Motion Planning for a Class of Planar Closed-Chain Manipulators

Guanfeng Liu, N. Shvalb†, Moshe Shoham†, J.C. Trinkle‡
Dept of CS, Stanford University, liugf@cs.stanford.edu
† Dept of ME, Technion-Israel Institute of Technology, Israel, shvalbn@techunix.technion.ac.il
‡ Dept of CS, Rensselaer Polytechnic Institute, trink@cs.rpi.edu

***Abstract*— We study the motion planning problem for planar *star-shaped* manipulators. These manipulators are formed by joining k "legs" to a common point (like the thorax of an insect) and then fixing the "feet" to the ground. The result is a planar parallel manipulator with $k-1$ independent closed loops. A topological analysis is used to understand the global structure the configuration space so that planning problem can be solved exactly. The worst-case complexity of our algorithm is $O(k^3N^3)$, where N is the maximum number of links in a leg. Examples illustrating our method are given.**

I. Introduction

The canonical robot motion planning problem is known as the "piano movers'" problem. In this problem, one is given initial and goal configurations of a "piano" (a rigid body that is free to move in an environment with fixed rigid obstacles) and geometric models of the piano and obstacles. The goal is to find a continuous motion of the piano connecting the initial and goal configurations. Lozano-Perez studied this problem in configuration space, or C-space, a space in which a configuration of the piano maps to a point, a motion maps to a continuous curve, and the obstacles map to the C-obstacle, *i.e.,* the set corresponding to overlap between the piano and an obstacle [3]. The dimension of C-space is equal to the number of degrees of freedom of the system. The free space, or C-free, is what remains after removing the C-obstacle from C-space. In C-space, the motion planning problem becomes a path planning problem. That is, one must construct a continuous path connecting the initial and goal configurations that lies entirely within C-free. Theoretical results for the piano movers' problem were first obtained by Schwartz, Sharir, and Hopcroft [27], [19]. They found that the problem is PSPACE hard, and proposed an algorithm based on Collins' decomposition to find a path. Since the worst-case running time of Collins' decomposition algorithm is doubly exponential in the dimension of C-space, it is impractical.

The more complex generalized movers' problem, is the problem in which there are multiple rigid bodies moving simultaneously in a workspace. The bodies are the links of one or more robots, and thus may be required to obey constraints corresponding to their kinematic structures and joint limits. Given the importance of motion planning problem in robotics, researchers worked to find more efficient algorithms despite the depressing complexity results found earlier. The most efficient exact method known is Canny's algorithm, which has time complexity that is only singly exponential in the dimension of C-space [40]. He also made the important observation that this bound is worst-case optimal, since the worst-case number of components in C-space is exponential in its dimension. Canny's algorithm is very difficult to implement - to date no full implementation exists.

In the 1990's, the intractability of exact motion planning for general problems stimulated a paradigm shift to randomized methods. The method of Barraquad and Latome combined potential field methods with random walk [13]. In essence, a potential field method defines an artificial potential field on C-space such that the goal configuration is the global minimum of the potential function and no saddle points or other local minima exist. When the function has this property, motion planning can be done by any gradient following algorithm. An important class of such functions are navigation functions [2], [6], [8]. Ideally, the potential function will be a function of the goal configuration, and the global minimum property will hold for all possible goal configurations. Since such potential functions can be difficult to design, Barraquad and Latome suggested the use of random walks to escape local minima [13]. This modification yielded a method that is practically effective and probabilistically complete.

When possibly many motion planning queries must be handled for a single static environment, a different type of randomized method has been found to be more efficient than rerunning the Barraquand-Latombe algorithm for each query. The probabilistic roadmap method (PRM) of Kavraki *et. al* [42], is an easy-to-implement randomized version of Canny's [40]. In this method, a graph is built that approximates the global structure of C-free. One chooses points at random in C-space and tests them for collision. Those that are not in collision are retained as nodes in the roadmap graph. Pairs of nodes are then tested for connectivity by using a fast "local" planner. If a pair is found to be connected, then an arc (containing the connecting path) is placed in the graph between the corresponding nodes. The roadmap becomes useful for motion planning after it attains the following two attributes: (1) a one-to-one correspondence between the graph's connected components and those of C-free; (2) for each point in C-free, it is easy to construct a path to the roadmap. Once the roadmap is completed, motion planning is essentially reduced to graph searching.

Because PRMs have been successful in solving problems in C-spaces with dimension approaching 100, many researchers have worked to make the method more efficient (*e.g.,* [29],

[30], [31]) and to modify it to solve more challenging types of problems, such as those with closed kinematic loops, nonholonomic constraints, dynamics, and intermittent contact (*e.g.*, [48], [28], [1], [32], [36], [21]). There have also been efforts to develop randomized methods for use in planning methods based on a cell decomposition of C-space [22]. Recent research revisited the theoretical basis that could explain the success of PRMs [33], [34], [35]. These works discussed the failure probability of finding a path between two given configurations in terms of several parameters like the path clearance, the number of sample points, and so on. In [35], probability measure theory was adopted to develop a more strict and complete theoretical basis for PRMs. If one reviews the set of problems for which PRMs have worked well, it appears that it is important that C-space be "expansive" [7] and possess a representation from which it is easy to obtain well-distributed points in C-space.

In this paper, we are particularly interested in planar *star-shaped manipulators*. These manipulators are formed by joining k planar "legs" to a common point (like the thorax of an insect) and then fixing the "feet" to the ground. The result is a planar parallel manipulator with $k-1$ independent closed loops. They are important because they arise in parallel manipulators, walking robots, and dexterous manipulation, and motion plans are difficult to obtain using PRMs. In such systems, C-space is often most naturally viewed as a lower-dimensional space embedded in an ambient space (typically the joint space). The embedding results from equality constraints corresponding to kinematic loop closure. In such settings, it is difficult to obtain an explicit description of C-space with minimal number of parameters and a suitable metric to guide sample generation. These problems make it difficult to construct a roadmap with the requisite properties, and hence difficult to solve motion planning problems for systems with kinematic loops using PRMs. The RLG (random loop generator) method [4], [5] improves the sampling techniques through estimating the regions of sampling parameters. However, its efficiency relies on the accuracy of the estimation, which often varies case by case. Moreover, it ignores the global structure of C-space, and may fail to sample globally important regions.

The difficulties associated with applying randomized motion planning methods to manipulators with closed chains and the availability of new results in topology [12], [41], [44], [10] have recently led to renewed interest in exact planning algorithms. Trinkle and Milgram derived some topological properties of the C-spaces (the number of components and the structures of the components) of single-loop closed chains with spherical joints in a workspace *without* obstacles [45], [44]. These properties drove the design of a complete, polynomial-time motion planning algorithm that works roughly as follows.

1) Choose a subset $\mathcal{A}$ of the links that can be positioned arbitrarily, and yet the remaining links can close the loop;
2) Move the links in $\mathcal{A}$ to their goal orientations along an arbitrary path while maintaining loop closure;
3) Permanently fix the orientations of the links in $\mathcal{A}$;
4) Repeat until all link orientations are fixed.

The main result that guided the algorithm's design is Theorem 2 in [45]. In generic cases, the C-space is the union of manifolds that are products of spheres and intervals. The joint coordinates corresponding to the spheres are those that can contribute to the subset $\mathcal{A}$ mentioned above and the structure of the C-space suggests a local parametrization for each step.

Here, the previous methods for C-space connectivity analysis are extended to planar star-shaped manipulators with revolute joints. These manipulators have a common junction point and k ($k > 0$) legs connecting the junction to the fixed base. Following a topological analysis of the global structure of C-space, the motion planning problem is solved completely in polynomial time. In Section III, kinematics and singularities of the manipulator are analyzed. In Section IV, necessary and sufficient conditions for C-space connectivity and path existence are derived, based on which a complete polynomial-time algorithm is developed in Section V. Section VI addresses path optimization and robustness issues. Section VII shows simulation results that tests the effectiveness of our algorithm. Finally VIII ends this paper with a brief conclusion.

