UNIVERSITY OF CALIFORNIA,
IRVINE

Automated Configuration Analysis of Planar Eight-Bar Linkages

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Mechanical and Aerospace Engineering

by

Brian Edward Parrish

Dissertation Committee:
J. Michael McCarthy, Chair
David A. Eppstein
David J. Reinkensmeyer

2014

DEDICATION

To my wife Deborah whose encouragement to pursue this effort was instrumental in beginning and completing the work. To my children Kyle and Kaeli who are pursuing God's purpose for their lives.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to thank Professor Michael McCarthy for providing the guidance to pursue this contribution to linkage design that enabled a general approach for ensuring the usefulness of synthesized planar linkages. He encouraged me to pursue the aspect of research that was of interest to me.

I would like to thank Professor David Eppstein for providing guidance in the field of graph theory that revealed the need for the improvements now implemented. With his guidance this research now implements a consistent approach for finding the cycle basis so that unique linkages can be uniquely identified.

I would like to thank my employer Northrop Grumman for their financial support through the NGAS Fellowship Program.

I appreciate Kaustubh Sonawale for joining the algorithm produced by this research with his own algorithm for linkage synthesis to make a complete linkage synthesis and analysis suite. His success revealed the value of some of the detailed interface conventions generated as part of this research. The FTLA convention and the enhanced adjacency matrix were demonstrated to be very effective.

I am grateful to Gim-Song Soh whose work on the Dixon determinant yielded a starting point for a portion of this automation.

I owe many thanks to my family for encouraging me and supporting me, especially when I felt like I was in over my head. At several moments my wife had more faith in me than I did and the value of her encouragement cannot be over stated.

I would like to thank my church family for keeping me in their prayers. There are many that could be listed here but specifically I want to thank Pastors Paul and Bonnie Thiemens.

The text of this dissertation contains a reprint of the material as it appears in ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. The co-author listed in this publication directed and supervised research which forms the basis for the dissertation.

CURRICULUM VITAE

Brian Edward Parrish

EDUCATION

Doctor of Philosophy in Mechanical and Aerospace Engineering **2014**
University of California, Irvine *Irvine, California*

Master of Science in Mechanical Engineering **1998**
California State University, Long Beach *Long Beach, California*

Bachelor of Science in Mechanical Engineering **1993**
California State University, Long Beach *Long Beach, California*

RESEARCH EXPERIENCE

Graduate Research Assistant, Robotics and Automation Laboratory **2010–2014**
University of California, Irvine *Irvine, California*

WORK EXPERIENCE

Mechanical Engineer, High Reliability Electronics **1993–2006**
TRW *Manhattan Beach, California*

Department Manager, Materials and Processing **2006–2009**
Northrop Grumman Aerospace Systems *Manhattan Beach, California*

Risk Manager, High Reliability Electronics **2009–Present**
Northrop Grumman Aerospace Systems *El Segundo, California*

Qualification Manager, High Reliability Electronics **2009–Present**
Northrop Grumman Aerospace Systems *El Segundo, California*

REFEREED CONFERENCE PUBLICATIONS

Identification of a Usable Six-Bar Linkage for Dimensional Synthesis **Sept 2012**
EUCOMES 2012: 4th European Conference on Mechanism Science

Use of the Jacobian to Verify Smooth Movement in Watt I and Stephenson I Six-Bar Linkages **Aug 2013**
IDETC 2013: 37th Mechanisms and Robotics Conference

Automated Generation of Linkage Loop Equations for Planar 1-DoF Linkages, Demonstrated up to 8-bar **Aug 2014**
Accepted for presentation at IDETC 2014: 38th Mechanisms and Robotics Conference

PATENTS

Integrated Design and Manufacturing System **Sep 18, 2001**
Patent No. US6292707 B1

Continuously Variable Transmission **May 14, 2002**
Patent No. US6387004 B1

Captivated Jackscrew Design **Jun 18, 2002**
Patent No. US6406210 B1

SOFTWARE

Planar Linkage Analysis Tool Up To 8-Bar
Mathematica algorithm to automatically produce the configuration analysis equations and factored Jacobian for planar 4, 6 and 8-bar linkages.

ASIC Electrical Pin-Out Evaluator
EXCEL tool to ensure the design of the I/O of custom ASICs complies with multiple design rules.

Risk Management Database and User Interface
EXCEL tool set to improve the user interface to a risk management database and to provide export capability to adjacent tool sets.

PROJECT EXPERIENCE

Human Powered Vehicle Lead **1991–1992**
California State University, Long Beach *Long Beach, California*

Leaning Vehicle with Dynamic and Static Stability **2006-2009**
Personal Development *Garden Grove, California*

ABSTRACT OF THE DISSERTATION

Automated Configuration Analysis of Planar Eight-Bar Linkages

By

Brian Edward Parrish

Doctor of Philosophy in Mechanical and Aerospace Engineering

University of California, Irvine, 2014

J. Michael McCarthy, Chair

This dissertation presents research into the automation of the configuration analysis of eight-bar linkages based on the known 16 linkage topologies. The eight-bar linkage topologies do not distinguish a ground link or an input link, and this research shows that the selection of a ground link yields 71 unique mechanisms. In addition, the selection of a driving link attached to ground results in 153 eight-bar linkages with distinct sets of loop equations. The research automates the configuration analysis process and has been demonstrated for four-bar and six-bar linkages as well as eight-bar linkages. A check of the process includes example 10-bar linkages.

The automation of linkage analysis begins with an adjacency matrix, which defines how the joints connect the links known as the topology of the linkage. The ground link and driving link are selected by the designer, and the automated analysis process determines the smallest set of independent loops. A naming convention is used to automate the definition of link features, dimensions, and joint angles, which yields a text version of the loop equations for the linkage. These loop equations are solved using the Dixon determinant to find the configuration of the linkage for each value of the driving link angle. This formulation also yields the Jacobian of the loop equations, and factors its determinant which is used to find the singularities of the linkage.

The contribution of this research is an automated analysis of eight-bar linkages that applies to four-bar and six-bar linkages, and has been useful in the analysis of 10-bar linkages. This approach has provided a new classification of the 153 eight-bar linkages by family, link assortment, topology, mechanism, and linkage. The automation process can formulate the loop equations and Jacobian conditions for all 153 cases, as well as for all four-bar and six-bar cases.

Automated configuration analysis for eight-bar linkages provides an important tool for evaluating the range of movement of linkages obtained in mechanism synthesis algorithms, allowing identification of linkages that achieve a required task.

Chapter 1

Introduction

To put the contribution of this research into perspective, this section provides an overview of the linkage design process, some key definitions, and a summary of the literature discussing kinematic chain topology enumeration, linkage synthesis and configuration analysis.

Portions of this chapter are from Parrish and McCarthy [31], used with permission.

1.1 Computer Aided Invention

The design process creates solutions to meet customer requirements. Typically many options are considered and the best candidates are selected for detailed study. When there exists a design for a similar set of requirements, modification of the known design to meet the new set of requirements is an attractive option. Ultimately a design selection is made and formalized. The process may be iterated based on results obtained during any of the steps.

Tsai [48] documents a formal design methodology based on the application of graph theory and combinatorial analysis. This approach provides the designer with many possible solu-

tions and may reveal design options that have been historically overlooked. The process follows seven steps that may be iterated to achieve a final product. An application of this process is shown by Shieh [35] to design planar leg mechanisms.

1. Identify the functional requirements.
2. Determine the nature of the motion (planar, spherical, etc.), the degrees of freedom, and the complexity of the mechanism.
3. Identify the structural characteristics.
4. Enumerate all possible kinematic chains that satisfy the desired characteristics.
5. Sketch and evaluate the possible kinematic chains to yield a set of feasible mechanisms.
6. Select the most promising option and perform detailed design: dimensional synthesis, simulation, modeling, prototype.
7. Produce the design.

Enumerating the kinematic chains, evaluating each candidate to identify the set of feasible mechanisms, performing dimensional synthesis, and simulating the motion are all tasks best performed by computer automation. As early as 1973, Bona et al. [2] provided ideas for how to use the computer to automate the design process. The process of automating the synthesis and analysis of a linkage on the computer to develop candidate designs is sometimes called computer aided invention [22].

The ideal computer aided invention process would automatically evaluate all of the viable kinematic chains produced by the enumeration step.

1.2 Definitions and Scope

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.

Some authors such as Manolescu [21] refer to linkages as driving mechanisms. The editor's note in Manolescu's paper states "A word to the American reader may be helpful: the author follows European custom in making distinction between three types of linkage, where Americans usually think only of two types. These are the kinematic chain, having no fixed link; the mechanism, which has one link fixed as a frame; and the 'driving mechanism', having both a fixed frame and one link specified as the driving link. The German terms for these are respectively Kette, Mehcanismus, and Getriebe; there is no English term at present which translates Getriebe."

The scope of this research is the mechanical linkage, Getriebe, up to linkages with eight bars. The research is limited to linkages that move in a plane, use only revolute joints and have input links that are adjacent to, connected to, ground. The research excludes linkages with a passive degree of freedom, over-constrained linkages, fractionated kinematic chains, and kinematic chains with partially locked sub-chains.

1.3 Enumeration of Kinematic Chains

Kinematic chain enumeration identifies all of the topologies capable of producing the desired motion. For the purposes of this research one type of kinematic chain is explored, the planar one degree of freedom (one-DoF) linkage. Even after selecting this kinematic type there is a sufficiently high quantity of kinematic chains that dimensional synthesis and configuration analysis are often performed on only a subset of the available topologies. For example, the six-bar planar linkage family contains two distinct topologies named the Watt and the Stephenson families. Within those two families there are five distinct mechanisms, the Watt I, Watt II, Stephenson I, Stephenson II, and Stephenson III and nine distinct linkages, the Watt Ia, Watt Ib, Watt IIa, Watt IIb, Stephenson I, Stephenson IIa, Stephenson IIb, Stephenson IIIa and Stephenson IIIb. The eight-bar family contains 16 distinct topologies and 71 distinct mechanisms have been identified [49]. All of these are planar one degree of freedom linkage topologies and all are potentially capable of solving the same design requirements.

1.4 Linkage Dimensional Synthesis

Dimensional synthesis solves for the geometric features of a linkage to produce design candidates from one of the enumerated kinematic chain topologies. Several, or ideally all, of the enumerated kinematic chains should be used to synthesize candidate linkages. There are three general types of linkage dimensional synthesis.

Path generation synthesizes the linkage geometric features so that a point on a part of the linkage traces a particular path. For an example see Kim et al. [18]. Function generation synthesizes a linkage so that the input and the output have a numerical relationship that approximates a function. For an example see McLarnan [25].

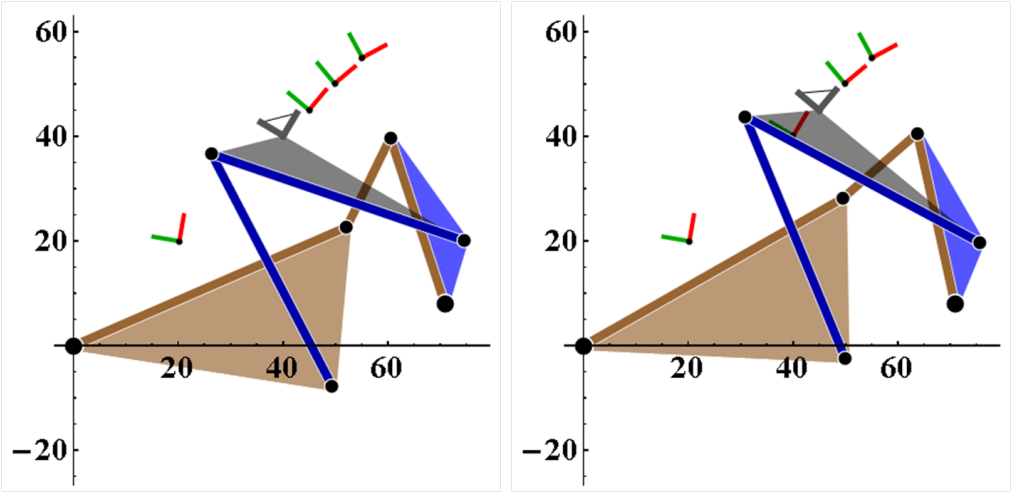


Figure 1.1: Example Stephenson Ia linkage reaching the second and third task positions of a set of five task positions.

A third synthesis method called task generation synthesizes a linkage to cause the end-effector to reach a set of task positions defined by the global locations and the orientations of the end-effector. As described by Soh and McCarthy [40] two RR dyads are synthesized using Burmester theory to constrain a 3R chain and form a six-bar linkage with one degree of freedom. To generate several candidate linkages, the desired task positions are varied within small tolerance zones. An example task generation problem is shown in Fig. 1.1 where the task is specified by five task positions, meaning five global locations and five global angles for the end-effector [31].

