An Exploration of Sensorless Manipulation

MICHAEL A. ERDMANN AND MATTHEW T. MASON

Abstract—**An autonomous robotic manipulator can reduce uncertainty in the locations of objects in either of two ways: by sensing, or by motion strategies. This paper explores the use of motion strategies to eliminate uncertainty, without the use of sensors. The approach is demonstrated within the context of a simple method to orient planar objects. A randomly oriented object is dropped into a tray. When the tray is tilted, the object can slide into walls, along walls, and into corners, sometimes with the effect of reducing the number of possible orientations. For some objects a sequence of tilting operations exists that leaves the object's orientation completely determined. This paper describes an automatic planner that constructs such a tilting program, using a simple model of the mechanics of sliding. The planner has been implemented, the resulting programs have been executed using a tray attached to an industrial manipulator, and sometimes the programs work. The paper also explores the issue of sensorless manipulation, tray tilting in particular, within the context of a formal framework first described by Lozano-Pérez, Mason, and Taylor [5]. It is observed that sensorless motion strategies perform conditional actions using mechanical decisions in place of environmental inquiries.**

I. Introduction

ROBOTS MUST successfully plan and execute tasks in the presence of uncertainty. Uncertainty constitutes an inability to know precisely the relative locations of the objects in a task. Uncertainty arises from model error, control error, and sensor error. Model error produces uncertainty throughout the planning and execution phases of a task. Control error introduces uncertainty as actions are executed. Sensor error limits the certainty with which an environment can become known through inspection.

There are two methods for overcoming uncertainty: sensory operations and motion strategies. Motion strategies use the mechanics of the task to reduce uncertainty in the relative locations of two or more objects. For example, consider the problem of placing a book on a table. A sensor-based strategy is to move the book down to the table, while watching proximity and force sensors to detect contact. A motion strategy is to move the book within a reasonable height above the table, then simply drop the book. The book is guaranteed to wind up on the table. The motion strategy is, in this case, simpler and faster than the sensing strategy.

The encompassing goal of this research is to develop automatic methods for analyzing and using the mechanics of the task to manipulate objects without the aid of sensing. Of interest are both methods for automatically solving manipulation problems in fixed environments, as well as methods for automatically designing environments conducive to accomplishing particular tasks. This paper reports on an initial investigation.

The first part of the paper considers orienting objects in a simple domain, using only the mechanics of the domain. The study of this domain offers some general insight into the problem of sensorless manipulation. An automatic planner is developed. The planner consists of one phase that determines the possible motion transitions, and a second phase that searches a space whose states reflect uncertainty at execution time.

The second half of the paper considers the relationship of sensorless manipulation to sensor-based manipulation. The main observation is that both sensorless motion strategies and sensor-based operations perform conditional actions. In both cases, the effect of a conditional action is determined by the actual state of the system at execution time. However, whereas a sensor-based system selects an action based on explicit sensor readings of the current state, a sensorless system lets the current state select the effect of an action mechanically.

A. An Example

The example we have chosen to illustrate the issues of sensorless manipulation is a system for orienting planar parts. The system consists of a tray onto which randomly oriented parts may be dropped. The tray may be tilted, causing the part to slide and rotate while making contact with the tray walls and corners. The objective is to construct a sequence of such tilting operations that uniquely orients the part.

The tray-tilting domain offers a simple setting in which to study sensorless manipulation, while still retaining the basic ingredients of many sensorless problems. In particular, the mechanics of the domain are complicated enough to require explicit analysis in order to synthesize predictable motions. Friction is important; part geometry is relevant. Most importantly, the tray-tilting domain forces us to consider general ways of representing and reducing uncertainty. The structure developed for the planner carries over to other domains, and provides a first basis for studying the relationship between sensorless and sensor-based manipulation.

Fig. 1 shows a tray-tilting operation, and one possible resulting displacement of an Allen wrench in the tray. The operation consists of starting with the tray horizontal, tilting the tray so that the arrow indicates the direction of steepest ascent, then lowering the tray back to horizontal. If we neglect inertial forces, this is equivalent to leaving the tray still while

Manuscript received August 6, 1986; revised July 16, 1987. This research was supported under Grant from the System Development Foundation.

M. A. Erdmann was with Carnegie-Mellon University, Pittsburgh, PA, on leave from the Massachusetts Institute of Technology. He is with the Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139.

M. T. Mason is with the Department of Computer Science and the Robotics Institute, Carnegie-Mellon University, Schenley Park, Pittsburgh, PA 15213.

IEEE Log Number 8821197.

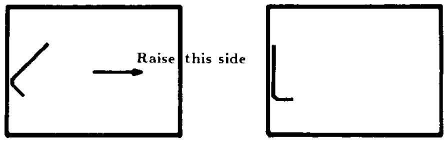

Fig. 1. Raising the right edge of the tray causes the Allen wrench to rotate to a new stable orientation.

changing the direction of the gravity vector. In any case, the arrow opposes the component of gravity tangential to the plane of the tray, once the tray is tilted. This gravitational force acts through the center of mass of the wrench, causing the wrench to rotate as shown. Throughout the paper we will refer to the direction of steepest ascent as the *azimuth*, and the angle of slope as the *elevation*.

Fig. 2 is a pictorial listing of a tray-tilting program that orients the Allen wrench. Starting from a completely arbitrary orientation, a completely determined orientation (and position) is obtained in nine steps. Each step is shown by a range of suitable azimuths, with the wrench drawn in every possible resulting orientation.

The program shown in Fig. 2 was produced automatically, based on a simple model of the mechanics of tray-tilting operations. The assumptions are as follows:

- Planar motion occurs.
- Inertial forces and impact forces are dominated by frictional forces.
- Frictional forces conform to Coulomb's law. The coefficient of friction is identical for both static and dynamic situations, and is constant both in space and time.
- The frictional forces occurring between the wrench and the bottom of the tray are neglected, except that the elevation is chosen large enough that sliding occurs.

The planner consists of two phases, the *contact* analysis phase and the *search* phase. Relying on the assumptions listed above, the contact analysis phase determines how a given azimuth affects the configuration of the wrench. The search phase uses simple forward-chaining to construct a sequence of tilting operations.

