A Probabilistic Roadmap Approach for Systems with Closed Kinematic Chains

Steven M. LaValle Jeffery H. Yakey

Department of Computer Science
Iowa State University
Ames, IA 50011 USA
{lavalle,yakeyj}@cs.iastate.edu

Lydia E. Kavraki

Department of Computer Science
Rice University
Houston, TX 77005 USA
kavraki@cs.rice.edu

Abstract

We present a randomized approach to path planning for articulated robots that have closed kinematic chains. The approach extends the probabilistic roadmap technique which has previously been applied to rigid and elastic objects, and articulated robots without closed chains. Our work provides a framework for path planning problems that must satisfy closure constraints in addition to standard collision constraints. This expands the power of the probabilistic roadmap technique to include a variety of problems such as manipulation planning using two open-chain manipulators that cooperatively grasp an object, forming a system with a closed chain, and planning for reconfigurable robots where the robot links may be rearranged in a loop to ease manipulation or locomotion. We generate the vertices in our probabilistic roadmap by sampling random configurations that ignore kinematic closure, and by performing randomized gradient descent to force satisfaction of the closure constraints. We generate edges in the roadmap by executing a randomized traversal of the constraint surface between two vertices. In this paper, we focus our presentation on the problem of planning the motions for a collection of attached links in a two-dimensional environment with obstacles. The approach has been implemented and successfully demonstrated on several examples.

1 Introduction

The Problem We address the problem of path planning for an articulated robot or structure that has many degrees of freedom, closed kinematic chains and moves in a cluttered environment (see Figure 1). This structure could correspond to a single robot, or, for example, to a pair of open-chain manipulators that cooperatively grasp and manipulate an object [19]. It could also correspond to a closed loop that may result from the rearrangement of the links of a reconfigurable robot in order to ease manipulation or locomotion [10, 20, 24].

We are working towards having an efficient planner for systems with closed kinematic chains. Our planner is based on the probabilistic roadmap approach to planning [16]. We construct a graph on the portion of the configuration space that satisfies kinematic closure constraints. The vertices of the graph are obtained by sampling random configurations that ignore kinematic closure and performing randomized gradient descent to force satisfaction of the closure constraints. The edges of the roadmap are computed by executing a randomized traversal of the constraint surface between two vertices.

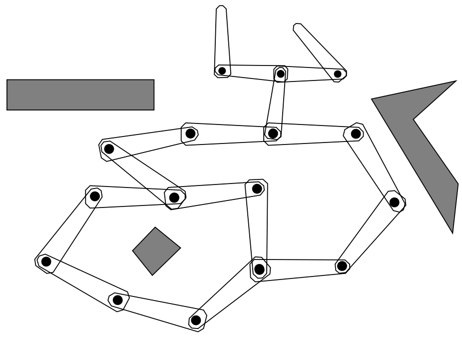


Figure 1: We consider structures that have closed kinematic chains and must avoid static obstacles.

Applications The immediate applications of our work are in systems with closed loops in which the connectivity of the loops does not change but the configurations of the loops change. One primary area of applications is manipulation planning. When two or more robots manipulate a single object, a closed kinematic chain is created involving the robots and the object. During manipulation, regrasping operations may be required to complete a task. A regrasping operation breaks the closed chain, while the object rests at a stable configuration, and forms a new closed chain with the robots grasping the object at different grasp positions. Between regrasping operations, there is a need to plan for the closed kinematic chain as a whole. Our approach provides an efficient way to compute the motion of the closed chain. Our approach may shed new light to manipulation planning and seed new research in that area. For example, manipulation planning algorithms consider all possible combinations of grasp positions [1], which can be very large, or first plan for the object and then use inverse kinematics to plan for the robots. In the latter case, regrasping is done in configurations where the arms can not move any further and frequently these configurations correspond to awkward configurations of the whole system.