A successful linkage synthesis will provide a linkage geometry that reaches each of these task positions however it does not guarantee that the linkage will be useful.

1.5 Configuration Analysis

Configuration analysis, sometimes called kinematic analysis, evaluates a linkage candidate design to determine if the linkage is useful. The first criterion requires that a useful linkage

move smoothly, continuously with an increasing input angle, through the task and all angles between the task positions. Although the dimensional synthesis approach provides linkage designs that reach the specified task positions, a candidate linkage may not move smoothly through the range of input angles necessary to reach the task positions. Linkages that do not provide smooth movement have branch or circuit defects. As defined by Chase and Mirth [3] a circuit is all possible orientations of the links which can be realized without disconnecting any of the joints and a branch is the continuous series of positions of the mechanism on the circuit between two stationary configurations. Branch and circuit defects arise when the task positions are on different branches or circuits.

A branch defect occurs when the linkage cannot reach one of the five task positions without passing through a stationary configuration, a singularity. At a singularity the linkage has reached the limit of its range of motion for the input link. Linkages with a branch defect can still reach all of the task positions by reversing the input direction but the linkage must also change the orientation of some of the links. Disassembly of the linkage is not required.

Another type of defect is a circuit defect. A linkage with a circuit defect can be assembled to meet all of the required output conditions but it cannot meet all of the required output conditions using one assembly. Such a linkage must be disassembled and reassembled into a different configuration to meet at least one of the required output conditions.

Configuration analysis is performed by constructing and solving the linkage loop equations to determine the link angles for every link in the linkage. The solutions must be obtained over the range of input necessary to reach the task for which the linkage was dimensionally synthesized. The method chosen for solving the loop equations in this research is the Dixon determinant.

Because there may be several valid assembly configurations for a given input angle, the solutions for the loop equations must be tracked to ensure the same assembly configuration reaches all of the defined tasks. This is typically done numerically.

1.6 Literature Survey

The mobility of kinematic chains was established for any number of bars in 1968 by Davies [4]. In 2005 Sunkari and Schmidt [42] point out that this work may be based on an erroneous assumption that planar linkages have planar graphs.

Franke [16] identified the 16 topologies of one-DoF eight-bar 10-joint kinematic chains using a concise notation that makes visualization of the linkage connections intuitive. Davies and Crossley [5] applied Franke's notation to enumerate the 230 distinct one-DoF 10-bar 13-joint kinematic chains. Woo [56] applied permutation groups to enumerate the 230 distinct 10-bar kinematic chains and provided a sketch of each. Tsai [48] published an atlas of the 16 one-DoF eight-bar 10-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 [43] apply a McKay-type algorithm [24] to show that there are approximately 20 million non-isomorphic topologies in the 16 bar 22 joint kinematic chain family. Ding and Huang [9, 10] established a canonical representation of the linkage graphs and published a method for isomorphism detection based on the largest perimeter loop and the degrees of the vertices. Ding et al. [8] published the enumeration of

graphs of kinematic chains up to 14-bars and recently published work extending the development of linkage graphs to linkages that contain multiple joints, joints on a common axis [11].

Davies [4] defines a mechanism as a kinematic chain with one link fixed to a global reference frame, ground. Manolescu [21] calls these planar jointed mechanisms while Tuttle [49] calls these inversions. In this research these are called mechanisms. Tuttle [49] determined the number of distinct mechanisms of the one, two and three-DoF kinematic chains by applying a procedure closely related to the approach originally proposed by Davies and Crossley [5]. The results show that for one-DoF linkages there are five distinct six-bar mechanisms and 71 distinct eight-bar mechanisms. Mruthyunjaya and Raghavan [27] propose a method to detect unique mechanisms based on determining the number of paths of lengths 1 to n from ground to each link but in a later paper Mruthyunjaya [26] recognize Tuttle's results as the most comprehensive enumeration of chains and mechanisms.

Using Baranov trusses and an inspection for symmetry Manolescu [21] totals the number of links with different characteristics to identify the unique mechanisms and linkages. The characteristics are the degree (binary, ternary, etc.) of the link, the degree of the links to which the link is adjoined, and the position of the link relative to the loops. Using this criteria Manolescu identifies the three distinct Stephenson six-bar mechanisms and the two distinct Watt six-bar mechanisms, total of five, as well as 19 unique linkages. Of those 19 unique linkages nine have a ground-connected input. Verho [50] also allows actuation through link pairs that are not grounded and identifies 25 unique six-bar linkages using Assur groups and visual inspection of the Assur group combinations. Of those 25 linkages nine have a ground-connected input and those nine match the nine identified by Manolescu. The discrepancy between the 19 found by Manolescu and the 25 found by Verho was discussed by Mruthyunjaya and Raghavan [27] who propose a computational method to detect unique linkages by determining the number of paths of lengths 1 to n from ground to the two

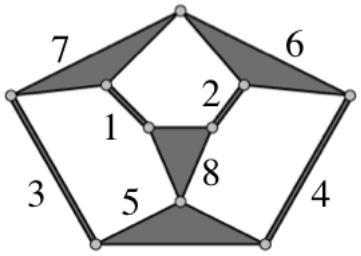


Figure 1.2: Double Butterfly eight-bar topology sketch.

ends of an actuation pair (AP). This method is based on the method used to detect unique mechanisms and attempts to identify unique linkages based on finding the structurally unique placement of the actuation pairs.

Simoni et al. [38] show a method using group theory to enumerate "parallel manipulators". They do not consider the input link but instead enumerate the number of unique combinations of ground and output links that are possible in a kinematic chain. Their conclusion states that future work will provide the "elaboration of criteria for the classification of the manipulators because the number of parallel manipulators which each chain can originate is generally very great and it is difficult to analyze the individual merits of each manipulator."

Linkage synthesis solves for the specific dimensions of the links using one of the enumerated topologies. Soh and McCarthy [39] published a methodology specific to the eight-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 linkage topologies [32, 41, 40, 23]. Some synthesis approaches incorporate prismatic joints [34]. Since all possible topologies of one-DoF linkages with revolute joints are known for the eight-bar family, every synthesized linkage must be in one of these topologies.

Linkage configuration analysis solves for the angles of all the output links. Approaches are typically shown for specific topologies. Various methods for solving the configuration of a linkage have been published such as the Gröbner-Sylvester method by Dhingra et al. [6] and

the linear relaxation method by Porta et al. [33]. For example, the topology shown in Fig. 1.2 represents a Double Butterfly eight-bar. Wampler [51], referencing Dixon [12], analyzed a Double Butterfly linkage using the Dixon determinant in a complex plane formulation. For completeness there is a continuation of the paper by Dixon with the same title [13]. The same linkage was also evaluated in rational formulation by Nielsen and Roth [29]. The Dixon determinant approaches provide all of the solutions, meaning all possible assembly configurations, of the linkage for a given input angle.

To determine if a particular assembly configuration is usable the singular configurations must be avoided. Linkages that encounter a singularity within the range of motion of interest have a branching defect. Branching defects depend on which link is selected to be ground and which link is the driven input link. Wang et al. [52] determines the branches of six-bar linkages using algebraic techniques involving the quadratic discriminant of the loop equations. Ting et al. [47, 45, 44, 46] and Dou and Ting [14] determine the branches of a linkage using the four-bar coupler curve and the five-bar joint rotation space. Chase and Mirth [3] use the sign of the determinant of the Jacobian to identify the branches of six-bar linkages. Myszka et al. [28] identifies the singularities and plots the singular configurations as curves that are a function of the length of one of the links.

Kecskeméthy et al. [17] 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 configuration analysis for all topologies of planar one-DoF linkages has not been published.

1.7 Contribution

The contribution of this research is the automation of the configuration analysis of a linkage in any planar eight-bar topology with revolute joints. The algorithm automatically constructs the linkage loop equations for any topology of planar one-DoF linkage with revolute joints up to eight bars and has been demonstrated on example 10-bar linkages including linkages with non-planar graphs and one with a quintenary link.

The automation is formulated in a structured way that determines a specific cycle basis through a common edge. This produces repeatable results enabling the unique identification of every unique linkage up to eight bars. This research identifies for the first time 153 distinct eight-bar linkages with a ground-connected input.

The unique identification of the linkages has been leveraged to sort every linkage and establish a linkage classification convention, NATML, where every linkage is uniquely identified by a specific five number index, $\{n, a, t, m, l\}$. This convention represents a linkage by family (number of bars), link assortment (number of links by type), topology (adjacency matrix), mechanism (selected ground link), and linkage (selected driving link). The classification is applicable to four-bar, six-bar, and eight-bar linkages.

The Dixon determinant is automatically derived from the loop equations and the configuration analysis is accomplished by solving the Dixon determinant for a given input angle to determine all possible assembly configurations. Using the eigenvalue, a method for identifying a linkage that partitions into simpler linkages has been established.

The interfaces between a linkage synthesis algorithm and this analysis algorithm are defined.

The automation is formulated in a general way to lay the ground work for future extensions of the approach that would analyze linkage topologies with more links, more degrees of freedom, and prismatic joints.

1.8 Introduction Conclusion

This section provided a frame of reference for automated configuration analysis of mechanical linkages. Definitions were provided for the kinematic chain, topology, mechanism, and linkage. An overview of the design process was provided as well as a summary of the literature covering enumeration of the kinematic chain topologies, dimensional synthesis of design candidates, and configuration analysis of design candidates. The contribution of this research was also provided.

Chapter 2

Mathematical Background

This chapter discusses the mathematical background that will be applied to the analysis of planar eight-bar linkages.

Portions of this chapter are from Parrish et al. [31], used with permission.

2.1 Linkage Kinematics

There are many types of joints that can be used in the design of a linkage and these can be combined in a variety of ways to produce a linkage with varying degrees of freedom. Table 2.1 shows some common joint constraints [48]. In this research we are considering only the simplest constraint, the revolute joint. Even considering only this simplest type of joint there are still many linkage topologies available.

Table 2.1: Degree of freedom for common linkage joints.

Kinematic Pair	Symbol	Joint DoF
Revolute	R	1
Prismatic	P	1
Cylindrical	C	2
Helical	H	1
Spherical	S	3
Plane	E	3
Gear Pair	G	2
Cam Pair	Cp	2

2.2 Mobility of Planar Linkages

The degree of freedom of a mechanism is given by Eqn. 2.1.

$$F = \lambda(n-1) - \sum_{i=1}^{j} c_i \tag{2.1}$$

where λ is the degree of freedom of the space in which the linkage will operate, n is the number of links, and c_i is the constraint imposed by joint i. For a planar linkage $\lambda = 3$.

The joint degree of freedom is related to the constraint imposed by the joint per Eqn. 2.2.

$$f_i = \lambda - c_i \tag{2.2}$$

The relationship between the mechanism degree of freedom and the joint degree of freedom is given by the Grübler or Kutzbach criterion [48, 19] Eqn. 2.3.

$$F = \lambda(n-j-1) + \sum_{i=1}^{j} f_i \tag{2.3}$$

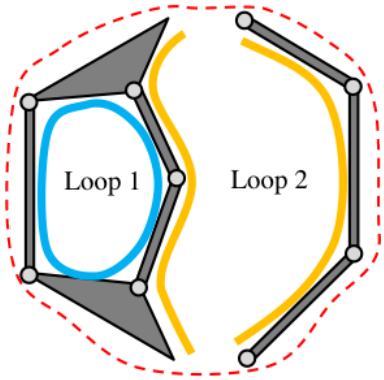


Figure 2.1: Adding a chain to a single loop creates two independent loops and three total loops.

Considering only one-DoF joints and planar motion the number of joints for a one-DoF planar linkage is given as a function of the number of links by the equation

$$j = 3n/2 - 2 \tag{2.4}$$

Fig. 2.1 shows that when a chain of links is added to the first loop the number of joints increases by one more than the number of added links and the number of independent loops increases by one. There are three total loops in Fig. 2.1 but only two independent loops.

The total number of loops is given by Euler's equation

$$\tilde{L} = j - n + 2 \tag{2.5}$$

Table 2.2: Link and joint arrangements for one degree of freedom linkages.

Mechanism	Links (n)	Joints (j)	Loops (L)	$\sum f_i$
Lever	2	1	0	1
4-Bar	4	4	1	4
6-Bar	6	7	2	7
8-Bar	8	10	3	10
10-Bar	10	13	4	13

The number of independent loops is given by

$$L = j - n + 1 \tag{2.6}$$

Plugging Eqn. 2.6 into Eqn. 2.3 produces the Loop Mobility Equation, Eqn. 2.7.

$$\sum_{i=1}^{j} f_i = F + \lambda L \tag{2.7}$$

Considering only one-DoF joints and planar motion, the quantity of independent loops is given by Eqn. 2.8.

$$L = n/2 - 1 \tag{2.8}$$

The valid link, joint and loop combinations for one-DoF linkages up to 10-bars are shown in Table 2.2

Table 2.3: Kinematic chain enumeration results up to 16 bars.

