Motion Planning for Planar n-Bar Mechanisms with Revolute Joints*

J.C. Trinkle
Intelligent Systems Principles Department
Sandia National Laboratory
Albuquerque, NM 87185-1004 U.S.A.
Email: jctrink@sandia.gov

R. James Milgram†
Department of Mathematics
Stanford University
Stanford, CA 94305 U.S.A.
Email: milgram@math.stanford.edu

Abstract

Maximizing the use of dual-arm robotic systems requires the development of planning algorithms analogous to those available for single-arm operations. In this paper, the global properties of the configuration spaces of planar n-bar mechanisms (*i.e.,* kinematic chains forming a single closed loop) are used to design a complete motion planning algorithm. Numerical experiments demonstrate the algorithm's superiority over a typical algorithm that uses only local geometric information.

1 Introduction

Today, software packages can efficiently compute the solutions to arbitrary forward dynamics and kinematics problems for general planar and spatial mechanisms with multiple kinematic loops (*e.g.,* ADAMS and Working Model). That is, given smooth driving inputs for a set of independent joints, the position, velocity, acceleration, and reaction force histories for the joints can be computed by methods based on numerical algorithms for solving differential algebraic equations (DAEs) [1, 2, 7]. The value of such forward analysis is that it provides the designer with detailed information about how a given design will perform under a range of expected operating conditions. If the performance is not acceptable, the designer might use multiple forward analyses to guide iterative modifications of the design.

Inverse problems are those in which one must find a driving input that can move a linkage through a sequence of configurations while satisfying various conditions on the positions, velocities, accelerations, and/or reaction forces. Generalizing inverse problems by expanding the search space to include kinematic and/or dynamic parameters leads to design problems. While one may argue that there is greater potential pay-off in solving inverse and design problems than forward problems, software for the former is lagging significantly behind. Part of the reason for this lag is the current lack of understanding of the global mathematical structure of the space of all configurations (C-space) for an arbitrary mechanism and the space of all C-spaces generated by varying the kinematic and dynamic design parameters, a so-called, *moduli space.*

Probing further, the solution to a forward problem requires no global information. One is given the model of the linkage and the driving input function, and a DAE method computes the solution using local (derivative) information. By contrast, if the objective is to find a continuous path connecting a pair of configurations of an n-bar mechanism (an inverse problem), then one must first be able to answer the question, "Does such a path exist?" This question cannot be answered without knowledge of the global properties of the C-space. In the case of planar four-bar mechanisms, this kind of information has been available for some time [3]. However, for planar n-bars with $n > 4$, previous results [4, 8, 9] are less complete. While the path existence question may seem unimportant, given the designer's knowledge and intuition about mechanisms and their applications, it is far from trivial when trying to build software to automate the design process. Further, it may be possible to achieve superior performance from designs outside a designer's normal range of design options (*i.e.,* from portions of the design space he would never consider). The encoding of global information in interactive design software will enable the designer to painlessly explore non-standard designs.

This paper takes a step toward solving general inverse and design problems and thus toward enabling enhanced computer-aided design software. It does so by building on recent results on the global properties of

*The research described herein was supported by the Research Foundations Program at Sandia National Laboratories. Sandia is a multi-program laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000

†The work of this author was also partially supported by the Nation Science Foundation.

the C-spaces of planar n-bar mechanisms [6]. The specific result presented here is a *complete*[1] algorithm for solving motion planning problems for arbitrary planar n-bar mechanisms with a single kinematic loop. Joint limits and collisions between links and obstacles are not considered.

Motion Planning Problem:
Given two valid configurations, S and G, of a planar n-bar mechanism with all revolute joints, determine if a continuous path connecting S and G exists, and if so, determine one.

The new theory and algorithm presented here, help to extend our fundamental understanding of the kinematics of planar n-bar mechanisms and clearly have applications in machine design. However, other important applications are in the area of robot manipulation planning. In robotics it is of interest to plan coordinated movements of two arms collaborating to move and manipulate an object too large or heavy to be handled safely by a single arm. While holding the object, the arms form a closed loop - an n-bar mechanism. Planning to move the object from one position to another is a motion planning problem. If one knew an acceptable motion for the held object, then inverse kinematics could be used to determine compatible motions of the arms. However, there are two reasons that this approach is not desirable. First, it requires that the individual arms have 6 or more degrees of freedom, which may not be required for successful task execution. Second, choosing a collision-free motion for the held object does not guarantee a valid motion for the arms; the motion could force collisions, joint limit violations, and/or near-singular configurations. The problem should be solved in its fully coupled form, not in an arbitrarily partitioned form.

