Complete Path Planning for a Planar 2-R Manipulator With Point Obstacles

G.F. Liu, J.C. Trinkle
Department of Computer Science
Rensselaer Polytechnic Institute
Troy, New York 12180–3590
Email: { liugf,trink }@cs.rpi.edu

R.J. Milgram
Department of Mathematics
Stanford University, CA 94305
Email: milgram@math.stanford.edu

***Abstract*— In this paper we develop a systematic topological approach to motion planning for a planar 2-R manipulator with point obstacles. By considering components in the free space for the second joint as the first joint varies, we build a two-dimensional array representing the cells of the free space and an asociated graph representing the boundaries of those cells. Using this graph, we derive a closed formula for the number of components of the free space. At the same time we solve the motion existence problem, namely, when are two arbitrary configurations in the same component? If so, we develop two explicit algoritms for constructing the path - a middle path method and a linear interpolation method. These algorithms give complete solutions to the path planning problem. Extensive examples are worked out which verify the correctness and efficiency of the resulting program. Then we briefly discuss how these methods generalize to a 3-R planar manipulator.**

I. Introduction

Three of the major problems in path planning for a robotic manipulator with obstacles are : (i) Determine the path components in the space of collision free configurations? (ii) Determine when two arbitrary points are in the same component? (iii) If they are in the same component, plan a path connecting them in the space of collision free configurations.

The most general motion planning scheme is the roadmap method, developed by Canny [1]. This method is exact and complete. However, its complexity is exponential and it has never been successfully implemented for practical examples. A more practical method is the Probabilistic Roadmap Method (PRM) [3], which is, however, not complete. [4], [6] employed topological methods for generating configuration space obstacles for path planning, while [5], [7] directly computed analytically the boundary of configuration space obstacles. Maciejewski and Fox [11] proposed a simple test for determining the connectedness of two configuration space obstacles, based on which a collision-free path was calculated.

The goal of this paper is to provide a closed form solution to Problem (i) and polynomial time algorithms for Problems (ii) and (iii) for a planar 2-R manipulator with point obstacles satisfying some weak assumptions. A minimal cell decomposition of the free configuration space is algorithmically constructed so that the boundaries of the cells are exactly the configurations that collide with the obstacle set. Thus the open cells are the components of the configuration space of collision free configurations. Next, methods from algebraic topology are applied to solve problems (i) and (ii). Simulations of various examples verify our approach.

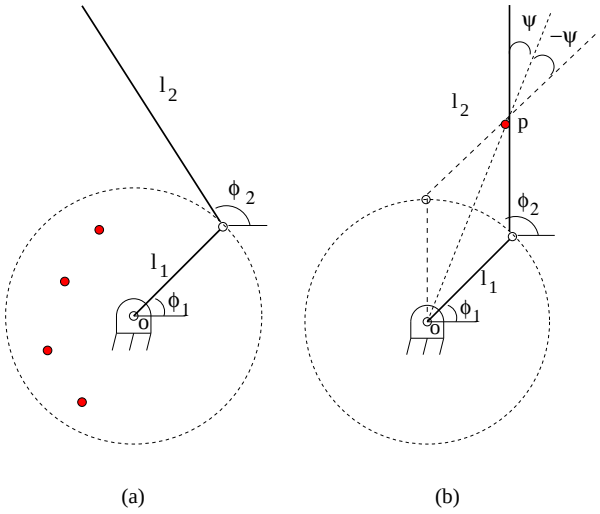


Fig. 1. (a) A planar 2-R manipulator with point obstacles (b) A triangle oAp forms when link 2 collides with p

II. Calculation of the Number of Components

In this section, we prove several important properties of the free configuration space of a planar 2-R manipulator with point obstacles. The end result is a closed formula for the number of components of the configuration space.

Let ϕ_1, ϕ_2 be the angles of the two joints, and l_1, l_2 be the link lengths of the manipulator as shown in Fig. 1-(a). If we ignore point obstacles, the configuration space of the manipulator is simply a torus $T^2 = S^1 \times S^1$. A rectangle $[-\pi, \pi] \times [-\pi, \pi]$ with the parallel boundary lines identified can be used as a picture for T^2. Suppose that the workspace of the manipulator contains a finite set of point obstacles $\mathcal{O} = \{p_1, p_2, \cdots, p_n\}$. The configurations that intersect the point obstacles form an arrangement of curves in T^2. We make the following assumption for convenience. It gives a maximally complex situation where all the collision curves will be closed. When the assumption is weakened the arrangement of curves becomes simpler. Some curves open up, others disappear.

Assumption *1:* $l_2 > 2l_1$ and $0 < l_{p_i} := \|p_i\| < l_1$ for all i. Moreover $\mathcal{O}$ is a generic set of point obstacles, i.e., (i) no three points are collinear; (ii) If $\mathcal{S}$ is the set of lines

