A Kinematics-Based Probabilistic Roadmap Method for Closed Chain Systems

Li Han, *Texas A&M University, College Station, TX 77843-3112*
Nancy M. Amato, *Texas A&M University, College Station, TX 77843-3112*

In this paper we consider the motion planning problem for closed chain systems. We propose an extension of the PRM *methodology which uses the kinematics of the closed chain system to guide the generation and connection of closure configurations. In particular, we break the closed chains into a set of open subchains, apply standard* PRM *random sampling techniques and forward kinematics to one subset of the subchains, and then use inverse kinematics on the remaining subchains to enforce the closure constraints. This strategy preserves the PRM sampling philosophy, while addressing the fact that the probability that a random configuration will satisfy the closure constraints is zero, which has proven problematical in previous attempts to apply the PRM methodology to closed chain systems.*

Another distinguishing feature of our approach is that we adopt a two-stage strategy, both of which employ the PRM *framework. First, we disregard the environment, fix the position and orientation of one link (the "virtual" base) of the system, and construct a* kinematic roadmap *which contains different self-collision-free closure configurations. Next, we populate the environment with copies of the kinematic roadmap (nodes and edges), and then use rigid body planners to connect configurations of the same closure type. This two-stage approach enables us to amortize the cost of computing and connecting closure configurations.*

Our results in 3-dimensional workspaces show that good roadmaps for closed chains with many links can be constructed in a few seconds as opposed to the several hours required by the previous purely randomized approach.

1 Introduction

Closed chain mechanisms arise in many practical problems, such as the Stewart Platform [22], closed molecular chains [21], reconfigurable robots [14, 20], and the closed chain system formed by multiple robots grasping an object [13] (see Figure 1). Closed chains are sometimes called parallel chains since they can be viewed as consisting of two or more serial/open chains that provide parallel linkages between two points. While closed chains can offer advantages over open chains in terms of the rigidity of the mechanism, motion planning and control of closed chains is complicated by the need to maintain the closed chain structure, the so-called *closure constraint.*

Figure 1: *The Stanford Assistant Mobile Manipulator [12]. (Photo Courtesy of Prof. O. Khatib.)*

In this paper we consider the motion planning problem for closed chain systems. The motion planning problem is to find a collision free path that takes the closed chain from one configuration to another. A real world example is to find a collision free path for a multi-fingered robotic hand, to move a grasped part from one station to another for machining. For some tasks, the robot might need to regrasp the object, *i.e.*, to change the grasp points and grasp fingers, so as to accommodate the workspace limits of the robot or to avoid collisions. In general, it is not easy to move and regrasp the object simultaneously. One approach [16] proposed for this problem is to interleave *transit paths*, which only implement the regrasp without moving the object, and

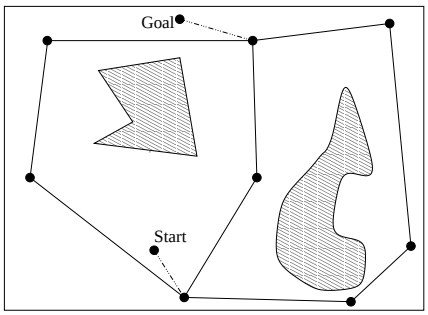


Figure 2: *A probabilistic roadmap (C-space).*

transform paths, which only transfer the object using fixed grasps. A manipulation system on a transform path with a fixed grasp can be viewed as a system with closed chains.[1] While the transform path planning problem has been studied for simple robot manipulation systems, the general problem remains open and is one motivation for our work. Other motivational applications of closed chain motion planning include animation, virtual reality and training.

Motion planning [15] is a challenging problem which involves complicated physical constraints and high-dimensional configuration spaces. The fastest existing deterministic planner [6] takes time exponential in the number of degrees of freedom of the robot. On the other hand, a class of randomized planners proposed during the last decade have successfully solved many previously unsolved problems. In particular, *Probabilistic Roadmap Methods (*PRM*s)* [2, 3, 4, 5, 10, 11, 24] have been used successfully in high-dimensional configuration spaces. The general methodology of PRMs is to construct a graph (the roadmap) during preprocessing that represents the connectivity of the robot's free configuration space (C-space), and then to query the roadmap to find a path for a given motion planning task (see Figure 2). Roadmap vertices are (generated from) randomly sampled configurations which satisfy feasibility requirements (*e.g.*, collision free), and roadmap edges correspond to connections between 'nearby' vertices found with simple local planning methods. For the most part, the major successes for PRMs have been limited to rigid bodies or articulated objects without closed chains. Recently, some efforts have been made to apply the PRM paradigm to closed chain systems [18] and to flexible objects [9].

The PRM planner for closed chain mechanisms proposed in [18] builds a roadmap in the portion of the configuration space that satisfies the closure constraints. The roadmap vertices are generated by first sampling points from the entire configuration space, and then performing a randomized gradient descent to try to transform them into configurations satisfying the closure constraints. Roadmap vertices are connected by a randomized gradient descent traversal of the constraint surface. When applied to closed chain linkages composed of line segments in the plane, this approach required several hours of computation to generate a well connected roadmap. Thus, while this work represents a crucial first step towards extending the PRM methodology to this important class of problems, the methods do not yet lead to efficient solutions for many practical problems.

1.1 Our Approach

In this paper, we propose a new approach for planning the motion of kinematic chains with closure constraints. Like [18], we believe the PRM methodology can be extended to closed chains. However, since the probability that a randomly generated node lies on the constraint surface is zero [18], we believe that the purely randomized philosophy of the PRM must be augmented with more deliberate techniques developed in the robotics community to deal with the closure constraints. In particular, we advocate the use of kinematics to guide the generation and connection of closure configurations. Briefly, the kinematics of an articulated object (such as a robotic finger) describes the relationship between the configuration/motion of the joints of the linkage and the resulting configuration/motion of the rigid bodies which form the linkage. Our planner uses both forward and inverse kinematics to generate closure configurations. In particular, we break the closed chain into a set of open subchains, apply standard PRM random sampling techniques and forward kinematics to one subset of the subchains, and then use inverse kinematics on the remaining subchains to enforce the closure constraints.

