On Moving and Orienting Objects

B.K. Natarajan

TR 86-775
(Ph. D. Thesis)
August 1986

Department of Computer Science
Cornell University
Ithaca, NY 14853

ON MOVING AND ORIENTING OBJECTS

A Thesis

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Balasubramaniam Kausik Natarajan

August, 1986

ON MOVING AND ORIENTING OBJECTS

Balasubramaniam Kausik Natarajan, Ph.D.

Cornell University, 1986

Many problems arising in the area of robotics are directly or indirectly motion related. In order to analyze such problems, it is necessary to incorporate the dynamics with the geometry in the mathematical formulation. With this in view, this thesis deals with two such problems – motion planning in the presence of uncertainty and the automated design of parts orienters.

Motion planning for robots with errors in position measurement, velocity and time is considered and shown to be decidable in polynomial time for a large class of inputs. The robot model is then extended to include damping – a limited form of force sensing. Motion planning for point objects in three-dimensional scenes and robots with damping is shown to be PSPACE-hard. A simplified version of the same problem is shown to be PSPACE-complete.

The problem of the automated design of parts orienters is rather closely related to motion planning. But the dynamics of the problem is so dominant that similar general formulations seem impossible. In this thesis, the alternative pursued is paradigm-by-paradigm analysis. Three paradigms are presented and analyzed – the "belt", for orienting convex polygons that are infinite in the third dimension, the "pan handler", for flat polygonal objects and the vibratory track for flat, convex polygons. Polynomial time algorithms are developed for the automated design of orienters in each of the paradigms.

Biographical Sketch

B.K. Natarajan earned a Bachelor of Technology degree in 1981 from the ıdian Institute of Technology, Madras and a Master of Science in Engineering om Princeton University in 1983.

Acknowledgements

I am extremely grateful to my advisor John Hopcroft for being advisor, friend and guru during my years at Cornell.

Many thanks to Charles Van Loan and David Caughey for serving on my committee and to John Gilbert, Juris Hartmanis and Dexter Kozen for their superb teaching, and for giving so freely of their time.

Despite winter's every effort, my years at Cornell were surprisingly pleasant. I thank all the staff, students and faculty who made them so.

Support for this thesis was provided in part by the U.S Army Research Office through the Mathematical Sciences Institute of Cornell University.

'able of Contents

List of Figures

CHAPTER 1

INTRODUCTION

Many problems arising in robotics are directly or indirectly related to the motion of objects in the physical world. In order to analyze such problems it is essential to model the motion involved. As a first approximation, one might consider a purely geometric or mathematical model that ignores the dynamics of the situation. For example, consider motion planning – determining whether a given robot can move a given object from one point to another in a three dimensional scene without colliding with the obstacles in the scene. A purely geometric model of the problem would be path planning, i.e, to simply determine whether there exists a path linking the two locations of the object along which the object can be moved without contacting the obstacles. This formulation of the problem is mathematically well-posed and can be tackled with conventional methods as is done in [Udupa, 1977, Reif, 1979, Lozano-Perez and Wesley, 1979, Schwartz and Sharir, 1982]. However, one would like to consider more realistic situations where the robot involved suffers from limited precision in its measurements of position, velocity, time etc. Such considerations are not easily assimilable into geometric formulations. Furthermore, we might be interested in robots whose sensory abilites are augmented by force sensing, vision, etc, features that are rather difficult to introduce into geometric formulations. [Lozano-Perez et al, 1984, Mason, 1983] document recent attempts at motion planning in the presence of uncertainty and limited force sensing.

In Chapter 2 of this thesis, we analyze the computational complexity of motion planning in the presence of uncertainties in position measurement, velocity and time. We show that despite these uncertainties, there are reasonably general conditions under which the problem can be reduced to the geometric formulation of motion planning – the classical "piano mover's problem". We then extend our model of the robot to include damping, a rather crude form of force sensing that allows the robot to negotiate obstacles by sliding along their surfaces. As is to be expected, damping enables the robot to perform tasks that would be impossible otherwise. The main result of the chapter is that deciding motion planning for robots with damping is PSPACE-hard. For completeness, we also exhibit a restricted version of the problem that is PSPACE-complete. These results are the first complexity analysis of the problem of motion planning with uncertainty, as the earlier work mentioned above consider solution procedures without regard to their running times. The results are significantly different from the related PSPACE-hardness results on generalized "piano moving" [Reif, 1979, Hopcroft, Joseph and Whitesides, 1985, Joseph and Plantinga, 1985] in that they concern motion planning for single rigid objects as opposed to composite objects with unboundedly many degrees of internal freedom. A more detailed introduction to the problem is provided in Chapter 2.

Another motion-related problem is the design of parts orienters. The motivation for this problem is as follows: Consider a robot that is to assemble a composite object by mating two parts. The precision to which the task is to be performed is governed by the tolerances in the mating parts. Even if the robot is capable of movements within this precision, considerable uncertainty is introduced when the robot grasps and picks up a part in the absence of vision. Since parts are often stored jumbled up in a bin of some sort, they have to be oriented precisely with respect to the robot before the assembly task can be attempted. Hence the need for an orienter – a device that accepts a particular part in any orientation from the bin and delivers it to the robot in some predetermined orientation.

Currently, parts orienters are designed entirely by hand by the ınufacturing engineer [Boothroyd et al 1982, Philip 1980]. In a Computer ded Design/Manufacturing environment, this is unacceptable as it only takes a v minutes to design a small part and it would be desirable to generate the sembly process for the part in comparable time. With this in view, we are erested in automating the design of parts orienters, i.e, to develop an gorithm that takes the shape of a part and the final orientation desired as outs, and determines the parts orienter that satisfies the requirements. If no enter satisfies the requirements, the algorithm should perhaps suggest odifications to the shape of the part so as to permit an orienter. Other work in s area may be found in [Mason and Erdmann, 1986, Lozano-Perez, 1986 and ossman and Blasgen, 1975].

Surprisingly enough, this problem is rather closely related to the motion nning problem mentioned earlier. In motion planning, we are given a scene of stacles and are interested in finding out whether a given object can be moved m one configuration in the scene to another. Here, we are to design the scene obstacles that forces the object to move from one set of configurations to other. Some caution should be exercised here so as not to draw the analogy too . While motion planning possesses a reasonable geometric formulation, the oblem of parts orienter design does not. This is because the problem is rather ongly governed by the physics of the motion involved. Notice that static stacles in the scene cannot cause the object to move from one configuration to other. Consequently, tacit in our discussion is some driving force that causes e object to move. Accounting for this force veers us away from simply inverting e statement of the motion planning problem to obtain a statement of the parts ienter problem. We are forced to resort to a paradigm-by-paradigm analysis of e problem where each paradigm is characterized by the model of the driving

force that moves the object. A detailed introduction to the problem may be fou in Chapter 3.

In Chapter 3, we develop algorithms for three different paradigms for pa orienters. The "belt" paradigm considers a horizontal belt moving at consta speed with steps at intervals. A two dimensional object placed on the belt m suffer repeated reorientation as it traverses the steps. The idea is to choose t height of the steps and their sequential order so that the object placed at one e of the belt in any orientation reaches the other end in a predetermined a unique orientation. The "pan handler" paradigm describes a flat, polygonal tr meant for orienting flat polygonal objects. The object is dropped on the tray a oriented by tilting the tray at various angles. At each tilt the object may reorie itself, the aim being to find a sequence of tilt directions that uniquely orients t object regardless of the orientation it starts with. Our third paradigm is t "vibratory track", a device that is rather commonly used in practice. It consists a horizontal track with a lip along which the planar polygonal object is vibrate The face of the object resting against the lip determines the orientation of th object. Select orientations of the object may be reoriented by means of pins place on the track or discarded by means of cut-outs in the track. The aim is to find sequence of pin and cut-out locations so that the object exits the track in som predetermined orientation regardless of the orientation in which it entered th track. Furthermore, for a given probability distribution of the object over it orientations at entry, the probability of the object exiting the track is to be maximum over all such sequences.

The development of algorithms for the above paradigms and their analysi give rise to rather interesting problems in graph pebbling and function composition over finite sets. We provide polynomial time algorithms for these problems without regard to constants, exponents or optimal solutions. Apart from being theoretically interesting, the results provide useful insight into the physical problems themselves. In short, we attempt to provide a theoretical basis for this practical problem.

CHAPTER 2

THE COMPLEXITY OF FINE MOTION PLANNING

2.1 INTRODUCTION

Motion planning for robots has received considerable attention in recent years. But the attention has been rather partial to gross motion planning at the cost of fine motion planning, which is a little studied although equally important problem. Loosely speaking, the term *gross motion* is applicable to situations in which the uncertainty in the relative position of the goal and the object to be moved is negligible, while the term *fine motion* is applicable to situations in which this uncertainty is significant. As an example, consider the typical problem of inserting a peg in a hole: Fig. 2.1a shows three possible positions for

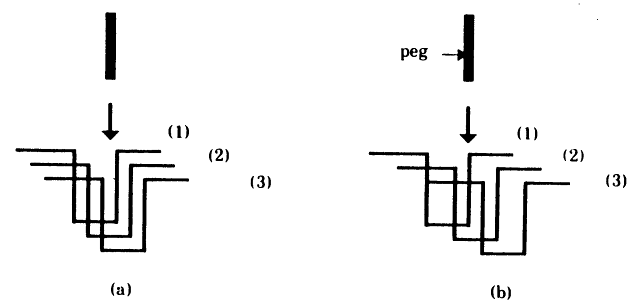

Fig. 2.1. Motion planning with uncertainty; an example

the hole (1), (2) and (3), with the peg above the hole. We are required to plan a motion for the peg that would insert it in the hole. We see that a straight line motion downward will successfully insert the peg regardless of whether the hole ıt position (1), (2) or (3). Therefore the uncertainty in the position of the hole is ignificant and this is an example of gross motion planning. Fig. 2.1b on the er hand is not so straightforward. There is no *a priori* motion that will form the insertion, and any attempt to insert the peg in the hole will require adjustments that depend on the exact location of the hole. Hence we conclude t the uncertainty in the position of the hole is significant here and that this is example of fine motion planning. A slightly different version of the same

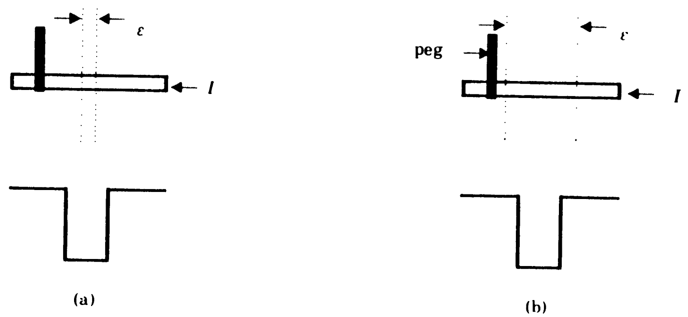

Fig. 2.2. Motion planning with uncertainty.

ɔlem is shown in Fig. 2.2. Here the position of the hole is fixed, but the initial tion of the peg can be anywhere in the region marked *I*. The difficulty arises n the position of the peg is known only within some error ε. If ε is smaller ı the clearance of the peg in the hole, Fig. 2.2a, the task is possible with tion feedback alone, since we could measure the position of the peg and utilize information to plan a motion that will successfully insert the peg in the hole. , Fig. 2.2b, attempts at inserting the peg in the hole will require additional rmation on the relative position of the peg with respect to the hole. This rmation could be obtained by any sensory mode – force sensing, generalized

damping, vision etc. However, Fig. 2.2a and Fig. 2.2b are both examples of fine motion planning as no single motion will perform either task.

2.1.1 The Robot Model and A Formal Statement of the Problem

Robots With Position Measurement

We now describe the robot model on which we base our discussion. The model is that of [Mason, 1983, Lozanó-Perez et al 1984]. In the following, we shall use the term *effector* to refer to a fixed reference point on the object gripped by the robot. The location of any point on the gripped object with respect to this point is known precisely at all times.

The effector has up to six degrees of freedom - three translational and three rotational. The location of the effector in some fixed global frame is known at any time to within a fixed error bound ε. In particular, the actual position of the effector is always within a sphere of radius ε centered at the observed position and vice versa. See Fig. 2.3a. Similarly, the orientation of the effector is known within a fixed error bound ε_0 at all times and the actual orientation of the effector is within a sphere of radius ε_θ centered at the observed orientation. We

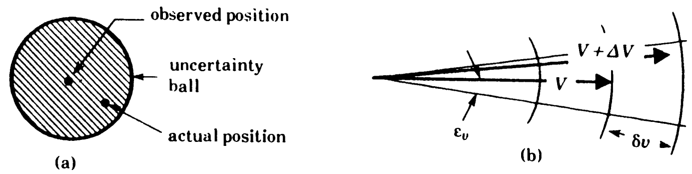

Fig. 2.3. (a) The uncertainty ball in position measurement
(b) The uncertainty in velocity

ll also allow for velocity errors in the robot – differences between the nmanded velocity and the velocity actually attained by the effector. (We use e following convention for vectors: vector quantities are in upper case while the rresponding scalars are in lower case). The attained velocity $(V + \Delta V)$ is within angle ε_v in direction and within δv in magnitude of the commanded velocity ctor V. The assumption is illustrated in Fig. 2.3b, where the shaded area is the t of possible locations for the arrowhead of the attained velocity vector. The agnitude of the commanded velocity, v, can be any real positive value. The ror magnitude δv is assumed to vary linearly with v, while ε_v is assumed variant. A motion plan M for the above effector consists of a sequence of moves $_1, m_2, \ldots$. Each move m_i is structured thus:

```
while F_i(p) do
    move with velocity f_i(p)
    for time interval τ;
od
```

Iere $F_i(p)$ is a boolean function of the observed effector position p and any rogram variables. The function f_i is the velocity function associatated with the h move and maps observed effector positions to command velocities. Every pplication of $f_i(p)$ to the effector is terminated after time interval τ, the *time-out* eriod, which can take any positive real value. Associated with the time-out eriod τ is an error $\delta\tau$, which is assumed to vary linearly with τ. The omputations involved in a motion plan are carried out to some fixed precision hat is independent of the aforementioned error parameters.

The fine motion planning problem for robots with position measurement is as ollows:

Problem 2.1

<u>Input:</u> A three dimensional *scene* S consisting of a finite set of planar walls in a closed and bounded three dimensional region D, a rigid polyhedral object O, a set of *initial positions* for the object I and a set of *goal positions* G.

<u>Property:</u> Is there a motion plan M for a given robot with a fixed error ε in position measurement that moves O from every position in I to some position in G without contacting any of the walls in S or leaving D? (We refer to such a plan M as a *fine* motion plan from I to G.)

We illustrate the problem with the following example in two dimensions.

