Path Planning and the Topology of Configuration Space

Anthony A. Maciejewski, *Member, IEEE*, and John J. Fox, *Student Member, IEEE*

***Abstract*—This work considers the path planning problem for planar revolute manipulators operating in a workspace of polygonal obstacles. This problem is solved by determining the topological characteristics of obstacles in configuration space, thereby determining where feasible paths can be found. A collision-free path is then calculated by using the mathematical description of the boundaries of only those configuration space obstacles with which collisions are possible. The key to this technique is a simple test for determining whether two disjoint obstacles are connected in configuration space. This test allows the path planner to restrict its calculations to regions in which collision-free paths are guaranteed *a* priori, thus avoiding unnecessary computations and resulting in an efficient implementation. Typical timing results for environments consisting of four polyhedral obstacles comprised of a total of 27 vertices are on the order of 22 ms on a SPARC-IPC workstation.**

I. Introduction

THE PROBLEM of path planning for robotic manipulators can be defined as that of finding a continuous motion that will take a manipulator from a given initial configuration to a desired final configuration, subject to the constraint that at no point in the motion does the manipulator collide with any obstacle in its workspace. A brief history of this subject can be found in [26] with [13] providing a recent and extensive survey. A large segment of the research that has been done in this area derives from the seminal work of Lozano–Pérez on configuration space, or c-space, representation [17]–[19]. These approaches can be grossly described as being comprised of two phases during which a data structure is first built that represents the geometric constraints imposed by the obstacles and then, subsequently, searched for a solution. Typically, the process of building the data structure consists of mapping the obstacles from the robot's workspace into the robot's c-space. The result of this operation is an explicit representation of all robot configurations that do not result in a collision with some obstacle, that is, the robot's free space.

In the past, a significant amount of work has employed a strategy that partitions free space into a set of subspaces called "cells" [9], [17], [25]. The adjacency relationship between cells is then captured by a data structure called the "connectivity graph." Establishing the existence of a collision-free path is thereby reduced to searching the connectivity graph for a sequence of adjacent cells, or a "channel," the first of which contains the initial configuration and the last containing the end configuration. The exact cell decomposition method can, in a sense, be thought of as a dual approach to road map type algorithms, such as the Voronoi diagram approach [8], [24], and the relationship between the two has been established in the literature [16].

The approach described by this paper is similar to those described above, however, it differs from traditional algorithms in that there is no explicit generation of free space. Rather, the connectivity of free space is determined by exploiting a fundamental characteristic of c-space obstacles for the case of revolute manipulators. In particular, the connectivity of adjacent subspaces of free space is established by testing for intersections between c-space obstacles. By employing this test judiciously, a channel is obtained in which a collision-free path is guaranteed *a priori*. Having once established where such a channel exists, any of the vast number of local planners, for example [14] or [15], can be used for maneuvering within and amongst the previously determined subspaces.

Previous researchers, notably [2], [5] and [6], have utilized topological information in generating an obstacle representation for planning. In these works, representations of c-space obstacles were developed by convolving a representation of a free flying robot with a representation for an obstacle. Because the mechanism used for performing these convolutions tended to produce a small number of vertices, edges, or faces that were either redundant or nonrealizable, a second stage was employed that utilized topological information to remove these extraneous pieces of information and, thereby, produce an accurate boundary representation of the obstacle in configuration space. Here, the topological properties of c-space obstacles are used as a filter for culling out regions of configuration space in which collision-free paths cannot exist.

Although the work described here does not require that the boundaries of the configuration space obstacles be computed, it is strongly related to work in which the boundaries of configuration space obstacles are analytically described [4], [7], [11], [12], [23]. The novelty of the mathematical representation used here is that the curves describing configuration space obstacles are not only described by their points but also by their tangents. This approach, while not related to the obstacle avoidance algorithm presented in [22], is strongly influenced by the analysis techniques used to describe kinematically redundant manipulators. As will be shown, the tangent information plays a crucial role in determining the topology of configuration space.

Manuscript received September 23, 1991; revised June 30, 1992. Parts of this work were performed at Sandia National Laboratories, Albuquerque, New Mexico. This work was supported in part by the U.S. Department of Energy under Contract DE-AC04-76DP00789 and in part by the National Science Foundation under grant CDR 8803017 to the Engineering Research Center for Intelligent Manufacturing Systems.

The authors are with the School of Electrical Engineering, Purdue University, West Lafayette, IN 47907–1285.

IEEE Log Number 92095585.

The feasibility of the approach presented here is demonstrated by considering the path planning problem for a two-dimensional (2-D) revolute manipulator operating in a workspace of polygonal obstacles. The study of this particular problem was initially motivated by the desire to plan pick and place operations for SCARA type manipulators moving amidst polygonal obstacles, an operation that is essentially planar. Although the majority of the work that is presented focuses upon what appears superficially to be a restricted problem, the techniques that are utilized appear to be generalizable to other problems of interest. To illustrate this, Section VII discusses the extensions required to perform motion planning for manipulators operating in a three-dimensional (3-D) workspace.

The remainder of this work is organized as follows. Section II reviews the kinematic transformation between a robot's workspace and its configuration space, presenting the relevant kinematic equations at the position level. Section III presents a novel approach for using the velocity transformation between the workspace and configuration space to analytically determine the tangents of configuration space obstacles. This representation is then used to develop a condition for determining when disjoint workspace obstacles intersect in configuration space. Section IV describes how this intersection test for obstacles in configuration space can be used to identify global properties of free space that are then used to guide a path planning algorithm. The implementation details and performance measurements are then provided in Section V in which timing results for environments consisting of four polyhedral obstacles comprised of a total of 27 vertices are shown to be on the order of 22 ms on a SPARC-IPC workstation. The presentation in Sections II-V is centered around obstacles modeled as points and 2-D manipulators modeled as line segments in order to reduce unnecessary detail. Section VI illustrates the straightforward extension of the algorithm to include robots and obstacles that are modeled as polygons and includes timing results obtained by running this algorithm on sets of randomly generated obstacles. Section VII outlines the extensions required for higher dimensional manipulators and, finally, section VIII presents the conclusions of this work and compares it to other recent results.

II. Transformation from Workspace to Configuration Space

The transformation of obstacles from a robot's workspace into a robot's configuration space is, naturally, strongly related to the inverse kinematics of the robotic mechanism [21]. Consider the two degree-of-freedom revolute manipulator with parallel joint axes depicted in Fig. 1. For a given set of joint angles, θ_1 and θ_2, the Cartesian position of the end effector is easily calculated using

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} l_1 c_1 + l_2 c_{12} \\ l_1 s_1 + l_2 s_{12} \end{bmatrix} \tag{1}$$

where θ_{12} denotes $\theta_1 + \theta_2$ and c_i and s_i denotes $\cos\theta_i$ and $\sin\theta_i$, respectively. Likewise, for the end effector to be in contact with a point obstacle located at the Cartesian coordinates x and y, one can use the following well-known inverse kinematic equations to solve for the two possible manipulator configurations:

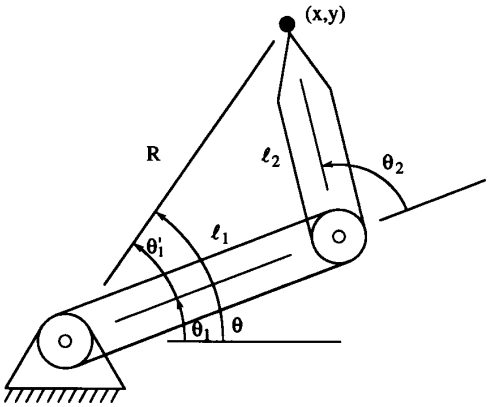


Fig. 1. The kinematics of a two degree-of-freedom revolute manipulator with parallel joint axes.

