Title
Automated Generation of Linkage Loop Equations for Planar One Degree-of-Freedom Linkages, Demonstrated up to 8-Bar

Journal
Journal of Mechanisms and Robotics, 7(1)

ISSN
1942-4302

Authors
Parrish, Brian E
McCarthy, J Michael
Eppstein, David

Publication Date
2015-02-01

DOI
10.1115/1.4029306

Peer reviewed

Automated Generation of Linkage Loop Equations for Planar One Degree-of-Freedom Linkages, Demonstrated up to 8-Bar

Brian E. Parrish[1]
Robotics and Automation Laboratory,
Department of Mechanical and
Aerospace Engineering,
University of California,
Irvine, CA 92697
e-mail: bparrish@uci.edu

J. Michael McCarthy
Robotics and Automation Laboratory,
Department of Mechanical and
Aerospace Engineering,
University of California,
Irvine, CA 92697
e-mail: jmmccart@uci.edu

David Eppstein
Information and Computer Sciences,
Department of Computer Science,
University of California,
Irvine, CA 92697
e-mail: eppstein@ics.uci.edu

In this paper, we present an algorithm that automatically creates the linkage loop equations for planar one degree of freedom, 1DOF, linkages of any topology with revolute joints, demonstrated up to 8 bar. The algorithm derives the linkage loop equations from the linkage adjacency graph by establishing a rooted cycle basis through a single common edge. Divergent and convergent loops are identified and used to establish the fixed angles of the ternary and higher links. Results demonstrate the automated generation of the linkage loop equations for the nine unique 6-bar linkages with ground-connected inputs that can be constructed from the five distinct 6-bar mechanisms, Watt I–II and Stephenson I–III. Results also automatically produced the loop equations for all 153 unique linkages with a ground-connected input that can be constructed from the 71 distinct 8-bar mechanisms. The resulting loop equations enable the automatic derivation of the Dixon determinant for linkage kinematic analysis of the position of every possible assembly configuration. The loop equations also enable the automatic derivation of the Jacobian for singularity evaluation and tracking of a particular assembly configuration over the desired range of input angles. The methodology provides the foundation for the automated configuration analysis of every topology and every assembly configuration of 1DOF linkages with revolute joints up to 8 bar. The methodology also provides a foundation for automated configuration analysis of 10-bar and higher linkages. [DOI: 10.1115/1.4029306]

Introduction

Dimensional synthesis solves for the geometric features of a linkage so that it is capable of moving the end-effector to each of a given set of positional requirements. One type of synthesis, task generation, solves the geometry to meet a set of task positions that are specified by their global positions and angles, Fig. 1. Synthesis does not guarantee that the linkage will move smoothly, meaning continuously with an increasing input angle, through all angles between the task positions.

A kinematic chain is an assembly of rigid bodies, links, connected by joints. The topology of a kinematic chain is the specific interconnection of the links and can be represented by either an adjacency matrix or an adjacency graph. A mechanism, sometimes called an inversion, is a kinematic chain of a specific topology where a link has been selected as ground. A linkage is a mechanism with a particular link selected as the input link; therefore, a linkage is a specific topology of a kinematic chain where one link is selected as ground and another link is selected as the input. The scope of this paper is limited to input links that are connected to ground.

The motivation for this paper is to automate the complete configuration analysis of a synthesized linkage to ensure continuous smooth movement through all input angles within the range of interest. In this paper, we present the first key step, an algorithm that automatically constructs the linkage loop equations for any topology of planar 1DOF linkage with revolute joints up to 8 bar.

To complete the automation of the configuration analysis, the Dixon determinant can be derived from the loop equations and used to solve for all possible assembly configurations. Continuous smooth movement can be determined from the linkage loop equations by deriving and solving the Jacobian to identify singularities for the configuration of interest within the range of desired input angles. The form of the automatically constructed linkage loop equations is sufficient for automated derivation of both the Dixon determinant and the Jacobian, but the details of that automation are beyond the scope of this paper.

This algorithm has been verified on all 4-, 6-, and 8-bar topologies for all unique combinations of the ground link and the ground-connected input link. Limiting the research to the 8-bar kinematic chains provided a convenient bounded scope. The approach is general; therefore, it also forms a basis for automating the analysis of 1DOF kinematic chains of higher link counts. To test the algorithm beyond the demonstrated 8-bar family, the present algorithm

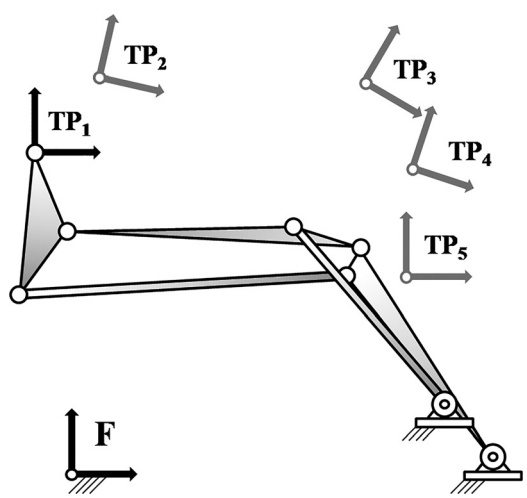

Fig. 1 Example Watt I linkage reaching a task position

[1]Corresponding author.

Manuscript received September 26, 2014; final manuscript received November 24, 2014; published online December 31, 2014. Assoc. Editor: Thomas Sugar.

successfully automated four 10-bar topologies including two with nonplanar graphs and one with a quintenary link.

Literature Review

The specific connectivity of a kinematic chain can be represented as an adjacency graph where the vertices represent links and the edges represent joints between the links. Tsai [1] published an atlas of the 16 1DOF 8-bar ten-joint kinematic chains and represented them in a set of linkage adjacency graphs and linkage sketches.

Much of the work today is enumerating the unique kinematic chains with high link counts. A key component of the enumeration process is the detection and elimination of isomorphic kinematic chains. Isomorphic kinematic chains are not unique because they have topologies that can be transformed into a topology that has already been enumerated by simply renumbering the vertices. Sunkari and Schmidt [2] apply a McKay-type algorithm [3] to show that there are approximately 20×10^6 nonisomorphic topologies for the 16 bar, 22 joint, kinematic chain. Ding and Huang [4] established a canonical representation of the kinematic chain adjacency graphs and published a method for isomorphism detection based on the largest perimeter loop and the degrees of the vertices. Ding et al. [5] published the enumeration of graphs of kinematic chains up to 14 bar and recently published work extending the development of linkage graphs to linkages that contain multiple joints, joints connecting more than two links on a common axis [6].

