FIND-PATH FOR A PUMA-CLASS ROBOT

Rodney A. Brooks

MIT Artificial Intelligence Laboratory
545 Technology Square
Cambridge, Massachusetts, 02139, U.S.A.

ABSTRACT

Collision free motions for a manipulator with revolute joints (e.g. a PUMA) are planned through an obstacle littered workspace by first describing 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.

I. 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 PUMA). It is based on a method presented at AAAI-82 for planning motions for a polygon through a two dimensional workspace (Brooks [1983a]).

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.

A key characteristic of our solution is that it solves a richer class of problems than merely finding safe paths for a manipulator with payload. On failure it can provide information to a higher level planner on how to alter the workspace so that it can find a solution to the new problem.

A. Previous approaches

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 Shamir [1982] have demonstrated the existence of a polynomial 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 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.

B. The problem to be solved

The algorithm presented below is not a complete solution to the find-path problem. It is restricted in the following ways.

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 for a six degree of freedom PUMA, joint 4 is kept fixed (a 5 *dof* PUMA has no joint 4), and joint 5 is coupled to the sum of the angles of joints 2 and 3 so that the axis of joint 6 is always kept vertical. 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 modelled 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 upper-arm and fore-arm.

The class of motions allowed suffice for most 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.

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.

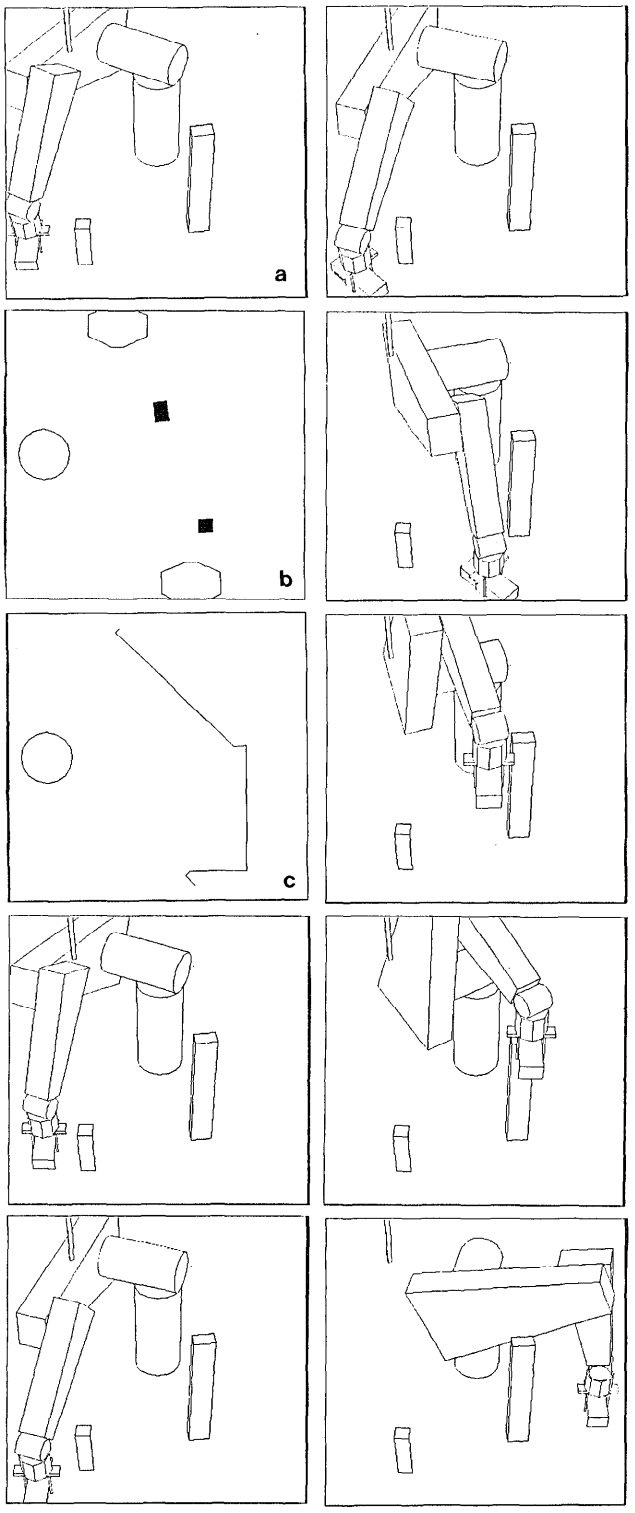

Figure 1. A path found by the algorithm. Part (a) is the initial configuration. Part (b) is a plan view (rotated $\frac{\pi}{2}$). Part (c) is a plan view of the payload path. The remaining images (top to bottom, left to right) show the path.