There is one final wrinkle to Allen wrench orienting. If the wrench is dropped into the tray, one must consider that the wrench may fall in either a left-handed or a right-handed orientation. No planar operation can switch from one to the other. Faced with this problem, the planner constructs a sixteen-step plan that reduces the number of possible orientations to two—one for each reflection.

B. Motivational Issues

Sensorless manipulation is an important aspect of manipulation in general. Many manipulation tasks require operations that are primarily sensorless, relying heavily on task mechanics in order to assure success. This section mentions a few of these tasks, along with some issues that further motivate the study of sensorless manipulation.

1) Some Sensorless Systems: Examples of systems employing sensorless manipulation include bowl feeders, assembly line feeders, coin sorters, and a variety of other mechanical devices. All these systems harness the mechanics of the task to orient and position objects without the aid of sensing. Bowl feeders move objects to some destination, forcing them past a gauntlet of gates and protrusions that reorient the objects. Objects not properly oriented pass through gates that mechanically recognize the orienting failure and throw the objects from the feeder back to their starting locations. Coin sorters similarly use the sizes and weights of coins as mechanical states in a decision procedure consisting of tracks and holes.

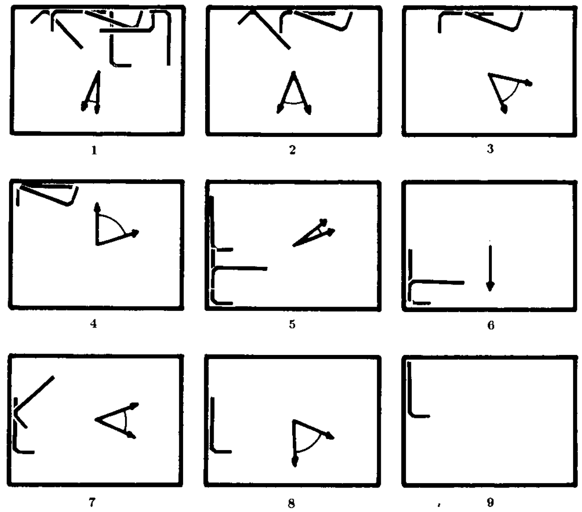

Fig. 2. Beginning at the upper left and moving from left to right, we can trace an automatically generated program that orients the wrench. Each frame shows the set of possible wrench contacts, and the operation to be applied. Each operation is represented by an interval of azimuths. The azimuth arrows indicate the tray's direction of steepest ascent; gravity acts in the opposite direction.

2) Sensorless Motions in a Sensor-Based Plan: Sensorless strategies are important as well in the context of sensor-based manipulation. Generally, sensorless manipulation plays a role in situations where sensing may not be readily performed or is inappropriate, where instead the task mechanics must be used to ensure success. Examples of tasks in which the mechanics are important include pushing, hitting, grasping, dropping, aligning, and throwing. For instance, during a grasping operation the precise forces acting and motions resulting may lie below the resolution of the available sensors. In particular, the relative positions of the fingers and the object may be indeterminate. Nonetheless, in many cases the remaining uncertainty may be eliminated with sensorless motion strategies. The mechanical interactions between the object and the fingers may be harnessed to aid the grasp (see [1] and [10]). For example, by squeezing or pushing with the fingers, the object being grasped can be forced to align itself with one of the fingers, thereby reducing uncertainty in orientation. Recognizing that such an alignment is possible and predicting how it will occur requires an understanding of the task mechanics.

3) Relation of Sensorless and Sensor-Based Strategies:

One source of potential confusion is to view sensorless manipulation solely in terms of program efficiency, especially in the context of tasks for which there exist both sensorless and sensor-based solutions. Previously, in the book example, we saw a sensorless strategy that was more efficient and required less knowledge than a corresponding sensing strategy. In general, this need not be the case. Nor should it be. In fact, precisely because sensorless manipulation must consider strategies that succeed independent of actual state at execution time, one would expect these strategies to include at times seemingly redundant or superfluous actions.

In general, one should view the uncertainty-reducing property of sensorless motions as just another mode of gathering information. Whether a sensorless strategy is superior, inferior, or complementary to a sensor-based strategy depends on the nature of the task. For complex tasks, a sensor-based system would employ a sensorless strategy as one step in an overall plan.

4) Feeders: One area in which the apparent redundancy of sensorless strategies is appropriate is the design and use of feeders. Feeders are high-volume systems in which all objects essentially move through the same set of stages. Depending upon an object's configuration upon entering a given stage, the object may or may not be affected by that stage. Once the feeder pipeline is full, the fact that a stage may sometimes perform no apparent action is irrelevant. The system as a whole is spewing forth objects at a constant rate. This uniformity of action application permits a simple and economical hardware design. In particular, the outputs of individual stages are well-defined configurations. Thus a planner can use the input–output specifications of various stages to chain together sequences of stages to construct feeders for particular applications.

The study of sensorless manipulation is important for determining what types of feeders and orienting devices are possible, that is, for deciding whether a desired set of operations is mechanically feasible. In particular, the construction of the individual stages in a feeder, and the efficient composition of these stages, is facilitated by an understanding of sensorless manipulation.

5) Scope of Sensorless Manipulation: The earlier interplay of mechanics and sensing raises an interesting theoretical question: What tasks are solvable using only the task mechanics, and what tasks require sensing? An answer to this question would greatly aid our understanding of manipulation. The question is interesting from a purely theoretical standpoint. Additionally, characterizations of solvable tasks in terms of sensing and mechanical requirements would allow us to allocate sensors and design environments in efficient and useful ways. We will return to a discussion of scope in the second half of the paper, focusing on the relationship between sensorless and sensor-based manipulation in terms of the kinds of decisions each approach performs.

C. Previous Work

Sensorless robotic manipulation has been addressed in the past. Mason [9] discusses the concept in general, and describes a number of examples. Some recent research includes the use of pushing and squeezing to eliminate uncertainty while grasping objects [1], [7], [10], [12] and to orient parts [6]. There is also some recent theoretical work on the complexity of sensorless manipulation [11].