Tuttle [7] determined the number of distinct inversions, mechanisms, of the 1, 2, and 3DOF kinematic chains. The results show that for 1DOF linkages there are five distinct 6-bar mechanisms and 71 distinct 8-bar mechanisms.

Using Baranov trusses, Manolescu [8] identifies three distinct Stephenson 6-bar mechanisms and the two distinct Watt 6-bar mechanisms, total of five, as well as 19 unique linkages. Of those 19 unique linkages, nine have a ground-connected input. Verho [9] allows actuation through link pairs that are not grounded and identifies 25 unique 6-bar linkages using Assur groups and visual inspection. Of those 25 linkages, nine have a ground-connected input and match the nine identified by Manolescu.

Linkage synthesis solves for the specific dimensions of the links using one of the enumerated topologies. Soh and McCarthy [10] published a methodology specific to the 8-bar family for synthesizing linkages that can be constructed from a pair of constrained 3-R chains. Linkage synthesis approaches are published for a variety of selected kinematic chain topologies [11–13].

Linkage configuration analysis solves for the angles of all of the output links. Approaches are typically shown for specific topologies. Wampler [14] referencing Dixon [15] analyzed a double butterfly 8-bar linkage using the Dixon determinant in a complex plane formulation. The same linkage was also evaluated in rational formulation by Nielsen and Roth [16]. Various other methods for solving the configuration of a linkage have been published such as the Gröbner–Sylvester method by Dhingra et al. [17] and the linear relaxation method by Porta et al. [18].

To determine if a particular assembly configuration moves smoothly through a range of input angles, the singular configurations must be avoided. Linkages that encounter a singularity within the range of motion of interest have either a branching defect or a circuit defect and have been addressed by several authors. As defined by Chase and Mirth [19] a circuit is all possible orientations of the links which can be realized without disconnecting any of the joints. A branch is the continuous series of positions of the mechanism on the circuit between two stationary configurations. Chase and Mirth use the sign of the determinant of the Jacobian to identify the branches of 6-bar linkages. Myszka et al. [20] identify the singularities and plot the singular configurations as curves that are a function of the length of one of the links.

Kecskeméthy et al. [21] published work automating the generation of the equations of motion of multibody systems. The method establishes a minimal cycle basis for the mechanism graph, generates local dynamics solutions for each mechanism loop, and then combines the local dynamics solutions into a global solution.

A general method for automating the derivation of loop equations for all topologies of planar 1DOF linkages has not been published.

Demonstrated up to the 8-bar family our contribution shows a method to automatically produce the linkage loop equations for any 1DOF linkage with revolute joints. The loop equations are in a form sufficient to complete the automation of the linkage configuration analysis by enabling automated derivation of the Dixon determinant and the Jacobian.

Linkage Graph and Adjacency Matrix

The number of joints in a planar 1DOF linkage is given as a function of the number of links by the equation,

$$j = 3n/2 - 2 \tag{1}$$

where j is the number of joints and n is the number of links.

The number of independent loops is given by

$$L = j - n + 1 \tag{2}$$

where L is the number of independent loops, j is the number of joints, and n is the number of links.

Beginning with the 4-bar kinematic chain, the planar 1DOF kinematic chains are comprised of loops of links. The 8-bar 1DOF kinematic chains exist in three link assortment families that are categorized by the quantity of links that are binary, ternary, quaternary, quintenary, etc. Binary links connect to two adjacent links, ternary links connect to three adjacent links, etc. For example, the 4400 link assortment family is comprised of 8-bar kinematic chains with four binary and four ternary links. The planar 1DOF link assortments and the quantity of topologies for each are listed in Table 1 up to 8-bar kinematic chains, Tsai [1]. Quintenary links appear in some 10-bar kinematic chains.

For planar 1DOF linkages with revolute joints, the number of connections between two links is limited to one since a revolute joint only allows 1DOF. Having two joints between the same two links forms a rigid structure.

Each topology can be represented in the form of an adjacency matrix where a "1" indicates a joint connecting two links and a "0" indicates no connection between the two links. The topology can also be represented as an adjacency graph where the vertices represent links and the edges represent joints connecting the links. For the double butterfly topology, the adjacency matrix and a nonplanar adjacency graph are shown in Fig. 2 along with a sketch of the linkage.

Rooted Cycle Basis

Before constructing the linkage loop equations, the loops must be identified. To do so, we use the adjacency graph to establish the smallest cycle basis through a common root edge, the edge connecting the ground vertex to the input vertex. We call this the rooted cycle basis.

Table 1 Planar 1DOF kinematic chain topologies

Class			Link assortment				Topology	
Loops	n	j	n_2	n_3	n_4	n_5	Quantity	Total
1	4	4	4	0	0	0	1	1
2	6	7	4	2	0	0	2	2
3	8	10	4	4	0	0	9	16
			5	2	1	0	5	
			6	0	2	0	2	

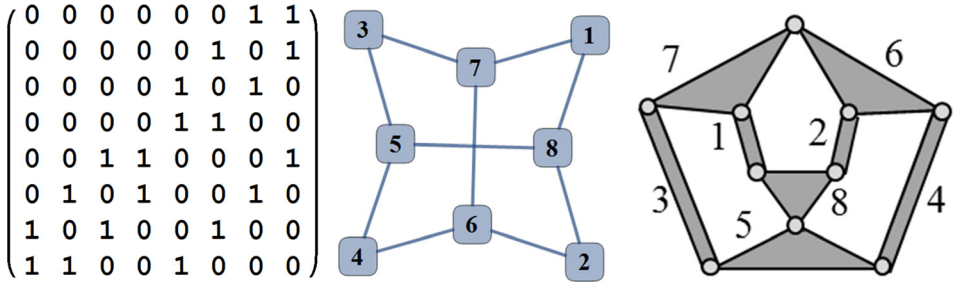

Fig. 2 Double butterfly 8-bar adjacency matrix, graph, and linkage sketch

A cycle basis is a set of independent cycles that form a basis for the graph such that every other cycle can be formed by a linear combination of the basis cycles. Our automation requires every loop of the cycle basis to pass through the edge connecting the ground vertex to the input vertex.

The graphs of planar 1DOF linkages are in the family of graphs called two-vertex connected. To divide the graph into two separate graphs, two vertices must be removed. Removing one vertex leaves a connected graph. Two-vertex connected graphs have the property that the graph can be decomposed into a set of ears, an ear decomposition. Per Whitney [22] any nonseparable graph based on a loop, or circuit, remains a nonseparable graph with the addition of ears, also called "suspended chains." The first ear is a loop, or cycle. The second ear, significantly higher, is a simple path that has only the end joints in common with the previous loops. An independent loop can be obtained by following the ear and the previous loops to a common vertex. Similarly, an open ear decomposition starts with a single edge and adds ears to form the graph.