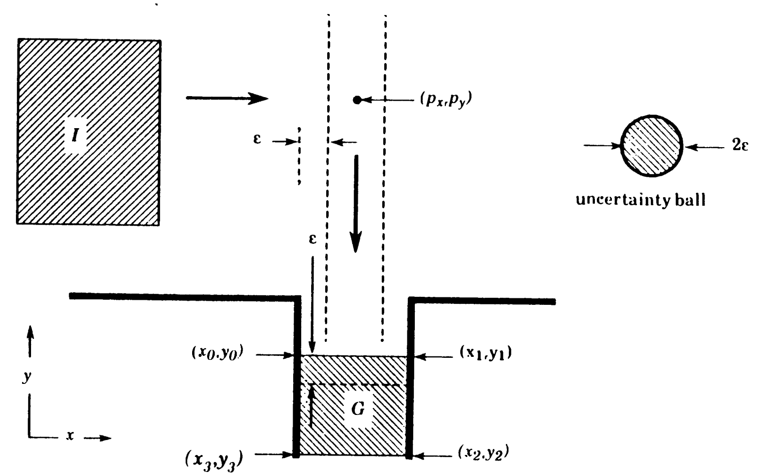

Fig. 2.4. Fine motion planning with position measurement; an example.

Example 2.1

The problem as shown in Fig. 2.4 requires finding a motion plan to move a point object from the set of initial positions I to the set of goal positions G. The uncertainty ball associated with position measurement is shown in the top right corner of Fig. 2.4. We notice that a point directly above G can be moved downward into G. What constraints should the observed position (p_x, p_y) of the effector satisfy if its actual position is to be directly above G? Points with $p_x \geq x_0+\varepsilon$ and $p_x \leq x_1-\varepsilon$ are acceptable as they always correspond to actual positions that are directly above G. Our plan now is to move the effector to the right until $x_0+\varepsilon \leq p_x \leq x_1-\varepsilon$ is satisfied. Then, a downward motion until $y_3+\varepsilon \leq p_y \leq y_1-\varepsilon$ will put the effector in the goal. In particular, Fig. 2.5 is a two-

```
(px,py) := observed effector position;
pick τ such that (τ + Δτ) ≤ (1/v) min(x1-x0-2ε, y0-y2-2ε);
while px ≤ x0-ε do
    move in the +x direction at speed v for time τ;
od
while py ≥ y0-ε do
    move in the -y direction at speed v for time τ;
od
```

Fig. 2.5. A motion plan for Example 2.1.

move fine motion plan for the problem. Here τ is picked to be small enough that overshoot of the goal and subgoal regions does not occur. □

Robots with Damping and Position Measurement

Next we extend our robot model to include *damping*. Damping is a limited ability to conform to obstacles encountered during a motion. The concept is best explained with an example. Fig. 2.6a shows the path traversed by the effector of a robot with damping. The commanded velocity is V at position (1) on the path and is equal to the attained velocity. When the effector strikes the wall at

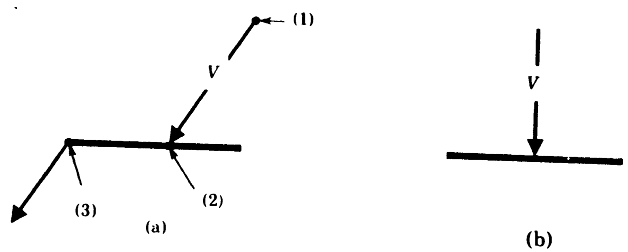

Fig. 2.6. Damped motion.

position (2) it conforms to the wall by attaining a velocity equal to the component of the commanded velocity parallel to the wall. The effector slides along the wall until it reaches position (3) at which point it is no longer constrained by the wall and hence reattains the commanded velocity. In Fig. 2.6b, the commanded velocity is perpendicular to the wall and hence the effector sticks to the wall as the component of the commanded velocity parallel to the wall is zero. In general, the effector will stick or slide depending on whether or not the commanded velocity is within the friction cone of the wall. The friction cone is a range of velocities about the normal to the wall that is determined by the coefficient of friction between the wall and the effector. For our purposes we need only know that incidence within the friction cone leads to sticking while outside the friction cone sliding occurs. See Fig. 2.7. A more complete discussion of friction cones can be found in [Mason, 1983]. In the presence of directional velocity errors, if there are to be commanded velocities that guarantee sticking on a wall, the friction cone should be larger than the directional velocity error. We will assume this to be the case in our discussion.

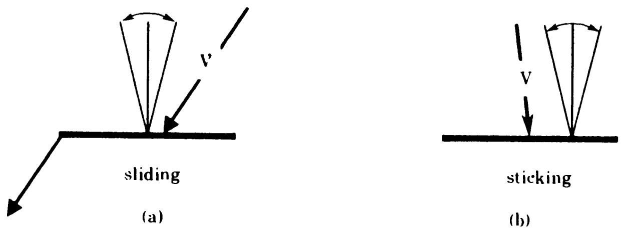

Fig. 2.7. The friction cone.

A motion plan in this setting is almost identical to the one for robots without damping. The only difference lies in the range of values the time-out period τ can take. It is essential that the following restriction be placed on v, δv, τ, $\delta\tau$:

$|(V + \Delta V)(\tau + \Delta\tau) - V\tau| \geq \varepsilon.$

Otherwise, it may be possible to reach any location in the scene with better than ε accuracy, making the uncertainty parameter ε meaningless. This is made clear in the following example.

Example 2.2

Consider the scene of Fig. 2.8. Here, the corner vertex at the intersection of the two walls can be used as a reference point. The effector moves towards the wall, slides along it and sticks at the corner (labeled reference point). Now the position of the effector is known to the same precision as the location of the corner, which can be much better than the uncertainty ball. The effector can then be moved to any location with no more than $|(V + \Delta V)(\tau + \Delta\tau) - V\tau|$ error. Hence

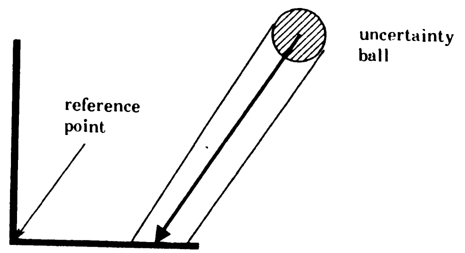

Fig. 2.8. Finding a reference point.

$|(V + \Delta V)(\tau + \Delta\tau) - V\tau| \geq \varepsilon$ should hold if the uncertainty parameter ε is to be meaningful. □

The statement of the problem in this setting is similar to the one given earlier with the exception that the robot now has the additional capability of damped motion.

Problem 2.2

Input: A three dimensional *scene S* consisting of a finite set of planar walls in a closed and bounded three dimensional region D, a rigid polyhedral object O, a set of *initial positions* for the object I and a set of *goal positions* G.

Property: Is there a motion plan M for a given robot with a fixed error ε in position measurement and damping that moves O from every position in I to some position in G without leaving D?

We illustrate the problem with the following example in two dimensions.

Example 2.3

The problem shown in Fig. 2.9 is similar to the one of Fig. 2.4 and requires

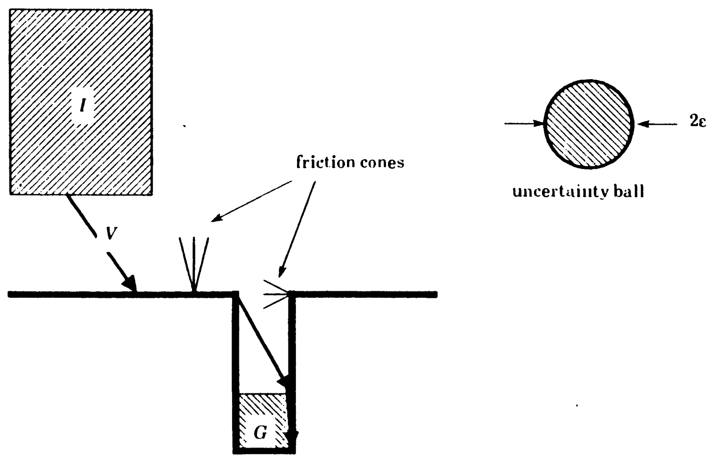

Fig. 2.9 Fine motion planning with damping; an example

moving a point object from the initial set I to the goal set G. Notice that there is no point x in G such that the uncertainty ball around x is entirely in G. Hence, position measurement alone cannot guarantee that the robot has attained the goal. It follows that there is no motion plan from I to G based on position measurement. (These intuitive ideas will be formalized later in Theorem 2.1). Assuming that the directional velocity error ε_v is zero, it is easy to verify that the velocity V shown in the figure is sufficient to move every point in I to some point in G. Notice the friction cones on the two walls and that V is outside both of them.

2.1.2 Overview of Results

In Section 2.2 we exhibit a direct reduction from fine motion planning f robots with position measurement to the path planning problem for rigid objects the"classical piano mover's" problem – which can be solved in polynomial tim In Section 2.3 we show that fine motion planning for robots with damping PSPACE-hard. This is the main result of this chapter. Although there ar algorithms in the literature [Mason, 1983, Lozano-Perez et al, 1984] for thi problem, this is the first attempt at the complexity of the problem. The result i significantly different from other PSPACE-hardness results [Reif, 197 Schwartz and Sharir, 1982, Joseph and Plantinga, 1985] related to motio planning in that it concerns rigid objects as opposed to objects with unbounde degrees of freedom. In Section 2.4 we show a restricted version of the problem t be PSPACE-complete. We then identify the key to the hardness of the problen and suggest a restriction that allows a polynomial time solution.

2.2. FINE MOTION PLANNING FOR ROBOTS WITH POSITION MEASUREMENT

2.2.1 Some Basic Results

We notice in our first example that a fine motion plan from I to G may exist even if I cannot be moved to G as a rigid body. In fact, a fine motion plan from I to G may be viewed as a translation of any continuous deformation of I to G. The limitation is that I cannot be deformed into an object smaller than an uncertainty ball. Before we make this more precise in the following lemma, we make some modifications to our robot model. In particular, in this section and the next, we ıll consider robots that are capable of traversing arbitrary paths without any locity error. (This is equivalent to piecewise linear motion with infinitesimal ıves and can be realized by the model of Section 2.1.1 as a limiting case.) We ıo assume that the motion plan is executed on a machine capable of real ithmetic. These modifications are in the interest of clarity and in Section 2.2.3, address the issue of approximate solutions for robots with piecewise linear ıtion plans and finite precision arithmetic.

ımma 2.1: Let G be the goal set for a point object in a three dimensional scene Let x be any point in S such that $B(x)$, the uncertainty ball centered at x, does t contact any of the walls. Then, there exists a fine motion plan from $B(x)$ to G and only if there exists a path along which $B(x)$ can be translated as a rigid dy to a position in which it is entirely in G.

roof: Let x_G be a point in G such that $B(x_G) \subseteq G$. Let O be the rigid body defined the boundary of $B(x)$ and let R be the path connecting O centered at x to O ntered at x_G. We now describe a fine motion plan M from $B(x)$ to G. In ırticular, M moves the point object from $B(x)$ to $B(x_G)$ by tracing out the path R. learly M maintains the point object within the volume swept by O along path R. ence there can be no contact between the point object and the walls in S. Also, hen M terminates, the point object is guaranteed to be in G as the uncertainty ıll around x_G is completely contained in G. We conclude that M is a legal fine ıotion plan from $B(x)$ to G.

Suppose there exists a fine motion plan M from $B(x)$ to G. Let R be the path aced out by x when M is applied to x. At any point x' on R, $B(x')$ cannot contact ny wall in S. For otherwise, there exists a location of the effector contacting a all in S with observed position x', implying that M is not a valid fine motion lan. Furthermore, let R terminate at a point x_G in G. Now, if $B(x_G)$ is not

entirely in G, M is invalid. It follows that R is a path along which $B(x)$ can b translated to a position where it is entirely in G. □

Before we proceed to the main theorem of this section, we need the followin definition: A set I is *star shaped* about a point x with respect to the uncertaint ball B, if $B(x) \subseteq I$ and for any $y \in I$ and $z \in B(x)$, the line yz is in I. Fig. 2.10 is a example of a two dimensional set I that is star shaped with respect to the ball B.

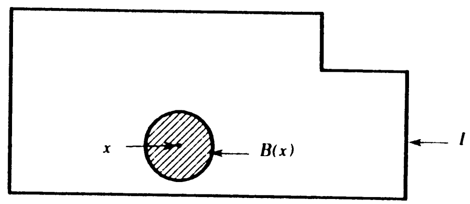

Fig. 2.10. A star shaped set.

Proposition: If a set I is convex and contains a point x such that $B(x) \subseteq I$, then is star shaped with respect to B.

Proposition: A closed set I is star shaped with respect to B if there exists a point x in I such that $B(x) \subseteq I$ and for any boundary point y of I and any boundary point of $B(x)$, yz is in I.

Theorem 2.1: Let I and G be the initial and final sets for a point object in a three dimensional scene. If I is star shaped about x with respect to B, then there exists a fine motion plan from I to G iff there exists a path along which $B(x)$ can be translated as a rigid body to a position where it is entirely in G.

Proof: Suppose there exists a fine motion plan M from I to G. Then M is also a fine motion plan from $B(x)$ to G. It follows from Lemma 2.1 that there exists a path along which $B(x)$ can be translated to a position in which it is entirely in G.

Suppose there exists a path R along which $B(x)$ can be translated into G. Then by Lemma 2.1 there exists a fine motion plan M from $B(x)$ to G. We now construct a fine motion plan $M' = mM$ from I to G where m is the move given by

begin

  $p :=$ observed position;

    trace the line joining p and x;

end

Since $B(x) \subseteq I$, at the end of move m the effector is guaranteed to be in $B(x)$ regardless of its inital position in I. Since I is star shaped with respect to B, the effector is always within I during m and hence cannot contact any of the walls in S. Since M is a fine motion plan from $B(x)$ to G it follows that mM is a fine motion plan from I to G. □

2.2.2 A Polynomial Time Reduction to the Classical Piano Mover's Problem

Theorem 2.1 suggests a direct reduction from fine motion planning with position measurement to the "classical piano mover's problem" – path planning for rigid objects with *no* uncertainty – which is decidable in polynomial time [Reif, 1979, Schwartz and Sharir, 1982]. We reproduce below the statement of the problem as presented in [Reif, 1979].

The classical piano mover's problem in d-space is:

<u>Input:</u> (R,S,p_I,p_F) where R is a set of polyhedral obstacles fixed in Euclidean d-space, and S (say, a sofa) is a rigid polyhedron with distinguished positions p_I and p_F.

<u>Property:</u> Can S be moved (by a sequence of translations and rotations in d-space) from position p_I to p_F without contacting any element of R?"

We now exhibit a polynomial time reduction from fine motion planning in our setting to the classical piano mover's problem. Consider the set core(G) defined on the goal set G as follows:

  core$(G) = \{x | x \in G$ and $B(x) \subseteq G\}$.

