MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

A. I. Memo No. 725 May, 1983

Planning Collision Free Motions for Pick and Place Operations

Rodney A. Brooks

Abstract. An efficient algorithm which finds collision free paths for a manipulator with 5 or 6 revolute joints is described. It solves the problem for four degree of freedom pick and place operations. Examples are given of paths found by the algorithm in tightly cluttered workspaces. The algorithm first describes free space in two ways: as freeways for the hand and payload ensemble and as freeways for the upperarm. Freeways match volumes swept out by manipulator motions and can be "inverted" to find a class of topologically equivalent path segments. The two freeway spaces are searched concurrently under projection of constraints determined by motion of the forearm.

Acknowledgements. This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the Laboratory's Artificial Intelligence research is provided in part by the System Development Foundation, in part by the Office of Naval Research under Office of Naval Research contract N00014-81-K-0494, and in part by the Advanced Research Projects Agency under Office of Naval Research contracts N00014-80-C-0505 and N00014-82-K-0334.

1 Introduction

A key component of automatic planning systems for robot assembly operations is a gross motion planner for the manipulator and its payload. Motions of the manipulator should avoid collisions with obstacles in the workspace.

In this paper we present an new approach to collision free planning motions for a manipulator with revolute joints (e.g. a Unimation 560, known as a PUMA). It is based on a method previously presented for planning motions for a polygon through a two dimensional workspace (Brooks [1983]).

Free space is described in two ways: as freeways for the hand and payload ensemble and as freeways for the upperarm. Freeways match volumes swept out by manipulator motions and can be "inverted" to find a class of topologically equivalent path segments. The two freeway spaces are searched concurrently under projection of constraints determined by motion of the forearm. The sequence in figure 1 illustrates a path found by the algorithm.

Section 1.2 below describes carefully the class of find-path problems to which the algorithm is applicable. Section 2 describes the algorithm qualitatively and section 3 gives the detailed equations necessary to implement it.

1.1 Manipulator motions; useful and otherwise

In this section we argue two points. First, we argue that the find-path problem for a manipulator with revolute joints is inherently difficult. Second we argue that this should be of small concern as there is little need to solve the most general problem; almost all realistic problems are members of one of a number of classes of less general problems – our algorithm solves one of those classes.

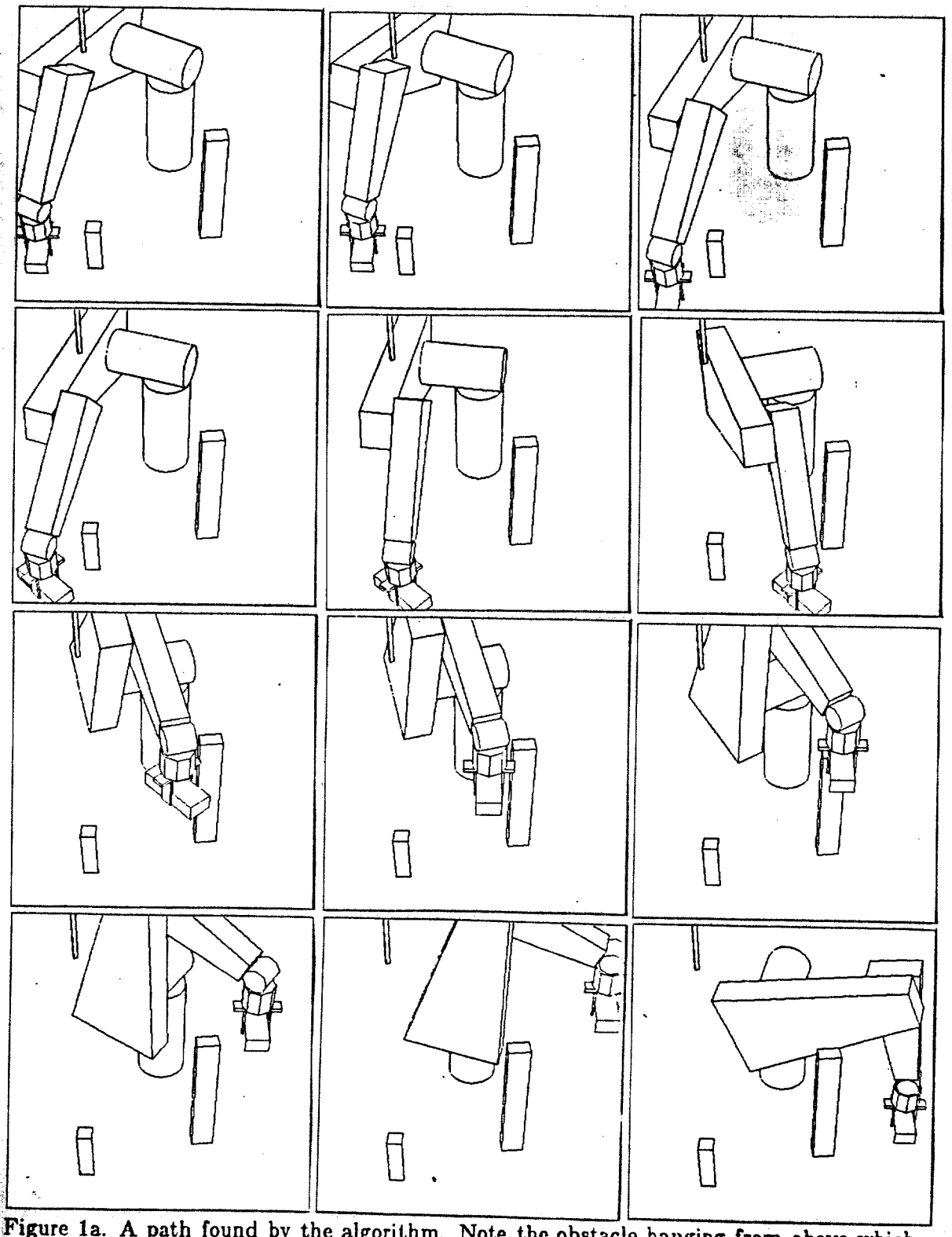

Figure 1a. A path found by the algorithm. Note the obstacle hanging from above which forces the upperarm to stay low until it is out from under it.

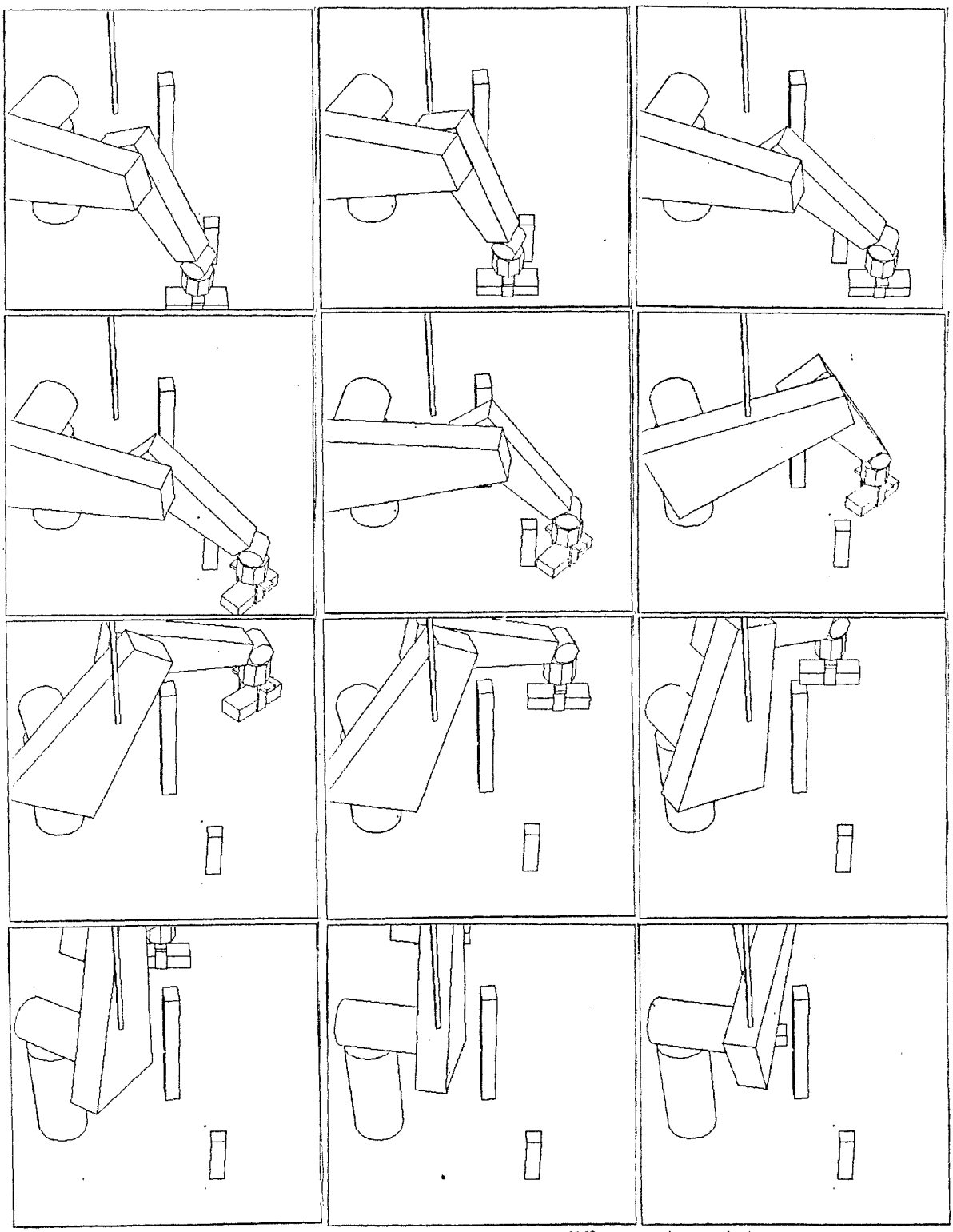

Figure 1b. The same path as in figure 1a but from a different viewpoint.

We claim that finding a collision free path for a manipulator and its payload through an obstacle cluttered space is inherently hard.

We do not mean hard in the algorithmic complexity sense, as it is known that there exists a polynomial time (in the number of obstacles) algorithm for any particular manipulator (Schwartz and Sharir [1982]). The best known time bounds of these algorithms, however, have exponents which make them impractical and indeed their implementation complexity is staggering and untried.

We mean hard in the sense that we expect all algorithms for the general problem to have large exponent worst case behavior and to have constant factors which make them slow even for simple problems. In addition we expect all such algorithms to be complex to implement. Our reasons for these beliefs are two fold: (1) the non-decomposability of the problem, and (2) the difficulties associated with three-space rotations.

Locally, at any point in the interior of the working volume of a six degree of freedom manipulator the payload can be made to follow an arbitrary curve in three-space with arbitrary re-orientation. Even for simple motions such as a straight line with fixed orientation, however, all of the first three joints of the manipulator are involved (see figure 2 for an illustration of the PUMA's joints). Thus, for instance, the elbow moves when the payload is translated. The direction and magnitude of the elbow movement is a complex function of the direction and location of the payload motion segment. Thus, in considering a motion which enables the payload to avoid an obstacle it is necessary to consider the collision behavior of the elbow at a distant locale. Analysis of the motions would be simpler if they could be considered to be uncoupled.

Unfortunately the elbow behavior can not be generally characterized over the range of possible payload motions in any simple way. This means that the general problem can not be decomposed into considering the payload and the elbow separately. If one had a

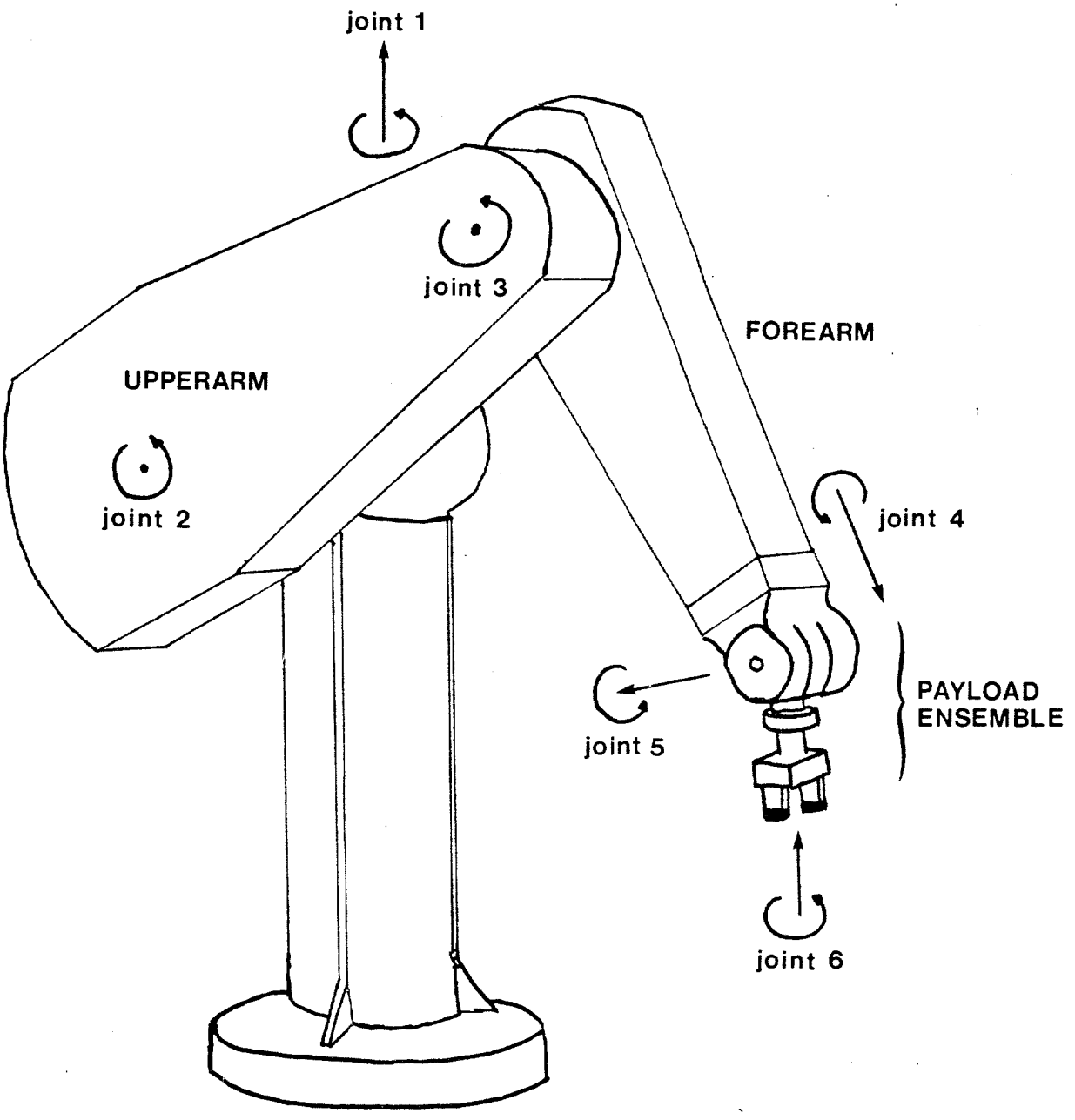

Figure 2. The PUMA has six revolute joints. It can be decomposed into three components: the upperarm, the forearm, and the combined wrist, hand and payload.