A rooted cycle basis can be derived from an open ear decomposition by constructing a path that follows the edges of each ear to the endpoints of that ear, then from each endpoint follows part of the next lowest ear to its end points. This repeats until arriving at the endpoints of the edge connecting the ground and input vertex. Since each cycle formed in this way contains an ear that is not part of the previous cycles, each cycle contains at least one edge that is not part of the previous cycles. Therefore, the set of cycles found by this method is independent. The quantity of cycles constructed in this way is the same as the number of independent loops necessary to form a cycle basis. Since the cycles are independent and of the proper count to form a cycle basis, a set of cycles formed in this manner is a valid cycle basis that meets the additional constraint that all cycles in the basis pass through the common edge connecting the ground and input vertices.

Deriving the Rooted Cycle Basis

To find the rooted cycle basis, we do not find the ear decomposition directly, rather we find the loops directly. The list of vertices along a loop from the input vertex to the ground vertex is called a path and the length of the path is its number of vertices. We use a shortest path algorithm to find loops that are of minimal length.

There are several means of identifying the shortest path between two vertices. Because all of the edges in these graphs represent joints they all have a positive distance, therefore, Dijkstra's shortest path algorithm [23] is sufficient for finding the shortest distance between two vertices. Mathematica contains a shortest path function that can be constrained to use the Dijkstra algorithm, but the default settings are also suitable.

To generate the loops, the list of edges is produced for the entire graph and the root edge, the edge connecting the ground vertex to the input vertex, is eliminated. Through the remaining portion of the linkage, the shortest path back to ground is identified and forms the first path. To find more independent paths, an edge in the first path is eliminated along with the root edge. Each edge in the first path is eliminated one at a time to find one or more independent paths. These paths can be visualized as being at the same level, the second level of paths. To find the third level of paths, an edge in a second level path is eliminated along with all the edge eliminations that created that second level path. This process continues until no new paths are found and every edge elimination has been attempted.

At any of these levels, if there are multiple paths of equal length one of them may be selected arbitrarily. At the next level, the path not selected will still be the shortest path; therefore, it will not be overlooked.

Because these are two-vertex connected graphs, elimination of two edges can separate the graph into two components such that there are no paths from the input vertex back to ground. When this occurs the elimination is not valid and the algorithm tests the elimination of the next edge.

From all of the paths identified, all of the unique paths are collected and formed into cycles by appending the root edge. The direction along each cycle is consistently defined such that the first vertex in each cycle is ground, the second vertex is input, and the last vertex is ground.

Some of the cycles may not be independent. An independent cycle will contain an edge that is not in any of the previous cycles. The cycles are ordered by length and then by vertex degree along the cycles. The smallest cycle is chosen as the first cycle in the cycle basis. In the sorted list of cycles, the next cycle which contains the fewest new edges not presently in the cycle basis is an independent cycle and is added to the cycle basis. This continues until a full set of independent cycles have been selected. The 8-bar family has three independent cycles.

Example Automated Cycle Basis Derivation

We apply the automation to an example 8-bar linkage from one of the nine topologies in the 4400 link assortment family, Table 1. The adjacency graph and adjacency matrix are shown in Fig. 3. Vertex 5 is the ground and vertex 2 is the input.

The first elimination is the root edge connecting ground to input (5–2), and the first level shortest path for the example 8-bar is shown in Fig. 4.

To find the shortest paths at the second level, each of the edges along the first shortest path is eliminated one at a time along with the edge connecting ground to input. The next shortest path back to ground is found through the rest of the linkage. Elimination of

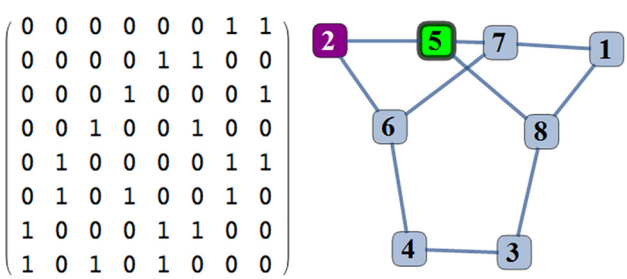

Fig. 3 Example 8-bar adjacency matrix and adjacency graph

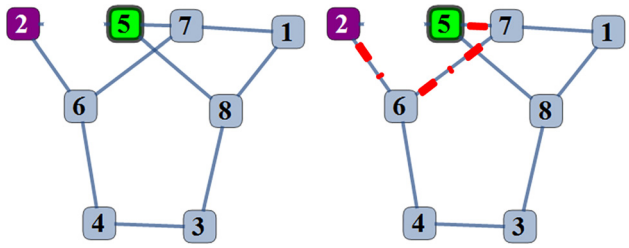

Fig. 4 The first level shortest path

the edge set (5–2) and (2–6) will eliminate all connections to the input vertex and is not valid. To produce the second shortest path, the first valid edge elimination set is (5–2) and (6–7), Fig. 5.

There remains one more edge to eliminate in the first loop, the elimination of the edge set (5–2) and (7–5). This elimination produces two shortest paths of equal length, Fig. 6. Either path can be selected for the next shortest path and in this example we select the first one. Although this is the same path as previously found, the other path will still be the shortest path during a later step and will not get overlooked.

To find the third level paths, an edge in each of the second level paths is eliminated along with the edge eliminations that formed the second level path. The first valid elimination set is the edges (5–2), (6–7), and (8–5). This elimination produces the shortest path shown in Fig. 7.

The last unique third level path is found by eliminating the edge set (5–2), (7–5), and (6–4). This elimination produces the shortest path shown in Fig. 8.