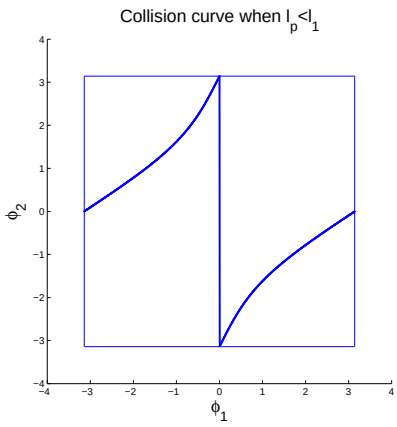


Fig. 2. The collision curve when $l_p < l_1$

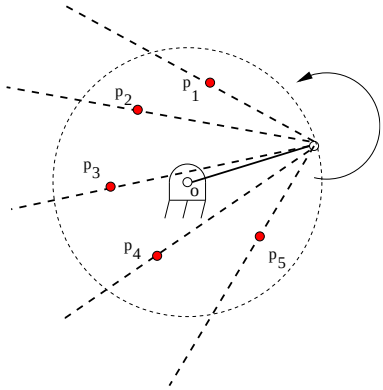


Fig. 3. Fixing the first joint and rotating the second joint counterclockwise, the manipulator hit $\mathcal{O}$ n times

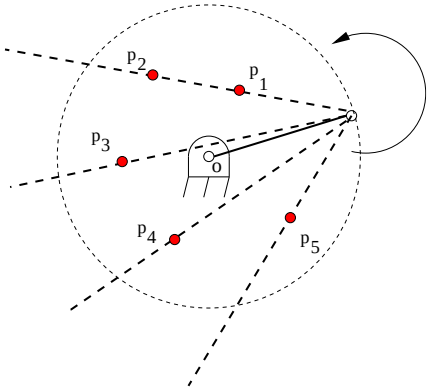


Fig. 4. Fixing the first joint and rotating the second joint counterclockwise, the manipulator hit $\mathcal{O}$ n-1 times

that contain 2 points of $\mathcal{O}$, then the point of intersection of any two lines in $\mathcal{S}$ does not lie on the circle of radius l_1 about the origin.

Let V_{p_i} be the collision curve which consists of all the configurations where the manipulator intersects p_i. V_{p_i} is the union of two curves, the first a circle, where ϕ_2 is arbitrary and l_1 intersects p_i, while the second also a circle consisting of all configurations where p_i intersects l_2. These two circles have a single point in common. It is direct to give equations for these curves. Fig. 2 gives an explicit V_{p_i}. Taking account of the identifications of the parallel edges, this graph represents two circles with common point on the x-axis, i.e., a "figure 8" or $S^1 \vee S^1$.

A. A Key Property of V_{p_i}

Let $C_{V_{p_i}}$ be the complement of V_{p_i} on T^2, i.e., $C_{V_{p_i}} = T^2 - V_{p_i}$.

Proposition *1:* $C_{V_{p_i}}$ is the product of two open intervals.

Proof: The projection of $C_{V_{p_i}}$ to the ϕ_1 axis will be the interval $S^1 - \{\theta_1\}$, where θ_1 is the angle of the first joint for which the first link lies on the line joining p_i to the origin. For each $\phi_1 \in S^1 - \{\theta_1\}$, there is only one configuration (ϕ_1, θ_2) at which the second link of the manipulator will collide with p_i. Therefore, the inverse image of ϕ_1 will be an open interval $S^1 - \{\theta_2\}$. This decomposition is locally a product so the total space of $C_{M_{p_i}}$ fibers over the interval with fiber an interval. Since any fibration over the interval is a product of the fiber with the interval, the proposition follows. □

Remark *1:* This result generalizes directly to m-R manipulators as long as for every configuration of the first $m-1$ joints, there is an angle for the remaining joint so that that link will pass through p_i. The complement of $C_{V_{p_i}}$ on $T^m = S^1 \times \cdots \times S^1$ will be the direct product of m open intervals, and V_{p_i} will be the $m-1$-skeleton of T^m, the union of the m coordinate $m-1$ torii of T^m.

By Proposition 1, we know that if there is one point obstacle, the number of components of $T^2 - \text{Sing}$ is one because $T^2 - \text{Sing}$ is connected. Of course, things change considerably when there are multiple point obstacles.

B. Counting the Number of Components

Define the singular set, $\text{Sing} \subset T^2$, as the set of all configurations of the manipulator that contain at least one obstacle of $\mathcal{O}$. Clearly,

$$\text{Sing} = \cup_i V_{p_i}.$$

Sing is a graph given as a union of circles in T^2 but will be a union of sub-tori, T^{m-1}, in T^m. To compute the number components of $T^2 - \text{Sing}$, we proceed as in the proof above. We project $T^2 - \text{Sing}$ onto S^1, $\pi_1 \colon T^2 - \text{Sing} \mapsto S^1$ by $(\theta_1, \theta_2) \mapsto \theta_1$. The image, $B \subsetneq S^1$, consists of n disjoint open intervals, $B = S^1 - \{\theta_{1,1}, \cdots, \theta_{1,n}\}$ where $\theta_{1,i}$, $i = 1, \cdots, n$, is the angles of the first joint when that link 1 contains p_i. Now, we look at the inverse image of π_1 over each point of B.

Proposition *2:* Under Assumption 1, there are exactly $n(n-1)$ points $\phi_j \in B$ over which the inverse image, $\pi_i^{-1}(\phi_j)$ consists of $n-1$ open intervals. For all the remaining points $\phi \in B$, $\pi_1^{-1}(\phi)$ consists of exactly n open intervals.