manipulator with more than six degrees of freedom then perhaps a decomposition would be possible. The payload could have all six of its degrees of freedom in motion through space, but some of the manipulator joint freedoms could be traded for a decoupling of links near

the base of the manipulator from the motions of the payload. In this case however we would not be solving the general find-path problem for this particular manipulator. There would exist obstacle configurations which could only be navigated by using joint motions which destroyed the decoupling effort. The algorithm we present below takes another approach – the degrees of freedom of motion of the payload are reduced.

Regardless of the above it is also true that even the simpler problem of moving a re-orientable polyhedron through three dimensional space littered with polyhedra has no practical algorithm for problems that are at all difficult. Brooks and Lozano-Pérez [1983] demonstrated an algorithm for polygons in two dimensional space. Our attempts to generalize that algorithm to three dimensions were frustrated by the increased complexity for three dimensional rotations relative to that of rotations in two dimensions.

Two dimensional rotations depend on only one parameter – the rotation magnitude. Small changes in the parameter lead to only small changes in the rotation operator. E.g. let x be a point in two dimensional space and $r(\theta)$ be a rotation of magnitude θ about the origin. Then

$$||r(\theta)x - r(\theta + \epsilon)x||$$

is independent of θ and is monotonic in $||\epsilon||$ (in the range 0 to π). General rotations of three dimensional objects have three degrees of freedom and thus form a three parameter family. Various parameterizations are possible, but no matter which is chosen and whatever metric is used for parameter space, no "continuity" property such as for two dimensions seems to hold. It is possible to get "continuity" for three dimensional rotations by embedding them as 3-d manifolds in higher dimensional spaces. For instance, quaternions can be used to represent rotations as a sphere in 4 space with diametrically opposite points identified. Schwartz and Sharir [1982] represent them as quadratic surfaces in 9 space. In a higher dimensional search space, however, one cannot move about arbitrarily in the parameter space and still have a rotation — care must be taken to satisfy additional constraints. In our case the result of all these problems was that we were unable to generalize our method of chopping parameter space into boxes with simple characterizations of their contents. In general the poor behavior of rotations in three dimensions bodes ill for simple algorithms

to handle them.

In our case even if we had been able to overcome the rotation difficulties and extend the algorithm to three dimensions and beyond it was clear that our space representation was exponential in the number of degrees of freedom of the moving object and hence our algorithm was also time exponential in the degrees of freedom. This behavior matches that of other known general find-path algorithms, for instance Schwartz and Sharir [1982] (see below for more details).

Lastly we note the danger of the super-human/human fallacy. Humans can not necessarily negotiate an arbitrary payload through a cluttered workspace even when a collision free path exists. They are especially poor at the task when a lot of re-orientation (throughout orientation parameter space) is needed.

It doesn't matter

Many motions of a manipulator do not require a full six degrees of freedom. We consider two classes of motions: (1) pick and place and (2) insertions or fitting.

(1) Consider the problem of picking up an object in one place and putting it down in another. For most applications the only re-orientations necessary for the object will be about the vertical axis or perhaps re-orientations at the ends of the gross motion using pure wrist motions. Arbitrary re-orientations of the object during gross motion of the manipulator will only be necessary when either the object is much larger than the manipulator wrist and hand assembly, or when the workspace is extremely cluttered. Both cases are extreme and there really must be a combination of the two to make the gross transfer motion difficult.

(2) For most insertion operations four degrees of freedom typically suffice (see Nevins and Whitney [1978]), perhaps modulo some very fine accommodating motions. Typically there is a direction of insertion (an *approach vector*) and the only required rotations are about a vector in that direction. Thus locally, at least, three translational degrees and one rotational

degree of freedom suffice.

The algorithm we present below solves the first of these problems, and the second in the case that the approach vector is vertical. In addition, for the case of a six joint manipulator with a wrist with intersecting axes, it is generalizable to arbitrary approach vectors, although each different approach vector requires a recomputation of the representation of obstacles.

Some assembly operations really require arbitrary fine motion of a workpiece and hence it would seem that the complete find–path problem must be solved for these cases. However it is becoming increasingly obvious that a six degree of freedom manipulator is not sufficient for these tasks. Any motion requires control of the first few large joints of the manipulator and such control must be extremely accurate to achieve even moderate accuracy at the end effector. There have therefore been moves towards the development of extremely dextrous hands, providing all six degrees of motion freedom. Fine motion and force control require a decomposition of the manipulator into gross and fine subcomponents. Exactly the same decomposition allows us to avoid the path planning task for a tightly coupled payload and manipulator upperarm. Thus a general findpath planner can again be replaced by a a more restricted path planner.

But ...

There is another common task for manipulators that we have ignored above. That is re–orienting a workpiece. For instance it may be necessary to take a piece of metal from a pallet, rotate $\frac{\pi}{2}$ about a horizontal axis and place it in a machine tool. Such motions need an additional level of planning.

The motion with re–orientation can be split into three steps: (a) gross motion, (b) re–orientation, and (c) gross motion. During each gross motion phase only one rotational degree of freedom for the payload need be allowed, and then the path planner of this paper can be used. Step (b) can be done in a large open volume with the first few joints of the

manipulator held constant while the wrist re-orients the payload. The re-orientation phase can be planned with a special purpose planner – it is a straightforward task if there is plenty of open volume available, e.g. Lozano–Pérez [1981].

1.2 The problem solved here

In the algorithm we present below we buy decomposability of the problem by spending two degrees of freedom of the manipulator to partially decouple the payload and the upperarm of the manipulator. In the six degree of freedom PUMA we keep joint 4 fixed (there is no joint 4 for the 5 degree of freedom PUMA) and use joint 5 to compensate the payload orientation for the motions of the upper and forearms. Joint 6 is free to re–orient the payload about its vertical axis, but such re–orientation does not require motion of either the upper or the forearm – it is completely decoupled. This is only two dimensional rotation. There is still coupling of translations of the payload and the motion of the upper links of the arm. A major new contribution of the algorithm is that motion of the components can at first be analyzed separately and then, later, constraints are propagated between the solutions to account for the remaining coupling.

We find paths where the payload is moved in straight lines, either horizontal or vertical, and is only re–oriented by rotations about the vertical axis of the world coordinate system. Thus we consider only 4 degrees of freedom for the PUMA.

The payload and the hand are merged geometrically, and the payload is considered to be a prism, with convex cross section. The payload can rotate about the vertical, as joint 6 rotates.

Obstacles in the work space are of two types: those supported from below and those hanging from above. Both are prisms with convex cross sections. Non–convex obstacles can be modeled by overlapping prisms. Prisms can be supported from below if they rest on the workspace table or on one another as long as they are fully supported. Thus no point in free space ever has a bottom supported obstacle above it. Work is currently under

way to extract such obstacle descriptions from depth measurements from a stereo pair of overhead cameras.

Similar pre-defined obstacles may also hang from above intruding into the workspace of the upperarm and forearm. Obstacles are precluded from a cylinder surrounding the manipulator base.

The class of motions allowed suffice for many assembly operations, and, with appropriate algorithms for re-orienting the payload without major arm motion, the algorithm can provide gross motion planning for all but the most difficult realistic problems. Fine motion planners may still be required for difficult terminal motions of a hand and workpiece.

1.3 Previous path planning work

There have been numerous algorithms detailed for finding collision free paths for polygons and polyhedra through space littered with similar obstacles.

The problem is much harder for general articulated manipulators. Schwartz and Sharir [1982] have demonstrated the existence of a polynomial time algorithm for a general hinged device. Unfortunately the best known time bound for the algorithm for a six degree of freedom manipulator is $O(n^{64})$ where n is polynomially dependent on the number of obstacles. The algorithm is of theoretical interest only.

Practical algorithms have been few, and fall into two classes.

1. Lozano-Pérez [1981, 1983] restricted attention to cartesian manipulators. The links of the manipulator can not rotate and so the joint space of the manipulator corresponds exactly to the configuration space for motion of the payload alone.

2. Udupa [1977] and Widdoes [1974] presented methods for the Stanford arm. Both rely on approximations for the payload, limited wrist action, and tesselation of joint space to

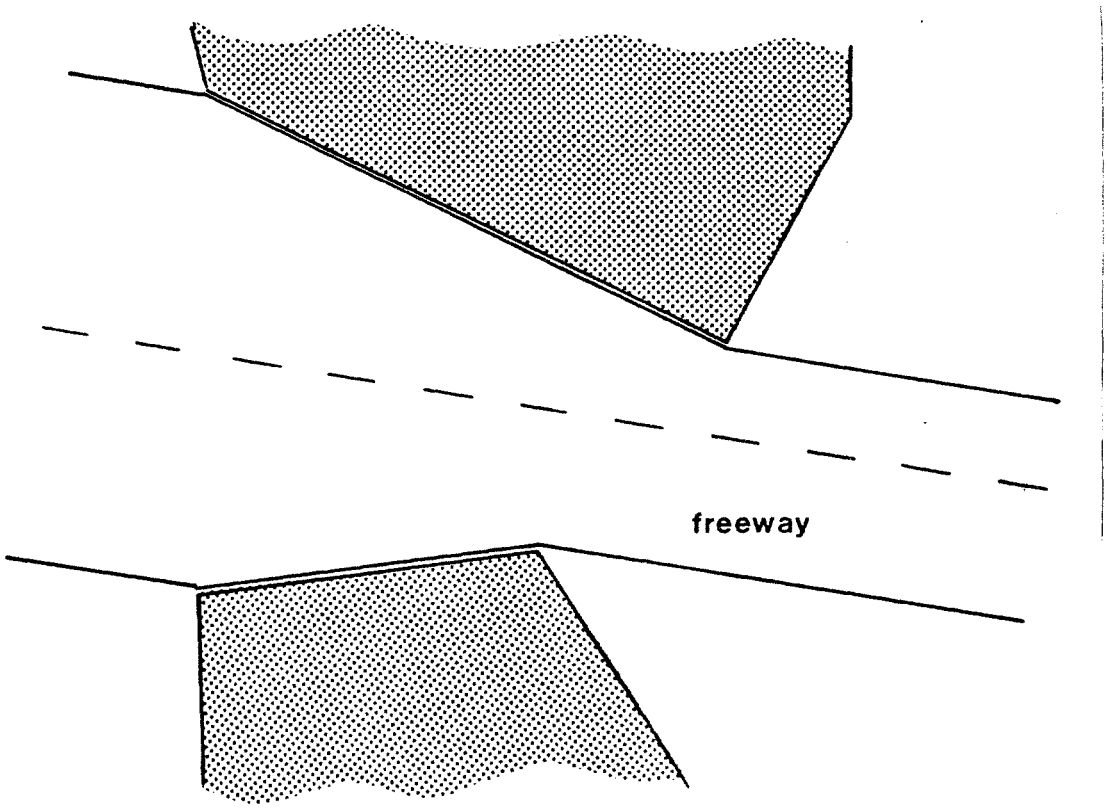

Figure 3. A freeway is an elongated piece of free space which describes a path between obstacles.

describe forbidden and free regions of real space. The problem with tesselation schemes is that to get adequate motion control a multi-dimensional space must be finely tesselated.

2. The Find-Path Algorithm

The key idea is that free space should be explicitly represented in such terms that it is easy to determine the collision free motion segments that can be made by the manipulator and its payload. Individual legal motion segments are linked to form a complete motion for the manipulator.

This was the main idea introduced in Brooks [1983]. There the problem was to find a collision free path for a convex polygon through a workspace littered with polygons. Free space was described as overlapping "generalized cones" or freeways: essentially as straight lines, or "spines", with left and right free space width functions. Figure 3 shows an example

freeway. The possible motions, object orientation as a function of position along the straight line, were found by determining a subset of legal orientations at each point on the spine. Motion segments consisted of translations, with re-orientation, along segments of spines. A complete path was found by transferring from one spine to another where they intersected. At such points the moving object must have an orientation legal for both freeways. Thus a transfer from one freeway to another puts boundary conditions on each of the motion segments, in terms of the legal object orientations at segment end points.

The fundamental idea which makes this method workable is to describe free space in such terms that locally a large class of legal motions can be determined. In addition it must be easy to filter these legal motions with boundary conditions determined by the choice of transfer from one local motion segment to another.