This paper is directly inspired by an example described by Grossman and Blasgen [4]. Grossman and Blasgen considered dropping an object into a tray consisting of a planar surface surrounded by bounding walls. The plane of the tray had been tilted so that an object dropped onto the plane would ultimately slide into a trihedral corner formed by the plane and two of the walls. The set of final orientations that the object might achieve in the corner could be predicted from the shape of the object. By choosing the tilt angles appropriately, this set was guaranteed to be of finite size. Once in the corner, the object's actual orientation could be determined by a sequence of sensing and probing operations.

The key idea underlying the Grossman and Blasgen scheme is the realization that many objects, in particular polyhedral objects, have only a finite number of stable resting configurations under the influence of gravity. Thus simply dropping an object onto a plane reduces the orientation uncertainty from an infinity of possibilities to a small and finite number. Forcing the object to slide along a wall further constrains the set of orientations, while forcing the object into a corner constrains the object's position as well.

The comparative roles of sensing and action were also considered in a formal framework by Lozano-Pérez, Mason, and Taylor [5]. The particular model employed assumed the presence of position and force sensors, but the planner described would use motion strategies when an advantage could be obtained. Mason [8] described a variant planner, and demonstrated that the planner was correct and complete, that is, if a plan for a given problem exists, the planner converges on a correct plan. Erdmann [2] showed that Mason's variant planner is not generally computable; implemented a less powerful, but computable, version; and demonstrated the plans in simulation.

II. A Tray-Tilting Planner

This section describes the planner in more detail, starting with the search phase. Following the description of the planner is a summary of our experience with the physical implementation, and some of the limitations we discovered.

First, it is necessary to discuss the planner's model of the tray-tilting domain. The planner does not model the global geometry of the tray. It models the tray as a string of walls, joined together at square corners. Each wall also has a facing, parallel, wall. In practice, this structure was implemented by a rectangular tray, but the planner does not know that the facing wall and the wall reached by two left turns are the same wall. It would not care anyway, since all walls, and all corners, are indistinguishable, as far as the planner is concerned.

The planner expects as input a geometric description of the part to be oriented, in the form of a convex polygon. Notice that for any polygonal object it is sufficient to consider the convex hull of that object. This is because any feasible contact between the object and the tray walls can consist only of vertices and edges on the convex hull of the object. The planner also expects as input the ratio of moment of inertia to mass of the part. This information is needed to determine the relative rate of rotation to rate of sliding during certain part motions.

The planner models four different types of contact, shown in Fig. 3. (The term "edge" refers to an edge of the convex hull of the part.)

- (M, i) Object edge i is against a wall, roughly in the middle.
- (L, i) Edge i is against a wall, and the object is in the left corner.
- (R, i) Edge i is against a wall, and the object is in the right corner.
- (J, i) Object edge i is facing away from the wall, with the object in the middle. The vertex or edge opposite edge i is in contact with the wall. This contact occurs when the object is allowed to slide away from contact state (M, i) to make contact with the facing wall, an operation that we call a *jump*.

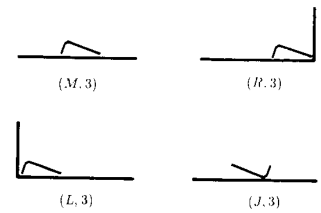

Fig. 3. The four contact types are illustrated using the wrench.

A. The Search Phase

Let us consider the search phase first, as this will motivate the contact analysis phase. The search constructs a graph, where each node of the graph is a set of possible contacts, and each edge is an interval of azimuths. For example, suppose node A consists of three contacts, that is, $A = \{C_1, C_2, C_3\}$. Suppose further that every azimuth between 45° and 90° causes contact C_1 to move to contact C_4, causes contact C_2 to move to either contact C_3 or contact C_4, and leaves contact C_3 unchanged. Let B be the node given by $B = \{C_3, C_4\}$. Then the graph includes an edge [45°, 90°] directed from A to B. The mapping on contacts defined by azimuths is determined during the contact analysis phase. The nodes and edges of the graph, however, are constructed as needed during the search.

A node in the graph thus reflects the uncertainty with which the configuration of the part being oriented is known at execution time. An edge in the graph describes a transition from one state of uncertainty to another state of uncertainty as a result of tilting the tray. The edges between nodes are constructed from knowledge of how individual contact states behave as a result of tilting the tray. This knowledge is supplied by the contact analysis phase.

The initial node in the search is the set of contacts that might result from dropping the object to be oriented onto the middle of the tray, then tilting the tray so that the object slides to the middle of a wall. Hence the initial node might consist of all possible edge-wall contacts, for a particular wall. In the wrench example, this means that the initial node consists of six possible contacts if the reflection state of the wrench is known, and twelve possible contacts if the reflection state is not known.

The objective of the search phase is to determine a sequence of edges leading from the initial node to a node of minimal cardinality. Other factors, such as the length of the sequence, may also be considered. The result is a sequence of azimuths at which the tray should be tilted. The final node is the set of possible contacts at the end of the tilting program. If there is only one such contact, then the object can be oriented unambiguously. If there is more than one such contact, then some other operation must distinguish between the contacts. A different sensorless strategy, such as executing a grasp at each possible location, could be used, or a sensory operation might be used to discriminate among the remaining possibilities.

The search phase was implemented as a breadth-first search. Such a search guarantees that the resulting path is the shortest path to the final node. In other words, the tilting program contains the fewest number of steps. Some pruning of nodes that need to be expanded is possible. For instance, it is never necessary to pursue edges from a node that is a superset of an already visited node. This is because any strategy that could successfully orient a part given the uncertainty of the larger node, would also successfully orient the part given the reduced uncertainty of the smaller node. For the Allen wrench example, our search trees had depths of about nine levels, with between one and five nodes expanded per level.

B. The Contact Analysis Phase

The objective of the contact analysis phase is to determine the mapping from contacts to contacts, as required for the search phase. Consider, for example, a block in contact with a horizontal wall, as shown in Fig. 4. Assume that the coefficient of friction is μ, and let $\alpha = \tan^{-1} \mu$. Then the space of azimuths may be divided into four regions, as shown in Fig. 5. The top central region in Fig. 5 corresponds to the friction cone. Any orientations in this range result in force applications that lie inside the friction cone, hence cause no object motions. Directions in the top left region cause the object to slide towards the right, while tray orientations in the top right region cause the object to slide to the left. All orientations in the lower region cause the object to break contact with the tray wall, and "jump" to the facing wall.