Another distinguishing feature of our approach is that we adopt a two-stage strategy, both of which employ the PRM framework. First, we disregard the en-

[1] Besides the collision-free and closure constraints, more manipulation constraints need to be taken into account for regrasp planning, which will be addressed in a follow up paper.

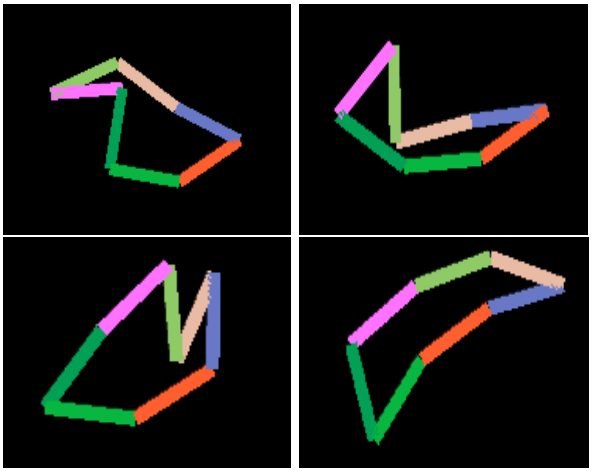


Figure 3: *Snapshots of a kinematic roadmap path for a 7-link closed chain.*

vironment, fix the position and orientation of one link of the chain (which can be viewed as a "virtual" base of the system)[2] , and construct a roadmap which contains different self-collision-free closure configurations. We call this roadmap a *kinematic roadmap* since it deals solely with the robot's kinematics and utilizes both forward and inverse kinematics in its construction. Figure 3 shows snapshots from a path contained in a kinematic roadmap found by our planner. Next, we populate the environment with copies of (portions of) the kinematic roadmap. This stage again employs the PRM strategy. In particular, we select random configurations (position and orientation) in the environment for the base in the kinematic roadmap, and retain those portions of the kinematic roadmap that are collision-free. Finally, local planning methods for rigid bodies are used to connect configurations of the same closure type. Note that this strategy restricts the connection of different closure configurations to the kinematic roadmap, *i.e.*, the only edges between different closure types in the main roadmap are copied from the kinematic roadmap.

Our motivation for this two-stage approach is that it amortizes the cost of computing and connecting the closed chain configurations. Another benefit of this strategy is that we do not waste time trying to connect unconnectable closure configurations when constructing the final roadmap. Indeed, our experimental results in 3-dimensional workspaces show that good roadmaps for closed chains with many links can be constructed in a few seconds as opposed to the several hours required by the previous purely randomized approach [18].

This paper is outlined as follows. In Section 2, we describe the related work in more detail. We formally define the closed chain motion planning problem in Section 3. The details of our approach are described in Sections 4 through 6. We present some experimental results in Section 7, and some concluding remarks in Section 8.

For simplicity and clarity, two-dimensional figures are used throughout this paper to illustrate the closed chain motion planning problem and our kinematics-based PRM motion planning approach. It should be noted that the general discussion and planning framework are applicable to both two-dimensional and three-dimensional workspaces.

2 Related Work

PRMs. As mentioned in Section 1, the success of PRMs for rigid body and articulated open chain robots [2, 3, 4, 5, 10, 11, 24], has motivated the extension of the PRM strategy to planning for elastic objects (FPRM) [9] and closed chains [18]. The major challenge here is in finding an effective way to deal with the additional constraints imposed on the feasible robot configurations. In particular, while the only constraint on feasible configurations for rigid bodies and open chains is that they be collision free (constraints on the joint variables of an open chain linkage can generally be encoded in the robot's configuration space), elastic objects can only achieve deformations with (local) minimum elastic energy and closed chains need to satisfy the closure constraints.

As previously mentioned, while [18] pioneered the use of PRMs on closed chain systems, the relative inefficiency of the randomized gradient descent technique used to generate closure configurations from randomly sampled configurations, and to connect closure configurations, emphasized the need for better techniques. The fundamental problem is that since the probability that a sampled node lies on a constraint surface is zero [18], it is very difficult to find (and connect) configurations satisfying the closure constraints using purely

[2]The virtual base can be chosen in other ways. For example, in a hand-object manipulation system, we can choose the object as the virtual base of the system.

randomized techniques. This is what motivates our kinematic roadmap, whose construction employs both forward and inverse kinematics. We note that although it was not used, the possibility of placing pre-computed closure configurations at different locations in the environment was mentioned in [18].

FPRM [9] deals with the energy requirement for elastic objects in a manner similar to the kinematic roadmap we use for closed chains: minimal energy deformations are computed *a priori*, disregarding the obstacles in the environment, and then copies of these deformations are placed at randomly selected locations (positions and orientations) in the environment. However, a difference between FPRM and our approach is that we also use the PRM strategy to construct the kinematic roadmap and populate kinematic roadmap edges, which correspond to connections between closure configurations. In contrast, FPRM populates minimal energy deformations only and performs node connection directly in the environment. We note that our two-stage PRM strategy can be applied to flexible object motion planning. More specifically, we can first generate a "deformation" roadmap with connections between minimal energy deformations in a clear environment and then populate it to the real environment containing obstacles.

Other Methods. A random exploration strategy, based on the *Ariadne's Clew Algorithm* [1], has been used to solve point-to-point inverse kinematics problems for redundant manipulators. Given an initial configuration of a robot, the problem was to find a reachable configuration that corresponds to a desired position and orientation of the robot end-effector and to find a feasible path connecting the initial and the goal configuration. Central to their approach was the construction of a roadmap[3] which took into account constraints due to joint limits, self-collision, and collision with environment obstacles. A point-to-point inverse kinematics problem was then solved by querying the roadmap.

In [25], the recently proposed *Rapidly-exploring Random Trees (*RRT*)* [17] strategy was used to greatly decrease the computation time required for several of the examples studied in [18].

[3]While the roadmap in [1] is called a *kinematic roadmap*, it is different from our kinematic roadmap which we construct without any knowledge of the obstacles in the environment.