2.1 Vision

Before we can decompose free space we must consider the input descriptions we can expect for it. There are two sources of descriptions we could expect and, orthogonally, two forms which such descriptions might take.

The geometry of the workspace may come either from geometric models or from direct observation using some sort of sensors. The free space might be described explicitly or it might be the occupied volumes which are described explicitly.

The examples in this paper were all generated using geometric models of space filling volumes. The free space descriptions were computed from those models. We plan on using a pair of directly overhead cameras to collect this data in the future. Initially, at least, we will be using algorithms which compute coarse surface descriptions of objects in the workspace. The descriptions will be 2-d surface patches parallel to the image planes of the camera, along with a depth measurement (we will use the system developed at MIT by Nishihara [1983]). A reasonable "safety assumption" for collision avoidance tasks is that there may be obstacles anywhere that it is not explicitly known that there are not.

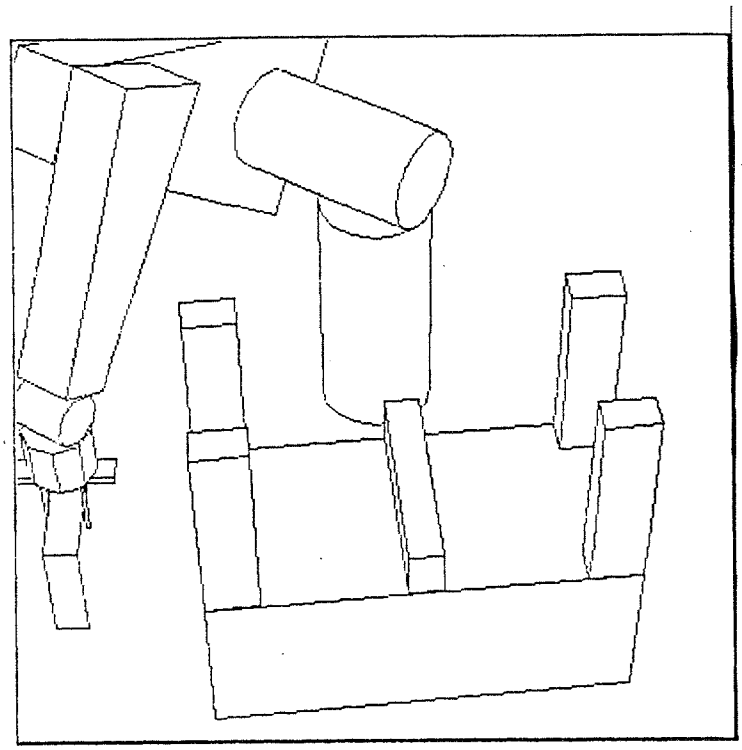

Figure 4. The world is modeled as fully supported stacked prisms.

Visible surface patches imply that the volume above them is obstacle free. We must assume, however, that any volume below them may be filled with obstacles. If the surface patches are polygonal in shape then this leads to world models as described in section 1.2 – namely fully supported vertical prisms. Figure 4 shows an example such world with the manipulator in the background. We make the following observation about our prism world representation:

Free space skyward property. If a point is in free space then all points above it are in free space. (This implies that if a point is in an obstacle then all points below it are also.)

It turns out that the analysis of prism world for interactions with the upper and fore arms is equally valid for prisms hanging from above, and furthermore the interactions they have with table–based obstacles are simple to manage. The same is not true for the interactions of hanging obstacles with the space representations for the payload. Therefore the algorithm is restricted to dealing only with hanging obstacles which may extend into the working volume of the upper and fore arms, but may not extend into the working

volume of the payload. In the later case obstacles must be considered to extend all the way down to the table level. Since the hanging obstacles and their downward extent will not be visible to an overhead camera we assume that they are pre-described by geometric models.

2.2 Decomposing the manipulator into parts

The classes of possible motions for different links of the manipulator and the payload differ substantially. Refer to figure 2. Consider the upperarm of the manipulator. Points on its axis for joint 2 are constrained to lie on circles. The upperarm can swing up and down about that axis while it swings around joint 1. Compare this to the motions possible for the payload. Locally the payload can translate or rotate in any direction.

If we wish to locally characterize the free space in terms of possible collision free motions the above suggests that we need different characterizations of space for different parts of the manipulator and its payload.

The payload

Consider the payload. Following our discussion in section 1.1 it will suffice to move it in horizontal and vertical straight lines only, with re-orientation only about the vertical axis. For convenience we will choose the axis of joint 6 as the axis of payload rotation. We develop here a representation of free space which enables us to quickly compute a large class of motions for the payload which are collision free for it – in later sections we must take into account collisions such motions would cause for the upper and fore arms.

We first note the following. Suppose we bound the payload and hand ensemble with a vertical polygonal prism. Then the *free space skyward property* tells us that a collision free path for the base polygon, through three-space, generates a collision free path for the payload and hand.

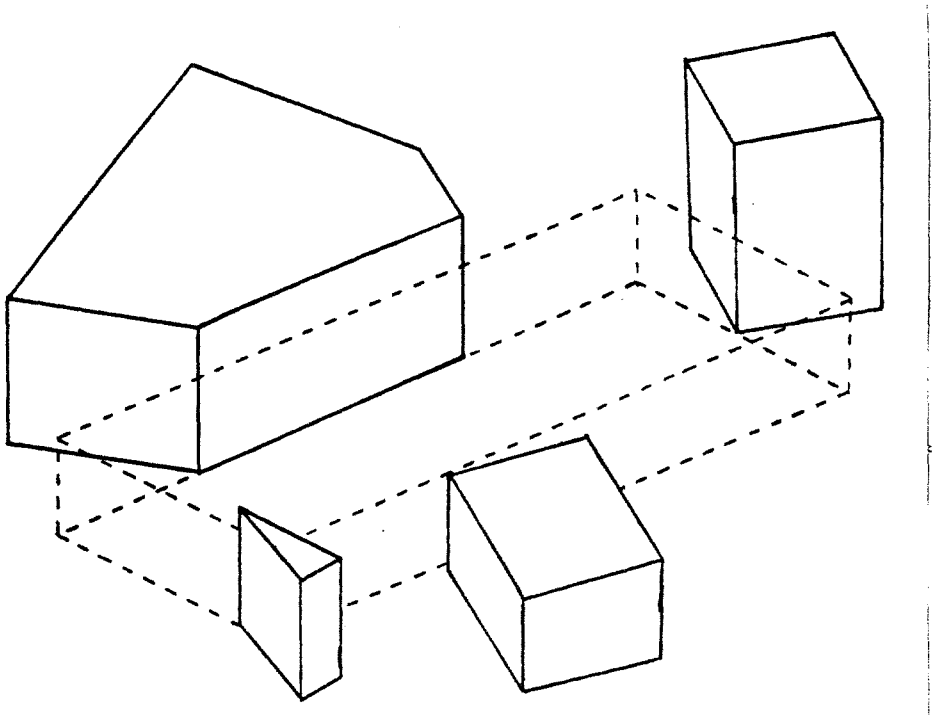

Figure 5. A 3-d freeway is defined by a vertical face of a prism and is terminated by other prismatic obstacles.

Now consider finding a representation of free space which will enable us to compute horizontal motions. At any particular height we have reduced the problem to finding a path for a polygon through a plane. Furthermore since all the obstacles in three space are polygonal prisms the obstacles within the particular plane will be polygonal. Brooks [1983] presented an algorithm for that problem where the moving object was complex. Thus if we assume a convex bounding approximation for the payload we already have a representation and algorithm within a given plane. The representation is overlapping (2-d) freeways.

If there are only a finite number of obstacles represented in the workspace, then there are only a finite number of different horizontal cross sections of the workspace. These can be grouped into horizontal slices which are uniform within their height bounds. For each such slice, algorithms similar to those of Brooks [1983] can be used to compute 2-d freeways. Vertical faces of prisms provide defining edges in a cross section for the swept sides of freeways. Other prisms provide polygonal obstacles which limit the extent of sweeping.

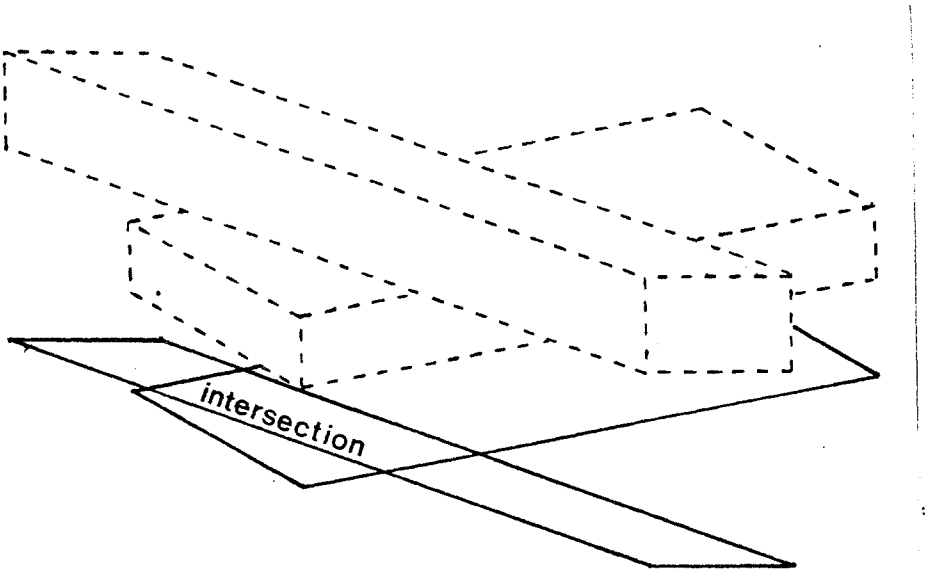

Figure 6. The vertical height of freeways need not be considered in testing for intersection. Only the intersections of their projections into the table top is important.

The 2-d structures so computed can be extended to three dimensions by sweeping them uniformly vertically over a range of heights determined by the continued existence of their defining edges and limiting obstacles. See figure 5 for an illustration.

If it is possible to move the payload cross section through the 2-d freeway, along its spine, then the payload can be moved horizontally along the spine so that its base is within the 3-d freeway's height range. Due to the *free space skyward property* it is also possible to move the payload along the spine, but above the height bounds of the freeway. That makes little sense, however, as not all the defining constraints (due to obstacles) are still valid so it is better to use the looser descriptions of free space provided by examining the higher payload workspace cross section slice.

Possible vertical motions are simply captured by this representation – again due to the *free space skyward property*. If the payload prism base can be placed in a collision free position in three space, then the payload can be lifted vertically upwards from that position and it will not collide with any obstacle.

A result of the observation of the previous paragraph is that we do not need to consider

relative heights of 3-d freeways in deciding whether the payload can be moved from one to another. We need only consider the intersections of their 2-d projections into the table top. If two 2-d freeways intersect there, and if it is possible to move the payload cross section from one freeway to another there, then we can transfer the payload from one 3-d freeway to another by introducing a vertical motion above the projected intersection. Figure 6 illustrates this point.

The upperarm

The upperarm (see figure 2) is large and can easily collide with any significantly sized obstacles in the workspace. Its motion is both controlled and constrained by joints 1 and 2. Let the state of these joints be described by variables ϕ and α respectively. If $\alpha = 0$ then the upperarm is sticking out horizontally.

The class of possible motions of the arm can not be easily characterized as translations as was the case with the payload. It is better to describe freeways for it in the configuration space (Lozano–Pérez [1981, 1983]) of ϕ and α. Furthermore the upperarm can not change orientation in ϕ–α space and its shape is fixed (in contrast to the payload which changes shape as things are picked up and put down) over all time. Therefore we can compile in special knowledge of that shape (in contrast to the general shape characterization used for the payload).

In general the task of mapping three space polyhedra into obstacles in the joint space of a manipulator is horrendously complex with non–convex and non–linear results. It turns out, however, that the restrictions we have placed on the classes of obstacles we describe (polygonal vertical prisms) conspire to make it easy to find simply described (closely) bounding shapes in this particular joint space.

The sides of the prisms map into constraints in ϕ–α of the form $\phi \leq \phi_0$ and $\phi \geq \phi_0$, i.e. straight lines parallel to the α axis. The constraints forced by the top or bottom faces of the prisms correspond to the upper arm hitting an edge of those faces. Each edge provides

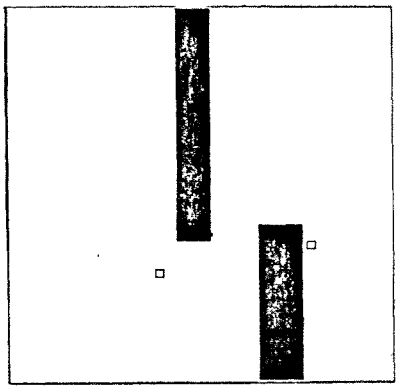

Figure 7. The obstacles in ϕ–α space (ϕ is the horizontal axis of the diagram and α the vertical) derived from the workspace of figure 1. The small obstacle in figure 1 is out of the reach of the upperarm so does not appear here. The small squares indicate the initial and goal configurations of the upperarm.

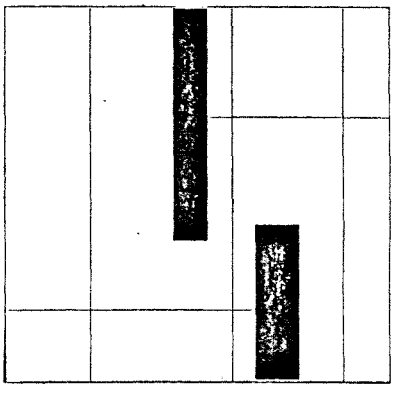

Figure 8. The freeways in ϕ–α space derived from the obstacles of figure 7.

