Towards Task-Level Planning: Action-Based Sensor Design

Michael Erdmann

CMU-RI-TR-92-03

Also available as Computer Science Technical Report CMU-CS-92-116.

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

February 1992

Support for the research leading to this proposal was provided in part by Carnegie Mellon University, in part by an equipment grant from AT&T, and in part by a Research Initiation Award from the National Science Foundation under grant IRI-9010686.

Contents

1 Project Summary 3

2 Introduction 4

3 What is the information needed to solve a task? 4
- 3.1 Repositories of Information . . . 5
- 3.2 A Bewildering Array of Strategies . . . 5
- 3.3 Designing Sensors for Tasks . . . 5

4 Complexity 6
- 4.1 Planning with Perfect Sensing and Uncertain Control is Easy . . . 6
- 4.2 Design Philosophy . . . 6

5 A Family of Sensors for the Point-Into-Disk Problem 7
- 5.1 Actions and Uncertainty . . . 8
- 5.2 An Ideal Sensor . . . 8
  - 5.2.1 A World-Centered Sensor: Poor Degradation with Noise . . . 8
  - 5.2.2 A Task-Centered Sensor: Respects Uncertainty . . . 9
- 5.3 Designing a Sensor with Time-Indexed Backprojections . . . 9
- 5.4 Planning A Strategy . . . 12
- 5.5 Differential Backchaining . . . 12
- 5.6 Backchained Actions Provide an Implicit Progress Measure . . . 14
- 5.7 Cones of Progress . . . 14
- 5.8 Coverings by Cones of Progress . . . 15
- 5.9 Noise . . . 15
- 5.10 Summary . . . 16

6 Ongoing and Proposed Research 16
- 6.1 Two-Pin Two-Hole Task . . . 16
- 6.2 Theoretical Issues . . . 17
- 6.3 Proposed Experimental Domain . . . 18

7 A History of Manipulation Methodologies 20
- 7.1 Symbolic Planning . . . 20
- 7.2 Configuration Space . . . 20
- 7.3 Guarded Moves and Compliant Motion . . . 20
- 7.4 Skeleton Strategies . . . 21
- 7.5 Preimages . . . 21
- 7.6 Sensorless Manipulation . . . 22
- 7.7 Error Detection and Recovery . . . 22
- 7.8 Randomization . . . 22
- 7.9 Interwoven Strategies and Environments . . . 23
- 7.10 A Progression of Ideas . . . 23

8 Statement of Impact **24**

9 Acknowledgments **26**

10 Bibliography **26**

Abstract

This research proposes a method for automatically designing sensors from the specification of a robot's task, its actions, and its uncertainty in control. The sensors provide precisely the information required by the robot to perform its task, despite uncertainty in sensing and control. The key idea is to generate a strategy for a robot task by using a backchaining planner that assumes perfect sensing while taking careful account of control uncertainty. The resulting plan indirectly specifies a sensor that tells the robot when to execute which action. Although the planner assumes perfect sensing information, the sensor need not actually provide perfect information. Instead, the sensor provides only the information required for the plan to function correctly.

This report is a revised version of a proposal currently submitted to NSF.

Keywords: Uncertainty, Robotics, Information, Randomization, Automatic programming, Planning, Manipulation, Preimage

1 Project Summary

This research proposes a method for automatically designing sensors from the specification of a robot's task, its actions, and its uncertainty in control. The sensors provide precisely the information required by the robot to perform its task, despite uncertainty in sensing and control. The key idea is to generate a strategy for a robot task by using a backchaining planner that assumes perfect sensing while taking careful account of control uncertainty. The resulting plan indirectly specifies a sensor that tells the robot when to execute which action. Although the planner assumes perfect sensing information, the sensor need not actually provide perfect information. Instead, the sensor provides only the information required for the plan to function correctly.

In the past, models of sensing have been driven from the bottom, by questions of the form "What are the characteristics of sensor X?", "What is the error?", "How can we use the sensor?". We are proposing models of sensing that are driven from the top, by questions of the form "Given a task, what sensor characteristics are required?", "How much uncertainty can be tolerated?", "How does sensing depend on action?".

This research will investigate special purpose sensors for simple assembly tasks. These tasks include multiple-peg-in-hole assembly, parts identification, parts orienting, and parts sorting. The long-range goal of this research is to develop task-level planners capable of translating high-level specifications into low-level robot commands. Sensor design is a means of understanding the information requirements of manipulation tasks and thus a means of improving task specifications. In addition, sensor design is one approach for breaking the complexity barrier that impedes practical planning in the presence of uncertainty. This research could lead to improved parts design, improved sensor design, and more efficient automatic programming tools.

2 Introduction

The goal of robotics is to develop autonomous systems. In order to achieve this goal, we must understand the relationship between action and sensing, and we must be able to convert that understanding into robots capable of operating productively and successfully in an uncertain world.

The nature of the relationship between action and sensing raises some fundamental questions:

- What is the information needed to solve a given manipulation task?
- What tasks can be solved by a given repertoire of operations?
- How sensitive are solutions of tasks to particular assumptions about the world?

Obtaining answers to these questions forms the central research agenda in robotics. In this proposal we focus on the first question.

Our goal in pursuing this line of research is to improve and automate robot programming tools. Our long-range research agenda is to develop a high-level representation of tasks that facilitates the decomposition of robot tasks into modular subtasks and the translation of high-level specifications into low-level robot commands. This translation and decomposition process is severely complicated by uncertainty. We review some of the relevant work in Section 7. Given uncertainty, our current best approach for improving robot programming and developing task-level planners is to understand the information requirements of robot tasks.

We propose a method for connecting action to sensing, by designing sensors particularly suited to the actions available to the robot. The sensors provide precisely the information required by the robot to function, despite uncertainty. This approach stands in contrast to the idea of using available sensors and manipulators to accomplish a given task. Indeed, the approach in this proposal is to control sensing as one controls action.

Although the problem of controlling sensing and action may seem to be more complicated than the problem of controlling only action, by using some simple guiding principles, this extra degree of controllability can actually simplify the problem of planning and executing manipulation strategies in the presence of uncertainty. The primary principle is to generate a strategy that tightly integrates action and sensing by first temporarily decoupling action from sensing. The key idea is to generate a strategy for a given task by using a backchaining planner that assumes perfect sensing but that accounts for control uncertainty. Given such a strategy, one then designs a sensor. Although the planner assumes perfect sensing information, the sensor provides only the information required for the strategy to function correctly, despite noise in the sensor and uncertainty in control.

3 What is the information needed to solve a task?

This question is a most important question in robotics. Within it are contained the basic issues of task definition, action specification, environment design, and sensing capabilities.

3.1 Repositories of Information

When we ask "what information is required?" some answers seem to appear immediately. Yet, when we look more closely these answers become fuzzy. The answers that appear immediately are of the form: some information arises from sensors, other information is encoded in the mechanism performing the task, some information is encoded in the mechanism's interpretation of the world, some in its predictive ability, and still other information acts subtly through the environment. It is obvious why these answers are fuzzy. They are fuzzy because there is no clear notion of how the information is distributed between these different repositories of information.

3.2 A Bewildering Array of Strategies

As an example, imagine the task of placing a pencil into an electric pencil sharpener in order to sharpen the pencil. There are numerous methods for accomplishing this task. One is to grasp the pencil then move one's hand slowly while carefully looking at the pencil and the sharpener, continuously readjusting one's aim as the pencil comes close to the sharpener's hole. Another is to close one's eyes, place one hand on the pencil sharpener, then with the other hand bring the pencil to the sharpener, and guide it manually into the hole using tactile feedback. A third strategy is to visually memorize the state of the world then move the pencil into the sharpener with no feedback whatsoever. A fourth strategy consists of randomly stabbing at the sharpener with the pencil until the pencil enters the sharpener. A fifth strategy consists of building a linkage of some sort that can insert the pencil into the sharpener from a large set of initial states. The linkage is specially designed as a function of the sharpener's geometry and the possible starting locations of the pencil. A sixth strategy is to dispense with the notions of pencil and sharpener, and instead build a series of tightly coupled sensory feedback loops, implemented on top of hardware that can grasp and move long thin objects. The feedback loops might implement such operations as pencil-seeking, pencil-grasping, and hole-seeking. Numerous other strategies may be constructed as hybrids of these six strategies.

This enormous array of strategies leaves one with a distinct feeling of unease. In particular, the strategies seem to gain their information very differently. Some rely heavily on sensors, others on mechanical devices, and yet others on prediction. This makes it very difficult to answer the question "what information is required to solve the task of putting a pencil into a sharpener?" Indeed, we could complicate the question even more by varying the sizes and shapes of the pencil and the sharpener. One is tempted to dismiss the question as non-sensical, based on the apparent realization that there is no intrinsic measure of required information and even no intrinsic task of the form Put the Pencil in the Sharpener.

3.3 Designing Sensors for Tasks

We will resist this temptation, and instead introduce time as a measure relative to which one can study information requirements. Specifically, one can first seek the information required to accomplish a task as quickly as possible. Relative to this base standard one can then compare other strategies and their information requirements.

A related approach is to realize that while all the strategies mentioned above are defined relative to a given world and a given repertoire of actions, the strategies have different applicability as one varies these worlds. In particular, if one increases uncertainty, or if one changes the tolerances of the world, or if one adds new features to the world, then some strategies will remain in effect, while others will falter. The degree to which the original world may be be modified gives a measure of the true information requirements of the task and thus lends substance to the task abstraction. Equivalently, the extent to which information must be added, either by adding new sensors or mechanically constraining the motions of the robot in the world, gives a measure of the intrinsic nature of the task. This reasoning thus also leads us to the idea that one should study the problems of designing sensors and of designing mechanical devices that provide precisely the information required to accomplish a task with a given repertoire of actions in a given amount of time. In this proposal we focus on sensor design.