This set has the property that if a point is observed to be in core(G) then it is definitely in G. Also, it follows from Theorem 2.1 that if core$(G) = \phi$ then G is not attainable for any I such that core$(I) \neq \phi$. We also observe that if a and b are two points in core(G) that are path connected in core(G) then for any point x in S, there exists a rigid translation of $B(x)$ to a iff there exists a rigid translation of $B(x)$ to b. We are now ready for the reduction to the piano mover's problem.

Problem

Given a set I of initial positions for a point object such that I is star shaped about a point x with respect to the uncertainty ball B and a convex polyhedral set G of goal positions in a three dimensional scene S, is there a fine motion plan from I to G?

Reduction

(1) If core(G) is empty, answer NO.

(2) Else, pick a point x_G in core(G). Answer YES iff there exists a rigid translation of $B(x)$ to $B(x_G)$.

Proof: Immediate from Theorem 2.1. □

Now, Step(1) is computable in polynomial time as G is convex and polyhedral. Since the piano mover's problem is decidable in polynomial time, we conclude that Step(2) and hence the fine motion planning problem as stated above is decidable in polynomial time.

2.2.3 An Approximate Solution

Next we consider the problem of fine motion planning for robots with non-zero :locity error, piecewise linear motion plans and fixed computational precision. ere, a tight reduction to rigid body motion planning is impossible as the set of ›ssible locations for the object grows as it is moved around. Consequently, we ·esent an approximate solution in the form of weaker versions of Lemma 2.1 and ıeorem 2.1. First, we need some definitions. The *discretization parameter* δl is a easure of the distance moved by the effector in any iteration of a move. It ›sorbs the error in the velocity, the time out period as well as the piecewise near approximation. The *computational error parameter* α allows for fixed recision execution of the motion plan and should be chosen accordingly. xtending our definition of the uncertainty ball B, we define B_δ to be the ncertainty ball B expanded by $\delta l + \alpha$.

emma 2.1': Let G be the goal set for a point object in a three dimensional scene . Let x be any point in S such that $B(x)$, the uncertainty ball centered at x, does ot contact any of the walls. Then,

(1) there exists a fine motion plan from $B(x)$ to G if there exists a path along which $B_\delta(x)$ can be translated as a rigid body to a position entirely in G.

(2) there does not exist a fine motion plan from $B(x)$ to G if $B(x)$ cannot be translated as a rigid body to a position where it is entirely in G.

'roof: Let x_G be a point in G such that $B_\delta(x_G) \subseteq G$. Let O be the rigid body efined by the boundary of $B_\delta(x)$ and let R be the path that translates O from x to G. If R is not piecewise linear, break it up into segments at points $x_1, x_2, x_3, \ldots$, uch that the line joining x_{i-1} and x_i deviates no more than $\delta l_i < \delta l$ from R. We now describe a fine motion plan $M = m_1 m_2 \ldots$ from $B(x)$ to G. In particular, the i^{th} move m_i moves the point object from $B(x_{i-1})$ to $B(x_i)$ and is structured thus:

While observed position $p \neq x_i$ **do**

move along the line joining p and x_i towards x_i

at speed v for time τ:

od

The speed v and the time interval τ are to be chosen for each move to satisfy:

$|(V + \Delta V)(\tau + \Delta\tau) - V\tau| + \delta l_i \leq \delta l.$

Clearly M maintains the point object within the volume swept by O along path R. Hence there can be no contact between the point object and the walls in S. Also, when M terminates, the point object is guaranteed to be in G as the uncertainty ball around x_G is completely contained in G. We conclude that M is a legal fine motion plan from $B(x)$ to G.

The second claim in the lemma follows from Lemma 2.1. □

Theorem 2.1': Let I and G be the initial and final sets for a point object in a three dimensional scene. If I satisfies both of the following conditions:

(1) I is star shaped about a point x with respect to B

(2) I grown by $\alpha + \delta l$ is a valid set of configurations for the point object

then

(1) there exists a fine motion plan from I to G if there exists a path along which $B_\delta(x)$ can be translated to a configuration in which it is entirely in G.

(2) there does not exist a fine motion plan from I to G if there does not exist a path along which $B(x)$ can be translated to a configuration in which it is entirely in G.

Proof: An application of Lemma 2.1' as in Theorem 2.1. □

2.2.4 Objects Other than Point Objects

Finally we tackle the case of a three dimensional object O in a three ensional scene S. Rather than work with the solid object in three dimensions, transform the problem to moving a point object in six-dimensional figuration space as follows. For any three dimensional scene S we can struct a configuration scene S' in six dimensional space such that a point $x_1,x_2,x_3,\theta_1,\theta_2,\theta_3)$ in S' denotes O at coordinates (x_1,x_2,x_3) and in orientation $\theta_2,\theta_3)$. If **i**, **j** and **k** are the orthogonal unit basis vectors of the scene S, ntation (0,0,0) is the object in its canonical orientation and orientation $\theta_2,\theta_3)$ is obtained from it by rotating the object about an axis in the $+\theta_2\mathbf{j}+\theta_3\mathbf{k}$ direction through an angle of $(\theta_1^2+\theta_2^2+\theta_3^2)^{1/2}$. Obstacles are ned in S' by the following condition: $x=(x_1,x_2,x_3,\theta_1,\theta_2,\theta_3)$ is an obstacle point nd only if O in the corresponding configuration intersects a wall in S'. S' is a set of the product space $\mathbf{R}^3\times\mathbf{C}^3$, where $\mathbf{C}=\mathbf{R}$ mod2π. The initial and goal I and G are also subsets of $\mathbf{R}^3\times\mathbf{C}^3$. Since $\mathbf{R}^3\times\mathbf{C}^3$ is periodic and non-lidean, we need to define a straight line and a star shaped set afresh. Let $x=$ $x_2,x_3,\theta_1,\theta_2,\theta_3)$ and $y=(y_1,y_2,y_3,\phi_1,\phi_2,\phi_3)$ be two positions of the object. Also, $\psi_1,\psi_2,\psi_3)$ and ψ be such that starting in orientation $(\theta_1,\theta_2,\theta_3)$, rotating the ct about the unit vector $\psi_1\mathbf{i}+\psi_2\mathbf{j}+\psi_3\mathbf{k}$ through an angle ψ brings it into ntation (ϕ_1,ϕ_2,ϕ_3). Then, a straight line xy joining x and y is the following ametric in t, $t\in[0,1]$:

$$x + t(y_1-x_1, y_2-x_2, y_3-x_3, (\psi+n2\pi)\psi_1, (\psi+n2\pi)\psi_2, (\psi+n2\pi)\psi_3)$$

arbitrary integer n. We say xy is generated by n. A set I in $\mathbf{R}^3\times\mathbf{C}^3$ is star ped about x with respect to B if both the following conditions hold:

(1) $B(x)\subseteq I$

(2) let u be any point in $\mathbf{R}^3\times\mathbf{C}^3$ and let $D=B(u)\cap I$. Then there exists intege n such that for every $y\in D$ and $z\in B(x)$, the line zy generated by n is in I.

With these definitions, it is not hard to verify that Lemma 2.1 and Theorem 2.1 stand in their original form. Although the above definition is not friendly, it says little more than its Euclidean counterpart that we saw earlier.

2.2.5 Remarks

Thus far we have restricted our discussion to cases where the initial set I completely contains an uncertainty ball. This can be a strong restriction. What can be said about the problem when this is not the case? We have yet another version of Theorem 2.1 for some restricted shapes of the initial set I.

A convex set $I\subseteq\mathbf{R}^n$ is *compressible* (to $B(x_1)\cap I$) by a convex set $B\subseteq\mathbf{R}^n$ if

$\exists x_1\in I : \forall x_2\in\mathbf{R}^n : B(x_2)\cap I$ can be *translated* to a position where it is entirely in $B(x_1)\cap I$.

Fig. 2.11 illustrates the property. It shows two rectangles I and B and their

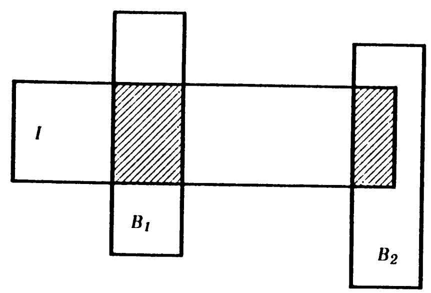

Fig. 2.11. Compressibility. I is compressible by B.

intersections at two different positions B_1 and B_2 of B. It is easy to see that $B_2\cap I$

can be fitted into $B_I \cap I$ and so can any other intersection of I and B. Exactly what pairs of classes of objects satisfy the compressibility property is not known. However, the following merit mention:

(1) If B is spherical then it appears that only spherical regions I satisfy the property.

(2) If B is cuboidal then cuboidal regions I that are axially aligned with B satisfy the property.

(3) Line segments are compressible by all B.

Theorem 2.1'': Let I and G be the initial and final sets for a point object in a three dimensional scene. If I is convex and compressible by the uncertainty ball B to $B(x_I) \cap I$ for some x_I in I, then there exists a fine motion plan from I to G if and only if there exists a path along which $B(x_I) \cap I$ can be translated as a rigid body to a position where it is entirely in G. □

If the velocity error is non-zero, any region of possible positions that is smaller than an uncertainty ball will grow as it is moved around. This means that a straightforward reduction to rigid body motion planning is impossible. Hence, there is no straightforward extension of Theorem 2.1' in this setting.

2.3 FINE MOTION PLANNING WITH DAMPING AND POSITION MEASUREMENT

2.3.1 A Complexity Result

In Example 2.3, we saw that robots with damping can have feasible fine motion plans in situations where robots with just position measurement cannot. Unfortunately, attendant to this increased effectiveness is the increa complexity of deciding the existence of a motion plan.

Theorem 2.2: The fine motion planning problem for point objects in th dimensional scenes and robots with damping and position measuremen PSPACE-hard.

Proof: Given a Turing machine T with a binary tape alphabet that operates i polynomial space bound $S(n)$ and a binary string W, we construct a scene which a fine motion plan exists if and only if T accepts W. The proof borro from [Reif, 1979] and [Joseph and Plantinga, 1985].

The following is an overview of the construction: Externally, the sce appears as shown in Fig. 2.12. The heavy lines are *conduits* – a bunch of tul

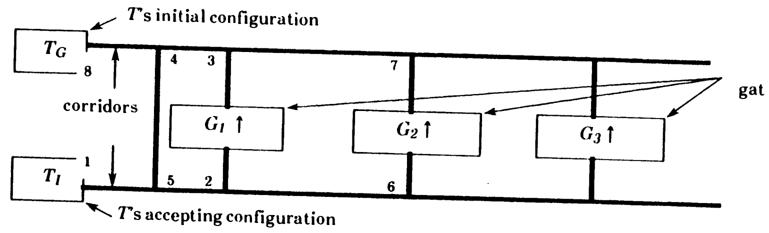

Fig. 2.12. Schematic of the scene (heavy lines represent conduits).

called *channels*. A conduit contains three channels for each tape cell of th Turing machine T and one for each state in its finite control. The error in th position measurement of the robot is chosen to be so large that it is impossible fc the robot to identify which channel of a conduit it occupies. This uncertainty i used to encode the configuration of the Turing machine as the set of channels tha the robot could possibly occupy at any instant. The boxes labelled T_G and T_I i g.2.12 terminate the channels by simply closing them off. Initially the robot is the box labelled T_I and could occupy any of the channels that encode T's cepting configuration. The walls in T_G terminating those channels that encode s initial configuration are made goal surfaces. The aim of the motion plan is to ove the robot from T_I to any of the goal surfaces in T_G. In the course of its ecution, the motion plan may take the robot through any of the gates G_1, G_2... iese gates change the set of possible locations of the robot in a manner nsistent with the transitions of the Turing machine T. Specifically, if T has a ansition from configuration C_0 to C_1, then the scene has a gate such that if the bot enters the gate with its possible locations encoding C_1, then the possible cations on exit from the gate encode C_0. The path labelled 1 through 8 in g.2.12 is indicative of what a motion plan may attempt.

In essence, the constructed problem demands a motion plan that simulates ınning the Turing machine T backwards trying to attain its initial nfiguration from its final configuration. Such a plan exists if and only if T cepts.

We now present the details of the proof. Assume that T has a set of states Q ith a starting state q_0 and one accepting state q_f that is distinct from q_0 and has) transitions out of it. Also, T accepts by printing zeros on all the tape squares ıd entering q_f. The building blocks of our scene are conduits of $3S(|W|)+|Q|$ *ıannels* – three channels for each tape square of T and $|Q|$ channels to represent ıe state of T's finite control. See Fig. 2.13. We say a channel is *active* at any ›int in a motion plan if executing the remainder of the plan translates every ›int in the channel to the goal. We use the activity of the channels to encode T's ›nfiguration, which consists of its tape-contents, the state of its finite control ıd its head position. The three channels for each tape square are used thus: he first channel is active if the tape square contains a 1, the second channel is

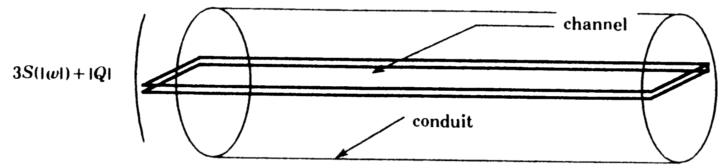

Fig. 2.13. A conduit of channels

active if it contains a 0. The third channel is active if the tape square is currentl scanned by T's head. Of the $|Q|$ state channels, the i^{th} one is active if T's finit control is in state q_i.

The overall layout of the scene is shown in Fig. 2.12. There are two corridoı linked by a series of gates G_1, G_2.... One of the corridors is linked to a termina area T_G containing the goal region G. T_G encodes the initial configuration of by simply blanking out all the channels with walls and designating the walls i the channels that are to be active as the surfaces that make up the goal region C See Fig. 2.14. The other corridor is connected to a terminal area T_I containin

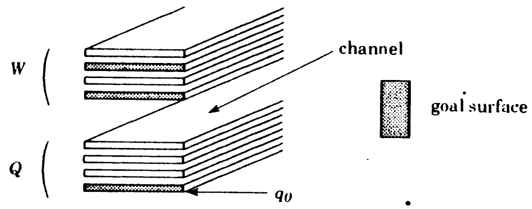

Fig. 2.14. The terminal area T_G

the initial region I. T_I encodes the final configuration of T by blanking out all th channels and designating the blanking walls on those channels that are to bı active as the surfaces that make up the initial region I.