$$\begin{aligned} \theta_1 &= \theta \pm \theta_1' \\ &= \theta \pm \cos^{-1} \frac{R^2 + l_1^2 - l_2^2}{2 l_1 R} \end{aligned} \tag{2}$$

and

$$\theta_2 = \pm \cos^{-1} \frac{R^2 - l_1^2 - l_2^2}{2 l_1 l_2} \tag{3}$$

where

$$R = \sqrt{x^2 + y^2} \tag{4}$$

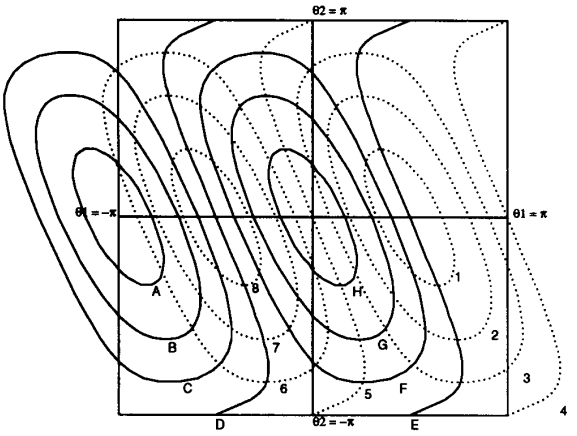
and

$$\theta = \tan^{-1} \frac{y}{x}. \tag{5}$$

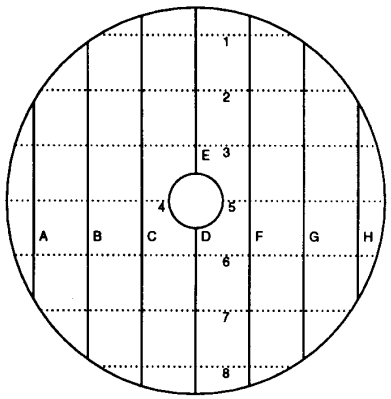
It is also easy to see that there is a very simple relationship between the angles θ_1' and θ_2 given by

$$\frac{\sin\theta_1'}{\sin\theta_2} = \frac{l_2}{R}. \tag{6}$$

The above inverse kinematic relationship can be interpreted as the specification of how the workspace is deformed in order to be transformed into configuration space. A graphical interpretation of this transformation is depicted in Fig. 2 where lines of constant x and y in the workspace are transformed into the configuration space of a manipulator whose link lengths have a ratio of $l_2/l_1 = 3/4$. This is clearly a complex, highly nonlinear mapping, however, it does possess rotational symmetry, with the same pattern in configuration space being shifted in θ_1. This, of course, simply visually confirms what (2)–(5) have revealed, i.e. the transformation to configuration space is simplified by describing the workspace in terms of polar coordinates. This fact is visually illustrated in Fig. 3 where curves of constant R and θ in the workspace are transformed into the configuration space of the same manipulator used to develop Fig. 2. From Fig. 3 it is clear that

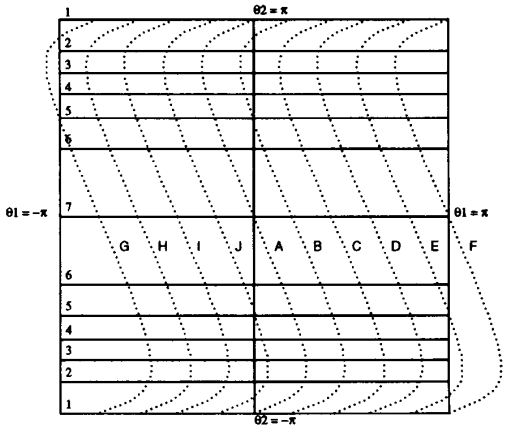


(a)

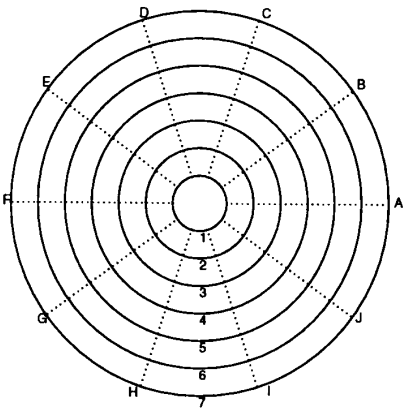


(b)

Fig. 2. The transformation between the position of the end effector in the workspace into confignration space. Lines of constant x and y in the manipulators Cartesian workspace are mapped into the joint values required to achieve these end-effector positions. The manipulator has a geometry as defined in Fig. 1 with $l_2 = \frac{3}{4}l_1$. The labels A through H and 1 through 8 show the correspondence between lines in the workspace and curves in c-space.(a) Configuration space. (b) Work space.



(a)



(b)

Fig. 3. The transformation between the position of the end effector in the workspace into configuration space. Lines of constant R and θ in the manipnlator's polar workspace are mapped into the joint values required to achieve these end eflector positions. The manipulator has a geometry as defined in Fig. 1 with $l_2 = \frac{3}{4}l_1$. The labels A through J and 1 through 7 show the correspondence between curves in the workspace and curves in c-space. (a) Configuration space. (b) Work space.

the distance of an obstacle from the base of the manipulator is the dominant characteristic in determining its shape in configuration space. It is also important to note the monotonic relationship between R and θ_2 as well as R's independence of θ_1. Physically, one can consider θ_2 as responsible for setting the distance of the end effector from the base (R) with θ_1 responsible for determining angular position (θ).

III. Configuration Space Obstacles

The preceding section has reviewed the transformation from end-effector positions in the workspace to joint positions in configuration space. However, when dealing with the transformation of obstacles from the workspace into configuration space, one is concerned with contact along the entire length of the robot and not only at the end effector. Thus, despite the fact that the manipulator depicted in Fig. 1 has only two variable parameters, θ_1 and θ_2, it will be convenient to consider the length of the second link, l_2, to be a variable as well, in effect creating a planar redundant manipulator with two rotary joints and one prismatic joint. The value of l_2 for this redundant manipulator's pseudo-end-effector being located at the position of an obstacle therefore gives the point of contact of the arm with the obstacle. Since the manipulator is now redundant, there is no longer a unique solution to (2) and (3) but a one-dimensional infinity of solutions parameterized by the now variable l_2. Thus a point obstacle in the manipulator's workspace becomes transformed into a curve in configuration space.

The value of R for the position of the pseudo-end-effector of the redundant manipulator is now no longer strictly a function of θ_2 but of l_2 as well. If θ_2 is fixed at a given value then the value of l_2 can be obtained using

$$l_2 = -l_1 c_2 \pm \sqrt{l_1^2 c_2^2 - l_1^2 + R^2} \tag{7}$$

or vice-versa using (3). Thus the shape of a point obstacle in configuration space can be mapped out by stepping along

either θ_2 or l_2 and then plotting the resultant values of θ_1 and θ_2. However, the analysis of the geometry, and ultimately the topology, of configuration space obstacles is simplified by considering the kinematic transformation between workspace and configuration space at the velocity level rather than the position level.

By differentiating (1) with respect to time, one obtains the following relationship between the end-effector velocities and the velocities of the manipulator variables:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} & -l_2 s_{12} & c_{12} \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} & s_{12} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{l}_2 \end{bmatrix} \tag{8}$$

where the 2 by 3 matrix on the right hand side is the Jacobian, which will be denoted by $\boldsymbol{J}$. A description of the null space of the Jacobian transformation is easily obtained by performing the cross product on the rows of $\boldsymbol{J}$ to obtain the vector

$$\boldsymbol{n}_J = \begin{bmatrix} -l_2 \\ l_2 + l_1 c_2 \\ l_1 l_2 s_2 \end{bmatrix}. \tag{9}$$

The physical significance of this vector is that it represents the ratios of how the three manipulator variables must change in order to maintain contact with an obstacle. In addition, the ratio of the first two elements of $\boldsymbol{n}_J$ represents the slope of the curve that describes an obstacle in configuration space.

At this point it is instructive to consider the shape of obstacles in configuration space as a function of their position in the workspace (see also [4], [23]). Fig. 4 shows the shape of nine point obstacles that are equally spaced in terms of R and θ, with R increasing at a rate of 10% of the total reach of the manipulator and θ going from $-\pi$ to π in $\pi/4$ increments. In this figure, for those obstacles that are at $R \leq l_1$, the collisions with the first link are shown as dashed lines in configuration space to distinguish them from collisions with the second link. For all of the obstacles the value of θ_1 approaches θ as l_2 approaches its minimum value. For obstacles that are further from the base than l_1, (points F through I), only the second link can contact the obstacle and the value of θ_2 approaches 0 as l_2 approaches its minimum value. Notice that the inverted S-shape of the obstacles becomes more pronounced as R approaches l_1. The most striking change in the shape of the obstacles as a function of R is, of course, the transition that occurs at $R = l_1$ (point E). At this point, and only this point does θ_2 approach $\pi/2$ as l_2 approaches its minimum value, which is zero. For $R < l_1$, (i.e. points A through D), θ_2 approaches $\pm\pi$ when l_2 approaches its minimum value. Clearly, the vertical lines that occur for all obstacles with $R \leq l_1$ represent the contact of the first link with the obstacle, since this case is independent of θ_2.