4 Complexity

This section argues for sensor design based on an examination of the complexity of several different approaches to motion planning. We draw our complexity estimates from two domains. One domain consists of discrete state spaces with discretely specified actions. The actions themselves have non-deterministic or probabilistic transitions. See [Papadimitriou 1985], [Papadimitriou and Tsitsiklis 1987], [Bertsekas 1987], and [Erdmann 1990] for further details. The other domain consists of continuous spaces with continuous actions, analogous to those in the LMT [Lozano-Pérez, Mason, and Taylor 1984] preimage methodology. Again, the actions are either non-deterministic or probabilistic. See [Natarajan 1986, 1988], [Canny and Reif 1987], [Canny 1988a, 1989], and [Donald 1988b] for further details.

4.1 Planning with Perfect Sensing and Uncertain Control is Easy

Table 1 summarizes the basic results. We omit many details. One notices the following pattern: for perfect sensing, planning seems to be easy; for general sensing, planning seems to be intractable; and for sensorless strategies, planning seems to be hard, but not as hard as in the general case.

The interesting observation is that planning with perfect sensing is usually easy, even in the presence of control uncertainty. Indeed the complexity of deciding the existence of a strategy for attaining the goal with uncertain control seems to be nearly that of determining the connectivity of the state space.

4.2 Design Philosophy

Previous work on planning in the presence of uncertainty has considered the general problem, in which both sensing and control are subject to uncertainty, and the special problem, in which sensing is completely absent and control is uncertain. These problems correspond to the top two rows of Table 1. These problems naturally lie in an intractable complexity class. The results of Table 1 suggest that it makes sense to look at problems in which sensing is perfect. One is reluctant to do so, since perfect sensing is impossible. There is a much more attractive

		Control Uncertainty Task Space; Strategy Type		
		Perfect Discrete; Optimal	Probabilistic Discrete; Optimal	Adversarial Continuous; Guaranteed
Sensing Uncertainty	Sensorless	P	NP	$2^{(n\,k)^{O(1)}}$
	Imperfect	PSPACE	PSPACE	$2^{2^{O(n\,k)}}$
	Perfect	P	P	P

Table 1: A rough complexity picture. In each case the problem is to decide the existence of a k-step strategy for attaining the goal from some initial region. The right-most column concerns tasks on continuous spaces, with the objective of finding a guaranteed strategy that attains the goal. The other two columns concern tasks on discrete spaces, with the objective of obtaining an optimal strategy. The complexity classes are classifications, except for the two exponentials, which are known upper bounds. The number of obstacle constraints is n. ("Perfect" means that there is <u>no</u> uncertainty of that type; "Adversarial" refers to unknown but bounded errors.)

view, however. This view envisions sensing as a design problem. For any task, one can assume perfect sensing in determining the control strategy for accomplishing that task. Having done so, one then builds an imperfect sensor that nonetheless provides the information required for this control strategy to operate. One big research issue is understanding the degree to which degradation of information still permits the control strategy to operate. Another research issue to decide whether an abstract sensor generated by this design process may be built, both geometrically and physically.

5 A Family of Sensors for the Point-Into-Disk Problem

The basic thesis of this proposal is that one should explore the special purpose design of sensors based on a system's available actions, in order to more easily accomplish a given manipulation task in the presence of uncertainty. The encompassing research objective is to obtain an understanding of the knowledge requirements of manipulation tasks.

In this section we illustrate these goals with a simple task. The task is to move a point in the plane into a circular disk centered at the origin of the plane. We will look at the design of a sensor for this task, that is specifically structured to guide the point into the disk.

In the process, we will see that it is possible to design a family of sensors, indexed by the speed with which the point approaches the goal. **The family of sensors defines a tradeoff between the optimal approach strategy that is possible with perfect sensing and the motions possible given the actual information available from an imperfect sensor.**

Let us note in passing that the point-into-disk problem is the archetypical abstraction of the assembly problem. Whenever two parts must be mated, the problem can be viewed as a point-into-volume problem in some appropriate parameter space [Lozano-Pérez 1981, 1983]. With this configuration space transformation in mind, we sometimes refer to the moving point as a "robot".

5.1 Actions and Uncertainty

Let us define the actions available for the point-into-disk problem. The system is a simple first-order system, in which the point is moved by commanding velocities that are subject to unknown but bounded errors. Specifically, if the execution system commands velocity $\mathbf{v}_0$, then the range of possible velocities is given by the ball $B_{\epsilon_v |\mathbf{v}_0|}(\mathbf{v}_0)$, where

$$B_{\epsilon_v |\mathbf{v}_0|}(\mathbf{v}_0) = \left\{ \mathbf{v}^* \;\middle|\; |\mathbf{v}_0 - \mathbf{v}^*| \leq \epsilon_v |\mathbf{v}_0| \right\}.$$

Another often useful way to think of velocity uncertainty is as a cone of possible velocity directions. For a nominal commanded velocity $\mathbf{v}_0$ and error radius $\epsilon_v |\mathbf{v}_0|$, the cone has axis given by $\mathbf{v}_0$, and error angle $\sin^{-1}(\epsilon_v)$.

A basic action consists of commanding a velocity $\mathbf{v}_0$ for some duration of time Δt, where Δt can be arbitrarily small. This model of action resembles closely the idea that the executive system will in a nearly continuous fashion execute a velocity for a small amount of time, determine the robot's new location, then execute a new velocity.

Now, suppose that the initial position of the point is known to lie in some set R, and suppose that one commands velocity $\mathbf{v}_0$ for time Δt. Assuming no obstacles, the resulting locations of the point are described by the set sum $R \oplus B_{\epsilon_v |\mathbf{v}_0| \Delta t}(\mathbf{v}_0 \Delta t)$. In other words, each possible initial location is offset by a possible velocity scaled by the elapsed time.

5.2 An Ideal Sensor

Let us ask ourselves what the ideal sensor for the point-into-disk task might look like. A sensor that reports precisely the current position of the point would certainly be nice. In particular, given a position reported by this sensor, the execution system can calculate a velocity vector that aims directly towards the goal. We will see presently that this perfect position sensor is actually superideal, in that it provides more information than is needed.

5.2.1 A World-Centered Sensor: Poor Degradation with Noise

One issue to keep in mind is how one might implement a perfect position sensor. One possibility is to have a physical sensor that estimates the robot's position by observing it. A camera is a good example. Another implementation might consist of a series of encoders attached to the control system for the robot. This is a form of dead-reckoning. A third implementation might consist of a grid of rectangular wires implanted in the plane, that report back the (x, y) position of the robot.

Observe that all of these sensors are world-centered. Specifically, they report back the configuration of the robot in some world coordinate system. Given this configuration, the run-time executive then calculates the difference in position between the robot and the goal in order to suggest a commanded velocity. This procedure is fine for a truly perfect sensor, but degrades poorly as the sensor becomes noisy or as the position of the goal becomes uncertain. In particular, suppose we model the uncertainty of such a world-centered sensor as an error ball with a fixed error radius. If the error radius is larger than the goal, then as the robot approaches close to its goal, the system will be unable to decide on which side of the goal the robot is located. This makes it impossible to execute a velocity guaranteed to attain the goal.

While this example is simple, the problem is fundamental. It exists in even greater magnitude for general manipulation tasks. In these tasks the information available to the robot may consist not just of positional information, but also of information coming from such sources as force sensors, history, and model-based prediction. It is the interaction of all these sources of world-centered information that makes the planning problem so difficult.

5.2.2 A Task-Centered Sensor: Respects Uncertainty

A second approach is to build a task-centered sensor. What might such a sensor for the point-into-disk task look like? A good representation for a perfect sensor is not a cartesian grid but a polar coordinate grid that is centered at the goal. As with the perfect cartesian sensor, the perfect polar coordinate sensor reports back the configuration of the robot relative to the hole, but in polar coordinates. The coordinates are then again used to compute a velocity vector along which the robot should move.

Fortunately, we can go yet a step further. The polar coordinate sensor reports back both the distance of the robot from the goal and the angle that measures the relative direction between the robot and the goal. Only the angle is used to calculate a motion command; the distance is ignored. This observation immediately suggests that the ideal sensor need only report the angular coordinate of the polar coordinate representation. In short, the ideal perfect sensor is a directional beacon situated at the goal, that reports the direction from the goal to the robot. This sensor senses one degree of informational freedom.

Let us observe that this task-centered sensor degrades nicely with noise. In particular, an error of a few degrees in the directional beacon simply means that the robot will move slightly in the wrong direction. Near the goal, the sensor can quickly detect and correct such inaccurate motions. In contrast, for the cartesian sensor, slight positional errors near the goal can lead to large directional uncertainties that remain undetected for long times.

There are two principles at play here. First, the sensor is task-centered. Second, the sensor error mirrors the control error. We will explain this second principle in the next few paragraphs.