The contact analysis phase considers the various edge-wall, vertex-wall, and edge-wall-corner contacts mentioned previously. For each, it partitions the circle of azimuths, similar to the partition in the block example shown in Fig. 5. Each region is labeled with the contacts that would result from tilting the tray at an angle contained in the region. This information is passed to the search phase.

In general, the classical friction cone is not quite enough, for two reasons: first, one must consider rotating objects; and second, one must consider multiple contacts. Both of these difficulties are addressed by using a generalized friction cone derived by Erdmann [2]. Using this construction, the analysis is ultimately no more complicated than that used in the block

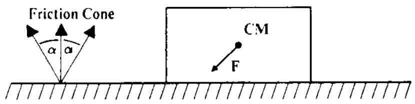

Fig. 4. A block sliding on a horizontal surface. Forces are applied at the center of mass.

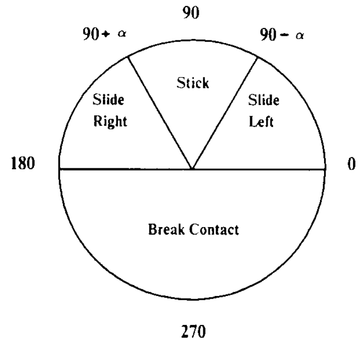

Fig. 5. A partition of azimuths into regions representing different motions of the block of Fig. 4.

example previously. For example, for one-point contact, it is again possible to break up the range of azimuths into four regions. One region corresponds to azimuths that cause the object to break contact with the wall. Another region corresponds to azimuths that cause pure rotations about the contact point. The remaining two regions correspond to azimuths that cause sliding motions at the point of contact. Thus it is possible to predict the instantaneous behavior of the object given any particular tilting motion. From this information it is possible to surmise the extended behavior of the object, and predict the contact transitions required by the search phase.

C. The Generalized Friction Cone

This section sketches briefly the form of the generalized friction cone that we used in the contact analysis phase. Consider a planar object that is permitted to rotate as well as translate. The objects considered by the tray tilter are modeled in this manner. In order to describe and predict the behavior of such an object it is necessary to consider all the forces and torques acting on the object. In general, any force acting on the object will also induce a torque about the object's center of mass. In particular, any contact force, such as a frictional force, will induce a (possibly zero) torque about the center of mass. The purpose of the generalized friction cone is to represent both the forces and induced torques arising from contact forces.

The classical friction cone is a subset of force space, that represents the range of reaction forces possible due to contact with some object. Similarly, the generalized friction cone is a subset of generalized force space that represents the range of reaction forces and torques possible as a result of contact with some object. For single-point contact, the generalized friction cone is a two-dimensional cone in the three-dimensional force–torque space. To see this, consider the classical friction cone (see Fig. 4), and imagine it to be embedded in three-space. To each force in the classical friction cone add a vector perpendicular to the plane of the cone, that represents the torque induced about the center of mass by that force. This vector depends on the particular force considered, but it varies linearly with the forces in the classical friction cone. The result of this transformation is that the classical friction cone is turned and tilted out of the plane of the paper. The result is the generalized friction cone.

A nice property of the generalized friction cone is that it may be used to predict the behavior of an object in single-point contact, which is subject to an applied force and torque, such as a gravitational or pushing force. Specifically, the applied force and torque are viewed as a vector in force–torque space. This vector is projected in a particular manner onto the generalized friction cone to determine the reaction force and torque resulting from the point of contact. Knowing this, the net motion of the object may be determined. In essence, the generalized friction cone provides a geometric method for solving Newton's equations.

Finally, for multiple-point contact, the generalized friction cone is the vector sum of all the single-point contact friction cones. Again, this representation describes the range of possible reaction forces and torques. Again, the friction cone may be used to determine the motion of an object in contact which is subject to applied forces and torques.

D. Preliminary Experimental Results and Limitations

The tray-tilting method was physically implemented by attaching trays to a PUMA 560 industrial manipulator. The trays were lids of cardboard boxes, or in one case, a plastic kitchen utensil box. The tray was tilted using the wrist motions of the manipulator. The objects we tried were two different Allen wrenches, and a binder clip (Fig. 6).

For each object, our planner generated a program in the form of a sequence of tilt angles. We ran these programs several dozen times. While we were not always able to satisfy all the assumptions listed in Section I-A, to the extent that the assumptions were satisfied, the programs were successful. All in all, the programs succeeded about half the time; perhaps more. Even of the failures, a surprising number actually ended with the object in the correct orientation, having gotten there by an unplanned path.

We believe that the failures can be attributed to violations of the basic mechanical assumptions. These assumptions are fairly restrictive. We will mention below the ways in which the assumptions could be violated, and suggest possible remedies. Any practical system would probably want to consider these suggestions. A good approach would be to extend the scope of the planner in order to effectively relax the assumptions.

It is perhaps misleading to view tray tilting as the basis for a practical orienting system. While one can imagine a home robot occasionally resorting to orienting objects situated on trays, the domain is too simple for general-purpose use. What the tray-tilting domain does show, is that sensorless manipulation is an important ingredient of manipulation in general. Even with such simple features as walls and right-angled

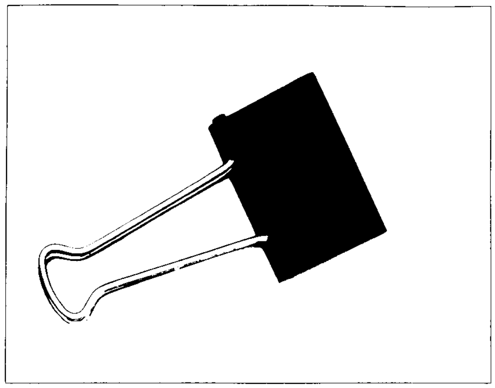

Fig. 6. A binder clip. The planner can orient and position this object unambiguously.

corners, nontrivial manipulation is possible. Understanding the ways by which task mechanics may be used to solve robot problems is an important question. The practical restrictions of our assumptions, as evidenced by the failure modes, suggest further directions of exploration.

