they have the same angular velocity ω. Hence we have:

$$\omega = \frac{u_1 - \dot{p}_{y1}}{r_{x1}} = \frac{u_2 - \dot{p}_{y2}}{r_{x2}} \quad (23)$$

This equation specifies a linear relationship between u_1 and u_2 which must hold for all values of angular velocity. However, not all possible values of the coefficients can occur. If the new contact occured while

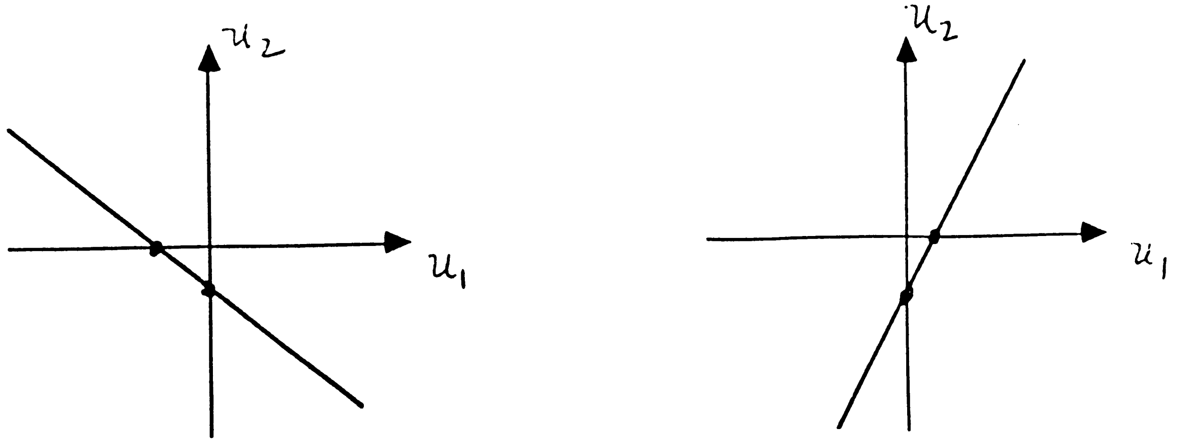

Figure 12: Possible lines in the u_1–u_2 plane.

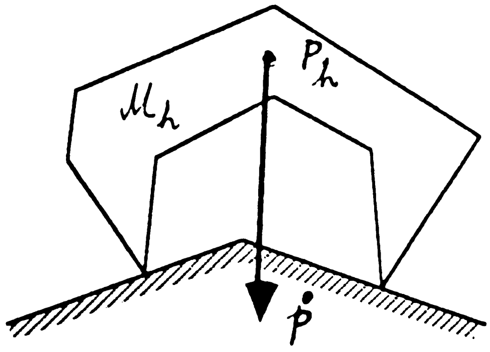

Figure 13: Sticking due to incompatible constraints.

the contact point was sliding on the interior of an edge, $u_1 = 0$ and the departure velocity u_2 must be negative (recall that $\boldsymbol{n}_j$ is the *outward* normal). Hence only a line for which the intercept with the u_2 axis is negative is possible (see Figure 12). Of these, if the u_1-intercept is negative, it is clear that it is not possible to find an angular velocity such that neither of the departure velocities are negative. Hence the pawl must stick due to incompatible constraints. Figure 13 depicts a situation in which such sticking occurs. Similarly, if the u_1-intercept is positive, the only possible way to maintain contact with the environment is to have $u_2 = 0$ (sliding along the second contact), while $u_1 > 0$ (the first contact departing from its edge).