5.3 Designing a Sensor with Time-Indexed Backprojections

The basic procedure for designing both the sensor and the strategy that together accomplish the task is to backchain from the goal under the assumption of a perfect sensor. The reason for

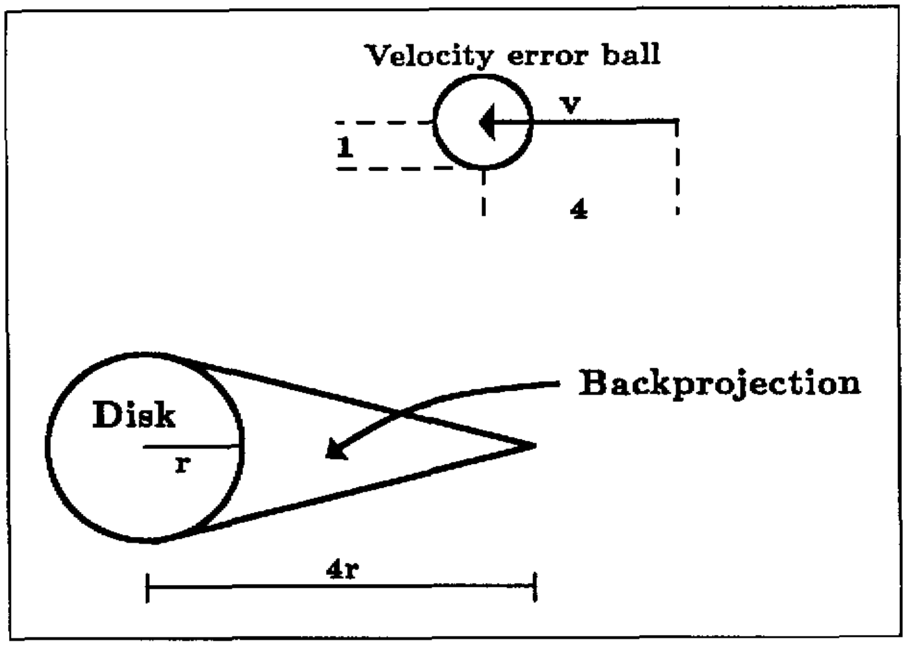

Figure 1: This figure shows a typical backprojection of a two-dimensional disk.

using a perfect sensor is that it considerably simplifies the planning problem, without forcing us into a particular implementation of the sensor. The backchaining process tells us what information is required, how it is used, and how to design a sensor so that the information degrades nicely in the presence of uncertainty.

We can describe the backchaining process inductively. Initially, the planner constructs a collection of backchaining regions from the goal disk. At each subsequent stage, the planner constructs a collection of backchaining regions from some other circular region. The semantics are that at each stage there is some disk within which a strategy for accomplishing the goal in the presence of perfect sensing and imperfect control has been constructed. The job of the planner is to enlarge this strategy to a slightly larger disk.

In order to understand the operation of the planner, consider first the problem of forming a *backprojection* of a circular disk relative to some commanded velocity. See [Erdmann 1984, 1986]. In brief, the backprojection of a goal relative to some commanded velocity consists of those configurations from which the commanded velocity is guaranteed to cause the robot to pass through the goal, despite uncertainty in the actual velocity of the robot. Figure 1 shows a typical backprojection of a disk of radius r. The commanded velocity in this case is $\mathbf{v}$, which points in the direction $(-1, 0)$, and the velocity error ball has radius $\epsilon_v\,|\mathbf{v}|$, with $\epsilon_v = \frac{1}{4}$. For each possible velocity direction one can compute such a backprojection.

These backprojections are larger than what we want. Recall that our basic model of an action is a pair: a velocity $\mathbf{v}_0$ and a time Δt. This model of action leads to time-indexed backprojections. See Figure 2. The semantics are as follows. If the robot starts off in the crescent-shaped region depicted in the figure, then upon execution of velocity $\mathbf{v}_0$ for time Δt, the robot is guaranteed to be somewhere inside the disk of radius r_k, despite control uncertainty. Formally, we have that the time-indexed backprojection of a goal $\mathcal{G}$ under the

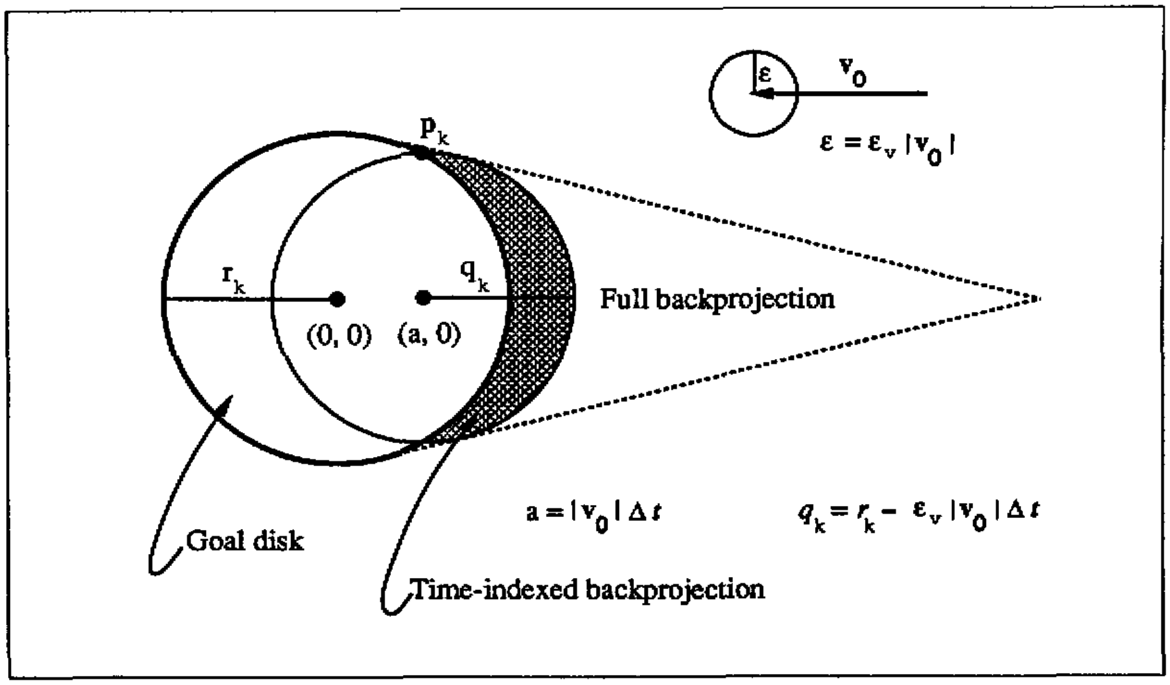
ε
v_0
$\varepsilon = \varepsilon_v |v_0|$
p_k
r_k
q_k
(0, 0)
(a, 0)
Full backprojection
$a = |v_0| \Delta t$
$q_k = r_k - \varepsilon_v |v_0| \Delta t$
Goal disk
Time-indexed backprojection

Figure 2: This figure shows a typical time-indexed backprojection of a two-dimensional disk. The goal disk has radius r_k and is centered at the origin. The action is to execute the nominal velocity $\mathbf{v}_0$ for time Δt. The velocity uncertainty is given by an error ball with radius $\epsilon_v |\mathbf{v}_0|$. The resulting time-indexed backprojection is a disk with radius $q_k = r_k - \epsilon_v |\mathbf{v}_0| \Delta t$ centered at $(a, 0) = -\mathbf{v}_0 \Delta t$. The shaded crescent-shaped area is the portion of the backprojection that lies outside of the goal disk. The point $\mathbf{p}_k$ is one of the intersection points of the circles that circumscribe the two disks. For comparison, the dashed lines outline the full backprojection. See also Figure 1.

action $(\mathbf{v}_0, \Delta t)$ is the set $\mathcal{B}_{\mathbf{v}_0,\Delta t}(\mathcal{G})$, with

$$\mathcal{B}_{\mathbf{v}_0,\Delta t}(\mathcal{G}) = \left\{ \mathbf{p} \;\middle|\; B_{\epsilon_v |\mathbf{v}_0| \Delta t}(\mathbf{p} + \mathbf{v}_0 \Delta t) \subseteq \mathcal{G} \right\}.$$

Here $\mathbf{p}$ is a point in the plane and $B_{\epsilon_v |\mathbf{v}_0| \Delta t}(\mathbf{p} + \mathbf{v}_0 \Delta t)$ is the two-dimensional ball of radius $\epsilon_v |\mathbf{v}_0| \Delta t$ centered at the position $\mathbf{p} + \mathbf{v}_0 \Delta t$. (We are assuming no obstacles.)

Now, observe that if $\mathcal{G}$ is a disk of radius r centered at the origin, then we can rewrite $\mathcal{B}_{\mathbf{v}_0,\Delta t}(\mathcal{G})$ as

$$\begin{aligned}
\mathcal{B}_{\mathbf{v}_0,\Delta t}(\mathcal{G}) &= \left\{ \mathbf{p} \;\middle|\; |\mathbf{p} + \mathbf{v}_0 \Delta t + \Delta \mathbf{p}| \leq r, \text{ for every } \Delta \mathbf{p} \text{ with } |\Delta \mathbf{p}| \leq \epsilon_v |\mathbf{v}_0| \Delta t \right\} \\
&= \left\{ \mathbf{p} \;\middle|\; |\mathbf{p} + \mathbf{v}_0 \Delta t| \leq r - \epsilon_v |\mathbf{v}_0| \Delta t \right\} \\
&= B_{r - \epsilon_v |\mathbf{v}_0| \Delta t}(-\mathbf{v}_0 \Delta t).
\end{aligned}$$

In other words, the time-indexed backprojection of a disk of radius r is formed by first shrinking the radius of this disk by the positioning uncertainty $\epsilon_v |\mathbf{v}_0| \Delta t$, then shifting the resulting disk by the negative of the commanded positional change $\mathbf{v}_0 \Delta t$. This is exactly what one would expect intuitively. Figure 2 shows both a goal disk of radius r_k and its time-indexed backprojection.

5.4 Planning A Strategy

During the first of the backchaining stages, the goal disk in Figure 2 corresponds to the task-level goal. In later stages, the disk constitutes a composite subgoal, formed from the union of several crescent-shaped backprojections.

During a given stage of the backchaining process, the planner expands the current composite subgoal of radius r_k to a new composite subgoal of radius $r_k + (1 - \epsilon_v)\,|\mathbf{v}_0|\,\Delta t$. Specifically, the planner computes time-indexed backprojections for all possible velocity directions. It holds fixed the magnitude of the commanded velocity $|\mathbf{v}_0|$ and the duration of time Δt, but rotates the direction of the commanded velocity through 2π radians. The union of the resulting crescent-shaped backprojections forms a band of width $(1 - \epsilon_v)\,|\mathbf{v}_0|\,\Delta t$ about the old composite subgoal of radius r_k. Taken together, the band and the old subgoal form the new subgoal.