II. Notation

Manipulator Notation	
M	- Manipulator
A	- Root junction or thorax of M
o_i	- Grounding point of foot i of M
M_j	- Leg j of M with foot fixed at o_j and other end free, $j = 1, ..., k$
n_j	- Number of links in M_j
$l_{j,i}$	- Length of link i of M_j; $i = 1, ..., n_j$
$\theta_{j,i}$	- Angle of link i relative to link $i-1$
$\tilde{M}_j(p)$	- Leg j of M with foot fixed at o_j and other end fixed at p
$\tilde{M}(p)$	- Manipulator with A fixed at p
L_j	- Sum of lengths of links of $\tilde{M}_j$
$L_{j,0}$	- Sum of lengths of links of M_j
$\mathcal{L}_j(p)$	- A set of long links of $\tilde{M}_j(p)$
$\lvert\mathcal{L}^*_j(p)\rvert$	- Number of long links of $\tilde{M}_j(p)$
Workspace Notation	
W_A	- Workspace of A
${}^dU_i$	- Cell of dimension d of W_A
p	- Point in the plane of M
$\gamma = p(t)$	- Curve in the plane of M
f	- Kinematic map of A
f_j	- Kinematic map of endpoint of M_j
Σ	- Critical set of f in W_A
Σ_j	- Critical set of f_j
Configuration Space (C-space) Notation	
$\mathcal{C}$	- C-space of M
$\tilde{\mathcal{C}}(p)$	- C-space of $\tilde{M}(p)$
$\mathcal{C}_j$	- C-space of M_j
$\tilde{\mathcal{C}}_j(p)$	- C-space of $\tilde{M}_j(p)$
c	- Point in C-space

III. Preliminaries

A star-shaped manipulator is composed of k serial chains with all revolute joints (see Fig. 1). Leg M_j is composed of

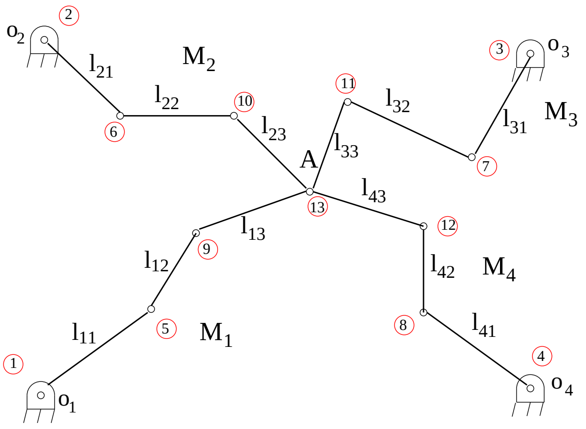


Fig. 1. Star-shaped manipulator with $k = 4$.

n_j links of lengths $l_{j,i}, i = 1, ..., n_j$ and joint angles $\theta_{j,i}, i = 1, ..., n_j$. At one end (the foot), M_j is connected to ground by a revolute joint fixed at the point o_j. At the other end, it is connected by another revolute joint to a junction point denoted by A. Note that when k is one, a star-shaped manipulator is an open serial chain. When k is two, it is a single-loop closed chain.

Assuming that the foot of M_j is fixed at o_j, let $f_j(\Theta_j) = p$ denote the kinematic map of M_j, where $\Theta_j = (\theta_{j,1}, \cdots, \theta_{j,n_j})$ is the tuple of joint angles, and p is the location of the endpoint of the leg (the thorax end). When M_j is detached from the junction A, the image of its joint space is the reachable set of positions of the free end of the leg, called the workspace W_j. In the absence of joint limits, the workspace W_j is an annulus if and only if there exists one link with length strictly greater than the sum of all the other link lengths. Otherwise it is a disk. Clearly, the workspace W_A of A when all the legs are connected to A is given by:

$$W_A = \bigcap_{j=1}^{k} W_j. \qquad (1)$$

In our study of $\mathcal{C}$, it will be convenient to refer to several other C-spaces. The C-space of leg M_j when detached from the rest of the manipulator will be denoted by $\mathcal{C}_j$. When the endpoint is fixed at the point p, leg j will be denoted by $\tilde{M}_j(p)$, where the tilde is used to emphasize the fact that the endpoint has been fixed. Note that $\tilde{M}_j(p)$ is a single-loop planar closed chain, about which much is known (see [45]), including global structural properties of its C-space, denoted by $\tilde{\mathcal{C}}_j(p) = f_j^{-1}(p)$.

When the junction A of a star-shaped manipulator is fixed at point p, its C-space will be denoted by $\tilde{\mathcal{C}}(p)$. Since collisions are ignored, the motions of the legs are independent, and therefore the C-space of the manipulator (with fixed junction) is the product of the C-spaces of the legs with all endpoints fixed at p:

$$\left.\begin{aligned} \tilde{\mathcal{C}}(p) &= \tilde{\mathcal{C}}_1(p) \times \cdots \times \tilde{\mathcal{C}}_k(p) \\ &= f_1^{-1}(p) \times \cdots \times f_k^{-1}(p) \\ &= f^{-1}(p) \end{aligned}\right\} \qquad (2)$$

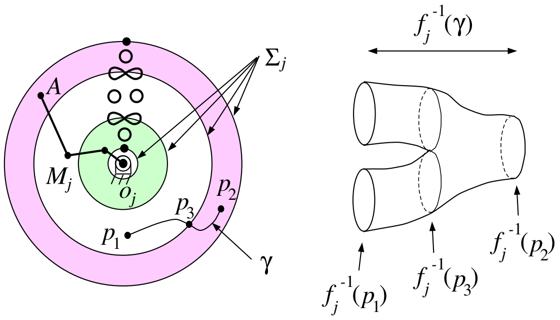


Fig. 2. **Left:** The workspace W_j of a three-link open chain M_j based at o_j. The critical set Σ_j of the kinematic map f_j is four concentric circles. The small circles, figure eights, and points at 12 o'clock show the topology of the C-space $\tilde{\mathcal{C}}_j(p)$ of the leg when its endpoint is fixed at a point in one of the seven regions delineated by the critical circles (one of the four circles or one of the three open annular regions between them). **Right:** The inverse image of the curve γ - a "pair of pants."

where by analogy, f is a total kinematic map of the star-shaped manipulator. Loosely speaking, the union of the C-spaces $\tilde{\mathcal{C}}(p)$ at each point p in W_A gives the C-space of a star-shaped manipulator:

$$\mathcal{C} = \bigcup_{p \in W_A} \tilde{\mathcal{C}}(p). \qquad (3)$$

Several properties of the C-spaces $\mathcal{C}_j$ and $\tilde{\mathcal{C}}_j(p)$ are highly relevant and so are reviewed here before analyzing the C-space of star-shaped manipulators. It is well known that the C-space of M_j is a product of circles (*i.e.*, $\mathcal{C}_j = (S^1)^{n_j}$) [1]. The workspace W_j contains a critical set Σ_j which is composed of all points p in W_j for which the Jacobian of the kinematic map $Df_j(\Theta_j)$ drops rank for some $\Theta_j \in f_j^{-1}(p)$. These points form concentric circles of radii $|l_{j,1} \pm l_{j,2} \pm \cdots \pm l_{j,n_j}|$, as shown in Fig 2. When A coincides with a point in Σ_j, the links can be arranged such that they are all colinear, in which case the number of instantaneous degrees of freedom of the endpoint of the leg is reduced from two to one.

Now consider the case where the endpoint of leg j is fixed to the point p. In other words, we are interested in the C-space $\tilde{\mathcal{C}}_j(p)$ of $\tilde{M}_j(p)$. In the 12 o'clock position in Fig. 2, points, circles, and figure eights are drawn to represent the global structures of $\tilde{\mathcal{C}}_j(p)$ in the seven regions of W_j. Specifically, when A is fixed to a point p on the outer-most critical circle, $\tilde{\mathcal{C}}_j(p)$ is a single point. For p fixed to any point in the largest open annular region, C-space is a single circle. Continuing inward, the possible C-space types are a figure eight (on the second largest critical circle), two disconnected circles, a figure eight again, a single circle, and a single point (on the inner-most critical circle).

A detailed analysis of $\tilde{\mathcal{C}}_j(p)$ with an arbitrary number of links in $\tilde{M}_j(p)$ can be found in [45]. The results that will be particularly useful in the analysis of star-shaped manipulators follow. First, the connectivity of $\tilde{\mathcal{C}}_j(p)$ is uniquely determined by the number of "long links." Consider the augmented link

[1] Recall the assumption of no joint limits.

set composed of the links of M_j and $\overline{o_j p}$, which will be called the fixed base link with length denoted by $l_{j,0}$. Let L_j be the sum of all the link lengths including the fixed base link (*i.e.,* $L_j = \sum_{i=0}^{n_j} l_{j,i}$). Further, let $\mathcal{L}_j(p)$ be a subset of $\{0, 1, ..., n_j\}$ such that $l_{j,\alpha} + l_{j,\beta} > L_j/2;\ \alpha, \beta \in \mathcal{L}_j(p),\ \alpha \neq \beta$. Over all such sets, let $\mathcal{L}_j^*(p)$ be a set of maximal cardinality. Then the number of long links of $\tilde{M}_j(p)$ is defined as $|\mathcal{L}_j^*(p)|$, where $|\cdot|$ denotes set cardinality.

Lemma *1:* **Kapovich and Milson [41], Trinkle and Milgram [45]**
The C-space $\tilde{\mathcal{C}}_j(p) = f_j^{-1}(p)$ has two components if and only if $|\mathcal{L}_j^*(p)| = 3$, and is connected if and only if $|\mathcal{L}_j^*(p)| = 2$ or 0. No other cardinality is possible.

Let us return to the discussion of Fig. 2. Viewing W_j as a base manifold and the C-space corresponding to each end point location as a fibre, it is apparent that the critical set Σ_j partitions W_j into regions over which the C-spaces $\tilde{\mathcal{C}}_j(p)$ form a trivial fibration. The implications of this observation are useful in determining the C-space of more complicated mechanisms. Consider a modification to $\tilde{M}_j(p)$ that allows the endpoint to move along a one-dimensional curve segment γ within W_j. Then as long as γ is entirely contained in one of the regions defined by the critical circles, $\tilde{\mathcal{C}}_j(\gamma) = \tilde{\mathcal{C}}_j(p) \times I$, where I is the interval. If γ crosses a critical circle transversally, then $\tilde{\mathcal{C}}_j(\gamma) = (\tilde{\mathcal{C}}_j(p_1) \times I) \bigcup \tilde{\mathcal{C}}_j(p_3) \bigcup (\tilde{\mathcal{C}}_j(p_2) \times I)$, where p_1 is a point in one of the two open annular regions containing γ, p_2 is a point in the other, and p_3 is a point on the critical circle crossed by γ, and $\bigcup$ denotes the standard "gluing" operation. In Fig. 2, an example γ and the corresponding C-space $\tilde{\mathcal{C}}_j(\gamma)$ are shown.

