COLLISION DETECTION AND AVOIDANCE IN COMPUTER CONTROLLED MANIPULATORS

Shriram M. Udupa
California Institute of Technology
Pasadena, California

ABSTRACT - The problem of planning safe trajectories for computer controlled manipulators with two movable links and multiple degrees of freedom is analyzed, and a solution to the problem proposed.

The key features of the solution are:

1. the identification of trajectory primitives and a hierarchy of abstraction spaces that permit simple manipulator models,
2. the characterization of empty space by approximating it with easily describable entities called charts - the approximation is dynamic and can be selective,
3. a scheme for planning motions close to obstacles that is computationally viable, and that suggests how proximity sensors might be used to do the planning, and
4. the use of hierarchical decomposition to reduce the complexity of the planning problem.

KEYWORDS - Manipulator, planning, representation, heuristics, abstraction.

1. INTRODUCTION

The problem of planning safe trajectories for computer controlled manipulators with two movable links and multiple degrees of freedom is analyzed, and a solution to the problem is presented.

The trajectory planning system is initialized with a description of the part of the environment that the manipulator is to maneuver in. When given the goal position and orientation of the hand, the system plans a complete trajectory that will safely maneuver the manipulator into the goal configuration. The executive system in charge of operating the hardware uses this trajectory to physically move the manipulator.

The solution is applied to a simplified two-dimensional manipulator (2D system) and a full-fledged three-dimensional manipulator - the Scheinman arm - (3D system). All the examples in this paper are from the 2D system. Once the 2D solution is understood and similarities between the 2D and 3D solution noted, it is easy to visualize the solution for the 3D manipulator.

Section 2 of this paper presents an example, and Section 3 a statement and analysis of the problem. Sections 4 and 5 present the solution. Section 6 summarizes the key ideas in the solution and indicates areas for future work.

2. AN EXAMPLE

This section describes an example (Figure 2.1) of the collision detection and avoidance problem for a two-dimensional manipulator. The example highlights features of the problem and its solution.

2.1 The Problem

The manipulator has two links and three degrees of freedom. The larger link, called the boom, slides back and forth and can rotate about the origin. The smaller link, called the forearm, has a rotational degree of freedom about the tip of the boom. The tip of the forearm is called the hand. S and G are the initial and final configurations of the manipulator. Any real manipulator's links will have physical dimensions. The line segment representation of the link is an abstraction; the physical dimensions can be accounted for and how this is done is described later.

The closed polygons in the figure represent polygonal approximations to obstacles; these polygons may be concave or convex, and there is no limit to the number of sides.

The problem is to plan a collision free trajectory that will get the manipulator from S to G. (A trajectory specifies the manipulator configuration as a function of time).

2.2 The Solution

Since the boom is much larger than the forearm, the boom is the more constraining of the two links. Therefore, a safe boom trajectory is first planned, and then the forearm is maneuvered safely along the boom tip locus. If for some reason, the forearm cannot be safely maneuvered, the boom trajectory is revised and the process repeated.

Boom Planning: The boom planning problem is to find a trajectory for the two joints associated with the boom. We can try to get the boom tip from S to G along the shortest path

*
Present Address
Bell Telephone Laboratories
Naperville, Illinois 60540

between the two boom tip locations - a straight line boom tip locus. In Figure 2.2, the shaded area represents the area that the boom sweeps when its tip traces a straight line from S to G. Since the shaded area intersects the L-shaped object, the boom will collide with that object. This collision can be avoided if an intermediate point P is chosen and the boom tip is required to go through P. The above procedure is then applied recursively to the sections SP and PG until a safe trajectory is found. Figure 2.3 shows the final boom tip locus that guarantees boom safety.

Forearm Planning: This refers to finding a trajectory for the forearm joint. The basic idea is to juggle the forearm back and forth so that it avoids any collisions as one end of it travels along the boom tip locus. This is not easy. Further complications arise because the maneuverability of the forearm near the goal configuration is very restricted. So as not to clutter the diagram, the forearm tip locus has been supressed from Figure 2.3.

Execution: The above planning results in a sequence of intermediate configurations leading to the goal configuration. The trajectory calculation routines use this sequence to generate a trajectory. The executive system in charge of operating the hardware uses the trajectory to move the manipulator.

Embellishments: The planning described is called mid-section planning. This mode of planning does not make use of the nature of obstacle configurations, and is good for planning maneuvers far from obstacles. Another mode of planning, called terminal phase planning, makes use of the nature of obstacle configuration around the hand, and can be used for planning maneuvers near the start and goal configurations. The use of both the mid-section and the terminal phase planning results in a simpler trajectory (a smaller sequence of intermediate configurations). Figure 2.4 shows the boom tip locus for the simpler trajectory.

The representation of the manipulator, obstacles, empty space and trajectories, the two planning modes and the associated heuristics, the use of the nature of obstacle configurations for terminal maneuvers etc. are all discussed in Sections 4 and 5.

2.3 Historical Perspective

Collision avoidance problems became manifest when computer controlled manipulators came into existence during the sixties. Pieper (1968) was the first one to investigate the problem. Paul (1972) tackled trajectory calculation and servoing. Lewis (1974) attacked a very restricted version of the collision avoidance problem, and Widdoes (1974) did the same. None of the earlier attempts can handle complexities similar to the ones illustrated in the example of Figure 2.1. The representations used by these earlier programs are inconvenient, or the planning strategies inadequate for handling the situation.

3. THE PROBLEM AND ITS ANALYSIS

The problem is to plan a trajectory to get a computer controlled manipulator with two movable links and multiple degrees of freedom safely into a desired goal configuration.