The idea is that if the robot starts off in a crescent-shaped backprojection, and if the executive commands velocity $\mathbf{v}_0$ for time Δt, then the robot is guaranteed to wind up in the composite subgoal disk. Once in this disk, the robot can then execute whatever strategy it has recursively computed for attaining the next composite subgoal, and so forth.

5.5 Differential Backchaining

Suppose that one repeatedly computes time-indexed backprojections, starting with the goal disk of radius r, and backchaining over and over. At the $k + 1^{st}$ stage one has a composite subgoal of radius r_k centered at the origin. Here k runs from 0 on upward, with $r_0 = r$. Given r_k and $\mathbf{v}_0\,\Delta t$, one computes a time-indexed backprojection, which is a disk of radius q_k centered at $-\mathbf{v}_0\,\Delta t$. The circles bounding these two disks intersect at the point $\mathbf{p}_k$ and its mirror image (relative to the line given by $\mathbf{v}_0$).

For simplicity, let us assume that $\mathbf{v}_0$ is parallel to the x-axis and that it is pointing in the negative x direction. Let us define $a = |\mathbf{v}_0\,\Delta t|$, and write $\mathbf{p}_k = (x_k, y_k)$. Then we observe that $q_k = r_k - \epsilon_v\, a$ and that $r_k = r + k\,(1 - \epsilon_v)\, a$. By solving for the intersection of two circles, it follows that

$$x_k = r\,\epsilon_v + k\,(1 - \epsilon_v)\epsilon_v\, a + \frac{1}{2}(1 - \epsilon_v^2)\, a.$$

Furthermore, y_k is implicitly defined by the equation $x_k^2 + y_k^2 = r_k^2$.

The interesting question is what happens as we make the time constant Δt of the feedback loop very small. Suppose we let $a \to 0$. In order to backchain out to some location $\mathbf{p}$, we must therefore let k approach infinity in such a way that the product $k\,a$ remains constant, say at some value c that depends on $\mathbf{p}$. Thus we have that

$$\begin{aligned} x_k &\to \epsilon_v \Big(r + c\,(1 - \epsilon_v)\Big), \\ y_k &\to \sqrt{1 - \epsilon_v^2}\,\Big(r + c\,(1 - \epsilon_v)\Big). \end{aligned}$$

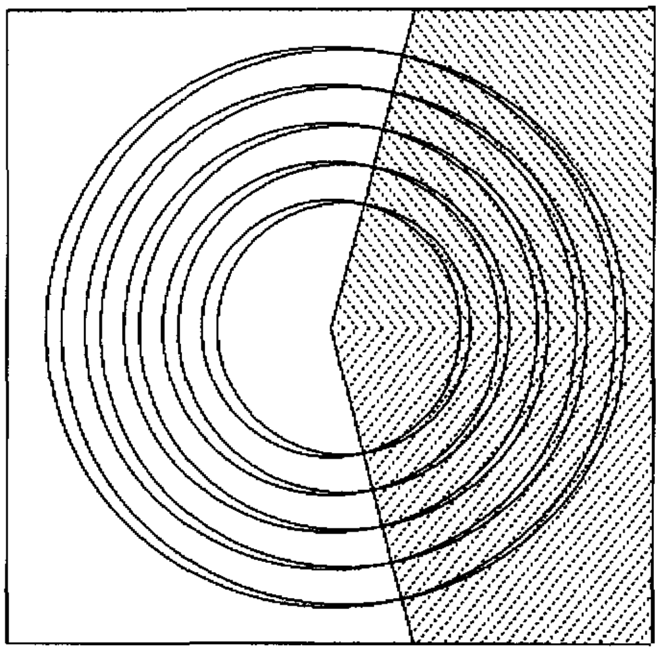

Figure 3: The two solid lines bound a cone of progress, which is indicated by the shaded region. The nominal commanded velocity $\mathbf{v}_0$ is the vector $(-1,0)$, with uncertainty $\epsilon_v = 0.25$. If the robot starts within the cone of progress, then commanding velocity $\mathbf{v}_0$ for a differential amount of time is guaranteed to move the robot closer to the origin, despite the uncertainty in control. Each pair of circles comprises a composite subgoal disk and its time-indexed backprojection relative to the velocity $\mathbf{v}_0 = (-1,0)$ and the time duration $\Delta t = 0.1$. The radius of the first subgoal is $r_0 = 1$. Observe that the solid lines nearly trace out the intersection points of each subgoal disk with its time-indexed backprojection. As Δt becomes smaller, this match becomes better. See also Figure 2.

We therefore see that

$$\frac{y_k}{x_k} \to \frac{\sqrt{1-\epsilon_v^2}}{\epsilon_v}.$$

In other words, the intersection points $\{\mathbf{p}_k\}$ line up in a straight line with slope $\sqrt{1-\epsilon_v^2}\big/\epsilon_v$.

One way to interpret this is as follows. Suppose we perform the backchaining process for all velocities and all execution times, until we have covered the entire plane with composite subgoals. Now, fix the velocity vector $\mathbf{v}_0 = (-1,0)$, and allow the time Δt to approach zero. Consider the union of all the crescent-shaped portions of the time-indexed backprojections computed during the backchaining process relative to the actions $\{(\mathbf{v}_0, \Delta t)\}_{\Delta t \downarrow 0}$. The union is a semi-infinite cone, with apex at the origin. The cone is bounded by two lines whose slopes are $\pm\sqrt{1-\epsilon_v^2}\big/\epsilon_v$. Figure 3 depicts the bounding cone. Also shown in the figure are pairs of circles. In each case one circle represents a composite subgoal of radius r_k for some k, while the other circle has radius q_k and is a time-indexed backprojection of the subgoal, for a small value of Δt.

What does this cone mean? The interior of the cone consists of all those locations at which the nominal velocity $\mathbf{v}_0$ is guaranteed to make progress towards the goal for a differential amount of time. This is clear from the manner in which the cone was constructed. It is also clear when one observes that each of the bounding edges of this cone is perpendicular to one of the extreme velocity vectors in the velocity error ball about the nominal velocity $\mathbf{v}_0$. This means that if the robot is located on the edge of the cone then there is a possible velocity which will cause the robot to remain differentially at the same distance from the goal.

5.6 Backchained Actions Provide an Implicit Progress Measure

It is useful to understand the reason that backchaining generates such cones. Effectively, the backchaining process defines an implicit progress measure on the plane. For instance, consider the case in which we hold $|\mathbf{v}_0|\,\Delta t$ constant, while varying the direction of $\mathbf{v}_0$. In that case, the number of backchaining steps k required in order to include a given point $\mathbf{p}$ of the plane within some subgoal disk of radius r_k defines a discrete progress measure. In the general case, as $k \rightarrow \infty$ and $\Delta t \rightarrow 0$, we see that for a given point $\mathbf{p}$ the product $k\,(1-\epsilon_v)\,a$ simply reduces to the distance of the point from the task goal.

With this description in mind, we will refer to the cone as a *cone of progress.* Specifically, it is a cone of progress relative to the commanded velocity $\mathbf{v}_0$.

5.7 Cones of Progress

The previous construction should come as no surprise. The cone appeared in a different form in [Erdmann 1990] in the context of randomization. That work discusses a simple feedback loop for accomplishing the point-into-disk task, despite uncertainty in sensing and control. The feedback loop operated as follows. The robot would sense its current position. If all position interpretations consistent with the observed sensor value lay wholly within some cone of progress, then the robot would determine a velocity vector and a maximum execution time that would permit the robot to move closer to the goal. On the other hand, if no single cone of progress contained all the possible locations of the robot, then no action was guaranteed to move the robot closer to the goal. In that case, the robot would execute a random motion.

There are two important ideas contained in the last paragraph. The first concerns speed of progress. The second is a description of a better sensor. First, in the randomization example of [Erdmann 1990], the sensor was a cartesian sensor whose uncertainty was modelled as an error ball. This meant that for observed sensor values close to the origin, the system could not decide that its position lay in any particular cone of progress. Conversely, for observed sensor values far away from the origin, the location of the sensor interpretation set inside a cone of progress determined the amount of progress that the system could make. In other words, we see that it is the relationship of a particular sensor and sensed value to all the cones of progress that determines whether progress is possible and if so, how much progress. This is the sensing-speed tradeoff that we mentioned earlier.

The second idea concerns the design of a task-specific sensor. We argued earlier based on simple symmetry that the sensor should be an angular sensor that is centered at the goal and that looks out from the goal toward the robot. That argument finds further support in the current geometric reasoning. In order for the robot to be certain that a particular action will move it closer to the goal, the robot needs to know that its current position lies within the cone of progress associated with that action. This requirement points towards the design of a sensor that recognizes cones of progress. For the example above, this means that we need a sensor that can decide whether the robot is located in some conical region with half-angle $\sin^{-1}(\epsilon_v)$ that is centered at the goal.

In summary: **Good sensors should recognize cones of progress.**

A corollary: **Sensors should not recognize states;**
Sensors should recognize applicable actions.

5.8 Coverings by Cones of Progress

There are two technicalities that we have yet to address. First, designing a sensor to recognize a single cone of progress is not enough. The entire state space need not be covered by a single cone of progress. This leads us to a family of sensors. At one extreme we have a sensor that is capable of recognizing the robot's presence or absence in each and every cone of progress. For the example above this amounts to a perfect angular sensor. Given such a sensor we would at run-time see the robot executing a nearly straight-line trajectory from its initial location to the goal.

At the other extreme we have a sensor that is capable at run-time of recognizing the robot's presence in at least one cone of progress, but not necessarily more. One view of such a sensor is as follows. We, as sensor designers, decide on a covering of the state space by some small collection of cones of progress. We then design a sensor that can recognize this covering. In other words, at run-time the sensor can accurately report the robot's position in at least one cone of progress. There may be many such sensors, since there may be many different minimal coverings of the state space.