The ability to calculate the shape of configuration space obstacles is crucial to determining collision-free paths. However, it is arguably even more desirable to know the topology of these obstacles in configuration space, i.e. their connectivity, and thereby ascertain the connectivity of collision-free paths. The fundamental problem therefore, is determining whether two disjoint obstacles in the workspace intersect in configuration space. Physically this means that there is a configuration

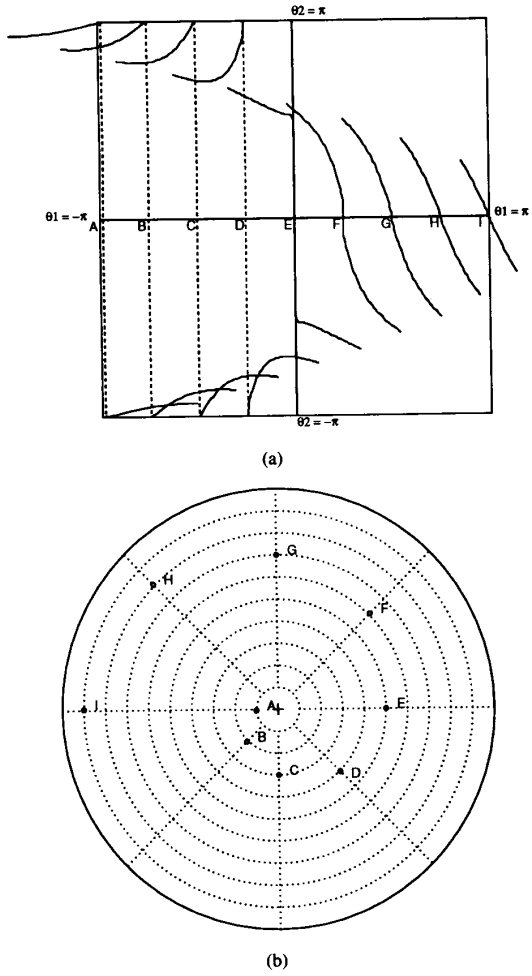


Fig. 4. The transformation of point obstacles from a manipulator's workspace into the corresponding curves in configuration space. The obstacles are located at equally spaced values of R and θ. The manipulator has a geometry as defined in Fig 1 with $l_2 = l_1$. The dashed lines on obstacles A through D represent collisions with the first link. (a) Configuration space. (b) Work space.

in which the manipulator is simultaneously in contact with both obstacles (referred to as a "critical point" in [25]). There are three cases to consider:

1) both obstacles are in contact with the first link,
2) one obstacle is in contact with the first link and the other with the second link, and
3) both obstacles are in contact with the second link.

The first of these cases is trivial, since both obstacles must have the same value of θ and $R \leq l_1$. The second case is only slightly more complicated since one must only check to see whether the two values of θ_1 for the end-effector contacting the one obstacle bracket the value of θ for the other obstacle located at $R \leq l_1$. The third case is the most interesting.

Consider the obstacles labeled F and G in Fig. 4. Clearly, in order for the second link to be simultaneously in contact with both of these obstacles its orientation must be parallel to the line connecting the two obstacles. Since the orientation of the second link is given by θ_{12} this condition leads to the

constraint equation

$$\theta_1 + \theta_2 = \tan^{-1} \frac{y_F - y_G}{x_F - x_G}. \tag{10}$$

Note that it is extremely fortuitous that this is a line in configuration space with a slope of -1. Thus if the configuration space obstacles F and G intersect, they must do so somewhere along these lines. Therefore, to check for a possible intersection, one can simply check to see if both curves representing the obstacles F and G intersect with the lines defined by (10).

Now the intersection of a line and a general nonlinear curve is not trivial, however, in this case there is a simple equation that can be extracted from (9), which gives the tangents to the configuration space obstacle curves. The minimum and maximum distances from a curve to a line will occur at the points along the curve at which the tangent matches the slope of the line. Thus, setting the tangent equation of the configuration space curve equal to the slope of the constraint equation (10), results in

$$\frac{-l_2}{l_2 + l_1 c_2} = -1 \tag{11}$$

the solution of which is given by $\theta_2 = \pm\pi/2$. Therefore, to determine if the lines defined by (10) intersect a configuration space curve one must evaluate the curve at only two points, i.e. those at $\theta_2 = \pm\pi/2$ and check to see if they bracket any of the lines defined by (10). If the value of l_2 is greater than the actual link length at $\theta_2 = \pm\pi/2$ then the end points of the curve should be used in their intersection test.

The above procedure is illustrated by using Fig. 5 and considering the steps required to determine if the configuration space image of obstacles F and G actually intersect. First one would evaluate (10) in order to determine the line on which any possible intersection must lie. Next, one must determine whether both of these obstacles intersect this line since if they don't then it is impossible for them to intersect each other. For obstacle F, one need only determine the value of θ_1 at the points F_1 and F_2, which have values of $\theta_2 = \pi/2$ and $\theta_2 = -\pi/2$ respectively. This is due to the fact that one knows that the tangents associated with these points will have slopes that are identical to that of the line described by (10). Having determined the points F_1 and F_2 it is trivial to determine whether they lie on the same side of the line described by (10). In case (a) of Fig. 5, both F_1 and F_2 lie on the same side of this line so one can immediately guarantee that the configuration space obstacles F and G do not intersect. However, if one moves obstacle G by changing its θ position to be closer to F then one would have the situation shown in case (b). In this case the intersection test between obstacle F and the line of possible intersection is satisfied since moving G has changed the intercept of this line. Now one must also check whether obstacle G intersects this line. Once again, one only needs to determine the points G_1 and G_2 which have values of $\theta_2 = \pi/2$ and $\theta_2 = -\pi/2$, respectively. If the value of R for this obstacle had been increased, then it would not have extended to $\pm\pi/2$ in the θ_2 direction so that the endpoints of the obstacle curve would be used in the intersection test. Once again it is trivial to show that G_1 and G_2 lie on opposite sides of the line described by (10). With this information one can easily show that the two obstacles F and G do in fact intersect.

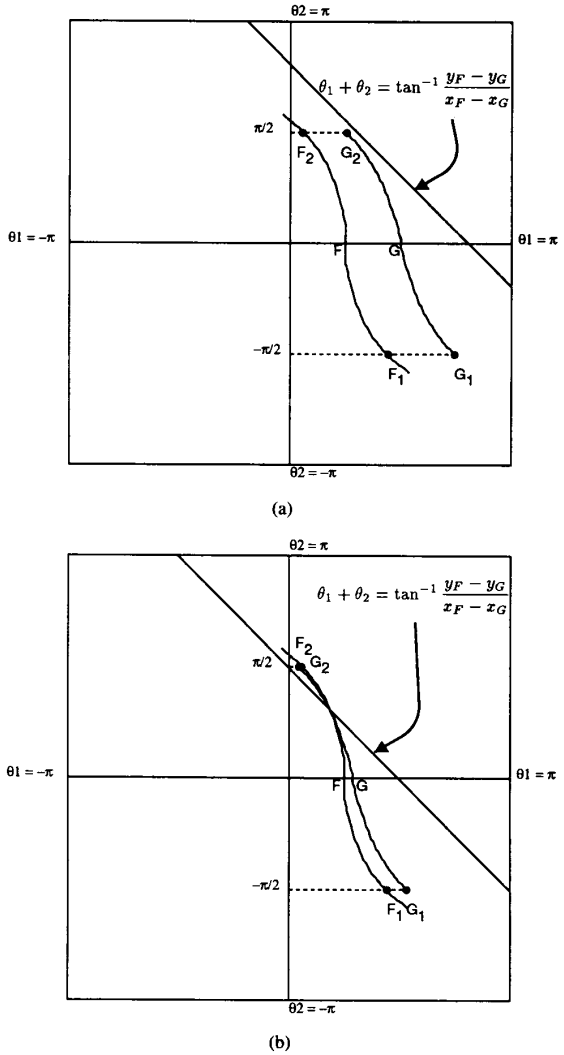


Fig. 5. An illustration of the intersection test used to determine if the configuration space images of two point obstacles are disjoint. In (a) the obstacles F and G correspond to their positions in Fig. 4 so that the resulting line defining a possible intersection does not intersect either of the obstacles. Not.e that one only needs to identify points F_1 and F_2 a.nd show tli"t they both lie on the sa.me side of this line to prove that there is no intersection. In (b) the obstacle G has been shifted in its θ position to lie closer to F. The line of possible intersections now identifies where these two curves intersect. The fact that these obstacles intersect is known by simply showing that the points G_1 and G_2 as well as the points F_1 a.nd F_2 lie on opposite sides of this line.