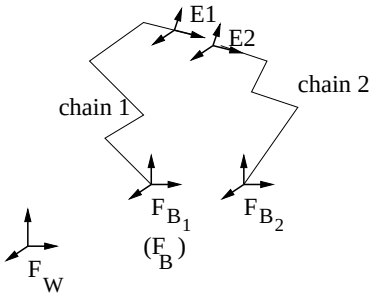


Figure 4: *Breaking a Closed Chain into 2 Open Chains*

3 Problem Formulation

In this section, we study the configuration spaces of multi-link chains and discuss the effect of closure constraints on these spaces. When a linkage system involves multiple closed chains, the overall closed chain constraint is satisfied if and only if each closed chain constraint is satisfied. Therefore, we discuss in detail only the case for one closed chain, with the understanding that the problem involving multiple closed chains can be similarly formulated and handled by our planner.

We first note that a closed chain system can alternatively be viewed as a linkage system consisting of a collection of open chains, where we 'break' each closed chain, and then satisfy the closure constraints, if any, by forcing the break points to coincide. For example, in Figure 4, chain 1 and chain 2 form a closed chain where the frames E_1 and E_2 attached to the breakpoint (the "end effector") must coincide to satisfy the closure constraints.

Consider a closed chain system that can be broken into k open chains. The configuration of an open chain i can be specified by its base configuration and its joint variables. In particular, the configuration of the base can be specified by the Euclidean (rigid body) transformation from the world frame F_W to the body frame F_{Bi}: $g_{wb_i} = (p_{wb_i}, R_{wb_i}) \in SE(d)$ where $d \in \{2, 3\}$ is the dimension of the workspace, $p_{wb_i} \in \mathrm{R}^d$ and $R_{wb_i} \in SO(d)$ are, respectively, the position and orientation of F_{Bi} relative to F_W, and $SE(d)$ denotes the special Euclidean group. Denote by $\beta_i = (\beta_{i1}, \ldots, \beta_{in_i})$, the vector of joint variables for chain i. For a revolute joint, the joint variable is an angle $\beta_{ij} \in [0, 2\pi)$, with the angle 2π equated to angle 0, which is naturally associated with a unit circle in

the plane, denoted by S^1, and hence we write $\beta_{ij} \in S^1$. A prismatic joint is described by a linear displacement $\beta_{ij} \in \mathrm{R}$ along a directed axis. In summary, the *configuration space* of a multi-link robot can be represented as

$$\mathcal{C} = \{(g_{wb_1}, \beta_1, \cdots, g_{wb_k}, \beta_k) | \\ g_{wb_i} \in SE(d), \beta_i \in S^{r_i} \times \mathrm{R}^{p_i}, i = 1, \cdots, k\}. \quad (1)$$

where r_i and p_i are the number of revolute joints and prismatic joints for link i, respectively.

One of the reasons PRMs work well for systems without closed chains is that any configuration q sampled from $\mathcal{C}$ is a *valid* configuration (when collision constraints are ignored). However, this is not true for systems involving closed chains where q must also satisfy the closure constraints. For example, the two end-effector frames in Figure 4 must coincide:

$$g_{we_1} = g_{wb_1} g_{b_1 e_1}(\beta_1) = g_{wb_2} g_{b_2 e_2}(\beta_2) = g_{we_2} \quad (2)$$

where $g_{b_i e_i}(\beta_i), i = 1, 2$, is the *forward kinematic transformation* [7, 19] of chain i which determines the end-frame configuration based on joint variables. (*Inverse kinematics* solves the inverse problem of determining proper joint variables to achieve some specified end-frame configuration.)

The closure constraint is often expressed in the form $f(q) = 0$ (*e.g.*, $g_{we_1} - g_{we_2} = 0$). When multiple closed chains exist in a linkage system, each closed chain imposes one closure constraint; the lth such constraint is denoted by f_l, and we use $f(q) = 0$ to denote $f_l(q) = 0$, for all $1 \leq l \leq K$, where K is the number of closed chain constraints. In general, the valid configurations of a closed chain system lie in the set:

$$\mathcal{C}_{closure} = \{q | q \in \mathcal{C} \text{ and } f(q) = 0\}. \quad (3)$$

Notice that closure constraints such as Equation 2 can be transformed to the zeros of polynomials using the projective transformation. Then, the valid configurations of the system, $\mathcal{C}_{closure}$, define a lower-dimensional *algebraic variety* embedded in the higher-dimensional configuration space $\mathcal{C}$. This is roughly analogous to embedding a 2-dimensional surface or a 1-dimensional curve in a three dimensional space. The fact that the volume measure of a low-dimensional entity in a high-dimensional ambient space is zero is why the probability that a random configuration $q \in \mathcal{C}$ will satisfy the closure constraint is zero. While closure constraints pose difficulties for standard PRM planners, as we will see, the structure of $\mathcal{C}_{closure}$ can be utilized to guide the generation and connection of closure configurations.

Finally, for both open and closed chain systems, feasible configurations should not involve collision between the robot and an obstacle, or self-collision among the links. We denote by $\mathcal{C}_{free}$ the set of robot configurations $q \in \mathcal{C}$ which do not cause any collision in the system.

Using the notation defined above, the closed-chain motion planning problem can be defined as follows:

Problem 1 *Given a start configuration q_0 and a goal configuration q_1, the objective of the planner is to find a path $q(t), t \in [0, 1]$, such that $q(0) = q_0$, $q(1) = q_1$, and $\forall t \in [0, 1], q(t) \in \mathcal{C}_{closure} \cap \mathcal{C}_{free}$.*

4 A Kinematics-Based PRM

In this section, we describe the high-level strategy of our kinematics-based PRM for closed chain systems.

We begin by noting that the closure constraint in Equation 2 can be reduced to:

$$g_{wb_1}(g_{b_1 e_1}(\beta_1) - g_{b_1 b_2} g_{b_2 e_2}(\beta_2)) = 0 \quad (4)$$

$$g_{b_1 e_1}(\beta_1) - g_{b_1 b_2} g_{b_2 e_2}(\beta_2) = 0 \quad (5)$$

where $g_{b_i e_i}, i = 1, 2$, are the end-frame configurations described in the body frame F_{Bi}, and $g_{b_1 b_2}$ is the transformation from the base link of chain 1 to the base link of chain 2. If we think of the base link of the first chain as the virtual (mobile) base of the system, its configuration g_{wb_1} can be interpreted as a rigid body motion on the system. In other words, Equation 4 reveals one important property of closure configurations: *rigid body transformations preserve closure configurations.* Equation 5 further shows that closure constraints can be defined independent of the base configuration.