As an example, for small values of ϵ_v, the state space of the point-into-disk task can be covered by four cones of progress. For instance, we might take these four cones to be the cones of progress whose axes are aligned with the four half-axes of the x-y coordinate system. Given this covering, the sensor merely has to decide in which of four angular regions the robot is located. At run-time we would observe the robot executing a series of velocities whose commanded directions are aligned with the axes. The net effect would be to pull the robot in towards the goal along a path near the diagonals of the state space. The speed of progress in this case is less than that achievable with a perfect angular sensor.

5.9 Noise

The second technicality that we need to address is noise. Consider a particular physical implementation of a sensor. We can view the sensor as mapping an observed sensed value to a collection of cones of progress, each of which is certain to contain the actual position of the robot. A perfect sensor maps an observed sensed value to all cones of progress that contain the robot's actual location. As the sensor becomes noisier, the sensor maps the observed sensed value to a smaller collection of cones of progress.

So long as the mapping from observed sensor values to collections of cones of progress is well-defined, there is no problem. Progress may be slow, but so long as there is at least one cone of progress for each possible sensor value, progress is certain. In the case of the cartesian sensor mentioned earlier, we saw that the mapping was not always well-defined. In particular, for observed robot positions close to the goal, there was no cone of progress that was certain to contain the actual position of the robot.

For the point-into-disk task above, we can imagine a physical sensor that returns some angle θ^* as an estimate of the robot's position relative to the goal. To say that the sensor is perfect is to say that the robot is located precisely at that orientation relative to the goal. To say that the sensor is noisy may mean that the actual angular location of the robot lies in some interval $(\theta^* - \gamma, \theta^* + \gamma)$, where γ is an error angle that measures the sensor's uncertainty. So long as γ is less than $\sin^{-1}(\epsilon_v)$, we can be sure that the sensor can recognize a cone of progress. This condition thus creates a design constraint on any physical implementation of the abstract sensor defined by the cones of progress.

5.10 Summary

The cones of progress specify exactly what knowledge the robot must have in order to make progress, and how the available information affects the speed of progress. A sensor should be viewed as recognizing cones of progress. Given this description, one can then design physical sensors that try to recognize as many cones of progress as one requires in order to accomplish a task quickly.

6 Ongoing and Proposed Research

6.1 Two-Pin Two-Hole Task

We are currently studying the problem of inserting a two-pin part into a pair of circular holes. In particular, we are examining the character of a robust sensor that provides precisely the information needed to perform this insertion as quickly as possible. The problem forces us to consider rotations in addition to translations. We are studying this problem from several perspectives. In particular, we have performed a theoretical analysis, we have written a simulator, and we have programmed the task as an assembly operation on the Zebra Zero manipulator. The theoretical considerations suggest that the ideal sensor is a sensor that recognizes radial lines in the (x, y, θ) configuration-space of the two-pin part. It is difficult to implement such a sensor physically. However, by combining two of the translational radial sensors used for the single peg-into-hole problem, one can obtain an approximation to the ideal sensor. See Figure 4. This approximation provides a close match to the ideal sensor for configurations of the part near the goal, where the sensor is needed the most. In particular, the approximation improves as the part moves towards its goal, obtaining an ideal match at the goal configuration. Such limiting behavior is unusual, in that standard cartesian sensors, that are not explicitly designed for a specific task, exhibit growing divergence as the part moves towards the goal.

We expect that for many tasks the design of a task-specific sensor will involve steps similar to those for the two-pin part. The design process will use a configuration-space analysis coupled with a backchaining planner to generate an ideal sensor. The resulting configuration-space sensor will provide a description of the precise information required to

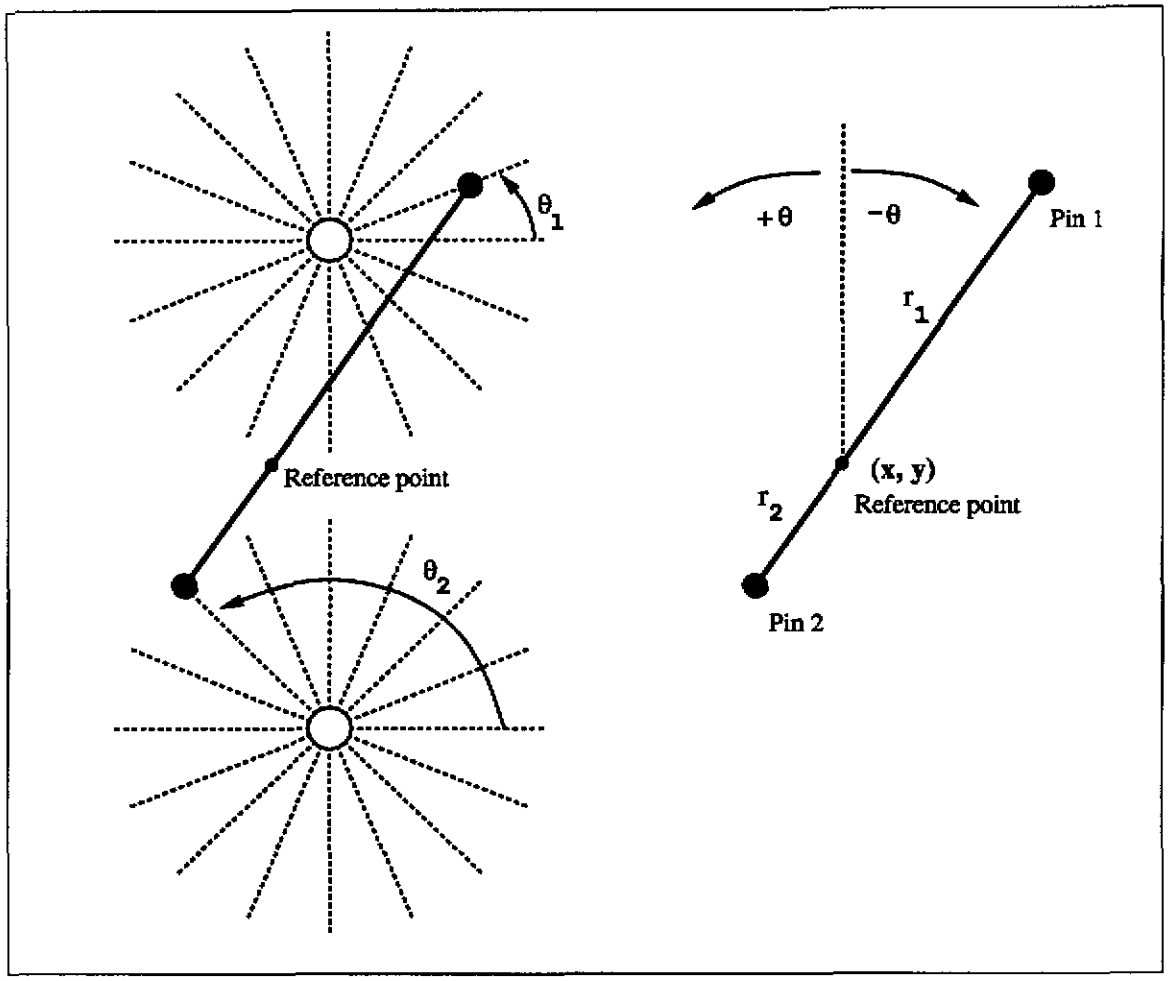

Figure 4: Relationship of two radial sensors to the configuration of a two-pin part. Each sensor is centered at one of the goal holes. The sensors report back two angles, (θ_1, θ_2). θ_i is the angle that pin i makes relative to hole i.

solve the task. The ideal sensor will generally have no immediate physical realization. Instead, the sensor serves as a guide, suggesting different physical implementations as approximations. An important and as yet unexplored step of the design process is the transformation of an ideal configuration-space sensor into a physically valid sensor.

6.2 Theoretical Issues

There are numerous interesting theoretical questions that we would like to explore:

- The construction of Section 5 is almost a general algorithm. We would like to describe the general process for designing an ideal sensor. We believe that in many cases this general process is of the same character as the preimage procedures for dealing with uncertainty. Part of the motivation for our research is the belief that this process is in many practical cases an algorithm, and that it is an algorithm of low complexity. This belief needs to be tested.

- We have described the ideal sensor in this proposal as a physical device that supplies information on demand. This description is an oversimplification. The required information can arise in many different ways; the information need not be the direct output of a physical sensor. For instance, a robot might obtain the information it requires to make progress by performing information-gathering motions and by maintaining a history of past motions and sensor values. The research question is whether it is possible to combine history and prediction with a tight sensory feedback loop of the type that we are assuming.

  As an example, in a particular case of the peg-in-hole problem, it may be necessary for a robot to use a cartesian sensor with a Kalman filter, instead of the ideal radial sensor that falls out of the design process. As the peg approaches close to the hole, the sensor may suddenly not provide the information required to make further progress. In particular, if ever the sensing error overlaps several cones of progress, then progress may not be possible. Fortunately, the cones of progress also provide a description of the amount of information required to again make progress. Thus, one can compute the expected amount of time that the robot must simply sit and repeatedly consult its sensor. While sitting, the robot uses the Kalman filter to reduce the variance in its position estimate, until the sensor error lies wholly within one cone of progress. Similarly, once the peg overlaps the hole, the robot could use torque sensing to obtain the information specified by the cones of progress.

  The research question is whether one can efficiently automate the construction of information-gathering subtasks for robots that are not equipped with ideal sensors.

- One of the primary motivations of the proposed research is to relax the assumption that sensors are fixed and given devices. Instead, a system should design sensors to suit its available actions and its current task. An extension of this idea is to mechanically and physically modify the environment of the task, so that a given action will make progress towards the goal from a larger set of initial conditions than is first possible. The cones of progress specify the extent to which a sensor must distinguish between actions. The cones of progress thus also specify which actions could be merged by redesigning the mechanics of the task.