Manipulation problems arise in industrial settings, virtual prototyping simulations, and in computer graphics animation. In graphics animation, closed kinematic chains are formed when a human-like character holds an object with two arms and carries the object, when two characters fight with each other, etc. A planner with the capability to handle closure constraints in addition to standard collision avoidance will be most useful in automating parts of the animation task. Furthermore, our work may have applications in planning for reconfigurable robots [20, 24]. These robots support multiple modalities of locomotion and manipulation. Closed chains are often present in locomotion gaits [24], or arise during the reconfiguration task. There is also is a growing interest in computational chemistry to develop tools that efficiently search the configuration space of flexible molecules [11]. The atoms and bonds can be modeled as joints and links, and the cyclic chains that commonly arise in molecules, impose closure constraints on the kinematics that describe atom positions for given configurations.

Use of Randomized Techniques There already exist complete algorithms that solve the general planning problem; therefore, our efforts to develop a randomized, incomplete planner need to be justified. The subset of the configuration space that satisfies kinematic closure constraints can generally be formulated as an algebraic variety (the zeros of the system of implicit polynomial equations). Many techniques exist that exactly and completely determine the structure of a variety [3, 12, 7]. Although there have been many interesting recent advances in computational algebraic geometry [22, 6], the complexity of these algorithms still limits their use to simpler problems. This observation has also been the basis for the development of randomized path planning techniques for rigid robots or articulated robots with open kinematic chains.

It is next important to ask whether existing randomized path planning techniques can be applied to our problem with minor adaptation. The major difficulty is that existing methods are designed for planning in a configuration space, which is assumed to be a parameterized manifold [21]. When closed kinematic chains exist, such luxuries disappear because path planning must be performed on a variety. In this case it is less straightforward to generate allowable configurations or local motions, which are needed for successful methods such as the potential field method [5, 8, 18] or probabilistic roadmaps [2, 17]. Our approach is to extend the probabilistic roadmap paradigm to generate a roadmap on a variety instead of a configuration space. This paradigm was chosen because of its recent success, its simplicity, and its ability to accommodate the case of open and closed chain mechanisms under a unified framework as explained in the next paragraph.

A General Framework Our longer-term goal is to develop a general, probabilistic roadmap framework for structures that include a complicated mixture of open and closed chains. The principles introduced in this paper serve as the closed-chain component, which can be combined with existing tools for handling open chains. In many applications the connectivity of the structure can vary frequently (e.g., a reconfigurable robot [10, 20, 24] or two robots and an object during a manipulation task [19]). When closed-chains are formed, our approach will produce vertices and edges that satisfy the constraints on the subset of configuration variables that are involved in the closed chains. For a given closed chain, our approach can also be used to precompute vertices and edges that satisfy the kinematic closure constraints, and could then be used in any system that contains the closed chain. Additional chains might be attached to this closed chain, or various obstacles might be added to the world, but the primary burden of exploring the subset of the configuration space that satisfies the kinematic closure constraints will already have been handled.

2 Problem Formulation

Our problem will be defined in a bounded 2D or 3D world, $\mathcal{W} \subset \Re^N$, such that $N = 2$ or $N = 3$. Let a *link*, L_i, be a rigid body in the world, which is considered as a closed, bounded point set. Let $\mathcal{L}$ be a finite collection of n_l links, $\{L_1, L_2, \ldots, L_{n_l}\}$. Let $J(L_i, L_j)$ (or simply J) define a *joint*, which indicates that the pair of links L_i and L_j are attached. Additional information is associated with a joint: the point of attachment for L_i, the point of attachment for L_j, the type of joint (revolute, spherical, etc.), and the range of allowable motions. Let $\mathcal{J}$ be a collection of n_j joints that connect various links in $\mathcal{L}$. Let $\mathcal{M} = (\mathcal{L}, \mathcal{J})$ define a *structure*. An example is shown in Figure 1. It will sometimes be convenient to consider $\mathcal{M}$ as a graph in which the joints correspond to vertices and the links correspond to edges. Therefore, let G_M denote the underlying graph of $\mathcal{M}$.