The planning system is initially given a description of the environment in which the manipulator is to operate. This environment undergoes minor changes when objects in it are transported around by the manipulator. The environment may also change drastically, as would happen if the robot (to which the manipulator belonged) moved to a new place. It is assumed that such drastic changes are infrequent compared to the total number of trajectories planned. This assumption is referred to as the infrequent environment initialization hypothesis.

The input to the planning system consists of the position and orientation of the hand in the goal configuration. The output is a list of intermediate configurations that will be used by the trajectory calculation programs to run the hardware.

A solution that will perform well in simple and commonly occuring situations is desired. The system should recognize when things go awry and should ask for human assistance when that happens. Optimal plans are not needed; at the same time, however, blatantly stupid plans are not permitted.

Planning begins with hypothesizing a trajectory. Following this is an iterative step that involves a check for collision and a trajectory modification (if there is danger). Under normal circumstances, the loop terminates when a safe trajectory is found.

Good heuristics for hypothesizing trajectories are essential. Ideally, the system should propose a collision-free trajectory on the first try. Since a trajectory designed to pass through large empty spaces is likely to be safe, a characterization of large empty spaces is desirable. Also, a characterization of obstacle configurations in the immediate vicinity of the hand is desirable; for, special heuristics, that increase the chances of proposing a collision-free trajectory, can be associated with these configurations.

Good techniques for making trajectory modifications are required; the modifications should ensure that the same problem does not recur and that new problems do not arise.

A few terms that will enable us to talk about the collision detection problem are now introduced.

The manipulator's state can be described either as a vector specifying its joint angles or as a position and orientation of the hand. The former is a representation in joint variable space or joint space, and the latter is a representation in cartesian space. The subspace of joint space generated by the boom joints is called boom space. When the manipulator moves its links trace a volume (surface, in two dimensions) called the trajectory envelope (see Figure 2.2).

Collision dectection involves checking intersections of the trajectory envelope and obstacles. Trajectory envelopes are most conveniently described in joint space and obstacles in cartesian space. Consequently, intersection checking requires constant transformations between the two spaces. This is expensive and we need to determine whether obstacles should be represented in joint space or trajectories in cartesian space, or whether it is possible to use the two spaces effectively and avoid the transformation problem. We need to see whether trajectory envelopes and obstacle descriptions can be simplified since it would lead to inexpensive intersection checks. Again, safe trajectory planning can be viewed as maneuvering in free space or as avoiding obstacles. Can these complementary views be used to advantage?

The solution presented in the next two sections provides answers to all the points and questions raised in the above analysis, making fast collision avoiders a distinct possibility.

4. REPRESENTATION

Trajectory planning gets easier with simple trajectory envelopes. Simple envelopes are possible only with well-chosen trajectory primitives and simple manipulator models. Simple manipulator models make the task of hypothesizing and modifying trajectories easy. And simple trajectory envelopes and numerically manageable obstacle representations make collision detection inexpensive. With this in mind, polyhedra models of obstacles, a hierarchy of abstraction spaces permitting simplified manipulator models and useful trajectory primitives are introduced in this section. Their use in planning is presented in section 5.

Starting with a simple and direct model of two connected cylinders, the abstraction spaces permit the manipulator to be modelled as two connected line segments, as a single line segment and, incredibly, as a point! Furthermore, the transformations generating the abstraction spaces are inexpensive. This is important; otherwise, the advantages gained by operating in these alternate spaces would be lost in the process of generating them.

The manipulator and its environment are described first, the three problem spaces next and finally the trajectory primitives are presented. Table 1 summarizes the relationship between the different problem spaces.

4.1 The Manipulator And Its Environment

The manipulator of Section 2 is an abstraction of the class of computer controlled manipulators with two movable links and multiple degrees of freedom. The Scheinman Arm shown in Figure 4.1 is another example. Details on the hardware and algorithms for trajectory calculation and servoing of this manipulator are described in Paul (1972), Dobrotin and Scheinman (1973) and Lewis (1974). The manipulator is a six degree freedom device allowing the hand to be positioned anywhere (within the maneuverable space) and with any orientation.

Figure 4.1 shows the six joints and the links between the points. Linkl is called the post, link2 the shoulder, and link3 the boom. Link4 and link5 are non-existent because the manipulator design has the last three joints at the tip of the boom. Link6 is called the forearm. Except for joint3, which is a sliding joint, all the joints are revolute. Jointl is called phi, joint2 theta, joint3 r, joint4 f theta, joint5 f_ phi, and joint6 *f*_ psi. The prefix "f" indicates that the angles refer to the forearm. The forearm tip is called the hand.

When looking along the boom at the hand, the boom is either on the right or the left side of the shoulder. This gives rise to the notion of a right-handed and left-handed manipulator respectively, and is called the lateral property. To simplify the presentation, the lateral property of the manipulator will henceforth be ignored. This concludes the description of the manipulator.

The term environment will be used to denote the set of obstacles in the workspace of the manipulator. Since the hand can touch the manipulator post, the post, too, is considered an obstacle. Obstacles can be both regular and irregular in shape. Objects on a robot such as the wheels, the TV rack, the platform etc. are regular and can be described as cylinders, parallelopipeds, or unions of these shapes. Boulders and rocks in the maneuverable space would be examples of irregular shaped objects.

4.2 Real Problem Space

The real problem space is a simple-minded and direct computer representation of the manipulator and its environment. The manipulator is modelled as a sequence of connected cylinders, one each for the post, boom and forearm. The boom and forearm representations correspond to the minimum bounding cylinders that enclose them. The trajectory envelope is, consequently, a two-element three-dimensional solid.

Obstacles are approximated by polyhedra - plane-faced objects. There is no restriction on the number of faces, and both concave and convex polyhedra are allowed. Better approximation implies more faces, and, consequently, more storage required to save the description and more time required to analyze collisions. Thus there is a trade-off involved. However, the representation is compact and intersection checks are inexpensive because obstacle surfaces are linear. The set of polyhedra, each approximating a real obstacle, is called a map.