IV. Analysis of Star-Shaped Manipulators

For star-shaped manipulators with one or two legs, the global topological properties of the C-space $\mathcal{C}$ are fully understood (for one, see [43]; for two, see [45], [44]). The goals of this section are to study the global properties of $\mathcal{C}$ when M has more than two legs and to derive necessary and sufficient conditions for solution existence to the motion planning problem.

1) Local Analysis: As a direct generalization of the critical set of a single leg, we define the critical set of a star-shaped manipulator as a subset Σ of W_A such that for every $p \in \Sigma$, there exists a configuration c such that at least one of the Jacobians $\{Df_1(c), \cdots, Df_k(c)\}$ drops rank. By definition we have:

$$\Sigma = \left(\bigcup_{i=1}^{k} \Sigma_i \right) \bigcap W_A. \qquad (4)$$

An advantage of this definition is that Σ can be used to stratify W_A such that each stratum is trivially fibred. Figure 3 shows a star-shaped manipulator with two legs. The critical set Σ is the boundary of the lune formed by the intersection of the outer critical circles of their individual workspaces. For every point interior to the lune, the fibre is two circles (the direct product of two points with one circle). The fibres associated to the vertices of the lune are single points, which correspond to simultaneous full extension of the two legs.

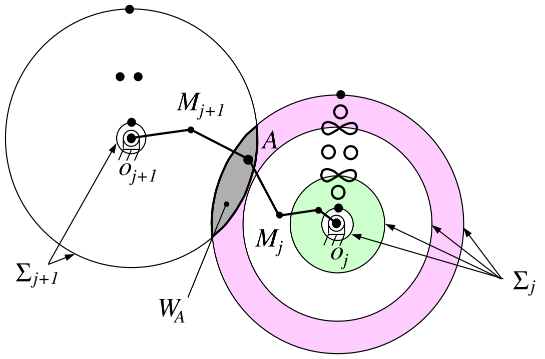


Fig. 3. The workspace W_A of A for a star-shaped manipulator with $k = 2$ is the intersection of the workspaces of A for each leg considered separately. The critical set Σ is composed of the black circular arcs where they bound or intersect the gray area.

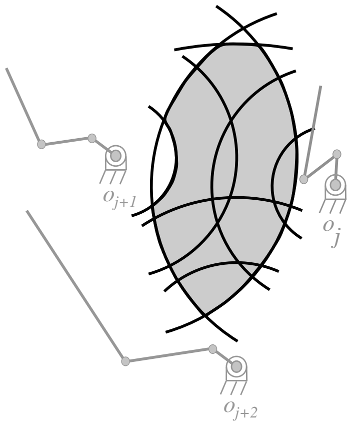


Fig. 4. Workspace (shaded gray) of a star-shaped manipulator with three legs. The critical set partitions W_A into 12 two-dimensional, 32 one-dimensional, and 21 zero-dimensional chambers.

Fig. 4 shows a possible workspace for a star-shaped manipulator with three legs. The critical set defines 65 distinct sets ${}^{d}U_i$ of varying dimension d, where i is an arbitrarily assigned index that simply counts components. We will refer to these sets as *chambers*. There are 12 two-dimensional, 32 one-dimensional, and 21 zero-dimensional chambers, each of which is trivially fibred. Removing the ${}^{0}U_i$ from Σ partitions it into open one-dimensional chambers ${}^{1}U_i,\ i = 1, \cdots, {}^{1}m$. Removing ${}^{0}U_i$ and ${}^{1}U_i$ from W_A yields open two-dimensional sets ${}^{2}U_i,\ i = 1, \cdots, {}^{2}m$, for which the following relationships hold:

$$\Sigma = \left(\bigcup_{i=1}^{{}^{0}m} {}^{0}U_i \right) \bigcup \left(\bigcup_{i=1}^{{}^{1}m} {}^{1}U_i \right) \qquad (5)$$

$$W_A - \Sigma = \bigcup_{i=1}^{{}^{2}m} {}^{2}U_i. \qquad (6)$$

Proposition. *1:* For all $d = 0, 1, 2$ and i, $f^{-1}({}^{d}U_i) = {}^{d}U_i \times f^{-1}(p)$, where p is any point in ${}^{d}U_i$ and the operator $\times$

denotes the direct product. Gluing the $f^{-1}({}^dU_i)$ for all i and d gives the total C-space $\mathcal{C}$.

Proof: When $d = 0$, ${}^0U_i$ contains a single point, the result follows. When $d = 1$, ${}^1U_i$ belongs to one critical circle of one leg, say $\tilde{M}_j$. Any two points $p_1, p_2 \in {}^1U_i$ are related by a Euclidean rotation $p_2 - o_j = R(p_1 - o_j)$, indicating that $\tilde{\mathcal{C}}_j(p_1)$ and $\tilde{\mathcal{C}}_j(p_2)$ are homotopic. Thus $\tilde{\mathcal{C}}_j(p)$ for all $p \in {}^1U_i$ have equivalent topological structure. For the other legs $\tilde{M}_l$, $l \neq j$, according to [44] (Lemma 6.1 and Corollary 6.5) $\tilde{\mathcal{C}}_l(p)$ for all $p \in {}^1U_i$ have equivalent topological structures as ${}^1U_i$ is free of critical points of $\tilde{M}_l(p)$. Thus $f^{-1}(p) = \tilde{\mathcal{C}}_1(p) \times \cdots \times \tilde{\mathcal{C}}_k(p)$ for all $p \in {}^1U_i$ have equivalent topological structures. The case when $d = 2$ can be proved by applying Lemma 6.1 and Corollary 6.5 of [44] to all legs. ■

Proposition 1 and the fact that ${}^dU_i$ is a simply connected set, reveal that each component of $f^{-1}({}^dU_i)$ is a direct product of one component of $\tilde{\mathcal{C}}_j(p)$, $j = 1, \cdots, k$, with a d-dimensional disk. Using $|\mathcal{L}_j^*(p)|$, $j = 1, \cdots, k$ and Lemma 1, one can show that the number of components of $f^{-1}({}^dU_i)$ is 2^{k_0}, where $k_0 \leq k$ is the number of legs for which $|\mathcal{L}_j^*(p)| = 3$.

2) Local Path Existence: Before considering the global path existence problem, consider motion planning between two valid configurations c_{init} and c_{goal} for which the junction A lies in the same chamber. Since the fibre over every point in ${}^dU_i$ is equivalent, path existence amounts to checking the component memberships of the configurations c_{init} and c_{goal}.

For a single leg $\tilde{M}_j(p)$, if the number of long links $|\mathcal{L}_j^*(p)|$ is not three, then any two configurations of $\tilde{M}_j(p)$ are in the same component. When $|\mathcal{L}_j^*(p)| = 3$, choose any two long links and test the sign of the angle between them (with full extension taken as zero). There are two possible signs, one corresponding to *elbow-up* and the other to *elbow-down.* If for two distinct configurations of $\tilde{M}_j$, A lies in the same chamber, there is a continuous motion between them while keeping A in this chamber, if and only if the elbow sign is the same at both configurations (naturally, one must perform the sign test with the same two links and in the same order for both configurations). Considering all the legs together, a continuous motion of A in ${}^dU_i$ exists if and only if a motion exists for each leg individually. The previous discussion serves to prove the following result.

Proposition. *2:* Restricted to $f^{-1}({}^dU_i)$, two configurations $c_1, c_2 \in f^{-1}({}^dU_i)$ are path connected if and only if for each leg $\tilde{M}_j$ with $|\mathcal{L}_j^*| = 3$ in ${}^dU_i$, the elbow angle of $\tilde{M}_j$ has the same sign at c_1 and c_2.

Proposition 2 completely solves the path existence problem if W_A consists of a single chamber. However, things become complex when W_A has more than one chamber.

3) Singular Set and Global C-space Analysis: Recall that the C-space $\mathcal{C}$ is a union of $f^{-1}({}^dU_i)$, $d \in \{0, 1, 2\}$, $i = 1, \cdots, {}^dm$ and that $f^{-1}(p)$, $p \in {}^dU_i$ for $d \neq 2$ and all i is a set containing at least a singularity of f. Combining the local C-space and singular set analysis yields the global structure of C-space.

Proposition. *3:* For all $p \in \Sigma_j$, $f_j^{-1}(p)$ is a singular set containing isolated singularities. If a singularity separates its neighborhood V in $f_j^{-1}(p)$, then it is these singularities which glue the two separated components in $f_j^{-1}(q)$ where $q \in W_A - \Sigma_j$ is a point sufficiently close to p.