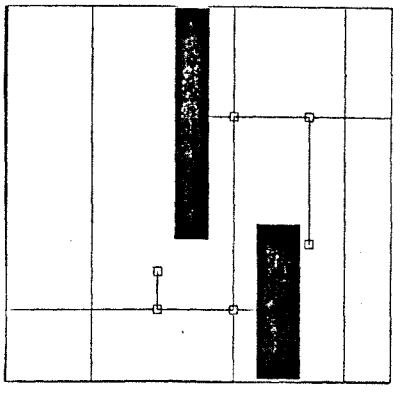

Figure 9. A path for the upperarm from initial to goal configurations.

a non-linear constraint. For moderately sized obstacles, however, or for any moderately small range of ϕ the constraints can be conservatively bounded simply, and reasonably accurately, by constraints of the form $\alpha \leq \alpha_0$ and $\alpha \geq \alpha_0$.

Thus we can populate ϕ-α space with overlapping rectangular obstacles. Figure 7 shows an example. This space is two dimensional and so we can apply our descriptive machinery from Brooks [1983] to describe freeways in this space - see figure 8. Notice that we only need to move a point through this space and that all freeways are actually rectangles aligned with the axes. Since there is no shape associated with the moving object there is no need to restrict motion to the spines of the freeways. Thus, although figure 9 shows a particular path for the upperarm we will only use paths which are an ordered list of freeways, indicating that the upperarm should move in some path through each one in turn.

The forearm

The class of possible motions of the forearm can not be as easily characterized as were those for the payload and the upperarm. It is the link between the two parts of the manipulator which we try to keep decoupled in the path planning process. In the next section we show how the forearm is ensured a collision free path by generation of constraints dependent upon its shape, given a recoupling of the upperarm and payload over a small class of path segments. We directly compute the constraints on payload height, over a proposed motion segment.

Note that this part of the algorithm is not yet implemented in the system used to generate the examples of figure 1 and figure 13.

2.3 Constraint propagation and search

The find-path algorithm first finds a list of ϕ-α freeways for the upperarm. Then it searches the space of payload freeways for a path which lets the upperarm follow the ϕ-α path already determined. The ϕ-α freeways both provide direction to the payload search as to which freeways to choose and they place constraints on the payload height within a freeway. When a freeway is being considered as a path segment for the payload, the path

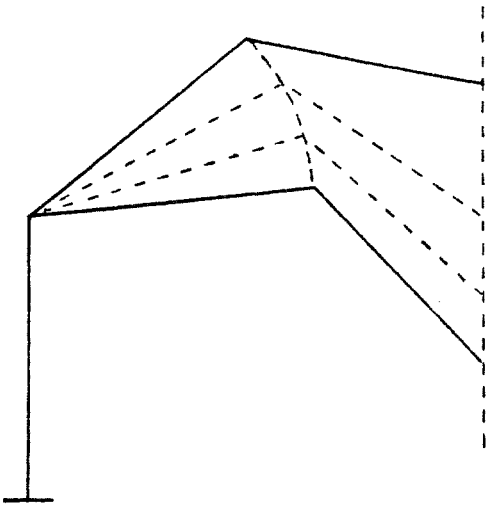

Figure 10. The extreme positions of the forearm during a vertical motion bound a volume which can not be intercepted by obstacles if each of the extreme positions is collision free.

it implies for the forearm is checked. On that basis the path segment may be rejected, as the forearm might place further height of payload motion constraints upon it.

Vertical motions are easy

The following observation is critical to developing a strategy for searching for a collision free path for the payload and the manipulator.

Observation: A vertical translational motion of the payload is collision free for the manipulator and payload if the static positions for the manipulator and payload at the extremes of the motion are collision free.

For the payload this fact follows from the *free space skyward property.* Consider figure 10, and the volume swept by the forearm during a vertical motion of the payload. A prismatic obstacle hanging from above can only protrude into that volume by piercing the top surface of the forearm in its upper position. Conceivably an obstacle from below can protrude into the volume through the swept cylindrical surface show in the diagram, avoiding completely the lower surface of the forearm in its lower position. Then however,

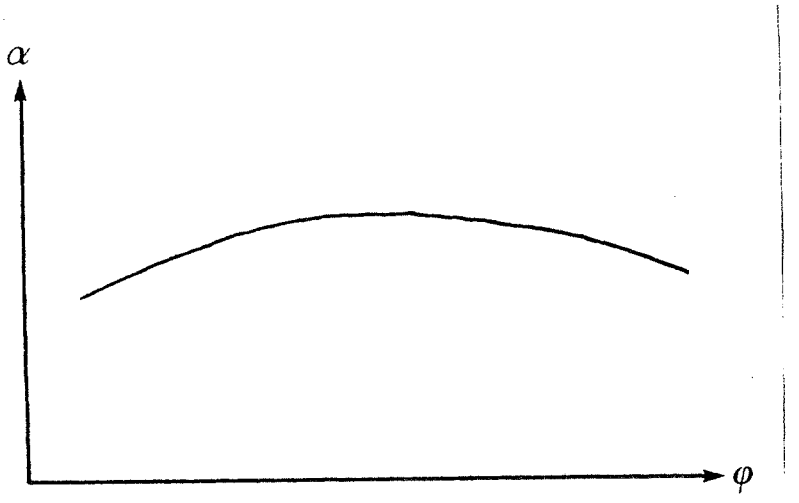

Figure 11. A typical motion of the upperarm when the payload is moved along a straight horizontal segment.

that obstacle must be piercing the lower surface of the upperarm in its lower position. The observation is thus illuminated for the forearm. Similar reasoning illuminates it for the upperarm.

Upperarm constraints

Now consider the motions of the upperarm that are induced by trying to move the payload in a horizontal straight line.

A straight line horizontal motion of the payload results in a motion of the upperarm in ϕ-α space which typically has the shape of the curve in figure 11. The maximum α value occurs when the center of joint 2 is closest to the line of motion. That point may not be reached for a particular segment. In any case a particular straight motion segment can be bounded in a box with sides parallel to the ϕ and α axes. Thus it is easy to determine whether the required upperarm motion is constrained to lie in a particular ϕ–α freeway. Conversely, given the spine of a 2-d payload freeway, and a ϕ-α freeway it turns out to be analytically (and hence computationally) simple to determine the constraints on the height of the motion which keep the upperarm within the freeway.

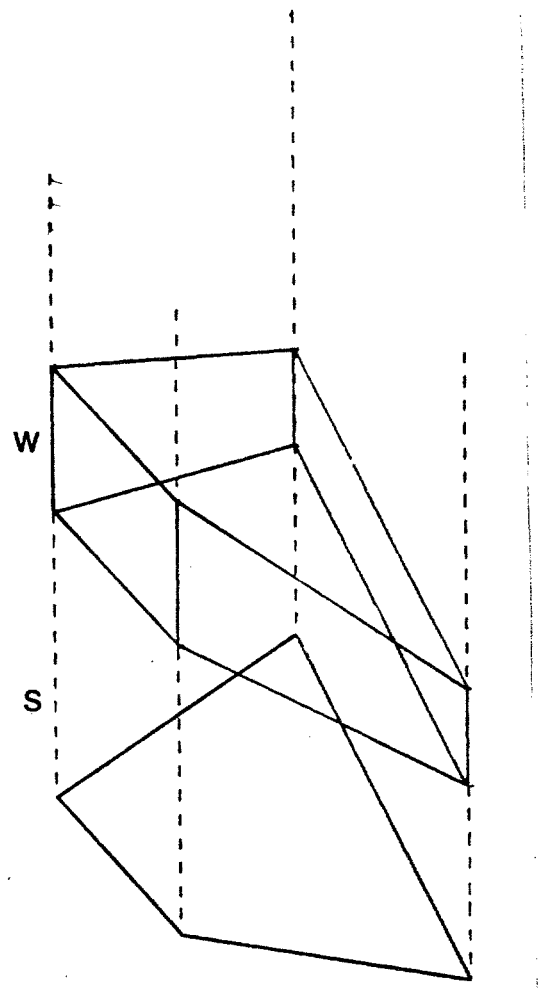

Figure 12. Wedge W defined by two bounding planes and a shadow prism S is a bound on the volume swept by the forearm during horizontal payload motion.

Forearm constraints

Consider moving the payload along a horizontal segment. The forearm sweeps out a volume. It is computationally simple to bound that volume by two planes parallel to the motion segment intersected with a vertical prism whose cross section is a quadrilateral bounding the projection onto the ground plane of the actual swept volume. Figure 12 shows an example such volume. Let the swept wedge be called W. Let the vertical quadrilateral prism be called S (for *shadow*).

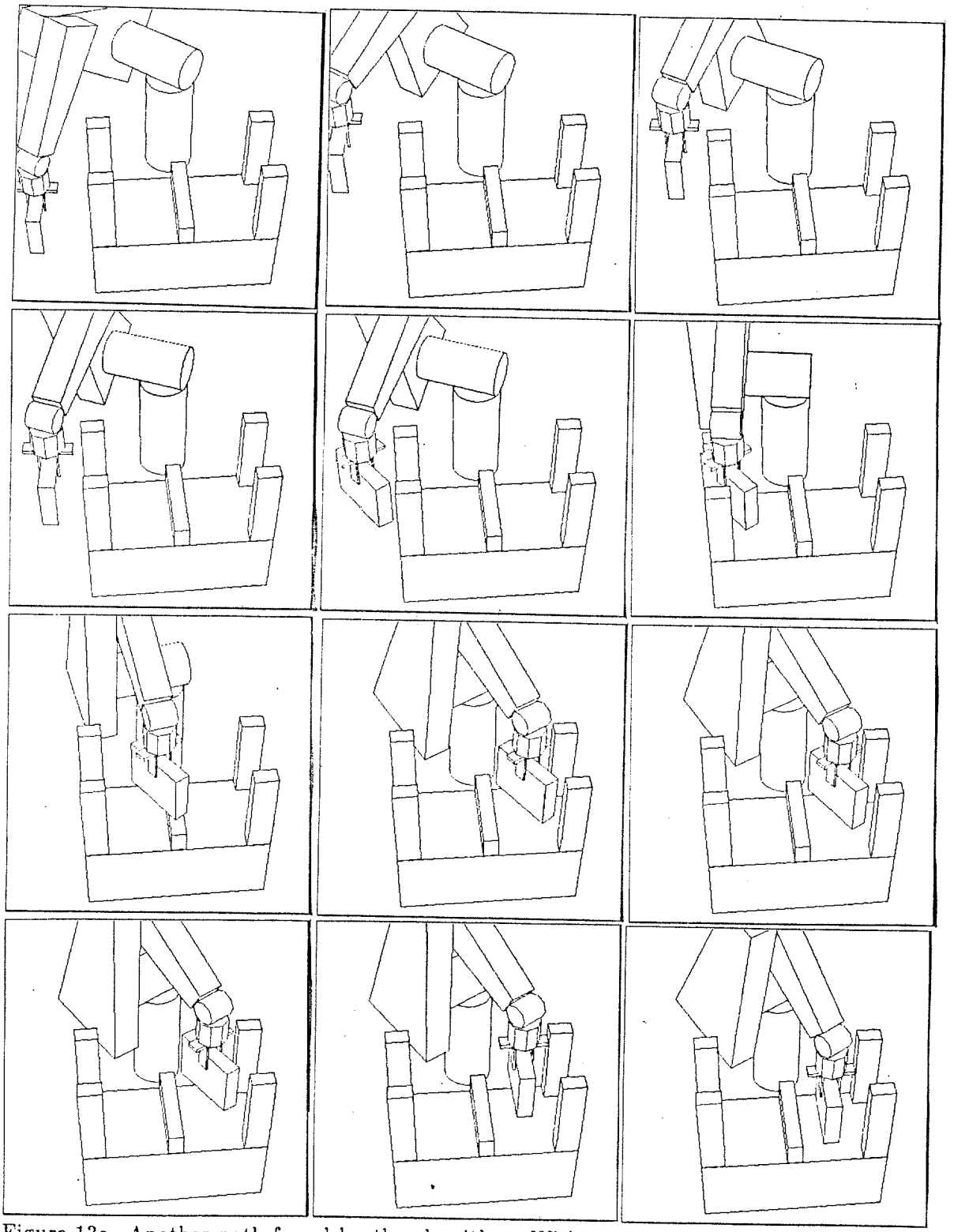

Figure 13a. Another path found by the algorithm. With a payload which is less high the algorithm simply lifts it over the top of the obstacles. In this case it can not reach high enough, so must slide it between them.