The maneuverable space is the complement of the volume occupied by elements of the map, with respect to the manipulator's work space.

4.3 Primary Problem Space

The primary problem space permits the manipulator to be viewed as consisting of a single line segment and having no lateral property.

First consider a two-line segment model of the manipulator. The finite axes of the cylinders bounding the boom and forearm are used for this model. The equivalence between the primary problem space and real problem space is preserved by enlarging the polyhedra by the radius of the manipulator links. The enlarged polyhedra are a called primary obstacles. The set of primary obstacles is called the primary map. With line-segment models of the manipulator links, the trajectory envelope is two connected surfaces, one called the boom surface and the other the forearm surface. The maneuverable space is called primary free space and is the complement of the volume occupied by primary obstacles with respect to the manipulator's workspace. Note that the enlargement of obstacles needs to be done just once for a given environment.

Finally, the single line segment description of the manipulator is made possible by a transformation called survey. Survey permits the boom to be viewed as a single point, and the trajectory envelope then reduces to the forearm surface - generated by the motion of the forearm. The transformation survey is applied to free space and results in a chart. A chart for primary free space in called a primary chart.

Consider the set of all points in free space such that the entire boom is safe from collision if the boom tip were positioned there. This subset of free space is called navspace (navigational space). The survey transformation approximates navspace by boxes in r-theta-phi space called regions and the set of regions is called a chart.

Regions are structured entities. They are made up of sectoroids and (in 3D) sectorolds are composed of pases. The pasc (parallelopiped in spherical coordinates) is the smallest unit. Figure 5.1 shows six 2D regions bounded by the radial arrows; it also shows one of the regions (R) and its four component sectoroids - S1,S2,S3 and S4. Pases, sectoroids and regions are bound by constant phi and constant theta surfaces. All pases in a sectoroid have the same phi limits, and all sectoroids in a region have the same theta limits. Pases, sectoroids and regions all have associated with them a maximum and minimum r value, called rmax and rmin respectively, indicating the safe limits of the boom extension. The difference between the maximum and minimum r values is called the safe limit interval. A region, sectoroid or pasc is considered impassable if the safe limit interval is less than some prespecified value. These items are illustrated in Figure 5.1.

Regions are an approximation to the points in navspace. This approximation is dynamic and can be changed by higher-level programs. The approximating procedure is called refinement, and the refinement level is called resolution. The initial approximation is done to a default resolution. If the resolution of a particular part of the environment is not adequate, the system further refines that portion of the chart. This is termed the selective refinement capability. This capability makes incremental modifications (necessitated by minor changes to the environment) to the charts inexpensive. Since refinement is dynamic, survey is not a one-time transformation. This is the price that has to be paid for the flexibility. Since there is a limit to the precision of placement of the hardware, the process of refinement will not continue indefinitely.

The concept of navspace permits considering the boom as a single point. Navspace and its approximation by charts is thus crucial to safe trajectory planning. The reason for imposing a structure on charts (in terms of regions, sectoroids and pases) is to have, some selectivity in terms of what parts of navspace should be refined and to what level. It is important to note that the exact nature of a region and its components is irrelevant, and the choice of a box in r-theta-phi space as the unit was dictated by the choice of a particular planning strategy described in Section 5.

4.4 Secondary Problem Space

The secondary problem space permits the manipulator to be viewed as a single point.

To get the single point description of the manipulator, let us start with the two-line segment model of the manipulator. We ignore the forearm, and account for it by enlarging the primary obstacles by the length of the forearm; this enlargement results in secondary obstacles and a secondary map, and the maneuverable space is called secondary free space. With the forearm accounted for, the manipulator consists of only the boom and the trajectory envelope is the boom surface.

It is now possible to arrive at the single point description of the manipulator. We apply the survey transformation to secondary free space. This results in a secondary chart, composed of secondary regions. Whenever the boom tip is in a secondary region the following holds:

1. by definition of the region, the entire boom is free of collision, and

2. since secondary regions are generated using secondary obstacles, the forearm is free from collision irrespective of its orientation.

The trajectory envelope at this level then is the line generated by the motion of the boom tip. A complex trajectory solid that consisted of two solids has thus been reduced to a line. The refinement process for secondary charts is similar to primary charts and so are all the attributes and transformations discussed in the context of primary charts.

If the manipulator needs to maneuver close to obstacles, secondary problem space is of no use since the 'gross' representation of the forearm engulfs free space near obstacles. Of course this does not imply that a trajectory does not exist. The finer model of the forearm as a line segment (as in primary problem space) should be used.

Looked at from a different angle, the ideas of secondary problem space representations (the secondary charts in particular), are a formal characterization of the intuitive ideas of ease of maneuvering in large chunks of empty space. The simplification of the trajectory envelope from two solids to a line makes the expectation come true.

4.5 Trajectory Primitives

Since obstacle faces are planes in cartesian "space, if the trajectory envelope were a plane (primary problem space) or a line (secondary problem space) in cartesian space, collision checking would be simple.

Since the manipulator joints can be operated independently, the boom tip can be made to trace cartesian space straight lines. Planning the cartesian space straight line locus for the boom tip is easy in the 2D case, while it is beset with problems in the 3D case. Hence in the 3D case, we settle for boom space - subspace of joint space generated by the boom joints - straight line locus for the boom tip.

Safety of the boom tip locus implies the safety of the entire manipulator only when the locus passes through a secondary chart; elsewhere the safety of the forearm needs to be ensured. To make forearm safety checks tractable, the following are chosen as primitives for the forearm trajectory.