Proof: First it is obvious that $f_j^{-1}(p)$ contains isolated singularities for there are finite ways to colinearize all the links of a close chain. Second, let

$$\gamma : (-\varepsilon, \varepsilon) \to W_A, \; \gamma(0) = p$$

be a curve that is transverse to Σ_j. According to Corollary 6.6 of [44], the distance function $s(\gamma(t)) = \int_0^t |\dot{\gamma}| dt$ defines a Morse function on $f_j^{-1}(\gamma)$

$$s \circ f_j : f_j^{-1}(\gamma) \to \mathbb{R}.$$

Note that 0 is a singular value of $s \circ f_j$ and the isolated singularities of $f_j^{-1}(p)$ are also singularities of $s \circ f_j$. The result of Morse theory applying to $s \circ f_j$ yields that $(s \circ f_j)^{-1}(0) = f_j^{-1}(p)$ is given by attaching a handle to $(s \circ f_j)^{-1}(\varepsilon_0) = f_j^{-1}(q)$ for a sufficiently small ε_0 and q a point sufficiently close to p. The Proposition follows. ■

Next, we establish necessary and sufficient conditions for the connectivity of $\mathcal{C}$. Let J be the index set such that for all $j \in J$, $|\mathcal{L}_j^*| = 3$ for at least one chamber ${}^dU_i$. We prove the following theorem.

Theorem *1:* Suppose $W_A = \bigcup_{d=0}^{2} \left(\bigcup_{i=1}^{{}^dm} {}^dU_i \right)$. Then $\mathcal{C} = f^{-1}(W_A)$ is connected if and only if:

1) W_A is connected;
2) $\Sigma_j \bigcap W_A \neq \emptyset$ for all $j \in J$.

Proof: (i) "Necessity:" Since $\mathcal{C}$ is a fibration of the base manifold W_A, it can have one component only when W_A has one component. Thus item 1 of Theorem 1 is required. Second, in order that $\mathcal{C}$ be connected, for each leg M_j restricted to W_A, the C-space $\tilde{\mathcal{C}}_j(W_A) = f_j^{-1}(W_A)$ must be connected. By definition, for all $j \in J$, there exists a chamber ${}^dU_i$ such that $|\mathcal{L}_j^*| = 3$. The result of Proposition 3 means that $\tilde{\mathcal{C}}_j(W_A)$ is connected only if $W_A \bigcap \Sigma_j \neq \emptyset$.

(ii) "Sufficiency:" Item 1 and 2 imply that $\tilde{\mathcal{C}}_j(W_A)$ are path connected for all j. Moreover, $\mathcal{C}$ is a fibration over W_A. The result follows. ■

Fig. 5 illustrates the global connectivity for an example W_A corresponding to a star-shaped manipulator with two legs and a workspace for which there are two chambers ${}^2U_1$ and ${}^2U_3$ where leg 1 has three long links and another chamber ${}^2U_4$ where both legs have three long links. Among these chambers, ${}^1U_1$ and ${}^1U_2$ belong to Σ_1, and ${}^1U_3$ belongs to Σ_2. According to Theorem 1, the C-space is path connected. In this example, the $\mathcal{C}$ is the product of the two structures shown.

Corollary *1:* Two configurations c_1 and c_2 of a star-shaped manipulator are in the same component if and only if

1) $f(c_1)$ and $f(c_2)$ are in the same component of W_A;
2) For each leg j with $|\mathcal{L}_j^*| = 3$ for all chambers ${}^dU_i$ in the component of W_A which contains $f(c_1)$ and $f(c_2)$, the elbow sign is same at both c_1 and c_2.

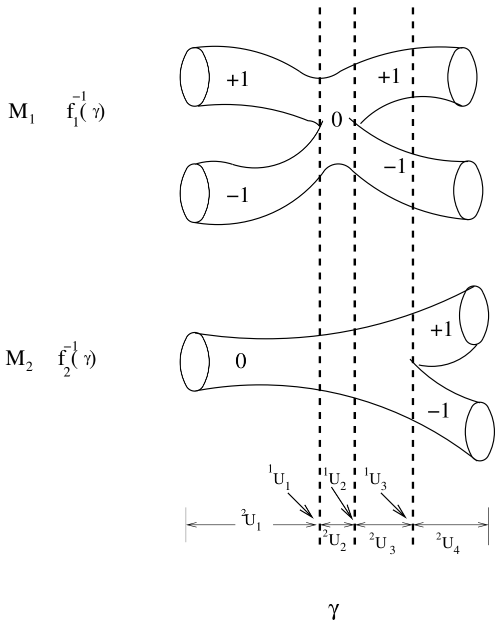


Fig. 5. C-space of a star-shape manipulator with two legs. For simplicity, only the portion of $f^{-1}(\gamma)$ is shown, where γ is a continuous curve in W_A that visits all chambers.

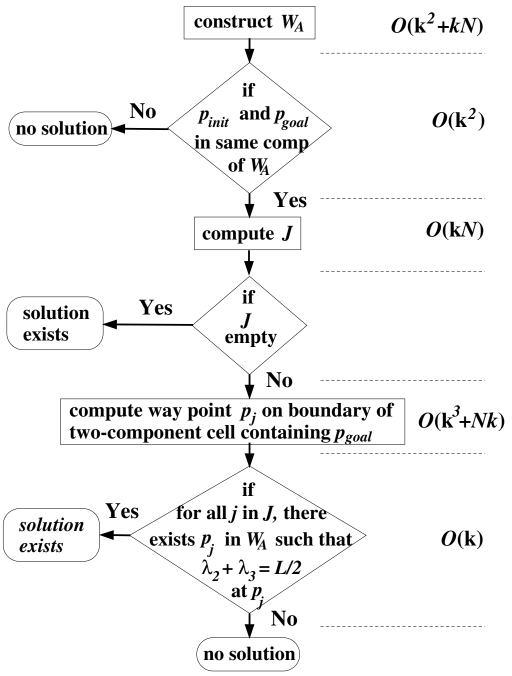
Remark *1:* As a matter of fact, Σ completely determines the connectivity of C-space. When computing a path between two given configurations, often motions of the junction to points on Σ are incorporated to allow the adjust the signs of the leg angles. However, inevitable deviations of the junction from Σ caused by numerical errors, make it impossible to adjust the sign of legs while fixing its end point. For these reasons, points in 2D chambers are preferred for sign adjustment.

V. A Polynomial-Time, Exact, Complete Algorithm

Our algorithm consists of two main routines, `PathExists` and `ConstructPath`. The logical flow of `PathExists` is illustrated in Figure 6. Its input is the topology and link lengths of a star-shaped manipulator and two valid configurations, c_{init} and c_{goal}. The output is the answer to the path existence question. Below we will show that the complexity of `PathExists` is $O(k^3N^3)$, where N is the maximum number of links in a leg and k is the number of legs.

The approach taken is to compute W_A and then, for each leg with its end point constrained to lie in W_A, to determine if its initial and goal configurations are path connected. Since the C-space of a leg is guaranteed to be connected if one of its critical circles Σ_j intersects W_A, the most straight forward way to test connectivity is to explicitly perform the intersections. However, since there are as many as 2^{n_j-1} critical circles, any algorithm based on this approach will have worst-case complexity that is at least exponential in N. The key contribution of `PathExists` is a polynomial-time algorithm for checking the existence of an intersection between W_A and a critical circles - even though there is an exponential number of these circles.

1. Construct W_A We compute W_A in three steps.



construct W_A
$O(k^2+kN)$
if p_{init} and p_{goal} in same comp of W_A
No
no solution
$O(k^2)$
Yes
compute J
$O(kN)$
if J empty
Yes
solution exists
No
compute way point p_j on boundary of two-component cell containing p_{goal}
$O(k^3+Nk)$
if for all j in J, there exists p_j in W_A such that $\lambda_2+\lambda_3=L/2$ at p_j
Yes
solution exists
$O(k)$
No
no solution

Fig. 6. Logical flow and complexity of the major steps of `PathExists`.

Step 1: Compute the boundary circles of W_j. In general, W_j is an annulus. The radius of its outer boundary circle is $r_{\max} = \sum_{i=1}^{n_j} l_{j,i}$, while that of its inner boundary circle, $r_{\min}$, can be determined by comparing $l_{\max} := \max_i l_{j,i}$ and $r_{\max} - l_{\max}$. If $l_{\max} > r_{\max} - l_{\max}$, then $r_{\min} = 2l_{\max} - r_{\max}$, else, $r_{\min} = 0$;
Step 2: Decompose the whole plane into cells using all boundary circles of all legs (e.g., the line sweeping algorithm can do this), and construct the cell adjacency graph;
Step 3: Pick a point from the interior of each cell, compute its distance from each base point, and compare the distance with the radii of the two boundary circles of W_j. The set of cells which can be reached by all legs constitute W_A.
The complexity of this 2-D cell decomposition algorithm is $O(k^2 + kN)$.

2. Are p_{init} and p_{goal} in same component of W_A? As an immediate consequence of the cell decomposition, this can be answered directly by searching the cell graph.