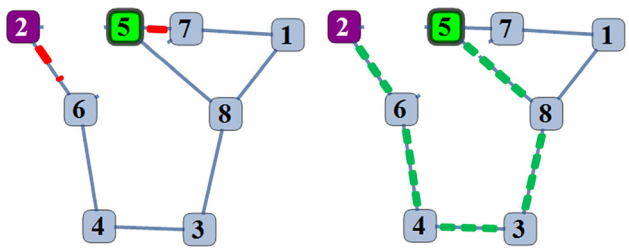

Fig. 5 A second level shortest path

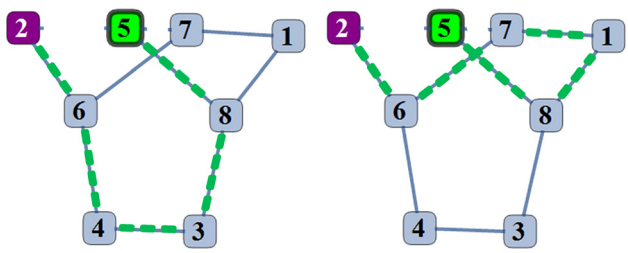

Fig. 6 Two second level shortest paths

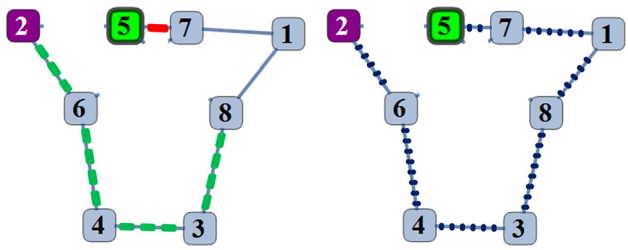

Fig. 7 A third level shortest path

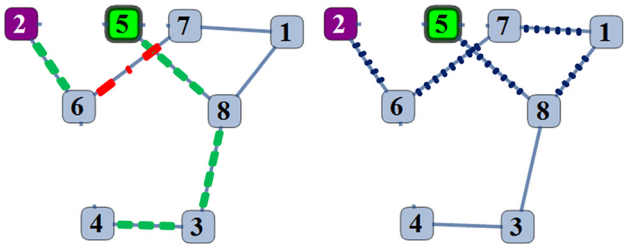

Fig. 8 Another third level shortest path

The unique shortest paths are formed into cycles by adding the root edge. The four unique cycles are sorted by length and loop vertex degree and shown in Fig. 9.

The first cycle {5, 2, 6, 7, 5} is the smallest and is selected as the first cycle in the cycle basis. The cycle {5, 2, 6, 7, 1, 8, 5} contributes the fewest new edges so it is selected next. Both of the remaining cycles contribute three new edges so the first one in the sorted list is added to the cycle basis {5, 2, 6, 4, 3, 8, 5}. For an 8-bar linkage, there are only three independent cycles; therefore, the remaining cycle is not independent of the selected three cycles and is discarded. The selected cycles are once again sorted by

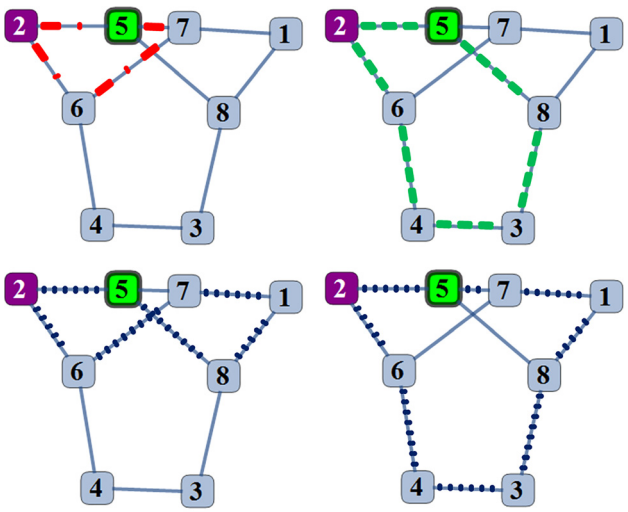

Fig. 9 The four unique cycles

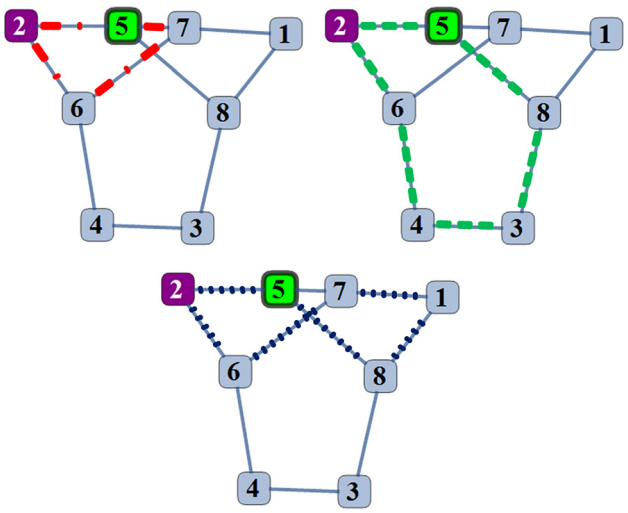

Fig. 10 The cycle basis

Table 2 Example linkage rooted cycle basis and vertex degree list

Loops	Cycle basis	Vertex degree list
1	{5, 2, 6, 7, 5}	{3, 2, 3, 3, 3}
2	{5, 2, 6, 4, 3, 8, 5}	{3, 2, 3, 2, 2, 3, 3}
3	{5, 2, 6, 7, 1, 8, 5}	{3, 2, 3, 3, 2, 3, 3}

length and loop vertex degree to produce the rooted cycle basis shown in Fig. 10. The loops and the vertex degrees are shown numerically in Table 2.

Unique Mechanisms and Unique Linkages

The automation must be able to analyze the configuration of all unique linkages, specifically, all unique combinations of topology, ground link, and ground-connected input link. To test the analysis routine, we enumerated and analyzed all of the unique linkages.

For each topology, there may be several choices for the ground link that produce the same mechanism. Similar to the definition of a graph isomorphism, nonunique mechanisms will have graphs with a one-to-one correspondence of vertices that preserve the incidence, meaning the degree of the adjacent vertices, as well as the correspondence of the selected ground link. For example, the double butterfly 8-bar mechanism has only two unique choices for the ground vertex, Fig. 11. Every other selection for the ground vertex can be made into one of the two unique mechanisms by renumbering the vertices.

Several selections of ground and ground-connected input link may produce the same linkage. Similar to a graph isomorphism and a nonunique mechanism, the graphs of nonunique linkages will have a one-to-one correspondence of vertices that preserve the incidence as well as the correspondence of the selected ground link and input link. For the double butterfly 8-bar linkage, there are only three choices for the ground and ground-connected input vertex that are unique, Fig. 12. Every other selection for the ground and ground-connected input vertices can be made into one of these three linkages by renumbering the vertices.