1. When the boom is moving, the hand shall trace a line parallel to the cartesian space straight line approximation of the boom tip locus; this is called pgram motion (for parallelogram motion).

2. When the boom is stationary, the forearm shall move in a single plane; this called circle motion (2D) or sphere motion (3D).

These constraints on the boom and forearm trajectories result in the decomposition of the trajectory surface into a sequence of parallelograms and sectors of a circle, considerably simplifying the collision detection task.

5. PLANNING

The process of planning a trajectory was discussed in Section 3; the aim is to plan a safe trajectory, and plan it fast.

Hierarchy, separability and reversibility are the key concepts in planning. The principle of reversibility states that if a trajectory from S to G is collision free then the same trajectory backwards from G to S is also collision free. Separability means the decomposition of the goal into independent subgoals. Hierarchy is used in the usual sense. For each goal, the most important aspects are tackled first and the lesser ones next. This is applied to every stage of the process. If some decisions made at a higher level do not pan out, local corrections are made. If the local fixups do not solve the problems, the system returns to the next higher level for replanning. At each stage it is ensured that the system will terminate its activities in a finite amount of time. If the system is unsuccessful in solving the problem, it gives up and asks for human help.

5.1 Overview Of Planning

The natural approach to hierarchical planning is to plan in secondary problem space first (since the trajectory envelope is the simplest there) and then refine the trajectory in primary problem space. The difficulty of interfacing the two problem spaces makes this approach unattractive. So instead, planning is "done" in primary problem space and secondary problem space is used for simplifications. The details of this approach are presented next.

The trajectory planning problem is separated into three phases. The first is a goal feasibility analysis phase, the second is the mid-section planning phase and the last is the terminal planning phase. At the feasibility analysis stage, goal feasibility is checked and any necessary refinements of the charts is carried out. The terminal phase activities use the reversibility principle and plan trajectories near the initial and final configurations. The mid-section phase deals with midway trajectory planning. For the terminal phase, forearm and boom planning iterate until a satisfactory boom tip location for starting the mid-section trajectory is found. For the midsection, planning proceeds hierarchically. Boom trajectory is first planned using the primary charts alone. For portions of the boom tip that do not lie in the secondary chart, forearm planning is done. The separability principle is used in boom planning; the trajectory for the theta-phi joints is planned first and the r -joint is fixed next.

If a safe forearm trajectory cannot be found, the boom trajectory is modified and another attempt at forearm planning made. If the system is unable to come up with a safe trajectory even after a prespecified number of attempts, it resorts to a configuration switch. The same techniques are used to plan a trajectory to get the manipulator to the goal, this time however, in a different lateral configuration. If this also fails, the system gives up.

Note that planning incorporates simple strategies. It may so happen that the system fails to find a solution when there exists one in the real world. It is unlikely that such situations will be encountered except in some pathological obstacle configurations.

5.2 Initialization

The system is initialized with a description of the environment. The system uses the input polyhedra and generates primary and secondary obstacles for the left and right, secondary and primary maps. All the charts are generated for a default resoultion (see Figure 5.1, for example). The regions of the charts will be further refined as and when necessary. The initialization needs to be done once for every new environment.

5.3 Goal Feasibility and Impossible Situations

Goal feasibility is done before planning begins. It includes boom placement and forearm placement safety checks. It determines whether the boom tip lies within a pasc of a primary region. If not, the appropriate region is repeatedly refined until either the goal boom tip position is within a pasc or the resolution limit is reached and the system returns complaining that the goal is not feasible. The forearm feasibility study involves checking whether in the final configuration, the forearm is safe from collision; if not, the goal is not feasible. Figures 5.2 and 5.3 are refinements of Figure 5.1 to get points S and G inside the charts. The environment is the one used in the example of Section 2.

During mid-section phase boom planning, the system keeps a watch for situations which would get the boom stuck (see Figure 5.4, for example). If the boom cannot maneuver out of an area, the system complains. Again, during forearm planning along a proposed boom tip locus, the system looks out for situations which would get the forearm stuck (see Figure 5.5, for example). Such situations are called impossible situations.

5.4 Mid-Section Planning

Boom Planning

Boom planning is equivalent to finding the path of a point through charts. Cartesian space straight line locus for the boom tip are used in the 2D system. The features of the 2D system and the required extensions for the 3D system are presented.

Point path planning is based on an adaptation of a well-known algorithm used for approximating a curve by a sequence of straight lines; the approximation is such that every point on the curve is within a specified distance from the line segment (approximating the portion of the curve the point is on). The recursive algorithm is illustrated in Figure 5.6. Point C is the farthest point on the curve from the approximating straight line AB. If the distance of C from AB exceeds the tolerance limit, the curve is split at C and the algorithm applied recursively to sections AC and CB.

The above algorithm works even if different thresholds are used for different parts of the curve. Each component (region, sectoroid, or pasc (in the 3D case)) has associated with it an rmax and rmin that specify the safe interval limit, within which the boom tip must lie when it goes through that component.

Figure 5.7 shows the working of the modified linear approximation algorithm. The dotted lines show adjacent regions of a chart and their safe limit intervals. Every pair of regions has a nontrivial intersection of their (rmin, rmax) interval. S and G are the start and goal boom tip locations. The arrow in the Figure shows the point on SG that is farthest from being inside the regions. A subgoal P is introduced in the region contributing to the violation and the algorithm is recursively applied to SP and PG.

Figure 5.8 shows how adjacent regions Rl and R2, which have trivial (rmin, rmax) interval intersections are handled by the introduction of additional subgoals PO and Pl. Note that the line along P0P1 is safe from the rmin of R2 to rmax of Rl.

A further generalization of the simple recursive curve-approximation algorithm is to make the approximating line segments be any desirable curve. In fact, for the 3D system this generalization is used to plan a boom tip locus that is linear in the boom joint angles.