3. Compute J This step is used to filter out easy solution existence checks, based on the cardinality and members of the sets $\mathcal{L}_j^*(p_{\text{init}})$ and $\mathcal{L}_j^*(p_{\text{goal}})$. For each leg $\tilde{M}_j(p_{\text{init}})$, compute L_j (see Section III) and find the three longest links of the set $\{l_{j,0}, ..., l_{j_{n_j}}\}$. Denote these links by $(p_{\text{init}}; \lambda_{j,1}, \lambda_{j,2}, \lambda_{j,3})$. Do the same for (p_{goal}) and define $(p_{\text{goal}}; \lambda_{j,1}, \lambda_{j,2}, \lambda_{j,3})$. This requires $O(N)$ work. Finally, $|\mathcal{L}_j^*(p_{(\cdot)})| = 3$ if and only if $\lambda_{j,2} + \lambda_{j,3} > L_j/2$. If $\mathcal{L}_j^*(p_{\text{init}}) = \mathcal{L}_j^*(p_{\text{goal}})$ and $|\mathcal{L}_j^*(p_{\text{init}})| = 3$, and if the signs of the long links are different at c_{init} and c_{goal}, then add j into J. Computing J is $O(kN)$.

4. Does the set of long links vary for all $j \in J$? If and only if $q \in W_A$ exists such that $\mathcal{L}_j^*(q) \neq \mathcal{L}_j^*(p_{\text{init}})$, then it is possible to make the long links colinear and thus change the

signs of their relative angles. This can be done by computing a point $q \in W_A$ on the boundary of the cell that contains p_{goal} and keeps the same set $\mathcal{L}_j^*(p)$ for all p in this cell. This boundary is characterized by $\lambda_{j,2} + \lambda_{j,3} = L_j/2$. Since $l_{j,0}$ is the only link whose length varies along with p, this boundary must be one or two circles (called inner and outer circles, respectively) whose radii, denoted $d_{\max}$ and $d_{\min}$, depend on the link lengths of the leg. Let $L_{j,0} = \sum_{i=1}^{n_j} l_{j,i}$ and suppose the four longest links at p_{goal} are $(\lambda_{j,1} > \lambda_{j,2} > \lambda_{j,3} > \lambda_{j,4})$ with $\lambda_{j,2} + \lambda_{j,3} > L_j/2$, we deduce the radii of the boundary circles for four different cases:

Case 1: if $l_{j,0}(p_{\text{goal}}) = \lambda_{j,1}$, then $d_{\max} = 2(\lambda_{j,2}+\lambda_{j,3}) - L_{j,0}$, and $d_{\min} = \max\{L_{j,0} - 2\lambda_{j,3}, 2(\lambda_{j,3} + \lambda_{j,4}) - L_{j,0}\}$.

Case 2: if $l_{j,0}(p_{\text{goal}}) = \lambda_{j,2}$, $d_{\max} = 2(\lambda_{j,1} + \lambda_{j,3}) - L_{j,0}$, and $d_{\min} = \max\{L_{j,0} - 2\lambda_{j,3}, 2(\lambda_{j,3} + \lambda_{j,4}) - L_{j,0}\}$.

Case 3: if $l_{j,0}(p_{\text{goal}}) = \lambda_{j,3}$, $d_{\max} = 2(\lambda_{j,1} + \lambda_{j,2}) - L_{j,0}$, and $d_{\min} = \max\{L_{j,0} - 2\lambda_{j,2}, 2(\lambda_{j,2} + \lambda_{j,4}) - L_{j,0}\}$.

Case 4: Otherwise, $d_{\max} = \min\{2(\lambda_{j,2} + \lambda_{j,3}) - L_{j,0}, L_{j,0} - 2\lambda_{j,2}\}$, and $d_{\min} = 0$.

If there is no overlap between the two boundary circles and the component of W_A that contains p_{init} and p_{goal}, then no path exists between c_{init} and c_{goal}. Otherwise, path exists and we obtain way points p_j for all leg $j \in J$. Computing $d_{\max}$, $d_{\min}$, and the way points p_j is $O(kN)$.

The basic idea of `ConstructPath` is that when moving from c_{init} to c_{goal}, those legs $j \in J$ may require a change in the signs of relative angles between long links, which is always possible at the way point p_j or other critical points of the corresponding leg. A natural approach then is to use two motion generation primatives: *accordion move* and *sign-adjust move*. The former moves the thorax endpoint (at A) along a specified path segment with all legs moving compliantly so that all loop closures are maintained. The latter keeps the endpoint fixed at a way point $q_j \in \Sigma_j$ (e.g., $q_j = p_j$ or other critical points) while moving leg j into a singular configuration and then to a nearby configuration with the sign of the relative angle between a pair of long links in this leg chosen to match those of c_{goal}.

The input of `ConstructPath` is W_A and its cell graph, c_{init}, c_{goal}, and the set of way points $p_j \in W_A, j \in J$ computed during the execution of `PathExist`.

1. Construct an initial path `ConstructPath` explores the cell graph of W_A, and constructs a path in W_A connecting p_{init} to p_{goal} and visiting all of the way points. Since there are at most k way points, this can be done in $O(k^3)$ time (the path has $k+1$ segments each with $O(k^2)$ arcs).

2. Construct *guards* and insert the guards into the path Notice that when one accordion moves a leg in a cell in which the number of long links is not 3 (called one-component cell), neither the signs of concatenating angles, nor the sign between any pair of links in this leg will be kept invariant. Thus even the sign between a pair of long links is adjusted to the desired one at a way point, it still could change if the leg keeps moving in a one-component cell. For this reason, we set *guards* for legs which have three long links at p_{goal}. These are the set of points q_j, each of which is the last intersection point between the above constructed path in W_A and the boundary of the two-component cell of leg j containing p_{goal}. Thus the number of guards (q_j's) may be more than the number of way points since the number of legs that have three long links at p_{goal} may be more than the cardinality of J. Next the *guards* are inserted into the path. Later when we construct the path in $\mathcal{C}$, sign-adjust moves are only performed at *guards* q_j (but not p_j) for after that the thorax endpoint gets into the two-component cell and the sign between a pair of long links will not change during accordion moves, i.e., the leg will always remain in the right component of its C-space. Assuming each arc in the path is approximated by a fixed number of line segments, finding guards is $O(k^3)$.

3. Accordion moves and sign-adjust moves The path in $\mathcal{C}$ then is produced by using accordion moves along the path and sign-adjust moves at the *guards*. At each *guard*, one checks the sign between a pair of long links of the corresponding leg. If it does not match the goal one, then the junction point is fixed while a sign-adjust move is executed, otherwise, the accordion move continues. Once A is coincident with p_{goal}, one is assured by the previous steps, that with A fixed at p_{goal}, the configuration of each leg is in the same component of its current C-space $\tilde{C}_j(p_{\text{goal}})$ as c_{goal}. The final move can be accomplished using a special accordion move algorithm found in [45]. At this stage, we remark that finding the set of way points p_j and planning an initial path visiting all p_j is necessary for otherwise, an arbitrary path between p_{init} and p_{goal} may not intersect the boundary of the two-component cell of a leg that contains p_{goal}.

The complexity of the accordion move algorithms reported in [45] are $O(N^3)$. Since the path has $O(k^3)$ line segments the complexity of `ConstructPath` is $O(k^3N^3)$. Note that accordion move algorithms with the required behavior can be designed to be $O(N^2)$, so the complexity of `ConstructPath` could be reduced.

Overall, our path planning algorithm is $O(k^3N^3)$.

VI. Path Optimization and Robustness

The above algorithm can be refined with path optimization and the consideration of robustness. It is obvious that in our algorithm the way points and thus the path between two given configurations is not unique if these exists a path. So a natural problem is path optimization in the sense of finding the shortest path. Here optimization is multidimensional since both the choice of way points, the order of way points, and the path between two consecutive way points could be optimized.

Since $\mathcal{C}$ is a fibration over W_A, a meaningful optimization problem is to construct a shortest path in W_A (corresponding to the minimal motion of the thorax endpoint) that connects p_{init} and p_{goal} and visits all p_i, followed by determining the minimal motion of all legs that maintains all loop closure constraints.

The former problem is generally a nonconvex and nonlinear optimization problem, since W_A could be nonconvex, and the objective function (the distance function) as well as the constraints are nonlinear. Let $p(t)$ be this path with $p(0) =$

p_{init} and $p(1) = p_{\text{goal}}$. The problem can be described as

$$\begin{gathered}\min \int_0^1 \|dp\| \\ p(t) \in W_A, \forall t \in [0,1] \\ p_i \in \{p(t)\}, \forall i \\ p_i \in \mathcal{A}_{\delta(i)}\end{gathered}$$

where $\mathcal{A}_j, j \in J$ is the boundary arcs of the two-component cell of leg j that contains p_{goal}. $\delta(J)$ is a permutation of J with $\delta(J) = J$. Solving this problem exactly is extremely hard, but a random search method can be used to obtain a good approximate solution.

To solve the latter problem, we notice that for a local motion $dp = [dx, dy]$ of the thorax endpoint, $d\Theta_j^T d\Theta_j$ is minimized if and only if.

$$d\Theta_j = J_j^+ dp$$

where $J_j = \frac{\partial f_j}{\partial \Theta_j}$ is the Jacobian of leg j, and $J_j^+ = J_j^T (J_j J_j^T)^{-1} d\theta_j$.