As a preview of our results, Figure 1 shows two motion planning problems for two six-bars with the same base link (bold and horizontal), but with slightly different link lengths. The start and goal configurations are shown in bold and thin lines, respectively. While not intuitively obvious, only one of these two problems has a solution, which our algorithm (implemented in Matlab on a 366 MHz Pentium II PC) found in less than 0.1 cpu seconds. This same algorithm has been successful in solving motion planning problems for n-bars with up to 10,000 links for one class of n-bars, but only up to about 40 links for the other class. The reason for this disparity will become clear later.

[1] By "complete" it is meant that if a path exists, our algorithm will find one in finite time, and if not, it will report that this is the case, also in finite time.

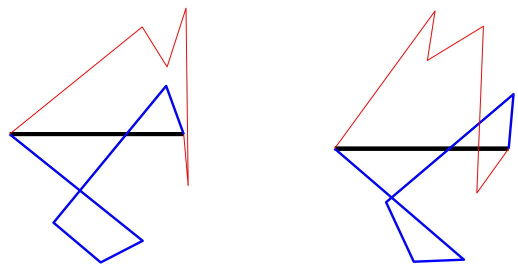

Figure 1: Start and goal configurations of two similar six-bars. For one no connecting path exists.

2 Configuration Spaces of Planar n-Bars

To develop provably maximally efficient algorithms for solving inverse problems for planar n-bars, one must completely understand the structure and properties of their C-spaces and know how to properly parameterize them.

2.1 C-spaces of Planar Four-Bars

The global structure of C-space for any four-bar mechanism is characterized by the sum of the longest and shortest links relative to the sum of the other two [3]. Letting L_i denote the i^{th} longest link, then the Grashof conditions for a planar four-bar are:

$$L_1 + L_4 < L_2 + L_3 \tag{1}$$

The four-bar shown in Figure 2, satisfies the Grashof conditions with $L_1 = l_2$ and $L_4 = l_4$. Two well-known properties of Grashofian four-bars are, first, the shortest link is a crank (*i.e.*, it can rotate 2π radians) relative to the other links. Second, when the linkage is in an "elbow-up" ("elbow-down") configuration, it cannot reach an "elbow-down" ("elbow-up") configuration without breaking at least one joint connection.[2] These properties are direct consequences of the fact that the structure of C-space is two disconnected (topological) circles [4].

When the sum of the longest and shortest ($L_1 = l_1$ and $L_4 = l_4$ in Figure 3) is greater than the sum of the other two, a four-bar is referred to as non-Grashofian. In this case, C-space is topologically equivalent to a single circle, and thus, all configurations are connected. However, some non-Grashofian four-bars do not have cranks (for example, the linkage in Figure 3). This

[2] "Elbow-up" ("elbow-down") configurations as those for which l_2 and l_3 lie above (below) the dashed line λ passing though the joint axes shown in Figure 2.

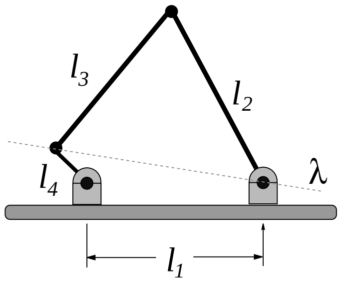

Figure 2: Grashofian planar four-bar.

means that when planning smooth motions connecting pairs of valid configurations, one cannot parameterize all motions by a single joint angle. Referring to Figure 3, suppose one were planning to move from the "elbow-up" configuration shown to the corresponding "elbow-down" configuration. Any path to accomplishing this task requires ϕ_4 (the angle of l_4 relative to the base link) to reach an extreme value along the way. Since l_2 and l_3 are colinear for extreme values of ϕ_4, ϕ_4 cannot be used at those values to control the linkage to obtain an "elbow-down" configuration.

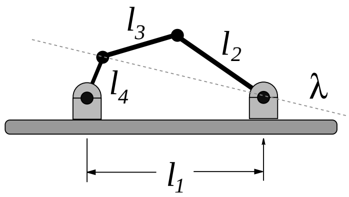

Figure 3: Non-Grashofian planar four-bar.