As with any algorithm in computational geometry, the effects of uncertainty, whether due to the modeling of the environment or to finite precision arithmetic, need to be addressed. Fortunately, the c-space obstacle connectivity test is not inherently ill-conditioned and the algorithm described above naturally provides a measure of proximity to an intersection. To illustrate, note that the principal calculation for ascertaining whether an intersection exists involves determining the location of four points relative to a line whose slope is

known to be -1. This is done by adding the two coordinates of the points, one of which will be typically known to be $\pm\pi/2$, and comparing this magnitude to the magnitude of the lines intercept. The norm of this comparison is proportional to the minimum distance from the point to the line, and thus provides a reliable metric for determining the probability of an intersection in the face of uncertainty.

The above analysis has illustrated how the topology of configuration space obstacles can be efficiently determined without having to evaluate the actual curves. These basic connectivity tests allow one to determine where connected portions of collision-free configuration space exist. This ability, along with the capability to evaluate the actual shape of a configuration space curve when required, form the basis of an efficient path planning algorithm.

IV. Path Planning

Before discussing the path planner used in this work, it is instructive to consider what type of global information one can obtain about free space. The identification of global features that determine regions of topologically equivalent collision-free paths can greatly improve the efficiency of most path planners. To illustrate this concept, consider the configuration space shown in Fig. 6, which contains seven obstacles. An abstract representation of its free space is shown in the lower half of the figure. One of the most basic global properties to identify is that free-space will be partitioned into disjoint maps, referred to here as "continents," by any obstacles that are closer than the length l_1 to the base. Thus, for the example in Fig. 6, obstacle F partitions free space into two continents and therefore path planners need only consider those obstacles that lie in the same continent as the start and goal configuration of the manipulator.

An additional global feature of free space for 2-D revolute manipulators that has been previously identified, is the existence of "highways" [23]. Physically, highways correspond to distinguished subspaces of configuration space for which a collision-free path can be planned simply by using a line segment parallel to the θ_1 axis for some relatively large range of θ_2 values. Fig. 6 illustrates the portions of configuration space corresponding to the highways for the continent depicted to the left. Note that the concept of a highway as illustrated here is more general than that presented in [23], in which obstacles at $R < l_1$ were not considered. In this work, the highway that exists for large values of θ_2, will be referred to as the "north highway" with the other highway referred to as the "south highway." Other global features of free space that are not guaranteed to exist include a path from one highway to the other. This feature, which if it does not exist implies that the free space is further partitioned, will be referred to as an "isthmus". Regions of free space that are connected to only one of the two highways will be referred to as "peninsulas" and finally those portions of free space that are entirely surrounded by obstacles and not connected to either highway are called "islands."

If there are no limitations on joint rotations then the 2-D configuration space of a revolute manipulator is mathematically described by a torus. However, physical manipulators are rarely without joint limits, hence the configuration space of such manipulators is homeomorphic to a closed subset of a plane in R^2. Throughout this work, the rather arbitrary joint limits of π and $-\pi$ are imposed, but the work is in no way restricted by this assumption. Notice, however, that if one removes the joint limits then the calculation of highways must be slightly modified to account for the fact that the top and bottom edges of c-space are connected as well as the left and right edges. Planning paths on the resulting torus is, in some ways, easier than planning under the assumption of joint limits since the calculation of isthmuses can now be avoided.

If, on the other hand, the ranges of θ_1 and θ_2 are such that they are more restrictive or are translated within (θ_1, θ_2) space, then many of the definitions that have been established above must be modified. In particular, continents need not be delimited by obstacles with $R < l_1$ since it is possible that suitably placed obstacles might form an impassable barrier in configuration space under particular joint limits. Also, with

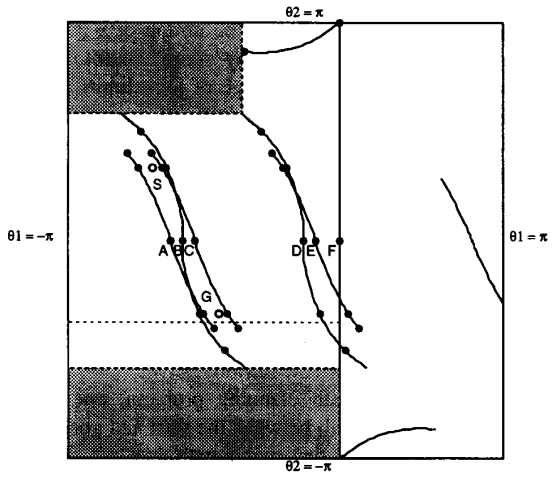


Fig. 6. An example of seven obstacles in configuration space along with an abstract representation of the resulting free space. For the given start (S) and goal (G) configuration one can identify the location of S and G in the a.bstra.ct. free space as well as a set of topologically equivalent collision-free paths by evaluating only those points shown in configuration space. These points correspond to those required to order the obstacles and those required to perform the intersection tests. The grey areas represent the north and south highways for the left continent. The dashed line below the goal point is an example of an alternativeθ_2 joint limit that would modify the abstract free space. (a) Configuration space. (b) Abstract-free space.

appropriately chosen joint limits, the notion of a highway as defined above need not exist. Finally, some features that would have existed under a different range of joint limits might not exist or may need to be reclassified. As an example, consider the situation if Fig. 6 is modified so that the range of θ_2 extends only to the dashed line in the figure. In this case, there is no south highway as defined above. Furthermore, the two peninsulas connected to the south highway are now islands and the two isthmuses are now peninsulas connected to the north highway. Clearly, while modifications in permissible joint limits would require minor modifications in the algorithms to establish the various features, the taxonomy for the subspaces of configuration space is still reasonable.

When solving the path planning problem, one typically requires only a single path between the start and goal configuration or the information that such a path does not exist. Thus one does not need to consider the global features of the entire free space but only those associated with the start and goal configuration. The following example will serve to illustrate how one can determine if and where a set of topologically equivalent collision-free paths exists.

Consider Fig. 6 once again with the given start and goal configurations identified as S and G, respectively. To determine on what type of feature the start configuration is located, one first determines the points on the obstacles A through F that have the same value of θ_2 as S. Ordering these points by their θ_1 values identifies that obstacle A lies to the left of S and that obstacles B through F lie to the right. Now in order to determine whether S lies on an island, peninsula, or isthmus one needs to apply the intersection test between obstacle A and each of the obstacles B through F. Applying these tests will identify that obstacles A and B intersect at a value of θ_2 smaller than that of S so that one can immediately conclude that S lies on a peninsula that is connected to the north highway. Applying the same procedure to the goal configuration results in intersection tests between the obstacles A and B against D through F, which determines the intersection between B and C and identifies G as lying on a peninsula attached to the south highway.

Since S and G are not connected to the same highway, one still needs to determine if an isthmus exists, and, if so, where it is located. This is done by ordering the obstacles A through F by their values of θ and determining the most likely location of an isthmus by comparing the differences in θ for adjacent obstacles. Note that when $\theta_2 = 0, \theta_1 = \theta$. This heuristic identifies the gap between obstacles C and D to be the most likely location for an isthmus, so intersection tests are applied between the set of obstacles A, B, and C and the set of obstacles D, E, and F. This results in the identification of an isthmus and, hence, the location of a set of topologically equivalent collision-free paths. This global information was obtained by only evaluating a few points of the configuration space obstacles, identified in Fig. 6 with dark circles, as well as the repeated evaluation of a simple intersection test. Clearly, if S and G were found to lie on different islands or if the search for an isthmus had failed one could guarantee that no path existed between S and G since they were located on disjoint portions of free space.

TABLE I
TIMING DATA FOR THE PATH PLANNING ALGORITHM AS A FUNCTION OF THE NUMBER OF OBSTACLES

No. Obstacles	Intersection Tests			Total Time
n	No.	Time	$\%T$	T
10	59	4.7 ms	33.8%	13.9 ms
20	204	16.3 ms	48.2%	33.8 ms
30	398	31.9 ms	61.0%	52.3 ms
40	686	54.9 ms	66.3%	82.8 ms
50	1079	86.5 ms	73.1%	118.3 ms

Note: Times are in milliseconds on a SPARC-IPC

Assuming that the start and goal are connected, one now has information about which obstacles form the boundary around a set of topologically equivalent collision-free paths in free space so that one can easily calculate a specific path. The salient point here is not which particular path is generated but that the performance of any path planner would benefit from the global properties obtained from the above analysis. In this implementation, both the position information, from (2), (3), and tangent information, from (9), of the obstacles bracketing the path are used in determining the shape of the path. A weighted average of the position and tangent information is used in order to try and maintain the path near the center of the bounding obstacles.