Another important issue about our planning algorithm is robustness. The sign-adjust move of leg j performed at a guard q_j is only feasible when $\mathcal{C}_j(q_j)$ is connected. Since $q_j \in \Sigma_j$ which is only 1-D, a small perturbation of the junction point in W_A (e.g., due to numerical errors) will violate the condition $p \in \Sigma_j$. When p moves into a two-component cell, then the sign-adjust move will fail. A remedy to this is to modify the path of the thorax in the neighborhood of $q_j \in \Sigma_j$ so that a point q'_j in the interior of a one-component cell is reached. After the sign of leg j is adjusted to the desired one with p fixing at q'_j, we apply a constrained accordion move algorithm to ensure that the leg j stays in the right component of $\mathcal{C}_j(p)$ just before its thorax endpoint enters the two-component cell containing p_{goal}.

VII. Examples

In this section, we demonstrate the correctness and complexity of our algorithm through two examples: a manipulator with three three-link legs, and a manipulator with three five-link legs. Movies of the motion plans are very helpful in understanding the figures. They can be found at `http://www.stanford.edu/~phwu1/multiloop`.

In the first example, two of the three legs of the manipulator have three long links when A is fixed at p_{goal}. Figure 7 shows the manipulator in its starting and goal configurations. Our algorithm predicts $J = \emptyset$. Then the algorithm constructs a path in W_A from p_{init} to p_{goal}, drawn as the dark solid lines in Fig. 8. This path intersects the boundary circles of the two-component cells of legs j containing p_{goal} several times, among which q_j, $j = 1, 2$ are the last ones. These two points are the guards (drawn as

diamonds in Fig. 9) where sign-adjust moves are performed.

At q_j, $j = 1, 2$, we check the sign of a pair of long links of leg j and see if it matches its sign at the goal. If not, we fix the other two legs and adjust the sign of the chosen long links in leg j. In this particular example, we chose the two longest links as the pair of long links. Before leaving q_j via the next accordion move, the pair of long links of leg j was moved to the elbow-opposite configuration (recall that there

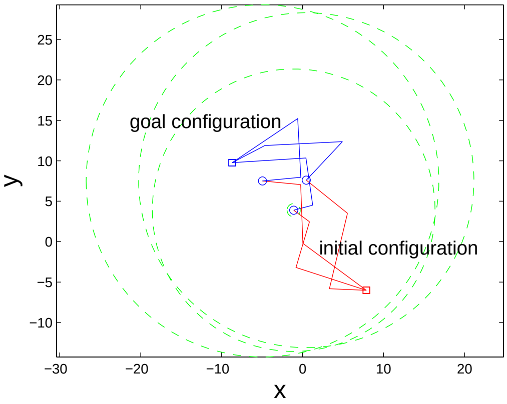


Fig. 7. Manipulator's initial configuration (junction on the right, drawn red) and goal configuration (junction just below the top left, drawn blue.) The boundary circles of W_j are drawn as dashed green lines.

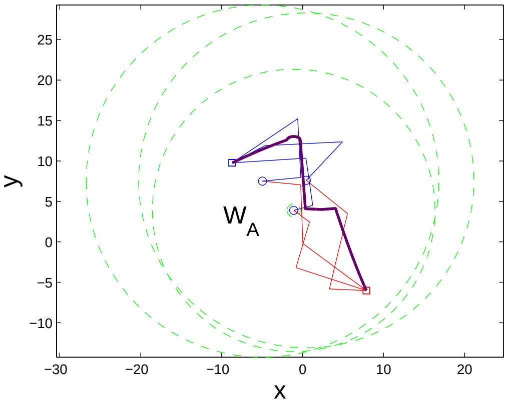


Fig. 8. A path between p_{init} and p_{goal} that is completely contained in W_A.

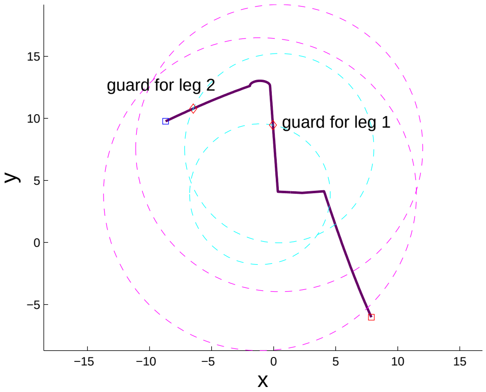


Fig. 9. The two guard points are the last intersection points between the path of A and the boundary circles of two two-component cells.

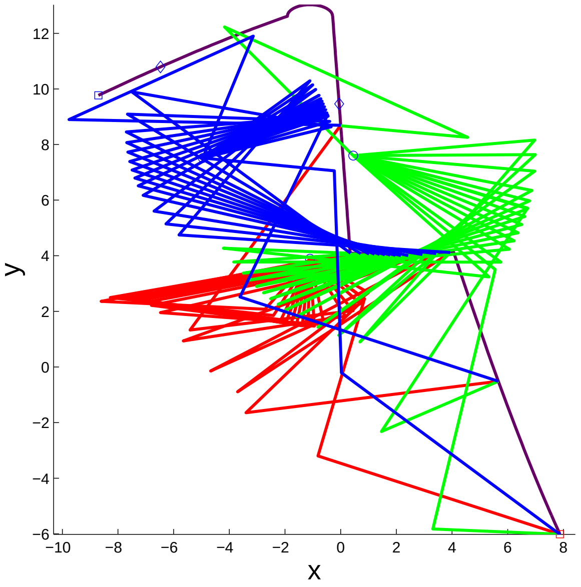


Fig. 10. All legs use an accordion move to move the junction A to the first guard q_1 of leg 1.

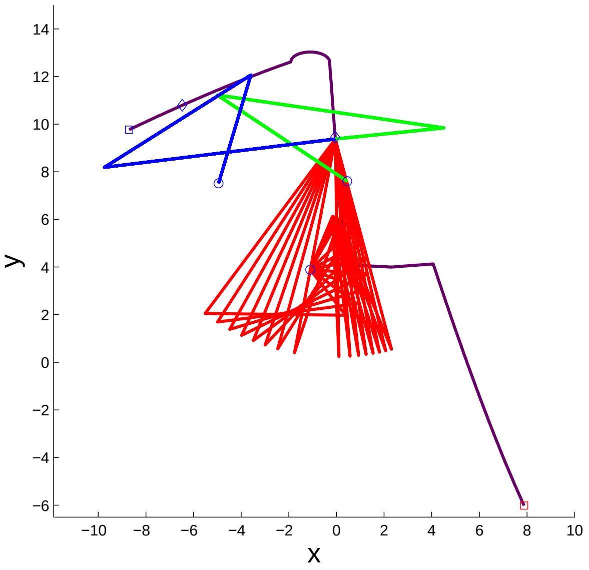


Fig. 11. With the junction A at q_1, the joint angles of leg 1 can be adjusted to achieve the signs required at the goal configuration. All other legs are fixed in place.

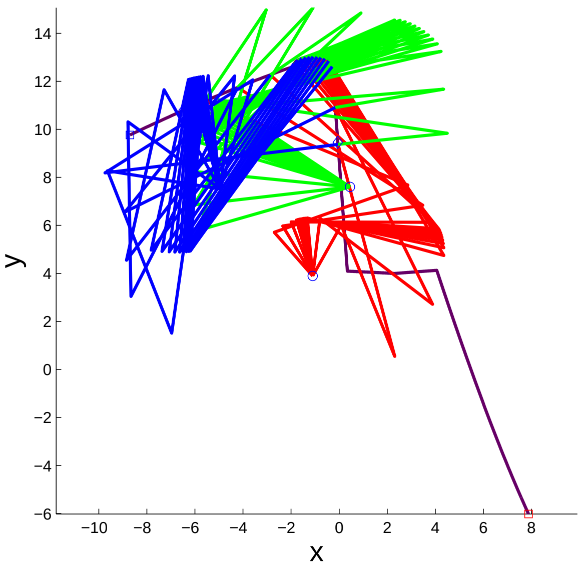


Fig. 12. All legs use an accordion move to move the junction A to the second guard q_2 of leg 2.

are two configurations for these two links, one is “elbow up”, the other is “elbow down”), which has exactly the same sign as the goal configuration. The Trinkle-Milgram algorithm [45] is used to plan such a motion between the two elbow-opposite configurations. Figures 10, 11, 12 13, 14, and 15 show the progress of the manipulation plan as the steps of the complete planning algorithm are carried out.

Next, we apply our algorithm to a bit more complex example, a star-shaped manipulator with three five-link legs. The initial and goal configurations of the manipulator and the motion of the manipulator that achieve the goal configuration are shown in Figs. 16, 17, 18, 19, 20, 21, and 22. Among which, 16, 17, and 18 respectively show the initial and goal configurations of the manipulator, the computed way point for a leg (leg 2 in our numbering scheme) that has two components at $p_{\rm init}$ and $p_{\rm goal}$ (i.e., J=$\{2\}$)., the path for the junction point that connects $p_{\rm init}$, the way point, and $p_{\rm goal}$, and the guard which is the last intersection point between this path and the boundary circle of the two-component cell.

The computation time for path existence for star-shaped manipulators with less than 10 legs, and legs of less than 10 links is typically from less than 1 second to a few seconds when run in a Matlab, P4, WindowsXP system.