Now consider the kinematics of $\mathcal{M}$. Using a standard parameterization technique, such as the Denavit-Hartenburg representation [13, 15], the *configuration* of $\mathcal{M}$ can be expressed as a vector, q, of real-valued parameters. Let $\mathcal{C}$ denote the configuration space or $\mathcal{C}$-space. Let $\mathcal{M}(q)$ denote the transformation of $\mathcal{M}$ to the configuration given by q. Note that the graph G_M is a tree if and only if there are no closed kinematic chains. If G_M is a tree, then any configuration q yields an acceptable position and orientation for each of the links if we ignore collision issues and planning for such systems as has been considered in [5, 9, 14, 17].

In this paper, we are primarily concerned with the case in which $\mathcal{M}$ contains closed kinematic chains, which implies that G_M contains cycles. In this case, there will generally exist configurations that do not satisfy closure constraints of the form $f(q) = 0$. Constraints can be defined by breaking each cycle in G_M at a vertex, v, and writing the kinematic equation that forces the pose of the corresponding joint to be the same, regardless of which of the two paths were chosen to v. Let $\mathcal{F}$ represent the set, $\{f_1(q) = 0, f_2(q) = 0, \ldots, f_m(q) = 0\}$ of m closure constraints. In general, if n is the dimension of $\mathcal{C}$, then $m < n$. Let $\mathcal{C}_{cons} \subset \mathcal{C}$ denote the set of all configurations that satisfy the constraints in $\mathcal{F}$.

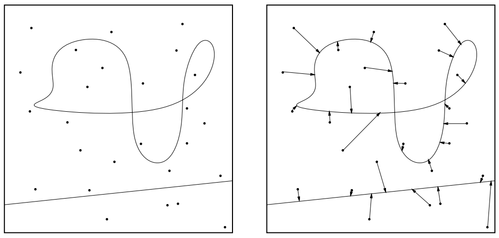
A collision is defined for $\mathcal{M}(q)$ if any of the links of $\mathcal{M}(q)$ collides with any of the workspace obstacles or the other links $\mathcal{J}$. Consecutive links usually do not give rise to collisions. Using standard terminology, let $\mathcal{C}_{free}$ denote the set of all configurations such that $\mathcal{M}(q)$ is not in collision. In addition to the usual complications of path planning for articulated bodies having many degrees of freedom, we are faced with the challenge of keeping the configuration in $\mathcal{C}_{cons}$. Although $\mathcal{C}$ is typically a manifold, $\mathcal{C}_{cons}$ will be more complicated. It can be expressed as an algebraic variety (the zero set of a system of polynomial equations), and it is assumed that a parameterization of $\mathcal{C}_{cons}$ is not available.

The task can now be defined as follows. Let $\mathcal{C}_{sat} = \mathcal{C}_{cons} \cap \mathcal{C}_{free}$, which defines the set of configurations that satisfy both kinematic and collision constraints. Let $q_{init} \in \mathcal{C}_{sat}$ and $q_{goal} \in \mathcal{C}_{sat}$ be the *initial configuration* and *goal configuration*, respectively. The task is to find a continuous path $\tau : [0, 1] \to \mathcal{C}_{sat}$ such that $\tau(0) = q_{init}$ and $\tau(1) = q_{goal}$. We also intend to build a network of paths that share common endpoints, forming a roadmap in $\mathcal{C}_{sat}$.

3 The General Approach

Our approach to path planning in $\mathcal{C}_{sat}$ follows the same philosophy presented in [17] for constructing a roadmap in $\mathcal{C}_{free}$. In that approach, a set of roadmap vertices is generated by selecting several thousand configurations at random, and keeping the ones that are not in collision. A distance metric is defined on the configuration space, and a neighborhood of vertices is determined for each vertex in the roadmap. A local planner (e.g., connect points $\mathcal{C}$ by a line segment) is used to attempt to connect each vertex to other vertices in its neighborhood. Collision detection is performed along the path determined by the local planner to determine whether the path should be added as an edge in the roadmap.