2.2 C-spaces of Planar n-Bars

Recognizing that scaling the link lengths uniformly does not change the global topological properties, it will be assumed henceforth, that the link lengths have been normalized to sum to 1. In previous work by Kapovich and Millson [4], it was shown that the properties of the C-space of a planar n-bar depend strongly on the number of *long links.*

Definition 1 *A subset $\mathcal{L}$ of the links is referred to as the "long links" if and only if the sum of the lengths of any two distinct links in $\mathcal{L}$ is greater than $\frac{1}{2}$.*

Since $\mathcal{L}$ is not unique, the number of long links is defined as the maximum cardinality of all possible $\mathcal{L}$. However, when there are three long links, $\mathcal{L}$ is unique.

Three conclusions applicable to the set of all n-bar mechanisms are immediately obvious:

1. By definition, there cannot be only one long link.
2. If an n-bar has a link longer than 0.5, then its C-space is empty.
3. There can be no more than three long links.

Thus the number of long links can be only zero, two, or three.

The following two less obvious Theorems by Kapovich and Millson can be viewed as a generalization of the Grashof/non-Grashof designation:

Theorem 1 *C-space is connected if and only if the mechanism does not have three long links.*

Theorem 2 *If there are three long links, then C-space has exactly two components and each component is a torus $(S^1)^{n-3}$.*

The implication of the toroidal structure of C-space in Theorem 2 is that for n-bars with three long links, all the other links (the short links) are independent cranks. That is, each of these links can rotate through a full circle regardless of the orientations of the other short links, and the long links can comply to maintain the closed-loop structure of the mechanism. In motion planning, as long as the start and goal configurations are in the same C-space component, one can drive the mechanism from the start to goal configuration by any continuous path connecting the start and goal angles of the short links. No such path exists that will violate the closed-loop constraint of the mechanism. It was our exploitation of this theorem that allowed us to generate a path for a 10,000-bar mechanism in several seconds in Matlab.

While the results of Kapovich and Millson provide important global information about the C-spaces of all planar n-bars, the development of our complete motion planning algorithm was enabled by the following two theorems first presented and proved in [6] (and specialized to the planar case here):

Theorem 3 *Let $\mathcal{N}$ be an n-bar with given lengths, $l_1, \ldots, l_n$ Then:*

a) Except for a finite number of base lengths, l_1, the C-space of $\mathcal{N}$ is a closed compact manifold of dimension $n-3$.

b) Whenever the C-space is a manifold, it is the boundary of a manifold of dimension $n-2$, which is given as the union of sub-manifolds of the form

$$(S^1)^s \times I^{(n-s-2)}$$

where S^1 is the (topological) circle and the set of s that occur depend on the link lengths.

Figure 4 shows a two-dimensional manifold, whose boundary (at the left side) is the C-space of a four-bar with l_1 a bit less than $l_2 + l_3 - l_4$. The C-space can be seen to be $S^1 \bigcup S^1$, which is the boundary of the "thickened circle" (generalized cylinder).

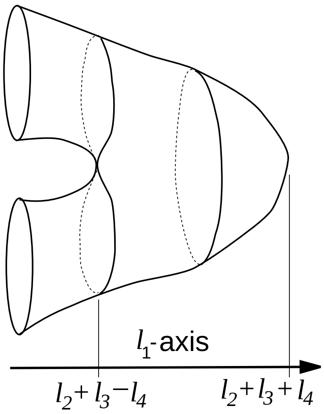

Figure 4: Two-dimensional manifold of four-bar C-spaces parameterized by the length of the base link, l_1.

Theorem 4 *If there are exactly two long links, then C-space is the double along the boundary of a thickening of the type of a union of* $(S^1)^s \times I^{n-s-2}$.

Here, "double along the boundary" means that the manifold is duplicated in place, and connected to the original manifold only along the boundary. The disconnected surfaces can then be separated to yield the new manifold composed of "mirror image" copies on either side of the original boundary. In the case of a four-bar, the initial manifold is an arc whose boundary is the two end points. The double along the boundary is a circle.

3 Complete Motion Planning Algorithm

We have developed a complete motion planning algorithm, which is divided into two cases based on the number of components of C-space - one or two. Given a normalized set of link lengths, C-space is a connected manifold if and only if the sum of the second and third longest links is less than 0.5 (*i.e.*, $L_2+L_3 < 0.5$). If and only if the sense of the inequality is reversed, C-space is the union of two disconnected toroidal manifolds. If the inequality is replaced by equality, C-space is not a manifold.