With this in mind, we now recount the failures observed. One difficulty is that we must use an elevation somewhat larger than the estimated angle of friction, in order to assure that the object will really move. This means that the object will accelerate, and, in some cases, the resulting inertial forces may dominate the frictional forces. For instance, the object might keep rolling along a wall, past the edge where it should have stopped, or it might hit a wall hard enough to rotate to an unplanned contact.

Another problem is lack of uniformity in the coefficient of friction, which can cause unplanned rotations during a "jump" from one wall to the facing wall. In severe cases, associated with gross surface defects in the tray, the effective coefficient of friction would be large enough to prevent sliding.

The plastic kitchen utensil box had fillets along the junction of the floor and the walls. Occasionally, a wrench would hit the fillet and be reflected from a right-handed to a left-handed configuration, or *vice versa*. Obviously, this violates the assumption of planar motions.

A final difficulty arises from the planner's inability to determine a duration for a tilting operation. It has no means of estimating velocities, nor does it know the dimensions of the walls. The main problem is that occasionally the planner needs to move from a corner contact to a middle contact. We specified durations that were determined empirically.

There are three possible approaches to all of these difficulties. First, we can construct a better implementation of the planner's model of a tray. More careful control of materials and construction could reduce the problems associated with inertial and impact forces, and nonuniformity of the coefficient of friction. Altering the dimensions of the tray can also help, by shortening the length of jump operations, for example. The second approach is to add planning code that avoids problematic operations. For instance, longer jumps might be avoided while still allowing shorter jumps. The third approach is to relax some of the assumptions, and extend the planner to cover the broader model of the task mechanics. Perhaps inertial forces and impact forces could be exploited rather than avoided.

The third approach is the most promising and interesting, as it forces us to consider further the role of mechanics in manipulation. More generally, along the lines of extending the planner's scope, it is desirable to understand and predict the continuous behavior of objects. The planner employed in the current tray-tilting scheme only considers discrete contact states. It analyzes the effect of various applied forces for each such contact, using the local differential information obtained to predict global behavior. The extent to which this planning approach is valid or should be augmented depends on a better understanding of dynamic issues.

E. Observing Manipulation Programs in Action

One aspect of these experiments was particularly striking. Generally, a human watching a manipulation task in progress can recognize the purpose and method of each operation. We see picking, transfers, placing, tapping, tilting, aligning, insertions—all operations that we use ourselves, and which we can usually perform better than robots. By contrast, a human observing the tray-tilting experiments is usually baffled. We found it very difficult to construct tray-tilting programs by hand, or to orient an object in a tray with our eyes closed. When some tilting operation is in progress, its purpose is usually inscrutable, because we see only the actual orientation, not the set of possible orientations. Within the narrow domain of tray-tilting planning, robots can synthesize where humans cannot even analyze.

III. Theoretical Issues

To characterize the scope of an automatic planner, that is, to determine the set of manipulation problems the planner can solve, is of great importance. In the context of this paper, the question is: how does the scope of a sensorless planner compare with a sensor-based planner; and more specifically, what is the scope of the tray-tilting planner? Both of these questions will, for the time being, remain open, although some observations are possible. It is clear, for instance, that giving sensory operations to a planner will not reduce the planner's scope. After all, the robot could always choose to ignore the sensors. It is also not too hard to construct a problem that requires sensing. So the scope of sensor-based manipulation is a superset of sensorless manipulation. But we would like to know much more—how large the difference might be, and the characteristics that place a problem outside the scope of sensorless manipulation.

In this section, we will compare sensor-based and sensorless manipulation using a formal framework described by Lozano-Pérez, Mason, and Taylor [5] for sensor-based manipulation. An obvious characteristic of sensor-based programs is that they have conditional statements, choosing among alternative actions based on sensory information. Although it is tempting to say that sensorless manipulation programs do not make decisions, we will see that an alternate view exists, for which the decisions occur, but are implemented by the mechanics of the task.

A. The Formal Framework

This section briefly recapitulates the framework described by Lozano-Pérez, Mason, and Taylor [5], modified to reflect the lack of sensors. We should note at the outset that the only explicit role of sensors in that framework is to decide when an action has achieved its goal, and to determine which of several possible goals it achieved. The identity of an achieved subgoal determines which of several possible next actions should be taken. In general, sensors might be used in other ways—to construct or modify a world model, to implement actions, or to detect errors, for instance. None of these functions of sensors are explicitly modeled by the formal framework, and hence will not be considered here.

The basic premise is that, given a collection of goals and a commanded action, the planner constructs a collection of *pre-images* of the goals under the specified action. This collection of pre-images constitutes precisely those regions in space from which the commanded action is guaranteed to achieve one of the goals in a recognizable manner, despite uncertainty in sensing and control. Recognizability is crucial, as otherwise a robot might attain its goal but never know it. The pre-images thus defined constitute a new collection of subgoals, from which another level of pre-images may be formed. This process is repeated. When a subgoal is constructed that contains the initial robot configuration, the chain of subgoals represents a valid plan.

The pre-image methodology constitutes a formal description of the problem of planning in the presence of uncertainty. The particular form advanced by Lozano-Pérez, Mason, and Taylor assumes generalized damper dynamics and Coulomb friction, but the formal results are valid independent of any particular model of control and task dynamics.

In the pre-image methodology, sensors provide information to the *termination predicate,* whose job is to detect attainment of one of the goals, terminating the motion in progress, and initiating the appropriate next motion. The termination predicate used by Lozano-Pérez, Mason, and Taylor, took the current sensory information, and the current time, as input. The termination predicate described by Mason [8] took the entire sensor history, and the current time. The termination predicate described by Erdmann [2] took only current sensor readings, and had no sense of time. Using Erdmann's termination predicate, the complex pre-image computation problem reduces to a simpler *back-projection problem*, which we will be able to use in this paper.

For the present case, sensory information is not available, so the termination predicate can only use elapsed time. This simplifies the description of the pre-images somewhat. In particular, suppose that the initial state of a system is known to lie in some start region of an appropriate state space. Upon application of an action, the state of the system will evolve over time. At any given time, the state of the system must lie in some other region of state space, called the *forward projection.* The forward projection is a function of the start region, the commanded action, the control uncertainty, and the elapsed time. In the absence of sensory information, the forward projection describes precisely the certainty with which the state of the system is known. In order that the start region be a pre-image, there must exist at least one point in time at which any motion that originated in the start region may be terminated with certainty inside a goal. Notice that this termination time must be independent of the actual trajectory followed. Thus given a collection of goals and a commanded action, the start region is a pre-image if and only if its forward projection is wholly inside some goal at some time.