As illustrated, in the particular case of planning motions for a planar manipulator it is not necessary to generate the entire abstract free space. Specifically, the portion of asbtract free space that is generated is the minimum required to find a path. In the example described above, the only portions of the abstract free space that needed to be generated were the two peninsulas on which the start and goal resided, the two highways, and enough of the abstract free space to determine where an isthmus existed. In many cases, the number of intersection tests between obstacles that are actually performed is significantly lower than the number of possible intersections between obstacles, i.e. $(n(n-1))/2$, where n is the number of point obstacles. Besides reducing the number of intersections that must be calculated, this has the useful side effect of tending to reduce the length of the path that is generated. In particular, one does not run the risk of generating a path that visits spurious isthmuses or peninsulas because these features have not been added to the free space representation. Since many of these paths are homotopic to the paths that are being calculated, the channel that is being generated in the first stage of planning is actually representative of a subset of the topologically equivalent paths.

V. SIMULATION RESULTS

The above path planning algorithm has been implemented in the C language and extensively tested on a SPARC-IPC workstation containing a 15.7 MIPS RISC architecture that results in a comparatively modest 1.7 Mflops performance. Timing data for various different environments are presented in Table I, with two representative examples given in Figs. 7 and 8. Fig. 7 shows a workspace with ten point obstacles randomly scattered throughout the workspace and with the start and goal configurations widely separated. Note that the algorithm

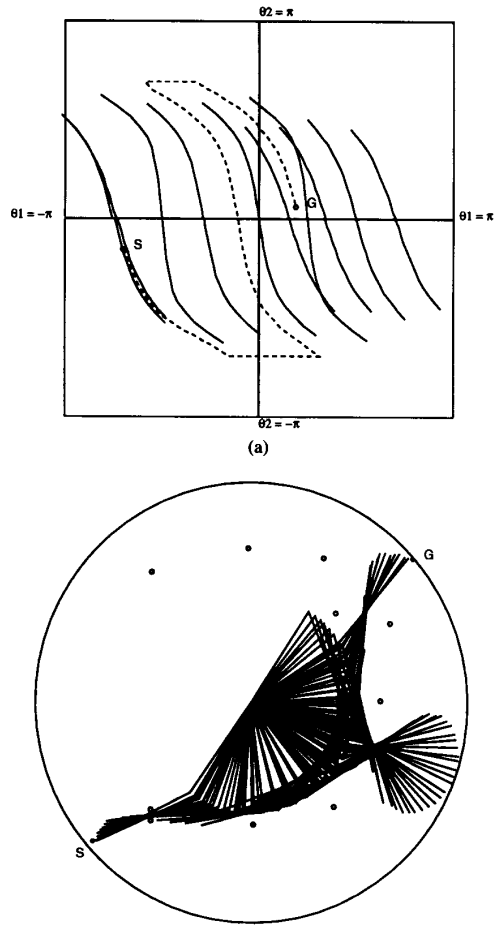


Fig. 7. A collision-free path, shown in dotted lines, from the start (S) configuration to the goal (G) configuration found by the path planning algorithm. The execution time for this environment, which has 10 point obstacles, was 19.8 ms on a SPARC-IPC. (a) Configuration space. (b) Work space.

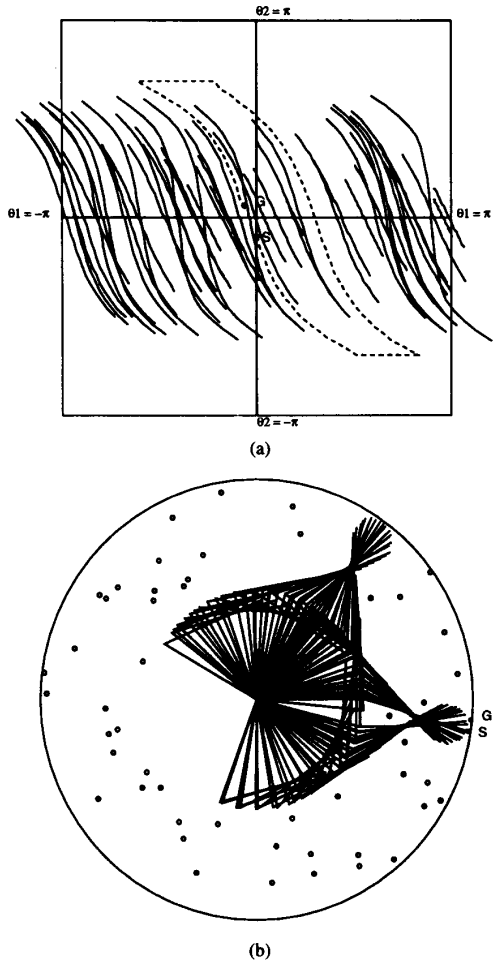


Fig. 8. A collision-free path, shown in dotted lines, from the start (S) configuration to the goal (G) configuration found by the path planning algorithm. The execution time for this environment, which has 50 point obstacles was 132 ms on a SPARC-IPC. (a) Configuration space. (b) Work space.

has identified that both the start and the goal configurations are bracketed by obstacles that intersect, thereby leaving no choice in which highway must be used. The collision-free path between the bracketing obstacles, despite the especially tight constraint near the start, is easily calculated because of the exact expression for both the positions and tangents of the obstacle curves. Note that the intersection test around the goal identifies a third obstacle that must be considered when calculating the path. Since the start and goal portions of the path exit into different highways the algorithm must find a connecting path between them in order to complete the path. In this particular example, the isthmus that is chosen is the one with the largest change in θ between adjacent obstacles, and is, therefore, the first potential isthmus selected by the heuristic described above. The execution time to find the total path was 19.8 ms. Fig. 8 illustrates the algorithm working in a much more cluttered workspace that consists of fifty random obstacles. In this case the start and goal configurations are very close together, however, the most direct path is blocked by an obstacle so that a relatively circuitous path is taken. The execution time to find the total path in this case was 132 ms. Note that the connecting path between the two highways is not the shortest path, due to the ordering based on obstacle separation. It would also be possible to identify all paths that connect the two highways, which could be performed off-line for static environments, and then select the best based on some other criteria such as total path length.

The data presented in Table I, which summarizes the performance of this algorithm, was calculated based on several hundred trials using the Unix program execution profiling utility, "gprof." In each case the positions of the specified number of obstacles was randomly assigned along with random start and goal configurations. The obstacles were uniformly distributed within the reachable cartesian workspace, while

the start and goal configurations were randomly assigned in configuration space. The number of obstacles, denoted by n, only considers those obstacles that lie in the same continent as the start and goal configuration. Those trials that did not have solutions or whose solutions yielded direct paths were not included in the timing statistics since they resulted in substantially shorter execution times. Each entry in the table is averaged over twenty of those trials that required a connecting path between the highways so that they are an average of the worst-case scenario. The trials were conducted with environments that ranged from a minimum of ten obstacles to a maximum of fifty. The maximum number of intersection tests required to determine if there exists an isthmus between two obstacles or to determine if a particular configuration is on an island or peninsula is $n^2/4$, which occurs when there are $n/2$ obstacles on either side of the point in question since each obstacle with a smaller value of θ_1 must be checked against each obstacle with a larger value of θ_1. The actual number of intersection calculations performed is given along with the CPU time required to perform the tests and the percentage of the total time that this represents. Note that the intersection tests that are responsible for determining the topology of configuration space represent an increasingly larger fraction of the total execution time as the environment becomes more complex. However, the implementation of this algorithm has not been optimized and so it is likely that this percentage can be reduced by applying a simpler "necessary condition" intersection test for those obstacles that are far from the path in question.

VI. Modifications for Polygonal Links Among Polygonal Obstacles

The extension of the above analysis to include robots whose joints are modeled by polygons working in an environment filled with polygonal obstacles is straightforward. At a conceptual level, the analysis described above determines all possible contacts between a point (the vertex of an obstacle) and a line segment (the robot link) that happens to be constrained in its allowable configurations. All that needs to be done to extend this algorithm is to include a scheme for determining all the possible contacts between a line segment (the edge of a polygonal obstacle) and a point (a vertex on the robot link) that is constrained in its allowable positions. This description is simply an inverse kinematics problem. Thus for any object that can be described by line segments, i.e. a polygon, the configuration space mapping of that object consists of the vertices mapped as described in section III along with the configurations required for the end effector to traverse the line segments. An example of such a mapping is presented in Fig. 9 where a heptagon in the workspace is mapped to configuration space. Note the seven inverted S-shapes resulting from the seven vertices and the image of the warped heptagon that connects these vertices. Now, by definition, the end effector traces out the heptagon in the workspace when the warped heptagon is traversed in configuration space. A useful workspace design tool results by considering the path traced by the end effector when the remainder of the configuration space obstacle is traversed. The resulting curves in the workspace identify regions that become unreachable due to the placement of the obstacle as well as identifying regions that are reachable in restricted configurations. In effect this mapping back into the workspace identifies how the usable workspace will be affected by the placement of objects in different positions and orientations. Thus it provides an efficient method for calculating the obstacle "shadows" [21].