Proof: When the angle of the first joint is constrained to be $\phi_1 \in B$, as ϕ_2 varies from 0 to 2π the second link will, in general, hit $\mathcal{O}$ exactly n times as shown in Fig. 3. However, when ϕ_1 is the angle made by one of the $n(n-1)$ intersections of the circle about the origin traversed by the endpoint of the first link with the $\binom{n}{2}$ lines through two distinct points of $\mathcal{O}$, there will only be $n-1$ angles, ϕ_2, where the second link intersects $\mathcal{O}$. See

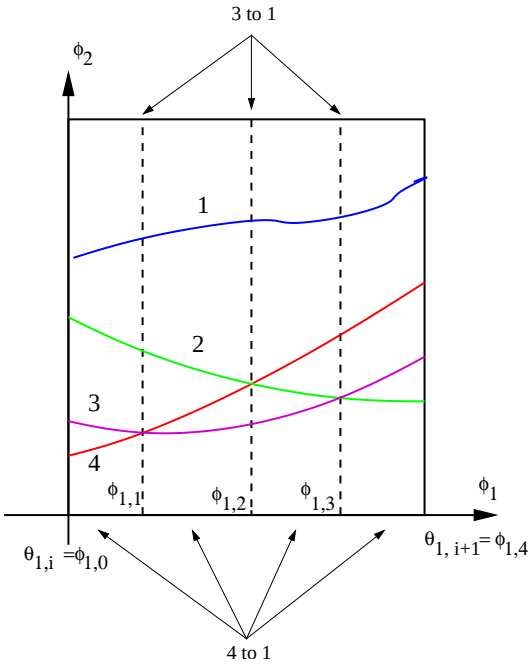
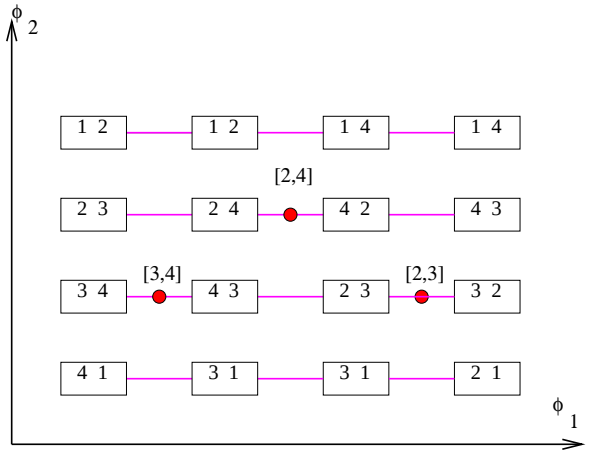


Fig. 5. Component sheaves above ϕ_1 axis



[a,b] means curve a intersects with curve b at this point

Fig. 6. A two-dimensional array representing the relations between components in Fig. 5

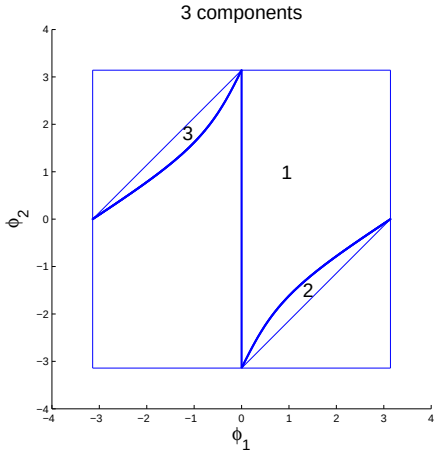


Fig. 7. 3 components for one point obstacle satisfying $l_{p_1} < l_1$

Fig. 4. (This is due to (ii) in 1.) □

Corollary *1:* Let $C \subsetneq B$ be the set of angles, ϕ_1, where $\pi^{-1}(\phi_1)$ is not equal to $(n-1)$ open intervals. Then C is a disjoint union of $\binom{n+1}{2}$ open intervals and $\pi^{-1}(C)$ fibers over C with fiber n open intervals.□

Since a fibration over an interval is a product, it follows that $\pi^{-1}(C)$ is the product of n intervals with $\binom{n+1}{2}$ intervals, or $\pi^{-1}(C)$ is exactly the disjoint union of $n \times \binom{n+1}{2}$ open rectangles.

What happens at one of the $n(n-1)$ points where there are $(n-1)$ intervals in the inverse image, as Fig. 5 and Fig. 6 show, is that one component ends on each side, and the remaining $n-1$ components extend across the line. The procedure indicated in Fig. 5 and Fig. 6 allows us to replace $T^2 - \mathrm{Sing}$ by a union of open curves $\mathcal{G}$ together with a projection $p\colon \mathcal{G} \longrightarrow B$.

To obtain these curves, each open interval in $\pi^{-1}(B)$ is replaced by a point, - perhaps thought of as the midpoint of that interval - so each inverse image is replaced by either n or $(n-1)$ distinct ponts. Moreover, replacement points in the inverse images of nearby points of B are close together if and only if the intervals they correspond to lie in the same component of $T^2 - \mathrm{Sing}$. Note that the set of components of $\mathcal{G}$ is exactly identified with the set of components of $T^2 - \mathrm{Sing}$. In fact, $\mathcal{G}$ is a deformation retract of $T^2 - \mathrm{Sing}$.