Finally, to ascertain which of the two cases occur, we only need to check the sign of the u_1-intercept of the line. From equation 23, this is given by

$$sgn\left(\dot{p_{y1}} - \dot{p_{y2}}\frac{r_{x1}}{r_{x2}}\right). \tag{24}$$

The other three possible transitions are similar. As before, the fact that departure speeds are linked by being on the same pawl leads to a linear relationship between u_1 and u_2, and as before, the pawl will stick due to incompatible kinematic contraints or the second contact will replace the first. We can check which of the two occurs by checking the sign of the u_1 intercept – if it is negative, the pawl sticks. We list the expressions for the sign of the u_1 intercept for the three possible cases below. When the two contacts are of different type, we shall superscript r by the contact type for clarity, since the interpretation of $\boldsymbol{r}$ depends on the contact type.

If both contacts are Type-A:

$$sgn\left(\dot{r_{y1}} - \dot{r_{y2}}\frac{r_{x1}}{r_{x2}}\right). \tag{25}$$

If the first contact is Type-B and the second contact is Type-A:

$$sgn\left(\dot{p_{y1}} + \dot{r^A_{y2}}\frac{r^B_{x1}}{r^A_{x2}}\right). \tag{26}$$

If the first contact is Type-A and the second contact is Type-B:

$$sgn\left(\dot{r^A_{y1}} + \dot{p_{y2}}\frac{r^A_{x1}}{r^B_{x2}}\right). \tag{27}$$

5.8 The Algorithm

Putting all the pieces developed in the previous sections together, we present the following algorithm for the motion of a single pawl. In order to keep the number of special cases small, we describe the algorithm for type-B contacts only. It is straightforward to handle type-A contacts as well.

Preprocessing: Determine the possible collisions of the pawl with $\mathcal{N}$ while undergoing a pure translation of $\boldsymbol{p}$. Let the first such time be t_b. Also determine the time t_e at which the root polygon $\mathcal{M}_0$ collides with $\mathcal{N}$. The algorithm can now be concerned only with the time interval $[t_b, t_e]$.

Step 1: For the current contact, determine the possible direction of slide, using the results of Section 5.5.

Step 2: Determine if sticking occurs due to friction, using the results of Section 5.4.

If it sticks, report this, and END.

Else, determine the time t_s at which the contact enters the sticking region, using the results of Section 5.5.

Step 3: Determine if other collisions between the pawl and $\mathcal{N}$ occur while the pawl is sliding while maintaining the current contact, using the results of Section 5.6. Let t_c be the time of the first such collision. Note that one also obtains the time at which the contact reaches the end of the current edge e_j, i.e., the time at which a collision occurs between the contact vertex and the adjacent edge. Let this time be t_f.

If $\min\{t_c, t_s\} = t_s$ then report that the pawl has stuck due to friction, END;

Else, if $t_c = t_f$, the contact vertex has reached the end of the current edge. The pawl now undergoes a pure rotation towards its rest position. Determine the next collision during this rotation. Note that the transition to an adjacent edge is included in this test.

- If a new contact is found, report it. GOTO Step 1 and repeat with the new contact.
- Else find the next collision under pure translation after the rotation has finished. Report the collision found. GOTO Step 1 and repeat with the new contact. If no collision is found, END.

Else, if $\min\{t_c, t_s\} = t_c$, report the new contact that has occurred. Analyze the multiple contacts using the results of Section 5.7.

- If the new contact is incompatible, report this, and END; the pawl has stuck due to incompatible kinematic constraints.
- Else, the new contact is effective. GOTO Step 1 and repeat with the new contact.

End

Each collision detection step above takes time $O(mn)$. In addition, suppose the contact changes c times. Then the loop of Steps 1–3 is repeated c times. Therefore the algorithm takes time $O(cmn)$. We expect (but do not prove) that $c = O(mn)$, and that it will typically be much smaller. However, in order to obtain an upper bound, we offer the following modification to the above algorithm, yielding a running time of $O(m^2n^2 \log mn)$.