Number of Links (n)	Unique Link Assortments	Unique Topologies
4	1	1
6	1	2
8	3	16
10	7	230
12	15	6856
14	30	318 162
16	58	19 819 281

2.3 Enumeration of Kinematic Chains

The quantity of unique kinematic chains that meet Eqn. 2.4 grows rapidly with the link count. Mruthyunjaya [26] and Simoni [37] summarize the work enumerating kinematic chains of mechanisms.

The driving equations for kinematic chain enumeration are shown in Eqn. 2.9.

$$\begin{aligned} n_2 + n_3 + n_4 + ... &= n \\ 2n_2 + 3n_3 + 4n_4 + ... &= 2j \end{aligned} \tag{2.9}$$

The variable n_2, n_3, and n_4 etc. represents the quantity of links with 2, 3, 4, etc. connections and are called binary, ternary, quaternary, etc. To be physically meaningful each term of the solution to this pair of equations, n_i where $i = 1$ to $n/2$, must be zero or a positive integer. The highest order link possible in the link assortment is $n/2$, therefore, the eight-bar family does not contain any links of higher order than quaternary. The number of solutions to this equation set is the number of unique link assortments. Table 2.3 compiles the kinematic chain enumeration results from various sources, [48, 49, 43].

The four-bar and the six-bar linkage families are comprised of only one unique link assortment each, however, the eight-bar family contains three unique link assortments. The link assortment is the quantity of links that comprise the kinematic chain listed in increasing order by the number of link connections (binary, ternary, quaternary, quintenary, etc.). The notation used by Tsai [48] to express the link assortment is a single digit representing the quantity of each link type in the kinematic chain. The first eight-bar link assortment, 4400, contains four binary links and four ternary links. The second link assortment, 5210, contains five binary links, two ternary links and one quaternary link. The final link assortment, 6020, contains six binary links and two quaternary links.

Within each link assortment there may exist several ways of connecting the links in the link assortment. Each unique interconnection arrangement is called a topology and can be represented by either an adjacency graph or an adjacency matrix.

One of the challenges in kinematic chain enumeration is the detection and elimination of chains based on isomorphic graphs. Two graphs are isomorphic if there is a one-to-one correspondence between their vertices and edges that preserve their incidence [48]. For our purposes we start with the results of the enumeration for the eight-bar topologies. Tsai [48] published an atlas that includes the eight-bar family of one-DoF kinematic chains. For this research the adjacency matrix for each of these unique topologies in the atlas is the starting point. From these adjacency matrices the unique linkages are derived.

The planar one-DoF link assortments and the count of topologies for each link assortment are shown in Table 2.4 up to 10-bars, Tsai [48].

Table 2.4: Link assortments and topologies of planar one-DoF kinematic chains up to 10-bars.

Class			Link Assortment				Topology	
Loops	Bars	Joints	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	
4	10	13	4	6	0	0	50	230
			5	4	1	0	95	
			6	2	2	0	57	
			6	3	0	1	15	
			7	0	3	0	3	
			7	1	1	1	8	
			8	0	0	2	2	

2.4 Graph and Matrix Representation of Linkages

Graphs represent connectivity of objects. Graphs have far broader applications than mechanical linkages but they are used here to represent the connectivity of the links in a linkage. Connected links are called adjacent and can be represented by an adjacency graph or by an adjacency matrix.

In an adjacency graph the adjacent objects are represented by vertices which are often drawn as a dot. Connections between the adjacent vertices are connected by an edge, drawn as a line between the vertices. When a vertex has two adjacent vertices the vertex is called binary. A vertex with three adjacent vertices is called ternary. For linkages, the vertices represent the links and the edges represent the joints.

A planar graph is embedded in a plane, meaning it is drawn such that none of the edges cross over one another. A non-planar graph will have edges that cross over one another. Drawing a graph as planar or not planar does not change the connections. The planar graph should not be confused with a planar linkage. A planar linkage moves in a plane however some

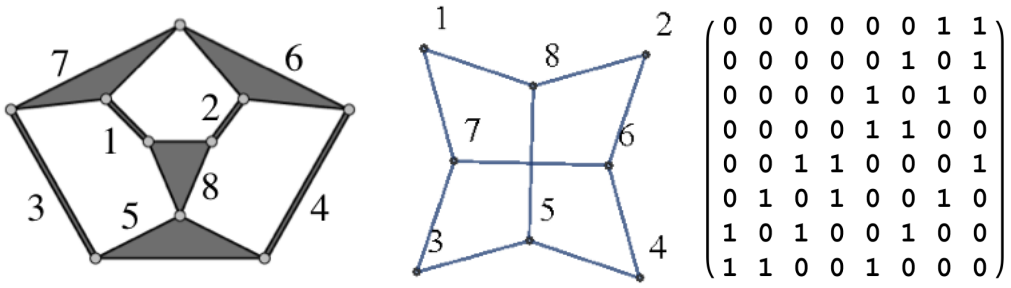


Figure 2.2: Double Butterfly eight-bar topology, adjacency graph, and adjacency matrix.

planar linkages with 10 bars or more have graphs that cannot be embedded in a plane. This distinction between the graph and the linkage has caused some confusion among researchers [1]. For the scope of this research, the eight-bar linkages can all be embedded in a plane.

An adjacency matrix numerically represents the connections between vertices. The rows and columns of the adjacency matrix represent the vertices. The numerical values indicate how many connections exist between the vertex represented by the row and the vertex represented by the column. The result is a symmetric matrix such that a non-zero value indicates when two vertices are adjacent.

For a planar one degree of freedom linkage there is only one joint between adjacent links so the adjacency matrices for any one degree of freedom linkage will contain only the numbers one and zero. If a second revolute joint existed between adjacent links, the two joints would reduce the degree of freedom of that connection to a value less than one and the two links would behave as a single rigid link. There are no connections from a link to itself therefore the diagonal of the adjacency matrix is always zero.

An example eight-bar linkage called the Double Butterfly, or Double Flyer, is shown in Fig. 2.2. The image on the left is a sketch of the linkage, followed by a non-planar embedding of the adjacency graph, and followed by the adjacency matrix.

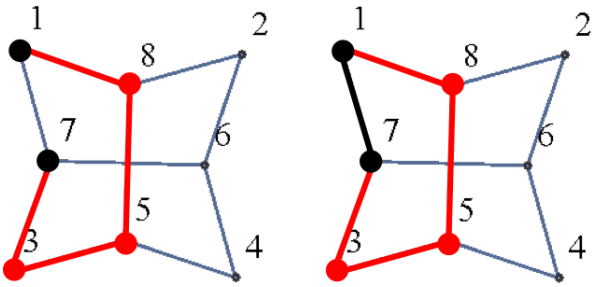


Figure 2.3: An example simple path and a cycle formed from that simple path.

2.5 Path, Spanning Tree and Non-Tree Edges

In a graph a path is defined as a set of connected edges that starts with a vertex and ends with a vertex. A simple path is such a set of connected edges with the additional constraint that the path does not repeat any vertices. The length of a simple path is the number of edges in the path. A cycle, or loop, is a path that starts with a vertex and ends with the same vertex and does not repeat any vertices. Fig. 2.3 shows an example simple path for the Double Butterfly adjacency graph and the cycle formed by closing that simple path through the edge connecting vertex 1 and 7.

A rooted spanning tree is a set of paths that start from a single vertex, the root, and connects every other vertex in the graph without repeating any vertices. A spanning tree forms no loops. There are many ways of constructing a spanning tree for a particular graph. Even from a single vertex multiple trees can often be constructed. Fig. 2.4 shows two example spanning trees for the Double Butterfly adjacency graph where both are rooted at vertex number 7.

The edges that complete a graph that are not part of the rooted spanning tree are called non-tree edges. The number of non-tree edges is the same as the number of independent loops in the graph.

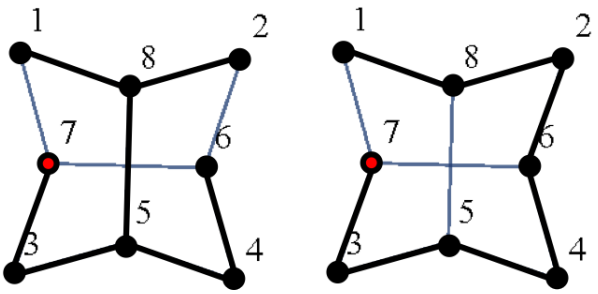


Figure 2.4: Two spanning trees for the Double Butterfly eight-bar.

2.6 Properties of Graphs for Planar Linkages

Planar one-DoF linkages are in the family of graphs called 2-vertex connected. To separate the graph into two disconnected components, two vertices must be removed. Removing one vertex leaves a connected graph. For example, removing a single link from a four-bar linkage does not separate the linkage into two linkages. Instead, the remaining links are still connected by the remaining joints. The four-bar linkage is represented by a graph comprised of one loop. Like the linkage, removing a single vertex from the associated adjacency graph does not separate the graph. The remaining vertices are still connected through the remaining edges. This graph property enables the automation to derive the linkage loop equations.

2.7 Open Ear Decomposition

2-vertex connected graphs have the property that the graph can be decomposed into a set of ears called an ear decomposition. Per Whitney [54] any non-separable graph based on a loop remains a non-separable graph with the addition of ears, also called "suspended chains". In Whitney's construction the first ear is a loop that starts and ends at a single vertex. The second ear, and higher, are simple paths whose end points are vertices that belong to earlier ears, or the first loop. One or both end points of an ear may also be an endpoint of previous ears.

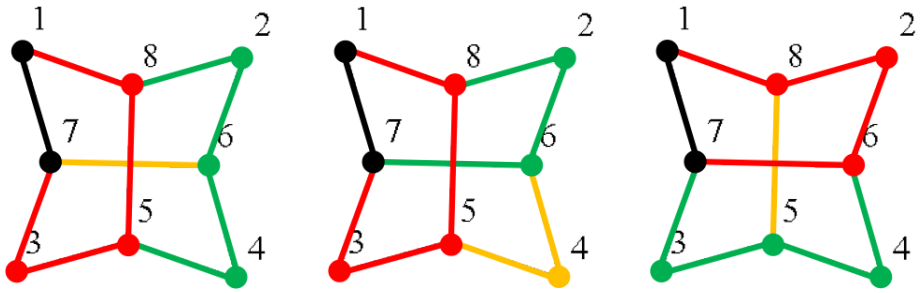


Figure 2.5: Three open ear decompositions for the Double Butterfly eight-bar.

Slightly different from the construction shown by Whitney an "open ear decomposition" of a graph is a sequence of simple paths where the first path is a single edge, each edge of the graph belongs to exactly one path, the two endpoints of each path belong to earlier paths, and none of the interior vertices of any path belong to earlier paths. An open ear decomposition for a graph has one more ear than the quantity of independent loops for the graph.

An open ear decomposition for a graph is not unique. Variations in the open ear decomposition include the selection of a different edge for the first ear and the specific selection of the edges to include in each ear. Fig. 2.5 shows three valid open ear decompositions for the Double Butterfly eight-bar linkage using the edge between vertex 1 and 7 as the first ear.

2.8 Cycle Basis

A cycle basis is 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. The cycle basis will contain every edge of a graph.

The cycles formed by adding a non-tree edge to a spanning tree form a cycle basis for the graph. The cycle basis formed in this way is called the Strictly Fundamental Cycle Basis [20].