As an example, we consider the components pictured in Fig. 5 and Fig. 6. Here, the curves in Sing are labeled 1, 2, 3, 4, and the components in the inverse image of the $\phi_i \in C$ are labeled by the two curves that bound them, choosing the orientation going from top to bottom, so in the first region the labeling is $[4,1]$ for the region that contains the point on the x-axis, which is identified with the point where $y = 2\pi$ at the top, while $[3,4]$ represents the component that pinches off at $\phi_{1,1}$. In the second region the labeling of the region containing the x-axis is $[3,1]$, while the region on the other side of the pinch is labeled $[4,3]$. The named regions over each segment of C are joined by a line if they are represent the same component. Otherwise, the line is broken. Note that they represent the same region if at least one of the two ordered coordinates is the same, and a break occurs when the two coordinates are the same, but the order reverses, as happens with $[3,4]$ in the first region and $[4,3]$ in the second.

From Fig. 6, we see that the number of components above an interval in B is $n + k_i$ with k_i being the number of intersections between collision curves which lie in this region. Whenever two collision curves intersect in this region, the number of components increases by one. Since there are n open intervals in B, and $n(n-1)$ total intersections the total number of components of $T^2 - \mathrm{Sing}$ is $\sum_{i=1}^{n}(n+k_i) = n^2 + \sum_i k_i = n^2 + n(n-1) = 2n^2 - n$.

Theorem *1:* If Assumption 1 is satisfied and we ignore self-intersections, then the number of components of $T^2 - \mathrm{Sing}$ is $2n^2 - n$. The number of components when self-intersections are not allowed is $2n^2 + n$.

Proof: It suffices to take self-intersections into account. Self-intersections give rise to a single additional curve, γ, in the singular set. It is easily seen that the coordinates

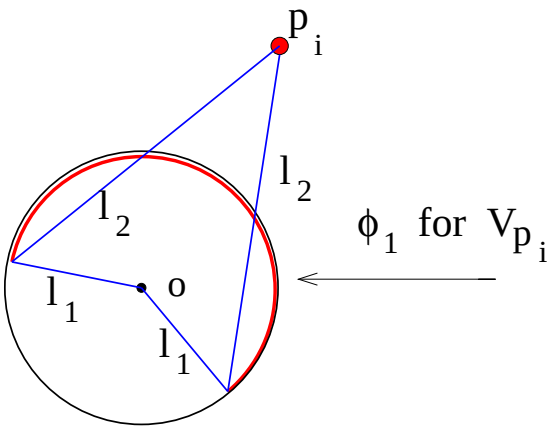


Fig. 8. V_{p_i} is an arc for $l_2 + l_1 > |p_i| > l_2 - l_1$

on γ $(\phi, \phi + \pi)$, and for each $p_i \in \mathcal{O}$ there are exactly two ϕ so that the configuration $(\phi, \phi + \pi)$ contains p_i. Consequently, each of the V_{p_i} intersects γ exactly twice, and the number of components now becomes $n(n+1) + n(n-1) + 2n = 2(n^2 + n)$, since, except for $n(n-1) + 2n$ points of B the number of components over each point is now $n + 1$. □

Example *1:* **Number of components when there are** 1 **or** 2 **point obstacles satisfying** $l_{p_i} < l_1$

Consider the case of a 2-R manipulator subject to 1 point obstacle satisfying $l_{p_1} < l_1$. Suppose Assumption 1 is satisfied. Then the number of disconnected components will be $2 + 1 = 3$, as shown in Fig. 7. If there are 2 obstacles satisfying $l_{p_1}, l_{p_2} < l_1$, and o, p_1, p_2 are not in the same line, then this number will be $2 \cdot 4 + 2 = 10$.

Weakening Assumption 1. Suppose that some of the $p_i \in \mathcal{O}$, (still in general position), lie outside of the l_1-circle (but at least one lies inside). If $p_i \in \mathcal{O}$ satisfies $l_1 + l_2 > |p_i| > l_1$, then V_{p_i} is a single circle when $l_2 - l_1 > |p_i| > l_1$ and otherwise is an arc bounded by the two configurations where the end-point of l_2 equals p_i as Fig. 8 illustrates. If it is a circle, then each point of B has exactly one point of V_{p_i} over it. If it is an arc, the two angles ϕ_1 for the bounding points have to be added to the special points in B, since, at these angles, the points of V_{p_i} will only exist on one side.

The previous analysis is changed in two ways. First one has to explicitly determine, for all $p_i \neq p_j \in \mathcal{O}$ with $l_1 + l_2 > |p_i| > l_1$ the structure of $V_{p_i} \cap V_{p_j}$.

- This intersection will be empty if the line joining p_i and p_j does not intersect the l_1-circle, or if it intersects it in two points, both of which lie *between* p_i and p_j.
- It will consist of 2 points if the line intersects the l_1-circle in two points, both of which lie *outside* the segment between p_i and p_j.
- Otherwise, it will consist of 1 point.

The second way in which the analysis changes is that the graph in Fig. 6 *bifurcates* at the first boundary point of the V_{p_i}-arc, and *amalgamates* at the second. See Fig. 9

III. Algorithm for the Existence Problem

Now that the components have been determined, the next step is the motion planning existence problem:

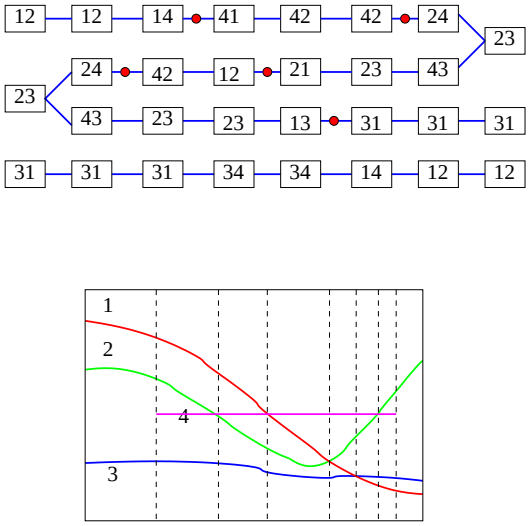


Fig. 9. An outside p_i, the associated intersection curves, and array

Problem *1:* **Motion planning existence problem**

Given two arbitrary configurations of the manipulator $\phi(0) = (\phi_1(0), \phi_2(0))$ and $\phi(T) = (\phi_1(T), \phi_2(T)$, both in $T^2 - \text{Sing}$, determine whether $\phi(0)$ and $\phi(T)$ are in the same component.