A transition of the Turing machine T is a function from the set of ›nfigurations to the set of configurations and represents a move of T – reading a ›pe square, change of state of the finite control, writing on the tape square and oving the tape head. Each of the transition gates $G_1, G_2 \ldots$ represent a legal ansition of T running on a $S(n)$ tape bound. If $S(|W|) = m$, then there are at ost $2m|Q|$ transitions requiring at most $2m|Q|$ gates. G_i checks to see if the coming set of active channels represents a configuration valid for the ith ansition. If so, it sets the outgoing channels to reflect the transition. We now scribe how these gates are to be built. To implement the transitions of T using ese gates we need only be able to perform logical AND's on the activity of the annels. First we set ε, the error in position measurement, to be larger than any mension in the scene. This ensures that position measurement is insufficient to fer to which channel the observed position corresponds. To compute $C = A \wedge B$, : connect A and B to a diffuser as shown in Fig. 2.15. C is connected to a nozzle

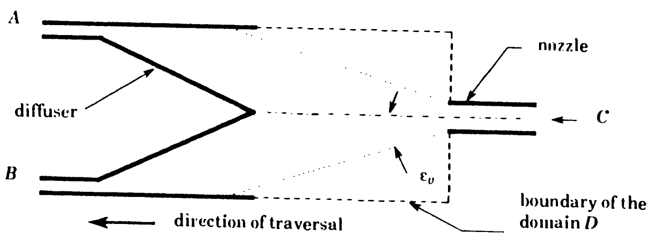

Fig. 2.15. An AND gate

ing the diffuser. The dimensions of the diffuser are chosen with respect to the ocity error ε_v (assumed to be non-zero) so that a point at the mouth of the :zle moved towards the diffuser cannot enter A or B preferentially. As paths side the domain D are not permitted, C can be translated to $A \wedge B$ or not at all, , C can be made active iff both A and B are. Also, the dimension of the nozzle

is chosen with respect to ε_v so that the gate canot be entered at A or B and ex through either of the other two ports. This makes the AND gate unidirectiona that $A \wedge B \Rightarrow C$ without side effects like $C \Rightarrow A$, $C \Rightarrow B$ etc.

If a transition gate G_i is to check to see if T is in state q and whether the h is at square k and reads 1, it performs AND's on the corresponding channels. set the entry channels to reflect the transition, G_i sets the jth entry channel tc the AND of the jth exit channel and the result of the above validity check. course, it resets some channels to reflect the rewriting of tape square k and h movement by T. Fig. 2.16 shows the schematic of a gate that implements transition (q_1, 1 in tape square 1) to (q_2, print 0 in tape square 1, move to t square 2). We note that a transition gate is unidirectional as the AND gates a

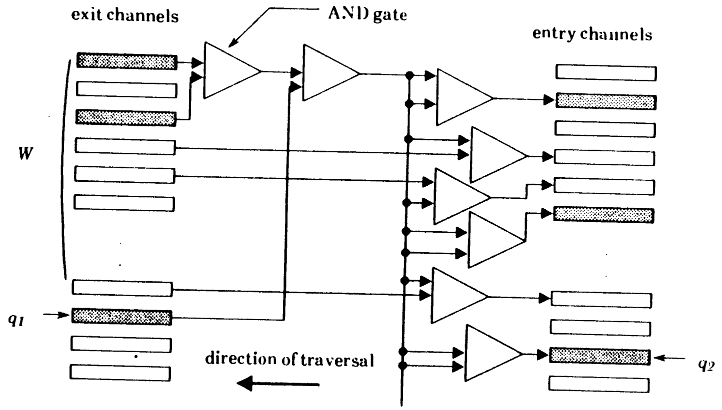

Fig. 2.16. Schematic of a typical transition gate

Hence the gates cannot be traversed the wrong way to realize illegal transition

Recall that the error in position measurement ε is set to be larger than an dimension in the scene. Hence position measurement cannot be used t electively enter a transition gate. To make selective entry of gates posssible, the ates are arranged in a cascade. At the i^{th} level in the cascade, a motion plan has ɔ choose between entering the i^{th} transition gate and continuing on to gates $i+1$, $+2, \ldots$, as shown in Fig. 2.17. Finally, we set the friction coefficient of the walls

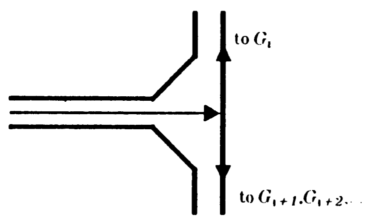

Fig. 2.17. Cascading the gates.

ɔ be large enough that the friction cone angle is bigger than the directional elocity error ε_v.

Let S_I and S_G be two activity assignments to the channels encoding two valid onfigurations C_I and C_G of T.

laim 2.1: There exists a fine motion plan translating S_I to S_G iff T started in C_G ttains C_I.

roof: In the following, we refer to a motion plan by the sequence of gates it raverses. The length of a plan is the length of the sequence. We proceed by nduction on the plan-length.

asis plans of length zero; immediate.

nduction Assume true for for plans of length k. Let $P = G_{P_I} P'$ be a plan of length $+1$, where P' is a plan of length k and G_{P_I} is the first gate traversed by P. Let $S_{I'}$ be the result of applying the plan G_{P_I} to S_I. By the correctness of the gates, S_I an be translated to $S_{I'}$ iff T started in $C_{I'}$ attains C_I. By the inductive hypothesis,

$S_{I'}$ can be traslated to S_G by P' iff T started in C_G attains $C_{I'}$. It follows that S_I be translated to S_G by a path of length $k+1$ iff T started in C_G attains C_I. □

From the above claim it follows that there exists a fine motion plan from I t iff T accepts W. □

2.3.2 Remarks

Remark

In the proof of Theorem 2.2, a non-zero directional velocity error ε_v was key the construction of the AND gates. Even if ε_v is zero, AND gates can constructed provided the range of velocity directions available to the effector discrete. In particular, we exhibit an AND gate for a cartesian robot - one th can command velocities only along three-fixed and mutually orthogonal axe See Fig. 2.18. The construction is similar to that of Fig. 2.15. As shown, tl

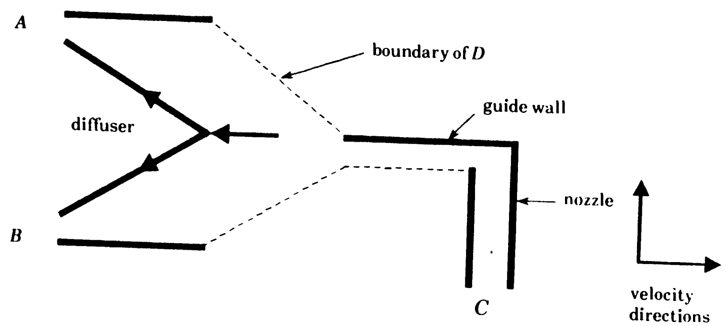

Fig 2.18. An AND gate for a cartesian robot.

nozzle is incident on a guide wall. All the walls are frictionless so that the only way to exit the nozzle is to slide along the guide wall which is positioned directly opposite the apex of the diffuser. Assuming that the effector striking the apex es not stick but enters either A or B, we see that C can be active iff both A and B e, obtaining the desired relationship. It is easy to see that this version of the ND gate is unidirectional as well.

Since our proof of Theorem 2.2 only required motions in orthogonal directions, holds for cartesian robots with zero directional velocity error as well, using the ove AND gates.

mark

Our construction of Theorem 2.2 utilized an initial set I that was made up of a umber of disconnected components. This is not an essential ingredient of the oof. The proof goes through even for a single-point initial set, utilizing scaded AND gates to generate the final configuration of the Turing machine T. g. 2.19 illustrates the idea.

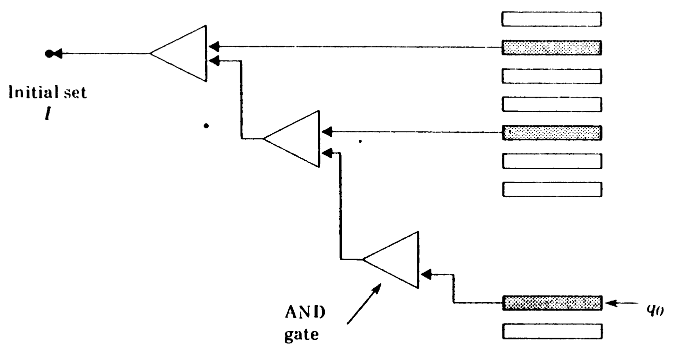

Fig 2.19. Generating configurations from single-point initial sets.

2.3.3 More Complexity Results

Our next result is mainly of technical interest. Having shown the existence problem in Theorem 2.2 to be PSPACE-hard, we would like to show at least a restricted version of the problem to be PSPACE-complete. If we can eliminate much of the haze and show a pared down version of the problem to be PSPACE-complete, we can then inquire into the reason for the jump in complexity between the problems for robots with and without damping. We shall proceed with an algorithm for fine motion planning in rectilinear scenes. First, some definitions. A three dimensional scene is *rectilinear* if

(1) All the walls in the scene are rectangular planes of zero thickness.

(2) The scene has a fixed orthogonal reference frame and each wall has at least one of its edges parallel to a reference axis.

(3) No two walls intersect, except on their boundaries.

It is easy to verify that our construction in Theorem 2.2 is a rectilinear scene. A scene S is *fine* with respect to a robot if there exists a point x in S such that $S \subseteq B(x)$, the uncertainty ball around x. In other words, S can be entirely contained by an uncertainty ball. Because of our lower bound condition on the time-out period τ for damped robots, a move in a fine scene S that commences at a point in S will terminate in S if and only if the effector sticks on a wall in S. Hence goal regions in fine scenes must be wall surfaces to be attainable.

The 'Pacman' Problem

Input: Given are a damped cartesian robot with zero directional velocity error, a fine rectilinear scene in three dimensions with reference frame aligned with the robot axes, a single-point initial set I and a goal set G for a point object.

<u>Property:</u> Is there a fine motion plan with damping and position measurement from I to G in S?

Fig. 2.20 gives a non-deterministic algorithm for the above problem.

Algorithm 2.1

start:

  $P := I$; /* set of possible positions of the point object */

  If $(P \subseteq G)$ then halt and report success;

  $d :=$ pick a direction of the possible six of the cartesian robot.

  if d was picked twice in a row then halt and report failure;

  simulate move in direction d;

  $P' :=$ new set of possible locations for the point object;

  if $P' \not\subseteq S$ then halt and report failure;

    else $P := P'$; goto *start*;

Fig 2.20 Algorithm for the 'pacman' problem

Claim 2: Algorithm 2.1 can be implemented in deterministic polynomial space.

Proof: In order to prove the above claim, we prove two subclaims:

(1) P can be never have more than polynomially (with respect to the number of vertices in the scene S) many points in it. Furthermore, these points need only be represented to the same precision as the vertices in S.

(2) Computing P' from P after each move takes polynomial space.

Proof of subclaim (1): Let X, Y and Z be the reference axes of the rectilinear scene and let the input be specified as coordinates in this frame. Consider the set

$$S_X = \{x \mid \exists \text{ an edge in } S \text{ parallel to } Y \text{ or } Z \text{ with } X\text{-coordinate } = x\} \cup \{x \mid (x,y,z) \in I\}$$
$$\cup \{x \mid \exists \text{ an edge along which two planes meet with } X\text{-coordinate } = x\}$$

and similarly S_Y and S_Z. Clearly there at most $O(n^2)$ points in each of the sets where n is the number of vertices in the scene S. Whence it follows that

$$|S_X \times S_Y \times S_Z| = O(n^6).$$

It is appropriate to point out that it is insufficient to refer to a point in S_{XYZ} by its coordinates alone. If the point is on a plane, it is necessary to know on which side of the plane, for instance. In general, a point could lie on an edge along which many planes meet. We assign a unique number to each plane and a sign say + and -, to refer to the two sides of a plane. Then, to locate a point we need its coordinates, the planes on whose surfaces it lies and the signs (+,-) for those surfaces. Since we need only three planes to uniquely identify a point representing this information takes $O(\log n)$ space per point and hence requires no change in the argument. The cardinality of the set of possible locations needs no adjustment on this account as we are considering the walls pairwise anyway.

We now show that $S_{XYZ} = S_X \times S_Y \times S_Z$ is closed under cartesian moves in the sense that applying a cartesian move to an element of S_{XYZ} will translate it to a set of points in S_{XYZ} or move it outside the scene S. Let (x_1, y_1, z_1) be a point (see Appendix A in this regard) in S_{XYZ} and let (x_2, y_2, z_2) be reached from (x_1, y_1, z_1) in a single cartesian move along direction d. Without loss of generality assume d is in the $+X$ direction. Then, (x_2, y_2, z_2) must lie on a wall normal to the x axis or on an edge along which two planes meet. In either case $x_2 \in S_X$. Let y be the Y-coordinate of the last edge parallel to the Z direction that was traversed (made contact with) in the move. Then $y_2 = y$ clearly. If no such edge was traversed, $y = y_1$. In either case $y_2 \in S_Y$. Similarly z_2 is in S_Z. Whence it follows that $(x_2, y_2, z_2) \in S_{XYZ}$. The argument is illustrated in Fig. 2.21, where the point is 'split' and the X-coordinate of both of the resultant points is left unaltered but the Z-coordinate of one is changed by deflection against the oblique wall in the path.

Proof of subclaim (2): We must keep in mind here that a point p in P anslated in direction d could 'split' by striking an edge as shown in Fig. 2.21. At

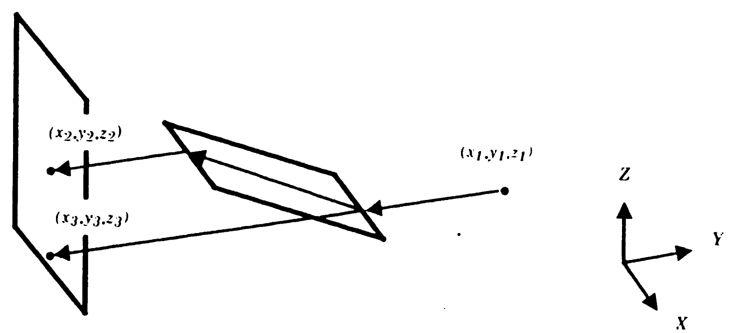

Fig 2.21 Computing moves in rectilinear scenes.

ch split, we can identify one of the resultant points with p, so that we can say at each $p \in P$ is split at most $O(n)$ times - once for each wall in S – before it sticks id thereby becomes an element of P' or leaves the scene S. Consequently, we n recursively compute P' from P with the depth of the recursion being $O(n)$ as own in Fig. 2.22.

early, computing P' from P as shown above takes polynomial space. In fact, the ne marked (*) in Fig.2.22 takes polynomial time in a straightforward nplementation.

From the two subclaims it follows that Algorithm 2.1 uses polynomial space id hence can be implemented in non-deterministic polynomial space. Since PSPACE = PSPACE by Savitch's thorem [Hopcroft and Ullman, 1979], it llows that Algorithm 2.1 can be implemented in deterministic polynomial ace. □

```
for each p∈P do
      Move(p,d);
  od;
  procedure Move(p:point;d:direction);
      /* simulate moving a point in direction d */
  (*) simulate moving p in direction d
      if p sticks then P' := P'∪{p};
      if p hits an edge then let p split into p1 and p2;
                  Move(p1,d);
                  Move(p2,d);
      else p goes outside S; abort and report failure;
  end;{Move}
```

Fig. 2.22 Computing moves for the 'pacman' problem.