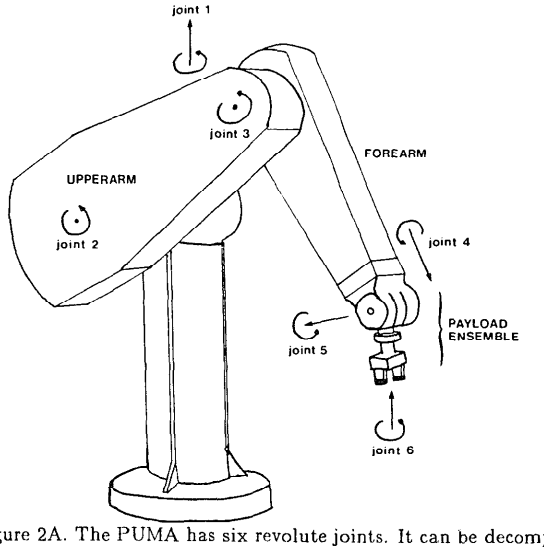

Figure 2A. The PUMA has six revolute joints. It can be decomposed into three components: the upperarm, the forearm, and the combined wrist, hand and payload.

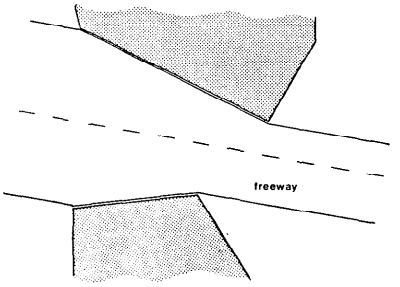

Figure 2B. A freeway is an elongated piece of free space which describes a path between obstacles.

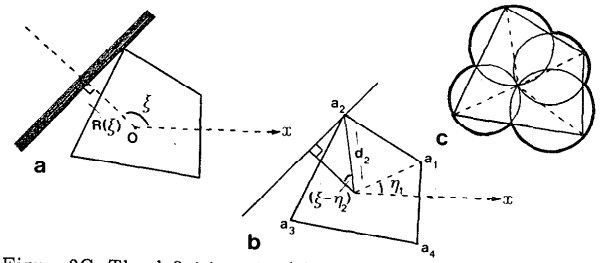

Figure 2C. The definition of $R(\xi)$, the radius function of an object, (a). Part (b) shows the geometric construction of $R(\xi) = d_2 \cos(\xi - \eta_2)$, (locally). Part (c) shows function R in polar coordinates.

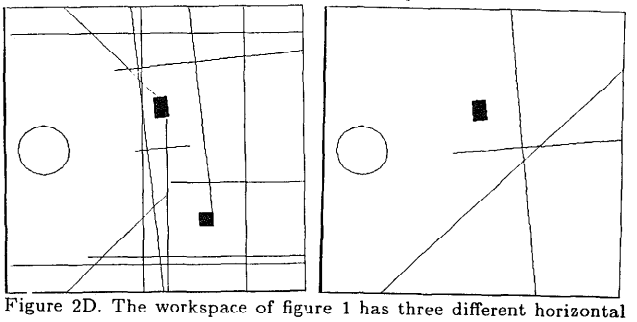

Figure 2D. The workspace of figure 1 has three different horizontal cross sections. Here we see the spines of freeways for the payload in the lowest cross section in which they are valid.

II. Payload Space

Figure 2A is a side view of a PUMA robot grasping an object. During gross motion we treat the wrist, hand and payload as a single object. It can translate through space (joints 1, 2 and 3 provide the motion while joint 5 compensates for orientation changes to keep the wrist stem axis vertical) and rotate about the vertical axis (joint 6).

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.

Since obstacles must be supported completely and motions will only be horizontal or vertical it follows that it suffices to find a path for the bottom cross section of the payload prism.

A. The problem in two dimensions

Brooks [1983a] 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 2B illustrates.

(2) The moving object can be characterized by its *radius function*, defined in figure 2C. 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. 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.

In the implementation described in this paper freeways are currently restricted to having constant left and right radii along their length. The first part of figure 2D shows the "spines" of freeways found at table top level of the scene in figure 1.

B. Adding vertical variations

There are at most as many different horizontal cross sections through the workspace as there are obstacles. We can apply the algorithms of Brooks [1983a] to each different cross section and derive a set of freeways valid over some horizontal slice of the workspace.