We handle the collision detection while sliding via the following data structure: with each constraint equation, we associate two balanced search trees, labeled POS and NEG. The leaves of the POS tree contain, sorted by the time t, all the tuples (t, k) where $k = 1, \ldots, mn$, is the index of a contact constraint equation, and t is the time at which the constraint is satisfied by $\boldsymbol{p}(t)$, given that $\boldsymbol{r} \times \boldsymbol{n}_j$ is positive. Similarly, the NEG tree corresponds to $\boldsymbol{r} \times \boldsymbol{n}_j$ being negative. Generically, the number of different times t for a given k has a fixed upper bound which can be obtained from Bezout's Theorem [Waerde50b]. Hence, there are $O(mn)$ nodes in each tree, and hence each tree takes $O(mn \log mn)$ to build, and hence the entire data structure takes $O(m^2n^2 \log mn)$ time to build. Each collision detection in Step 3 of the algorithm now takes $O(\log mn)$.

We claim that the collision detection in Step 3 can be requested at most as many times as there are leaves in the data structure. Suppose we request the next collision at time t, while maintaining constraint k_1, answer is the tuple (t_c, k_2). From the algorithm it follows that only three things can happen:

1. The pawl sticks due to friction, or incompatible constraints, in the time interval $[t, t_c]$, and the program terminates.

2. The pawl slides off the end of the edge $t_c = t_f$. This implies that leaf of the tree with time t_f and constraint corresponding to the contact vertex touching an adjacent edge, are used up, and the time is advanced to t_f.

3. (t_c, k_2) is the new contact. Hence this leaf is used up and the time is advanced to t_c.

Hence, no tuple can be the answer to collision detection request more than once, proving our claim. Since the entire data structure has $O(m^2n^2)$ leaves, the collision detection while sliding takes $O(m^2n^2 \log mn)$ time. We can similarly show that the collision detection problem for pure translation and pure rotation can be done in $O(m^2n^2 \log mn)$.

5.9 Proposed Research

In this report, we addressed the problem of predicting the motion of a flexible object amidst a polygonal environment. To this end, we presented a careful analysis of the physics of the interaction of the flexible body with the environment. The analysis yielded several simple tests to determine the motion of the body, which were integrated into an algorithm for predicting the motion under a given motion plan.

This motion prediction problem is the first step in reasoning about such devices. We propose to continue research on design for assembly along the lines of this report. Specifically, future work will include implementation of the algorithm, generation and testing of motion plans, and the design of flexible mechanisms given the desired performance. We hope to extend our model to include uncertainty in the position and orientation of the moving body, and to include continuously deformable bodies as well. We will also extend our algorithm to compute the force history (the forces on the flexible object as a function of time). In addition to building an experimental system that implements the algorithm sketched above, we will run experiments to analyze designs.

We believe that our algorithmic analysis could be used as a basic step in an approach to "design as search." For example, suppose we are given geometric models of two parts to mate, and an approximate "assembly plan" (direction of mating). We envision an algorithm that searches around the boundaries of the models, and modifies the shape by introducing snap-fasteners and ratchet and pawl mechanisms. After each modification, the analysis algorithm described above could be run, to determine how the parts will mate, and the forces required to mate and disassemble them. Depending on the results of the analysis, the design change could be modified, retracted, or declared suitable. Of course, there is a host of conceptual and algorithmic problems that must be solved to make such an approach viable, but we believe that with the algorithmic underpinnings described in our analysis algorithm, a research program in this area is now viable. The first step is to refine, develop, implement, test, and extend the idea above in the analysis phase; the second, and less clear step, would involve the eventual generalization into the realm of "design as search."

Please Note: this section describes only our proposed research in design for assembly. At the end of each section is a subsection proposing future work in that area. These sections are, respectively, 3.10, 4.6, and 5.9. It is in these sections that we outline the next steps in our research agenda.

6 Review of Previous and Related Work

Broadly speaking, previous work falls in the following categories: algorithmic motion planning, compliant motion planning with uncertainty, model error, error detection and recovery, and kinodynamic motion planning. There has been a great deal of work in these areas; in this bibliographic essay we give a flavor for many of the key ideas.