In this section, we construct an efficient algorithm for Problem 1 with n point obstacles. We assume that Assumption 1 is satisfied. In the previous section, we constructed an array and an associated union of curves. Using this construct, we can assign numbers to nodes of this array to label the distinct components. denote the i^{th} row, j^{th} column node, as $\text{Array}[i, j]$. The graph allows us to associate a component field, *comp*, to each node, numbered lexicographically. Using Fig. 6 as an example we have

$$\begin{array}{lll} \text{Array}[1,i].comp = 1 & , & i = 1, \cdots, 4 \\ \text{Array}[2,1].comp = 2 & , & \text{Array}[2,2].comp = 2 \\ \text{Array}[2,3].comp = 3 & , & \text{Array}[2,4].comp = 3 \\ \text{Array}[3,1].comp = 4 & , & \text{Array}[3,2].comp = 5 \\ \text{Array}[3,3].comp = 5 & , & \text{Array}[3,4].comp = 6 \\ \text{Array}[4,i].comp = 7 & , & i = 1, \cdots, 4. \end{array}$$

Given $\phi(0), \phi(T) \in T^2 - \text{Sing}$, we initially check whether their first coordinates lie in the same component of B. If not, they are not in the same component. If so, we check the array to find their respective components. $\phi_1(0)$, $\phi_1(T)$ determine the columns in the array, while $\phi_2(0)$, $\phi_2(T)$ determine the rows. Checking the assigned numbers answers Problem 1. A general algorithm that ignores self-intersections is summarized as follows, though the self-intersections are easily accounted for by simply using the associated array with the self-intersection curve considered as well as the V_{p_i}.

Algorithm *1:* **Algorithm for the Existence Problem**

Input: n point obstacles p_i, $i = 1, \cdots, n$, a planar 2-R manipulator satisfying Assumption 1, and two arbitrary configurations $\phi(0)$ and $\phi(T)$ representing, respectively, the initial and final configurations;

Output: True if the two configurations are in the

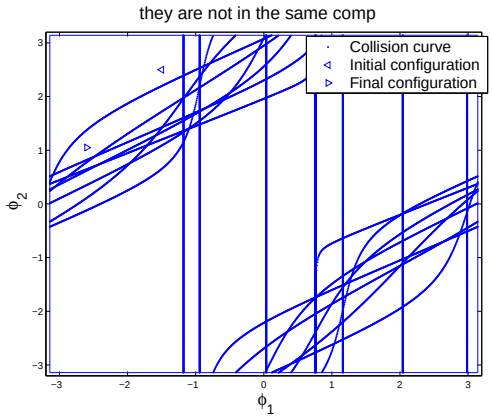


Fig. 10. $\phi(0)$ and $\phi(T)$ are not in the same component

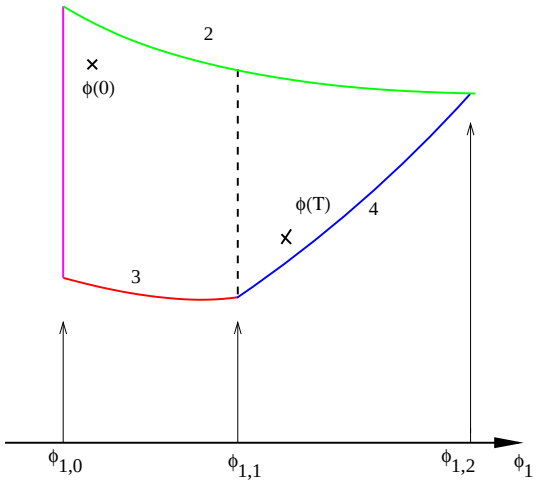


Fig. 11. Cell Structure

same component, otherwise, False

Step 1: Determine the components in B that contain $\phi(0)$ and $\phi(T)$ by comparing $\phi_1(0)$ and $\phi_1(T)$ with the $\theta_{1,i}$. If they are not in the same interval, return False and stop;

Step 2: If they are in the same interval, return the pair $(\theta_{1,i}, \theta_{1,i+1})$ that determines the interval;

Step 3: For this interval calculate the intersection points, i.e., all the intersections of the lines connecting two distinct points of $\mathcal{O}$ with the interval $\theta_{1,i} < \phi < \theta_{1,i+1}$. For each intersection point, record the angles

$$Insc(i).\phi = (\phi_1, \phi_2)$$

and the intersection points in question p_i, p_j are stored as $Insc(i).index(1), Insc(i).index(2) \in \{1, \cdots, n\}$ with the closer point stored in $index(1)$. (For ϕ slightly less than ϕ_1, the second coordinate of the configuration (phi, ϕ_2) that intersects the point in $index(1)$ will be *less* than the corresponding angle for the point in $index(2)$.);

Step 4: Sort all the intersection points in this interval on ϕ_1;

Step 5: If there are m intersection points in this interval, construct the associated $n \times (m+1)$ array: $Array$.