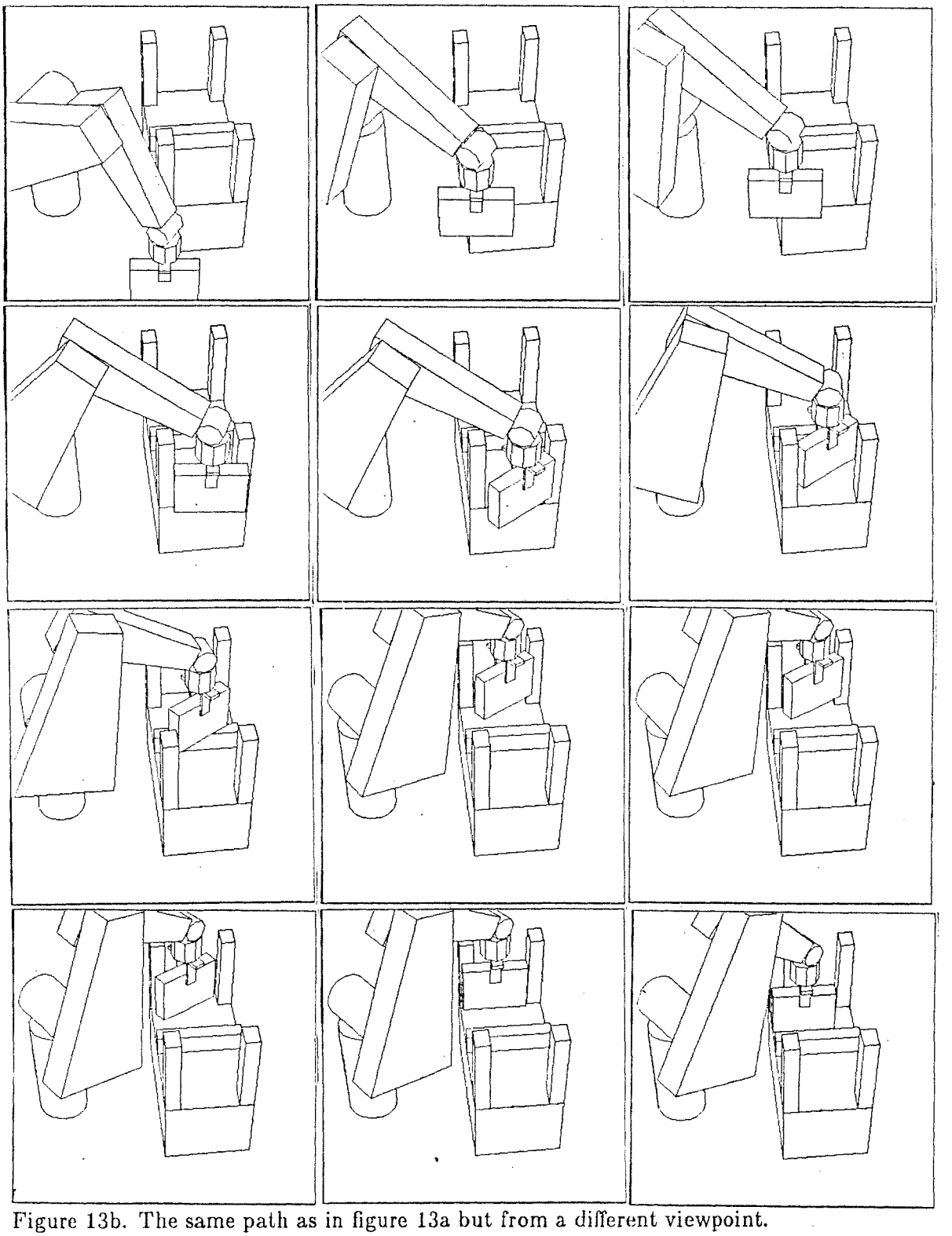

Figure 13b. The same path as in figure 13a but from a different viewpoint.

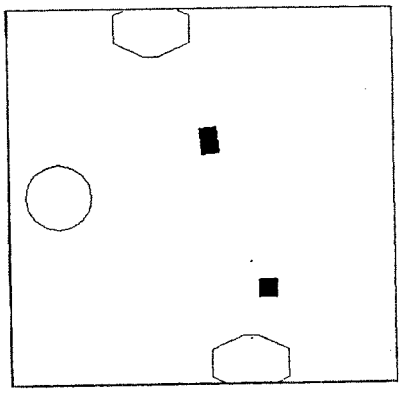

Figure 14. The convex hull of the payload and hand is shown at the initial and goal positions of figure 1.

Consider a prismatic obstacle, anchored below, which is completely contained in S. If all its top surface vertices are below the lower bounding plane of W then it is guaranteed that the forearm does not hit it while the payload follows the horizontal line segment. Conversely if there were a collision it must be the case that one of the top surface vertices is above the lower bounding plane. In general it suffices to intersect an obstacle prism with S, then make sure all the top surface vertices lie below W. Similarly reasoning holds for obstacles protruding into the work space from above.

By making W more oblique, the same wedge moved to different heights can act as a bounding volume for the forearm, for different heights of travel for the payload. Now the vertices of obstacle prisms intersected with S can be used to determine legal heights for W and hence legal heights for motion of the payload.

3. The Algorithm Details

In this section we give the full details of the analysis of the manipulator and obstacle space. The equations derived below provide sufficient information to implement the algorithm which produced the paths shown in figure 1 and figure 13.

3.1 Payload space

The first step of the find-path algorithm is to find a prism (the payload prism) which contains the wrist, hand and payload, with axis of extension parallel to the wrist stem. This can be simply done by taking the convex hull of the projection of these parts into the horizontal plane, then sweeping it up from the base of the payload object through the wrist. See figure 14.

From the discussion of section 2.2 it remains only to find free path segments for the polygonal cross section of that bounding prism through horizontal slices of the workspace.

Brooks [1983] demonstrated a new approach for the problem of moving a two dimensional polygon through a plane littered with obstacle polygons. It was based on two ideas.

(1) Free space can be represented as overlapping "freeways". A freeway is a channel through free space with a straight axis and with a left and right radius at each point. Figure 3 illustrates.

(2) The moving object can be characterized by its *radius function*, defined in figures 15 through 17. The radius function characterizes the left (i.e. $R(\theta + \frac{\pi}{2})$) and right (i.e. $R(\theta - \frac{\pi}{2})$) radii of the object as it is swept along in some direction θ.

The key point is that radius functions and sums of radius functions can be easily inverted (linear in the number of vertices). Thus given a freeway and the radius function of a moving object it is simple to determine the legal range of orientations which lead to no collisions when the object is swept down the freeway.

Finding freeways

This aspect of the complete manipulator-find-path algorithm is the one with most room for improvement.

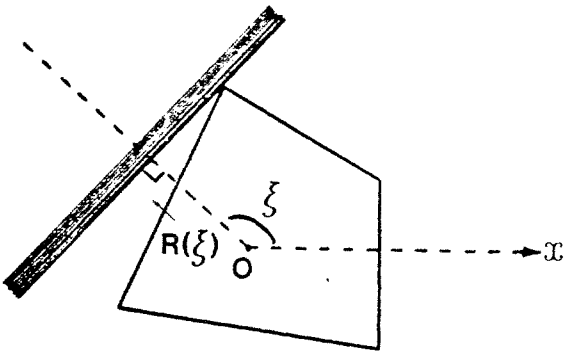

Figure 15. The definition of $R(\xi)$, the radius function of an object.

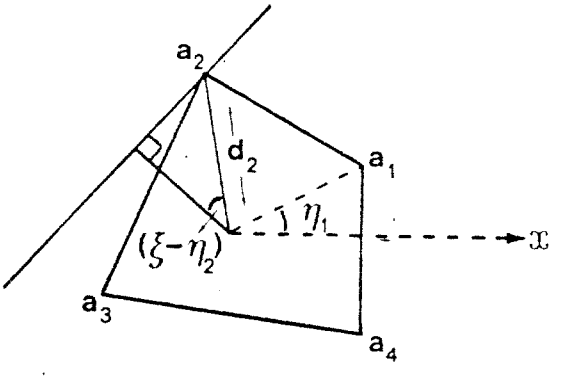

Figure 16. The geometric construction of $R(\xi) = d_2 \cos(\xi - \eta_2)$, (locally).

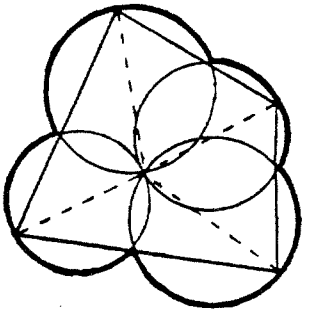

Figure 17. Function R in polar coordinates.

In the implementation described in this paper freeways are currently restricted to having constant left and right radii along their length. Figure 18 shows the "spines" of freeways found at table top level of the scene in figure 1. Figure 19 shows freeways at the next level, and figure 20 shows the top level freeways.

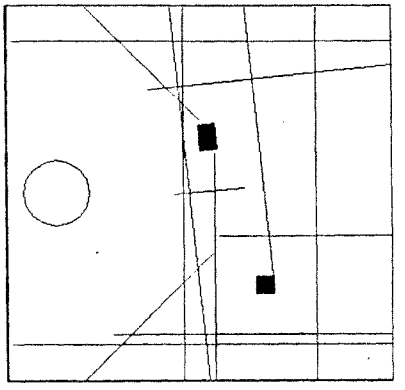

Figure 18. Spines of the freeways for the payload found at table top level of the example of figure 1.

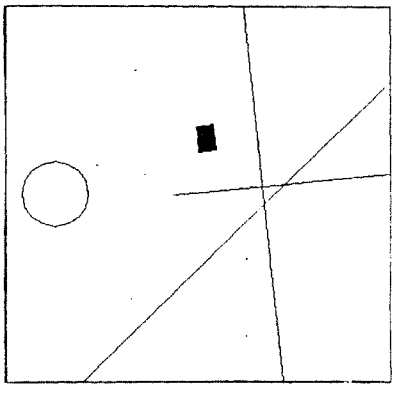

Figure 19. One level up from figure 18. This is a horizontal slice of the workspace above the small obstacle.

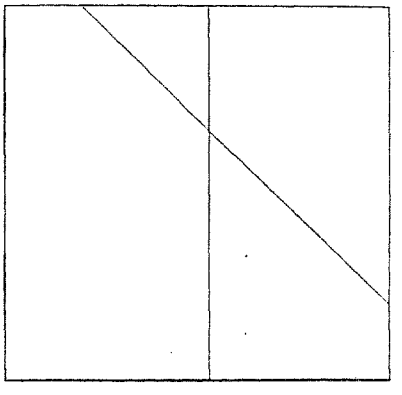

Figure 20. The top of the workspace of figure 1 – now we are above both obstacles.

Within a given horizontal slice of the workspace each edge of each obstacle is used to attempt to generate a freeway. In addition the manipulator upright base is approximated by an octohedron and its edges are used as generators. The generating edge forms one

side of the freeway. A parallel line segment is "moved" out from the chosen edge until an obstacle is hit – this is done by examing all other obstacles in the workspace slice. Often the segment can be safely moved out to infinity. Thus the width of the freeway is determined. Next it is extended lengthwise (i.e. perpendicular to the generating segment) in each direction until an obstacle is reached. Again it may be extensible to infinity. The result is a rectangular freeway, possibly infinite, and with direction parallel to the original generating edge.

The height at which the payload is moved along a freeway is determined by projection of constraints from the upperarm and forearm. The spine, i.e. trajectory, along a freeway is chosen independently of the upperarm and forearm. This is a definite weakness of the algorithm and it may be possible to introduce such interactions with more detailed analysis of the flavor that is presented below for constraints on the payload height.

In the implementation used to generate the examples given in this paper the payload cross section is examined and the following constants computed:

$$r_s = \min_\theta R(\theta)$$
$$r_l = \max_\theta R(\theta)$$
$$w = \min_\theta (R(\theta) + R(\theta + \pi))$$

If the width of the freeway is less than w then the payload can not be moved along the freeway so it can be discarded from further consideration. If the width of the freeway is less than $2r_l$ then then the middle of the freeway is chosen as the spine. Otherwise the freeway is turned into multiple freeways. A spine for one is chosen at a distance $1 + r_s$ away from the original generating edge, with additional spines every $0.5 + r_l$ further out, until the opposite edge is reached.

At a particular height the reach of the manipulator can be determined (see the appendix). Over a horizontal slice of the workspace the minimum reach is used to bound the extent of motion along each freeway. Figure 21 shows the spines of all the freeways found for the situation in figure 1.

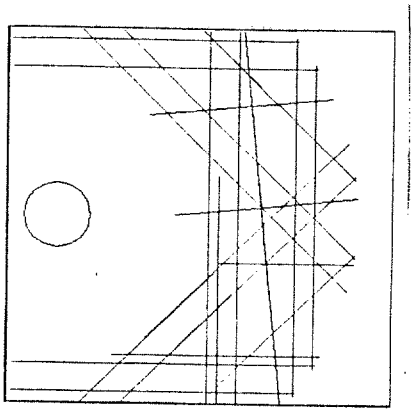

Figure 21. Here we see all the freeways (from figure 1) for the payload, viewed from above. They are limited by the reach of the manipulator.

Legal orientations

All the freeways are checked for intersection in x and y coordinates. The height, or z coordinate, is ignored for this step.

At each point of intersection we can determine for each freeway the legal orientations of the object such that it is contained within the freeway. The original 2-d paper [1983] gives details on this process. Essentially it consists of intersecting set of the form

$$\{\theta \mid R(\theta + \eta) \leq v\}$$

where v is the distance to an edge of the freeway, and η is the orientation of the edge's normal.

If the legal orientations for two freeways at a single point happen to overlap then we can move the payload from one of the freeways to the other at that point. If the legal orientations at two points on the spine of a single freeway overlap then we can move the payload along that freeway from one point to the other without the payload ever protruding out of the freeway.

3.2 Upperarm space

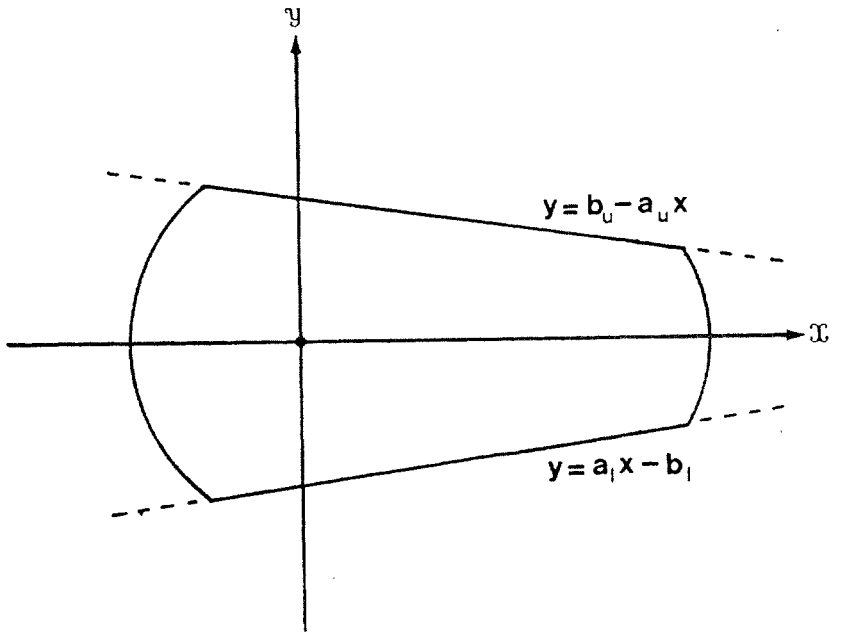

Figure 22. The cross section of the upperarm.