Each of these steps is considerably more challenging for the case of constructing a probabilistic roadmap in $\mathcal{C}_{sat}$. To determine the vertices, we can no longer simply guess random configurations because they will normally not lie in $\mathcal{C}_{sat}$. Since $\mathcal{C}_{sat}$ is generally of lower dimension than $\mathcal{C}$, the probability is zero that a randomly-chosen point will lie in $\mathcal{C}_{sat}$. In Section 3.1 we present a randomized descent algorithm that takes a randomly-chosen point in $\mathcal{C}$ and attempts to move it incrementally to $\mathcal{C}_{sat}$. Repeated use of this algorithm results in a set of vertices for our roadmap. To construct the edges, the local planner becomes more complicated because the path generated must satisfy the kinematic constraints. In Section 3.2 we present a randomized scheme that attempts to connect two nearby vertices while ensuring that the constraint is maintained to within a specified tolerance.



a. b.

Figure 2: (a) Configurations are first chosen at random in $\mathcal{C}$; the curves depict $\mathcal{C}_{cons}$. (b) Randomized error minimization is performed on the samples to force as many as possible onto $\mathcal{C}_{cons}$.

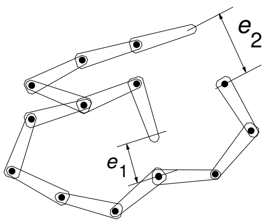


Figure 3: Each kinematic loop is broken, and $e(q)$ is measured in terms of Euclidean distances.

3.1 Generating the Roadmap Vertices

Figure 2 illustrates the problem of generating vertices in $\mathcal{C}_{sat}$. A random sample in $\mathcal{C}$ can be easily generated (of course its distribution depends on the parameterization of $\mathcal{C}$). The algorithm in Figure 4 gives pseudocode for a randomized descent technique that iteratively attempts to reduce an error function, $e(q)$, which can be defined in terms of each $f_i(q)$. If the constraints are expressed in polynomial form, the algebraic distance can be minimized, or an approximation to the Euclidean distance in $\mathcal{C}$ may easily be minimized [23]. We chose an alternative approach, though, which is to break the kinematic loops and minimize the sum of squares the Euclidean distances of each joint that is not where it should be to satisfy kinematic closure. See Figure 3.

The algorithm GENERATE_RANDOM_VERTEX requires three constants: ϵ, which is the numerical tolerance on the error function, I, which is the maximum number of search steps, and J which is the maximum number of consecutive failures to close the kinematic chains. Assume that RANDOM_NHBR always generates a random configuration in $\mathcal{C}_{free}$, and the distance between q and q' is fixed and small. RANDOM_NHBR may have to guess many nearby configurations to produce one that is collision-free. Rather than compute a complicated gradient of $e(q)$, any random configuration that is better than q is kept. This was observed in [4] to be much faster than computing an analytical gradient for high-degree-of-freedom problems. If the algorithm becomes trapped in a local minimum and returns FAIL-

```
GENERATE_RANDOM_VERTEX()
q ←RANDOM_CONFIGURATION();
i ← 0;  j ← 0;
while i < I and j < J and e(q) > ϵ do
    i++;  j++;
    q' ←RANDOM_NHBR(q);
    if e(q') < e(q) then
        j ← 0;  q ← q';
if e(q) ≤ ϵ then Return q
         else Return FAILURE
```

Figure 4: This algorithm iteratively attempts to reduce the kinematic error.

```
CONNECT_VERTICES(q, q')
1   i ← 0;  j ← 0;  k ← 0; L = {q};
2   while i < I and j < J and k < K and
                  ρ(LAST(L), q') > ρ0 do
3       i++;  j++;
4       q'' ←RANDOM_NHBR(LAST(L));
5       if e(q'') ≤ ϵ then
6           j ← 0;  k++;
7           if ρ(q'', q') < ρ(LAST(L), q') then
8               k ← 0;  L ← L + {q''};  q' = q'';
9   if ρ(LAST(L), q') ≤ ρ0 then Return L
10                          else Return FAILURE
```

Figure 5: This algorithm iteratively attempts to bring the system from one vertex to another while keeping the kinematic error within tolerances on $\mathcal{C}_{cons}$.