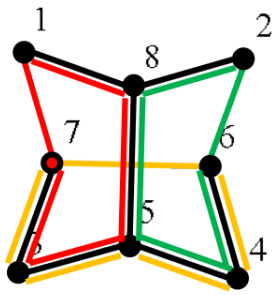


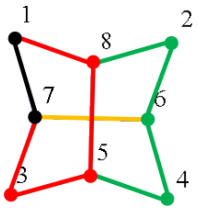
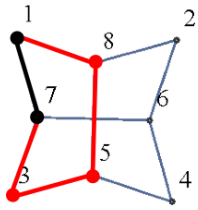
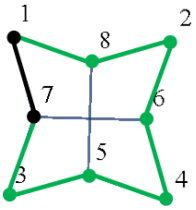
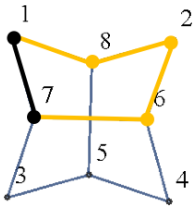
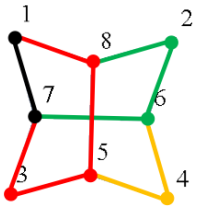
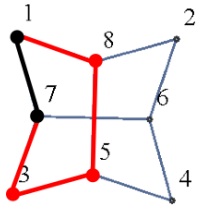
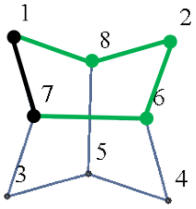
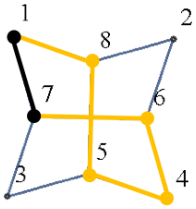
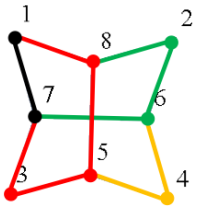
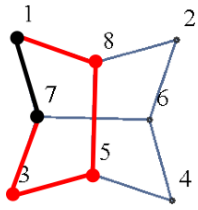
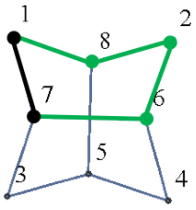
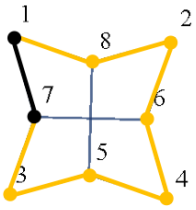
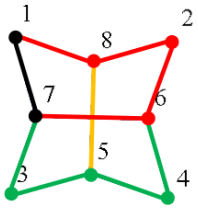
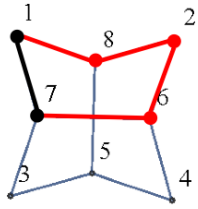
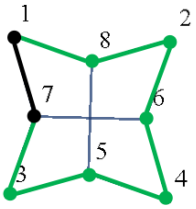
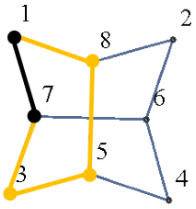
Figure 2.6: A strictly fundamental cycle basis for the Double Butterfly linkage formed from a spanning tree.

Fig. 2.6 shows the strictly fundamental cycle basis formed from the first Double Butterfly spanning tree shown in Fig. 2.4.

A cycle basis for the 2-vertex connected graphs can also be obtained such that every cycle passes through a common edge. Such a 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 the next lowest ear, and repeats until arriving at the endpoints of a the first ear. The first ear of the open ear decomposition is a single edge. Appending this single edge, the first ear, to each of the paths completes each path to form a set of cycles. 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 independent 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.

A cycle basis formed from an ear decomposition for a graph is not unique. Variations in the open ear decomposition and variation in the exact edge added to an ear result in various cycle bases. Table 2.5 shows four valid cycle bases derived from three open ear decompositions of the Double Butterfly eight-bar.

Table 2.5: Four different cycle bases formed from three different open ear decompositions for the same Double Butterfly eight-bar.

Number	Ear Decomposition	Cycle Basis		
1				
2a				
2b				
3				

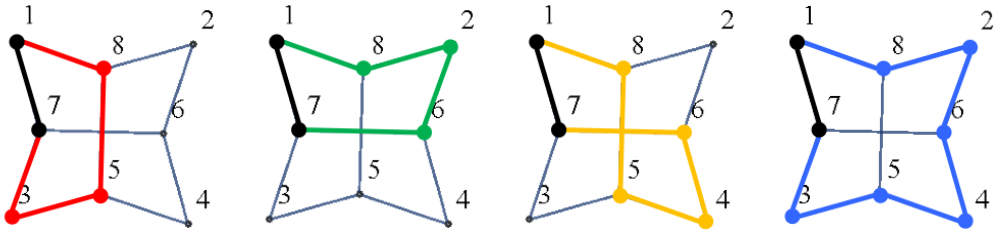


Figure 2.7: Four unique cycles for an example Double Butterfly linkage.

Because planar linkage graphs are 2-vertex connected they always have an open ear decomposition. The single edge selected for the first ear is arbitrary so it can always be selected as the edge connecting the ground vertex to the input vertex. Because these linkages always have an open ear decomposition based on the edge connecting the ground vertex to the input vertex, a cycle basis always exists such that every cycle passes through a single edge, the edge connecting the ground vertex to the input vertex.

2.9 Smallest Cycle Basis Through A Common Edge

Our automation requires that every cycle of the cycle basis must contain the edge connecting the grounded vertex to the input vertex. For many graphs there will exist more such cycles than required to form a cycle basis.

The automation produces the smallest loop equations when the cycle basis is as small as possible, meaning the cycles are as short as possible. To find the smallest cycle basis that passes through the edge connecting the ground vertex to the input vertex, we find all of the cycles that pass through that edge and select the smallest independent set as the cycle basis. An independent cycle through the graph will contain at least one edge that is not in any of the other cycles in the cycle basis.

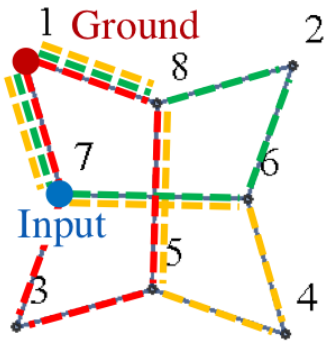


Figure 2.8: Double Butterfly cycle basis, binary ground link and ternary input link.

Besides those shown in Table 2.5 there are other open ear decompositions based on the edge connecting vertices 1 and 7, however, there are only four unique cycles that contain the edge connecting vertices 1 and 7. In Fig. 2.7 the four unique cycles are shown. These four cycles are unique but not independent.

To select the smallest cycle basis we find all of the cycles that contain the edge connecting the ground vertex to the input vertex and order them by length and then by vertex degree. The vertex degree is established in the order of the vertices along the cycle and each cycle is ordered such that the first vertex is ground, the second vertex is the ground-connected input vertex, and the last vertex is ground. Each cycle that contributes at least one new edge is an independent cycle, therefore, beginning with the smallest cycle each cycle that contributes at least one new edge is added to the cycle basis until a sufficient quantity of cycles is found. Of course once the sufficient quantity is found, no larger cycle will contribute a new edge since by definition the cycle basis contains every edge in the graph.

For the Double Butterfly linkage with a binary ground link and a ternary ground-connected input link, the four unique cycles are sorted in the appropriate order in Table 2.6. The first three cycles form the smallest cycle basis and the edges that each cycle contributes is also shown. Shown in Fig. 2.8 is the smallest cycle basis for the Double Butterfly linkage with

Table 2.6: Double Butterfly unique cycles and vertex degrees, binary ground link and ternary input link.

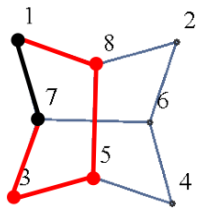
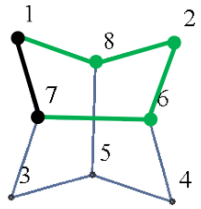
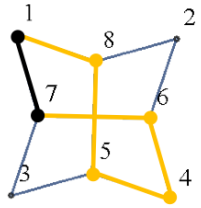
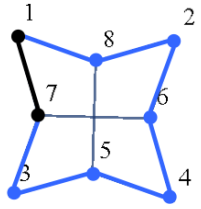
Num	Cycle	Vertex Degree List	Contributes Edges	Graph
1	{1,7,3,5,8,1}	{2,3,2,3,3,2}	{1 ↔ 7} {7 ↔ 3} {3 ↔ 5} {5 ↔ 8} {8 ↔ 1}	
2	{1,7,6,2,8,1}	{2,3,3,2,3,2}	{7 ↔ 6} {6 ↔ 2} {2 ↔ 8}	
3	{1,7,6,4,5,8,1}	{2,3,3,2,3,3,2}	{6 ↔ 4} {4 ↔ 5}	
4	{1,7,3,5,4,6,2,8,1}	{2,3,2,3,2,3,2,3,2}	None	

Table 2.7: Double Butterfly cycle basis and vertex degree list, binary ground link and ternary input link.

Loop	Cycle Basis	Vertex Degree List
1	{1,7,3,5,8,1}	{2,3,2,3,3,2}
2	{1,7,6,2,8,1}	{2,3,3,2,3,2}
3	{1,7,6,4,5,8,1}	{2,3,3,2,3,3,2}

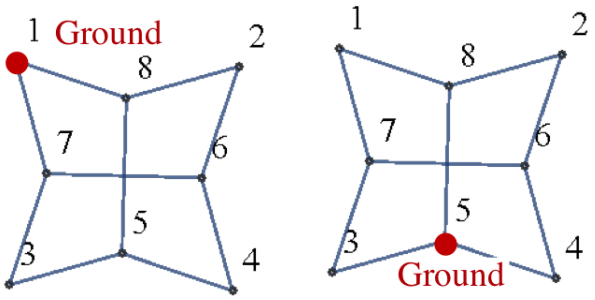


Figure 2.9: Two unique mechanisms can be derived from the Double Butterfly eight-bar topology.

a binary ground link and a ternary ground-connected input link. The loops and the vertex degrees for this cycle basis are shown in Table. 2.7.

2.10 Unique Mechanism

A mechanism is a kinematic chain with one link selected to be the grounded link. In a kinematic chain motion is described as motion of the links relative to each other while in a mechanism motion of all of the links can be determined relative to ground. Choosing different links to be ground is called linkage inversion.

For each linkage there may be several choices for the ground link that produce the same mechanism. Similar to the definition of a graph isomorphism a non-unique mechanism will have a graph with a one-to-one correspondence of vertices that preserve the incidence as well as the correspondence of the selected ground link. For example, the Double Butterfly eight-bar mechanism has only two choices for the ground vertex that are unique, Fig. 2.9. Every other selection for the ground vertex can be made into one of these two forms by renumbering the vertices.

There are 71 unique mechanisms that can be constructed from the eight-bar kinematic chain topologies [49].

2.11 Unique Linkage

A linkage is a mechanism with a selected input link. Therefore a linkage is a kinematic chain with a selected ground link and a selected input link. Since a unique mechanism depends only on the selection of the ground link and does not consider the selection of the input link, there is often more than one unique linkage that can be derived from a particular mechanism. For the automated analysis to be complete it must be able to analyze any linkage in the eight-bar kinematic chain family therefore the analysis must be able to evaluate all unique combinations of ground link and ground-connected input link.

The conditions that limit the motion of a linkage depend on which link is chosen to be ground and which link is chosen to be the input link. In this research the selection of the input link is limited to links that are connected to ground. Linkages that contain an actuation pair that is not ground connected can be analyzed by selecting one of the links of the actuation pair as the analytical ground. In this case the movement of every link relative to the actual ground link can be obtained by tracking the motion of each link relative to the "motion" of the actual ground link. For example the earth moving machinery shown in [4] contains an actuation pair between two moving links that can be analyzed by choosing one side of the actuator as the analytical ground.

Linkages where the input link is not directly connected to ground may have an actuation pair that has been omitted from the model. These linkages can be analyzed by increasing the number of bars so that the linkage now includes the actuation pair as part of the linkage. For example the robotic hand shown in [55] is modeled as an eight-bar linkage but could be analyzed as a 10-bar linkage by incorporating the prismatic actuator.

Several selections of ground and ground-connected input link may produce the same linkage. Similar to a graph isomorphism and a non-unique mechanism, the graph of a non-unique linkage will have a one-to-one correspondence of vertices that preserve the incidence as well as

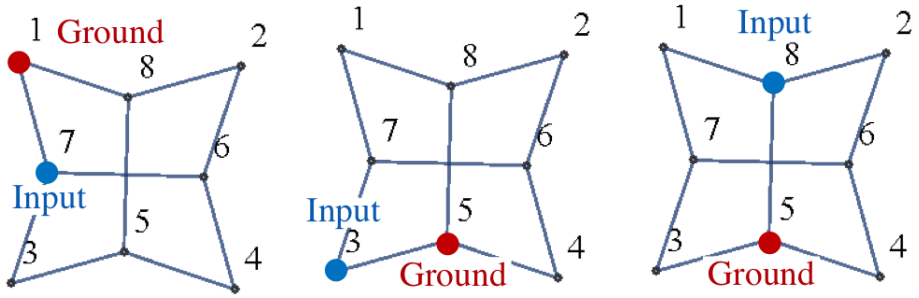


Figure 2.10: Three unique linkages can be derived from the Double Butterfly eight-bar topology.

the correspondence of the selected ground link and input link. For the Double Butterfly eight-bar linkage there are only three choices for the ground and ground-connected input vertex that are unique, Fig. 2.10. Every other selection for the ground and ground-connected input vertices can be made into one of these three forms by renumbering the vertices.

Because we use a specific cycle basis, the smallest cycle basis through the common edge connecting ground and input, the non-unique eight-bar linkages can be identified by comparing the vertex degrees along the 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. Eight-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. Defined by the ground-input common edge cycle basis every eight-bar has a unique set of three vertex degree lists.