In this section we examine the problem of finding freeways for the upperarm in the ϕ–α space for the manipulator. We first consider obstacles for the upperarm for a fixed ϕ, i.e. for a fixed joint 1 position. Then we develop a method for partitioning the problem in ϕ space and bounding obstacles over those partitions.

Constraints in α–space

Consider figure 22. It is a vertical cross section of the upperarm normal to the axes of joints 2 and 3. The x–axis of the diagram intersects the axes of those two joints, and the y–axis intersects the axis of joint 2. The upper and lower surfaces of the link are flat with equations given by

$$y = b_u - a_u x$$

and

$$y = a_l x - b_l$$

where each of a_u, b_u, a_l and b_l are positive.

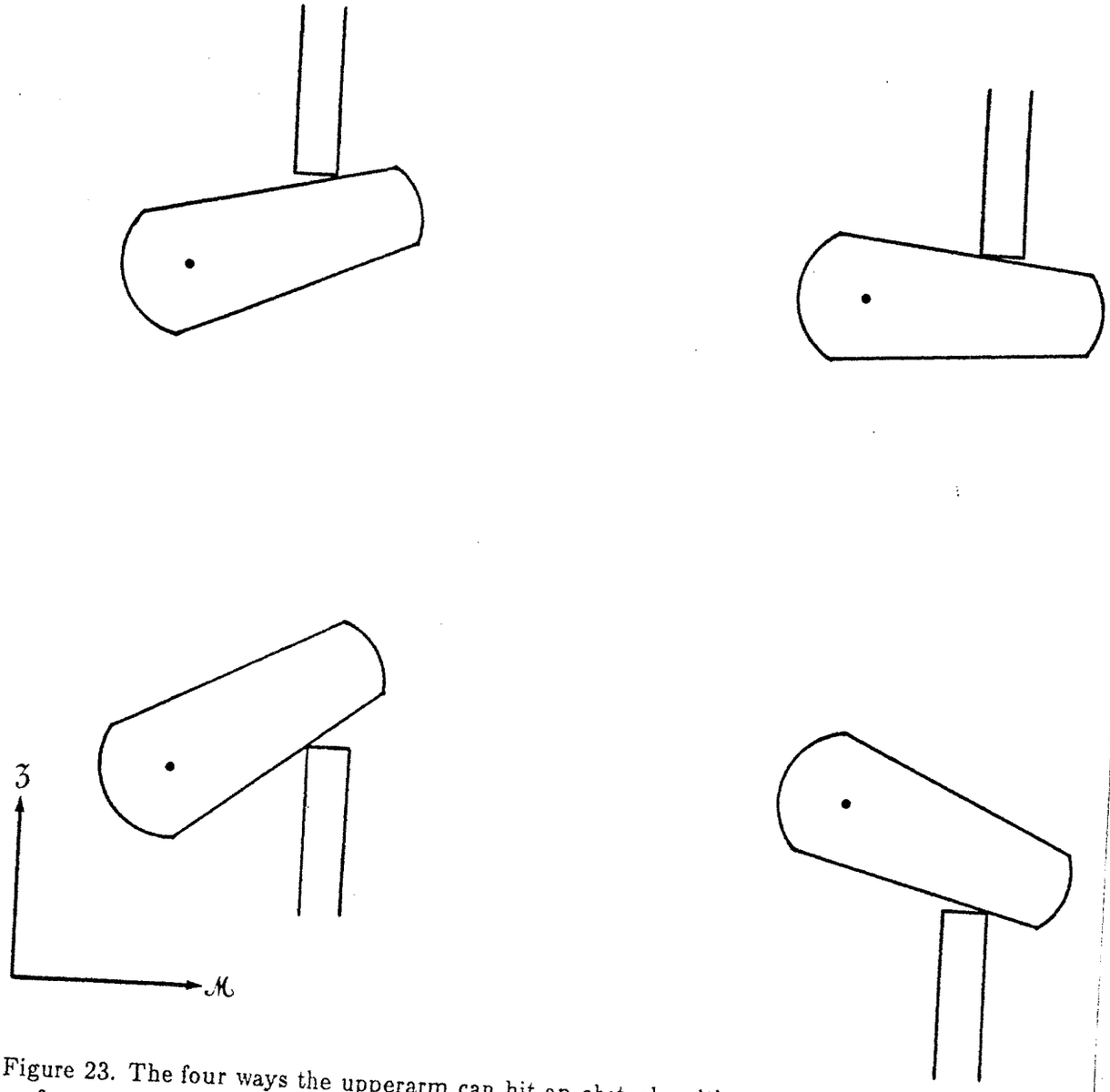

Figure 23. The four ways the upperarm can hit an obstacle with one of its upper or lower surfaces.

Now consider cutting the workspace with a vertical plane so that it intersects such a cross section of the upperarm – see figure 23. The space has axes labeled $\mathcal{M}$ and $\mathcal{Z}$, with the latter vertical, and the point $(0, 0)$ corresponds to the point of rotation for joint 2 of the manipulator. A prism cut by the plane becomes a rectangle, protruding into the workspace from above or below.

Figure 23 shows four possible configurations of an obstacle relative to the upperarm.

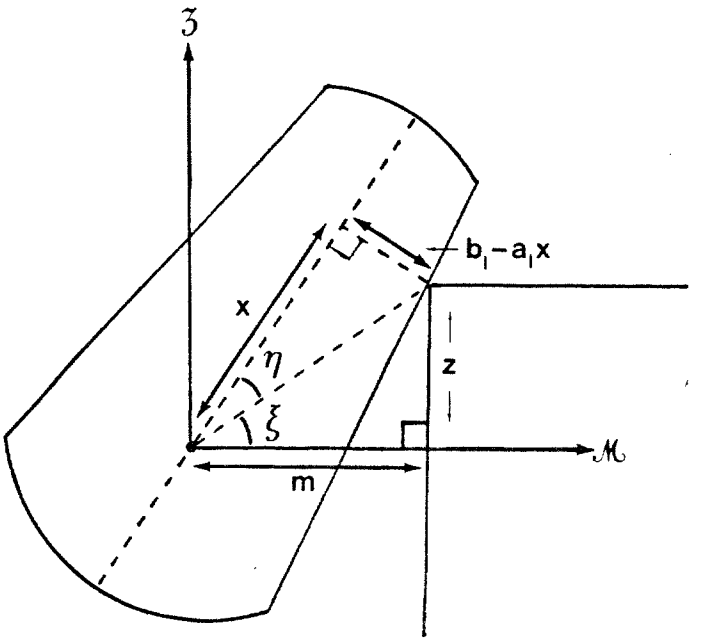

Figure 24. A cross section of the upperarm and its detailed interaction with a prism obstacle.

A hanging obstacle can be touched by the upper surface of the link. The vertex of the obstacle which is touched is determined by the z coordinate of the lower surface of the obstacle. If

$$z > \frac{b_u}{\sqrt{a_u{}^2 + 1}}$$

then the vertex with larger m coordinate can be touched, otherwise the vertex with smaller m. Likewise an obstacle protruding into the workspace from below can be touched by the lower surface of the link. Again the z coordinate is the determining factor as to which vertex can be touched. If

$$z < \frac{-b_l}{\sqrt{a_l{}^2 + 1}}$$

then the vertex with larger m coordinate can be touched, otherwise the vertex with smaller m.

We consider in detail the constraints on α implied by the third of the four cases, and present the results for all four cases below. Consider figure 24. All lengths in the diagram

are labeled with positive values. We can assume for this special case that both ξ and η are between 0 and $\frac{\pi}{2}$. Clearly the presence of the obstacle implies that

$$\alpha > \xi + \eta = \text{atan}(z, m) + \eta.$$

(Note that we use the functional form "atan(s, c)" to refer produce an angle θ such that $\sin\theta = \frac{s}{\sqrt{s^2+c^2}}$ and $\cos\theta = \frac{c}{\sqrt{s^2+c^2}}$.) It remains to determine η. Let

$$r = \sqrt{m^2 + z^2}.$$

Then $x = r\cos\eta$ and so

$$r\sin\eta = b_l - a_l r\cos\eta,$$

whence

$$\frac{b_l}{r\sqrt{{a_l}^2+1}} = \frac{1}{\sqrt{{a_l}^2+1}}\sin\eta + \frac{a_l}{\sqrt{{a_l}^2+1}}\cos\eta,$$

and so

$$\eta = \arcsin\frac{b_l}{r\sqrt{{a_l}^2+1}} - \text{atan}(a_l, 1).$$

Completing the analysis for all four cases reveals the following constraints on α. For an obstacle protruding into the workspace from below

$$\alpha > \text{atan}(z, m) - \text{atan}(a_l, 1) + \arcsin\frac{b_l}{\sqrt{(m^2+z^2)\times({a_l}^2+1)}}, \tag{1}$$

and for one hanging from above

$$\alpha < \text{atan}(z, m) + \text{atan}(a_u, 1) - \arcsin\frac{b_u}{\sqrt{(m^2+z^2)\times({a_u}^2+1)}}. \tag{2}$$

There are three other classes of locations for an obstacle:

(1) An obstacle may not be touched by the upper or lower surfaces of the link, but perhaps it can be touched by the small end of the link. We ignore such collisions as they imply a collision also with the forearm and will be handled as such later.

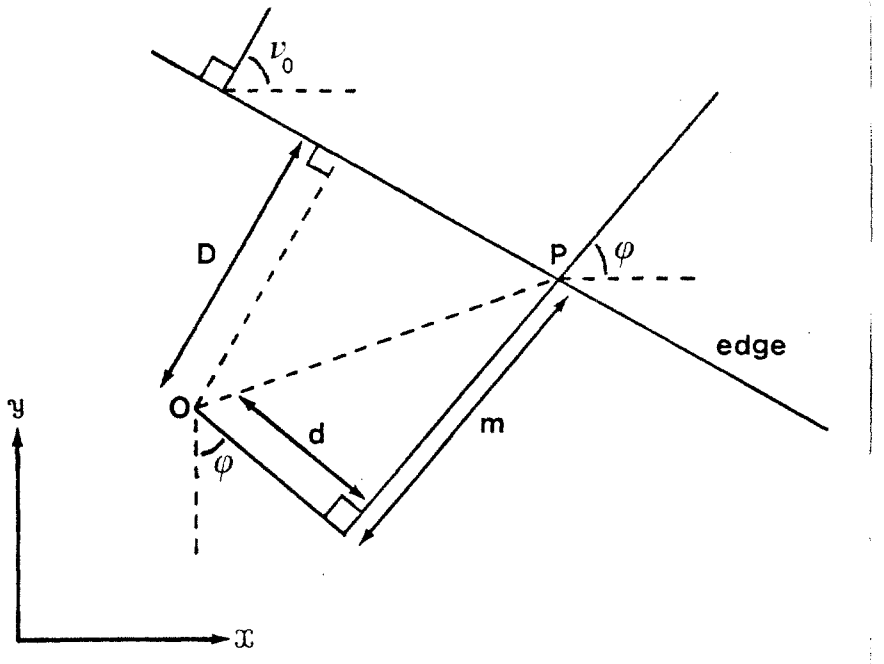

Figure 25. A plan view of the kinematics of the upperarm hitting an obstacle prism.

(2) The same may be true for the large end of the link. However in section 1.2 we explicitly outlawed such collisions by declaring an obstacle free cylinder about the upright base support of the manipulator.

(3) The obstacle may be unreachable by the upperarm. In this case it does not effect the legal motions of the upperarm so we can ignore it. Such an obstacle may restrict motions of the forearm or the payload and it will be dealt with by the appropriate parts of the algorithm.

Constraints in ϕ–space

Refer again to figure 24 and consider varying ϕ and the interaction of the upperarm with a prism edge. Clearly z remains constant, but m can vary.

Now refer to figure 25. This is a plan view of an obstacle and the kinematic linkage

of the manipulator. The upperarm of the manipulator intersects the edge (in plan view). Suppose the edge has a normal with orientation ν_0 and further that the edge has distance D from the origin (the base of the manipulator is centered at the origin). Let d be the displacement of the upperarm from the axis of joint 1 of the manipulator. Then

$$m = \frac{D - d\sin(\phi - \nu_0)}{\cos(\phi - \nu_0)}.$$

Direct analysis of the effects on α of this formulation for m is difficult. We can observe geometrically, however, that m depends on the distance from O to P. The minimum possible OP is D (so long as the upperarm intersects the edge for $\phi = \nu_0$) or the value at one of the end points for the edge. Similarly the maximum can only occur at the end of an edge. When OP has length D then

$$m = \sqrt{D^2 - d^2}.$$

Thus by examining at most three values for ϕ we can determine maximum and minimum values for m.

Now consider the three terms on the right of inequality (1). The second term is constant with respect to ϕ. The third term increases as m decreases, and thus has its maximum over an edge for the minimum m. The behavior of the first term depends on the sign of z. For positive z (i.e. for obstacles higher then the axis of joint 2) the first term has maximum value for minimum m, while for negative z the maximum occurs at maximum m. Thus for a given z we can quickly produce a conservative lower bound (the bound may be larger than the actually achievable α) on α over the range of ϕ where the upperarm might intersect the edge. Similar analysis of inequality (2) leads to a similarly structured conservative upper bound on α over the range of a single prism edge.