VIII. Conclusion

In this paper, we studied the global structural properties of planar star-shaped manipulators. Via the analysis of the critical set Σ, we derived the global connectivity of the C-space, and necessary and sufficient conditions for path existence. Based on these results, we devised a complete polynomial algorithm for motion planning. Simulation examples were used to illustrate the key ideas behind the motion planning problem of planar star-shaped manipulators.

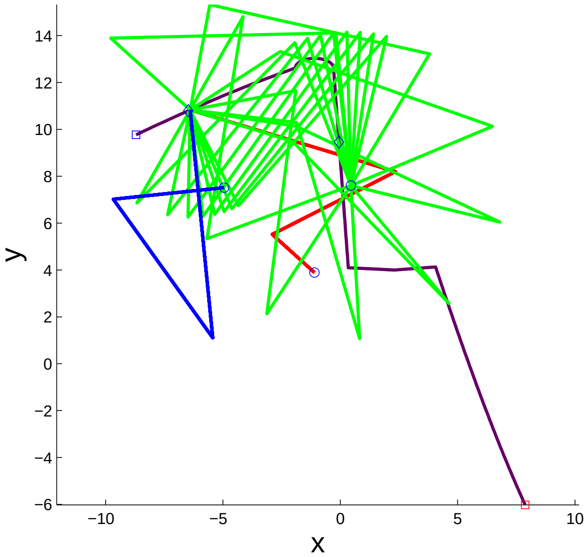


Fig. 13. With the junction A at q_2, the joint angles of leg 2 can be adjusted to achieve the signs required at the goal configuration. All other legs are fixed in place.

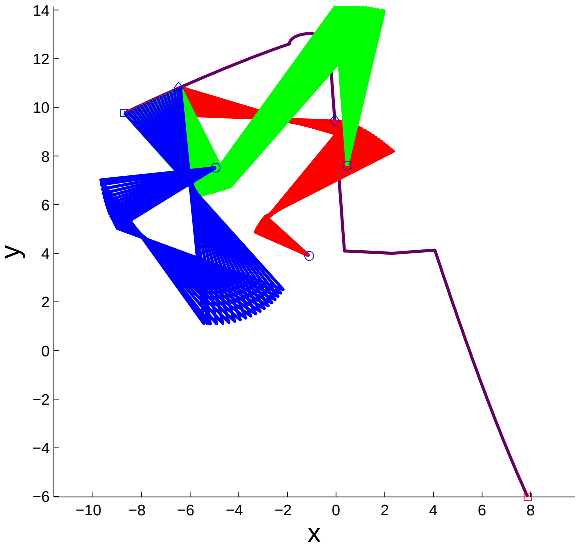


Fig. 14. All legs use an accordion move to move the junction A to its goal location. The signs of the joint angles are preserved guaranteeing that legs 1 and 2 will be in the correct C-space component once A is fixed at the goal position.

Acknowledgment

The authors would like to thank Ryan Trinkle for help on the analysis of the solution existence algorithm, and the National Science Foundation for its support through grants 0139701 (DMS-FRG), 0413227 (IIS-RCV), and 0420703 (MRI).

References

[1] M. Cherif and K.K. Gupta, *Planning Quasi-Static Fingertip Manipulation For Reconfiguring Objects*. IEEE Transactions on Robotics and Automation, Vol. 15, No. 5, PP. 837-848, 1999.

[2] D. E. Koditschek, *Exact Robot Navigation by Means of Potential Functions: Some Topological Considerations*. IEEE International Conference on Robotics and Automation, PP. 1-6, 1987.

[3] Lozano-Perez83, *Spatial Planning: A Configuration Space Approach*. IEEE Transactions on Computers, Vol. C-32, No. 2, PP. 108-119, 1983.

[4] J. Cortés, T. Siméon, and J.P. Laumond, *A Random Loop Generator for Planning the Motions of Closed Kinematic Chains using PRM Methods*. In Proceedings of the 2002 IEEE International Conference on Robotics and Automation, pages 2141-2146, 2002.

[5] J. Cortes and T. Siméon, *Probabilistic Motion Planning for Parallel Mechanisms*. In Proceedings of the 2003 IEEE International Conference on Robotics and Automation, pages 4354-4359, 2003.

[6] E. Rimon and D. E. Koditschek, *Exact Robot Navigation Using Cost Functions: The Case of Distinct Spherical Boundaries in E^n*. IEEE International Conference on Robotics and Automation, PP. 1791-1796, 1988.

[7] D. Hsu, J.C. Latombe, and R. Motwani, *Path Planning in Expansive Configuration Spaces*. IEEE International Conference on Robotics and Automation, 1997.

[8] E. Rimon and D. E. Koditschek, *The Construction of Analytic Diffeomorphisms for Exact Robot Navigation on Star Worlds*. IEEE International Conference on Robotics and Automation, PP. 21-26, 1989.

[9] M. Kapovich and J. Millson, *On the moduli spaces of polygons in the Euclidean plane*. Journal of Differential Geometry, Vol. 42, PP. 133-164, 1995.

[10] N. Shvalb, M. Shoham, D. Blanc *The Configuration Space of an Arachnoid Mechanism*. To appear in Forum Mathemaicum 2005.

[11] J.-C. Hausmann and A. Knutson, *The cohomology ring of polygon spaces*. Ann. Inst. Fourier, Vol. 48, PP. 281-321, 1998.

[12] Y. Kamiyama, M. Tezuka, and T. Toma, *Homology of the Configuration Spaces of Quasi-Equilateral Polygon Linkages*. Transactions of the American Mathematical Society, Vol. 350, No. 12, PP. 4869-4896, 1998.

[13] J. Barraquand and J.-C. Latombe, *Nonholonomic Multibody Mobile Robots: Controllability and motion planning in the presence of obstacles*. IEEE Interantional Conference on Robotics and Automation, PP. 2328-2335, 1991.

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

[15] V. Guillemin and A. Pollack *Defferential Topology*. Publisher:Prentice Hall PTR, 1974.

[16] J. Hopcroft and G. Wilfong, *On the motion of objects in contact*. Int. J. Robotics Research, Vol. 4, 1986, PP. 32-46.

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

[18] J.C. Latombe, *Robot Motion PLanning*. Kluwer Academic Publishers, 1992.

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

[20] 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.

[21] S. M. LaValle and J. J. Kuffner, *Rapidly-exploring random trees: Progress and prospects*. In B. R. Donald, K. M. Lynch, and D. Rus, editors, Algorithmic and Computational Robotics: New Directions, pages 293-308, A K Peters, Wellesley, MA, 2001.

[22] F. Lingelbach, *Path Planning using Probabilistic Cell Decomposition*. IEEE International Conference on Robotics and Automation, PP. 467-472, 2004.

[23] S.M. LaValle, M.S. Branicky and S.R. Lindemann, *On the Relationship between Classical Grid Search and Probabilistic Roadmaps*. The International Journal of Robotics Research, Number 23, Volume 7/8 (2004):673-692.

[24] S. R. Lindemann and S. M. LaValle, *Current issues in sampling-based motion planning*. In P. Dario and R. Chatila, editors, Proc. Eighth Int'l Symp. on Robotics Research. Springer-Verlag, Berlin, 2004. To appear.

[25] J. Yakey, S. M. LaValle, and L. E. Kavraki, *Randomized path planning for linkages with closed kinematic chains*. IEEE Transactions on Robotics and Automation, 17(6):951–958, December 2001.

[26] J.P. Merlet, *Parallel Robots*. Kluwer Academic Publishers, 2000.

[27] J.T. Schwartz and M. Sharir, *On the piano movers II. General techniques for computing topological properties on real algebraic manifolds*. Adv. Appl. Math., vol.4, PP. 298-351, 1983.

[28] L. Han and N.M. Amato, *A kinematics-based probabilistic roadmap method for closed chain systems*. in Algorithmic and Computational

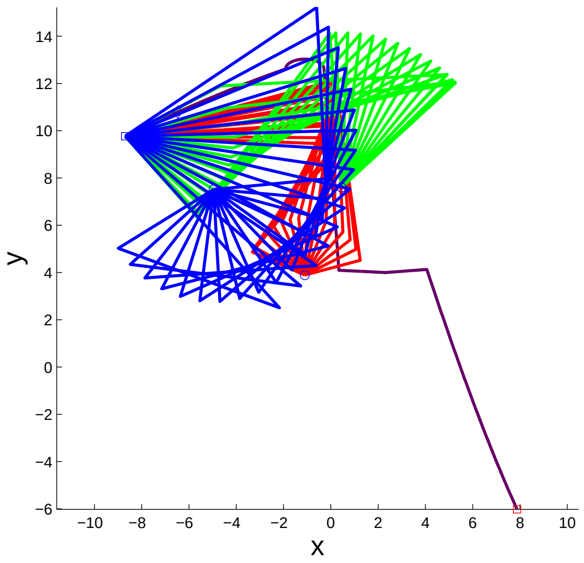


Fig. 15. All legs use the Trinkle-Milgram algorithm to achieve their goal configurations with the junction A fixed.