Step 6: Fix $\phi_1 \in (\theta_{1,i}, \phi_{1,1})$, and calculate the order in which link 2 hits the n point obstacles when it rotates counterclockwise;

Step 7: For $i = 2, \cdots, m$, write down, successively, the order for these columns. Simply change the order of the previous column by permuting the two indices in $Insc(i).index(1)$ and $Insc(i).index(2)$;

Step 8: Label the nodes of $Array$ according to Fig. 6;

Step 9: The labels at nodes in successive columns will agree if at most one number between two respective labels is changed. Otherwise, if the order of the labeling pair is reversed, the node in the new column will receive a new component label;

Step 10: Identify the nodes corresponding to the initial and final configurations. If the associated labels are the same, they lie in the same component and *true* is returned. Otherwise *false* is returned.

It is not difficult to see that steps 7 and 10 take the most time. Since the number of intersection points in a given B component is $2(n^2 - n/n) = n - 1$ on average, the average complexity of the above algorithm will be $O(n^2)$. Table I shows the computation time with respect to the number of point obstacles.

No. of point obstacles	2	4	6	8
Computation time (s)	0.031	0.047	0.0620	0.0630
No. of point obstacles	10	20	30	40
Computation time (s)	0.0780	0.1090	0.2970	0.7970

TABLE I

COMPUTATION TIME FOR DIFFERENT NUMBER OF POINT OBSTACLES

Example *2:* **A Planar** 2**-R Manipulator with** 8 **Obstacles**

First we apply Algorithm 1 to the system with the following point obstacles:

$$p_1 = [-7.3222, 1.1544]^T,\ p_2 = [3.2630, 7.5101]^T$$
$$p_3 = [4.0698, 3.8988]^T,\ p_4 = [-1.1067, 2.1812]^T$$
$$p_5 = [4.9228, -6.7600]^T,\ p_6 = [7.2367, 6.7572]^T$$
$$p_7 = [1.6568, -4.0035]^T,\ p_8 = [6.0669, 0.2177]^T.$$

and link lengths, $l_1 = 10$ and $l_2 = 20$. Algorithm 1 gives that

$$\phi(0) = [-1.5, 2.5]^T\ ,\ \phi(T) = [-3, -3]^T$$

are in the same components, while

$$\phi(0) = [-1.5, 2.5]^T\ ,\ \phi(T) = [-2.6, 1.05]^T$$

are in different components. Fig. 10 helps explain this.

IV. Algorithms for Path Planning

In this section, we will develop two algorithms for planning a path from $\phi(0)$ to $\phi(T)$ when they are in the same component.

In the previous section, Algorithm 1, constructs a two-dimensional array $Array$ for each component of B. Given any point in this region, there is a unique node in $Array$ corresponding to the component that contains the poin. In fact, as shown in Fig. 5, Fig. 6, Fig. 9, each node of $Array$ represents a cell-like,[1] region in $T^2 - \text{Sing}$, (or $T^2 - \text{Sing} - W$, where W represents the self-intersection curve). The connecting lines between nodes show that the two regions have a common arc on their boundaries.

For example, Fig. 11 shows one component of Fig. 5, which is the union of two of these regions. The two cells-like regions are represented by $Array[2,1]$ and $Array[2,2]$, respectively. They are connected since there is a line segment connecting them, as shown in Fig. 6. With this in mind we next describe how one can plan a path linking the two points $\phi(0)$ and $\phi(T)$ shown in Fig. 11, where $\phi(0)$ locates in $Array[2,1]$, and $\phi(T)$ in $Array[2,2]$.

A. Middle Path Method

Cell 1 (or $Array[2,1]$) is labelled as $(2,3)$, which means this cell is bounded from above by V_{p_2}, and from below by V_{p_3}. With the usual metric on $T^2 - \text{Sing}$, over each point$\phi \in B$, each arc in $\pi^{-1}(\phi)$ has a unique mid-point, and each of the cell-like regions has a smooth midpoint curve. Moreover, these curves agree along the common arcs between regions. Here is the local equation for this arc:

$$\begin{aligned} \overline{\phi}_2 &= atan2(p_2(2) - l_1 \sin\phi_1, p_2(1) - l_1\cos\phi_1) \\ \underline{\phi}_2 &= atan2(p_3(2) - l_1 \sin\phi_1, p_3(1) - l_1\cos\phi_1) \\ \phi_2 &= \tfrac{\overline{\phi}_2 + \underline{\phi}_2}{2}, \end{aligned}$$

for $\phi_1 \in [\phi_{1,0}, \phi_{1,1}]^T$. Here $p_2, p_3 \in \mathbb{R}^2$ are the Cartesian coordinates of obstacles p_2 and p_3. Similarly, in cell 2 (or $Array[2,2]$) which is labelled $(2,4)$, the middle path will be

$$\begin{aligned} \overline{\phi}_2 &= atan2(p_2(2) - l_1 \sin\phi_1, p_2(1) - l_1\cos\phi_1) \\ \underline{\phi}_2 &= atan2(p_4(2) - l_1 \sin\phi_1, p_4(1) - l_1\cos\phi_1) \\ \phi_2 &= \tfrac{\overline{\phi}_2 + \underline{\phi}_2}{2}, \end{aligned}$$

for $\phi_1 \in [\phi_{1,1}, \phi_{1,2}]^T$. A path from $\phi(0)$ to $\phi(T)$ can be constructed as follows: first move vertically from $\phi(0)$ to the mid-point of its arc in $pi^{-1}(\phi_1(0))$, then move on the mid-point curve to a point m with $\phi_2(m) = \phi_2(T)$, and then move vertically to $\phi(T)$.