In the following discussion, we will use g_{wb} to denote the configuration of the virtual base of the system. In addition, we will treat the transformation from the system base to the base of any chain, say chain i, as a virtual link with joint variables being the parameterization of the transformation g_{bb_i}, *e.g.*, a position vector

(prismatic joints) $p_{bb_i} \in \mathrm{R}^d$ and an orientation vector (revolute joints) $\alpha_{bb_i} \in S^{\frac{d(d-1)}{2}}$. Thus, every open chain can be viewed to be virtually extended to the system base. As a result, we can define a joint variable for the extended chain as $\theta_i = (p_{bb_i}, \alpha_{bb_i}, \beta_i), i = 1, \cdots, k$. For simplicity, we will call θ_i the joint variable of subchain i, with the understanding that it includes the virtual joint variables, when applicable. Define the joint variable of the system to be $\theta = (\theta_1, \cdots, \theta_k) \in S^r \times \mathrm{R}^p$, where p and r are the total number of prismatic joints and revolute joints, respectively. Then the system configuration space can be rewritten as

$$\mathcal{C} = \{(g_{wb}, \theta) | g_{wb} \in SE(d), \theta \in S^r \times \mathrm{R}^p\}. \quad (6)$$

Since the closure constraint (Equation 2) or its general form $f(q) = 0$, in fact, does not depend on the base configuration and only specifies constraints with respect to joint variables, we can define the equivalent constraints of f on the joint variables as

$$\tilde{f}(\theta) = 0 \quad (7)$$

Furthermore, define

$$\tilde{\mathcal{C}}_{closure} = \left\{\theta | \theta \in S^r \times \mathrm{R}^p, \tilde{f}(\theta) = 0\right\}. \quad (8)$$

Now, the subset of $\mathcal{C}$ corresponding to closure configurations can be expressed as:

$$\mathcal{C}_{closure} = \left\{(g_{wb}, \theta) | g_{wb} \in SE(d), \theta \in \tilde{\mathcal{C}}_{closure}\right\} \quad (9)$$

The following two observations summarize the above discussion, and form the basis of our two-stage PRM closed chain planner:

Observation 1 *Only the joint variables θ determine if a configuration $q = (g_{wb}, \theta)$ is closure or not. The closure constraint defines an algebraic variety (Equation 8) on $\theta \in S^r \times \mathrm{R}^p$, which can be parameterized almost everywhere. In other words, θ can be partitioned into θ_a and θ_p, $\theta = \theta_a \times \theta_p$, where θ_p can be determined from θ_a based on the closure constraints (Equation 7).*

Observation 2 *A given "closure configuration" θ, can be combined with different base configurations $g_{wb} \in SE(d)$, which corresponds to placing the same closure configuration at different locations in the environment.*

The first observation suggests an efficient way to generate closure configurations: ignore the environment (obstacles) and set a nominal base configuration, randomly generate θ_a and determine the corresponding θ_p by solving the closure constraints (Equation 7), and retain the self-collision free closure configurations. For example, consider the closed chain shown in Figure 4. Suppose we select the joint variables of chain 2, θ_2, as the *active variables* θ_a, and randomly generate values for them. We then use *forward kinematics* to determine the end-frame configuration g_{be_2}, and then use *inverse kinematics* to compute joint variables of chain 1 (the *passive variables* θ_p) which will make g_{be_1} coincide with g_{be_2} and satisfy the closure constraints.

The closure configurations generated will be the vertices of a (small) roadmap which records paths connecting self-collision-free closure configurations (again, with no dependence on the base configuration g_{wb}). The edges in this roadmap can be generated using straight-line, or any other simple local planner to connect the active variables θ_a of the two closure configurations, and then computing the corresponding passive variables θ_p along the local path. As with any PRM, a self-collision-free local path is recorded as a roadmap edge. By a slight abuse of terminology, we call such a roadmap a *kinematic roadmap* since it reveals the kinematic connectivity of the closure structures, and its construction involves the computation of both forward and inverse kinematics. Figure 5(a) shows a three-node kinematic roadmap for a 4-link closed chain.

The second observation suggests that we place multiple copies of the same closure configuration in the environment. There are two advantages to this approach. First, we quickly populate the environment with closure configurations and amortize the cost of computing a closure configuration. Second, we can treat configurations with the same closure structure as rigid body configurations, which can then be connected by efficient rigid body PRM local planning methods.

While one could generate a roadmap simply from configurations with *one* closure structure, the query process would have to connect any start configuration and goal configuration to this closure structure, which might become very hard, or might not be possible at all. Hence, instead of copying only one closure configuration, we copy an entire kinematic roadmap to the environment and retain all vertices and edges that are collision free (see Figure 5(b)). Next, we group con-

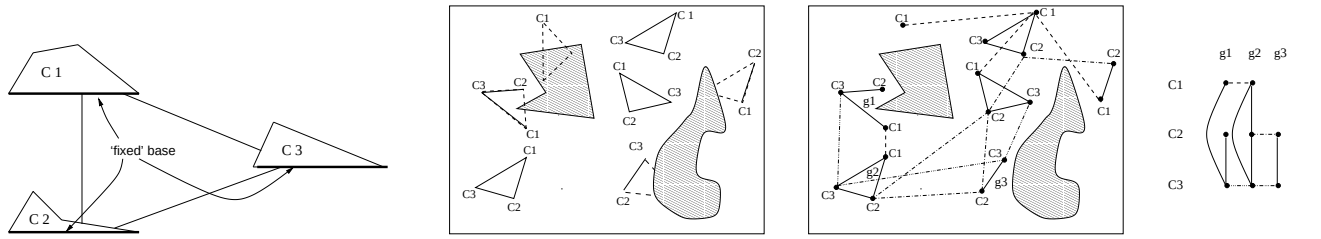


Figure 5: *(a) A kinematic roadmap (Workspace), (b) copying the kinematic roadmap to different base configurations (C-space), and (c) connecting configurations with the same closure structure (C-space).*

figurations by closure structure, and attempt to make (rigid body) connections within each group (see Figure 5(c)).

5 The Kinematic Roadmap

This section discusses a PRM planner to construct a kinematic roadmap. Recall that the kinematic roadmap encodes the connectivity of the closure configurations $\theta \in \tilde{\mathcal{C}}_{closure}$ and does not depend on the environment.