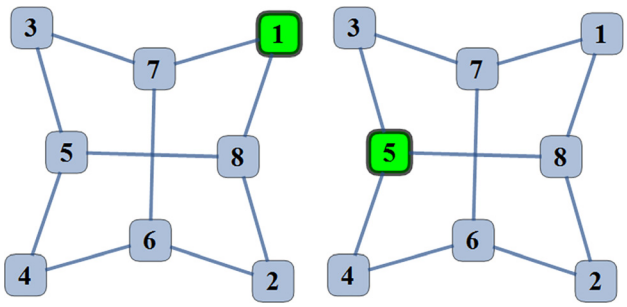

Fig. 11 Double butterfly unique mechanisms

To identify a nonunique linkage, the algorithm compares the incidence of the vertices along the cycles of the rooted cycle basis. The two cycle bases being compared have both been consistently sorted by cycle length and consistently ordered within each cycle such that the first vertex is ground, the second vertex is the ground-connected input, and the last vertex is ground. The incidence of each cycle is represented by the vertex degrees taken in order along the loop, Table 2. 8-bar linkages that are not unique preserve the incidence along the loops of the cycle basis, meaning they have the same set of three vertex degree lists in the same order. Comparing the vertex degree lists of every possible 8-bar linkage with a ground-connected input reveals that every unique linkage has a unique set of three vertex degree lists defined by the rooted cycle basis.

Defining the Linkage Dimensional Features

The only input needed to construct the loop equations is the rooted cycle basis. To automatically establish the linkage loop equations, we apply a naming convention to the linkage loops to construct unique names for the links, the joints, and the lines between joints along a loop. We also define the name and location of the link angles and the fixed angles representing divergence and convergence of loops on ternary and higher links. When two loops diverge or converge, we name the lines along the two loops and the angle between them.

The cycle basis provides an ordered list of vertices for each loop. These vertices are also the names of the links which are joined in order along each loop. Following the rooted cycle basis along each loop, we assign a unique name for the joints based on the links being joined. Because there is only one joint between any two links, these joint names are always unique. The first character of the joint name is "*j*," followed by the number of the first link being joined, followed by "*t*," and finished with the number of the second link being joined. For example, a joint between link 7 and link 5 is called *j*7*t*5. The *t* enables unambiguous distinction between links even when the link number has more than one digit.

Also following the order of the links as shown for each linkage loop, the physical distance between two joints on the same link is given a unique name based on the two end joints. The first character is "*L*," followed by the name of the first joint (with the *j* omitted), followed by *t*, followed by the last digits of the ending joint. For example, the dimension of the line on link 5 between the joints *j*7*t*5 and *j*5*t*2 is called *L*7*t*5*t*2. In a binary link, this dimension is intuitive, the distance between the two joints. In a ternary link, this is the distance between two of the three joints.

For defining angles, we use the convention that all angles are positive counter clockwise. The global angle of a link is defined from a global reference to a feature on the link. The selected global reference is the *x* axis. The selected feature on a binary link is intuitive, the line between the two joints. For ternary and higher links, there will be two or more features that could be selected. We define the global angle of the link as the angle from the global

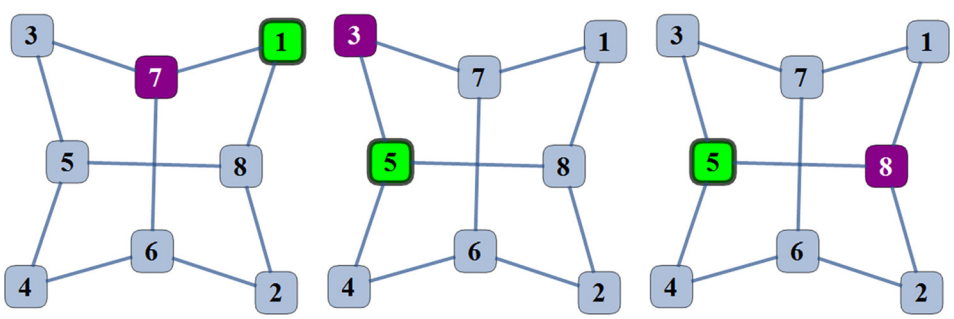

Fig. 12 Double butterfly unique linkages

x axis to the line between the joints along the first loop that contains the link. The origin of that angle is the first joint of the link encountered along the loop. The name assigned to this angle is "*th*" followed by the link number. For example, if dimension *L*7*t*5*t*2 is part of the first loop then the angle relative to the global frame for *L*7*t*5*t*2 is called *th*5 and the origin of that angle is at the joint *j*7*t*5.

The vertices where loops diverge (or converge) represent ternary or higher links. We need to define the angle for the line along the divergent (or convergent) loop. We do this by defining a fixed angle to describe the angle of the divergent (or convergent) loop relative to the reference loop from which the loop diverges (or converges). This fixed angle is added to the angle defining the line along the reference loop. The fixed angle is located at the common joint of the two loops and begins from the line along the reference loop and ends at the line along the divergent (or convergent) loop. The name of the fixed angle is based on the two lines. The name is "*fix*" followed by the name of the line in the reference loop (with the *L* omitted), followed by "*tt*," followed by the name of the line along the divergent (or convergent) loop (with the *L* omitted). In quaternary links, there will be three loops that pass through the link. The third loop may diverge or converge relative to a line that diverges or converges from the link angle. The same naming convention applies in this case so the final angle of the third loop will be a summation of the link angle and two fixed angles.

Derived automatically from the cycle basis shown in Fig. 10, we apply the naming convention to the example 8-bar linkage in Fig. 13.

Converting the Rooted Cycle Basis to Loop Equations

To automate the process of developing the loop equations, we apply the naming convention by automating the construction of a convention called FTLA. FTLA describes the link feature along each loop with four terms {from joint, to joint, link dimension, and angle}. Each loop is represented by a series of these four-term sets and the first of the four-term sets for each loop represents the line between two joints on the ground link. To sufficiently define a line between two joints, only three of these four terms are needed and the fourth can be derived; however, we choose to retain all four terms for ease of automation. The only input needed to develop the FTLA is the rooted cycle basis.

To create the FTLA for a linkage, first the rooted cycle basis is converted to the series of joints it represents. The joints are then paired to represent the end points of the link features along the linkage loops. The lines between the joints are named based on the joints and ordered such that the first line in each FTLA represents a line on the ground link.