We apply the mid-point algorithm to a case with 10 point obstacles in the workspace. The resulting path is shown in Fig. 12.

B. Linear Interpolation

The path obtained using the mid-point curves is continuous, but not generally smooth when the curve crosses between cell-like regions or where the mid-point curve intersects the two verticals. We can remove these last two breaks in the first derivative by using linear interpolation, so that the paths from the intial and final points meet the mid-point curve at the boundary with the next cell-like region or further on.

[1]More exactly, each of these regions is contractible

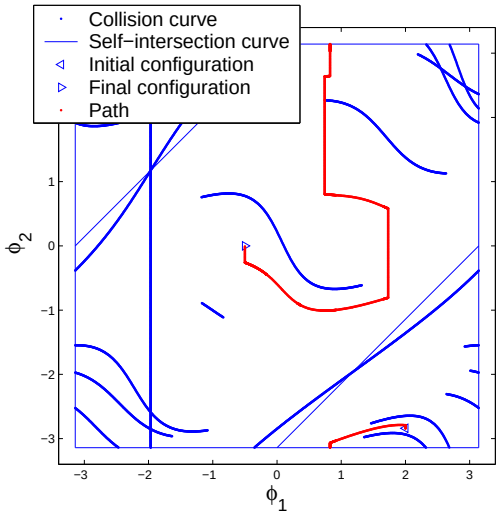


Fig. 12. Path planning using the mid-point curve

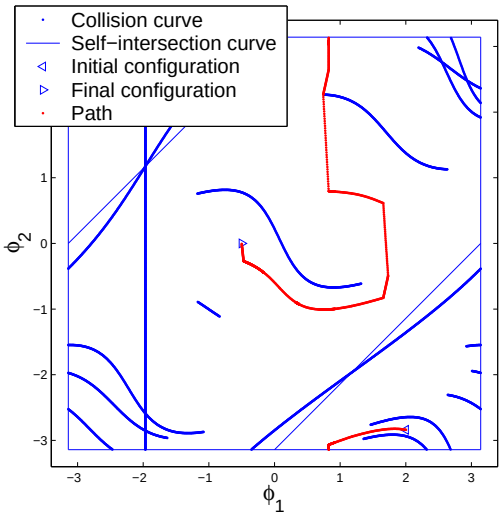


Fig. 13. Path planning using linear interpolation

We apply linear interpolation to another case with 10 point obstacles in Fig. 13. Note that this path is still not smooth at the intersections of cell-like regions. However, it is routine to modify these curves near the intersection points to make their derivatives agree to whatever level is desired.

V. Short Discussion of the General Theory

Underlying the algorithms above is the use of Poincaré duality to determine the number of components in an oriented manifold, M, when a closed subset, X, is deleted. The 0-dimensional (singular) homology group of any space, Y, $H_0(Y, \mathbb{R})$ has the form $\mathbb{R}^n$ where n is the number of path components in Y. When M is a k-dimensional, closed, oriented manifold with no boundary, and $Y = M - X$ for X a reasonable, proper, closed subspace, then

$H_0(M - X, \mathbb{R}) \cong H^k(M, X, \mathbb{R})$. On the other hand, this last group is determined by the exact sequence of the pair

$$\cdots \longrightarrow H^{n-1}(M, \mathbb{R}) \longrightarrow H^{k-1}(M, \mathbb{R}) \longrightarrow H^k(M, X, \mathbb{R}) \longrightarrow H^k(M, \mathbb{R}) \longrightarrow 0$$

We can apply this for $M = T^2$ and X either Sing, or $X = \text{Sing} \cup W$. Moreover, under Assumption 1, the cohomology map $H^1(T^2, \mathbb{R}) \longrightarrow H^1(X, \mathbb{R})$ has image a copy of $\mathbb{R}^2$ from Proposition 1. Hence, the exact sequence becomes

$$0 \longrightarrow \mathbb{R}^2 \longrightarrow H^1(X, \mathbb{R}) \longrightarrow H^2(T^2, X, \mathbb{R}) \longrightarrow \mathbb{R} \longrightarrow 0$$

since $H^1(T^2, \mathbb{R}) = \mathbb{R}^2$, and $H^2(T^2, \mathbb{R}) = \mathbb{R}$. From this it follows that $H^1(X, \mathbb{R}) = \mathbb{R}^{n+1}$, where n is the number of path components in $T^2 - X$. More generally, if there is at least one point $p_i \in \mathcal{O}$ with $|p_i| < l_1$ then the map $H^1(T^2, \mathbb{R}) \longrightarrow H^1(X, \mathbb{R})$ has image $\mathbb{R}^2$. If $|p_i| > l_1$ for all $p_1 \in \mathcal{O}$, but there is at least one p_j so that $l_2 - l_1 > |p_i| > l_1$, then the image is $\mathbb{R}$. Otherwise, the image is 0.