Since obstacles must always be completely supported by others below it is true that any freeway valid at some height h_0 is also valid for all $h \geq h_0$. Thus successively higher cross sections will inherit freeways from below. Figure 2D shows a series of cross sections through the example workspace of figure 1. These are plan views, rotated $\frac{\pi}{2}$ from the original scene. Each cross section includes the spines of all freeways which are valid for the first time at that height. Their validity extends vertically upwards through the remainder of the workspace. (Recall that we assume that the hanging obstacles do not protrude into the payload workspace.)

III. Upperarm Space

The upper-arm (see figure 2A) 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 upper-arm is sticking out horizontally.

The class of possible motions of the arm can not be easily characterized as translational constraints 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 upper-arm 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 $R(\theta)$ function used for the payload).

A. Constraints in α-space

Consider figure 3A: a cross section of the upper-arm and an obstacle for a particular fixed ϕ. All lengths in the diagram are labelled with positive values. The presence of the obstacle puts constraints on the valid range of angles of joint 2. Clearly the presence of the obstacle implies that

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

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

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

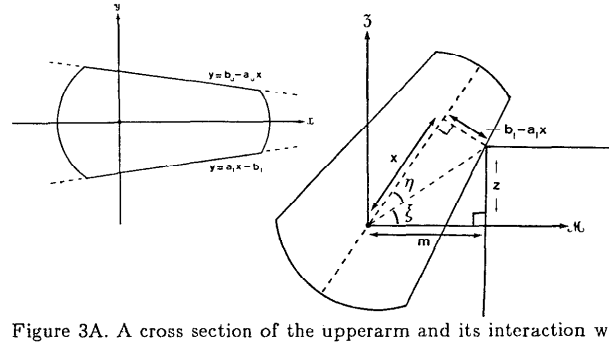

Figure 3A. A cross section of the upperarm and its interaction with a prism obstacle. Equation (3) is the constraint derived from this figure.

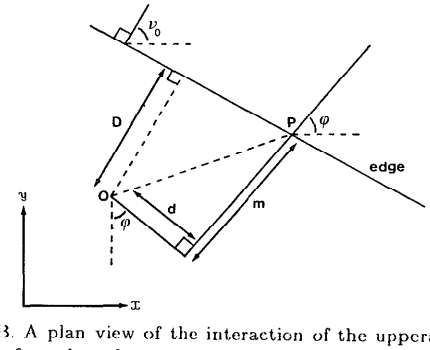

Figure 3B. A plan view of the interaction of the upperarm and the top edge of an obstacle prism.

and so

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

whence

$$\alpha > \operatorname{atan}(z, m) - \operatorname{atan}(a_l, 1) + \arcsin \frac{b_l}{\sqrt{(m^2 + z^2) \times (a_l{}^2 + 1)}}. \quad (3)$$

Due to the restrictions we have placed on our models of objects in the world we can conclude that vertices such as v in figure 3A must be points on the top edges of prisms. The problem to be solved is to determine how m and z vary with changing ϕ.

The inequality, with appropriate sign changes, is valid for obstacles intruding into the workspace from above the arm also.

We can ignore collisions with the end of the upper arm as such a collision would also result in a collision for the fore-arm. We find it more convenient to catch such collisions under our fore-arm analysis.

B. Constraints in ϕ–space

Refer again to figure 3A and consider a varying ϕ and the interaction of the upper-arm with a prism edge. Clearly z remains constant, but m can vary.

Refer to figure 3B. Suppose the edge has a normal with orientation ν_0 and further that the edge has distance D from the origin. Let d be the displacement of the upper-arm from the axis of joint 1 of the arm. 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. However we can observe geometrically that m depends rather directly on the distance from O to P. The minimum possible OP is D (so long as the upper-arm 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 (3). 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 second term has maximum value for minimum m, whilst for negative z the maximum occurs at maximum m. Thus for a given z we can quickly produce a conservative bound on α over the range of ϕ where the upper-arm might intersect the edge. Detailed analysis shows that for most obstacles this conservative bound is very good.

For a prismatic obstacle in the workspace we take each top edge and determine conservative α bounds over a ϕ range for each edge which "faces" the base of the manipulator. The resulting ϕ–α boxes are then merged and bounded by single ϕ–α box. For large obstacles with many edges it is better to approximate by a series of adjacent ϕ–α boxes. The Fig 3C series 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 upper-arm and so does not place any constraints on ϕ–α space.

The analysis above has been done for an upper-arm that is paper thin. An upper-arm with real thickness has infinitely many such cross sections, all of which can contribute constraints on α. Observe however that as the upper-arm 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).