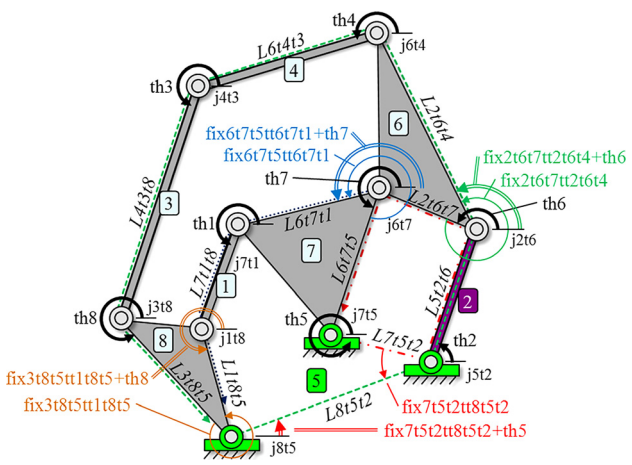

Fig. 13 Naming convention, example 8-bar

The angle for a given line is defined in two steps. First, we establish the global angle from the x axis to the link, specifically to the line between the joints along the first loop that contains the link. Second, we establish the fixed angle that must be added to the reference line to properly define the divergent or convergent line. When two loops converge the "to joint" will match, when two loops diverge the "from joint" will match. Subtracting the from joint and to joint terms of each previous loop from the same terms in the current loop reveals the locations where the current loop converges to (or diverges from) the previous loop. These fit the form of the following equation:

$$\begin{aligned}&\text{convergent loop form} : \{\text{Joint}_2 - \text{Joint}_1, 0\}\\&\text{divergent loop form} : \{0, \text{Joint}_2 - \text{Joint}_1\}\end{aligned} \tag{3}$$

Because we know which entries in the FTLA lists are subtracted, we know the locations of the loop divergences and convergences so we map the fixed angles to the appropriate location in the FTLA and the appropriate name for the fixed angle is added to the angle defining the reference line.

We convert from the FTLA form to the final loop equations by taking the sum along each loop of the product

$$\begin{aligned}X &: (\text{link dimension}) * \cos(\text{angle})\\Y &: (\text{link dimension}) * \sin(\text{angle})\end{aligned} \tag{4}$$

Example, Creating the Loop Equations

Continuing with our example 8-bar shown in Fig. 13, the first step in deriving the linkage loop equations uses text manipulation of the rooted cycle basis in Table 2 to define the terms {from joint, to joint, link dimension, and link angle}. At this first step, the link angle does not account for the fixed angles between divergent or convergent loops. The {from joint, to joint} pairs along the loops are shown in the following equation:

$$\begin{aligned}&\text{Loop1}\\&\{j7t5, j5t2\}, \{j5t2, j2t6\}, \{j2t6, j6t7\}, \{j6t7, j7t5\}\\&\text{Loop2}\\&\{j8t5, j5t2\}, \{j5t2, j2t6\}, \{j2t6, j6t4\}, \{j6t4, j4t3\},\\&\{j4t3, j3t8\}, \{j3t8, j8t5\}\\&\text{Loop3}\\&\{j8t5, j5t2\}, \{j5t2, j2t6\}, \{j2t6, j6t7\}, \{j6t7, j7t1\}\\&\{j7t1, j1t8\}, \{j1t8, j8t5\}\end{aligned} \tag{5}$$

Using Eq. (5), we subtract each term in loop 1 from each term in loop 2 to reveal where the loops diverge and converge. Loop 2 and loop 1 converge at *j*5*t*2 and diverge at *j*2*t*6. The difference between the two loops fits the desired pattern at these joints, as follows:

$$\begin{aligned}\{j8t5, j5t2\} - \{j7t5, j5t2\} &= \{j8t5 - j7t5, 0\}\\\{j2t6, j6t4\} - \{j2t6, j6t7\} &= \{0, j6t4 - j6t7\}\end{aligned} \tag{6}$$

The same process is applied to loop 3 relative to loop 1 and loop 2. Loop 3 converges to loop 1 at *j*5*t*2 following the same path along *L*8*t*5*t*2 as loop 2; therefore, the fixed angle about *j*5*t*2 is the same for both loop 2 and loop 3. Loop 3 also converges to loop 2 at *j*8*t*5 and diverges from loop 1 at *j*6*t*7.

The locations of the loop divergences and convergences are mapped to the appropriate location in the loops and the appropriate name for the fixed angle is assigned. The final angle of the linkage feature is the sum of the fixed angle and the link angle.

The final FTLA convention is shown in the following equation:

$$
\begin{aligned}
&\text{Loop1}\{j7t5, j5t2, L7t5t2, th5\}, \{j5t2, j2t6, L5t2t6, th2\},\\
&\qquad \{j2t6, j6t7, L2t6t7, th6\}, \{j6t7, j7t5, L6t7t5, th7\}\\
&\text{Loop2}\{j8t5, j5t2, L8t5t2, fix7t5t2tt8t5t2 + th5\},\\
&\qquad \{j5t2, j2t6, L5t2t6, th2\},\\
&\qquad \{j2t6, j6t4, L2t6t4, fix2t6t7tt2t6t4 + th6\},\\
&\qquad \{j6t4, j4t3, L6t4t3, th4\}, \{j4t3, j3t8, L4t3t8, th3\},\\
&\qquad \{j3t8, j8t5, L3t8t5, th8\}\\
&\text{Loop3}\{j8t5, j5t2, L8t5t2, fix7t5t2tt8t5t2 + th5\},\\
&\qquad \{j5t2, j2t6, L5t2t6, th2\}, \{j2t6, j6t7, L2t6t7, th6\},\\
&\qquad \{j6t7, j7t1, L6t7t1, fix6t7t5tt6t7t1 + th7\},\\
&\qquad \{j7t1, j1t8, L7t1t8, th1\},\\
&\qquad \{j1t8, j8t5, L1t8t5, fix3t8t5tt1t8t5 + th8\}
\end{aligned}
\tag{7}
$$

The final loop equations are shown in the following equation:

$$
\begin{aligned}
&\text{Loop1}\\
&X : L7t5t2\cos(th5) + L5t2t6\cos(th2)\\
&\quad + L2t6t7\cos(th6) + L6t7t5\cos(th7) = 0\\
&Y : L7t5t2\sin(th5) + L5t2t6\sin(th2)\\
&\quad + L2t6t7\sin(th6) + L6t7t5\sin(th7) = 0\\
&\text{Loop2}\\
&X : L8t5t2\cos(fix7t5t2tt8t5t2 + th5) + L5t2t6\cos(th2)\\
&\quad + L2t6t4\cos(fix2t6t7tt2t6t4 + th6) + L6t4t3\cos(th4)\\
&\quad + L4t3t8\cos(th3) + L3t8t5\cos(th8) = 0\\
&Y : L8t5t2\sin(fix7t5t2tt8t5t2 + th5) + L5t2t6\sin(th2)\\
&\quad + L2t6t4\sin(fix2t6t7tt2t6t4 + th6) + L6t4t3\sin(th4)\\
&\quad + L4t3t8\sin(th3) + L3t8t5\sin(th8) = 0\\
&\text{Loop3}\\
&X : L8t5t2\cos(fix7t5t2tt8t5t2 + th5) + L5t2t6\cos(th2)\\
&\quad + L2t6t7\cos(th6) + L6t7t1\cos(fix6t7t5tt6t7t1 + th7)\\
&\quad + L7t1t8\cos(th1) + L1t8t5\cos(fix3t8t5tt1t8t5 + th8) = 0\\
&Y : L8t5t2\sin(fix7t5t2tt8t5t2 + th5) + L5t2t6\sin(th2)\\
&\quad + L2t6t7\sin(th6) + L6t7t1\sin(fix6t7t5tt6t7t1 + th7)\\
&\quad + L7t1t8\sin(th1) + L1t8t5\sin(fix3t8t5tt1t8t5 + th8) = 0
\end{aligned}
\tag{8}
$$