On the basis of Claim 2.2 and Theorem 2.2 we can now state

Theorem 2,3: The 'Pacman' problem is PSPACE-complete.

Proof: By Theorem 2.2, Remarks on Theorem 2.2, and Claim 2.2. □

We now have a very restricted version of the existence question for fin motion plans that is PSPACE-complete. This will permit us to inquire into th reasons that contribute to the complexity of the problem. The key to th complexity is the number of 'degrees of freedom' of the object – the number possible locations for the object at any point in a motion plan. If this were to b bounded by some constant, the problem can be solved in polynomial time. T illustrate this we look at a modified version of the problem of Algorithm 2.1 an exhibit a polynomial time algorithm for it.

The Restricted "Pacman" Problem

Input: Given are a damped cartesian robot with zero directional velocity error, a fine rectilinear scene in three dimensions with reference frame aligned with the robot axes, a single point initial set I and a goal set G for a point object.

Property: Is there a fine motion plan with damping and position measurement from I to G such that at any point in the plan the object has at most k (for fixed k) possible locations?

Define a configuration to be the set of possible locations of the object at any instant. There are at most

$$\prod_{i=1}^{k} \binom{n^6}{i} = O(n^{6k^2})$$

onfigurations by our arguments in Claim 2.2. We now construct a graph with ne vertex per configuration. We place a directed edge between two vertices u nd v if one of the six directional moves of the robot translates u to v. For each ertex u we can find the outgoing edges using the procedure of Fig. 2.23. This rocedure takes polynomial time as at worst P can run through all the onfigurations, since no configuration can be repeated in a simulation as the oordinates of the points in P are non-decreasing in direction d. We now label a ertex a 'goal vertex' if every point in the corresponding configuration is in the oal set G. We then search the graph for a path from the vertex representing I to ny goal vertex. Clearly, there exists a motion plan from I to G that has no more an k possible locations for the object at any instant iff there is such a path.

Since the construction of the graph, the labelling and the searching of the raph are all feasible in polynomial time, Algorithm 2.2 can be implemented in lynomial time and space polynomial in $k \log n$.


```
Algorithm 2.2
for each direction d do
    P := u;        /* P is the set of possible locations */
    v := Φ;
    while P ≠ Φ do
        simultaneously simulate moving all points in P in direction d until some point p in P
        does one of
                (a)sticks
                (b)hits an edge
                (c)goes outside S;
        if (a) then v := v∪{p};
        if (b) then let p split into p1 and p2;
                    P := (P-{p})∪{p1}∪{p2};
        if (c) then P := Φ; v := Φ;
    od
    draw an edge from u to v;
od
```

Fig. 2.23 Constructing the graph for the 'restricted pacman' problem

2.4. SUMMARY

We began with fine motion planning for robots with position measuremen and showed that if the initial region contains an uncertainty ball, we can reduc the problem in polynomial time to motion planning for a rigid object. Since th latter is decidable in polynomial time, the former is as well. We then looked a fine motion planning for a point object in three dimensions and robots wit damping and showed a general class of problems to be PSPACE-hard. Discardin the unnecessary ingredients in the problem, we identified a seemingly innocuous class that is PSPACE-complete. Restricting this class to motion plans with a constant bound on the number of possible locations for the object reduced the ımplexity of the problem to polynomial time. From this we conclude that the ımber of disconnected regions in the set of possible locations for the object is the ey contributor to the complexity of the problem. A practical algorithm for fine otion planning should only look for solutions that obey a bound in this regard.

CHAPTER 3

AN ALGORITHMIC APPROACH TO THE AUTOMATED DESIGN OF PARTS ORIENTERS

3.1. INTRODUCTION

3.1.1 Motivation

Robots are seeing increasing use in assembly tasks. The typical assembl task of mating two parts follows the following paradigm; the robot picks up th first component and positions it at a convenient location. It then picks up th second component and mates it with the first. The problem is that today's robo cannot adjust well to variations in the position and orientation of the part when i attempts to grasp it. Consequently, it is necessary to feed the part to the robot iı specific orientations. Typically, this requires a parts orienter, which is a 'blacl box' that accepts the part in any orientation and outputs it in som predetermined orientation. Currently, parts orienters are designed by the 'pape and pencil' method and may require many hours of work. In the CAD/CAN environment, where the design of a simple part can be completed in a matter o minutes, it is desirable that the complete assembly process for the part be determined in about the same time. Hence, there is a need to automate the design of parts orienters for simple parts and integrate the design algorithm with the CAD/CAM system. What would such an algorithm look like? It should take as inputs

(a) Geometric information on the part, such as is typically resident in a CAD/CAM database,

(b) The set of orientations in which the part may enter the orienter and the set of orientations in which it should exit,

nd should output a suitable feeder for the part, if such exists within the design ·amework. If none exists, the algorithm should perhaps suggest modifications to ıe part which would permit such a feeder. Hopefully, such modifications will not lter the functional nature of the part.

The above problem is one of real-world interest and in many ways is similar to ıe problem of building compilers for programming languages faced in the fifties nd the sixties. Then, little was known about the theory of compilers or context ·ee grammars and the first compilers were hand-crafted around the language. Vith the development of a theory, it was possible to identify the class of ınguages that permitted simple and elegant compilers. Today, it is straight ›rward to generate compilers automatically for this class, eliminating many ours of hard labour. There is reason to believe that a similar situation exists in ›botics and that it is necessary to build abstractions and theory for what are enerally deemed as 'hard but gritty' real world problems. For example, it would e useful to identify the class of parts for which the design of orienters can be asily automated.

This paper attempts to abstract and analyze the design of parts orienters. The bstracted problems are not only interesting from a theoretical viewpoint, but lso provide useful insight into the physical domain.

3.1.2 The Problem Statement

It is elegant and intuitively helpful to define the problem of parts orienter esign as follows:

Given an object O and two sets of orientations I and G for it, design a set obstacles S that forces the object to move from any orientation in I to so orientation in G.

This set of obstacles S is the 'black box' orienter that will accept O in a orientation in I and output it in some orientation in G. Notice that this definiti is in some sense the inverse of the motion planning problem:

Given an object O and two sets of configurations I and G for it in a scene S obstacles, are there paths for the object from every configuration in I to sor configuration in G?

The difficulty with the above definition of the parts orienter design problem that it does not completely capture the physics involved. If the obstacles to l designed are static, how can they force the part to move? Tacit here is sorr external force that drives the part and without being precise about the exa nature of this driving force, it is impossible to proceed. Hence, it is necessary t study different paradigms for parts orienters and attempt to devise automatin algorithms for each of these paradigms. Towards this end, we classify part orienters as they are in practice, [Boothroyd et al, 1982, Philip, 1980].

3.1.3 Classifying Parts Orienters

Fig. 3.1 shows a heirarchical classification of parts orienters. The root of th tree is labelled *parts handlers* and includes any and all devices that may be usec to manipulate parts. This general class is subdivided into *passive handlers* anc *manipulators*. Manipulators are devices that explicitly grip the object that they handle. Typical examples of this class are robots and remote handlers for toxics. Passive handlers are devices that do not grip the object they handle. Examples of this class are conveyor belts, vibratory feeders, etc. The distinction between the

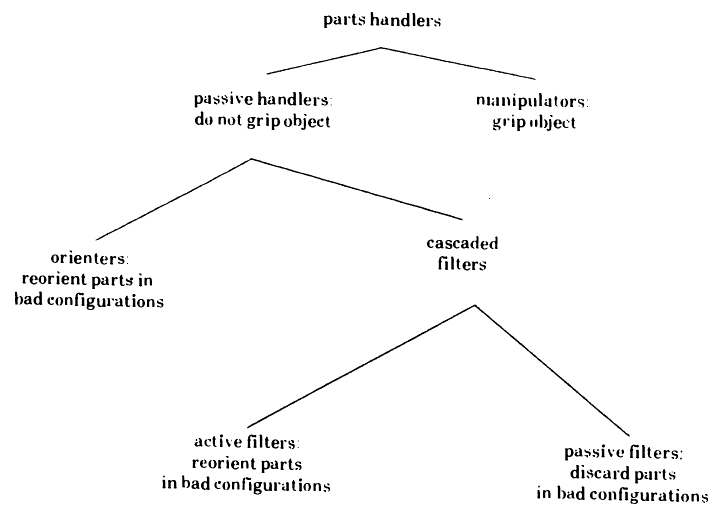

Fig. 3.1. Classifying parts orienters

› classes is not rigorous as one could well consider devices that reduce the nber of degrees of freedom of the object handled but do not constrain them irely.

[n this thesis we are primarily interested in passive handlers as ›processors" to the much studied class of manipulators, [Udupa, 1977, Lozano-ez and Wesley, 1979, Reif, 1979, Schwartz and Sharir, 1982]. Passive dlers can be further subdivided into *orienters* and *cascaded filters*. Orienters best described as devices that accept the object to be handled in any orientation and output it in the desired orientation. Cascaded filters are, as the name suggests, a sequence of filters. A filter is a three port device that accept the object in some set of orientations C_i and delivers it in another set o orientations C_o, possibly discarding the object if it enters the filter in some set o orientations C_d. Fig. 3.2 is a schematic of a filter. C_i will in general not be the

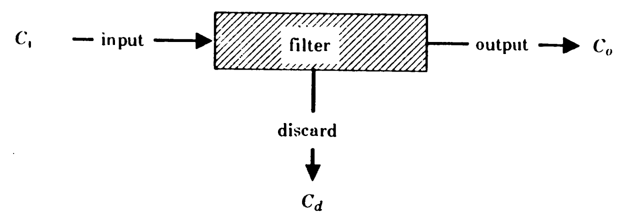

Fig. 3.2. Schematic of a filter

set of all orietations C_u and if the object is fed to the filter in a orientation in C_u–C_i the filter gets 'stuck' and its effect on the part is undefined. If the object traversing the filter starting in every orientation in C_i is either discarded or allowed to exit the filter without reorientation, then the filter is a *passive filter*. Otherwise, the filter is an *active filter*. Typically, cascaded filters are configured as shown in Fig. 3.3. Parts are fed to the filter sequence from a randomizing bin so that they occur in all configurations and the discarded parts from each filter are returned to the bin. Most vibratory feeders follow this paradigm, [Boothroyd et al, 1982].

The main difference between orienters and cascaded filters is that filters enjoy the additional freedom of feedback to recycle parts that end up in bad orientations. The distinction is not rigorous as, for example, a cascaded filter

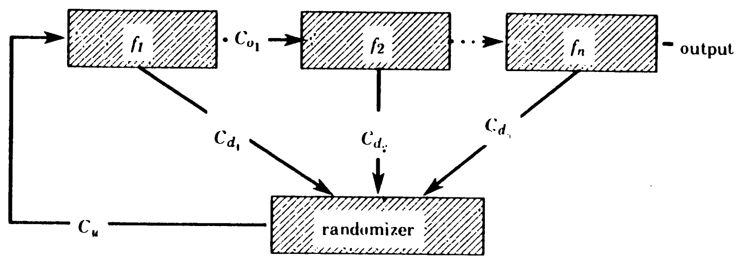

Fig. 3.3. Schematic of a cascaded filter

onsisting entirely of active filters that do not discard any orientations ($C_d = \phi$) s also an orienter. Cascaded filters are most useful in orienting small parts that re to be handled in large quantities, while orienters are suitable for few-of-a-ind parts, large parts, and task driven situations like those that a household obot might face.

3.1.4. Overview of Results

Having classified parts orienters in a general way, we proceed by studying articular paradigms for the two classes of passive handlers - orienters and lters. In the orienters class, we consider two paradigms - the belt and the anhandler. The belt is for two dimensional polygonal objects that are infinite in ie third dimension while the panhandler is for flat polygonal objects. In the ascaded filters class, we show how to automate the design of vibratory feeders for at, polygonal objects. In each case, the physical problem is first made precise by aking suitable assumptions and then abstracted to a computational problem.

The abstracted problems bring up interesting questions in graph pebbling and selective function composition on finite sets. We restrict our attention to polynomial time algorithms to these problems without regard to exponents or constants.

3.2 PARADIGMS FOR ORIENTERS

3.2.1 The Belt

Consider a two dimensional belt (infinite in the third dimension) as shown in Fig. 3.4. The belt moves from left to right and has vertical jumps at various

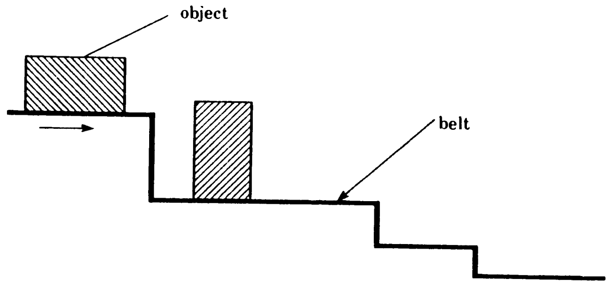

Fig. 3.4. The belt paradigm

points. The two dimensional convex polygonal object (infinite in the third dimension) is placed on the left end of the belt in some orientation. As the object traverses a jump, it could be reoriented depending on the height of the jump and the orientation in which the object approached the jump. The aim is to pick a sequence of jumps so that regardless of the orientation in which the object starts on the belt, it always attains the same predetermined orientation at the end of e belt. This paradigm could be useful for orienting large objects like say, rtons that are to be oriented before being unpacked by a robot.

The Problem Statement and Abstraction

We consider an n-sided convex polygonal object O and label the faces 1 ough n. The object is in orientation i if it lies on face i. Referring to Fig. 3.5,

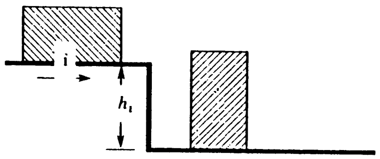

Fig. 3.5. The transition height h_i

define the *transition height* h_i of orientation i to be the minimum jump height which a change in orientation occurs when the object aproaches the jump in entation i. We assume that the transitions in the above definition result in que orientations. Furthermore, if the object in orientation i is subject to a p of height much greater than h_i, it will tend to tumble before attaining a ble configuration. Since such behaviour is unpredictable and inconsistent, we l disallow such possibilities by placing the restriction that if the object can be rientation i, it cannot be subject to a jump greater than h_i. We are now ready tate the problem.

Problem 3.1

Given a convex, n-sided polygonal, two dimensional object and the transi height for each orientation of the object, find a sequence of jumps so that object started on the belt in any orientation always attains s predetermined orientation at the end of the belt.