6.1 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 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.*].

6.2 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.

In [Donald 87b,88a], we introduced a combinatorial object call the *non-directional backprojection*, and analyzed its complexity. Our analysis led to efficient algorithms for certain subproblems in compliant planning with uncertainty. In particular, we gave an efficient algorithm for planning one-step strategies in the plane. By using results from computational algebra, we showed that planning a guaranteed planar multi-step strategy with sticking termination can be decided in time polynomial in the geometric complexity, and roughly singly-exponential in the number of steps in the plan. [Briggs 89] improved our one-step algorithm by a quadratic factor.

6.3 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, except for [Donald 87b, 88a;b, 89, 90]. In this work, we considered compliant motion planning problems with n degrees of motion freedom, and k dimensions of variational geometric model uncertainty. We reduced this planning problem to the problem of computing preimages in an $(n + k)$-dimensional generalized configuration space, which encompasses both the motion and the model degrees of freedom, and encodes the control uncertainty as a kind of non-holonomic constraint. We developed algorithms and planners using this approach.

6.4 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.

Our work on the EDR theory has been presented in [Donald 86a,b; 87b, 88a;b, 89, 90]. In [Donald 88b, 89], 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.

6.5 Trajectory Generation and Optimal-Time Kinodynamic Planning

There exists a large body of work on optimal control in the control theory and robotics literature. For example, see [Hol, BDG, Sch, SS1, SS2]. Much of this work attempts an analytic characterization of time-optimal solutions—for example, to prove that in certain cases piecewise-extremal ("bang-bang") controls, with a finite number of switchings, suffice. This has led to many interesting and deep subresults. For example, [BDG, Hol] show how to rescale the velocity of a given trajectory $\Gamma = (\boldsymbol{p}, \dot{\boldsymbol{p}})$ to make it time-optimal while respecting dynamics constraints. Using these ideas, a number of authors have proposed heuristic or approximation algorithms for time-optimal trajectory planning. In particular, Sahar and Hollerbach [SH] and Shiller and Dubowsky [SD] both implemented algorithms for robots with full dynamics, employing configuration-space or phase-space grids. They did not bound the goodness of their approximations, nor the running times of their algorithm; however, their algorithms take time that grows exponentially with the number of gridpoints.

[CR] prove that the 3D polyhedral Euclidian shortest path problem is $\mathcal{NP}$-hard; a problem reduction extends this result to show that optimal kinodynamic planning in 3D is also $\mathcal{NP}$-hard. [Pap] gives a fully-polynomial approximation algorithm for the shortest path problem. Ó'Dúnlaing [O] provides an exact algorithm for one-dimensional kinodynamic planning. Kinodynamic planning in 2D is related to planning with non-holonomic constraints, as studied by Fortune and Wilfong [FW, W] and Jacobs and Canny [JC]; they consider a wheeled robot with a bounded minimum turning radius.

The results of Papadimitriou [Pap] on 3D Euclidian shortest paths gave us the idea that polynomial-time approximation schemes might be devised for min-time kinodynamic problems. [CDRX] introduce the algorithmic analysis approach to kinodynamic planning and provide the first provably good approximation algorithm for 2- and 3-dimensional optimal kinodynamic planning. The [CDRX] approach has been actively extended in collaboration with our graduate students at Cornell, Berkeley, and Duke.

[DX1] outline the improvements to these results and report on an implementation. [DX2,DX3] extend our approach to robots with L_2 dynamics bounds, and to open chain manipulators with rotational and prismatic joints.

[JHCP] give the first polynomial-time approximation algorithm for optimal kinodynamic planning for open chain manipulators with rotational and prismatic joints. Their work introduces new algorithmic and proof techniques, including discretizing acceleration-space according to the problem parameters and reducing state-dependent dynamics to being locally constant. Our approach [DX2,DX3] toward state-dependent dynamics basically the same as theirs; however, our algorithm only considers accelerations approximately at torque extremals. In concurrent independent work, [RT] obtain a polynomial-time approximation scheme for point robots obeying L_2 dynamics bounds.