An explicit method of determining an element in $H^1(X, \mathbb{R})$ associated to two points in $T^2 - X$ is to take any smooth path, τ, between the two points so that its intersection with X does not contain any points in the intersections $V_{p_i} \cap V_{p_j}$ for $i \neq j$, and which intersects X transversally. (This is easily arranged since the intersections of the curves in Sing or $\text{Sing} \cup W$ are a finite set of distinct points.) Any two such paths give a closed path in T^2, and hence the difference between any two choices of path give an element of $H_1(T^2, \mathbb{R})$.

The intersection of τ and X defines a homomorphism of the 1-chains of the cell decomposition of X (where the 0-cells are the intersections of the curves in X, together with their boundaries, and the one cells are the components of the complement of the 0-cells) to $\mathbb{R}$. This homomorphism is given by assigning to each edge in X the sum of the number of intersections of the path with the edge, each intersection being counted as $+1$ or -1 depending on whether the orientation at the point given by first taking the vector along the path from the initial to final point and then the vector at the edge given by a chosen but fixed orientation of the edge, agree with the orientation of T^2 or not.

This homomorphism represents a class $\gamma \in H^1(X, \mathbb{R})$, well defined up to the image of $H^1(T^2, \mathbb{R})$. *Moreover, the two points lie in the same component of $T^2 - X$ if and only if $\gamma \in im(H^1(T^2, \mathbb{R}))$.* This is somewhat non-intuitive, but is a standard technique in mathematics. [12], [13] One thing that may help is to notice that the only way $H^1(X, \mathbb{R}) > \mathbb{R}^2$ is if there are at least two components in $T^2 - X$ - if there is only one component, then γ must lie in the image of $H^1(T^2, \mathbb{R})$.

This theory extends without essential change to n-links, $n \geq 3$. The only difference is that calculations tend to become quite long and involved, since the space of configurations that contain points in $\mathcal{O}$ can become very complicated for larger n.

VI. Conclusion

This paper developed a systematic topological approach to motion planning of a planar 2-R manipulator with point obstacles. By considering components of the configuration space that lie above the indiviual points ϕ_1, we constructed a two-dimensional array representing the cells of the free space and a graph describing the connected components. Using this graph, we derived a closed formula for the number of components of the configuration space under Assumption 1, and then solved the motion existence problem.

When Assumption 1 is removed, we need to consider cases where some collision curves are not closed. This is done by the process pictured in Fig. 9.

If we add a link (3-link manipulators), we consider all components above each 2-dimensional component in the 2-dimensional picture for the configuration space of the manipulator consisting of only link 1 and 2. Two configurations are in the same component only if when projecting to the (ϕ_1, ϕ_2) space, they are in the same component, but the inverse image of each point in each component is a union of disjoint arcs. Where the numbers of arcs change is when the endpoint of the first two links lies on one of the lines connecting two distinct points $p_i, p_j \in \mathcal{O}$. Moreover, the endpoint must lie on the *same side* of both points. These give curves in T^2 that are, under weak assumptions, circles that wind along the first coordinate. Two of these circles intersect if and only if the two lines intersect in the workspace of the first two links, but the determination of the intersections of these circles with the V_{p_i} for the first two links is more involved.

In a forthcoming paper, we will address these remaining problems in detail.

References

[1] J.F. Canny, *The Complexity of Robot Motion Planning*. Cambridge, MA: MIT Press, 1988.

[2] J. Schwartz, J. Hopcroft, and M. Sharir, *Planning, Geometry, and Complexity of Robot Motion*. Ablex, 1987.

[3] L.E. Kavraki, P. Švestka, J.C. Latombe, and M.H. Overmars, *Probablistic Roadmaps for path planning in high-dimensional configuration space*. IEEE Transactions on Robotics and Automation, 12(4):566-580, 1996.

[4] C. Bajaj and M.S. Kim, *Generation of configuration space obstacles: The case of a moving sphere*. IEEE Jounral on Robotics and Automation, RA-4:94-99, 1988.

[5] M.S. Branicky and W.S. Newman, *Rapid computation of configuration space obstacles*. in Proceedings of IEEE International Conference on Robotics and Automation, 304-310, 1990.

[6] R.C. Brost, *Computing metric and topological properties of configuration-space obstacles*. in Proceedings of IEEE International Conference on Robotics and Automation, 170-176, 1989.

[7] W. Meyer and P. Benedict, *Path Planning and the geometry of joint space obstacles*. in Proceedings of IEEE International Conference on Robotics and Automation, 215-219, 1988.

[8] C. Nash and S. Sen, *Topology and Geometry for Physicists*. Academic Press, 1983.

[9] W. Fulton, *Algebraic Topology: A First Course*. Springer-Verlag, 1995.

[10] J.R. Munkres, *Elements of Algebraic Topology*. Addison-Wesley Publishing Company, 1995.

[11] A.A. Maciejewski and J.J. Fox, *Path Planning and the Topology of Configuration Space*. IEEE Transactions on Robotics and Automation, Vol. 9, No. 4, Aug., 1993.

[12] F.R. Cohen, R.L. Cohen, B.M. Mann, R.J. Milgram, *The topology of rational functions and divisors of surfaces*, Acta Math., Vol. 166(1991), 162-221.

[13] V. A. Vassiliev, *Complements of Discriminants of Smooth Maps*. **Topology and Applications**, Amer. Math. Soc., (1992)