This method of detecting a unique linkage is applicable to linkages with fewer than eight links. There is only one unique four-bar linkage. All four-bar links are binary therefore the vertex degrees along the one and only loop are all degree 2 no matter which link is selected as the ground and no matter which ground-connected link is selected as the input. This

method also properly distinguishes the six-bar family. For example, the Watt Ia and the Watt Ib both use a binary link as ground but the Watt Ia uses a binary link as the input while the Watt Ib uses a ternary link as the input. The smallest cycle bases for these two linkages have unique vertex degree lists.

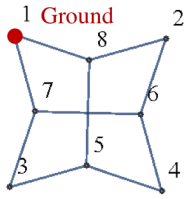
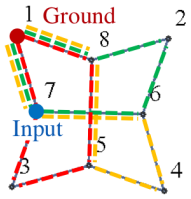
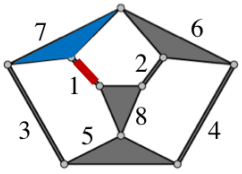
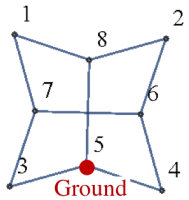
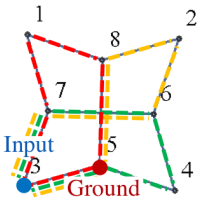
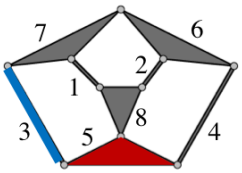
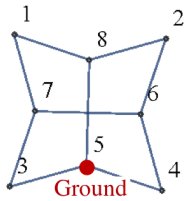
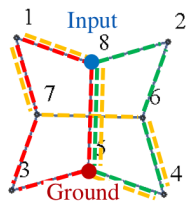
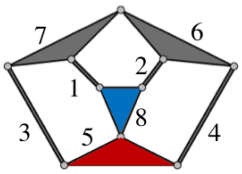
To verify that a unique set of vertex degree lists properly identifies the unique eight-bar linkages, two inspections are performed. First, an inspection of the unique vertex degree lists for the eight-bar family show that all instances of a unique set of vertex degree lists are within the same topology. If this had not been true, the loop vertex degree list method could overlook a non-unique linkage.

The second check is the connections between the loops. The ternary and higher links represent connections between the loops of the cycle basis. The arrangement of these connections should be consistent among the non-unique linkages. For example, suppose a linkage has a particular ternary link in the third position of the first and third loop while a different ternary link is in the fourth position of the same two loops. Suppose another linkage exists such that those ternary links in one of the two loops are swapped, meaning the connections between the loops are crossed. If this occurs the linkage could be a unique linkage but the vertex degree list could be the same in both cases. If this condition is possible the unique linkage could get overlooked by the vertex degree list method. An inspection of the linkages with a common set of loop vertex degree lists revealed that those linkages also have a consistent arrangement of loop-to-loop connections.

These two results may or may not be general when evaluating linkages with more links than eight but it is consistent for the linkages with four, six and eight bars. In the automation both checks have been coded so that it will flag inconsistent results that require investigation.

As examples, the two unique Double Butterfly mechanisms produce three unique Double Butterfly linkages. Shown in Table 2.8 are the loops, the unique vertex degree lists, the cycle

Table 2.8: Three unique cycle bases through a common edge can be derived for the Double Butterfly eight-bar topology.

Mechanism	Loops	Vertex Degrees	Cycle Basis	Linkage Sketch
	{1,7,3,5,8,1} {1,7,6,2,8,1} {1,7,6,4,5,8,1}	{2,3,2,3,3,2} {2,3,3,2,3,2} {2,3,3,2,3,3,2}		
	{5,3,7,1,8,5} {5,3,7,6,4,5} {5,3,7,6,2,8,5}	{3,2,3,2,3,3} {3,2,3,3,2,3} {3,2,3,3,2,3,3}		
	{5,8,1,7,3,5} {5,8,2,6,4,5} {5,8,1,7,6,4,5}	{3,3,2,3,2,3} {3,3,2,3,2,3} {3,3,2,3,3,2,3}		

bases, and the linkage sketches. Within this topology every selection of ground vertex and ground-connected input vertex fits one of these three forms and can be made into this exact set of loops by renumbering the vertices.

2.12 Shortest Path Algorithms

One way to identify all of the cycles through the edge connecting the ground vertex to the input vertex is by finding the shortest paths from the input vertex through unique paths in the graph and back to the ground vertex.

There are several means of identifying the shortest path between two vertices. One method published by Floyd [15] referencing Stephen Warshall [53] can handle negative edge distances. In our graphs the edges represent joints. Since all of the joints in our linkages are revolute joints they are all represented in the adjacency graph by an edge with a distance of one. Because all of the graph edges have a positive distance Dijkstra's shortest path algorithm [7] is sufficient for finding the shortest distance between two vertices. Mathematica has a built-in shortest path function that can be constrained to use Dijkstra's algorithm but the default settings are also suitable.

Dijkstra's shortest path algorithm establishes the shortest path between two vertices P and Q. The algorithm requires that the distance between vertices be positive but the distance can be direction dependent. The algorithm determines the shortest distance between a starting vertex P and all of the other vertices until the ending vertex Q is reached. Because there are multiple paths between the starting vertex P and an intermediate vertex R the shortest path from P to R is retained.

The vertices are placed in three sets.

- Set A (known): Vertices for which the path of minimum length from P is known.
- Set B (candidates): Vertices that are connected to a vertex in Set A but are not part of Set A.
- Set C (unknown): The remaining vertices.

The edges of the graph are placed in three sets.

- Set I (known): Edges that are part of a shortest path to a vertex.
- Set II (candidates): Edges that connect a vertex in Set A to a vertex in Set B.
- Set III (unknown): The remaining edges.

The candidates represent the best path found so far. Once the shortest path to a candidate vertex is known the vertex and the associated edge are transferred to the known sets. The shortest path found so far is a combination of the known shortest path to a known vertex plus a path through one candidate edge in Set II.

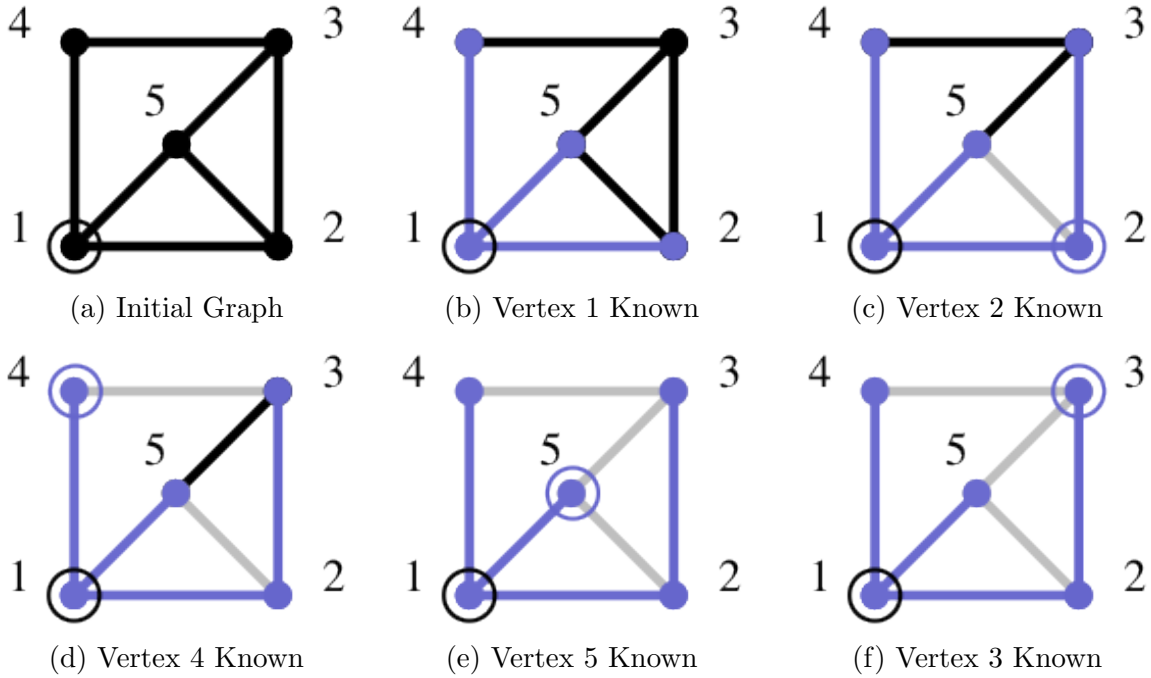
Initially all vertices are in Set C (unknown) and all edges are in Set III (unknown) and the algorithm is initiated by transferring vertex P to Set A (known).

Step 1: Consider all edges connecting vertices in Sets B or C to the vertex just transferred to Set A. If the connected vertex R belongs to Set B (candidates) and the associated edge r provides a shorter path from P to R than the best-so-far path, replace the edge in Set II with this new edge. If the connected vertex R belongs to Set B but does not provide a shorter path from P to R than the best-so-far path, the edge is rejected because it is not part of a shortest path to a vertex. If the connected vertex R is in Set C, transfer the vertex to Set B and transfer the associated edge to Set II.

Step 2: The next closest candidate vertex to P represents the shortest path from P. The next closest vertex to P is the shortest path through Set I (known) paths and the Set II (candidate) paths. Transfer to Set A (known) the vertex with the minimum distance from P and transfer the associated edge to Set I (known).

Repeat Steps 1 and 2 until vertex Q is added Set A (known). Keeping track of the vertices along the shortest path from P to Q provides the vertex list along the shortest path.

To demonstrate the algorithm an example of the procedure is shown in Fig. 2.11. The goal is to find the shortest path from the Root vertex, vertex 1, to the Destination vertex, vertex 3, for the graph shown in Fig. 2.11a using Dijkstra's algorithm. The algorithm starts with the Root vertex and adds it to the list of vertices with a "Known" distance, Fig. 2.11b. Next the algorithm finds the shortest distance to each adjacent vertex and identifies the adjacent vertices as "Candidates". The candidate that is closest to the Root vertex is selected as the next "Known" vertex. If several vertices are equally close to the Root, one is selected arbitrarily as the shortest path. In the example, Fig. 2.11c, there are three vertices equally close to the root and vertex 2 is selected as "Known". Through this closest adjacent "Known" vertex the algorithm finds the next set of closest "Candidate" vertices and rejects those that do not provide a shorter path to the root. The path $\{1, 2, 5\}$ is length 2 while the current shortest path $\{1, 5\}$ is length 1. Since this new path is not shorter than the current best path for vertex 5 the edge $\{2, 5\}$ is rejected and shown as gray in Fig. 2.11d. The remaining "Candidates" are combined and the process repeats by selecting the next vertex closest to the root vertex. In the example vertex 4 is selected next, Fig. 2.11d. The path $\{1, 4, 3\}$ is not shorter than the current shortest path $\{1, 2, 3\}$ to vertex 3 therefore edge $\{4, 3\}$ is rejected. Vertex 5, Fig. 2.11e, is selected next. The path $\{1, 5, 3\}$ is not shorter than the current best path $\{1, 2, 3\}$ to vertex 3 therefore edge $\{5, 3\}$ is rejected. Finally vertex 3 is added to the list of "Known" vertices signaling the end of the algorithm and the shortest path is length 2 through vertices $\{1, 2, 3\}$, Fig. 2.11f



(a) Initial Graph (b) Vertex 1 Known (c) Vertex 2 Known

(d) Vertex 4 Known (e) Vertex 5 Known (f) Vertex 3 Known

Figure 2.11: Dijkstra's shortest path algorithm.

2.13 Loop Equations

Portions of this section are from Parrish and McCarthy [31], used with permission.

Linkage loop equations describe the closure conditions for assembling a linkage. Following a loop the sum of the linkage features projected onto the x and y axis must sum to zero for a linkage to assemble.