B. Comparison of Sensor-Based and Sensorless Manipulation

Let us briefly contrast planning with and without sensing. The role of sensing in the pre-image methodology is to recognize attained subgoals. Due to uncertainty, the subgoal attained at the end of a motion is generally not uniquely identifiable at the beginning of a motion. Sensing aids in recognizing which of a class of possible subgoals was indeed attained. This permits the plan executor to decide on the next motion to execute.

In the absence of sensing, the plan executor cannot indulge in sensor-based decisions. The subgoal to be attained at the end of a motion must be known precisely at the beginning of a motion. In fact, all commanded actions and all subgoals encountered are known at the beginning of task execution.

It appears then that sensing facilitates conditional plans. In the absence of sensing, plans assume a linear structure. However, let us look more carefully at the nature of the goals in each of these methods. Each subgoal in a sensorless plan tends to be a union of some of the subgoals appearing in a sensor-based plan. This is apparent in the tray-tilting scheme. The nodes consist of sets of individual contacts, rather than individual contacts. The nodes attempt to capture the uncertainty in contact type by grouping together those contacts that behave similarly under the application of tilt operations. Thus the nodes in the search space are elements of the power set of the contact states. This construction is similar to the construction of a deterministic finite automaton from a nondeterministic one.

These observations suggest that sensorless plans perform decisions. The decisions are mechanical decisions rather than conditional actions based on environmental inquiries. In sensor-based plans, decisions are of the form "If state S_1, then perform action A, and state S_2 will result." In sensorless plans, decisions are of the form "Perform action A. If state S_1, then state S_2 will result."

C. Back-Projections and Sensorless Pre-Images

So far we have considered a formal description of sensorless plans, and have compared these to sensor-based plans. One approach to computing sensor-based pre-images is to back-project motion constraints from distinguished subsets of the goals. This section examines such back-projections in the context of sensorless pre-images.

Given a collection of goals and a commanded action, the *back-projection* of the goals is the region of space from which all trajectories under the commanded action are guaranteed to reach at least one of the goals. In other words, any motion that begins in the back-projection as a result of the commanded action will eventually pass through one of the goal sets. For some tasks, such as compliant motion tasks in planar environments, back-projections may be computed directly from geometric descriptions of the environment, the goals, and the control uncertainty. See [3] for further details.

Notice that, unlike pre-images, back-projections contain no notion of goal recognizability. In particular, sensor readings are irrelevant. The lack of sensor requirements suggests that back-projections may be related to sensorless pre-images. However, whereas sensorless pre-images use time to recognize goal attainment, back-projections contain no notion of time. Therefore, in order to apply back-projections to the problem of sensorless manipulation, we must deal with the role of time in the recognition of goals. In the process we shall see that certain sensorless pre-images may indeed be described using back-projections.

Consider a single goal, and some subset of the goal. Suppose that for some action the forward projection of the subset never leaves the goal. In other words, any motion originating in the subset of the goal, as a result of commanding the action, remains in the goal. We will refer to such a subset as a *stationary* subset of the goal (relative to the commanded action). For instance, suppose the space of interest is the real axis, and the goal is the nonnegative side of the real line. Then the forward projection of the origin under an action that says "move right" forever remains in the goal. Thus the origin is a valid stationary subset. Of course, so is any subset of the positive axis. As another example, suppose that a block is resting on a tabletop. Consider applying a force subject to uncertainty. So long as all forces that actually might be applied, that is, all forces in the error ball about the nominal commanded force, lie within the friction cone of the table and block, the block will remain forever on the tabletop. Thus the table is itself a stationary goal relative to any such applied force.

Now suppose we have a stationary subset of some goal relative to some commanded action, and consider its back-projection. Recall, this means that any point in the back-projection is guaranteed to eventually reach the stationary set, hence forever remain in the goal set. Thus the back-projection is a pre-image in the limit as time approaches infinity. More practically, consider the maximum time t_{max} required to reach the stationary set via any motion beginning in the back-projection, given the commanded action. If this maximum time is finite, then the back-projection is actually a pre-image. Indeed, the termination predicate can successfully halt any motion that began in the back-projection, signaling entry into the original goal set once time t_{max} has elapsed. The maximum time assumption is valid in finite polyhedral environments. In more general environments it may be necessary to select a subset of the back-projection for which the assumption is valid.

As a slight variation, let us consider subsets of goals that are only partially stationary. Specifically, suppose that the forward projection of some subset G of a goal is guaranteed only to remain inside the goal for a finite amount of time, say until time t_{exit}. Again, form the back-projection of this partially stationary subset G. Now choose a subset B of this back-projection, from which all motions are guaranteed to reach G by time t_{enter}. If t_{enter} is less than or equal to t_{exit}, then B is a pre-image. Indeed, at any time in the range $[t_{enter}, t_{exit}]$ the termination predicate can successfully halt any motion that began in B at time zero. A typical example is given by any task involving dead reckoning.

D. Relation to the Tray-Tilting Scheme

The tray-tilting scheme is a form of pre-image computation via the back-projection method just described. The tray-tilting scheme is not implemented via back-chaining or back-projection. In fact, the tray-tilting method is implemented using a forward-chaining search. However, there is a difference between problem implementation and problem description. The nodes of the tray-tilting search graph correspond to the individual subgoals of the pre-image methodology. The azimuths correspond to the commanded actions. The nodes of minimal cardinality comprise the initial collection of goals. These are not known until the search algorithm completes. Nonetheless, conceptually, the nodes of minimal cardinality form the goal states. Finally, the initial node, consisting of all edge–wall contacts possible after the object has been dropped into the tray, corresponds to the starting region of the pre-image methodology.

In fact, with one exception, each node created after application of a particular tilt operation is a stationary subgoal—continued application of the same operator does not result in any motions leaving the node. The search algorithm actually computes the transitive closure of contacts reachable under a given applied force. Thus a given node S_1 is essentially the forward projection of some other node S_2. The second node S_2 is therefore a subset of the back-projection of the first node S_1. By our previous comments, this back-projection and hence the node S_2 must be pre-images of the node S_1.