For a prismatic obstacle in the workspace we take each top edge between the cross section and the swept sides. Using the test derived above we can determine whether the edge can contact the manipulator. If so we determine conservative α bounds over the ϕ range of the edge. The resulting ϕ–α boxes are then merged and bounded by a single ϕ–α box. For large obstacles with many edges it is better to approximate by a series of adjacent ϕ–α boxes. Figure 7 shows the ϕ–α boxes generated by the scene in figure 1. Notice that

only the overhead obstacle and the larger obstacle on the table are represented. The small obstacle can not be reached by the upperarm and so does not place any constraints on ϕ–α space.

The analysis above has been done for an upperarm that is paper thin. An upperarm with real thickness has infinitely many such cross sections, all of which can contribute constraints on α. Observe however that as the upperarm is rotated downwards towards a horizontal edge then the arm will hit with one of *its* edges first (except when $\phi = \nu_0$, when the lower surface strikes simultaneously across a line segment). Thus we need consider the constraints arising only from the two extreme cross sections (note that they have different d's so the ϕ's which must be considered are different).

Freeways in ϕ–α space

We can easily compute "freeways" in this space and their spines are illustrated in figure 8. Searching these freeways for a path is also trivial, figure 9 illustrates the connected chain of ϕ–α freeways which must be negotiated to solve the problem shown in figure 1.

3.3 Projecting constraints

Our find-path algorithm is based on projecting constraints on motions of the upperarm and forearm to become constraints on the motion of the payload.

Upperarm to payload

In this section we examine the projection of constraints on the upperarm to become constraints on the payload. In section 2.3 we saw a typical (figure 11) motion of the upperarm in ϕ–α space, for a straight line motion of the payload.

Consider the plan view of figure 26. The kinematic linkage of the manipulator is shown

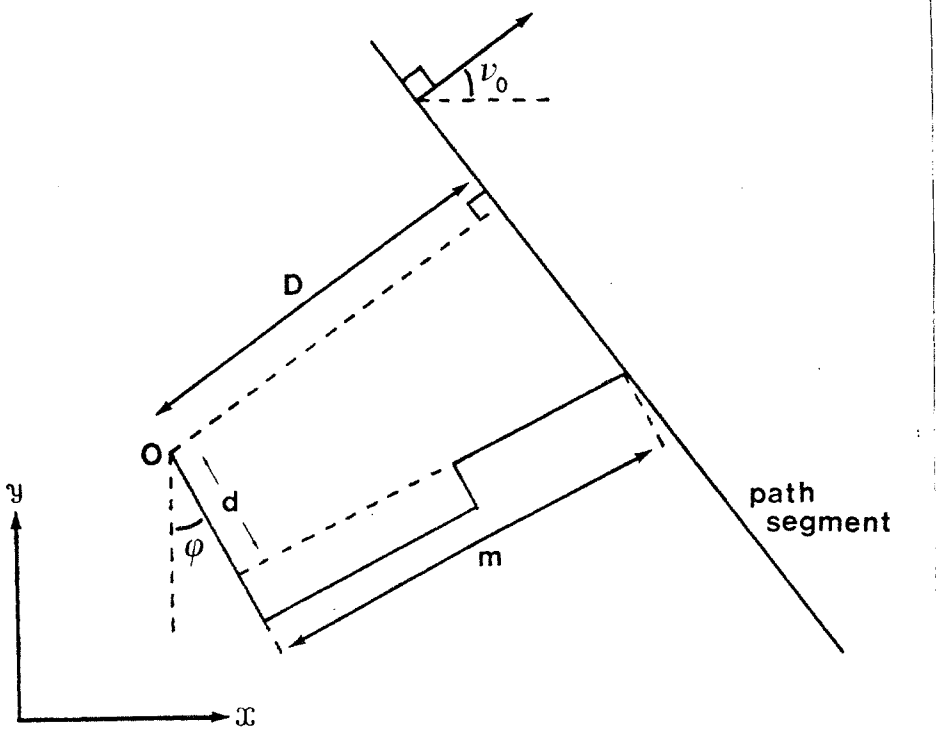

Figure 26. A plan view of the manipulator kinematics in moving the payload along a horizontal straight line segment.

along with a line segment along which the payload is to be moved. The maximal value for α must occur when m is minimal (note that this m and the d from the diagram are different (but similar in concept) from those of section 3.2). That will occur when

$$m = \sqrt{D^2 - d^2}$$

i.e. at

$$\phi_0 = \nu_0 + \text{atan}(d, \sqrt{D^2 - d^2}).$$

Thus the maximal value for α occurs at ϕ_0 if that is in the range of the motion segment or at one of the extremes of the segment. Similar reasoning shows that the minimal value for α must occur at one of the segment extremes. Notice that the points of maxima and minima do not depend on the height of the segment of motion. The values of those maxima and minima will however.

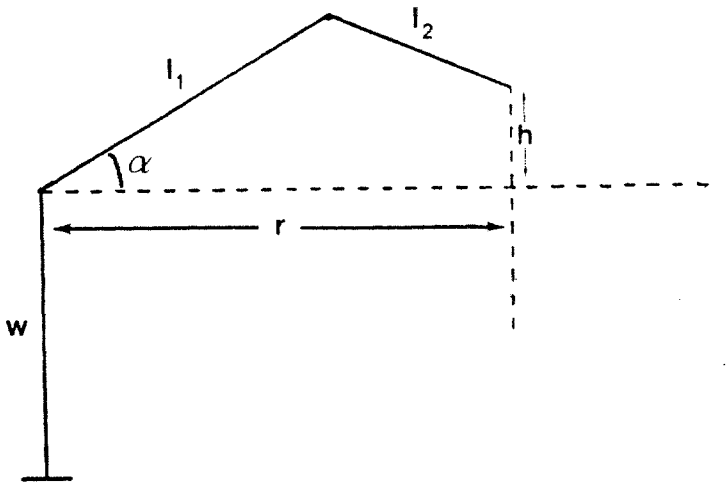

Figure 27. A side view of the manipulator kinematics.

Consider a fixed point (x, y) in the table plane and how the value of α varies as the payload is lifted vertically above that point. Refer to the side view of figure 27. It is a cross section through the arm parallel to the upper and fore arms. Notice first that

$$r = \sqrt{x^2 + y^2 - d^2}.$$

Clearly the maximum achievable α is

$$\alpha_m = \arccos \frac{r - l_2}{l_1}$$

Thus for an interval $[\alpha_1, \alpha_2]$ the minimum and maximum heights which can be achieved by the end of the forearm are given by

$$w + l_1 \sin\alpha - \sqrt{{l_2}^2 - (r - l_1 \cos\alpha)^2}$$

for $\alpha = \alpha_1$ and $\alpha = \min(\alpha_2, \alpha_m)$. From this we can readily compute the bounds on payload height for a particular ϕ while moving along a path segment subject to the constraints of a ϕ–α freeway.

The argument above however implies that by considering the two end points of the motion segment and the point ϕ_0 (when it is interior) and intersecting all the height constraints we are guaranteed a safe set of heights which we can use for the motion along the segment.

Forearm to payload

The forearm can also provide constraints on the legal height of a horizontal straight line motion of the payload. The argument in section 2.3 takes care of vertical motions – it suffices that a vertical motion should be preceded and succeeded by legal horizontal motions.

Note again that this part of the algorithm is not yet implemented in the program which generated the examples for figure 1 and figure 13. As with all unimplemented find-path algorithms (which happens to be the status of many of those published) the reader should be wary of the details.

First consider the problem of constructing a wedge W bounding the swept volume of the forearm as described in section 2.3 (also see figure 12) for a particular horizontal motion of the payload. We will then generalize the construction to include a range of heights.

The shadow prism S can be simply constructed by projecting the path line segment and the extreme sides of the forearm into the table top plane and adding in the fourth side of the quadrilateral. Thus the only real work is in constructing the upper and lower bounding planes of the wedge W.

The bounding planes are parallel to the motion segment and each have one of the four swept edges of the forearm lying in them. The lower bounding plane has one of the lower edges of the forearm, and the upper plane has one of the upper edges. Consider figure 28, the cross section of the forearm. Again a_u, b_u, a_l and b_l are all positive, and

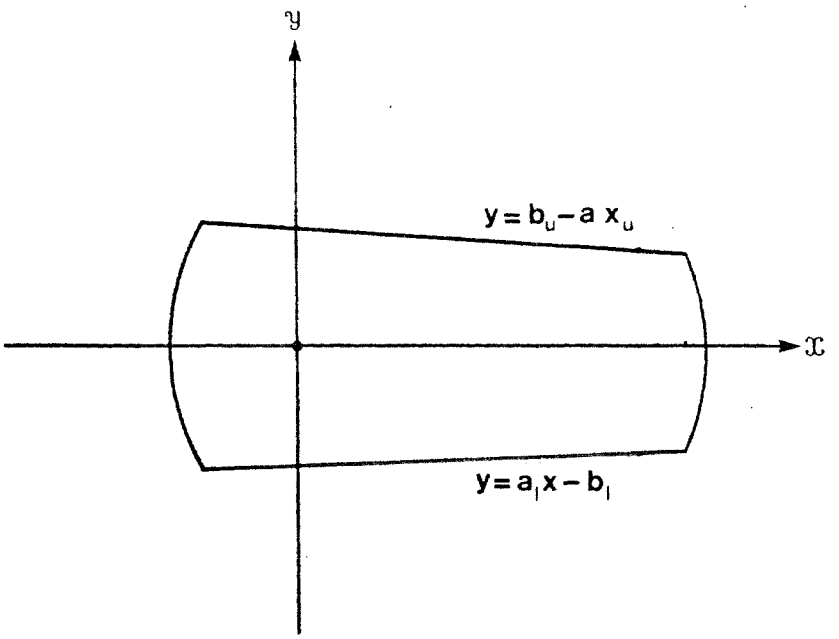

Figure 28. The cross section of the forearm.

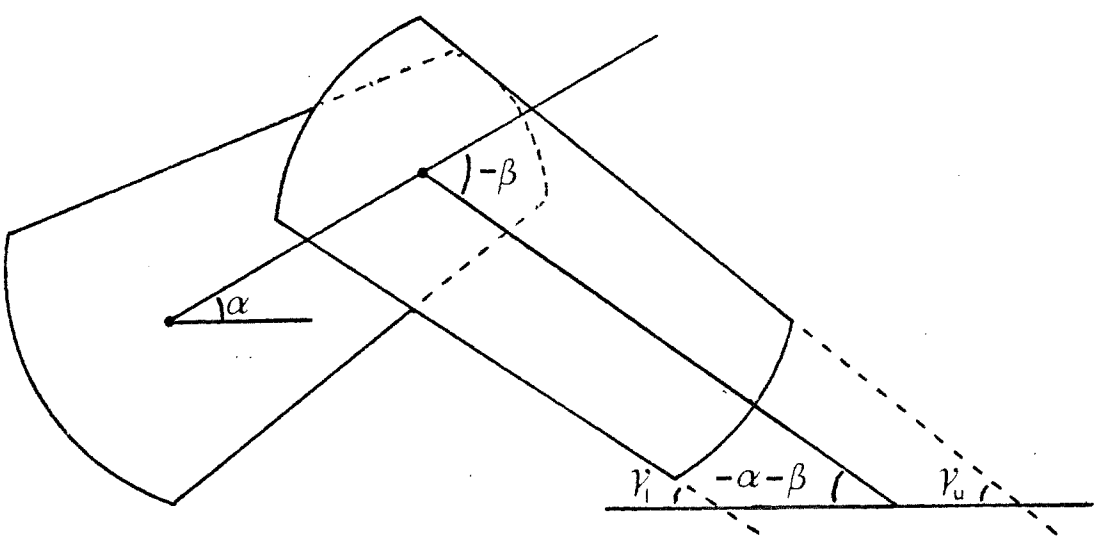

Figure 29. The angles made by the sides of the forearm with a horizontal plane.

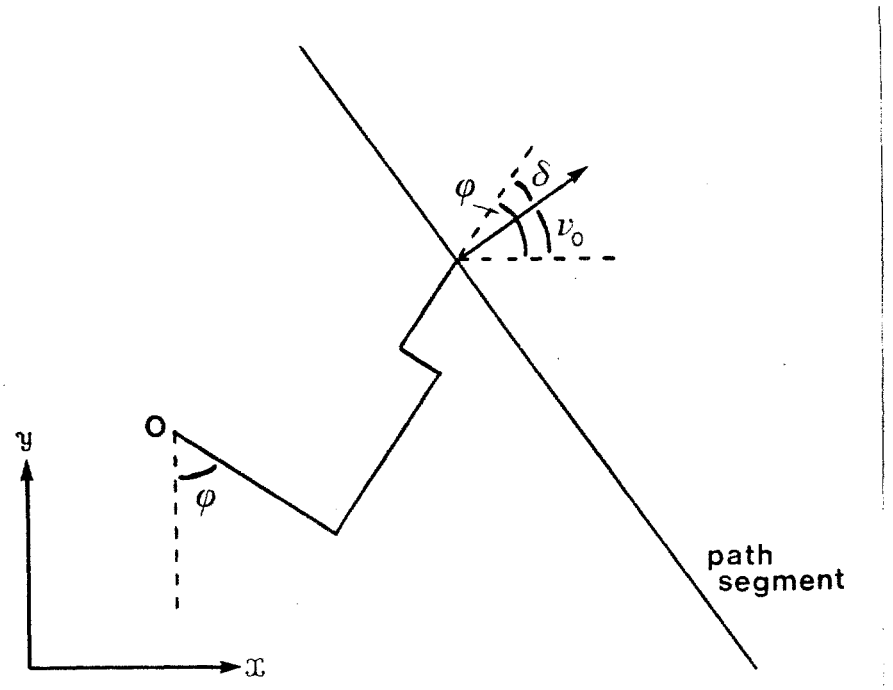

Figure 30. A plan view of the angle between the forearm and a horizontal path segment.

$$\xi_u = \text{atan}(a_u, 1)$$
$$\xi_l = \text{atan}(a_l, 1).$$

Now consider figure 29. All angles are labeled with positive quantities. The upper and lower edges of the forearm make angles

$$\gamma_u = \xi_u - (\alpha + \beta)$$
$$\gamma_l = -(\xi_l + \alpha + \beta)$$

to the horizontal plane.