The Watt I six-bar linkage has two independent loops which yield four component equations. Using the linkage geometry defined in Fig. 2.12 the component equations for the first loop through the joints ptO, ptA, ptB, ptC, ptO are

$$\begin{aligned} l_1 \cos\theta_1 + b_1 \cos(\theta_2 - \gamma) - b_2 \cos(\theta_4 + \eta) - l_0 \cos\theta_0 = 0, \\ l_1 \sin\theta_1 + b_1 \sin(\theta_2 - \gamma) - b_2 \sin(\theta_4 + \eta) - l_0 \sin\theta_0 = 0. \end{aligned} \tag{2.10}$$

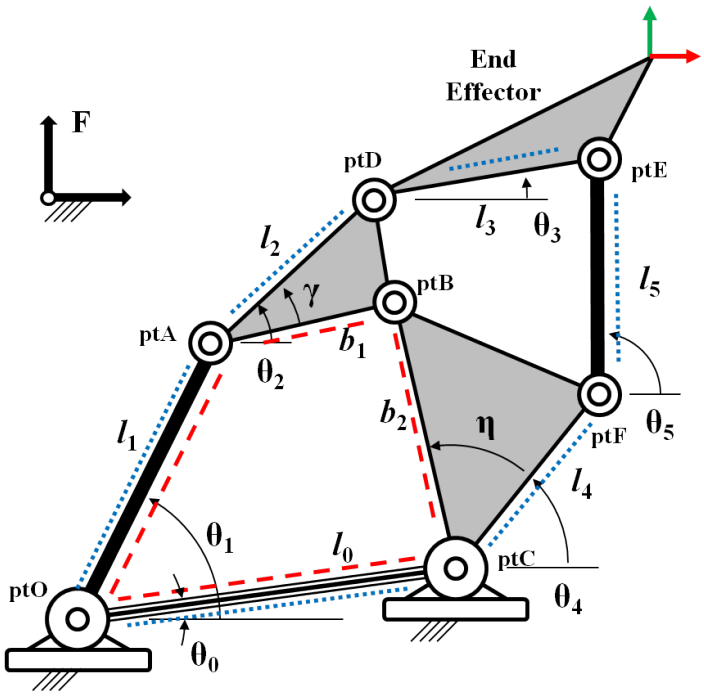


Figure 2.12: Watt I linkage has two independent loops through a common edge.

The component equations for the second loop through the joints ptO, ptA, ptD, ptE, ptF, ptC, ptO are

$$\begin{aligned} l_1 \cos\theta_1 + l_2 \cos\theta_2 + l_3 \cos\theta_3 - l_4 \cos\theta_4 - l_5 \cos\theta_5 - l_0 \cos\theta_0 = 0, \\ l_1 \sin\theta_1 + l_2 \sin\theta_2 + l_3 \sin\theta_3 - l_4 \sin\theta_4 - l_5 \sin\theta_5 - l_0 \sin\theta_0 = 0. \end{aligned} \quad . \tag{2.11}$$

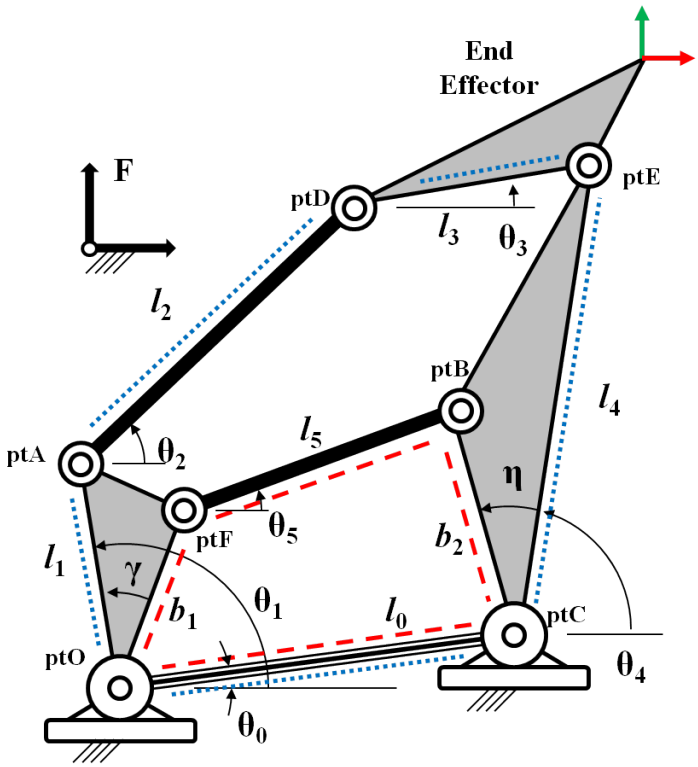


Figure 2.13: Stephenson I linkage has two independent loops through a common edge.

Using the linkage geometry defined in Fig. 2.13, the Stephenson I linkage component equations for the first loop through the joints ptO, ptF, ptB, ptC, ptO are

$$\begin{aligned} b_1\cos(\theta_1-\gamma)+l_5\cos\theta_5-b_2\cos(\theta_4+\eta)-l_0\cos\theta_0=0,\\ b_1\sin(\theta_1-\gamma)+l_5\sin\theta_5-b_2\sin(\theta_4+\eta)-l_0\sin\theta_0=0. \end{aligned} \tag{2.12}$$

The second loop component equations through the joints ptO, ptA, ptD, ptE, ptC, ptO are

$$\begin{aligned} l_1\cos\theta_1+l_2\cos\theta_2+l_3\cos\theta_3-l_4\cos\theta_4-l_0\cos\theta_0=0,\\ l_1\sin\theta_1+l_2\sin\theta_2+l_3\sin\theta_3-l_4\sin\theta_4-l_0\sin\theta_0=0. \end{aligned} \tag{2.13}$$

2.14 Singularities and the Jacobian

Portions of this section are from Parrish and McCarthy [31], used with permission.

Singularities can occur when two adjacent link features are collinear, such as when a four-bar sub-linkage is actuated through two of the bars and the two free links become collinear. Singularities can also occur when the lines along three link features have a common intersection point, such as when a four-bar sub-linkage is actuated by a dyad such that the line of action passes through the instantaneous roll center of the four-bar coupler link. The singularity conditions can be found through the Jacobian.

The singularities of a linkage depend on which link is selected as ground and which link is selected as the ground-connected input link. The remaining links are all output links. The Jacobian represents the rate of change of the output angles with respect to each other and is formed by taking the derivative of each of the loop equations with respect to each of the output angles and arranging the derivatives in matrix form. Each column is the derivative with respect to a particular output angle and each row is the loop equation for which the derivative is taken. To identify the singularities the determinant of the Jacobian is solved for the angles that cause the determinant to be zero.

Chase and Mirth [3] provide definitions which we apply to describe the criteria for smooth movement. A linkage that moves smoothly must remain on the same circuit, meaning, it must be able to reach all of the task positions using a range of continuous input angles such that disassembly of the linkage is not required. It must also remain on the same branch of the circuit, meaning, it must not pass through any stationary configurations, sometimes called singularities.

A linkage will have smooth movement when a consistent set of signs for the determinants exists throughout the continuous range of input angles required to reach all of the task

positions,

$$\theta_{1,start} < \theta_{1,k} < \theta_{1,end} \tag{2.14}$$

where k is an index of the input angle.

To analyze the linkage we construct the Jacobian from the loop equations and factor the determinant of the Jacobian into the determinant of individual 2x2 block matrices along the diagonal. This factoring enables not only the identification of singular configurations but also the source of the singular configuration. When one of the factors contains only two link angles, the singularity occurs when there is a collinear condition. When one of the factors contains three link angles, the singularity occurs when the lines along three link features intersect at a common point. When one of these 2x2 determinants is zero an undesirable stationary configuration has been reached. For an alternate approach that is applicable to the Watt I linkage see Parrish and McCarthy [30].

When a matrix is in block triangular form, Eqn. 2.15, the determinant factors as Eqn. 2.16. This result scales with the size of the matrix such that the determinant of a 6x6 block triangular matrix is the product of the three 2x2 blocks along the diagonal.

$$F = \begin{bmatrix} A & 0 \\ C & D \end{bmatrix} \tag{2.15}$$

A, C and D are 2x2 sub-matrices.

$$|F| = |A|\,|D| \tag{2.16}$$

If the determinant is not in block triangular form we can convert the Jacobian to block upper triangular form through a determinant preserving transform as shown by Silvester [36], Eqn. 2.17. This factors the determinant of the Jacobian into the determinant of the individual 2x2 block matrices along the diagonal. Since the transform involves an inverse of sub-matrix D the columns of the Jacobian may need to be sorted so that block D is full rank. Although this sorting can change the overall sign of the determinant of the Jacobian, we are interested in the determinants of the 2x2 blocks which remain unchanged after the columns are sorted.

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} I & 0 \\ -D^{-1}C & I \end{bmatrix} = \begin{bmatrix} A - BD^{-1}C & B \\ 0 & D \end{bmatrix} \tag{2.17}$$

The determinant of each of these 2x2 block matrices is assigned the name J_i, $i = 1, \ldots, n/2$.

There are linkages, such as the Stephenson III six-bar linkage, where specific geometric dimensions can allow a link to rotate more than 360 degrees before encountering a singularity [3]. For such linkages the factored Jacobian alone may not be sufficient to track the solutions uniquely because within certain ranges of input there will exist two assembly configurations with the same sign list.

We apply this now to a specific example. For the Watt I linkage shown in Fig. 2.12 we have selected θ_1 as our input parameter and θ_0 as a fixed angle defining the orientation of the ground link. Since θ_1 and θ_0 are known, not output angles, these are eliminated from the Jacobian. The Jacobian of this Watt I six-bar linkage with a grounded binary link oriented

by θ_0 and a binary input link oriented by θ_1 is

$$\begin{bmatrix} -b_1\sin(\theta_2-\gamma) & b_2\sin(\theta_4+\eta) & 0 & 0 \\ b_1\cos(\theta_2-\gamma) & -b_2\cos(\theta_4+\eta) & 0 & 0 \\ -l_2\sin\theta_2 & l_4\sin\theta_4 & -l_3\sin\theta_3 & l_5\sin\theta_5 \\ l_2\cos\theta_2 & -l_4\cos\theta_4 & l_3\cos\theta_3 & -l_5\cos\theta_5 \end{bmatrix} \tag{2.18}$$

We recognize that the Jacobian is already in the block triangular form. The upper left diagonal block of Eqn. 2.18 represents the singularity conditions for the four-bar loop, when $(\theta_2-\gamma)$ and $(\theta_4+\eta)$ are collinear. The lower right diagonal block represents the singularity conditions of the second loop, when θ_3 and θ_5 are collinear. The value of the determinant of the Jacobian for the entire linkage is 0 under either of these conditions. Therefore we can separate singularities of the four-bar loop from the singularities of the second loop.

J_1 is the determinant of the upper left diagonal block.

$$J_1 = \begin{vmatrix} -b_1\sin(\theta_2-\gamma) & b_2\sin(\theta_4+\eta) \\ b_1\cos(\theta_2-\gamma) & -b_2\cos(\theta_4+\eta) \end{vmatrix} \tag{2.19}$$

J_2 is the determinant of the lower right diagonal block.

$$J_2 = \begin{vmatrix} -l_3\sin\theta_3 & l_5\sin\theta_5 \\ l_3\cos\theta_3 & -l_5\cos\theta_5 \end{vmatrix} \tag{2.20}$$

We apply this to another example. For the Stephenson I linkage described in Fig. 2.13 the Jacobian is

$$\begin{bmatrix} b_2\sin(\theta_4+\eta) & -l_5\sin\theta_5 & 0 & 0 \\ -b_2\cos(\theta_4+\eta) & l_5\cos\theta_5 & 0 & 0 \\ l_4\sin\theta_4 & 0 & -l_2\sin\theta_2 & -l_3\sin\theta_3 \\ -l_4\cos\theta_4 & 0 & l_2\cos\theta_2 & l_3\cos\theta_3 \end{bmatrix} \tag{2.21}$$

This Jacobian is block diagonal and can be factored into J_1 and J_2 enabling us to separate the singularity conditions of the first loop from the singularity conditions of the second loop. J_1 is singular when the links located by $(\theta_4+\eta)$ and θ_5 are collinear. J_2 is singular when the links located by θ_2 and θ_3 are collinear.

$$J_1 = \begin{vmatrix} b_2\sin(\theta_4+\eta) & -l_5\sin\theta_5 \\ -b_2\cos(\theta_4+\eta) & l_5\cos\theta_5 \end{vmatrix} \tag{2.22}$$

$$J_2 = \begin{vmatrix} -l_2\sin\theta_2 & -l_3\sin\theta_3 \\ l_2\cos\theta_2 & l_3\cos\theta_3 \end{vmatrix} \tag{2.23}$$

2.15 Dixon Determinant

Portions of this section are from Parrish and McCarthy [31], used with permission.

For our evaluation we need to solve the loop equations for all of the linkage configurations available for a given input angle. There are many approaches to solving the linkage loop equations. We choose to solve the loop equations using the Dixon Determinant method

described by Wampler [51] and shown as applied here in detail in both McCarthy and Soh [23] and Parrish and McCarthy [30]. The Dixon Determinant provides all of the assembly configurations for a given input angle.

The loop equations are formulated as vector loop equations in the complex plane. The y direction is taken to be along the imaginary axis so we multiply both y component equations by i which is defined as $i^2 = -1$. We transform the loop equations into complex form by applying trigonometric identities Eqn. 2.24 and exponential identities Eqn. 2.25.

$$\begin{aligned}
\cos(A+B) &= \cos A \cos B - \sin A \sin B \\
\sin(A+B) &= \sin A \cos B + \cos A \sin B \\
\cos(A-B) &= \cos A \cos B + \sin A \sin B \\
\sin(A-B) &= \sin A \cos B - \cos A \sin B
\end{aligned} \tag{2.24}$$

$$\begin{aligned}
e^{i\theta_j} &= \cos\theta_j + i \sin\theta_j \\
e^{-i\theta_j} &= \cos\theta_j - i \sin\theta_j
\end{aligned} \tag{2.25}$$

Taking the complex conjugate of each equation produces the necessary quantity of independent complex equations to enable a solution.