URE, then the sample is simply discarded. This has no serious effect on the overall approach, except that some computation time is wasted. Other approaches, such as the Levenberg-Marquardt nonlinear optimization procedure could be used instead of randomized descent, but one must be careful not to introduce an unwanted deterministic bias on the solutions.

3.2 Generating the Roadmap Edges

Once a set of vertices have been generated for the roadmap, a local planner is used to determine edges. Let $\rho(q, q')$ denote a metric on the configuration space. This could be defined, for example, as the sum of squares of the Euclidean displacements of all of the joints. Pairs of vertices for which ρ is below some threshold are chosen for attempted connection (this step is also performed in standard probabilistic roadmap construction). The algorithm in Figure 5 attempts to reduce ρ, the distance from q to q', by a randomized gradient descent that simultaneously maintains the kinematic error to within ϵ and reduces ρ. The randomized descent is free to travel due to tolerances on the closure constraints. The structure of CONNECT_VERTICES is similar to GENERATE_RANDOM_VERTEX. An additional constant K is used to terminate after K consecutive failures to reduce ρ, even though kinematic closure is maintained. Also, the constant ρ_0 is introduced to stop the algorithm when the path from q is sufficiently close to q'. In some cases, it might be preferable to switch the order of lines 5 adn 7. The success of the algorithm is based on the assumption that the selected vertices are close enough to ensure that local minima and collision constraints are not likely to prevent connection. Path smoothing can be performed on the resulting output L to improve the solution quality.

A crucial element in the performance of CONNECT_VERTICES is how RANDOM_NHBR will be selected. Although our current implementation chooses samples at random, it is preferable to generate samples that locally follow the tangent space of the constraints. The differential configuration vector dq lies in the tangent space of $f_i(q) = 0$ if

$$\frac{\partial f_i(q)}{\partial q_1} dq_1 + \frac{\partial f_i(q)}{\partial q_2} dq_2 + \cdots + \frac{\partial f_i(q)}{\partial q_n} dq_n = 0. \tag{1}$$

This leads to the following homogeneous system:

$$\begin{pmatrix} \frac{\partial f_1(q)}{\partial q_1} & \frac{\partial f_1(q)}{\partial q_2} & \cdots & \frac{\partial f_1(q)}{\partial q_n} \\ \frac{\partial f_2(q)}{\partial q_1} & \frac{\partial f_2(q)}{\partial q_2} & \cdots & \frac{\partial f_2(q)}{\partial q_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_m(q)}{\partial q_1} & \frac{\partial f_m(q)}{\partial q_2} & \cdots & \frac{\partial f_m(q)}{\partial q_n} \end{pmatrix} \begin{pmatrix} dq_1 \\ dq_2 \\ \vdots \\ dq_n \end{pmatrix} = \mathbf{0} \tag{2}$$

Recall that $m < n$. If the rank of the matrix is $k \leq m$, then $n - k$ configuration displacements can be chosen independently, and the remaining k parameters must satisfy (2). This under-constrained homogeneous system can be efficiently converted into reduced row echelon form using singular value decomposition (SVD). This enables RANDOM_NHBR to locally follow the tangent space and only generate $n - k$ random scalar displacements. The remaining displacements are given by simple linear equations. On average this technique should allow RANDOM_NHBR to remain within tolerances for larger step sizes, thus improving the efficiency of CONNECT_VERTICES.