Dixon Determinant Derivation

To solve for the angles of all of the links, we solve the Dixon determinant using the complex plane formulation as shown by Wampler [14]. To convert the loop equations to complex form, we treat the Y direction as along the imaginary plane. Multiply the Y equations by i where $i^2 = -1$ and sum the X and Y equations. We apply trigonometric identities and exponential identities to transform the loop equations into imaginary form. To solve for the unknown link angles, the conjugates of the complex loop equations are used to provide the full equation set.

The Dixon determinant method requires the selection of one unknown angle to be used as a generalized eigenvalue while the remaining angles are solved as a generalized eigenvector. The Dixon determinant form is

$$
[\mathbf{M}\Theta_n - \mathbf{N}]\mathbf{t} = 0 \tag{9}
$$

where M and N are matrices with constant coefficients comprised of the linkage dimensions, the ground angle, the input angle, and the complex conjugate of the input angle. Θ_n is the unknown angle selected to be the eigenvalue and the vector $\mathbf{t}$ is the set of monomials representing the remaining unknown link angles.

Some unknown angles are poor choices for the eigenvalue Θ_n because the resulting eigenvector $\mathbf{t}$ cannot be used to solve for all of the remaining link angles. To cancel any scaling factors that may exist, the final step of the solution process takes the ratio of two elements of $\mathbf{t}$ to determine the true numerical value of each angle. With a poor selection of Θ_n, there is no combination of elements in $\mathbf{t}$ whose ratio defines one or more of the unknown angles. To automate this aspect of the configuration analysis, we simply test each unknown angle as a candidate eigenvalue, derive the eigenvector $\mathbf{t}$ and verify, symbolically, that there exists a monomial ratio that will produce every unknown angle. The first candidate Θ_n that meets these criteria is selected as the eigenvalue.

Some linkages cannot be solved as a whole linkage using the Dixon determinant process, not because of a flaw in the process but because there is no valid selection of the eigenvalue Θ_n. These linkages partition and can be solved as independent sublinkages.

When a valid Θ_n is found, numerical values for the link features determined from the linkage synthesis are substituted into the Dixon determinant and the Dixon determinant is then numerically solved for a given input angle. The output provides all of the possible linkage assembly configurations for that input angle.

Block Diagonal Jacobian Derivation

Because the Dixon determinant provides all of the possible real assembly configurations, a means of identifying a particular assembly configuration within the solutions is needed. McCarthy and Soh [13] show a numerical method to track a particular solution through the range of input angles. The present research identifies a particular linkage assembly by factoring the Jacobian into 2×2 blocks and seeking the assembly configuration where the signs of the determinant of the 2×2 blocks match a desired pattern. The desired pattern represents the assembly configuration that reaches the task positions used in linkage synthesis. When the desired pattern does not exist for a particular input angle, either a branch or circuit defect has been encountered and the linkage assembly configuration will not move smoothly through that input angle.

The Jacobian is the derivative of each of the six loop equations, Eq. (8), with respect to each unknown angle. We convert the Jacobian to block upper triangular form through a determinant preserving transform as shown by Silvester [24], Eq. (10). The Jacobian can now be factored as the product of the determinant of each 2×2 block matrix along the diagonal

$$
\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}
\begin{bmatrix} \mathbf{I} & 0 \\ -\mathbf{D}^{-1}\mathbf{C} & \mathbf{I} \end{bmatrix}
= \begin{bmatrix} \mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C} & \mathbf{B} \\ 0 & \mathbf{D} \end{bmatrix} \tag{10}
$$

Since the transform involves an inverse of submatrix D, the columns of the Jacobian are first sorted so that the blocks along the diagonal are full rank to ensure that D is full rank. For the 8-bar family, the Jacobian is a 6×6 matrix and we apply the transform twice, the first transform treats the lower right 2×2 along the diagonal as D, the second transform treats the lower right 4×4 as D.

The example linkage, Fig. 13, has the input link and the ground link within a 4-bar sublinkage; therefore, singularities occur when the following features are collinear: $L2t6t7$ and $L6t7t5$, $L6t4t3$ and $L4t3t8$, or $L7t1t8$ and $L1t8t5$. Taking the determinant of each 2×2 block along the diagonal produces the three Jacobian factors expected, Eq. (11). One of these factors is zero when one of the expected link pairs is collinear

$$
\begin{aligned}
J_1 &: -L2t6t7\,L6t7t5(\cos(th7)\sin(th6) - \cos(th6)\sin(th7))\\
J_2 &: -L4t3t8\,L6t4t3(\cos(th4)\sin(th3) - \cos(th3)\sin(th4))\\
J_3 &: -L1t8t5\,L7t1t8(\cos(fix3t8t5tt1t8t5 + th8)\sin(th1)\\
&\quad - \cos(th1)\sin(fix3t8t5tt1t8t5 + th8))
\end{aligned}
\tag{11}
$$

For many linkages, the sign combination of the Jacobian factors uniquely identifies the configuration of interest among the

Table 3 Count of unique mechanisms and linkages, 6-bar and 8-bar

Links	Assortment	Topology	Mechanism	Linkage
6	4200	2	5	9
8	4400	9	35	76
	5210	5	31	68
	6020	2	5	9
	Total	16	71	153

solutions of the Dixon determinant. However, there may be exceptions such as the Stephenson III 6-bar linkage that could contain a link that rotates more than 360 deg before encountering a singularity [19]. For such linkages, the factored Jacobian alone may not be sufficient because within a particular range of input angles there may exist two configurations with the same sign list.

Results

Following this procedure, we have automatically derived loop equations for the entire family of 4-bar, 6-bar, and 8-bar 1DOF linkages with revolute joints. Four 10-bar topologies have also been successfully automated including two with nonplanar graphs and one with a quintenary link.