The above planning procedure is first applied at the region level, then at the sectoroid level and finally at the pasc level.

In the 3D manipulator, the system first plans a trajectory in the theta-phi space making only certain minimal checks on the safe limit intervals of the regions through which the trajectory passes. Once this is done, the r-joint planning is done using the generalized linear approximation algorithm.

Forearm Planning

Forearm planning is just a sequence of applications of the two primitives for the forearm trajectory. Figure 5.9 illustrates circle motion; it shows the angular interval the forearm can move in, the most favored orientation of the forearm for the given direction of travel of the boom, the forearm orientation chosen and the boom tip locus SG. Figure 5.10 illustrates pgram motion; it shows the direction of travel (SG being the boom tip locus) the initial forearm orientation and the parallelogram generated by the forearm motion.

5.5 Terminal Phase Planning

Terminal phase planning plans trajectories close to obstacles, primarily near the start and the goal configurations. The strategy consists of planning pairs of adjust and move motions. A sequence of such pairs of motions puts the boom tip at a safe point, from which the mid-section strategies take over. A safe point, is a point in a secondary pasc, or if there is no secondary pasc with a reasonable safe interval then it is a point in a primary pasc whose safe limit interval exceeds a prespecified value.

During the move motion the boom tip moves along a line collinear with the forearm and away from the hand, and the forearm maintains its orientation in cartesian space. This motion continues until either the boom tip reaches a safe point or a potential collision is recognized.

The adjust motion orients the forearm to reduce chances of collision during the subsequent move motion. Figure 5.11 shows adjust motion heuristics - the "binary" choice of favorable orientations - for the 2D case. The numbers indicate the sequence in which these orientations will be tried. For a particular orientation, if it turns out that the subsequent move motion makes no progress, the next orientation in the sequence is tried. If the manipulator joint angles remain unchanged, even after a prespecified number of tries, the system returns a failure'.

In 3D a 3 x 3 square of solid angles is determined about the current forearm orientation. Each square is 5 degress in size, and has a 0 or 1 associated with it according as it is safe or not for the forearm to maneuver within the solid angle represented by the square. The 2D adjust motion strategy is applied in the plane taking the forearm through the center of the safe solid angle interval. If more than one solid angle square is safe, obvious generalizations of the binary choice searching strategy can be tried.

Instead of using software for gathering information for planning motions close to obstacles, proximity sensors can be mounted on the forearm to provide this information. The logic for analyzing realtime data from an array of such sensors is simple; the logic incorporates the simple adjust motion heuristics described above.

6. CONCLUSIONS

A solution to the safe trajectory planning problem for computer controlled manipulators with two movable links and multiple degrees of freedom was presented. The solution treats manipulators with a sliding joint, and permits transporting of objects which can be enclosed within the minimum bounding cylinder of the manipulator link. For modifications of the solution that permit handling larger objects, extensions to the solution for treating manipulators with only rotary joints, and details on how to account for the lateral property of the manipulator, the reader is referred to Udupa (1976). A significant portion of the ideas described here have been implemented in SAIL on a DEC PDP-10 computer. The output of the collision detection and avoidance system has yet to be interfaced with a real manipulator.

6.1 Key Ideas

1. Simplified Manipulator Descriptions And Trajectory Primitives: Alternate problem spaces of increasing abstraction that permit simplified manipulator models and primitive trajectory types are identified; these simplify collision detection and trajectory hypothesis and modification.

2. Navspace and Charts: The concept of navspace that permits the reduction of the boom to a single point is identified. Odd-shaped navspace is approximated by easily describable entities called charts; the approximation is dynamic and can be selective, thus permitting easy incremental modifications to the charts.

3. Transformations for generating the primary and secondary maps and charts need to be computed only once or a few times; otherwise the advantages of using the alternative problem spaces would have been offset by the expensive computations required to generate them.

4. Trajectory Planning In Empty Space vs. Collision Avoidance: These two complementary views can be used to advantage in the safe trajectory planning problem. Boom planning is treated as planning trajectories in empty space (the charts), and forearm planning is treated as a collision avoidance problem.

5. Cartesian Space vs. Joint Space: By decomposing planning into boom and forearm planning and maneuverable space into navspace

ELEMENTS IN WORLD / PROBLEM SPACE	MANIPULATOR	OBSTACLES/ ENVIRONMENT	EMPTY SPACE	TRAJECTORY ENVELOPE
SECONDARY	SINGLE POINT (BOOM TIP)		SECONDARY REGION/ SECONDARY CHART	LINE
	SINGLE LINE SEGMENT (BOOM)	SECONDARY OBSTACLE/ SECONDARY MAP	SECONDARY FREE SPACE	SURFACE (BOOM)
PRIMARY	SINGLE LINE SEGMENT (FOREARM)		PRIMARY REGION/ PRIMARY CHART	SURFACE (FOREARM)
	TWO LINE SEGMENTS (BOOM AND FOREARM)	PRIMARY OBSTACLE/ PRIMARY MAP	PRIMARY FREE SPACE	TWO CONNECTED SURFACES (BOOM AND FOREARM)
REAL	TWO SOLID SEGMENTS (BOOM AND FOREARM)	POLYHEDRA/ MAP	MANEUVERABLE SPACE	TWO CONNECTED SOLIDS (BOOM AND FOREARM)

TABLE 1 THE REPRESENTATION HIERARCHY

and space occupied by obstacles, the advantages of the representations in these alternate spaces can be capitalized on.

6. Planning: Hierarchical decomposition, different strategies for maneuvering far from obstacles and for maneuvering close to obstacles, and a formal characterization of large chunks of empty space all simplify the planning task.