3.3 Path Planning in $\mathcal{C}_{cons}$

A path planning query is given by q_{init} and q_{goal}. We can simply pretend that these two are new vertices in the roadmap, and use the technique from Section 3.2 to connect each of them to nearby vertices already in the roadmap. Graph search techniques can then be used to search the roadmap for a path that connects the two vertices. A path will not be found if the vertices lie in separate connected components. This could either mean that there is no solution, or that more vertices might be needed in the graph. The expansion technique discussed in [17] could be employed in this case to help increase the connectivity of the roadmap.

4 A Case Study: A 2D Line-Segment Structure

Although the approach described in this paper can be applied to a very general class of problems, we have made several simplifications in order to implement and illustrate the basics of our approach: 1) $\mathcal{L}$ is a collection of line segments in a 2D world; 2) joints only attach links at their endpoints; 3) every joint is revolute; 4) there are no joint limits (i.e., they have full freedom to rotate from 0 to 2π); 5) one of the joints attaches a link to the origin $(0, 0)$ in the world, $\mathcal{W}$; 6) the obstacle region is polygonal. Since each joint must attach two links, it will be convenient to consider a zero-length artificial link, L_0, that is permanently fixed at the origin. Below we describe the model components for a 2D line-segment structure and the implementation of our algorithm.

Kinematics Each configuration parameter, q_i can vary from 0 to 2π; thus, the configuration space $\mathcal{C}$ is $[S^1]^{n_l}$. The structure is in self collision if the interior of any line segments intersect. The free configuration space $\mathcal{C}_{free}$ contains the set of all configurations that avoid self collision and collision with static obstacles.

To determine the position and orientation of each L_i with respect to L_{i-1} in a chain of links $L_1, L_2, \cdots, L_i$, we will use a "homogeneous" transform matrix T_i that encodes both translation and rotation in $\mathcal{W}$:

$$T_i = \begin{pmatrix} \cos q_i & -\sin q_i & \ell_{i-1} \\ \sin q_i & \cos q_i & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}, \qquad (3)$$

in which ℓ_i is the length of L_i and $\ell_0 = 0$ (the length of the artificial link L_0). The transformation of any link is given by the product, $\forall (x, y) \in L_i$,

$$\begin{pmatrix} x(q) \\ y(q) \\ 1 \end{pmatrix} = T_1 T_2 \cdots T_i \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \qquad (4)$$

Closure Constraints To handle the closure constraints, we define a new structure, $\mathcal{M}' = (\mathcal{L}', \mathcal{J}')$, which is obtained by breaking cycles in the underlying graph G_M of $\mathcal{M}$. Let the set of links be the same, $\mathcal{L}' = \mathcal{L}$. Let $\mathcal{J}'$ be a superset of $\mathcal{J}$ and contain $n_j + m$ joints. A new joint is added for each of the m cycles in G_M. For each cycle in G_M, a corresponding joint can be selected arbitrarily. Denote this link by J_k. There will be two links from the cycle that are attached to J_k. For one of the links, disconnect it from J_k and form a new joint J'_k. If this insertion of joints is performed for each cycle, the result will be a structure $\mathcal{M}'$ which has no cycles ($G_{M'}$ is a tree). In $\mathcal{M}'$, the configuration of any link can be determined by applying (4) to the sequence of links on the unique path to L_0.

Neglecting self-collision, note that M' can achieve any configuration in $\mathcal{C}$. If J_k and J'_k have the same position in $\mathcal{W}$, then a closure constraint from $\mathcal{M}$ is satisfied. If this is true for all joints in $\mathcal{J}' \setminus \mathcal{J}$, then the configuration lies in $\mathcal{C}_{cons}$. The closure constraint f_i can be written by subtracting the expression for $J_k(q)$ using (4) from the expression for J'_k using (4). Let B be the indices of the set of joints that are broken in $\mathcal{M}$ to form $\mathcal{M}'$. The kinematic error function from Section 3.1 can be defined as

$$e(q) = \sum_{k \in B} \|J_k(q) - J'_k(q)\|^2. \qquad (5)$$

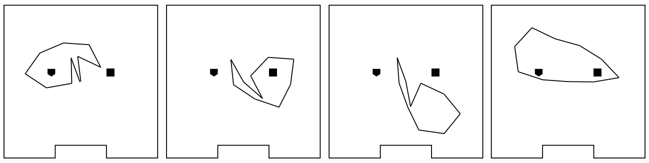


Figure 6: Four samples that lie in distinct connected components of $\mathcal{C}_{sat}$ for a 10-dof structure.