We assume that the transition heights of the object are given as they are strongly dependent on the physical parameters of the problem to be within scope of this paper. They might be determined by n simple experiment computations. Define the transition diagram of an object to be a graph with vertex per orientation of the object. An edge (u,v) with label t is present in diagram if and only if orientation u when subject to its transition height of t g to orientation v. Since the transitions are unique, the graph is a collectio cycles with zero or more trees attached by the root to each cycle. We can state the abstract version of the problem as a graph pebbling game on transition diagram.

Problem 3,1'

Pebbling Game

start : All vertices are pebbled.

move: Of all pebbled vertices, pick those with minimum labels on their respective outgoing edges. Move the pebbles along these edges.

Question

Find a sequence of moves that minimizes the number of pebbled vertices.

Fig. 3.6 is a simple algorithm to solve the problem. Essentially, the algorit considers the set of pebbled vertices of the graph and plays the pebbling game a predetermined number of moves. When the algorithm terminates, the set of

Algorithm 3.1

```
S ← {1,2,...n};        /* set of pebbled vertices */
σ ← ε;                 /* move sequence, initially null */
for i ← 1,2,...,m do
    h ← min{h_k|k is in S}; /* find minimum label on edges out of pebbled vertices */
    S ← h(S);          /* move pebbles along minimum label edges */
    σ ← σ·h;           /* update move sequence  */
od
```

Fig. 3.6 The pebbling algorithm

bbled vertices is of minimum size in the sense that no other sequence of jumps oduces a smaller set. The interesting question here is how big the sequence ngth m need be.

aim 3.1: The sequence length m is $\Theta(n^2)$.

oof: We first show that m can attain n^2.

Subclaim: The sequence length m attains n^2.

Proof: Consider the transition diagram of Fig. 3.7. It is clear that any

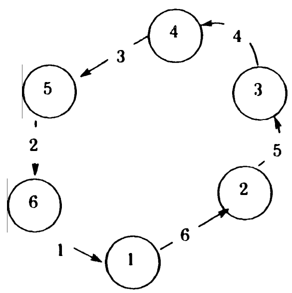

Fig. 3.7. Worst case transition diagram

sequence of jumps producing the minimum set of pebbled vertices for thi diagram is of length at least $1+2+3+4+5+6 = n(n+1)/2$ for $n=6$. Sinc there exists such a diagram for every n, we conclude that m is $\Omega(n^2)$. Not that we have said nothing about the objects that generates this family o diagrams and cannot do so as as we do not have sufficient knowledge of th physics. □

Next we show that m is $O(n^2)$. We say that an edge (u,v) with label t of th transition diagram is *used* by a move if u was selected by the move (i.e, befor the move u was a pebbled vertex and the label value t was a minimum).

Subclaim: Every edge in the transition diagram is used $O(n)$ times when th minimum set of pebbled vertices is attained.

Proof: There are two kinds of edges in a transition diagram. Those that ar in a cycle and those that are not. Let (u,v) be an edge that is not on a cycle. Clearly, (u,v) can be used only as many times as there are vertices z such tha there exists a path from z to u. Since there can be at most n such, the claim follows.

Now let (u,v) be an edge on a cycle. We work the proof in stages. Suppose the transition diagram consists entirely of a single cycle. It is not hard to see that when any edge of the cycle is used n times, every edge of the cycle must already have been used, including those edges on the cycle with maximum label value. Consider the set of edges on the cycle with maximum label value. It is straightforward to verify that the first time any of these edges are used, all of them will be used and that the minimum set of pebbled vertices is attained. Since the set of pebbled vertices can never grow, any further moves will maintain the minimality of the set.

Now extend the diagram to consist of a single cycle with some trees attached to it. Let t_m be the maximum label value of the edges on the cycle and let $U = \{(u_i,v_i)\}$ be the edges on the cycle with this label value. Also, let $R = \{(r_i,s_i)\}$ be the tree edges with a label value of t_m. Notice that either every edge in U is used at any move or none of them are used, and that if any edge of R is used then every edge of U is used. Also, edges with label value greater than t_m will never be used. Now, the first time an edge from U is used, the set of pebbled vertices is

$\{u \mid (u,v) \text{ has label } \geq t_m\}$.

Furthermore, no edge in the diagram has been used more than n times. Call a tree *active* if the set

$\{u \mid u \text{ is a pebbled vertex in the tree and the edge out of } u \text{ has label} \leq t_m\}$

is nonempty. Notice that trees that are not active cannot be reduced further. Between two succesive uses of edges in U, the number of pebbled vertices on each active tree decreases by one and the cycle edges between two edges (u_1,v_1) and (u_2,v_2) in U (as shown in Fig. 3.8) are each used no more than once

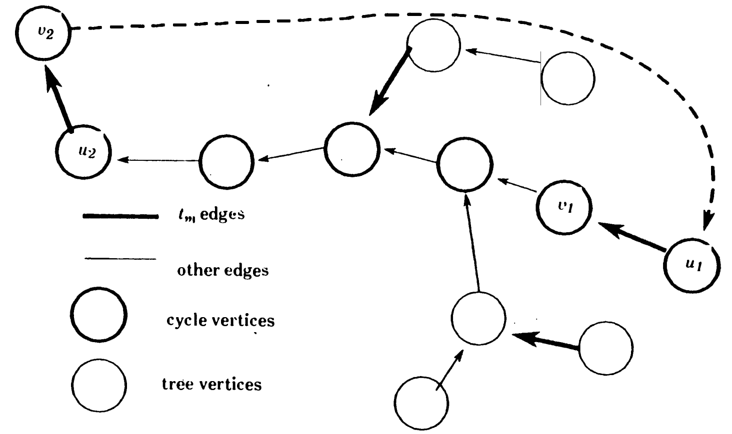

Fig. 3.8. The segment between cycle edges of maximum label value t_m.

for each active tree attached to the segment $u_1,v_1,\ldots,u_2,v_2$. Hence after C uses of the cycle edges, all the trees will be inactive and the minimim se pebbled vertices attained. This implies that no edge is used more than times before the minimum set of pebbled vertices is attained.

Now consider the transition diagram with multiple cycles and trees. Let

$$t_m = \min_{C_i \text{ is a cycle}} \; \max_{(u,v) \text{ in } C_i} \{\text{label}(u,v)\}.$$

Clearly, t_m is the maximium edge label that can be used. Pick any cycle in t diagram with an edge labelled t_m. Now the arguments made above apply this cycle. The first time an edge labelled t_m is used, no edge could have be be used more than n times. After that, each edge on the cycle is used once each vertex in an attached tree on this or some other cycle of the diagram.

The two subclaims completes our proof. □

With the above discussion, we can tighten the loop guards of Algorithm 3.1 obtain Algorithm 3.1' of Fig. 3.9.

Remarks

We considered unidirectional belts in the above treatment. It might be interest to consider bidirectional belts, where the object could be tranferred fro a belt moving right to a belt moving left and vice versa. See Fig. 3.10. Althoug this does not change the worst case jump sequence length, it might b significantly better in some cases.

An average case analysis of the algorithm might prove it to be better tha $O(n^2)$. In itself, an $O(n^2)$ bound is not very encouraging.

We assumed that the transitions at any jump were single valued. It is mor realistic and perhaps more interesting to relax this restriction.

Algorithm 3.1'

```
t_m ←        min            max        {label(u,v)};
        C_i is a cycle   (u,v) in C_i
σ ← ε;          /* move sequence, initially null */
S_m ← {u|(u,v) is a cycle edge with label t_m}∪
          {u|(u,v) is a tree edge with label ≥ t_m};  /* minimal set of pebbled vertices */
S ← {1,2,...n};      /* set of pebbled vertices */
  while |S| ≠ |S_m| do
     h ← min{h_k|k is in S};
     S ← h(S);          /* move pebbles along minimum label edges */
     σ ← σ·h;           /* update move sequence */
  od
```

Fig. 3.9. An improved pebbling algorithm

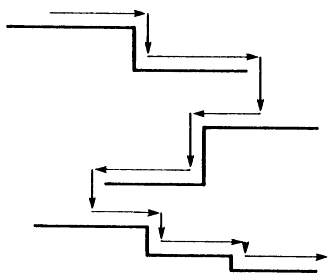

Fig. 3.10. A bidirectional belt.

3.2.2 The Pan Handler

Here we consider the problem of orienting planar rigid polygonal objects in k driven situations. i.e. the same orienter is to be used to handle a variety of such parts. The orienter consists of a planar regular polygonal tray with a lip a shown in Fig. 3.11. Suppose the object is dropped onto the tray and is in som

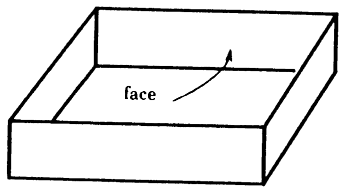

Fig. 3.11. The orienting tray

possible set of orientations against a certain face of the tray. If the tray is tilted the object leaves this face and strikes another face depending on the tilt direction possibly reorienting itself when it attains a stable configuration against the nev face. The problem is to choose a sequence of tilt directions that will reorient th object in some unique orientation against a predetermined face of the tray, if sucl a sequence exists. This scheme was first proposed in [Grossman and Blasgen 1975] and more recently discussed in [Mason and Erdmann, 1986]. The paradigm might be useful for household robots that could walk around with trays so that when one drops say, a polygonal coffee mug in the tray, the robot walks off tilting the tray until it has uniquely oriented the mug and picked it up.

Problem Statement and Abstraction

Before proceeding with the problem we need to make some assumptions and place some restrictions to state the problem more precisely. Our assumptions are:

(a) The object is allowed to contact only one face of the tray at a time.

(b) One and only one edge of the object makes contact with any face of the tray in any orientation. To support this, we assume the object to be convex. This is without loss of generality as we can always consider the convex hull of a non-convex object.

(c) The object always remains roughly in the middle of a face at any time. In light of assumption (e), this is equivalent to the tray being large compared to the object.

(d) All operations are planar in that the object is always dropped on a distinguished face of its two planar faces and it lies on that face throughout.

(e) The tray can only be tilted along median directions - lines joining midpoints of two faces as shown in Fig. 3.12. This is to discretize the set of tilt

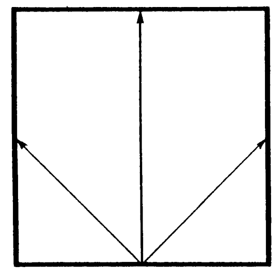

Fig. 3.12. Median directions

directions for two reasons: A fixed and finite set of tilt directions does not require a full freedom wrist to tilt the tray but can be achieved with a fixed and finite set of binary actuators. Secondly, as we shall show later, the problem is not as interesting if any tilt direction is attainable.

(f) The faces of the tray have a high coefficient of friction while the tray flo is smooth. This allows the object to rotate about the vertex of first conta without sliding when the tray is tilted. See Fig. 3.13.

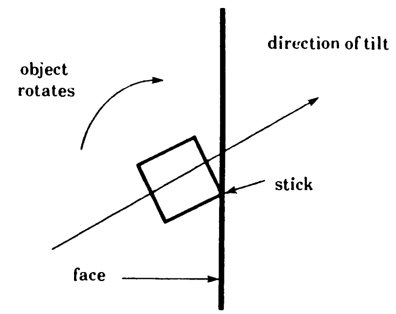

Fig. 3.13. Rotation about the vertex of first contact

(g) The tray is never tilted so steeply that the object tumbles, i.e. rotates pa its first stable configuration when it makes contact with a face, as shown i Fig. 3.13.

(h) The transitions are single valued in that if the object is in a certai orientation and the tray is tilted in a median direction, the resultin orientation is uniquely known.

(h) Initially, the object can be in any allowable configuration against sor predetermined face.

We are now ready to state the problem:

Problem 3.2

Given a k-sided regular polygonal tray and a n-sided convex planar polygon object, find a sequence of tilt directions that reduces the possible orientations t any single orientation under the above assumptions, if there exists such.

Since the tray is regular, we need not distinguish between the object lying in the same orientation against two different faces. Also, the effect of each tilt direction is independent of the face to within an isomorphism. We number the faces in say anti-clockwise order from any face so as to give the median directions with respect to that face a unique labelling. Fig. 3.12 shows the labels for the median directions with respect to face 4. Hereafter, we refer to the median directions by these labels so that the j^{th} direction corresponds to the j^{th} face in this ordering. Labelling the sides of the object in cyclic order (say anti-clockwise), we say that the object is in orientation i if it lies on side i. Corresponding to this labelling, we define S to be the set of n orientations of the object. We can now express the effect of median tilts as k functions $f_1, f_2, \ldots, f_k$ from S to S and state the abstract version of the problem in terms of these mappings.

Problem 3.2'

Given k mappings $f_1, f_2, \ldots, f_k : S \to S$ on a finite set S, find a mapping f_0 that is a composite of the f_i's such that f_0 is a constant function on S, i.e.,

$|f_0(S)| = |\{s \mid s = f_0(v) \text{ for some } v \text{ in } S\}| = 1.$

Fig. 3.14 gives an algorithm to solve the abstracted problem above. The algorithm is greedy and makes no attempt to be efficient or provide an optimal solution. Starting with the identity function, at each stage the algorithm attempts to find a composite function with range at least one less than the previous stage. To do this, it picks a pair of elements from the range of the previous stage, and looks for a sequence of function compositions that merges these two. It then composes this sequence with the function of the previous stage

Algorithm 3.2

```
  (1) determine the transitions for each face and each median direction.
  S←{1,2,...n};        /* possible orientations */
  f0←ε;       /* identity mapping */
  While |S| > 1 do
      (2) pick i and j in S such that there exists a sequence of function compositions s suc
          that s(i) = s(j). If no such exists, halt and report failure.
      (3) S←s(S);    /* update S */
          f0←s∘f0;        /* update composite sequence */
  od
```

Fig. 3.14. The function composition algorithm

to obtain the new function. If no such pair of elements exists it halts and repo failure.

The algorithm works because the problem satisfies the inclusion property: there exists $s \in S$ and composite function $f_0 : S \to s$, then for any subset $S' \subseteq S$ the exists a composite function f such that $f(S') = s$ (f_0 is trivially such a functio Consequently, the greedy algorithm can reduce the range set whenever a however it can, without ever getting into some set of elements that cannot reduced, although there might have been some other reducing sequence.

Time Analysis

Step (1) can be computed in $O(nk)$ time or by $O(nk)$ simple experiments.

Step (2) can be computed in $O(kn^2)$ time. To see this, notice that we c construct a transition diagram for pairs of elements in S. Specifically, this i directed graph $G = (V, E)$ where

$V = \{(i,j) \mid i,j \in S\}$

$E = \{(i_1 j_1) \to (i_2 j_2) \mid$ if $f_i(i_1) = i_2$ and $f_i(j_1) = j_2$ for some $1 \le i \le k\}$

label$((i_1 j_1) \to (i_2 j_2)) = i$, if $f_i(i_1) = i_2$ and $f_i(j_1) = j_2$ for some $1 \le i \le k$.

ery vertex has outdegree k and there are n^2 vertices and hence at most n^2k ges. Now, a depth-first search of this graph yields a path from a vertex $(i_1 j_1)$ to $j_2)$ where $i_1 \neq j_1$ and $i_2 = j_2$. The labels on this path form the composition quence s. The depth first search requires $O(kn^2)$ time and so does constructing e graph, although this need be done only once.