7. Planning At Execution Time: Guidelines have been suggested for incorporating proximity sensors into the manipulator system for doing terminal phase planning.

6.2 Suggestions For Future Work

Transporting objects comparable in size to the manipulator, collision avoidance for multiple manipulators, handling of a richer class of constraints (keep the hand vertical during motion, for example), and other manipulator hardware designs (telescoping manipulators, for example) are some topics that need investigation. Collision avoidance in humanoid manipulators (all rotary joints) can be handled by a slight extension of the solution presented in this paper.

Little is known about modifying trajectories dynamically based on any sensory data the system may acquire during execution. Further investigations on the use of proximity sensors, force and tactile sensors, and visual feedback to simplify planning is required.

7. ACKNOWLEDGEMENTS

The research reported here was done in connection with the Jet Propulsion Laboratory's (JPL) Robotics Research Program. I would like to express my sincere appreciation to Sankaran Srinvas, Scott Roth and Meir Weinstein of California Institute of Technology, and Len Friedman and Bill Whitney of JPL for the many thoughtful discussions, and their concern and encouragement at various stages of the research reported here. 1 would like to thank Ed Ballard of Bell Labs for reviewing a draft of this paper.

REFERENCES

1. Dobrotin, B. M. and Scheinman, V. D., "Design of a Computer Controlled Manipulator for Robot Research", 3IJCAI, Stanford University, August 1973.

2. Lewis, R. A., "Autonomous Manipulation in a Robot: Summary of Manipulator Software Functions", JPL Technical Report 33-679, March 1974.

3. Paul, R., "Modelling, Trajectory Calculation and Servoing of a Computer Controlled Arm", PhD thesis, Stanford University, November 1972.

4. Pieper, D. L., "The Kinematics of Manipulators Under Computer Control", PhD thesis, Stanford University, October 1968.

5. Udupa, S. M., "Collision Detection and Avoidance in Computer Controlled Manipulators", PhD thesis, California Institute of Technology, September 1976.