The process identified the five unique 6-bar mechanisms, Watt I–II and Stephenson I–III, as well as the nine unique 6-bar linkages with a ground-connected input, matching the known Watt and Stephenson families [8,9]. The quantity of 71 unique mechanisms identified for the 8-bar family matches the result published by Tuttle [7]. The process also provided a new result showing 153 unique 8-bar linkages with a ground-connected input. The results are summarized in Table 3.

The algorithm also properly identified the Watt IIb linkage as the one 6-bar linkage that partitions. Like the Watt IIb, the algorithm identified 24 linkages in the 8-bar family that do not have an acceptable selection for the eigenvalue angle Θ_n and cannot be solved as a whole linkage using the Dixon determinant. Inspection of these 24 linkages, and the Watt IIb, shows that the ground and input links are driving two 1DOF sublinkages whose assembly configurations are independent.

Conclusions

In this paper, we present a procedure to automatically create the linkage loop equations for the entire family of 1DOF linkages with revolute joints up to 8 bar. The process provides equations in a format suitable for the automation of the complete configuration analysis of planar 1DOF 8-bar linkages with revolute joints. The method is also general and forms the basis for automation of 10-bar and higher linkages.

Linkages containing multiple joints, joints connecting more than two links on a common axis, can be represented with these automated loop equations by setting the appropriate physical feature sizes to zero. Future work should determine if there are special considerations for the configuration analysis of multiple joint linkages through the automated Dixon determinant.

Extensions of the work are expected to include planar multidegree of freedom linkages, 10-bar and higher linkages, prismatic joints, inputs not connected to ground, and spherical linkages. We also expect to incorporate improvements in the algorithm for computational efficiency.

Acknowledgment

This material was based upon work supported by the National Science Foundation under Grant Nos. 1217322 and 1066082 and by the Office of Naval Research under Grant No. N00014-08-1-1015.

References

[1] Tsai, L.-W., 2000, *Mechanism Design: Enumeration of Kinematic Structures According to Function*, CRC Press, Boca Raton, FL.
[2] Sunkari, R. P., and Schmidt, L. C., 2006, "Structural Synthesis of Planar Kinematic Chains by Adapting a McKay-Type Algorithm," Mech. Mach. Theory, **41**(9), pp. 1021–1030.
[3] McKay, B. D., 1998, "Isomorph-Free Exhaustive Generation," J. Algorithms, **26**(2), pp. 306–324.
[4] Ding, H., and Huang, Z., 2007, "The Establishment of the Canonical Perimeter Topological Graph of Kinematic Chains and Isomorphism Identification," ASME J. Mech. Des., **129**(9), pp. 915–923.
[5] Ding, H., Hou, F., Kecskeméthy, A., and Huang, Z., 2012, "Synthesis of the Whole Family of Planar 1-DOF Kinematic Chains and Creation of Their Atlas Database," Mech. Mach. Theory, **47**, pp. 1–15.
[6] Ding, H., Yang, W., Huang, P., and Kecskeméthy, A., 2013, "Automatic Structural Synthesis of Planar Multiple Joint Kinematic Chains," ASME J. Mech. Des., **135**(9), p. 091007.
[7] Tuttle, E. R., 1996, "Generation of Planar Kinematic Chains," Mech. Mach. Theory, **31**(6), pp. 729–748.
[8] Manolescu, N., 1973, "A Method Based on Baranov Trusses, and Using Graph Theory to Find the Set of Planar Jointed Kinematic Chain and Mechanisms," Mech. Mach. Theory, **8**(1), pp. 3–22.
[9] Verho, A., 1973, "An Extension of the Concept of the Group," Mech. Mach. Theory, **8**(2), pp. 249–256.
[10] Soh, G. S., and McCarthy, J. M., 2007, "Synthesis of Eight-Bar Linkages as Mechanically Constrained Parallel Robots," 12th World Congress in Mechanism and Machine Science (IFToMM), Besancon, France, June 18–21.
[11] Perez, A., and McCarthy, J. M., 2005, "Clifford Algebra Exponentials and Planar Linkage Synthesis Equations," ASME J. Mech. Des., **127**(5), pp. 931–940.
[12] Soh, G. S., and McCarthy, J. M., 2008, "The Synthesis of Six-Bar Linkages as Constrained Planar 3R Chains," Mech. Mach. Theory, **43**(2), pp. 160–170.
[13] McCarthy, J. M., and Soh, G. S., 2010, *Geometric Design of Linkages*, Springer, New York.
[14] Wampler, C. W., 2001, "Solving the Kinematics of Planar Mechanisms by Dixon Determinant and a Complex Plane Formulation," ASME J. Mech. Des., **123**(3), pp. 382–387.
[15] Dixon, A., 1909, "The Eliminant of Three Quantics in Two Independent Variables," *Proceedings of the London Mathematical Society*, C. F. Hodgson & Son, London, UK, pp. 49–69.
[16] Nielsen, J., and Roth, B., 1999, "Solving the Input/Output Problem for Planar Mechanisms," ASME J. Mech. Des., **121**(2), pp. 206–211.
[17] Dhingra, A. K., Almadi, A. N., and Kohli, D., 2001, "A Gröbner–Sylvester Hybrid Method for Closed-Form Displacement Analysis of Mechanisms," ASME J. Mech. Des., **122**(4), pp. 431–438.
[18] Porta, J. M., Ros, L., and Thomas, F., 2009, "A Linear Relaxation Technique for the Position Analysis of Multiloop Linkages," IEEE Trans. Rob., **25**(2), pp. 225–239.
[19] Chase, T. R., and Mirth, J. A., 1993, "Circuits and Branches of Single-Degree-of-Freedom Planar Linkages," ASME J. Mech. Des., **115**(2), pp. 223–230.
[20] Myszka, D. H., Murray, A. P., and Wampler, C. W., 2012, "Mechanism Branches, Turning Curves and Critical Points," ASME Paper No. DETC2012-70277.
[21] Kecskeméthy, A., Krupp, T., and Hiller, M., 1997, "Symbolic Processing of Multiloop Mechanism Dynamics Using Closed-Form Kinematics Solutions," Multibody Syst. Dyn., **1**(1), pp. 23–45.
[22] Whitney, H., 1932, "Non-Separable and Planar Graphs," Trans. Am. Math. Soc., **34**(2), pp. 339–362.
[23] Dijkstra, E. W., 1959, "A Note on Two Problems in Connexion With Graphs," Numer. Math., **1**(1), pp. 269–271.
[24] Silvester, J. R., 2000, "Determinants of Block Matrices," Math. Gaz., **84**(501), pp. 460–467.