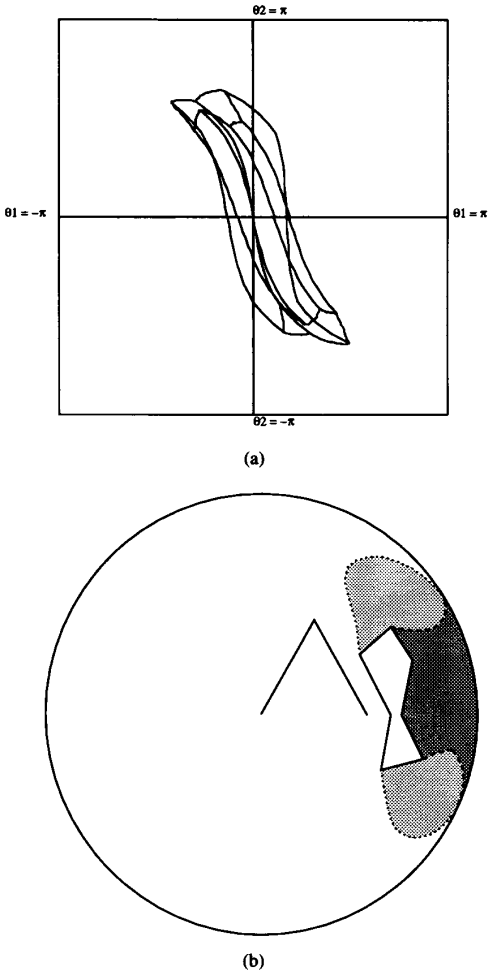


Fig. 9. The transformation of a polygonal obstacle from the workspace into the configuration space. Knowledge of the shape of the obstacle in configuration space then used to compute the "shadows" of the obstacle in the workspace. The dark gray regions are now unaccessible by the manipulator and the light, gray regions are attainable in only a single configuration. (a) Configuration space. (b) Work space.

The above discussion can be extended to include contact of any vertex of the robot link against a polygonal obstacle by simply considering the vertex to be a pseudo-end-effector. The modifications to the kinematics of the robot due to the line segments describing the link no longer passing through the joint axis are presented in Fig. 10. As an example, the relationship between the joint angles and a point on the line segments describing the top and bottom of the arm in Fig. 10

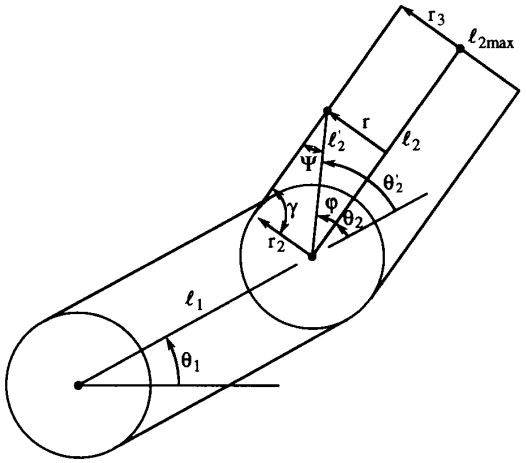


Fig. 10. Modifications to the kinematics of a two degree-of-freedom revolute manipulator with parallel joint axes to account for polygonal-shaped links.

is given by

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} l_1 c_1 + l_2 c_{12} \pm r s_{12} \\ l_1 s_1 + l_2 s_{12} \mp r c_{12} \end{bmatrix} \tag{12}$$

where

$$r = \left(1 - \frac{l_2}{l_{2\,\max}}\right) r_2 + \frac{l_2}{l_{2\,\max}} r_3. \tag{13}$$

Differentiating (12) results in the following modifications to (8)

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = [J] \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{l}_2 \end{bmatrix} + \begin{bmatrix} \pm r c_{12} & \pm r c_{12} & \pm \frac{r_3 - r_2}{c_{12\,\max}} s_{12} \\ \pm r s_{12} & \pm r s_{12} & \mp \frac{r_3 - r_2}{c_{12\,\max}} c_1 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{l}_2 \end{bmatrix}. \tag{14}$$

The most straightforward way of modifying the algorithm described above is to simply apply the transformations

$$\theta_2' = \theta_2 + \phi \tag{15}$$

$$l_2' = \sqrt{l_2^2 + r^2} \tag{16}$$

and then to use exactly the same software with θ_2 and l_2 replaced with θ_2' and l_2'. The value of ϕ is given by

$$\phi = \gamma + \psi - \frac{\pi}{2}. \tag{17}$$

using

$$\frac{r_2}{\sin\psi} = \frac{l_2'}{\sin\gamma} \tag{18}$$

where

$$\gamma = \tan^{-1} \frac{l_{2\,\max}}{r_2 - r_3}. \tag{19}$$

Note that the path planner must also be modified to include the link vertex against obstacle edge contacts into the intersection routine. Fig. 11 illustrates the transformation of polygonal obstacles from the workspace into configuration space for a manipulator whose links are described by a polygon. In theory, the above technique may be applied to much more arbitrarily shaped links by simply modifying (13) so that r is some arbitrary nonlinear function of l_2. Unfortunately, in practice, any nonlinearity in (13) complicates the resulting calculations to a point that approximating arbitrarily shaped links by polygons is typically more efficient. Fig. 12 shows a typical simulation result using this algorithm where the environment includes four obstacles consisting of a total of fifteen vertices. The total execution time for calculating the collision-free path was 27 ms on a SPARC-IPC. Table II summarizes the performance of the polygonal planner working within randomly generated workspaces. Each polygonal obstacle consists of a randomly determined number of vertices chosen from a uniform distribution that ranged between 3 and 10. As before, the vertices were uniformly distributed within the reachable cartesian workspace, and the start and goal configurations were randomly assigned in configuration space. Trials that did not have a solution or whose solutions yielded direct paths were once again discarded because of their shorter execution times. The data presented in the table resulted from averaging ten trials that required the generation of a connecting path between the highways. Thus, once again, this represents a worst-case scenario. The table summarizes the description of the workspace by presenting the number of obstacles and the average number of total vertices in the workspace. The table includes averages for the number of intersection tests performed, the amount of time consumed by these tests, the total amount of time required to find a path, and the percentage of this total time consumed by performing intersection tests.

TABLE II
TIMING DATA FOR THE POLYGONAL PATH PLANNING ALGORITHM AS A FUNCTION OF THE NUMBER OF OBSTACLES

Obstacles		Intersection Tests			Total Time
No. Obst.	Ave. No. Vert.	No.	Time	%T	T
2	12.6	30	2.4 ms	16.2%	14.8 ms
3	17.6	55	4.4 ms	23.8%	18.5 ms
4	27.3	73	5.8 ms	26.2 %	22.1 ms

Note: Times are in milliseconds on a SPARC-IPC

A comparison of Tables I and II reveals the existence of an apparent anomaly. The data in these tables indicates that the seemingly more difficult problem of planning amongst polygonal obstacles is being performed more quickly than solving comparable point obstacle problems. This issue can be resolved by considering the fact that one already has a significant amount of connectivity information about the vertices of a polygon as opposed to disjoint points. As an example, consider the situation depicted in Fig. 11. If the planner were to attempt to decide whether an isthmus existed between obstacles B and C, then it need not exhaustively test for intersections between all of the point obstacles representing the vertices. Therefore, although there may be a large number of vertices in the workspace, a comparatively small number play a part in establishing the presence of local topological features. As a result, the planner was able to generate paths for polygonal environments that consisted of, on the average, slightly more than 27 vertices in significantly less time than that required

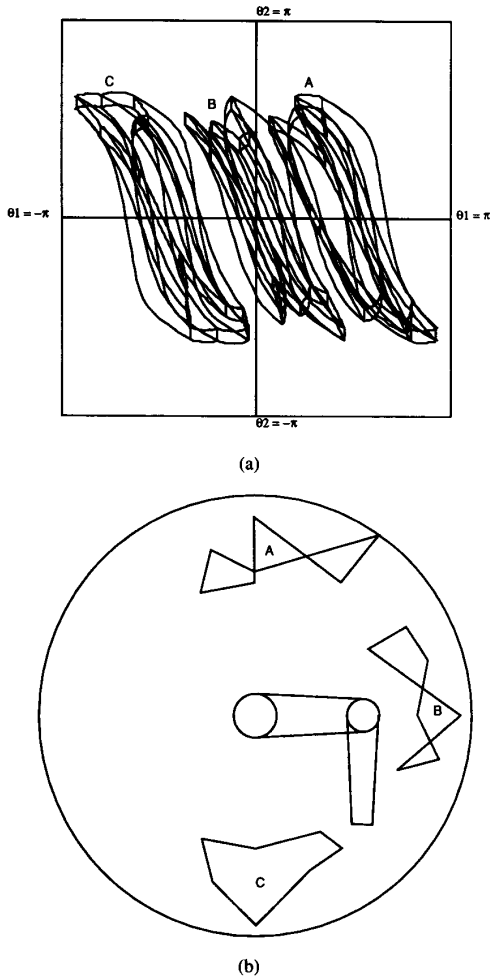


Fig. 11. The transformation of polygonal obstacles from the workspace into the configuration space for a manipulator whose links are also described as polygons. Note that there is no constraint that requires the obstacles to be convex. (a) Configuration space. (b) Work space.