6. Widdoes, C, "A Heuristic Collision Avoider for the Stanford Robot Arm", Stanford C. S. Memo 227, June 1974.

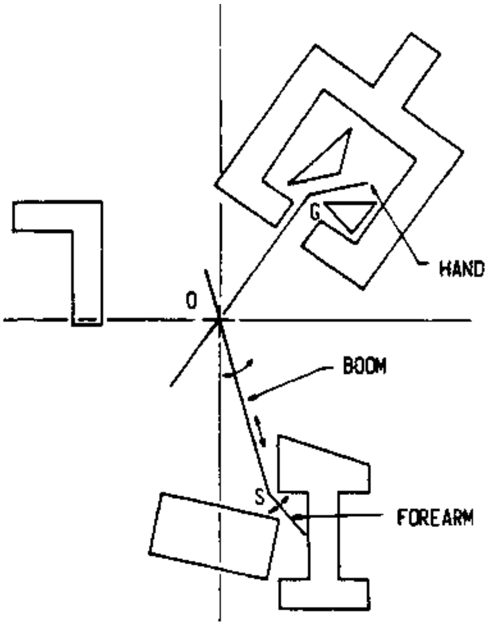

FIGURE 2.1 AN EXAMPLE

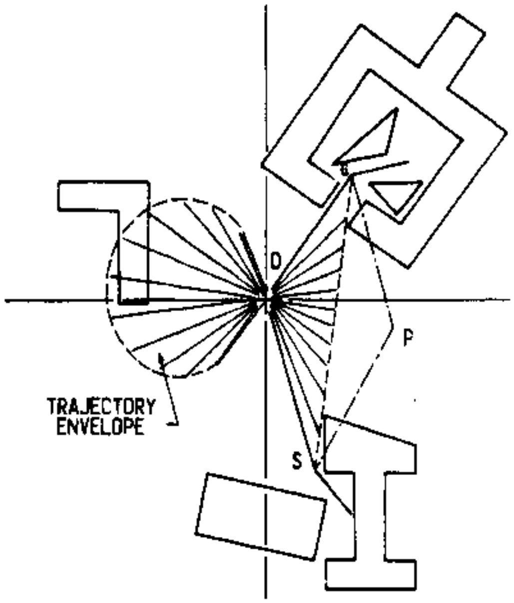

FIGURE 2.2 BASIC IDEAS

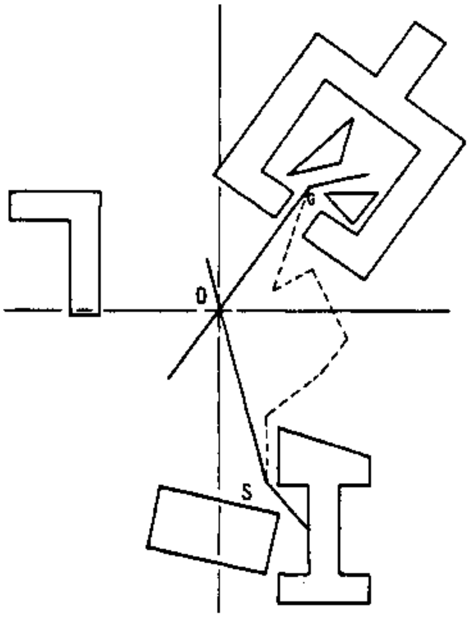

FIGURE 2.3 BOOM TIP LOCUS

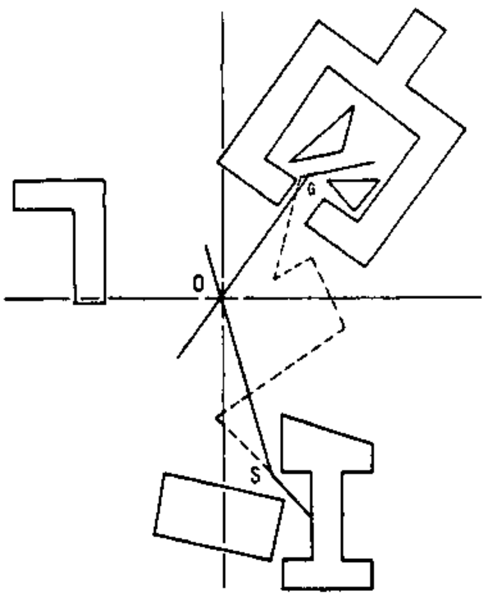

FIGURE 2.4 BOOM TIP LOCUS WITH TERMINAL PHASE PLANNING

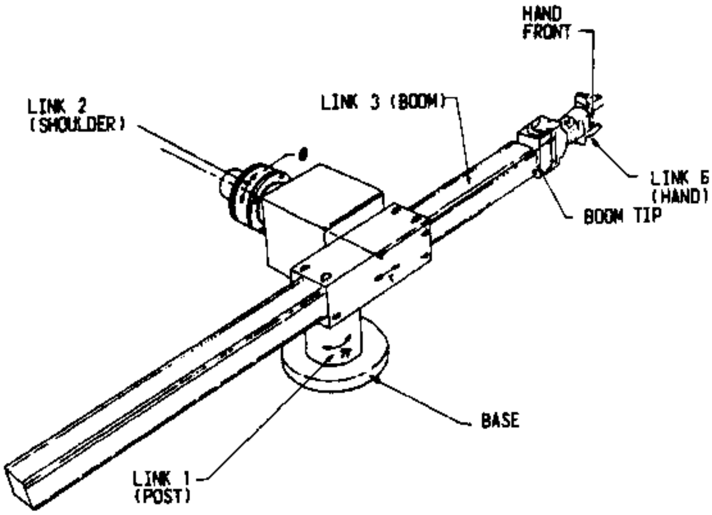

FIGURE 4.1 THE SCHEINMAN ARM

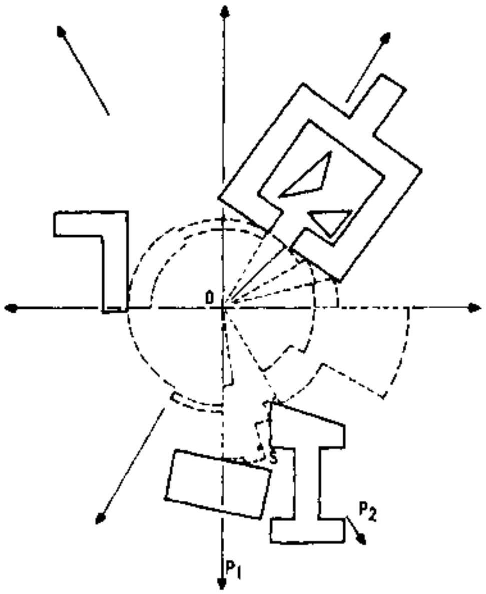

FIGURE 5.2 CHART REFINEMENT - 1

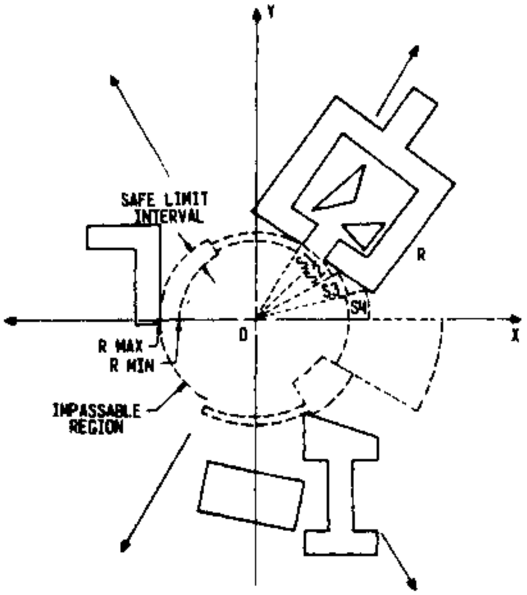

FIGURE 5.1 A CHART AND ITS ATTRIBUTES