The Dixon determinant is formulated as a matrix Δ that is constructed from the vector loop equations. The first entry in the top row is the vector loop equation for the first loop, the

second entry is the conjugate, the third is the vector loop equation for the second loop, and the fourth is the conjugate. The remaining rows of the matrix Δ are the same equations as the first row except the variables we wish to solve, Θ_j, are sequentially replaced with temporary variables α_j. The final matrix is square.

In each of the matrix entries one variable, an output angle Θ_n, is selected to be the eliminant and is treated as part of the constants. For example choosing Θ_3 as the eliminant and Θ_2, Θ_4 and Θ_5 as the variables we wish to solve, the Dixon determinant for a six-bar linkage is formed per Eqn. 2.26 where Θ_2, Θ_4 and Θ_5 are replaced successively with temporary variables α_2, α_4 and α_5 in each of the successive rows.

$$\begin{vmatrix} F_1(\Theta_2,\Theta_4,\Theta_5) & F_1^*(\bar{\Theta}_2,\bar{\Theta}_4,\bar{\Theta}_5) & F_2(\Theta_2,\Theta_4,\Theta_5) & F_2^*(\bar{\Theta}_2,\bar{\Theta}_4,\bar{\Theta}_5) \\ F_1(\alpha_2,\Theta_4,\Theta_5) & F_1^*(\bar{\alpha_2},\bar{\Theta}_4,\bar{\Theta}_5) & F_2(\alpha_2,\Theta_4,\Theta_5) & F_2^*(\bar{\alpha_2},\bar{\Theta}_4,\bar{\Theta}_5) \\ F_1(\alpha_2,\alpha_4,\Theta_5) & F_1^*(\bar{\alpha_2},\bar{\alpha_4},\bar{\Theta}_5) & F_2(\alpha_2,\alpha_4,\Theta_5) & F_2^*(\bar{\alpha_2},\bar{\alpha_4},\bar{\Theta}_5) \\ F_1(\alpha_2,\alpha_4,\alpha_5) & F_1^*(\bar{\alpha_2},\bar{\alpha_4},\bar{\alpha_5}) & F_2(\alpha_2,\alpha_4,\alpha_5) & F_2^*(\bar{\alpha_2},\bar{\alpha_4},\bar{\alpha_5}) \end{vmatrix} \tag{2.26}$$

For eight-bar linkages there are two more unknown angles and three total loops. The Dixon determinant for an eight-bar has two more columns and two more rows representing the third loop.

Performing row operations subtracting Row 2 from Row 1, Row 3 from Row 2, and Row 4 from Row 3 cancels terms in Row 1 through Row 3 that do not contain the variables Θ_2, Θ_4, Θ_5 and the associated α_2, α_4, α_5. Last, we factor out the extraneous roots where $\Theta_j = \alpha_j$ by applying the relationship $\Theta_j - \alpha_j = -\Theta_j\alpha_j(\bar{\Theta}_j - \bar{\alpha_j})$

Expand this determinant to obtain a polynomial in terms of the variables Θ_2, Θ_4, Θ_5 and the associated α_2, α_4, α_5. This polynomial can be expressed in matrix form

$$\delta = \mathbf{a}^T [W] \mathbf{t} \tag{2.27}$$

where

$$\begin{aligned} \mathbf{a} &= (\alpha_2, \alpha_4, \alpha_5, \alpha_2\alpha_4, \alpha_4\alpha_5, \alpha_2\alpha_5)^T, \\ \mathbf{t} &= (\Theta_2, \Theta_4, \Theta_5, \Theta_2\Theta_4, \Theta_4\Theta_5, \Theta_2\Theta_5)^T. \end{aligned} \tag{2.28}$$

The output angle chosen to be the eliminant, Θ_3 in this example, is part of the matrix W. The values $\mathbf{t}$ that satisfy the loop equations cause the Dixon determinant to be zero independent of the values of $\mathbf{a}$. Thus, the configurations $\mathbf{t}$ satisfy

$$[W] \mathbf{t} = 0. \tag{2.29}$$

The matrix W can be separated into two square matrices. One matrix is the coefficients of the terms containing the eliminant, Θ_3 in this example, and the other matrix is the terms that do not contain the eliminant. This yields the generalized eigenvalue problem

$$[M\Theta_3 - N] \mathbf{t} = 0, \tag{2.30}$$

The square matrices M and N contain only constants defined by the linkage dimensions, the ground angle Θ_0 and its conjugate $\bar{\Theta}_0$, and the input angle Θ_1 and its conjugate $\bar{\Theta}_1$.

The solutions for $\mathbf{t}$ of this generalized eigenvalue problem yield the output angles, for this example Θ_j $j = 2, 3, 4, 5$, that define the configuration of a six-bar linkage. The eliminant, Θ_3 in this example, is solved as the eigenvalue while Θ_j $j = 2, 4, 5$ are solved as the eigenvector. Equation Eqn. 2.30 can have as many as six roots, $\mathbf{t}_i$ $i = 1, \ldots, 6$, which means that for a given input value Θ_1 there can be as many as six assembly configurations for the six-bar linkage.

The final step of the process must resolve any scaling factors the come with the eigenvector solution. The ratio of terms in $\mathbf{t}$ is taken to produce the true value of each angle Θ_j $j = 2, 4, 5$. For example, the true value for Θ_2 is not the first term of $\mathbf{t}$, rather, it is the term $\Theta_2\Theta_4$ divided by the term Θ_4. This ratio cancels the scaling factor and the value of Θ_4. Any ratio of two terms will cancel the scaling factor so it would be equally valid to solve for Θ_2 by taking the ratio of $\Theta_2\Theta_5$ and Θ_5.

Applying the Dixon determinant derivation to the Watt I six-bar linkage, we convert Eqn. 2.10 and Eqn. 2.11 into vector loop equations

$$
\begin{aligned}
F_1 : &\quad l_1\Theta_1 + b_1\Theta_2 e^{-i\gamma} - b_2\Theta_4 e^{i\eta} - l_0\Theta_0 = 0, \\
F_1^* : &\quad l_1\bar{\Theta}_1 + b_1\bar{\Theta}_2 e^{i\gamma} - b_2\bar{\Theta}_4 e^{-i\eta} - l_0\bar{\Theta}_0 = 0, \\
F_2 : &\quad l_1\Theta_1 + l_2\Theta_2 + l_3\Theta_3 - l_4\Theta_4 - l_5\Theta_5 - l_0\Theta_0 = 0, \\
F_2^* : &\quad l_1\bar{\Theta}_1 + l_2\bar{\Theta}_2 + l_3\bar{\Theta}_3 - l_4\bar{\Theta}_4 - l_5\bar{\Theta}_5 - l_0\bar{\Theta}_0 = 0.
\end{aligned}
\tag{2.31}
$$

For the Watt I the Dixon Determinant is shown in Eqn. 2.32.

$$\Delta = \begin{vmatrix} -b_1 e^{-i\gamma}\Theta_2\alpha_2 & b_1 e^{i\gamma} & -l_2\Theta_2\alpha_2 & l_2 \\ b_2 e^{i\eta}\Theta_4\alpha_4 & -b_2 e^{-i\eta} & l_4\Theta_4\alpha_4 & -l_4 \\ 0 & 0 & l_5\Theta_5\alpha_5 & -l_5 \\ r_{41} & r_{42} & r_{43} & r_{44} \end{vmatrix} = 0 \tag{2.32}$$

where

$$\begin{aligned} r_{41} &= l_1\Theta_1 + b_1\alpha_2 e^{-i\gamma} - b_2\alpha_4 e^{i\eta} - l_0\Theta_0, \\ r_{42} &= l_1\bar{\Theta_1} + b_1\bar{\alpha_2} e^{i\gamma} - b_2\bar{\alpha_4} e^{-i\eta} - l_0\bar{\Theta_0}, \\ r_{43} &= l_1\Theta_1 + l_2\alpha_2 + l_3\Theta_3 - l_4\alpha_4 - l_5\alpha_5 - l_0\Theta_0, \\ r_{44} &= l_1\bar{\Theta_1} + l_2\bar{\alpha_2} + l_3\bar{\Theta_3} - l_4\bar{\alpha_4} - l_5\bar{\alpha_5} - l_0\bar{\Theta_0}. \end{aligned} \tag{2.33}$$

Applying the Dixon determinant derivation to the Stephenson I six-bar linkage, we convert Eqn. 2.12 and Eqn. 2.13 into vector loop equations

$$\begin{aligned} F_1 : &\quad b_1\Theta_1 e^{-i\gamma} + l_5\Theta_5 - b_2\Theta_4 e^{i\eta} - l_0\Theta_0 = 0, \\ F_1^* : &\quad b_1\bar{\Theta_1} e^{i\gamma} + l_5\bar{\Theta_5} - b_2\bar{\Theta_4} e^{-i\eta} - l_0\bar{\Theta_0} = 0, \\ F_2 : &\quad l_1\Theta_1 + l_2\Theta_2 + l_3\Theta_3 - l_4\Theta_4 - l_0\Theta_0 = 0, \\ F_2^* : &\quad l_1\bar{\Theta_1} + l_2\bar{\Theta_2} + l_3\bar{\Theta_3} - l_4\bar{\Theta_4} - l_0\bar{\Theta_0} = 0. \end{aligned} \tag{2.34}$$

For the Stephenson I the Dixon Determinant is shown in Eqn. 2.35.

$$\Delta = \begin{vmatrix} 0 & 0 & -l_2\Theta_2\alpha_2 & l_2 \\ b_2e^{i\eta}\Theta_4\alpha_4 & -b_2e^{-i\eta} & l_4\Theta_4\alpha_4 & -l_4 \\ -l_5\Theta_5\alpha_5 & l_5 & 0 & 0 \\ r_{41} & r_{42} & r_{43} & r_{44} \end{vmatrix} = 0 \tag{2.35}$$

where

$$\begin{aligned} r_{41} &= l_5\alpha_5 + b_1\Theta_1e^{-i\gamma} - b_2\alpha_4e^{i\eta} - l_0\Theta_0, \\ r_{42} &= l_5\bar{\alpha_5} + b_1\bar{\Theta}_1e^{i\gamma} - b_2\bar{\alpha_4}e^{-i\eta} - l_0\bar{\Theta}_0, \\ r_{43} &= l_1\Theta_1 + l_2\alpha_2 + l_3\Theta_3 - l_4\alpha_4 - l_0\Theta_0, \\ r_{44} &= l_1\bar{\Theta}_1 + l_2\bar{\alpha_2} + l_3\bar{\Theta}_3 - l_4\bar{\alpha_4} - l_0\bar{\Theta}_0. \end{aligned} \tag{2.36}$$

The final solution is obtained by taking this determinant to form the polynomial δ and constructing the matrix W by gathering the coefficients of the terms containing the monomials of $\mathbf{a}$ and $\mathbf{t}$ per the form of Eqn. 2.27. Matrix W is then separated into matrices M and N by gathering the coefficients of the eliminant. The numerical values of the linkage features, input angle, and ground angle are provided and the generalized eigenvalues and eigenvectors are solved numerically using an eigenvalue solver. For each eigenvector the ratio of the eigenvector elements is taken to provide the numerical solution for the output angles. After converting the solutions back to real values, each of the real solutions represent a valid assembly configuration for the linkage.

For these Watt I and Stephenson I linkages, with the input in the four-bar loop, the diagonal of the matrices M and N contain one zero which reduces the number of real solutions from six to four, Wampler [51].

Wampler [51] describes a minor modification to the Dixon determinant to incorporate prismatic joints. He calls them sliding joints. This modification has not been implemented in the present research.

2.16 Selection of the Eliminant

Some unknown angles are poor choices for the eigenvalue Θ_n because the resulting eigenvector $\mathbf{t}$ cannot be used to solve for all 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. This occurs when the ground link, the input link and the output link designated to be the eliminant are all part of a one-DoF sub-linkage.

The Double Butterfly linkage used by Wampler [51] to demonstrate the Dixon determinant process is unique in the eight-bar family. This linkage contains no one-DoF sub linkages therefore any output angle can be selected as the eliminant Θ_n.