The exception mentioned above deals with motions from corners into the middle of tray walls. In order to recognize transitions from corner contacts to edge–wall contacts, the search algorithm employs a crude form of the bounded time variation discussed previously. This permits the search algorithm to stop computing the transitive closure after the object to be oriented has slid from a corner contact to an edge–wall contact, but before the object has slid into the opposing corner.

E. Nondeterminism and Conditional Actions

In pursuing the view of sensorless manipulation as nondeterministic sensor-based manipulation, suppose that we create a plan under the assumption of perfect sensing. Suppose further that the plan executor actually does not possess any sensors. The question is how well can the plan executor execute the sensor-based plan without sensors. If the plan executor can operate nondeterministically, then it can execute the sensor-based plan faithfully. Specifically, at any decision point, the executor need merely guess the correct sensor values, nondeterministically choosing the next action to perform.

The point to observe is that sensing is entirely a matter of making decisions. Assuming that the physics of a task are fixed, sensing does not influence the class of actions possible, merely the recognition of success. Of course, the recognition of success decides the order in which actions are chained together. The extent to which this limits the class of solvable tasks is an open question.

Suppose that we insist on a deterministic plan executor, and we want to determine how well the executor can perform the sensor-based plan. There is no general answer, but we can at least set up the problem. Consider a single conditional action. Suppose that the start state is in one of the sets $\{R_i\}$. For convenience, suppose these start regions are nonoverlapping. Assume further that the conditional action says to command the action A_i if the actual start state is in the set R_i. The action is to be executed until one of the goals is known to have been attained.

Assume that the plan executor cannot distinguish between the different start regions $\{R_i\}$. We would like to know how well the plan executor can perform the conditional action. Actually, we do not require that the plan executor exhibit exactly the same effects as the conditional action, merely that it achieve one of the goals.

Observe that the question may be answered by applying the sensorless pre-image methodology described earlier, with initial start region $R = \cup R_i$. If there is any sequence of motions from R to one of the goals, the pre-image methodology will find it. The effects of this sequence may not parallel the effects of the conditional action, but at least they will achieve one of the goals. Unfortunately, applying the pre-image methodology offers only a procedural solution for particular cases, and only for single conditional actions. It does not provide in general terms a comparison of sensor-based manipulation with sensorless manipulation.

Let us consider a simple form of combining the component actions that comprise a conditional action. Specifically, suppose that there are only finitely many start regions $\{R_i\}$ and corresponding actions $\{A_i\}$, and suppose that we wish to perform the component actions in the order $A_1, \cdots, A_n$. This is analogous to converting a nondeterministic Turing machine into a deterministic machine. We can describe the conditions under which this choice of combination is guaranteed to achieve one of the goals. Specifically, there must exist times $t_1, \cdots, t_n$ such that commanding action A_1 until time t_1, followed by action A_2 until time t_2, and so forth, will eventually lead to one of the goals. Said differently, the repeated forward projection of the start region R under the actions $A_1, \cdots, A_n$ (with switching times $t_1, \cdots, t_n$) must be wholly contained in some goal.

For example, suppose that commanding a particular action A_i causes motions beginning in the corresponding start region R_i to enter a goal, and otherwise causes all motions beginning in a goal or in any other start region R_j, with $i \neq j$, to remain in either the goal or the set R_j. Then the previous conditions are satisfiable. In fact, any ordering of the commanded actions would achieve the desired result.

As a concrete example, consider a planar rectangular block that is not a square. Imagine sliding the rectangle on a horizontal edge. The block can be in two distinguishable states. Either a short side is resting on the edge, or a long side is resting on the edge. The goal is to slide the block into some region on its right, and to orient the block so that a long side is resting on the horizontal edge. A sensing strategy might contain the following conditional action: If the block is properly oriented, apply a force that will cause the block to slide to the right. If the block is improperly oriented, apply a force that will cause the block to orient itself properly while sliding to the right.

Suppose that the reorienting force is so chosen that it only rotates the block if the block is in the incorrect state. Suppose further that the goal is the entire region to the right of some point on the horizontal edge. Then the sensor-based strategy may be executed without using sensors to determine the orientation of the block. In fact, either ordering of the commanded forces will achieve the desired goal. The tray-tilting scheme is based on this method of combination.

IV. Conclusions

A. Solvable Tasks

Let us return to the issue of describing the scope of a planner, that is, the class of tasks the planner can solve. In this and previous papers we have explored a number of different planning methods. Some of our results relate to the scope of a planner, but usually in terms of some abstractly defined class of tasks. For instance, Mason [8] demonstrated that one variation of the pre-image planner can solve all tasks that have a valid solution. But what tasks are those? We have no means of determining whether a task is solvable, other than to try to construct a solution.

For the most part, various planners are compared not in terms of their scopes, but in terms of their descriptions. In other words, we compare the planning methods in terms of the termination predicates available and in terms of the back-projections used to compute pre-images. This type of comparison does not provide a characterization of the solvable tasks. While we have a notion of the generators of a methodology, we do not understand the relations between these generators. Thus we cannot predict the effect of removing or changing one of these generators. Missing are both absolute task descriptions and methods for extracting characterizations out of generator specifications.

For the present paper, it is desirable to characterize the tasks solvable by the tray-tilting scheme. As a necessary condition for an object to be orientable unambiguously, we know that it should not possess any rotational symmetries relative to its center of mass. The extent to which this condition is also sufficient is unknown. In fact, suppose that we eliminate the jump operator. Now consider the task of orienting a convex object whose interior angles are all acute. Suppose the coefficient of friction is zero. It is impossible to orient the object. The only transitions possible are sliding along a wall, which never rotates the object, and turning through a corner, which always rotates the object to an adjacent edge. It is impossible to reduce the number of possible orientations.

B. Future Work

Section II-D included some directions for improving the fidelity of the planner's models, by improving either the experimental tools or the planning system. In this section, we describe some other directions for future work.