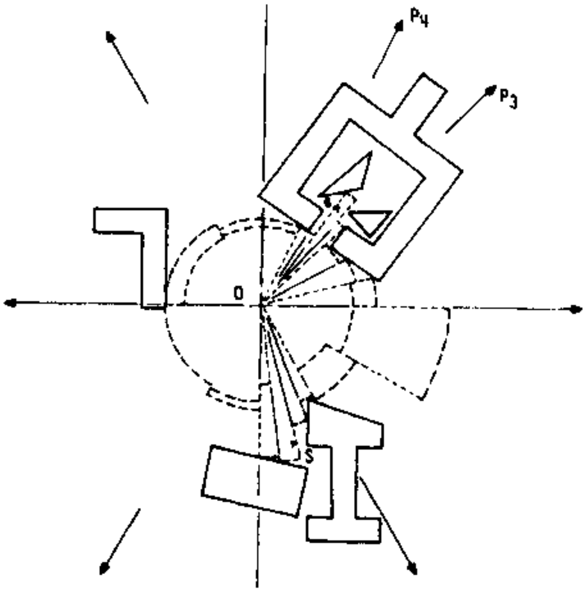

FIGURE 5.3 CHART REFINEMENT - 2

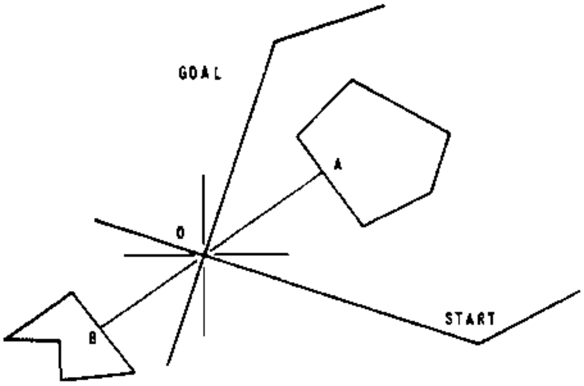

FIGURE 5.4 BLOCKED BOOM

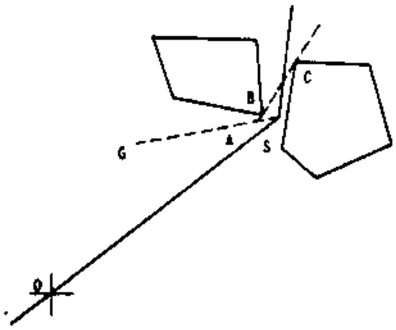

FIGURE 5.5 BLOCKED FOREARM

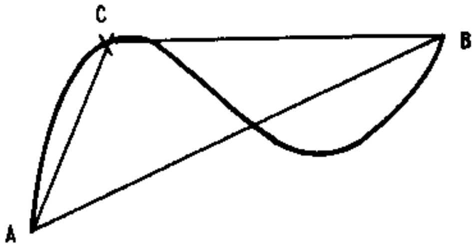

FIGURE 5.6 LINEAR APPROXIMATION

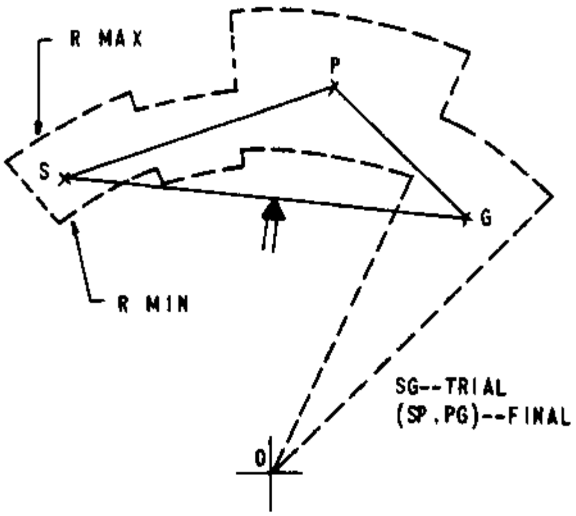

FIGURE 5.7 MODIFIED LINEAR APPROXIMATION

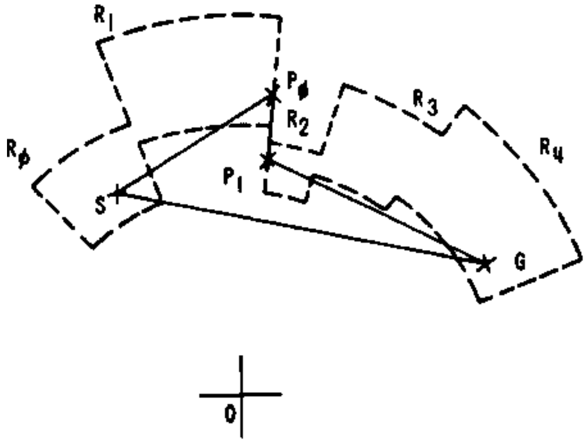

FIGURE 5.8 TRAJECTORY FIXING

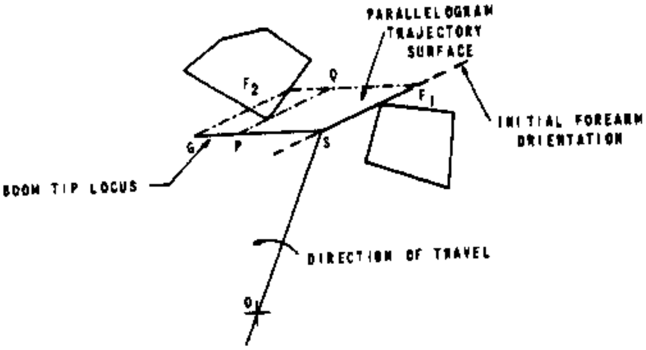

FIGURE 5.10 PGRAM MOTION

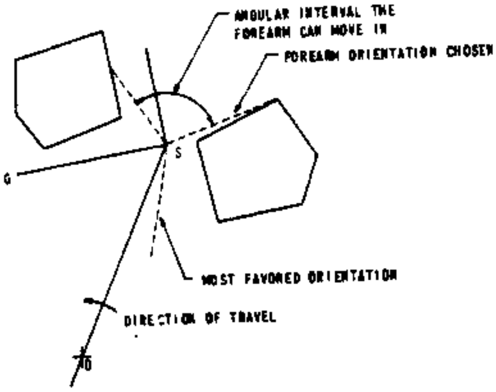

FIGURE 5.9 CIRCLE MOTION

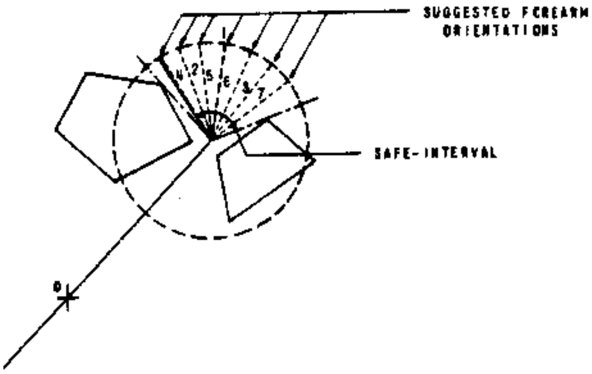

FIGURE 5.11 ADJUST MOTION