C. Freeways in ϕ–α space

We have successfully reduced the collision avoidance problem for the upper-arm in real space to one of an path finding problem for a point in a space populated with rectangular obstacles aligned with the axes of the space. Furthermore, since we allow only prismatic obstacles stacked on each other on the table, or hanging from above, there can be no "free-floating" obstacles in ϕ–α space.

We can easily compute "freeways" in this space and their spines are illustrated in the second part of figure 3C. Searching these freeways for a path is also trivial and the third part of 3C illustrates the connected chain of ϕ–α freeways which must be negotiated to solve the problem shown in figure 1.

IV. Projecting Constraints

Our find-path algorithm is based on propagating constraints on motion of the upper-arm, fore-arm and payload through the searches for paths for these three physical components.

In this section we examine the propagation of constraints on the upper-arm to become constraints on the payload. Consider the plan view of figure 4A, and the problem of moving the payload along a horizontal path segment.

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 III). That will occur when

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

i.e. at

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

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.

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 plan and side views of figure 4B. The side view 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 fore-arm 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 whilst 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.

V. Search

Paper length constraints preclude a detailed description of the final search process. In essence path planning proceeds as follows.

A path through ϕ–α space is found for the upper-arm. It is a list of ϕ–α freeways. Now a depth first search is done through payload space under the guidance of this list of constraints on α (as detailed in the previous section).

Notice that the upper-arm puts constraints on the height of travel for the payload within a particular freeway. If a freeway section is chosen in depth first search, and these constraints provide a non-empty range of heights then the path segment is certainly valid for the payload and upper-arm. The problem of fore arm collisions remains.

Such collisions are checked for each path segment during depth first search. Observe (figure 5) that over a given vertical line segment for the payload in the workspace, the fore-arm is closest to horizontal for the top of the segment, and more acute at the bottom. Furthermore for a particular height as we travel along a horizontal segment the arguments of section IV apply also to the fore-arm. Thus we can easily compute a bounding volume for the swept volume for the fore-arm travelling along the maximal height allowed on a segment. We project that swept volume down onto the table plane and determine which prisms it "shadows" from above. The prisms are intersected with this volume and then each prism upper vertex has a distance below the fore-arm swept volume. If any distance is negative then the motion along the horizontal segment is completely forbidden. Otherwise the minimal distance gives a safe bound on how far we can lower the payload during traversal of the horizontal segment without producing a collision for the fore-arm.

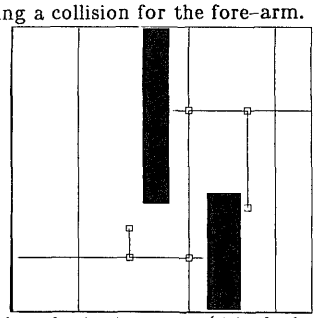

Figure 3C. The obstacles in ϕ–α space (ϕ is the horizontal axis and α the vertical) derived from the workspace of figure 1. The small obstacle in figure 1 is out of the reach of the upperarm so does not appear here. The freeways in this space and a planned path instance for the upperarm are also shown.

VI. 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 path of figure 1 is 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. Notice that besides lowering the upper-arm 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!

References

Brooks, Rodney A. (1983a). *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. (1983b). *Planning Collision Free Motions for Pick and Place Operations,* MIT AI-Memo 719, May.

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

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

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., 737-748.

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

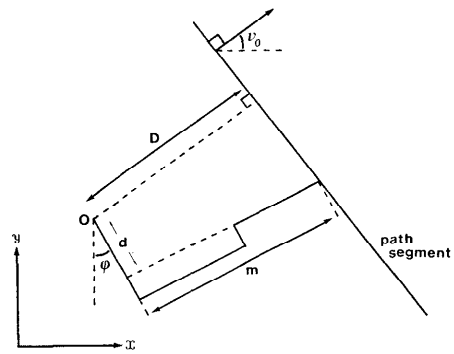

Figure 4A. A plan view of the manipulator kinematics in moving the payload along a horizontal straight line segment.

Figure 4B. A side view of the manipulator kinematics.

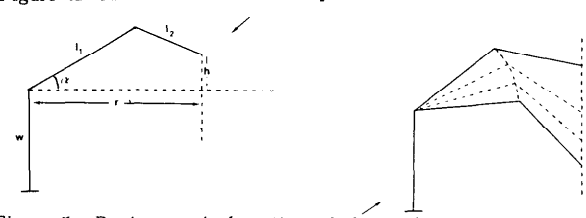

Figure 5. During vertical motion of the payload the manipulator forearm is closest to horizontal at the upper bound of payload travel and more acute at the lower bound.