3.1 Case 1: C-Space is not Connected

With two components, one must first be sure that the start and goal configurations are in the same component of C-space. Figure 5 shows a typical linkage with three long links in configurations belonging to different components. Regardless of how one stretches out the short links, the long links cannot move from "elbow-up" on the left, to "elbow-down" on the right. Therefore, membership in a particular component can be checked by simply determining the sign of the angle between any pair of long links.

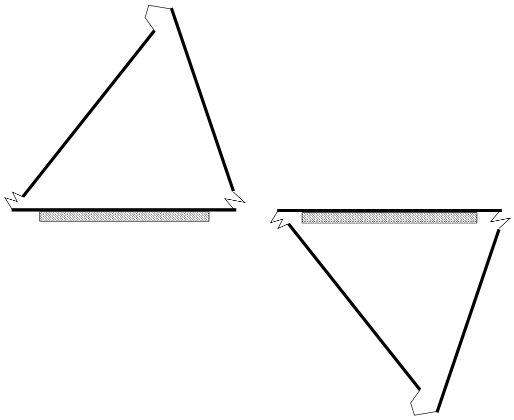

Figure 5: A pair of disconnected configurations of an n-bar with three long links.

If the start and goal configurations are in different components, then a connecting path does not exist. Otherwise, an infinite number of trajectories exist and planning is a simple matter of generating any continuous connecting path on the surface of the hyper-torus corresponding to the angles of the short links. In our implementation, the path is computed by linear interpolation (*i.e.*, we follow a geodesic of the hyper-torus).

3.2 Case 2: C-Space is Connected

When C-space is connected, planning is more difficult. The complexity is proportional to the number of sub-tori contained in C-space. This number, in turn, is roughly inversely proportional to number of distinct "critical radii," r_j, of the linkage, defined as follows:

$$r_j = \sum_{i=1}^{n-1} \sigma_i l_i \tag{2}$$

where $\sigma_i = \pm 1$. Since the number of critical radii is generally exponential in the number of links, it is not practical to construct the full information, that would be needed for planning "optimal" trajectories. An exception is when the number of distinct link lengths is restricted (as would be the case in modular robotic systems). For example, if there were only two distinct link lengths, then the number of distinct types of C-spaces would be proportional to n^2, and they are easy to construct.

For the remainder of this paper, it will be assumed that link lengths are arbitrary, and that "optimal" algorithms are impractical. The basis of our algorithm is a procedure we refer to as *accordion-move*. The link connected to the left end of the base, l_{left}, is moved into its correct final orientation with respect to the base, l_{base}, while the remaining links comply (see Figure 6). Next, the angle between l_{left} and l_{base}

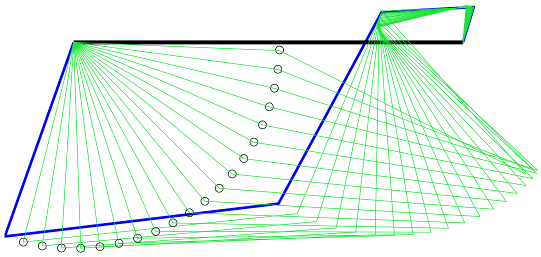

Figure 6: An *accordion-move* for a 6-bar.

is fixed, effectively fusing them into a single link and reducing the overall link count by one. Figure 7 shows an *accordion-move* of the five-bar resulting from the *accordion-move* shown in Figure 6. Note that since kinematic loop closure is independent of link order, for convenience, our implementation chooses the longest link as the base while processing each *accordion-move*.

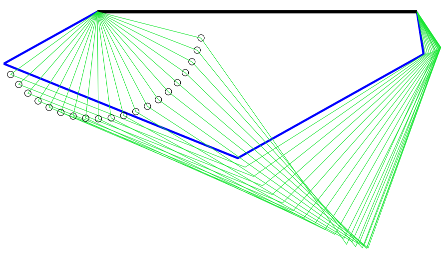

Figure 7: An *accordion-move* for the resulting 5-bar.