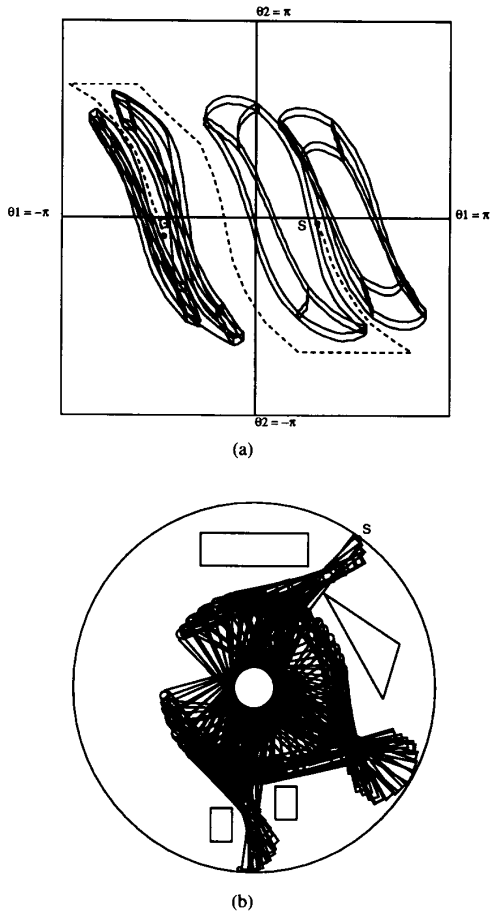


Fig. 12. A collision-free path for a manipulator whose links are described as polygons, shown in dotted lines, from the start (S) configuration to the goal (G) configuration found by the path planning algorithm. The execution time for this environment, which has four polygonal obstacles consisting of 15 vertices, was 27.2 ms on a SPARC-IPC. (a) Configuration space. (b) Work space.

to plan a path amongst 20 point obstacles. Clearly, additional information in the form of necessary conditions can be used to optimize performance, however no attempt has been made to exploit these in the planner described here.

VII. Generalizations to 3-D Workspaces

The motivation behind studying the path planning problem for planar two-link manipulators was discussed in Section I. This section outlines the generalizations required to consider a general positioning manipulator operating in a 3-D workspace. Nearly all such manipulators can be decomposed into a planar component that contains two parallel joint axes and an orthogonal joint axis that is solely responsible for motion in the third dimension. Motion planning for the first three links of a PUMA manipulator provides a familiar example of this typical geometry.

A major difference between planning for planar manipulators as opposed to planning in higher dimensions relates to the amount of the abstract free space that must be calculated. As discussed in Section IV, assumptions about the form of collision-free paths in a plane allow the planner to restrict the amount of the abstract free space that is actually calculated. In higher dimensions such assumptions are not valid and it appears to be necessary to generate the entire abstract free space and then to apply a search algorithm, such as A^*. While this approach is arguably more time consuming, there are advantages, even in the planar case, since optimal paths can be obtained and replanning is more efficient.

An algorithm has been developed for generating the complete abstract free space for a planar environment via a sweep line approach. The nodes of this graph are defined by the path-connected subspaces of free space that are bounded horizontally by the same obstacles. Computing these regions requires the evaluation of the obstacles only at their upper and lower extrema and at their points of intersection with other obstacles. The result of running this algorithm on a simple environment is illustrated in Fig. 13(a). The horizontal dashed lines on the graph delineate the extent of the

regions corresponding to the relevant nodes. Generation of these graphs can be performed in approximately 26 ms on a SPARC-IPC for environments consisting of 20 point obstacles randomly distributed in the same manner as described above. This algorithm is described in detail in [10].

To build a 3-D abstract free space, an approach that is analogous to [17] can be used where, in this case, ranges of the orthogonal joint define the slices. The algorithm begins by determining the abstract free space for a particular slice of the workspace and proceeds by incrementally constructing the graph as the orthogonal joint is swept through a sequence of critical slices. This process is illustrated in Fig. 13 where (a) and (b) represent the abstract free space for adjacent slices. Note, however, that the complete abstract free space for the slice depicted in Fig. 13(b) need not be calculated. Rather, only those edges depicted using dotted lines in Fig. 13(c) need to be added to the graph from Fig. 13(a) to represent the additional free space obtained by considering the second slice. These critical slices are determined by those changes in the topology of configuration space that cause changes in the abstract free space representation between neighboring slices and can be characterized by the following conditions:

1) the introduction of a new obstacle in c-space;
2) the elimination of an existing obstacle in c-space;
3) the introduction of an intersection of two previously disjoint obstacles in c-space; and
4) the elimination of an intersection of two obstacles in c-space.

The first two conditions can be easily identified, particularly for polygonal obstacles, by determining the range of values of the orthogonal joint for which contact with the workspace obstacle is possible. The third and fourth conditions are more difficult to identify and the search for algorithms to efficiently do so are the subject of ongoing research.

VIII. Conclusion

The main contribution of this work is a simple method for determining the connectivity of configuration-space obstacles for revolute manipulators. This is achieved by a novel approach for analytically describing the boundaries of the configuration-space obstacles. It is shown that the incorporation of such topological information into a path planner results in an extremely efficient implementation. The source of this efficiency is rooted in being able to determine global properties of the free space without exhaustively describing the obstacles.

It should be emphasized that this efficiency is inherently coupled to exploiting the constraints on the motion of articulated manipulators. Thus the techniques presented here are not applicable to the problem of moving unconstrained polygons (mobile robots) in a world of polygonal obstacles so that other methods must be employed for such problems [1], [20]. It should also be pointed out that since the algorithm presented is complete it possesses the inherent drawbacks of any global planner, i.e. the curse of dimensionality. Thus it is highly unlikely that the algorithm presented here can be extended to high-dimensional redundant manipulators without