KINEMATIC ROADMAP CONSTRUCTION
1. NODE GENERATION
   (find self-collision-free closure configurations)
2. CONNECTION
   (connect nodes and save paths with edges)
   (repeat as desired)

5.1 Node Generation

The task here is to generate joint variables θ which satisfy the closure constraints and retain self-collision-free closure configurations as kinematic roadmap nodes. Recall that θ can be partitioned into active and passive variables, θ_a and θ_p, where the value of θ_p can be determined for a given θ_a value based on closure constraints.

From an algorithmic point of view, θ_a needs to be chosen such that for a given value of θ_a, the corresponding value of θ_p satisfying the closure constraints, if any, can be computed efficiently. While solutions are not known for inverse kinematics problems for general linkages, the closed-form inverse kinematic solutions for simple chains (such as 4-link chains) and most industrial robots do exist. Therefore, we are most

Figure 6: *Closure configurations for some kinematic chains generated by our planner.*

interested in choosing θ_a and θ_p as consecutive joint variables. More specifically, for a system involving K closed chains, each closed chain is broken into two open chains: one an "active chain" with joint variables θ_{la} and the other a "passive chain" with joint variables θ_{lp}, where $l = 1, \ldots, K$, is the index of the closed chain. We also need to ensure that the chain corresponding to the passive joint variables θ_p has a closed-form inverse kinematic solution. This can always be done, for example, by choosing any three consecutive joint variables in each closed chain as θ_p.

NODE GENERATION FOR KINEMATIC ROADMAP
1. Randomly generate θ_a

2. Use forward kinematics for active chains to compute end-frame configurations $g_{la}, l = 1, \ldots, K$, at the break point of each closed chain;
3. Use inverse kinematics for passive chains to compute joint variables θ_p to achieve the end-frame configurations computed in Step 2.
4. If a solution is found in Step 3 (closure exists)
5. If closure configuration $\theta = (\theta_a, \theta_p)$ is self-collision free
6. retain θ as a kinematic roadmap node

In Step 3, if multiple solutions exist for the inverse kinematics problem, we can either keep all solutions or randomly choose one. Figure 6 shows some closure configurations generated by our planner.

Finally, we note that when a link or joint is involved in multiple closed chains, *i.e.*, when the system involves common loops, the closed chain constraints need to be carefully handled to guarantee that different closed chains will result in the same link configuration or the same joint variable. In particular, we can use the algorithm above to close loops one by one, by choosing joint values to avoid breaking loops while creating other loops. The first loop can choose arbitrary values for its active joint variables θ_a^1. If the loop cannot be closed, *i.e.*, there does not exist θ_p^1 satisfying the closure constraints, then discard it. Otherwise, continue working on its neighboring loop, say loop 2. Assume the joints θ_c^{12} are common for loop 1 and loop 2. Then we will use the values of θ_c^{12} that have been computed from loop 1 as part of the active joint values θ_a^2 of loop 2 (to keep loop 1) and then compute the corresponding θ_p^2. In general, if a loop shares joint variables with other loops, it has to keep the values of the common joint variables that have been determined from the closure constraints of other loops. Clearly, when a loop has more determined joint values, it is more constrained, and thus it is more difficult to close the loop. One heuristic for node generation of common loops is to start from the loop with the largest number of common joints. For the example shown in Figure 7, it would be better to start from the center loop.

For the efficiency of the PRM node generation, it is important to choose θ_a and θ_p in a way that the inverse kinematics for the passive chains, corresponding to the θ_p joint variables, has a closed form solution. It is also important to maximize the probability that a closure configuration can be obtained given a randomly generated θ_a. For example, consider the closed

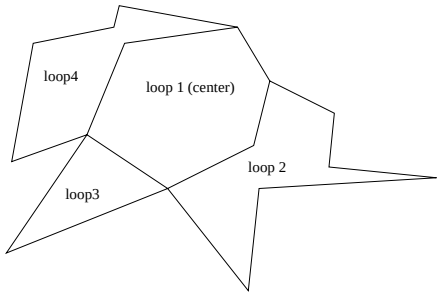


Figure 7: *A System with Common Loops*

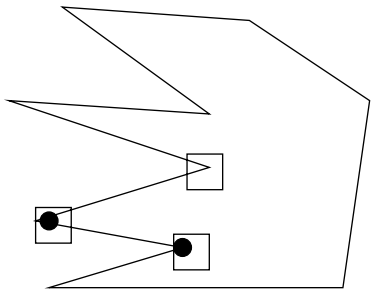


Figure 8: *Different joint partition schemes.*

chain shown in Figure 8. Two possible selections of θ_p are shown: (i) the two joints marked with black circles, or (ii) the three joints marked with squares. In this case, the three joint option would be preferable, since the intersection of the workspaces of the active and passive chains is larger. In the following, we make this argument more precise.

Recall that the workspace of an open chain with joint variables θ is defined as

$$W = \{g_{be} \in SE(d) | \exists \theta \in Q, s.t. g_{be} = g_{be}(\theta)\} \qquad (10)$$

where Q is the joint space, *i.e.*, the set of all possible joint variable values with joint limits taken into account, g_{be} is the end-frame configuration with respect to the base, and $g_{be}(\theta)$ is the forward kinematics of the open chain.

For one closed chain, we denote the end-frame configuration of the active chain by g_{ba}, and the workspaces of the active chain and passive chains by W_a and W_p, respectively.

Observation 3 *A randomly generated value of θ_a can result in a closure configuration, i.e., there exist passive*

joint variable values satisfying the closure constraint, if and only if *the end-frame configuration of the active chain, g_{ba}, falls in the workspace of the passive chain, i.e. $g_{ba} \in W_p$.*

Therefore, a rough estimate of the probability that a randomly generated θ_a results in a closure configurations is:

$$\text{prob(closure)} \approx \frac{Volume(W_p \cap W_a)}{Volume(W_a)} \qquad (11)$$

where $Volume$ denotes the volume of the workspace measured in $SE(d)$.