Step(3) takes $O(kn^3)$ as it requires applying the $O(kn^2)$ sequence to the set S iich is of size $O(n)$.

Since the algorithm can run for at most n iterations, the total running time is $kn^4)$ and the length of the sequence generated by the algorithm is $O(kn^3)$.

A Modified Problem

The above, assumed that the object can be in any orientation against a rticular face at the start of the tilt sequence (assumption (h)). Since all ientations of the object may not be simultaneously stable, this assumption is a t too strong. In particular, consider the case where the object starts in some ven set of orientations against a particular face. Although this is a relatively mple extension, the *abstract* problem as stated in Problem 3.2' becomes SPACE-complete. We shall not prove this claim here, but cite the related sults of [Kozen, 1977], where the problem of non-empty intersection of regular ts is proved PSPACE-complete. Intuitively, it seems rather unlikely that the ysical problem also exhibits such a jump in complexity, leading us to believe at the abstraction of Problem 3.2 to Problem 3.2' is not tight enough.

Consider the object lying on side 1 as shown in Fig.3.15. Suppose that tilting direction 1 leaves the object lying on side 2. Then, if the object had started on

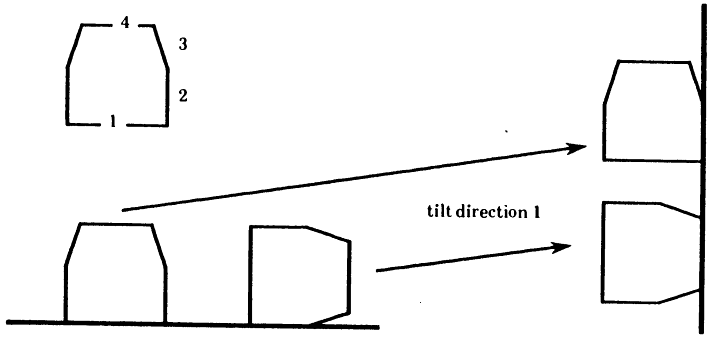

Fig. 3.15. Monotonicity of tilt functions

side 2, it must end up "at least" (in anti-clockwise order) on side 2 after the tilt. Ir short, the tilt functions $f_1, f_2, \ldots, f_k$ must be *monotonic* with respect to the cyclic order on the set of orientations S. To make this more precise, notice that any oriented simple cycle that contains every element of a finite set generates a cyclic order on the set. A sequence $s_1, s_2, \ldots, s_j$ of elements of the set is *ordered* if $s_1, s_2, \ldots, s_j$ are encountered in that order when the generating cycle of S is traced once, starting from s_1. A function $f: S \to S$ is monotonic if for any ordered sequence $s_1, s_2, \ldots, s_j$, the sequence $f(s_1), f(s_2), \ldots, f(s_n)$ is ordered. Now, recalling that S has a cyclic order on it generated by the anti-clockwise numbering of the sides of the object, we are ready to state the abstracted problem afresh.

Problem 3.2"

Given k monotonic functions $f_1, f_2, \ldots, f_k : S \to S$ on a finite set S with a cyclic order, and a subset U of S, find a function f_0 that is a composite of the f_i's such that $|f_0(U)| = 1$.

Before we can give an algorithm to solve the above problem, we need some pporting lemmas and definitions. Let S be a finite set with a cyclic order and $\sigma = s_1, s_2, \ldots, s_n$ be an ordered enumeration of S. The interval $[s_i, s_j]$ of σ is the $\{s_i, s_{i+1}, \ldots, s_j\}$ if s_i precedes s_j in σ. Otherwise, $[s_i, s_j] = [s_i, s_n] \cup [s_1, s_j]$. Let U be ubset of S. We say U *partitions* σ into the intervals $[u_1, u_2], [u_2, u_3], \ldots [u_j, u_1]$ of σ $u_1, u_2, \ldots u_j$ is an ordered enumeration for U such that u_1 appears before u_j in Notice that for a given cyclic order, the partition is unique.

mma 3.1: Let S be a finite set with a cyclic order and let $\sigma = s_1, s_2, \ldots, s_n$ be an lered enumeration for it. Also, let $f: S \rightarrow S$ be a monotonic function such that for ne s_i and s_j in S, $f(s_i) = f(s_j) = s$. Then, if $\{s_i, s_j\}$ partitions σ into $[s_i, s_j]$ and $[s_j, s_i]$, her $f([s_i, s_j]) = s$ or $f([s_j, s_i]) = s$.

oof: Immediate from the monotonicity of the function f. □

mma 3.2: Let S be a finite set with a cyclic order and let $\sigma = s_1, s_2, \ldots, s_n$ be an lered enumeration for it. Let $f: S \rightarrow S$ be a monotonic function such that $S)| > 1$, and let U be a proper subset of S partitioning σ into intervals $[u_1, u_2]$, $u_3], \ldots [u_j, u_1]$. (Let $u_{j+1} = u_1$). Then, $f(U)$ is a singleton set $\{s\}$ if and only if one the intervals $[u_i, u_{i+1}]$ is such that

(a) $f(u_i) = f(u_{i+1}) = s$

(b) for some $v \in [u_i, u_{i+1}]$, $f(v) \neq s$.

oof:

ly if)

There exists v in S such that $f(v) \neq s$. Now v belongs to some interval $[u_i, u_{i+1}]$ it satisfies condition (b). Since $f(U) = \{s\}$, $f(u_i) = f(u_{i+1}) = s$ satisfying condition . Hence the claim is proved.

(if)

Let v in $[u_i, u_{i+1}]$ be such that $f(v) \neq s$. Since f is monotonic, we can appl Lemma 3.1 to the partitioning of S by $\{u_i, u_{i+1}\}$ and conclude that eithe $f([u_i, u_{i+1}]) = \{s\}$ or $f([u_{i+1}, u_i]) = \{s\}$. But $f([u_i, u_{i+1}]) \neq \{s\}$ as $v \in [u_i, u_{i+1}]$ and $f(v) \neq s$ Therefore $f([u_{i+1}, u_i]) = \{s\}$ implying that $f(U) = \{s\}$ as $[u_{i+1}, u_i] \cap U = U$. □

Fig. 3.16 is an algorithm to solve Problem 3.2". Essentially, the algorithn

Algorithm 3.2'

(1) Use Algorithm 3.2 to decide if there exists a composite function f_0 such that $|f_0(S)| = 1$. If so, halt and report function.

(2) Construct a transition diagram for triples in S as was done for pairs in Algorithm 3.2.

(3) Let $\sigma = s_1, s_2, \ldots, s_n$ be an ordered enumeration for S and let $[u_1, u_2], [u_2, u_3], \ldots [u_j, u_1]$ be the partitioning of σ by U. Find a path in the transition diagram above from a vertex (u_i, a, u_{i+1}) to a vertex (s, b, s) where $a \in ([u_i, u_{i+1}] - U)$ and $b \neq s$.

(4) The labels on this path form the required composition sequence. If no such path exists, report failure.

Fig 3.16 A modified function composition algorithm

first checks to see if there exists a composite function that maps all of S to a single element. If such exists, it reports the function. Else, any composite function that maps U to a singleton $\{s\}$ must map at least one element a of S-U to some element other than s. Hence the algorithm searches for a composite function that satisfies conditions (a) and (b) of Lemma 3.2 on some interval $[u_i, u_{i+1}]$, reporting failure if none exists.

Time Analysis

Step (1) takes $O(kn^4)$ time as analysed earlier.

Step (2) takes $O(kn^3)$ time.

Step (3) can be performed in $O(kn^3)$ time using depth first search. The sequence length computed is $O(kn^3)$.

Total time is $O(kn^4)$ and the sequence length is $O(kn^3)$.

Remarks

Algorithm 3.2 finds the minimal cardinality set of orientations reachable from he set of all orientations. Algorithm 3.2' can be easily modified to to do so as vell. This is bit more than what we wanted it to do and might prove useful to educe modifications to the geometry of the object so as to permit unique rientation.

We assumed that all transitions were single valued. It would be interesting to tudy the problem with this restriction removed.

Using the notion of monotonicity, we can modify Algorithm 3.2 to produce a equence of length $O(kn^2\log n)$ in $O(kn^3\log n)$ time. In particular, the algorithm alves the number of posible orientations at each iteration.

Earlier we stated without discussion that if a continuous spectrum of tilt irections were available, the problem would be simplified. We now discuss this sue. Consider the object shown in Fig. 3.17. Each side subtends two *stability* ngles at the centroid, between the perpendicular through the centroid and the ne joining the centroid to the vertices of the side. Let $L=\{l_1,l_2,...,l_n\}$ be the left ability angles for the n sides of the object and let $R=\{r_1,r_2,...r_n\}$ be the right ability angles as shown in Fig. 3.17. Suppose R (or L) had a distinct maximum

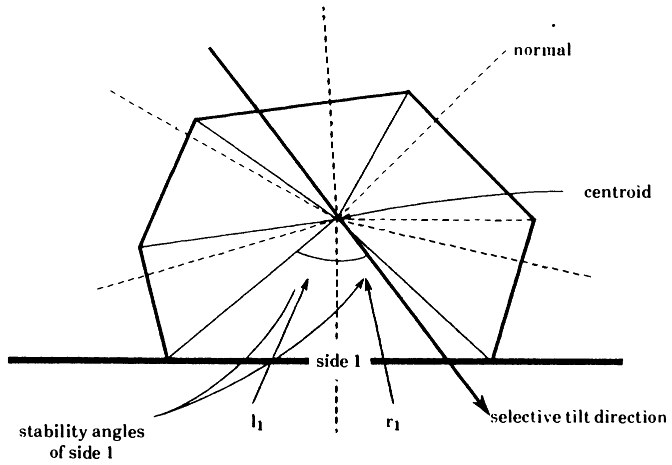

Fig. 3.17. Stability Angles.

as r_1 is for the object of Fig. 3.17. Then there exists a tilt direction for which ther is exactly one stable orientation – the object lying on side 1. In particular, this i any angle between the two largest elements of R. Consequently, tilting in thi direction will immediately orient the object uniquely making the problem rathe simple. However, for objects for which neither R nor S have distinct maxima, w can formulate a new scheme which utilizes the stability angles to orient th object in a more direct way than the pan handler. The statement of the probler would read about the same though as here we have done little more than develo or 'unravel' the tray.

Another question that is interesting is whether there exists a tray shap that is good for all or most objects. For any given tray, there exists a simple objec that is not uniquely orientable by that tray using only median tilts, althoug

:re exists a tray that can do so. To see this, define the *incidence angles* of a tray be the angles made by the tilt directions with the sides on which they are ident. See Fig. 3.18. Let a_l be the smallest non zero incidence angle of the

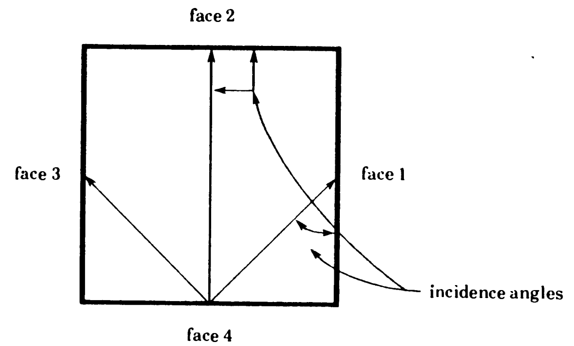

Fig 3.18 Incidence angles

en tray. Pick an even integer N such that $a = \pi/N$ lies between 0 and a_l and struct an 'almost regular' polygon as shown in Fig. 3.19. In particular, pick 0 such that $0 < a-\varepsilon < a < a+\varepsilon < a_l$ and construct a polygon whose vertices lie on rcle and whose sides subtend $2(a-\varepsilon)$ and $2(a+\varepsilon)$ at the centre alternately. See . 3.19. Now the object has two different orientations, but for every median tilt ection of the given tray, both these orientations behave identically. ısequently, the given tray cannot reduce the set of possible orientations to ıer of the two. Yet, a tray with an incidence angle of a will uniquely orient the ect in the orienatation in which it rests on the larger side. To see this, we only d notice that at an incidence angle of a, one of the orientations is unstable and ıce all transitions must be to the other

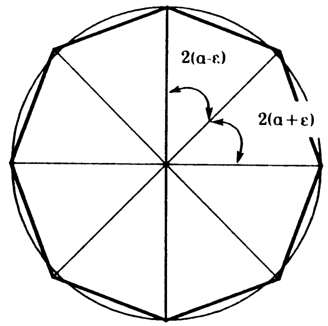

Fig. 3.19. An almost regular object

3.3. PARADIGMS FOR FILTERS

3.3.1 A First Attempt

We introduced the notion of a filter in Section 3.1.3. At first glance, the desig of filters appears to lend itself to the approach that parts orienter design is tl dual of motion planning. In fact, such an approach is explored in [Lozano-Pere 1986]. I contend that such an approach is not very effective with support from tl following example.

Example 3.1

Consider the planar rectangular object of Fig. 3.20a. Suppose we wish t orient it uniquely in one of its two orientations using planar operations. Conside a simple filter for the object where the object moves in a straight line along the axis and static obstacles are placed along the path so as to reject one of the two orientations of the object. We can depict the effect of the filter using the configuration space of the object - a two dimensional space (x,θ), x representing the position of the object along the x-axis and θ the orientation of the object. Since the object can only be in one of two orientations when it enters the filter, θ can be one of two values, say, $\theta=0$ and $\theta=\pi/2$. See Fig. 3.20b. Now, if we were to view

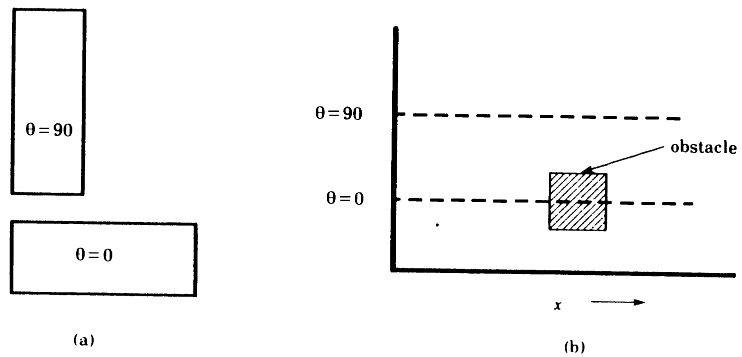

Fig. 3.20. (a) The rectangular object and its two orientations.
(b) The configuration space of the filter

the design of a filter to be the inverse of motion planning, we need only pick an obstacle in this configuration space that intersects say the $\theta=0$ configuration line, but not the other. Mapping this configuration space obstacle to a physical obstacle completes the design of a filter that allows objects entering in the $\theta=90$ orientation to pass through but rejects the $\theta=0$ orientation. The difficulty here is that every configuration space obstacle does not correspond to a physical obstacle. In particular, the obstacle that we chose in Fig. 3.20b is not generated by any physical obstacle. This is to be expected as the configuration space of an object is typically a higher dimensional entity than the physical domain and the mapping from configuration space to physical space is singular.