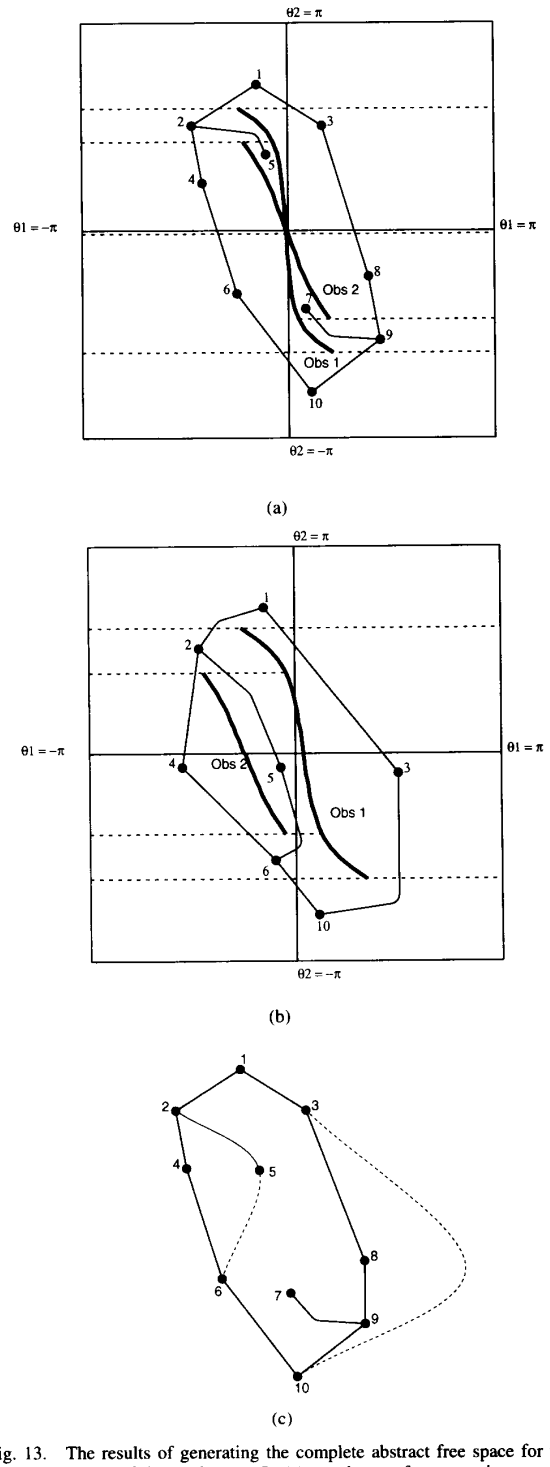


Fig. 13. The results of generating the complete abstract free space for two successive slices of the workspace. In (a), an abstract free space is generated for two obstacles that intersect In (b), the abstract free space is determined for the neighboring slice, in which the twoc-space obstacles no longer intersect. The graph of (c) illustrates the underlying abstract free space for the two slices when they are considered together. The edges depicted using dotted lines are those that are required to represent the incremental changes in the graph due to the change in workspace slices. (a) and (b) Configuration space. (c) Abstract-free space.

sacrificing completeness if it is to compare favorably with existing planners [3].

In conclusion, this paper has presented a novel approach to generating a decomposition of free space that is based upon examining the connectivity of configuration space obstacles. The methodology developed employs analytic techniques commonly associated with the analysis of redundant manipulators to develop a test for establishing the intersection of obstacles in configuration space. The test is extremely efficient and does not require the development of an exhaustive description of the robot's free space. Furthermore, a measure of the reliability of the results is easily calculated, thereby providing a mechanism for establishing the effects of uncertainty on the intersection test. The path planner that results has been demonstrated for planar manipulators modeled as line segments or polygons operating amidst obstacles that are points or polygons, respectively. Finally, extensions to manipulators operating in higher dimensions have been indicated.

ACKNOWLEDGMENT

The first author would like to acknowledge the assistance of numerous individuals at Sandia National Laboratory. He would particularly like to thank David R. Strip, Yong K. Hwang, and Pang-Chieh Chen for the many useful discussions on path planning for articulated manipulators. Also, the authors would like to thank the anonymous reviewers whose suggestions have greatly improved this paper.

REFERENCES

[1] F. Avnaim, J. D. Boissonnat, and B. Faverjon, "A practical exact motion planning algorithm for polygonal objects amidst polygonal obstacles," in *Proc. IEEE Int. Conf. Robotics Automat.*, Philadelphia, PA, May. 1988, pp. 1656–1661.

[2] C. Bajaj and M. S. Kim, "Generation of configuration space obstacles: The case of a moving sphere," *IEEE J. Robotics Automat.*, vol. RA-4, pp. 94–99, Feb. 1988.

[3] J. Barraquand, B. Langlois, and J. Latombe, "Robot motion planning with many degrees of freedom and dynamic constraints," in *Robotics Research: The Fifth Int. Symp.*, Cambridge, MA, 1990, pp. 435–444.

[4] M. S. Branicky and W. S. Newman, "Rapid computation of configuration space obstacles," in *Proc. IEEE Int. Conf. Robotics Automat.*, Cincinnati, OH, May 13–18, 1990, pp. 304–310.

[5] R. C. Brost, "Computing metric and topological properties of configuration-space obstacles," in *Proc. IEEE Int. Conf. Robotics Automat.*, Scottsdale, AZ, May 14–19, 1989, pp. 170–176.

[6] R. C. Brost, "Analysis and planning of planar manipulation tasks," Ph.D. dissertation, Carnegie Mellon University, January 1991.

[7] J. Canny, "Collision detection for moving polyhedra," *IEEE Trans. Pattern Anall. Machine Intell.*, vol. PAMI-8, pp. 200–209, 1986.

[8] J. Canny, *The Complexity of Robot Motion Planning*, M. I. T. Press, Cambridge, MA, 1987.

[9] B. R. Donald, "Motion planning with six degrees of freedom," M. I. T. AI Tech. Rep. 791, May 1984.

[10] J. J. Fox and A. A. Maciejewski, "Computing the topology of configuration space," in *Proc. 1992 IEEE Int. Conf. Syst., Man, Cybern.*, Chicago, IL, Oct. 18–21, 1992, pp. 31–36.

[11] Q. Ge and J. M. McCarthy, "An algebraic formulation of configuration-space obstacles for spatial robots," in *Proc. IEEE Int. Conf. Robotics Automat.* , Cincinnati, Ohio, May 13–18, 1990, pp. 1542–1547.

[12] Y. K. Hwang, "Boundary equations of configuration obstacles for manipulators," in *Proc. IEEE Int. Conf. Robotics Automat.*, Cincinnati, OH, May 13–18, 1990, pp. 298–303.

[13] Y. Hwang and N. Ahuja, "Gross motion planning—A survey," *ACM Computing Surveys*, vol. 24, no. 3, pp. 219–292, Sept. 1992.

[14] Y. Hwang and N. Ahuja, "Path planning using a potential field representation," in *Proc. 1988 IEEE Int. Conf. Robotics Automat.*, Philadelpia, PA, Apr. 24–29, 1988, pp. 648–649.

[15] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. J. Robotics Res.*, vol. 5, no. 1, pp. 90–98.

[16] J. Latombe, *Robot Motion Planning*. Boston, MA: Kluwer Academic, 1991.

[17] T. Lozano-Pérez, "A simple motion planning algorithm for general robot manipulators," *IEEE J. Robotics and Automation*, vol. RA-3, no. 3, pp. 224–238, June 1987.

[18] ——, "Automatic planning of manipulator transfer movements," *IEEE Trans. Syst., Man, and Cyber.*, vol. SMC-11, no. 10, pp. 681–698, Oct. 1981.

[19] ——, "Spatial planning: A configuration space approach," *IEEE Trans. Comput.*, vol. C–32, pp. 26–38, Feb. 1983.

[20] J. Lengyel, M. Reichert, B. R. Donald, and D. P. Greenberg, "Real-time robot motion planning using rasterizing computer graphics hardware," *Computer Graphics*, vol. 24, no. 4, pp. 327–335, Aug. 1990.

[21] V. Lumelsky, "Effect of kinematics on motion planning for planar robot arms moving amidst unknown obstacles," *IEEE J. Robotics and Automation*, vol. RA-3, no. 3, pp. 207–223, June 1987.

[22] A. A. Maciejewski and C. A. Klein, "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments," *Int. J. Robotics Res.*, vol. 4, no. 3, pp. 109–117, Fall 1985.

[23] W. Meyer and P. Benedict, "Path planning and the geometry of joint space obstacles," in *Proc. 1988 IEEE Int. Conf. Robotics Automat.*, Philadelphia, PA, Apr. 24–29, 1988, pp. 215–219.

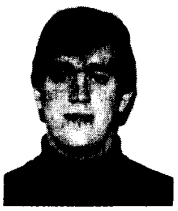
[24] C. Ó'Dúnlaing and C. Yap, "A retraction method for planning the motion of a disc," *J. Algorithms*, Mar. 1985.

[25] J. T. Schwartz and M. Sharir, "On the piano movers' problem—Part I: The case of a two-dimensional rigid polygonal body moving amidst polygonal barriers," *Commun. Pure and Applied Math.*, vol. 34, pp. 345–398, 1983.

[26] J. T. Schwartz and C. K. Yap, *Advances in Robotics, Vol. I, Algorithmic and Geometric Aspects of Robotics* . Hillsdale, NJ: Erlbaum, 1987.

Anthony A. Maciejewski (S'82–M'87) received the B.S.E.E., M.S., and Ph.D. degrees in electrical engineering from The Ohio State University, Columbus, in 1982, 1984, and 1987, respectively.

He is been an Associate Professor with the School of Electrical Engineering at Purdue University, West Lafayette, IN. His primary research interests are in the simulation and control of kinematically redundant systems.



John J. Fox (S'93) received the B.S. degree in electrical engineering in 1986 from Carnegie Mellon University, Pittsburgh, PA, in 1986 and the M.S. degree in electrical engineering in 1988 from Pennsylvania State University, University Park, PA. He is currently pursuing the Ph.D. degree at Purdue University, West Lafayette, IN, where his areas of research include robotic path planning, redundant manipulators, and programming environments for robots.

From 1986 to 1987, he worked on autopilot design for Texas Instruments Missile Systems Division.