Remark 1 *If both W_p and W_a can be computed, then a random closure configuration $\theta = (\theta_a, \theta_p)$ can be obtained by randomly choosing a configuration in $W_p \cap W_a$ and using inverse kinematics of both the active chain and the passive chain to compute θ_a and θ_p. This is one of the most effective ways to generate a random closure configuration. However, it is not easy to compute the workspaces. We will still use θ_a and θ_p in the following discussion with the understanding that the knowledge of the workspaces, when available, should be exploited to improve the effectiveness of node generation.*

Since multiple joint variable values may result in the same end-frame configuration, the above probability measure is not accurate because it does not take the multiplicity into account. Denote by $g^{-1}(g_{ba})$ the inverse kinematic solutions for one end-frame configuration of the active joints. Then, the probability of obtaining a closure configuration from a randomly generated θ_a is:

$$\text{prob(closure)} = \frac{Volume(g^{-1}(W_p \cap W_a))}{Volume(g^{-1}(W_a))} \qquad (12)$$

where $g^{-1}(W)$ denotes set of inverse kinematic solutions for each configuration in W, and where the volume is computed in the active joint space instead of $SE(d)$ as in Equation 11.

As different joint partition schemes may result in different probabilities of successfully generating a closure configuration, the joint partition scheme should ideally be chosen based on this probability. However, the computation of the probability measure (Equation 12) involves the computation of the workspaces, inverse kinematics, and volume integrals, which is probably too complicated to be practical for most linked systems. Nevertheless, the probability measure provides us with insight that can be used to develop heuristics to guide the partition of the joint variables. For example, balancing the lengths of the active and passive chains is one possible heuristic.

Finally, we note that increasing the probability that a closure configuration exists will in many cases complicate the inverse kinematics for the passive chain. For example, the length heuristic mentioned above might result in longer passive chains with more complicated inverse kinematics than the 3-joint/4-link chains we have selected.

5.2 Node Connection

An edge between two closure configurations in the kinematic roadmap consists of a sequence of intermediate closure configurations. Since it is relatively expensive to generate closure configurations, the edges of the kinematic roadmap are saved for future use. Node connection in the kinematic roadmap follows the standard PRM framework.

NODE CONNECTION FOR KINEMATIC ROADMAP
1. For any two 'nearby' closure configurations θ_i and θ_j
2. Use (simple) local planner to find path from θ_{ia} to θ_{ja}: $\theta_a(t), t \in [0,1]$, where $\theta_a(0) = \theta_{ia}$, $\theta_a(1) = \theta_{ja}$
3. For each intermediate point on the path $\theta_a(t)$
4. If inverse kinematics determines that no $\theta_p(t)$ exists to satisfy the closure constraints
5. return no-edge
6. Choose the closure configuration $\theta(t)$ that is continuous from previous step
7. If $\theta(t)$ involves self-collision, then return no-edge
8. endfor
9. save the edge (and with it the path $\theta(t), t \in [0,1]$)
10. return edge-exist

The connection of common loop configurations has to be processed similarly as for the node generation of common loops. In general, any connection strategy and local planner for rigid body robots or serial chains, such as the nearest neighbors connection strategy and the C-space straight line local planner, can be used to choose pairs of closure configurations for potential edge generation and to connect the active joint variables. When the system involves complicated closed chain structures, more sophisticated techniques such as the Jacobian method or a point-to-point inverse kinematic solver [1] can also be used to generate kinematic roadmap edges. The distance metric on the set $\tilde{\mathcal{C}}_{closure}$

can be, *e.g.*, Euclidean distance (Equation 13) with the modification that the distance between two joint angles is at most π.

$$dist(\theta_1 - \theta_2) = \|\theta_1 - \theta_2\| \tag{13}$$

6 Building a Roadmap from a Kinematic Roadmap

The kinematic roadmap provides us with a set of self-collision-free closure configurations $\theta \in \tilde{\mathcal{C}}_{closure}$ (Equation 8) and connections between them. Therefore, by randomly generating base configurations g_{wb}, we can 'populate' the environment with kinematic roadmap nodes and edges – this will only require collision detection with environment obstacles since we save the paths associated with the kinematic roadmap edges. Furthermore, roadmap nodes generated from the same closure configuration can be treated as rigid body configurations during roadmap connection.

Prototype Kinematics-Based PRM
I. Populate Environment with Kinematic Roadmap
  generate random base configurations and retain collision-free parts of kinematic roadmap in roadmap
II. Additional Connection of Roadmap Nodes
  connect roadmap nodes with the same closure structure using rigid body planners

6.1 Populating the environment with copies of the Kinematic Roadmap

Populating Env. with copies of Kinematic Roadmap
1. Choose random vertex θ from the kinematic roadmap
2. Generate random base configuration g_{wb}
3. If the configuration (g_{wb}, θ) is collision-free
4. Retain (g_{wb}, θ) as a roadmap vertex
5. For each neighbor of θ, say $\bar{\theta}$, in the kinematic map (repeat with their neighbors as needed)
6. If $(g_{wb}, \bar{\theta})$ is collision-free
7. Retain $(g_{wb}, \bar{\theta})$ as a roadmap vertex
8. Retrieve the path $\theta(t)$ connecting θ and $\bar{\theta}$ from the kinematic map
9. If $(g_{wb}, \theta(t))$ is collision-free for all intermediate closure configurations along the path
10. Add an edge between (g_{wb}, θ) and $(g_{wb}, \bar{\theta})$

(repeat as desired)

The generation of the random base configuration in Step 2 can be implemented with any node generation strategy developed for rigid body robots such as PRM [11], PRM with Gaussian filter [5], OBPRM [2], and the medial-axis PRM [23]. We next note that since the kinematic roadmap's nodes and edges are known to be self-collision free, the only collision checks needed in this stage are between environment obstacles and the robot, *i.e.*, it is not necessary to check the links for self-collision. Therefore, the reuse of the closure configurations and their connection edges can significantly reduce the total number of collision detection calls, which represent the major computation cost at this stage.