- In designing approximations to ideal configuration space sensors it is useful to have a small repertoire of primitive sensors, along with a calculus for combining these sensors to build more complicated sensors. We are interested in how simple sensors may be combined into more powerful sensors. A promising approach is to use a number of movable light-beam sensors that can measure distance very accurately in a small local volume. We are motivated here by the recent successes of [Canny and Goldberg 1991].

6.3 Proposed Experimental Domain

We propose to test our sensor design methodology in the domain of parts assembly. We will consider planar parts assembly and planar parts orienting. In addition we will consider three-dimensional assembly operations in which the essential character of the assembly can be represented in low dimensions, such as the cylindrical peg-in-hole and the two-pin-two-hole problems discussed above.

The types of tasks we envision are:

- Locating parts that enter the workarea of the robot via some transport mechanism, such as a feeder belt. The parts that we have in mind are:
  - Rigid planar parts, of different sizes and shapes. Some of these shapes are convex, others concave. Some contain holes for attaching springs or bolts.
  - Blocks of varying sizes.
  - Bolts of varying sizes, shapes, and thread types.
  - Nuts corresponding to the bolts.
  - Springs of varying lengths.
- Distinguishing between different parts that enter the workarea.
- Orienting and positioning the parts.
- Assembling the parts. The robot most accomplish several different kinds of assembly tasks, of the form:
  - Assemble two or more of the planar shapes, that are designed to fit together.
  - Fasten two or more parts together with a bolt.
  - Fasten two or more parts together with a spring.

The physical hardware that we would like to employ is a Zebra Zero force-controlled manipulator, a vision system, and a number of very simple sensors such as light beams, infrared proximity sensors, and contact switches. We currently have the Zebra manipulator running in the lab. We would like to purchase a vision system, the simple sensors, and better workstations on which to run our control and planning software. In particular, the Zebra may be controlled either directly from a PC or over the network from a Sun workstation. The PC does not permit us to run very extensive planning systems, while the Sun is an old 3/60 model with very little memory and a slow processor, resulting in significant paging and network delays. We would like to upgrade our hardware in order to perform efficiently the experimental part of our research.

The main objective of the experimental part of the proposed research is to test the versatility of an assembly system that can design its own sensors. We intend to write planning software to design the sensors. We envision a planner that expects as input a geometric description of the parts to be assembled, that has some knowledge of mechanics and uncertainty, and that can reason about combining primitive sensors to form more complicated sensors. Additionally, the planner will expect as input a geometric description of an assembly task. The planner will backchain from the goal of this task, in order to generate both a desired sequence of assembly operations that accomplishes the task as well as an ideal sensor that provides the information required to execute the assembly operations. Having found such a plan, the planner will then approximate the ideal sensor with a combination of the primitive sensors.

7 A History of Manipulation Methodologies

7.1 Symbolic Planning

The SHAKEY system from the late sixties (see [Nilsson 1980, 1984] for descriptions) is a splendid example of an early AI approach for planning the motions of a robot operating in a physical world. The system consisted of a mobile robot, a high-level planner based on STRIPS, and several layers of lower-level planning and control loops. The goal set forth by the SHAKEY system has driven research in robotics and AI for several decades. The goal was to develop a system that would permit specification of robot goals at the task level. The system was expected to then translate these task-level goals into appropriate commands for actually moving the robot and accomplishing the task.

Two difficulties became apparent with SHAKEY. The first is that uncertainty mattered. Someone had to implement low-level routines to deal with uncertainty. The second difficulty is that the symbolic states used to represent the robot's configuration were too simple. Once one proceeds to more complicated tasks, such as the assembly of complicated parts, these symbolic states are insufficient. Instead, it seems to be necessary to consider geometry in detail.

These two difficulties led to further work. Some of the work dealt with uncertainty, other work dealt with geometry. Later these two threads reconnected to form a general approach for dealing with uncertainty in geometrically specified environments.

7.2 Configuration Space

The work on geometry sought to understand the constraints imposed by obstacles in the world on the motions of a robot or manipulator. In a series of articles, Lozano-Pérez developed the notion of *configuration space* as a means of characterizing these constraints. See [Lozano-Pérez and Wesley 1979] and [Lozano-Pérez 1981, 1983].

The configuration space approach spawned more than a decade of work on motion planning. Some work sought to classify the complexity of the motion planning problem under the assumptions made above, and to devise and implement algorithms for planning solutions to this problem. See, for instance, [Reif 1979], [Schwartz and Sharir 1983a], [Donald 1984, 1987], [Canny 1986, 1988a, 1988b], [Khatib 1986], and [Koditschek 1987]. Ultimately the problem was classified as PSPACE-complete (see [Reif 1979] and [Canny 1988b]). Other work considered the problem of planning the motions for several cooperating robots. See, for instance, [Schwartz and Sharir 1983b], [Yap 1984], [Hopcroft, Schwartz, and Sharir 1984], [Reif and Sharir 1985], [Kant and Zucker 1986], and [Erdmann and Lozano-Pérez 1987]. Some books with excellent survey articles include [Brady et al. 1982], [Schwartz and Yap 1986], [Schwartz, Hopcroft, and Sharir 1987], [Khatib, Craig, and Lozano-Pérez 1989], and [Latombe 1990].

7.3 Guarded Moves and Compliant Motion

Early work on uncertainty sought to improve the performance of robot planning and execution systems by developing primitive operations specifically designed to overcome uncertainty. The intent was to retain the structure of high-level planning systems, while using the primitive operations to encapsulate the low-level activity needed to overcome uncertainty.

The new primitives were special feedback loops and force control strategies that coupled motion to sensing in a manner that reduced uncertainty. *Guarded moves* consisted of motions that terminated execution when some sensory condition was satisfied. Similarly, *compliant motion* strategies were developed to maintain contact with surfaces. These strategies were useful for establishing and maintaining contact with a surface even when the location, shape, or attitude of the surface was uncertain. See [Ernst 1961], [Will and Grossman 1975], [Drake 1977], [Whitney 1977], [Salisbury 1980], [Mason 1981], and [Raibert and Craig 1981], among others.

Related activity involved the construction of mechanical devices, such as the *RCC* of Draper Labs that could perform tight assembly operations without requiring precise control or sensing. The construction of such devices required a physical analysis of the task that involved modelling both geometry and friction. See [Nevins et al. 1975]. Indeed, this work motivated other work on modelling friction and on planning mechanical operations guaranteed to succeed in the presence of uncertainty (see, for instance, [Simunovic 1979], [Whitney 1982], [Erdmann 1984], [Caine 1985], and [Sturges 1988]).

7.4 Skeleton Strategies

Guarded moves and compliant motions provided local tools for overcoming uncertainty. The natural next step was to incorporate these local strategies into the global motion planners.

The first step was to relax the view of assembly strategies as fixed loops based on nominal robot and part trajectories. Early work considered parameterizing strategies in terms of quantities that could vary with particular problem instantiations. The skeleton strategies of [Lozano-Pérez 1976] and [Taylor 1976] offered a means of relating error estimates to strategy specifications in detail. [Brooks 1982] extended this approach using a symbolic algebra system. His system could be used both to provide error estimates for given operations, as well as to constrain task variables or add sensing operations in order to guarantee task success. Along a slightly different line, [Dufay and Latombe 1984] developed a system that observed execution traces of proposed plans, then modified these using inductive learning to account for uncertainty.

7.5 Preimages

In the previous research we see the first attempts to account for uncertainty. However, uncertainty is still treated merely as an irritant, capable of destroying a previously and independently developed plan. In 1983, Lozano-Pérez, Mason, and Taylor proposed a planning framework for synthesizing fine-motion strategies in the presence of uncertainty. This framework generates plans by recursively backchaining from the goal. Each backchaining step generates a collection of sets, known as *preimages*, from which entry into the goal is guaranteed, despite sensing and control uncertainty. The preimage framework directly incorporates uncertainty into the planning process. See [Mason 1984], [Erdmann 1986], [Buckley 1987], [Donald 1989], and [Canny 1989] for further work on preimages. See [Latombe

1990] for an excellent discussion of preimage work, and see [Bertsekas 1987] for related work in the stochastic domain.

7.6 Sensorless Manipulation

Open-loop strategies constitute an important class of manipulation strategies, some of which are explicitly designed to reduce uncertainty. Such sensorless strategies rely purely on the mechanics of their domain to accomplish a task, despite uncertainty in the actions and in the initial state of the system. See, for instance, [Mason 1982, 1986]. In particular, Mason looked at the problem of reducing uncertainty in the orientation of parts by pushing. Building on this work, [Brost 1988] implemented a system that could orient planar parts through a series of pushing and squeezing operations. See also [Mani and Wilson 1985], [Peshkin 1986], [Natarajan 1986], [Erdmann and Mason 1988], and [Goldberg, Mason, and Erdmann 1991].

7.7 Error Detection and Recovery

An important offspring of the LMT preimage planning methodology is Donald's work on Error Detection and Recovery [Donald 1988a, 1989, 1990]. This work deals with the related problems of representing model error and planning strategies for error detection and recovery. Donald's work moved away from the requirement that a strategy, in order to be considered a legitimate strategy, actually be guaranteed to solve a task in a fixed predetermined number of steps. This move represents an important change in perspective.

7.8 Randomization

Building on top of Donald's ideas, [Erdmann 1990] studied active randomization as a primitive strategy for accomplishing robot tasks. Randomization is one way to overcome the problem posed by Donald. This is the problem of deciding what to do if a system cannot guarantee task success with the actions and sensors that it has available. Randomization accomplishes a task probabilistically. By actively randomizing its actions, a system can blur the significance of unmodelled or uncertain parameters.