7 References

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).

[AS] Asada, H., and Slotine, J-J. E. *Robot Analysis and Control,* John Wiley & Sons, New York (1986).

[BDG] Bobrow, J., Dubowsky, S., and Gibson, J. *Time-Optimal Control of Robot Manipulators,* Int. Jour. Robotics Research 4 (3) (1985).

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.

[BR79] O. Bottema and B. Roth. Theoretical Kinematics. , North Holland, Amsterdam, 1979.

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).

Briggs, Amy *An Efficient Algorithm for One-Step Compliant Motion Planning with Uncertainty,* 5^{th} ACM Symposium on Computational Geometry, Saarbrucken, Germany. (1989).

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).

[CDRX] Canny, J., Donald, B., Reif J., and Xavier, P. *On The Complexity of Kinodynamic Planning,* 29^{th} Symposium on the Foundations of Computer Science, White Plains NY (1988).

[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).

[Can89a] John Canny. *On computability of fine motion plans.* , IEEE ICRA, Scottsdale, AZ, (1989)

[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 **3** (3) pp. 219-236. (1988).

[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,* Ph.D. Thesis, Dept. EECS, MIT Artificial Intelligence Laboratory, MIT-AI-TR 982 (1987b).

Donald, B. R. *The Complexity of Planar Compliant Motion Planning with Uncertainty,* Proc. ACM Symposium on Computational Geometry, Urbana, IL June, 1988. *To appear in* Algorithmica. (1988a).

[D] **Donald, B. R.** *A Geometric Approach to Error Detection and Recovery for Robot Motion Planning with Uncertainty,* Artificial Intelligence **37** (1-3), Dec. 1988. pp. 223-271. (1988b).

Donald, B. R. *Planning Multi-Step Error Detection and Recovery Strategies,* Proc. IEEE International Conference on Robotics and Automation, Philadelphia, PA April, 1988. (1988c).

Donald, B. R. *Error Detection and Recovery in Robotics,* Lecture Notes in Computer Science, Vol. 336, Springer-Verlag, New York (1989). 314 pp. (1989).

Donald, B. R. *Planning Multi-Step Error Detection and Recovery Strategies,* International Journal of Robotics Research, **3** (1), March, (In press). (1990).

[DP89] **Donald, B. and D. Pai** *On the Motion of Compliantly-Connected Rigid Bodies in Contact, part II: A System for Analyzing Designs for Assembly,* Cornell CS tech Report (1989).

[DX1] **Donald, B.,and Xavier, P.** *A Provably Good Approximation Algorithm for Optimal-Time Trajectory Planning,* IEEE Int. Conf. On Robotics and Automation, Scottsdale, AZ (1989).

[DX2] **Donald B. and Xavier P.** *Near-Optimal Kinodynamic Planning for Robots With Coupled Dynamics Bounds, Fourth IEEE International Symposium on Intelligent Control,* (Sept. 24-26, 1989) Albany, NY. (1989).

[DX3] **Donald B. and Xavier P.** *Provably Good Approximation Algorithms for Optimal Kinodynamic Planning for Cartesian Robots and Open Chain Manipulators,* Cornell University, Technical Report, Computer Science Department (1989).

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).

[FW] **Fortune, S., and Wilfong, G.** *Planning Constrained Motion,* STOC, Chicago (1988).

[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).

[GV] **Golub, G., and VanLoan, C.** *Matrix Computations,* Johns Hopkins University Press, Baltimore, MA (1983).

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).

[HP] **Heinzinger, G., and Paden, B.** *Bounds on Robot Dynamics,* IEEE Int. Conf. On Robotics and Automation, Scottsdale, AZ (1989).

[Hol] **Hollerbach, J. M.** *Dynamic Scaling of Manipulator Trajectories,* MIT A.I. Lab Memo 700 (1983).