Our algorithm begins with the starting configuration and performs *accordion-moves* until one of the following special states is obtained:

1. The linkage is a four-bar with a connected C-space.

2. The linkage has two components and the current and goal configurations are in the same component.

3. The linkage has two components and the current and goal configurations are in different components.

Upon reaching the first state, a complete solution is at hand, since algorithms exist to plan motions of arbitrary four-bars with connected C-spaces. Arrival in the second state implies that the remainder of the path can be completed by linear interpolation as discussed above. If, however, the algorithm obtains the third state, one can achieve the second state (and hence, a solution) with one carefully chosen accordion move. Since prior to the last *accordion-move* the C-space of the linkage was connected, the last *accordion-move* was responsible for disconnecting the C-space and placing the current and goal configurations in different components. However, one can show that a more carefully chosen *accordion-move* can be designed that will disconnect C-space in a way that puts the current and goal configurations in the same component (see [10] for details).

The algorithm is summarized below under the assumption that a path exists:

```
0. Given: start and goal states, S and G.
1. Perform an {accordion-move}.
2. Test for arrival in special state.
3. If 1st special state, then complete with
    one {4-bar-move}.  DONE.
4. Else if 2nd special state, then complete
    with linear interpolation.  DONE.
5. Else if 3rd special state, then backtrack
    and perform special {accordion-move}.
    Go to step 2.
6. Else, Go to step 1.
```

4 Numerical Results

Our algorithm was compared to a simple algorithm that used only local geometric information. The local algorithm can be thought of as a proportional controller applied to an open kinematic chain. Imagine breaking the linkage at the left end of the base link, yielding an open chain with $n-1$ revolute joints based on the right side of the fixed base link. The proportional controller generates corrections to the current configuration by comparing it with the goal configuration as follows:

$$\begin{bmatrix} \boldsymbol{J} \\ \alpha \boldsymbol{I} \end{bmatrix} d\boldsymbol{\phi} = \begin{bmatrix} \boldsymbol{x}_{desired} - \boldsymbol{x}_{current} \\ \boldsymbol{\phi}_{goal} - \boldsymbol{\phi}_{current} \end{bmatrix}. \qquad (3)$$

where $\boldsymbol{J}$ is the Jacobian matrix for position of the end point of the $(n-1)$-joint open chain, $\boldsymbol{I}$ is the identity matrix, $\boldsymbol{x}_{desired}$ is the location of the left end of the base link, $\boldsymbol{x}_{current}$ is the current location (numerically) of the left end of the base link, $\boldsymbol{\phi}_{goal}$ is goal configuration, of the linkage, $\boldsymbol{\phi}_{current}$ is the current configuration of the linkage, and α is a scalar weight that determines the relative importance of maintaining the loop closure constraint and attracting the mechanism to its goal configuration.

A number of tests were run to verify the correctness of our algorithm and to compare its performance with the local algorithm just described. For linkages with from 4-16 links, link lengths and start and goal configurations were chosen at random. Therefore, planning problems with connected and disconnected C-spaces were attempted. Since the complete algorithm was able to recognize solution existence and then find a solution whenever one existed, the table only contains the results of the local algorithm.

numLinks	numExist	numLocal
4	102	59
5	160	67
6	192	87
8	199	107
10	200	157
12	200	171
16	200	191

For each number of links considered (left-most column), 200 random problems were generated. The second column shows the number of problems for which solutions existed and the last column shows the number of problems solved by the local method. Notice that as the number of links increased, the probability of success of the local path planner increased. For problems with over about 30 links, the local method solved every problem we generated, but it is guaranteed that the local method (and all other purely local methods) will fail for certain problems even when solutions exists for large numbers of links.

Figure 8 shows a problem drawn from our test set for which the local algorithm failed. While the local method generated a smooth motion, it failed to connect the start and goal configurations. Figure 9 shows the individual joint angle trajectories not achieving the goal angle targets (the circles on the right side of the plot). On the other hand, the complete algorithm found a solution with two *accordion-moves* and one *four-bar-move* (see Figure 10). The slope discontinuities at the 20 and 40 steps point indicate the ends of the *accordion-moves*.

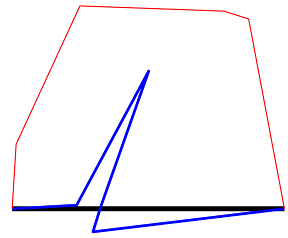

Figure 8: Planning problem with solution not found by the local algorithm.