An interesting consequence of this view is that there are tasks that may be solved with no information except for a sensor that signals goal attainment. For instance, for tasks in which there is no danger of catastrophic failure and in which there are no trap states, a robot can simply perform a random walk through state space until the goal is attained.

Randomization has been used explicitly for solving automation tasks by other researchers. In the domain of mobile robots, see for instance [Arkin 1989], who injects noise into potential fields to avoid plateaus and ridges. [Barraquand and Latombe 1989] investigated a Monte-Carlo approach for escaping from local minima in potential fields. Some probabilistic work was aimed at facilitating the design process. For instance, [Boothroyd et al. 1972] considered the problem of determining the natural resting distributions of parts in a vibratory bowl feeder. [Goldberg 1990] investigated probabilistic strategies for grasping objects, and developed a framework for planning optimal orienting strategies relative to various cost functions.

7.9 Interwoven Strategies and Environments

There is an interesting germ of an important idea in the work cited above. If we look at the skeleton strategies of Taylor and Lozano-Pérez, and the plan checker of Brooks, we see an underlying design philosophy. Specifically, strategies and environments are seen as interwoven. Given a robot task there are two ways to proceed. One is to develop a strategy for accomplishing the task in the specified environment. It is in this direction that the work on uncertainty has evolved over the last few years. Another direction is to redesign the environment so that some simple off-the-shelf strategy is guaranteed to succeed. For instance, one might develop a nominal plan for accomplishing the task under the assumption of no uncertainty. In order to ensure the plan's success in an uncertain world, one introduces a new sensor not originally postulated in the environment. This second direction for dealing with uncertainty has not received as much attention in recent years. An important research question is to determine the conditions under which redesigning the environment is a possible solution. Studying the information requirements of manipulation tasks is one approach to answering this question.

7.10 A Progression of Ideas

This proposal builds on the progression of research discussed above. It seeks to develop efficient methods for planning manipulation strategies in the presence of uncertainty. We see a progression from symbolic reasoning, to planning in the presence of uncertainty, to designing sensors and environments. At the heart of all of this research is an attempt to understand the interaction of a robot with its environment and to facilitate the automatic programming of robots.

8 Statement of Impact

This research is significant on three levels:

1. First, the research will examine directly the availability, design, and use of existing simple sensors to improve the reliability and effectiveness of assembly operations.

2. Second, the research is aimed at breaking the uncertainty barrier. While we have had formal methods for dealing with uncertainty in robot operations for about a decade, the computational complexity of these methods has made it nearly impossible to implement systems for dealing with uncertainty. Fundamentally, the difficulty lies with sensing uncertainty. The proposed research intends to circumvent this computational complexity by designing sensors that provide precisely the information required by the assembly system.

3. Third, the research is of long-range significance in understanding the information requirements of manipulation systems. Such systems include factory assembly cells as well as household robots or humans. Current robots are extremely limited in functionality. Additionally, it is cumbersome to program these robots. Much trial and error testing is necessary. We simply do not have a good understanding of how task strategies can fail due to poor modelling, uncertain control, and imperfect sensing. In order to automate the design and development of systems with broad capabilities it is essential to understand how sensing interacts with mechanics.

We envision the most immediate impact of our proposed research in the following specific areas.

- **Parts Identification, Sorting, and Orienting.** The proposed research will examine motion and sensing strategies for orienting small parts. A typical example consists of identifying and orienting parts such as screws and bolts, plastic components, and small subassemblies. These parts typically arrive semi-chaotically at a robotic workcell via some transport system. Existing systems for orienting such parts include bowl feeders, specially designed nests (such as Sony's APOS), and active vision-manipulation strategies. We would like to understand the design of easily reconfigurable special-purpose parts handlers that are composed of simple sensors and simple mechanical gates.

- **Assembly.** Parts identification and orienting is typically a prelude to an assembly task. The proposed research should be of interest to robot programmers. In particular, given an assembly task, the proposed research will suggest a motion strategy along with a set of sensor configurations for accomplishing the task quickly and successfully. Typical tasks that we will consider are insertions, screwing operations, and spring loading.

- **Task-Level Planning.** Studying the information requirements of robot tasks will lead to better ways for programming robots. The ultimate goal is to develop task-level specifications for programming robots. To date, the notion of a task is not even well-defined, given the problems generated by uncertainty. Exploring the information flow between different parts of a task is crucial to developing task-level systems. The design of action-based special purpose sensors is particularly important, as it tells us what information is required for a given task. Inverting this relationship circumscribes a task, that is, it provides a language for talking about tasks in high-level terms. For instance, rather than talk about the SCREW THE BOLT INTO THE HOLE task, one might refer to the task by its sensing and motion requirements: "This is the task that requires a rotationally varying sensor, moving in conjunction with a rotational motion of the arm." Both descriptions are reasonably high-level, yet the first conveys no information about the task. The proposed research will help in developing functional task-level descriptions.

- **Operation in Partially Unknown Environments.** The proposed research examines the design and construction of sensors for reducing uncertainty. Environments consisting of known objects in unknown locations or consisting of unknown objects of fixed type are conducive to this approach. For instance, we imagine designing sensors useful for locating and sorting recyclable parts such as those found in consumer trash or automobile recycling plants. Other applications include sorting and locating objects in hazardous waste dumps.

9 Acknowledgments

Many thanks to Tamara Abell, Bruce Donald, Tomás Lozano-Pérez, Matt Mason, and Nina Zumel for their insights, discussions, and comments.

10 Bibliography

[Arkin 1989] Arkin, R. C. 1989. Motor Schema-Based Mobile Robot Navigation. *International Journal of Robotics Research.* 8(4): 92–112.

[Barraquand and Latombe 1989] Barraquand, J., and Latombe, J.-C. 1989. Robot Motion Planning: A Distributed Representation Approach. Report No. STAN-CS-89-1257. Stanford University, Department of Computer Science.

[Bertsekas 1987] Bertsekas, D. P. 1987. *Dynamic Programming: Deterministic and Stochastic Models.* Englewood Cliffs, N.J.: Prentice-Hall.

[Boothroyd et al. 1972] Boothroyd, G., Redford, A. H., Poli, C., and Murch, L. E. 1972. Statistical Distributions of Natural Resting Aspects of Parts for Automatic Handling. *Manufacturing Engineering Transactions.* 1:93–105.

[Brady et al. 1982] Brady, M., Hollerbach, J. M., Johnson, T. L., Lozano-Pérez, T., and Mason, M. T. (eds). 1982. *Robot Motion: Planning and Control.* Cambridge, Mass.: MIT Press.

[Brooks 1982] Brooks, R. A. 1982. Symbolic Error Analysis and Robot Planning. *International Journal of Robotics Research.* 1(4):29–68.

[Brost 1988] Brost, R. C. 1988. Automatic Grasp Planning in the Presence of Uncertainty. *International Journal of Robotics Research.* 7(1):3–17.

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

[Caine 1985] Caine, M. E. 1985. Chamferless Assembly of Rectangular Parts in Two and Three Dimensions. M.S. thesis, Massachusetts Institute of Technology, Department of Mechanical Engineering.

[Canny 1986] Canny, J. F. 1986. Collision Detection for Moving Polyhedra. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* **PAMI-8**(2):200–209.

[Canny 1988a] Canny, J. F. 1988. *The Complexity of Robot Motion Planning.* Cambridge, Mass.: MIT Press.

[Canny 1988b] Canny, J. F. 1988. Some Algebraic and Geometric Computations in PSPACE. *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, pp. 460–467.

[Canny 1989] Canny, J. F. 1989. On Computability of Fine Motion Plans. *Proceedings of the 1989 IEEE International Conference on Robotics and Automation*, pp. 177–182.

[Canny and Goldberg 1991] Canny, J. F., and Goldberg, K. Y. 1991. "RISC-Robotics," talk given at the Robotics Seminar at CMU, Oct 11, 1991.

[Canny and Reif 1987] Canny, J. F., and Reif, J. H. 1987. New Lower-Bound Techniques for Robot Motion Planning Problems. *Proceedings, 28th Symposium on the Foundations of Computer Science.*

[Donald 1984] Donald, B. R. 1984. Motion Planning with Six Degrees of Freedom. AI-TR-791. S.M. thesis. Massachusetts Institute of Technology, Artificial Intelligence Laboratory.

[Donald 1987] Donald, B. R. 1987. A Search Algorithm for Motion Planning with Six Degrees of Freedom. *Artificial Intelligence.* **31**(3):295–353.

[Donald 1988a] Donald, B. R. 1988. Robot Motion Planning with Uncertainty. *Artificial Intelligence.* **37**(1-3):223–271.

[Donald 1988b] Donald, B. R. 1988. The Complexity of Planar Compliant Motion Planning with Uncertainty. *Proceedings, ACM Symposium on Computational Geometry.*

[Donald 1989] Donald, B. R. 1989. *Error Detection and Recovery in Robotics.* Lecture Notes in Computer Science, No. 336. Berlin: Springer-Verlag.

[Donald 1990] Donald, B. R. 1990. Planning Multi-Step Error Detection and Recovery Strategies. *International Journal of Robotics Research.* **9**(1):3–60.

[Drake 1977] Drake, S. H. 1977. Using Compliance in Lieu of Sensory Feedback for Automatic Assembly. Sc.D. thesis. Massachusetts Institute of Technology, Department of Mechanical Engineering.

[Dufay and Latombe 1984] Dufay, B., and Latombe, J. 1984. An Approach to Automatic Robot Programming Based on Inductive Learning. *Robotics Research: The First International Symposium*, eds. M. Brady, and R. Paul. Cambridge, Mass.: MIT Press, pp. 97–115.

[Erdmann 1984] Erdmann, M. A. 1984. On Motion Planning with Uncertainty. AI-TR-810. S.M. thesis. Massachusetts Institute of Technology, Artificial Intelligence Laboratory.