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.

[JC] **Jacobs, P., and Canny, J.** *Planning Smooth Paths for Mobile Robots,* IEEE Int'l. Conf. on Robotics and Automation, Scottsdale, AZ (1989).

J. Jennings, B. Donald, and D. Campbell *Towards Experimental Verification of an Automated Compliant Motion Planner based on a Geometric Theory of Error Detection and Recovery,* Proc. IEEE International Conference on Robotics and Automation, Scotsdale, Arizona (1989).

[JHCP] **Jacobs, P., Heinzinger, G., Canny, J., and Paden, B.** *Planning Guaranteed Near-Time-Optimal Planning in a Cluttered Workspace,* Int'l Workshop onf Sensorial Integration for Industrial Robots: Architectures & Applications, Zaragoza, Spain (1989).

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).

[LoP] **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).

[O] **Ó'Dúnlaing** *Motion Planning with Inertial Constraints,* NYU Robotics Laboratory TR 230 (1986).

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.

[Pap] **Papadimitriou, C.** *An Algorithm for Shortest Path Motion in Three Dimensions,* Information Proc. Letters, North Holland **20** pp. 259-263 (1985).

[Pai88] **Dinesh K. Pai.** Singularity, Uncertainty and Compliance of Robot Manipulators. , PhD thesis, Cornell University, Ithaca, NY, May 1988.

[PD89] **D. Pai and B. Donald** *On the Motion of Compliantly-Connected Rigid Bodies in Contact, part I: The Motion Prediction Problem,* Cornell CS tech Report (1989).

[PS] **Papadimitriou, C. and Steiglitz** *Combinatorial Optimization: Algorithms and Complexity,* Prentice-Hall, Englewood Cliffs, NY (1982).

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.

[RT] **Reif, J., and Tate, S.** *Approximate Kinodynamic Planning Using L_2-norm Dynamic Bounds,* Duke University Tech. Report (1989).

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).

[SH] **Sahar, G., and Hollerbach, J.** *Planning of Minimum-Time Trajectories for Robot Arms,* IEEE Int. Conf. On Robotics and Automation, St. Louis (1985).

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.

[Sch] **Schaettler, H.M.** *On The Optimality of Bang-Bang Trajectories in* $\Re^3$, Bull. AMS **16** (1). pp 113-6 (1987).

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).

[SD] **Shiller, Z., and Dubowsky, S.** *Global Time-Optimal Motions of Robotic Manipulators in the Presence of Obstacles,* IEEE Int. Conf. On Robotics and Automation, Philadelphia (1988).

[SS1] **Sontag, E., and Sussmann, H** *Remarks on the Time-Optimal Control of Two-Link Manipulators,* Proc. 24^{th} Conf. on Decision and Control, Ft. Lauderdale (1985).

[SS2] **Sontag, E., and Sussmann, H** *Time-Optimal Control of Manipulators,* Dept. mathematics, Rutgers, New Brunswick, NJ (1986).

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.

[vdW50] **B. L. van der Waerden.** Modern Algebra, *volume II.* , Frederick Ungar Publishing Co., 1950.

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).

[Whi76] **Whitney, D.**, "Force Feedback Control of Manipulator Fine Motions", *Journal of Dynamic Systems, Measurement, and Control,* June, 1977, pp. 91-97.

[Whi82] **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.

[W] **Wilfong, G.** *Motion Planning for an Autonomous Vehicle,* IEEE Int. Conf. On Robotics and Automation, Philadelphia (1988).

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.

8 Who Wrote This Paper

Prof. Bruce Donald is director of the Cornell Computer Science Robotics Laboratory. Dr. Dinesh Pai is a research associate in the laboratory. Patrick Xavier is a graduate student in Computer Science. Graduate students Jim Jennings, Amy Briggs, Mike O'Donnell, Marius Moscovici, and Bob Blondell also contributed in ways large and small to the work on this project.