6.2 Connecting same closure nodes

Closed-chain configurations with the same closure configurations can be viewed as configurations of a rigid body. Therefore, we can use rigid body PRM methods to connect them. More specifically,

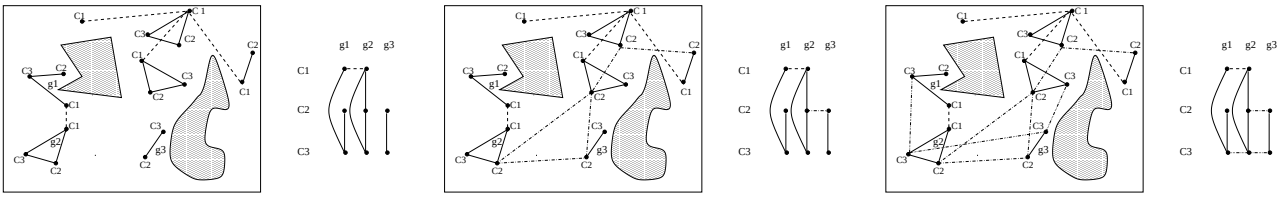
Connecting Nodes of Same Closure Type
1. For each closure configuration θ in kinematic roadmap
2. Collect all roadmap nodes with this closure configuration in a set
3. Use rigid body PRM connection methods to connect configurations in the set
4. Add the edges generated in Step 3 to the roadmap
5. endfor

Figure 9 shows a portion of the roadmap, being progressively built by connecting configurations with closure structures $C1, C2$ and $C3$, respectively.

7 Experimental Results

7.1 Implementation Details

Our prototype closed chain PRM planner was developed on top of the C++ OBPRM software package developed by the robotics group at Texas A&M University [2, 4]. This strategy was taken because the construction of the kinematic roadmap and the connection of the roadmap nodes with the same closure configuration are both basically simple PRM planners. It turned out to be fairly easy to incorporate the closed-chains into the PRM framework due to the object oriented design of the code. All experimental results reported in this section were performed on an SGI Octane and used the RAPID [8] package for 3D collision detection.



(a) (b) (c)

Figure 9: *Roadmap connection in three rigid body phases: connecting configurations of type (a) C1, (b) C2, and (c) C3.*

Table 1: *Kinematic roadmap construction times (seconds) and statistics for k-link closed chains, $k = 6, 7, 11$ and 15. In the table, planner and spatial chains are labeled respectively with P and S following their link numbers, and cfg and CC denote the number of nodes and the number of connected components respectively in the resulting kinematic roadmap. Note that the roadmap for a single planar closed chain should have two connected components.*

Kinematic Roadmap Construction				
Chain	Generation		Connection	
Links	sec	cfg	sec	CC
6(P)	0.84	203	24.66	2
7(P)	0.90	122	9.88	2
11(P)	1.24	16	0.48	4
15(P)	1.63	3	0.01	3
6(S)	0.97	28	0.88	4
7(S)	1.09	20	0.41	6
11(S)	1.76	1	0.00	1

7.2 Experiments

While our planner can handle complicated three-dimensional closed chains, the results presented here are for single loop closed chains in three-dimensional environments. In particular, the chains we consider consist of m identical links, all joints are revolute, and we partition the m joint angles into 3 consecutive passive angles and $m - 3$ active angles.

We first study the effectiveness of our method for generating the kinematic roadmap, which involves generating and connecting closure configurations in the absence of obstacles in the environment. We used the algorithm presented in Section 5.1 to check if it is feasible for a randomly generated active joint angle to achieve a closure configuration. As seen in Figures 3 and 6, this method was effective in generating and connecting closure configurations.

However, as predicted in Section 5.1, our success in generating closure configurations is closely tied to the "balance" of the partition of the joints into the active and passive sets. Table 1 shows the kinematic roadmap statistics for closed chains with various numbers of links (all cases contained three passive joint angles). The nodes were generated from 2000 trials using the algorithm sketched in Section 5.1. The edges were generated by trying to connect each node to its 20 nearest neighbors using a C-space straight line planner. The results shown in the table are consistent with the analysis of Section 5.1. Namely, for both planar and spatial types of chains, the planner generated more closure nodes for the chains with fewer links, which are also those for which the partition into the active and passive chains is most equal, and where their workspaces overlap the most. In addition, the results for planner chains are better than their spatial counterparts, again due to their larger overlapping workspaces. Clearly, the simple PRM kinematic roadmap planner does not work as well for the 15-link planar chain and 11-link spatial chain. We are working on incorporating more sophisticated planning methods, such as the Ariadne's Clew algorithm [1], into our system which should help in such situations.

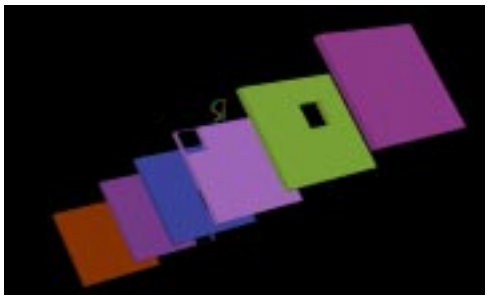


Figure 10: *The "Walls" Environment.*

We now analyze the benefit of the kinematic pre-

Table 2: *Roadmap construction times (seconds) and statistics for roadmaps constructed with and without the kinematic roadmap. In the table, planner and spatial chains are labeled respectively with P and S following their link numbers, and cfg and CC denote the number of nodes and connected components in the roadmap (there are two), respectively.*

Roadmap Construction							
Chain	Kinematic Map			Generation		Connection	
Links	sec	cfg	CC	sec	cfg	sec	CC
7(P)	1.25	31	2	2.44	341	12.56	2
7(P)	–	–	–	5.25	338	62.07	6
9(P)	0.44	13	3	0.99	132	9.27	3
9(P)	–	–	–	4.63	104	11.94	8
7(S)	0.31	7	6	0.09	43	3.93	7
7(S)	–	–	–	3.52	39	5.61	16

processing. That is, the benefit of using the kinematic roadmap as opposed to simply generating closure configurations directly in the environment (as was done in [18]). To study this issue, we compare roadmaps constructed with and without the kinematic preprocessing. We used the environment shown in Figure 10. Table 2 shows the statistics for 7 and 9-link chains. The roadmaps without kinematic preprocessing were generated using the PRM planner implemented in the OBPRM software package. The nodes were generated from 4000 attempts, using the generation method outlined in Section 5.1, the only difference being that now the base configuration was randomly generated and collision was checked with the obstacles in the environment as well. The edges were generated using the straight line planner to connect each node to its 20 nearest neighbors. The kinematic map nodes were generated from 400 attempts for planer chains and 500 attempts for the spatial 7-link chain, and, again, the 20 nearest neighbors were checked using the straight line planner. Then, five different base configurations were generated for each closure node when populating the environment with copies of the kinematic roadmap. The final rigid body connections between configurations with the same closure type also used the straight line planner with the 20 nearest neighbors. As can clearly be seen from the table, the roadmaps constructed using kinematic preprocessing are superior in all aspects: faster computation and improved roadmap quality (fewer connected components).