[Erdmann 1986] Erdmann, M. A. 1986. Using Backprojections for Fine Motion Planning with Uncertainty. *International Journal of Robotics Research.* **5**(1):19–45.

[Erdmann 1990] Erdmann, M. A. 1990. On Probabilistic Strategies for Robot Tasks. AI-TR-1155. Ph.D. thesis. Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science.

[Erdmann and Lozano-Pérez 1987] Erdmann, M., and Lozano-Pérez, T. 1987. On Multiple Moving Objects. *Algorithmica.* **2**(4):477–521.

[Erdmann and Mason 1988] Erdmann, M., and Mason, M. T. 1988. An Exploration of Sensorless Manipulation. *IEEE Journal of Robotics and Automation.* **4**(4):369–379.

[Ernst 1961] Ernst, H. A. 1961. MH-1, A Computer-Operated Mechanical Hand. Sc.D. thesis. Massachusetts Institute of Technology, Department of Electrical Engineering.

[Goldberg 1990] Goldberg, K. 1990. Stochastic Plans for Robotic Manipulation. Ph.D. Thesis. Carnegie Mellon University, School of Computer Science.

[Goldberg, Mason, and Erdmann 1991] Goldberg, K., Mason, M., and Erdmann, M.

1991. Generating Stochastic Plans for a Programmable Parts Feeder. *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, pp. 352-359.

[Hopcroft, Schwartz, and Sharir 1984] 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.

[Kant and Zucker 1986] Kant, K., and Zucker, S. W. 1986. Toward Efficient Trajectory Planning: The Path-Velocity Decomposition. *International Journal of Robotics Research.* **5**(3):72–89.

[Khatib 1986] Khatib, O. 1986. Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. *International Journal of Robotics Research.* **5**(1):90–98.

[Khatib, Craig, and Lozano-Pérez 1989] Khatib, O., Craig, J., and Lozano-Pérez, T. (eds.) 1989. *The Robotics Review 1.* Cambridge, Mass.: MIT Press.

[Koditschek 1987] Koditschek, D. E. 1987. Exact Robot Navigation by Means of Potential Functions: Some Topological Considerations. *Proceedings of the 1987 IEEE International Conference on Robotics and Automation*, Raleigh, North Carolina, pp. 1–6.

[Latombe 1990] Latombe, J.-C. 1990. *Robot Motion Planning.* Boston: Kluwer Academic Publishers.

[Lozano-Pérez 1976] Lozano-Pérez, T. 1976. The Design of a Mechanical Assembly System. AI-TR-397. S.M. thesis. Massachusetts Institute of Technology, Artificial Intelligence Laboratory. Reprinted in part in Winston, P. H., and Brown, R. H., eds. 1979. *Artificial Intelligence: An MIT Perspective.* Cambridge, Mass.: MIT Press.

[Lozano-Pérez 1981] Lozano-Pérez, T. 1981. Automatic Planning of Manipulator Transfer Movements. *IEEE Transactions on Systems, Man, and Cybernetics.* **SMC–11**(10):681–698. Reprinted in Brady, M., et al., eds. 1982. *Robot Motion.* Cambridge, Mass.: MIT Press.

[Lozano-Pérez 1983] Lozano-Pérez, T. 1983. Spatial Planning: A Configuration Space Approach. *IEEE Transactions on Computers.* **C-32**(2):108–120.

[Lozano-Pérez, Mason, and Taylor 1984] Lozano-Pérez, T., Mason, M. T., and Taylor, R. H. 1984. Automatic Synthesis of Fine-Motion Strategies for Robots. *International Journal of Robotics Research.* **3**(1):3–24.

[Lozano-Pérez and Wesley 1979] Lozano-Pérez, T., and Wesley, M. 1979. An Algorithm for Planning Collision-Free Paths Among Polyhedral Obstacles. *Communications of the ACM.* **22**(10):560–570.

[Mani and Wilson 1985] Mani, M., and Wilson, W. 1985. A Programmable Orienting System for Flat Parts. *Proc., NAMRI XIII.*

[Mason 1981] Mason, M. T. 1981. Compliance and Force Control for Computer Controlled Manipulators. *IEEE Transactions on Systems, Man, and Cybernetics.* **SMC-11**(6):418–432. Reprinted in Brady, M., et al., eds. 1982. *Robot Motion.* Cambridge, Mass.: MIT Press.

[Mason 1982] Mason, M. T. 1982. Manipulator Grasping and Pushing Operations. AI-TR-690. Ph.D. thesis. Massachusetts Institute of Technology, Artificial Intelligence Laboratory.

[Mason 1984] Mason, M. T. 1984. Automatic Planning of Fine-Motions: Correctness and Completeness. *Proceedings of the 1984 IEEE International Conference on Robotics and Automation*, Atlanta, Georgia, pp. 492–503.

[Mason 1986] Mason, M. T. 1986. Mechanics and Planning of Manipulator Pushing Operations. *International Journal of Robotics Research.* **5**(3):53–71.

[Natarajan 1986] Natarajan, B. K. 1986. 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 1988] Natarajan, B. K. 1988. The Complexity of Fine Motion Planning. *International Journal of Robotics Research.* **7**(2):36–42.

[Nevins et al. 1975] Nevins, J., Whitney, D., Drake, S., Killoran, D., Lynch, M., Seltzer, D., Simunovic, S., Spencer, R. M., Watson, P., and Woodin, A. 1975. Exploratory Research in Industrial Modular Assembly. Report R-921. C.S. Draper Laboratory, Cambridge, Mass.

[Nilsson 1980] Nilsson, N. 1980. *Principles of Artificial Intelligence.* Tioga Publishing, California.

[Nilsson 1984] Nilsson, N., ed. 1984. Shakey the Robot. Stanford Research Institute, Technical Note 323.

[Papadimitriou 1985] Papadimitriou, C. H. 1985. Games against Nature. *Journal of Computer and System Sciences.* **31**:288-301.

[Papadimitriou and Tsitsiklis 1987] Papadimitriou, C. H., and Tsitsiklis, J. N. 1987. The Complexity of Markov Decision Processes. *Mathematics of Operations Research.* **12**(3):441–450.

[Peshkin 1986] Peshkin, M. A. 1986. Planning Robotic Manipulation Strategies for Sliding Objects. Ph.D. Thesis. Carnegie Mellon University, Physics Department.

[Raibert and Craig 1981] Raibert, M. H., and Craig, J. J. 1981. Hybrid Position/Force Control of Manipulators. *Journal of Dynamic Systems, Measurement, and Control.* **102**:126–133. Reprinted in Brady, M., et al., eds. 1982. *Robot Motion.* Cambridge, Mass.: MIT Press.

[Reif 1979] Reif, J. H. 1979. The Complexity of the Mover's Problem and Generalizations. *Proceedings, 20th Symposium on the Foundations of Computer Science.*

[Reif and Sharir 1985] Reif, J., and Sharir, M. 1985. Motion Planning in the Presence of Moving Obstacles. *Proceedings, 26th IEEE Symposium on the Foundations of Computer Science*, pp. 144–154.

[Salisbury 1980] Salisbury, J. K. 1980. Active Stiffness Control of a Manipulator in Cartesian Coordinates. Paper delivered at *IEEE Conference on Decision and Control*, Albuquerque, New Mexico, December, 1980. Reprinted in Mason, M. T. and Salisbury, J. K. 1985. *Robot Hands and the Mechanics of Manipulation.* Cambridge, Mass.: MIT Press, pp. 95–108.

[Schwartz, Hopcroft, and Sharir 1987] Schwartz, J., Hopcroft, J., and Sharir, M. 1987. *Planning, Geometry, and Complexity of Robot Motion.* New Jersey: Ablex Publishing.

[Schwartz and Sharir 1983a] Schwartz, J. T., and Sharir, M. 1983. On the Piano Movers' Problem: II. General Techniques for Computing Topological Properties of Real Algebraic Manifolds. *Advances in Applied Mathematics.* **4**:298–351.

[Schwartz and Sharir 1983b] Schwartz, J. T., and Sharir, M. 1983. On the Piano Movers' Problem: III. Coordinating the Motion of Several Independent Bodies: The Special Case of Circular Bodies Amidst Polygonal Barriers. *International Journal of Robotics Research.* **2**(3):46–75.

[Schwartz and Yap 1986] Schwartz, J., and Yap, C. 1986. *Advances in Robotics.* Hillside, New Jersey: Lawrence Erlbaum Associates.

[Simunovic 1979] Simunovic, S. N. 1979. An Information Approach to Parts Mating. Sc.D. thesis. Massachusetts Institute of Technology, Department of Mechanical Engineering.

[Sturges 1988] Sturges, R. H., Jr. 1988. A Three-Dimensional Assembly Task Quantification with Application to Machine Dexterity. *International Journal of Robotics Research.* **7**(4):34–78.

[Taylor 1976] Taylor, R. H. 1976. A Synthesis of Manipulator Control Programs from Task-Level Specifications. AIM–282. Ph.D. thesis. Stanford University, Artificial Intelligence Laboratory.

[Whitney 1977] Whitney, D. E. 1977. Force Feedback Control of Manipulator Fine Motions. *Journal of Dynamic Systems, Measurement, and Control.* **98**:91–97.

[Whitney 1982] Whitney, D. E. 1982. Quasi-Static Assembly of Compliantly Supported Rigid Parts. *Journal of Dynamic Systems, Measurement, and Control.* 104:65–77. Reprinted in Brady, M., et al., eds. 1982. *Robot Motion.* Cambridge, Mass.: MIT Press.

[Will and Grossman 1975] Will, P. M., and Grossman, D. D. 1975. An Experimental System for Computer Controlled Mechanical Assembly. *IEEE Transactions on Computers.* **C-24**(9):879–888.

[Yap 1984] Yap, C. K. 1984. Coordinating the Motion of Several Discs. Technical Report No. 105. New York University, Computer Science Department, Courant Institute of Mathematical Sciences.