Distance Metric The following distance metric can be used to both determine whether to try connecting vertices, and to drive the actual search to connect them:

$$\rho(q, q') = \sum_{i=1}^{n_j} \|q_i - q'_i\| \qquad (6)$$

Experimental Results The planner was implemented in C++ and LEDA (Library of Efficient Datatypes and Algorithms) on a PC-based Linux workstation. The initial results appear promising; however, significant performance improvements can be made. The current results took several hours to generate, and we hope in the future to improve the edge connection speed by iteratively refining the allowed tolerances on the kinematic closure error. In the current implementation, the allowable error during connection is 0.04 for links of length 10. Figure 6 shows four samples that were obtained from by GENERATE_RANDOM_VERTEX. Note that each sample represents a distinct connected component of $\mathcal{C}_{sat}$. A roadmap was generated that contains about 3000 vertices and about 12000 edges. Figure 7 shows several computed solutions for a variety of examples.

5 Conclusions

We presented an extension of the probabilistic roadmap path planning approach to the case of systems that have closed kinematic chains. Closure constraints are common in many applications such as robotics, computational chemistry, virtual prototyping, and computer graphics. The difficulty is that path planning must be performed in a complicated subset, $\mathcal{C}_{sat}$, of the configuration space.

Our current experiments represent a preliminary evaluation of the approach. We are currently investigating potential performance improvements that can be made to the algorithms in Figures 4 and 5, and the application of these algorithms to more complicated structures in 2D

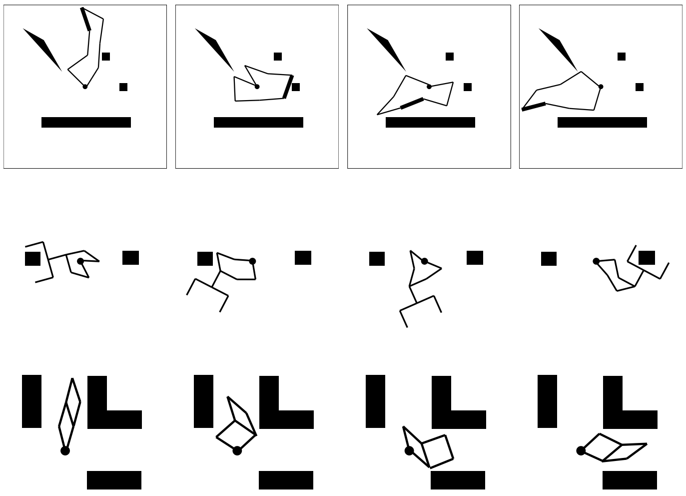


Figure 7: Snapshots along computed paths for three examples. The first is an 8-link loop that includes a manipulated bar; the second is a 6-link loop with an attached "end effector"; the third consists of two four-link loops. The disk in each frame indicates where the structure is attached to the world.

or 3D environments. The improvement of the vertex generation and edge connection procedures is a largely open issue.

Acknowledgments

We are very grateful to Jean-Claude Latombe who has contributed to many ideas explored in this paper. This research evolved from the work done by Paul Finn, Lydia Kavraki, Jean-Claude Latombe, and Steve LaValle, which was supported by Pfizer Limited and focused on developing conformational search techniques for drug molecules. Steve LaValle is partially supported by an NSF CAREER award. Lydia Kavraki is partially supported by NSF CAREER Award IRI-970228.

References

[1] R. Alami, T. Siméon, and J. P. Laumond. A geometrical approach to planning manipulation tasks. In *5th Int. Symp. Robot. Res.*, pages 113–119, 1989.