8 Conclusion

This paper presents a kinematics-based probabilistic roadmap planner for closed chains. The two-stage construction of our roadmap first builds a (small) kinematic roadmap that deals solely with the robot's kinematics and utilizes both forward and inverse kinematics in its construction. In the second stage, the environment is populated with copies of the kinematic roadmap, and rigid body connections are made between nodes with the same closure type. Both stages employ PRM planners to construct their roadmaps. Our preliminary experimental results indicate that the use of kinematics to guide the generation and connection of closure configurations is very beneficial, both reducing the computation costs and improving the connectivity of the resulting roadmap as compared to previous purely randomized approaches.

Therefore, we believe that augmenting the randomized philosophy of PRMs with more deliberate techniques developed in the robotics community is a promising direction to pursue for motion planning problems involving additional constraints.

While our results are promising, there are still many issues to be addressed. First, we intend to more fully exercise our current closed chain PRM planner on more complex linkages. We anticipate that the computational costs will grow, so that additional optimization will be required. One area we intend to explore in this regard is parallelization, which has been shown to be effective for traditional PRMs [3]. We also plan to study more complicated manipulation planning problems such as regrasp planning.

Acknowledgement

We would like to thank Drs. Steve Wilmarth, Jeff Trinkle, and Peter Stiller for valuable discussions. We are also grateful to the robotics group at Texas A&M, especially Lucia Dale and Guang Song, for their help regarding the OBPRM software package.

This research was supported in part by NSF CAREER Award CCR-9624315, NSF Grants IIS-9619850, EIA-9805823, and EIA-9810937, and by the Texas Higher Education Coordinating Board under grant ARP-036327-017.

References

[1] J. M. Ahuactzin and K. Gupta. The kinematic roadmap: A motion planning based global approach for inverse kinematics of redundant robots. *IEEE Trans. Robot. Automat.*, RA-15(4):653–670, 1999.

[2] N. M. Amato, O. B. Bayazit, L. K. Dale, C. V. Jones, and D. Vallejo. OBPRM: An obstacle-based PRM for 3D workspaces. In *Proc. Int. Workshop on Algorithmic Foundations of Robotics (WAFR)*, pages 155–168, 1998.

[3] N. M. Amato and L. K. Dale. Probabilistic roadmap methods are embarrassingly parallel. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 688–694, 1999.

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

[5] V. Boor, M. H. Overmars, and A. F. van der Stappen. The gaussian sampling strategy for probabilistic roadmap planners. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 1018–1023, 1999.

[6] J. F. Canny. On computability of fine motion plans. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 177–182, 1989.

[7] John J. Craig. *Introduction to Robotics: Mechanics and Control, 2nd Edition.* Addison-Wesley Publishing Company, Reading, MA, 1989.

[8] S. Gottschalk, M.C. Lin, and D. Manocha. Obb-tree: A hierarchical structure for rapid interference detection. Technical Report TR96-013, University of N. Carolina, Chapel Hill, CA, 1996.

[9] L. Kavraki, F. Lamiraux, and C. Holleman. Towards planning for elastic objects. In *Proc. Int. Workshop on Algorithmic Foundations of Robotics (WAFR)*, 1998.

[10] L. Kavraki and J. C. Latombe. Randomized preprocessing of configuration space for fast path planning. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 2138–2145, 1994.

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

[12] O. Khatib. Stanford Robotic Manipulation Group, robotics.Stanford.EDU/groups/manips/projects.

[13] O. Khatib, K. Yokoi, K. Chang, D. Ruspini, R. Holmberg, and A. Casal. Vehicle/arm coordination and multiple mobile manipulator decentralized cooperation. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 546–553, 1996.

[14] K. Kotay, D. Rus, M. Vona, and C. McGray. The self-reconfiguring robotic molecule: Design and control algorithms. In *Proc. Int. Workshop on Algorithmic Foundations of Robotics (WAFR)*, pages 375–386, 1998.

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

[16] J.-P. Laumond and M.H. Overmars. Algorithms for robotic motion and manipulation. In *Proc. Int. Workshop on Algorithmic Foundations of Robotics (WAFR)*, 1996.

[17] S. M. LaValle and J. J. Kuffner. Randomized kinodynamic planning. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 473–479, 1999.

[18] S.M. LaValle, J.H. Yakey, and L.E. Kavraki. A probabilistic roadmap approach for systems with closed kinematic chains. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 1999.

[19] Richard M. Murray, Zexiang Li, and S. Shankar Sastry. *A Mathematical Introducation to Robotic Manipulation.* CRC Press, Boca Raton, FL, 1994.

[20] A. Nguyen, L. J. Guibas, and M. Yim. Controlled module density helps reconfiguration planning. In *Proc. Int. Workshop on Algorithmic Foundations of Robotics (WAFR)*, 2000.

[21] A.P. Singh, J.C. Latombe, and D.L. Brutlag. A motion planning approach to flexiable ligand binding. In *7th Int. Conf. on Intelligent Systems for Molecular Biology (ISMB)*, pages 252–261, 1999.

[22] D. Stewart. A platform with six degrees of freedom. *Proc. of the Institute of Mechanical Engineering*, 180(part I(5)):171–186, 1954.

[23] S. A. Wilmarth, N. M. Amato, and P. F. Stiller. MAPRM: A probabilistic roadmap planner with sampling on the medial axis of the free space. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 1024–1031, 1999.

[24] S. A. Wilmarth, N. M. Amato, and P. F. Stiller. Motion planning for a rigid body using random networks on the medial axis of the free space. In *Proc. ACM Symp. on Computational Geometry (SoCG)*, pages 173–180, 1999.

[25] Jeffery H. Yakey. Randomized path planning for linkages with closed kinematic chains. Master's thesis, Iowa State Univ, Computer Science Dept., 1999.