Apart from such considerations, static design of the obstacles as discus above will cause the filter to be blocked each time the object enters in undesirable orientation. Designing obstacles to discard undesirable orientati cannot be done without knowledge of the dynamics of the situation.

We are forced to conclude that the dynamic issues in parts orienting ar strong that simple mathematical formulations of the problem are infeasil Consequently, we resort once again to considering specific paradigms.

3.3.2 The Vibratory Track

Here we consider the problem of orienting a convex, planar polygonal ob on a sloped vibratory track with a lip as shown in Fig. 3.21. The object cre

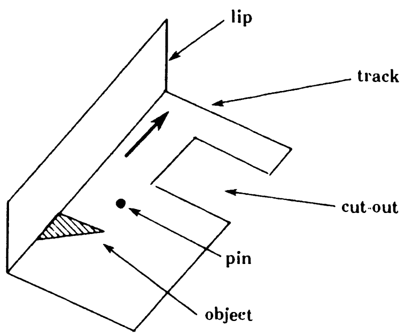

Fig. 3.21. The track.

along the track with one of its sides against the lip. The object may be reorient by means of pins placed along the track and undesirable orientations may discarded by means of cut-outs in the track. Assuming that the probabili ribution of the orientation of the object entering the track is known *a priori*, ıre to find a sequence of pin locations and cut-outs so that the object exits the k in some unique and predetermined orientation with maximal probability.

Problem Statement and Abstraction

n order to make a precise statement of the problem, we make the following ımptions:

a) Some side of the object is always in contact with the lip of the track. This letermines the orientation of the object.

b) A cut-out runs parallel to the lip and is specified by the distance of the edge of the track from the lip at the cut-out. It has the following effect on an orientation: If the centroid of the object is outside the track, the object falls off the track and is discarded. Otherwise, the object is unaffected.

(c) A pin has the following effect on an orientation: Referring to Fig.3.22, if

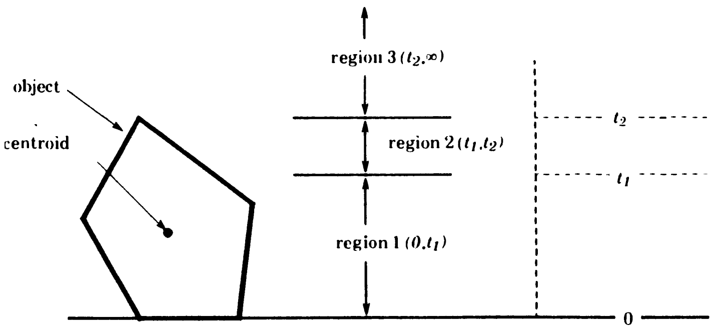

Fig. 3.22. The transition regions of an object

the pin is located in region 1, the object gets stuck at the pin and cannot

proceed. This is to be avoided. If the pin is in region 2 (end points excluded), the object changes orientation to the next (clockwise or anti-clockwise order) stable orientation and proceeds past the pin. In region 3, the object proceeds past the pin without suffering a change in orientation. We refer to these three regions as the *transition regions* defined with respect to each orientation of the object and represent them by the distance intervals from the lip – $(0,t_1)$, (t_1,t_2) and (t_2,∞) as shown in Fig. 3.22. Fig. 3.22 is only indicative and not definitive.

(d) No pin on the track should be in region 1 of any of the possible orientations of the object at that point on the track.

(e) All operations are planar in that one of the two planar faces of the object is distinguished and is always in contact with the track.

We are now ready to state the problem:

Problem 3.3

Given a planar, convex, polygonal object, the transition regions for each orientation of the object, and the probability distribution of the orientation of the object as it enters the track, find a sequence of pin locations and cut-outs so that the object reaches the end of the track in some unique orientation with maximum probability.

Define the transition diagram of the object to be a directed, labeled graph with one vertex per orientation. An edge (u,v) with label $[t_1,t_2]$ is present in the graph if and only if $(0,t_1)$, (t_1,t_2) and (t_2,∞) are the transition regions for orientation u and orientation u changes to v when traversing a pin in the range (t_1,t_2). A vertex has label d if the centroid of the object is at a distance d from the base in that orientation. Observe that the transition diagram of any object consists of a single

:ycle. We can now state the problem as a graph pebbling game on the transition liagram.

ʔroblem 3.3'

ʔebbling Game

start: All vertices are pebbled. The weight of the pebble on a vertex represents the probability of the object entering the track in that orientation.

move: (a) pick a value $t>0$ such that if u is a pebbled vertex and the label on its outgoing edge is $[t_1,t_2]$, $t>t_1$. For each pebbled vertex u with outgoing edge (u,v) with label $[t_1,t_2]$, if $t<t_2$, move the pebbles on u to v.

(b) pick a value $d_0>0$ and discard the pebbles on every vertex with a label $d>d_0$.

luestion: Find a sequence of moves that reduces the number of pebbled vertices) one, but maximizes the weight of the pebbles remaining on the graph at the nd of the game over all sequences that result in a single pebbled vertex.

Fig. 3.23 gives an algorithm to solve the above problem. In the following we ill assume that no edge has a trivial label ($[t_1,t_2]$ is trivial if $t_1=t_2$). This makes ıe algorithm technically simpler while allowing a straightforward extension to ıe general problem.

the algorithm is successful, the moves in Phase 1 together with the moves in ıase 2 form the required sequence. Else no solution exists. □

ime Analysis

ıase 1 takes $O(n^3)$ as there are n^2 moves played and each move costs $O(n)$.

Algorithm 3.3

Phase 1

Play the pebbling game for n^2 moves;

At each move

(a) pick t to be the least value that is legal.

(b) pick $d_0 = \infty$;

Phase 2

Construct a pairwise transition diagram for the object as follows. Each vertex is a pair of orientations (a,b). Edges are defined in the obvious way: if there is an edge from a to c labelled $[t_1,t_2]$ and b to d labelled $[q_1,q_2]$ respectively in the transition diagram then the edges to be included in the pairwise diagram are going to depend on the relationship between the label values. For example, suppose $q_1 \geq t_1$ and $t_2 \geq q_2$. Then the edges to be included are

(a,b) to (c,b) with label $[t_1,q_1]$

(a,b) to (c,d) with label $[q_1,q_2]$

(a,b) to (a,d) with label $[q_2,t_2]$.

If any of the above label intervals are trivial, the corresponding edge is omitted.

Colour a vertex (a,b) red if label(a) < label(b) else colour it blue.

while there is more than one pebbled vertex on the board **do**

let $v_1,v_2,\ldots,$ be the pebbled vertices on the board in decreasing order of pebble weight.

Find a path by depth-first search from (v_1,v_2) to a red vertex (a,b). If no such exists halt and report failure.

With $d_0=\infty$ play the pebbling game so that (v_1,v_2) traces this path. Then with $t=\infty$ and d_0 such that label$(a)<d_0<$label(b) play a move.

od

Fig. 3.23 The pebbling algorithm for the vibratory track.

Phase 2 costs $O(n^4)$:

$O(n^2)$ to construct the pairwise diagram.

$O(n^3)$ for each iteration of the while loop with at most n iterations.

ıere are $O(n^3)$ moves in the solution sequence.

ɔrrectness

Let t_{max} be the maximum label value in the transition diagram. Consider the
t

$S = \{u|$ edge out of u has label $[t_1, t_{max}]$ for some $t_1\}$

ıd the related family of sets defined by

$S_u = \{v| \quad u \in S, v \notin S$ and there is a path from v to u in the transition diagram that does not include an element of $S\}$.

nally define the equivalence relation $\equiv$ on S by $x \equiv y$ if $x, y \in S_u$ for some u.

aim 3.2: If $a \equiv b$ then no sequence of moves can cause the pebbles that started a and b to occupy the same vertex.

·oof: Follows from the observation that the transition diagram consists of a ıgle cycle. □

Given this, we see that if at the end of some sequence of moves there is only e pebbled vertex, then all the pebbles on this vertex started on vertices in the me set S_v, for some vertex v. We now proceed by proving the following claims.

aim 3.3: If there exists a sequence of moves σ that terminates with one pebbled rtex, then Algorithm 3.3 finds such a sequence.

·oof: Let σ terminate with a pebble starting on vertex x remaining on the ard. Distinguish this pebble from all others. Now, by Claim 3.2, any pebble ırting on say some vertex y such that $x \equiv y$ must be discarded by σ. At the time ıs pebble was discarded, say it occupied vertex b and the distinguished pebble :upied vertex a. Then, in our pairwise graph, (a,b) must be a red vertex.

At any stage in Phase 2 of the algorithm, let $v_1, v_2, \ldots$, be the pebbled vertices in decreasing order of pebble weight. Notice that Phase 1 ensures that $v_1 \equiv v_2 \equiv v_3 \ldots$. Furthermore, there exists a sequence of moves (with trivial d_0 values) that moves the pebbles on v_1 to x, as every edge in the transition diagram has a non-trivial label. Let this sequence move the pebbles on v_2 to some vertex y. Then $x \equiv y$ and the existence of σ ensures that there exists a path from (x,y) to a red vertex in the pairwise diagram. □

Claim 3.4: Algorithm 3.3 finds the sequence of moves that terminates with maximum pebble weight.

Proof: Let

$$w_{\max} = \max_{v \in S} \quad (w_v = \sum_{u \in S_v} \text{weight of pebble starting on } u)$$

That is, w_v is the weight of all the pebbles that start on vertices in S_v and w_{max} is the maximum of these. By Claim 3.2 w_{max} is the maximum weight that can be attained on a single vertex. If Algorithm 3.3 terminates successfully, it does so with w_{max}. Hence the claim is proved. □

Remarks

The paradigm of Section 3.3.2 and the subsequent analysis are admittedly simple. For one, three dimensional objects pose a greater challenge and it is unclear whether our anlysis of two dimensional objects is extensible to them. For another, our restrictions on the transitions of the object and the type of obstacles in the paradigm are rather strong.

3.4. Summary

We considered the problem of automated design of parts orienters. After an nitial attempt at viewing the problem as a simple dual to motion planning, we oncluded that the physics of the problem dominated it sufficiently to force a aradigm-by-paradigm analysis. Proceeding on this course, we analyzed three aradigms – the belt, the pan handler and the vibratory track. In each case we rmulated a theoretical abstraction of the problem and then proceeded in a aditional manner. The abstractions to the paradigms brought up interesting roblems on graph pebbling and function sequencing on finite sets. Polynomial me algorithms were developed for each of the abstracted problems.

CHAPTER 4

CONCLUSION

The results reported in this thesis concern two different problems – moti planning in the presence of uncertainty and the automated design of par orienters. Although these problems are rather closely related, they require ve different formulations and methods of analysis owing to the fact that the physi plays a much greater role in the latter problem.

With regard to motion planning, we showed that for a large class of inpu motion planning for rigid objects in three dimensional scenes and robots wi uncertainty is reducible to the classical piano mover's problem and hen decidable in polynomial time. We also showed that the same problem for robo with damping is PSPACE-hard. The significance of our results is as follow Despite uncertainty in velocity, time and position, motion planning for robo without damping is a tractable problem. The addition of damping makes tl robot more powerful, but makes motion planning for it intractable. There is go reason to believe that replacing damping with other sensory modes like visic and force-sensing will alter the complexity of the problem in a similar fashion. I fact, a study of motion planning with other sensory modes will be a goc extension to the results of this thesis.

With regard to parts orienters, the most important contribution of this thesi is the argument that the problem is difficult to formulate in a general manne and requires analysis on a paradigm-by-paradigm basis. Although thre paradigms are presented and analyzed in detail, the paradigms are far too simpl to be immediately useful. The study was largely an initial attempt at bringin together the necessary ingredients to approach this practical problem from th eo retical viewpoint. Further work along these lines could provide a firm eor tical basis for reasoning about this and other such problems. The results esented can withstand considerable extension in all directions, particularly in e way of more realistic assumptions and nondeterministic and less discrete dels.

REFERENCES

Boothroyd, G., Poli, C, and Murch L.E., 1982, Automatic Assembly, Merce Dekker.

Grossman, D.D. and Blasgen, M.W., 1975, Orienting Mechanical Parts by Computer Controlled Manipulator, IEEE Transactions on Systems, Man Cybernetics, Vol. SMC 5, No. 5, pp561-565.

Hopcroft, J.E. and Ullman, J.D., 1979, Introduction to Automata Theory, Languages and Computation, Addison-Wesley.

Hopcroft, J.E., Joseph, D.A., and Whitesides, S.H., 1985, On the Movem of Robot Arms in 2-dimensional Bounded Regions, SIAM Journal of Computi Vol. 14, No. 2.

Joseph, D.A and Plantinga, H.W., 1985, June 5-7, On the Complexity Reachability and Motion Planning Questions, Symposium on Computatio Geometry, Baltimore, Maryland.

Kozen, D., 1977, Oct. 31-Nov. 2, Lower Bounds for Natural Proof Systems, 1 Annual Symposium on Foundations of Computer Science, Providence, Rh Island.

Lozano-Perez, T., and Wesley. M.A., 1979, An Algorithm for Planni Collision-Free Paths Among Polyhedral Obstacles, Communications of the AC Vol. 2, No. 10, pp560-570.

Lozano-Perez, T., Mason, M.T., and Taylor, R.T., 1984, Automatic Synthe of Fine-Motion Strategies for Robots, The International Journal of Roboti Research, Vol. 3, No. 1, pp3-24.

Lozano-Perez, T., 1986, Private communication.

Mason, M.T., 1983, Automatic Planning of Fine Motions: Correctness a Completeness, Carnegie Mellon University., CMU-RI-TR-83-18.

Mason, M.T., and Erdmann, M., 1986, April, An Exploration of Sensorless Manipulation, 3rd IEEE International Conference on Robotics and Automation, San Francisco, California.

Philip, T.K., 1980, Lecture Notes on Automated Manufacturing, Dept. of Mechanical Engineering, Indian Institute of Technology, Madras.

Reif, J.H., 1979, Oct. 29-31, Complexity of the Mover's Problem and Generalizations, 20th Annual Symposium on Foundations of Computer Science, San Juan, Puerto Rico.

Schwartz, J.T., and Sharir., M., 1982, On the 'Piano Mover's' Problem II, Courant Institute of Mathematical Sciences, New York, Report No. 41.

Udupa, S., 1977, Collision Detection and Avoidance in Computer Controlled Manipulators, Ph.D. Thesis, Cal.Tech, Pasadena, California, 1977.