[2] N. M. Amato and Y. Wu. A randomized roadmap method for path and manipulation planning. In *IEEE Int. Conf. Robot. & Autom.*, pages 113–120, 1996.

[3] D. S. Arnon. Geometric reasoning with logic and algebra. *Artif. Intell.*, 37(1-3):37–60, 1988.

[4] J. Barraquand and J.-C. Latombe. Nonholonomic multibody mobile robots: Controllability and motion planning in the presence of obstacles. In *IEEE Int. Conf. Robot. & Autom.*, pages 2328–2335, 1991.

[5] J. Barraquand and J.-C. Latombe. Robot motion planning: A distributed representation approach. *Int. J. Robot. Res.*, 10(6):628–649, December 1991.

[6] S. Basu, R. Pollack, and M.-F. Roy. Computing roadmaps of semi-algebraic sets on a variety. Submitted for publication, 1998.

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

[8] D. Challou, D. Boley, , M. Gini, and V. Kumar. A parallel formulation of informed randomized search for robot motion planning problems. In *IEEE Int. Conf. Robot. & Autom.*, pages 709–714, 1995.

[9] P.C. Chen and Y.K. Hwang. Sandros: A motion planner with performance proportional to task difficulty. In *Proc. of IEEE Int. Conf. Robotics and Automation*, pages 2346–2353, Nice, France, 1992.

[10] G. Chirikjian, A. Pamecha, and I. Ebert-Uphoff. Evaluating efficiency of self-reconfiguration in a class of modular robots. *Journal of Robotic Systems*, 13(5):717–338, 1996.

[11] D. E. Clark, G. Jones, P. Willett P. W. Kenny, and Glen. Pharmacophoric pattern matching in files of three-dimensional chemical structures: Comparison of conformational searching algorithms for flexible searching. *J. Chem. Inf. Comput. Sci.*, 34:197–206, 1994.

[12] G. E. Collins. *Lecture Notes in Computer Science.* Springer-Verlag, Berlin, 1975.

[13] K. S. Fu, R. C. Gonzalez, and C. S. G. Lee. *Robotics: Control, Sensing, Vision, and Intelligence.* McGraw-Hill, New York, 1987.

[14] K. Gupta and X. Zhu. Practical motion planning for many degrees of freedom: A novel approach within sequential framework. *Journal of Robotics Systems*, 2(12):105–118, 1995.

[15] R. S. Hartenburg and J. Denavit. A kinematic notation for lower pair mechanisms based on matrices. *J. Applied Mechanics*, 77:215–221, 1955.

[16] L. E. Kavraki. *Random Networks in Configuration Space for Fast Path Planning.* PhD thesis, Stanford University, 1994.

[17] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. Robot. & Autom.*, 12(4):566–580, June 1996.

[18] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *Int. J. Robot. Res.*, 5(1):90–98, 1986.

[19] Y. Koga and J.-C. Latombe. On multi-arm manipulation planning. In *Proc. IEEE Int. Conf. on Rob. and Autom.*, pages 945–952, 1994.

[20] K. Kotay, D. Rus, M. Vora, and C. McGray. The self-reconfiguring robotic molecule: Design and control algorithms. In P.K. Agarwal, L. Kavraki, and M. Mason, editors, *Robotics: The Algorithmic Perspective.* AK Peters, 1998.

[21] J.-C. Latombe. *Robot Motion Planning.* Kluwer Academic Publishers, Boston, MA, 1991.

[22] B. Mishra. Computational real algebraic geometry. In J. E. Goodman and J. O'Rourke, editors, *Discrtete and Computational Geometry,* pages 537–556. CRC Press, New York, 1997.

[23] G. Taubin. Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. *IEEE Trans. Pattern Anal. Machine Intell.*, 13(11):1115–1137, November 1991.

[24] M. Yim. *Locomotion with a Unit-Modular Reconfigurable Robot.* PhD thesis, Stanford Univ., December 1994. Stanford Technical Report STAN-CS-94-1536.