First, the synthesis of various planning methods is desirable. While we have some understanding of individual operations, their combined behavior is not well understood. A planner should, for example, be able to use the pushing analysis of [1] or [10] to predict the outcome of the jump operator in the tray-tilting scheme. The pre-image methodology described above for dealing with uncertainty should be used to successfully handle errors arising from surface texture irregularities or control limitations. For example, it may be possible to bound the position and orientation of an object after some operation has been performed, using a dynamic analysis. If the bounded region includes points away from the desired contacts, small tapping or tilting operations might be used to regain contact with the desired walls and corners.

Second, one should not be bound to the use of a real tray. Rather, one should abstract the physical implementation of the tray-tilting scheme just as the planner abstracts the operators it applies. For example, instead of using a tray with immovable walls, one could simply use a tiltable plane with no walls, a couple of straight boards, and a couple of corners. Depending on the desired operator, one could arrange the boards and corners appropriately. The jump operator could be implemented by simply picking up the current contact wall and moving it to the other side of the object. The halfway sliding operator could be implemented by using boards of various lengths, or by moving a corner relative to the object. Naturally, this implementation introduces its own difficulties, related to the motion of the boards and corners. The point, however, is that a particular ideal behavior may be implemented in numerous ways. The particular method chosen should be influenced by such factors as the object type, the control capabilities, and an understanding of effects not explicitly modeled by the planner.

Third, one should consider more general mechanical operations than those available in the tray-tilting domain. The operations in the tray-tilting domain were limited to jumps, sliding, alignment with edges, and rotation at right-angled corners. In any practical application more versatile operations would probably be required. For instance, one could imagine trying to unravel the tray into a feeder of some sort. Suddenly right-angled corners seem unwieldy and restrictive, corners with arbitrary angles being more powerful. While the general methods of dealing with uncertainty and of sensorless planning carry over from the tray-tilting domain, the particular operators employed by the tray tilter may not. In any application, we should use the theoretical tools discussed earlier to guide our approach, while developing more robust and versatile actions.

C. Summary

This paper has explored sensorless manipulation in the domain of planar parts orienting. A planner was developed for orienting parts dropped onto a tray under the influence of gravity. The planner consisted of two phases. The first phase was a contact analysis phase. This phase analyzed the various edge–wall and edge–wall–corner contacts possible. The output of the contact analysis phase was a description of the transitions achievable between different contacts as a result of tilting the tray. The second phase of the planner consisted of a search phase. The search phase used the contact transitions to determine a sequence of tray-tilting operations that would orient the part unambiguously. In cases where unambiguous orientations could not be achieved, the search phase would try to minimize the number of final orientations.

This paper also explored some theoretical issues of sensorless manipulation. In particular, the paper described sensorless manipulation within the formal framework of the pre-image methodology. Special cases of pre-images were shown to be variants of back-projections. The tray-tilting scheme was recognized as one such special case. Finally, it was noted that comparisons between different manipulation schemes often occur at the level of generating components, rather than at the level of solvable tasks.

ACKNOWLEDGMENT

The authors wish to thank R. Brost and T. Wood for their contributions.

REFERENCES

[1] R. C. Brost, "Automatic grasp planning in the presence of uncertainty," in *Proc. 1986 IEEE Int. Conf. on Robotics and Automation* (San Francisco, CA, Apr. 7–10, 1986), pp. 1575–1581.

[2] M. A. Erdmann, "On motion planning with uncertainty," Artificial Intelligence Lab. MIT, Tech. Rep. AI-TR-810, 1984.

[3] ——, "Using backprojections for fine motion planning with uncertainty," *Int. J. Robotics Res.*, vol. 5, no. 1, pp. 19–45, 1986.

[4] D. D. Grossman and M. W. Blasgen, "Orienting mechanical parts by computer-controlled manipulator," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-5, pp. 561–565, Sept. 1975.

[5] T. Lozano-Pérez, M. T. Mason, and R. H. Taylor, "Automatic synthesis of fine-motion strategies for robots," *Int. J. Robotics Res.*, vol. 3, no. 1, pp. 3–24, 1984.

[6] M. Mani and W. Wilson, "A programmable orienting system for flat parts," in *Proc. NAMRI XIII* (Berkeley, CA, May 19–22, 1985).

[7] M. T. Mason, "Manipulator grasping and pushing operations," Artificial Intelligence Lab., MIT, Tech. Rep. AI-TR-690, 1982.

[8] ——, "Automatic planning of fine-motions: Correctness and completeness," in *Proc. 1984 IEEE Int. Conf. on Robotics and Automation* (Atlanta, GA, Mar. 13–15, 1984), pp. 492–503.

[9] ——, "The mechanics of manipulation," in *Proc. 1985 IEEE Int. Conf. on Robotics and Automation* (St. Louis, MO, Mar. 25–28, 1985), pp. 544–548.

[10] ——, "Mechanics and planning of manipulator pushing operations," *Int. J. Robotics Res.*, vol. 5, no. 3, pp. 53–71, 1986.

[11] B. K. Natarajan, "An algorithmic approach to the automated design of parts orienters," in *Proc. 27th Annu. IEEE Symp. on Foundations of Computer Science* (Toronto, Ont., Canada, Oct. 27–29, 1986), pp. 132–142.

[12] M. A. Peshkin, "Planning robotic manipulation strategies for sliding

objects," Ph.D. dissertation, Physics Dept., Carnegie-Mellon Univ., Pittsburgh, PA, 1986.

Michael A. Erdmann was born in Augsburg, Bavaria, on June 10, 1959. He received the B.S. degree in mathematics from the University of Washington, Seattle, and the S.M. degree in computer science from the Massachusetts Institute of Technology, Cambridge.

Currently he is a graduate student at MIT, working in the Artificial Intelligence Laboratory.

Matthew T. Mason received the B.S., M.S., and Ph.D. degrees from the Massachusetts Institute of Technology, Cambridge, in 1976, 1978, and 1982, respectively.

Since the Fall of 1982 he has been an Assistant Professor in the Computer Science Department, with a joint appointment in the Robotics Institute, at Carnegie-Mellon University, Pittsburgh, PA. His research interests are in robotics and automatic planning. He is co-editor of *Robot Motion: Planning and Control* (MIT press, 1982) and *Robot Hands and the Mechanics of Manipulation* (MIT Press, 1985).