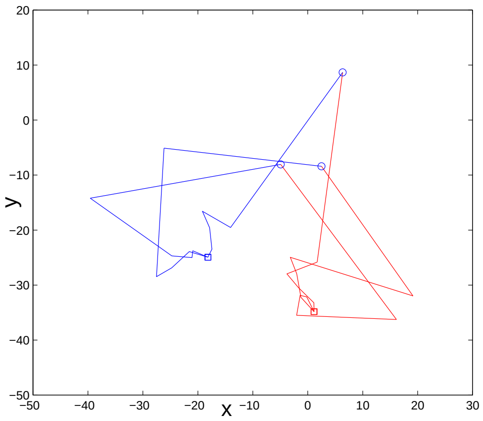


Fig. 16. Manipulator's initial configuration (drawn red, junction drawn as small square) and goal configuration (drawn blue).

Robotics: New Directions, B.R. Donald, K.M. Lynch, and D. Rus, Eds. AK Peters, Wellesley, PP. 233-246, 2001.

[29] N. Amato, B. Bayazit, L. Dale, C. Jones, and D. Vallejo, *OBPRM: An obstacle-based PRM for 3d workspaces*. in Robotics: The Algorithmic Perspective, P. Agarwal, L. Kavraki, and M. Mason, Eds. Natick, MA: A.K. Peters, 1998, PP. 156-168.

[30] V. Boor, M. Overmas, and A.F. van der Stappen, *The Gaussian sampling strategy for probabilistic roadmap planners*. IEEE International Conference on Robotics and Automation, 1999.

[31] R. Bohlin and L. Kavraki, *Path planning using lazy PRM*. IEEE International Conference on Robotics and Automation, PP. 521-528, 2000.

[32] J.J. Kuffner and S.M. LaValle, *RRT-Connect: An Efficient Approach to Single-Query Path Planning*. IEEE International Conference on Robotics and Automation, PP. 995-1001, 2000.

[33] D. Hsu, R. Kindel, J.-C. Latombe, *Randomized Kinodynamic Motion Planning with Moving Obstacles*. International Journal of Robotics Research, Vol 21, No. 3, PP. 233-255, 2002.

[34] L. Kavraki, M.N. Kolountzakis, and J.-C. Latombe, *Analysis of Probabilistic Roadmaps for Path Planning*. IEEE Transactions on Robotics and Automation, Vol. 14, PP. 166-171, 1998.

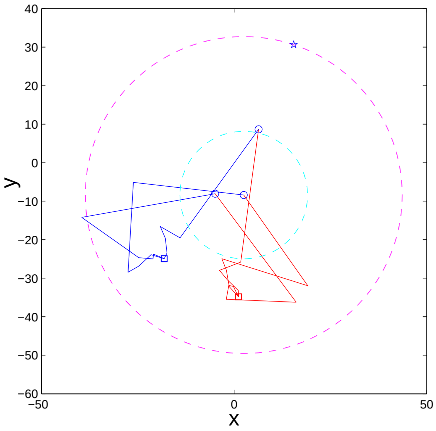


Fig. 17. A way point (drawn as star) for leg 2 (the leg with its base anchored at the center of two circles drawn as dashed lines) for adjusting the sign of this leg.

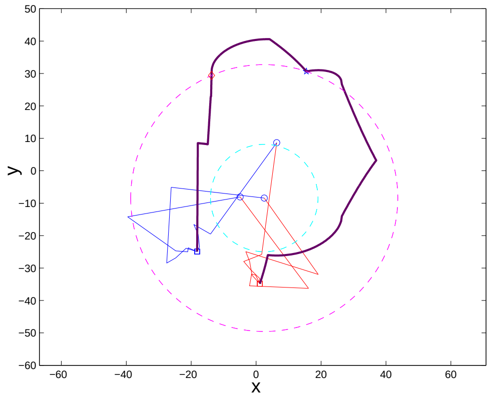


Fig. 18. A path for the junction that connects p_{init} and p_{goal} that visits the way point. The last intersection point of this path with the boundary circle of the two component cell that contains p_{goal} is the guard (drawn as diamond).

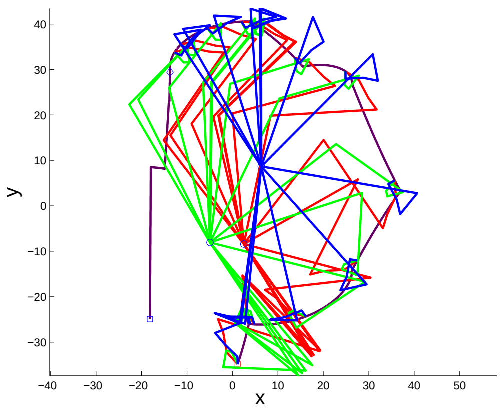


Fig. 19. All legs use an accordion move to move the junction A to the guard q_2 of leg 2.

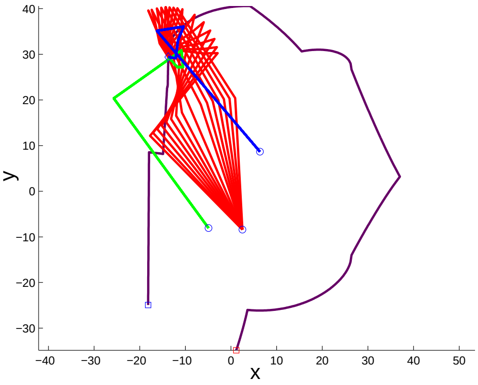


Fig. 20. With the junction A at q_2, the joint angles of leg 2 can be adjusted to achieve the signs required at the goal configuration. All other legs are fixed in place.

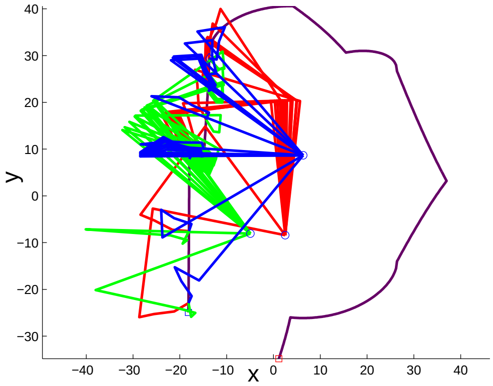


Fig. 21. All legs use an accordion move to move the junction A to its goal location. The signs of the joint angles are preserved guaranteeing that legs 2 will be in the correct C-space component once A is fixed at the goal position.

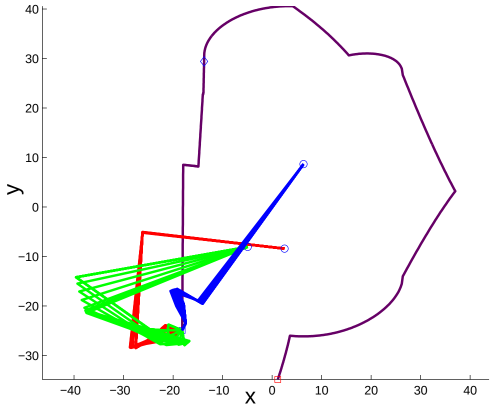


Fig. 22. All legs use the Trinkle-Milgram algorithm to achieve their goal configurations with the junction A fixed.

[35] A.M. Ladd and L.E. Kavraki, *Measure Theoretic Analysis of Probabilistic Path Planning*. IEEE Transactions on Robotics and Automation, Vol. 20, No. 2, PP. 229-242, 2004.

[36] S.M. LaValle and J.J. Kuffner, *Randomized kinodynamic planning*. IEEE International Conference on Robotics and Automation, 1999.

[37] M.S. Branicky, S.M. LaValle, K. Olson, and L.B. Yang, *Quasi-Randomized Path Planning*. IEEE International Conference on Robotics and Automation, PP. 1481-1487, 2001.

[38] J.C. Trinkle and R.J. Milgram, *Complete Path Planning for Closed Kinematic Chains with Spherical Joints*. International Jounral of Robotics Research, 21(9):773-789, December, 2002.

[39] R.J. Milgram and J.C. Trinkle, *The Geometry of Configuration Spaces for Closed Chains in Two and Three Dimensions*. Homology Homotopy and Applications, 6(1):237-267, 2004.

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

[41] M. Kapovich and J. Millson, *On the moduli spaces of polygons in the Euclidean plane*. Journal of Differential Geometry, Vol. 42, PP. 133-164, 1995.

[42] 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.

[43] J.C. Latombe, *Robot Motion PLanning*. Kluwer Academic Publishers, 1992.

[44] R.J. Milgram and J.C. Trinkle, *The Geometry of Configuration Spaces for Closed Chains in Two and Three Dimensions*. Homology Homotopy and Applications, 6(1):237-267, 2004.

[45] J.C. Trinkle and R.J. Milgram, *Complete Path Planning for Closed Kinematic Chains with Spherical Joints*. International Jounral of Robotics Research, 21(9):773-789, December, 2002.

[46] G.F. Liu, J.C. Trinkle and R.J. Milgram, *Toward Complete Path Planning for Planar 3R-Manipulators Among Point Obstacles*. In Algorithmic Foundations of Robotics VI, STAR 17, Springer-Verlag, 329-344, 2005.

[47] G.F. Liu and J.C. Trinkle, *Complete Path Planning for Planar Closed Chains Among Point Obstacles*. In Robotics: Science and Systems, MIT Press, 2005.

[48] J. Yakey, S. M. LaValle, and L. E. Kavraki, *Randomized path planning for linkages with closed kinematic chains*. IEEE Transactions on Robotics and Automation, 17(6):951–958, December 2001.