Consider figure 30. The motion segment has a horizontal normal with orientation ν_0 about the vertical. The first joint of the manipulator has angle ϕ, and hence so does the forearm. Thus the forearm intersects the normal to the motion segment with angle

$$\delta = \phi - \nu_0.$$

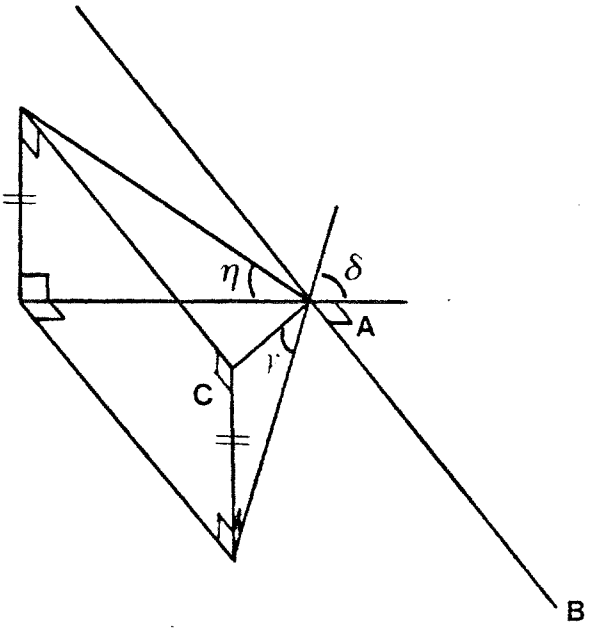

Figure 31. AB is the path segment. AC is an edge of the forearm used to generated a bounding plane.

Now consider figure 31. A plane parallel to line segment AB, but including line segment AC has an orientation η about AB relative to the horizontal. Clearly

$$\tan \eta = \frac{\tan \gamma}{\cos \delta}. \tag{3}$$

Over the length of a path segment we would like to find the η with minimal tangent for the lower surface of wedge W where

$$\gamma = \gamma_l = -(\xi_l + \alpha + \beta),$$

and with maximal tangent for the upper surface, where

$$\gamma = \gamma_u = \xi_u - (\alpha + \beta).$$

Unfortunately both the numerator and denominator of equation (3) move in the same direction as the payload is moved along a line segment. The numerator is maximal when the payload is closest to the manipulator base, i.e. when $\phi = \nu_0$. But that is precisely

when the denominator reaches its maximal 1. Similarly the numerator is minimal at one of the extremes of a line segment, but then δ is maximal and so the the denominator is also minimal.

Over a line segment path it is possible to get bounds on the orientations of the upper and lower planes simply by chopping up the segment into pieces and computing worst case bounds for each of the numerator and denominator of equation (3) and dividing. Since the numerator and denominator are both monotonic away from $\phi = \nu_0$ the accuracy of the bounds can be made arbitrarily good by comparing the values for each at each end of a sub–segment and splitting again if necessary. Thus we can determine a wedge W bounding the swept forearm for a particular horizontal translation of the payload.

If we again examine equation (3) we notice that the denominator does not depend on the height of the payload. Furthermore the numerator is smaller for greater height and larger for lesser height. Thus besides bounding the orientations for the upper and lower surfaces of W over a motion segment at a fixed height we can get weaker bounds which are true for all wedges over a range of heights. Again, to ensure that the bounds are not too weak, we may want to split a given range of heights into a smaller range. Once we have a single wedge W valid over a range of heights we can determine constraints on the legal payload heights by comparing vertices of prismatic obstacles contained in the shadow prism S with the position of the wedge as a function of payload height. This process was explained in section 2.3.

4. Conclusion

By restricting the class of solutions we look for in the general find–path problem for a robot with revolute joints we have developed a practical path planner.

The complex example paths of figure 1 and figure 13 are found in less than 1 minute on an original MIT lisp machine – such machines have no floating point hardware and in general are much slower than, say, a VAX 11/780.

```
Topologically equivalent paths ---

Destination        Height              Orientations
( 20.000,-18.000)
( 18.685,-16.685)  [ 0.000, 2.028]  -- [0.000,0.000]
( 19.360,-16.010)  [ 0.000, 2.028]  -- [0.000,6.283]
( 27.000,-16.010)  [ 0.000, 2.069]  -- [0.000,0.853] union [2.288,3.995] union [5.430,6.283]
( 27.000,  1.576)  [ 0.000, 3.315]  -- [0.718,2.424] union [3.859,5.566]
( 25.463,  1.422)  [ 8.000,23.447]  -- [0.000,0.753] union [2.188,3.895] union [5.330,6.283]
(  9.442, 17.442)  [20.000,25.158]  -- [0.000,6.283]
( 10.000, 18.000)  [20.000,24.982]  -- [0.000,6.283]

Final move list ---

Destination                Orientation
( 20.000,-18.000,  1.000)  0.000
( 18.685,-16.685,  1.000)  0.000
( 19.360,-16.010,  1.000)  0.000
( 27.000,-16.010,  1.000)  0.000
( 27.000,-16.010,  1.000)  0.785
( 27.000,  1.576,  1.000)  0.785
( 27.000,  1.576, 21.000)  0.735
( 25.463,  1.422, 21.000)  0.735
( 25.463,  1.422, 21.000)  0.000
(  9.442, 17.442, 21.000)  0.000
( 10.000, 18.000, 21.000)  0.000
( 10.000, 18.000,  0.000)  0.000
```

Figure 32. Constraints on the path segments for the problem in figure 1 and the finally chosen path.

In figure 1 notice that besides lowering the upperarm to get under the obstacle protruding into the workspace from above, the planner had to rotate the payload so that it could squeeze around the outside of the obstacle on the table top! Figure 32 gives the descriptions of the possible paths for the payload along each freeway in the final path. There is a height range and a set of orientation intervals. Following that is a list of "move locations" for the final path, giving a list of destination positions and orientations for the payload. Between each destination either a pure translation or pure re-orientation suffices.

Figure 13 illustrates a path found when there were no overhead obstacles constraining the upperarm and forearm. With a payload shorter in the vertical direction, the algorithm found a path which simply lifted it over the vertical obstacles and placed it at the goal position. In the illustrated example, however, the arm was unable to reach collision free positions above the obstacles. Nor was it able to move the payload around the obstacles, either on the side towards the manipulator or the other side. The only solution is to move

```
Topologically equivalent paths ---

Destination        Height              Orientations
( 16.000,-18.000)
( 15.651,-14.517)  [ 0.000,19.055]  -- [0.000,0.000]
( 21.597,-13.920)  [ 0.628,19.846]  -- [0.000,0.732] union [5.351,6.283]
( 19.501,  6.975)  [13.000,20.053]  -- [0.639,2.303] union [3.780,5.444]
( 18.017,  6.826)  [10.000,20.139]  -- [0.000,0.732] union [2.210,3.874] union [5.351,6.283]
( 18.000,  7.000)  [10.000,20.136]  -- [0.000,0.660] union [5.423,6.283]

Final move list ---

Destination                 Orientation
( 16.000,-18.000, 14.000)  0.000
( 15.651,-14.517, 14.000)  0.000
( 21.597,-13.920, 14.000)  0.000
( 21.597,-13.920, 14.000)  0.685
( 19.501,  6.975, 14.000)  0.685
( 19.501,  6.975, 14.000)  0.649
( 18.017,  6.826, 14.000)  0.649
( 18.000,  7.000, 14.000)  0.649
( 18.000,  7.000, 14.000)  0.100
( 18.000,  7.000, 10.000)  0.100
```

Figure 33. Constraints on the path segments for the problem in figure 13 and the finally chosen path.

the payload between the vertical obstacles, twisting it so that it will fit. Figure 33 shows the path segment classes and finally chosen path.

Acknowledgements

Tomás Lozano-Pérez provided valuable comments on an earlier draft of this paper.

References

Brooks, Rodney A. (1983). *Solving the find-path problem by good representation of free space,* IEEE Trans. on Systems, Man and Cybernetics (SMC–13):190–197.

Brooks, Rodney A. and Tomás Lozano–Pérez (1983). *A Subdivision Algorithm In Configuration Space For Findpath With Rotation,* IJCAI–83, Karlsruhe, Germany.

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

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

Nevins, J. L. and D. E. Whitney (1978). *Computer Controlled Assembly,* Scientific American 238, 62–74..

Nishihara, Keith (1983). *in progress,* MIT Artificial Intelligence Lab.

Schwartz, Jacob T. and Micha Sharir (1982). *On the Piano Movers Problem II: General Properties for Computing Topological Properties of Real Algebraic Manifolds,* Department of Computer Science, Courant Institute of Mathematical Sciences, NYU, Report 41, February.

Udupa, Shriram M. (1977). *Collision Detection and Avoidance in Computer Controlled Manipulators,* Proceedings of IJCAI–5, MIT, Cambridge, Ma., Aug. 1977, 737-748.

Widdoes, L. Curtis (1974). *Obstacle avoidance.,* A heuristic collision avoider for the Stanford robot arm.Unpublished memo, Stanford Artificial Intelligence Laboratory.

Appendix – PUMA solution

Since we restrict our manipulator to only four degrees of freedom, the backward kinematic solution (given a position and orientation of the payload then determine the joint angles) is much simpler than the general solution.

Figure 34 shows plan and side views of the PUMA kinematics given our restrictions. Note that the line representing the forearm does not run along the axis of joint 4 of the six degree of freedom PUMA. Here we use the line joining the axes of joints 3 and 5. There are fixed offsets between our three angles ϕ, α and β and the actual joint angles.

The solution we use follows from the diagram. Given a point (x, y, z) in table top

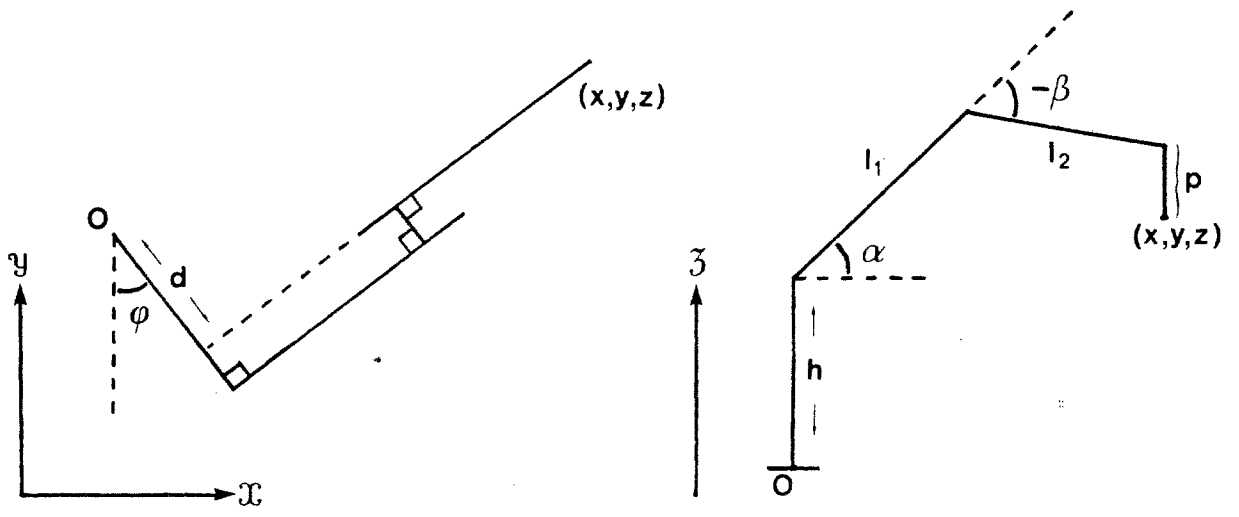

Figure 34. The kinematics of the PUMA regarded as a four degree of freedom device.

coordinates (the base of the manipulator is at $(0, 0, 0)$) for the payload reference point, we compute:

$$
\begin{aligned}
w &\leftarrow z + p - h \\
e &\leftarrow x^2 + y^2 - d^2 \\
r &\leftarrow \sqrt{e} \\
c &\leftarrow \frac{e + w^2 - {l_1}^2 - {l_2}^2}{2 l_1 l_2} \\
s &\leftarrow -\sqrt{1 - c^2} \\
a &\leftarrow l_1 + l_2 c \\
\phi &\leftarrow \mathrm{atan}(y, x) + \mathrm{atan}(d, r) \\
\alpha &\leftarrow \mathrm{atan}(aw - l_2 sr, ar + l_2 sz) \\
\beta &\leftarrow \mathrm{atan}(s, c)
\end{aligned}
$$

in sequence. If e is negative then the point is not reachable and there is no solution. The quantity r is the horizontal component of the distance from the manipulator base to the payload reference point. The quantity c corresponds exactly to $\cos\beta$, so if it is outside the range —1.0 to 1.0 there is again no solution. In order to ensure an "up elbow" solution, s, which is $\sin\beta$, is forced to be negative. The three joint angles then follow.