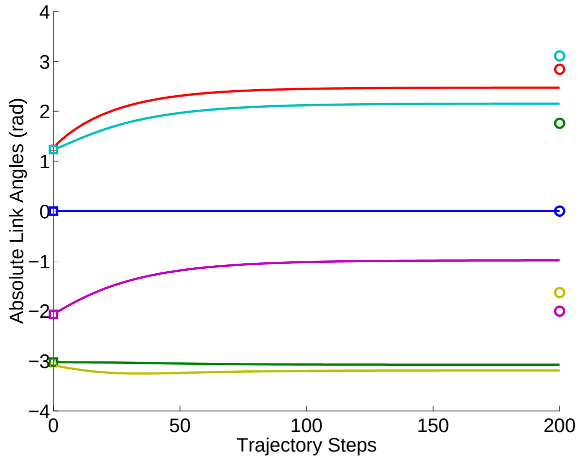

Figure 9: Faulty trajectory found by the local algorithm.

5 Conclusion

The motion planning problem discussed here appears at the outset to be a simple motion planning problem restricted to a smooth manifold described by holonomic constraints [5]. However, such planning problems are not easily solved without knowledge of the topology of the configuration space. The power of the this knowledge was demonstrated through the development of a complete algorithm for the planar, n-bars with all revolute joints. It is able to solve all problems with solutions, but is slower and produces less smooth trajectories than typical local algorithms. While not exhaustive, our numerical experiments suggested that the complete and local algorithms be used in tandem. The local algorithm should be applied first on large problems (more than about 15 links). Then, if it fails, one can call the global algorithm.

While our algorithm works in its current basic form, there are two primary improvements that we plan to make: 1) extending *accordion-move* to bringing more than one link into its goal orientation in a single move; and 2) enhancing link selection in *accordion-move* to obtain a two-component C-space more rapidly.

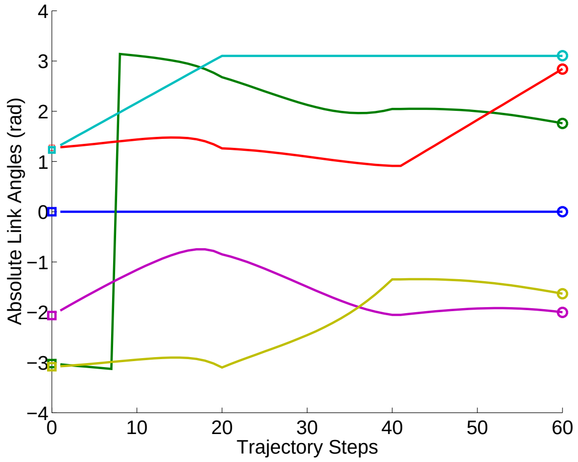

Figure 10: Trajectory found by our complete algorithm.

Acknowledgement

The authors would like to thank Tony Maciejewski for motivating the comparison of our complete algorithm to local ones.

References

[1] R.M. Ascher and L.R. Petzold. *Computer Methods for Differential Equations and Differential Algebraic Equations*. SIAM, 1998.

[2] D. Baraff. Linear-time dynamics using lagrange multipliers. *Computer Graphics* (*Proc. SIGGRAPH*), pages 137–146, 1996.

[3] K.H. Hunt. *Kinematic Geometry of Mechanisms*. Oxford University Press, 1978.

[4] M. Kapovich and J. Millson. On the moduli spaces of polygons in the euclidean plane. *Journal of Differential Geometry*, 42:133–164, 1995.

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

[6] R.J. Milgram and J.C. Trinkle. The geometry of configuration spaces for closed chains in two and three dimensions. *Homology, Homotopy, and Applications*, 2001. submitted.

[7] F.R. Potra. Numerical methods for differential algebraic systems with application to real-time simulation of mechanical systems. *Zeitschrift für Angewandte Mathematik und Mechanik*, 74:177–187, 1994.

[8] K.L. Ting. Mobility criteria of single loop n-bar linkages. *Trans. ASME, Journal of Mechanisms, Transmissions and Automation in Design*, 111(4):504–507, December 1989.

[9] K.L. Ting and Y.W. Liu. Rotatability laws for n-bar kinematic chains and their proof. *ASME Journal of Mechanical Design*, 113(1):32–39, 1991.

[10] J.C. Trinkle and R.J. Milgram. The geometry of configuration spaces for closed chains in two and three dimensions. Technical report, Sandia National Laboratories. in preparation.