2.17 Identifying Linkages That Partition

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. An inspection of these cases reveals that the linkage can be partitioned into two independent

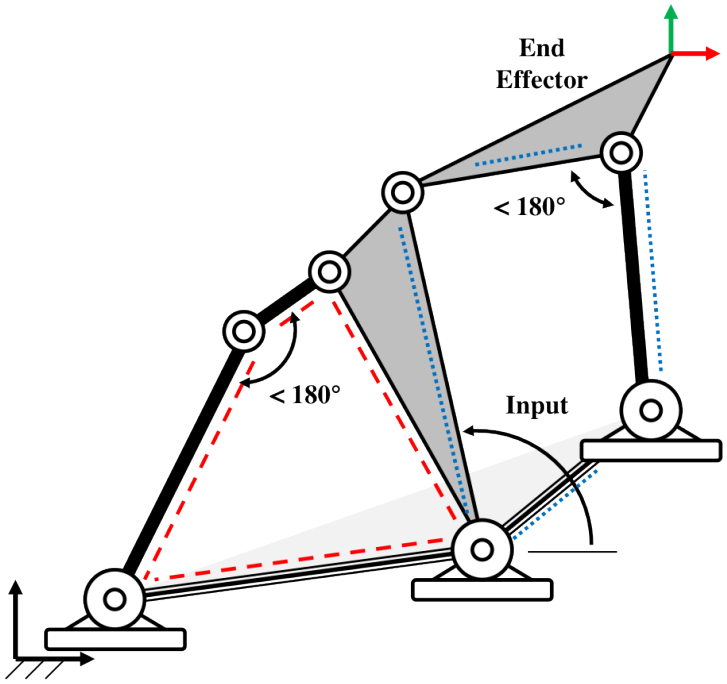


Figure 2.14: Watt IIb linkage partitions into two independent four-bar linkages.

one-DoF linkages with a single common joint, the joint between the ground link and the input link. Therefore both the ground and the input links are part of two independent one-DoF linkages. For these linkages no matter what angle is chosen for the eigenvalue Θ_n that output angle will always be a part of a one-DoF sub-linkage, therefore, the assembly configuration of the other one-DoF sub-linkage is independent.

In the six-bar family the Watt IIb, Fig. 2.14, is the one and only example of a linkage that partitions. This linkage has a grounded ternary link and the input is on a ternary link as well. The input drives two four-bar linkages through a common joint between the ground and input links. The final assembly configuration of each four-bar sub-linkage is independent of the other. Therefore each of the two four-bars can be assembled independently such that the two free links have a relative angle of either less than 180 degrees or greater than 180 degrees.

In the cases that partition, the independent sub-linkages should be independently solvable using the Dixon determinant as long as the sub-linkages themselves do not also partition. If the partitioning is carried down to the lowest level that no longer partitions, the Dixon determinant should be able to be applied at that level to solve each of the elements as an independent one-DoF linkage.

2.18 Mathematical Background Conclusion

This chapter provided the necessary mathematical background for analysis of eight-bar linkages. The scope of the linkage types discussed in this research has been defined as planar linkages with revolute joints. The chapter provided the equations describing the mobility of a linkage as well as the valid quantities of links and joints required for linkages with one degree of freedom. A summary of applicable kinematic chain enumerations have been provided.

A brief summary of the graph theory used for this research has been provided. The sections included a description of the adjacency graph, adjacency matrix, paths, and spanning trees. Properties of the graphs applicable to planar linkages have been discussed along with the cycle basis. A method to find the shortest path between two vertices has been described.

To solve for the specific linkage angles the loop equations were described. The Jacobian, used to identify singular configurations, was also described and applied to two six-bar linkages.

The Dixon determinant was introduced as the means to solve for the linkage output angles for all possible assembly configurations. The Dixon determinant was applied to a two example six-bar linkages.

The proper selection of the eigenvalue for solving the Dixon determinant was discussed. Finally linkages that partition, and therefore cannot be solved as a whole linkage by the Dixon determinant, were discussed.

Chapter 3

Automated Loop Equations

To automatically formulate the linkage loop equations for any eight-bar linkage we require a procedure that will operate on any topology of kinematic chain.

3.1 Finding Loops

The process to automatically construct the linkage loops for a particular linkage begins with the adjacency graph and the user defined selection for the ground link and ground-connected input link. A present limitation is that the input link must be adjacent to, meaning connected to, the grounded link.

The first step is to establish the smallest cycle basis for the linkage through a common edge, the edge connecting the ground vertex to the input vertex. To find this cycle basis we do not base our search directly on an open ear decomposition, rather, we find the cycles directly using a shortest path algorithm.

3.1.1 Breadth-First Search for Unique Shortest Paths

The search for unique cycles seeks the shortest path from the input vertex, through the linkage, and back to ground. Independent paths are found by eliminating an edge in the current shortest path and searching for the next shortest path. The shortest paths sought in this manner can be visualized in levels of depth. Once the shortest path is found an independent path at the next depth can be found by eliminating an edge in the current path and reapplying the shortest path algorithm. The algorithm is breadth-first, meaning at each depth the algorithm evaluates all of the shortest paths before evaluating the next depth.

To generate the loops the list of edges are produced for the entire graph and the edge directly 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. This path is the first level. To find the second level of paths, an edge in the first level path is eliminated along with the one edge elimination that created that first path, the edge connecting ground and input. All edges in the first level path are eliminated one at a time, along with the edge between ground and input, to produce 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. Every edge at the second level of paths is eliminated one at a time to identify unique paths for the third level. Once every edge at the third level has been tested, the process moves to the fourth level and repeats. This process continues until all edge eliminations have been attempted and no new paths are found.

In a one-DoF linkage there is a maximum of one joint between two links, otherwise the two joined links would form a structure. Since there is one and only one edge that directly connects the ground link to the input link, the direct path between the links has a distance of one. Every other path through the linkage from the input link to ground requires more than one edge. Seeking the shortest path from input to ground using the full graph of a linkage will

always return the direct connection between the ground and the input link as the shortest path. Eliminating the edge connecting the ground to the input forms a reduced graph that represents the starting condition for the search. Running a shortest path algorithm on the reduced graph forces the algorithm to find the first shortest path from the input vertex to ground that is not the direct connection between the ground and input vertices.

For the second depth and subsequent depths the algorithm must find a shortest path that is independent of the current shortest path therefore the algorithm must be forced to avoid at least one edge in the current shortest path and all the edges that were eliminated to produce the current shortest path. This is accomplished by eliminating one edge in the current shortest path and all the edges that were eliminated to produce the current shortest path. The set of edges that are eliminated to perform this function is called an elimination set. There are several elimination sets generated at each level.

A valid elimination set will produce a reduced graph that has a path from the input vertex to the ground vertex.

Euler's equation, Eqn. 2.6, shows that there are three independent loops for every linkage in the eight-bar family. Since this algorithm successively breaks each independent loop to find the next independent loop there will not be any short paths found at the fourth level for the eight-bar linkages.

3.1.2 Multiple Shortest Paths

Not every elimination set produces a reduced graph with only one unique shortest path from input to ground. When multiple paths have the same length back to ground, arbitrarily select one of them. There is no need to perform any weighting on paths of the same length to force a particular selection, any of them may be selected. The path or paths not selected

will still be the shortest path when the selected path is broken at the next depth of search. Therefore, one of the paths not selected will be selected at the next depth of the search.

3.1.3 No Paths and the Stopping Criteria

Not every elimination set produces a reduced graph with a path from input to ground. Removing an elimination set containing two or more edges could separate the graph into two pieces. The resulting reduced graph may have no connections to either the ground or the input vertex. Alternatively the removal of the elimination set could separate the graph into two linkages somewhere in the middle of the graph away from the ground and input vertices. Either case results in no paths from the input vertex back to ground. When this occurs the elimination set is not valid and the algorithm tests the next elimination set in the elimination list.

If all elimination sets in the elimination list fail to produce a reduced graph with a path from input to ground, the algorithm stops. All paths from input to ground have been found.

3.1.4 Cycle Basis

The unique paths are collected into one set and each is made into a cycle by appending the edge connecting the ground vertex to the input vertex. The cycles are represented by their vertices. Each cycle begins with the number for the ground vertex. The second vertex is the number for the input vertex. The rest of the vertices are listed as encountered in order along the loop, ending with ground.

The cycles are sorted in the following order.

1. Loop Length

2. Vertex Number

3. Vertex Degree

To perform the third sorting the vertex degree lists are padded on the left with zeros to equalize the length of each cycle. Then the loops are sorted in ascending ordered by depth. Loops that have the same padded vertex degree list are not re-ordered in step 3 therefore the vertex number order remains as in step 2, sorted by the vertex number.

The cycle basis for the graph of a planar linkage is formed by selecting $n/2 - 1$ independent cycles. For the eight-bar there are three independent loops, or cycles.

We know that a cycle basis through a common edge always exists based on the ear decomposition theory. Since this automation has found every cycle through a common edge, there exists enough cycles within the unique cycles to form a cycle basis. To find the cycle basis the algorithm follows the same order as Whitney's ear decomposition, the smallest cycle is selected as the first ear. An independent cycle contains an edge not contained in any of the previous cycles. The next cycle in the sorted list of cycles that contains the fewest new edges not presently in the cycle basis is an independent cycle and is added to the cycle basis.

3.2 Converting Cycles to Loop Equations

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 as well as the fixed angles that represent divergence and convergence of loops on ternary and higher links. When two loops diverge or converge we only name the lines along the two loops and the

angle between them. This provides a complete geometric description of the triangle formed by the three joints. The only input needed to construct the loop equations is the cycle basis.

3.2.1 Naming Convention for Links, Joints, Lines, and Angles

The cycle basis provides an ordered list of vertices for each loop. These vertices are also the names of the linkage links which are joined in order along each loop. Following the 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 1 and link 5 is called $j1t5$. The "t" enables unambiguous distinction between links even when the link number is more than one digit.

Also following the order of the links as shown for each linkage loop, the dimension of the line 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", and followed by the last digits of the ending joint. For example, the dimension of the line on link 5 between the joints $j1t5$ and $j5t2$ is called $L1t5t2$. 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 evaluating 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. The global angle of the link is the angle from the global 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 $L1t5t2$ is part of the first loop then the angle relative to the global frame for $L1t5t2$ is called $th5$ and the origin of that angle is at the joint $j1t5$.

The vertices where the 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 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 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).

As an example we apply the naming convention to the Stephenson six-bar. The adjacency graph of the Stephenson six-bar is shown in Fig. 3.1. Choosing link 1 as ground and link 5 as input, the cycle basis is $\{1, 5, 2, 6, 1\}, \{1, 5, 4, 3, 6, 1\}$ and results in feature names shown on a sketch of the linkage in Fig. 3.2. The two loops diverge at $j1t5$ and converge at $j6t1$. Notice that the fixed angles are defined positive counter clockwise starting from the first loop and ending at the second loop. The fact that the angle defining the line $L3t6t1$ is accurately represented is shown in Fig. 3.3. The angle $th6$ defines the line $L2t6t1$ while the angle $fix2t6t1tt3t6t1$ is the fixed angle at the common joint $j6t1$ of the two loops. Transferring these angles to be about $j3t6$ shows how the summation accurately represents the angle for the line $L3t6t1$.

In quaternary links there will be three loops that pass through the link. The three loops may not all diverge from or converge at a common joint. It is possible that the quaternary

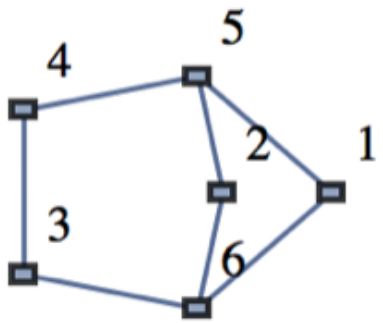


Figure 3.1: Stephenson six-bar adjacency graph for demonstrating the automation naming convention.

link will contain two loops that diverge and a third loop that converges to one of the other two. The lines that connect the joints on such a quaternary link form a "Z" on the link, Fig. 3.4. In those cases the final angle for the third link dimension is the summation of the global link angle and two fixed angles, one describing the angle between the divergent loops and the second describing the angle between the convergent loops. If all three loops do diverge from or converge at a common joint then the fixed angle for the third line can be defined relative to either of the lower two loops. However, we choose to define it relative to the second loop so that the final angle of the third loop is represented by a sum of two fixed angles, one describing the angle between the first and second divergent or convergent loops and one describing the angle between the second and third divergent or convergent loops. This makes a consistent means of representing all of the cases of divergent and convergent loops. For quintenary and higher links, links that only exist in 10-bar and higher linkages, a similar situation will exist that we believe will be adequately addressed by the summation of three fixed angles defining the orientation of each line upon which the new line is based.

A non-standard naming scheme causes variation that makes automation difficult. The same Stephenson six-bar linkage was shown in Fig. 2.13 using arbitrary names [31]. The variation in the link names, angle names, fixed angle names, and the placement of the angles

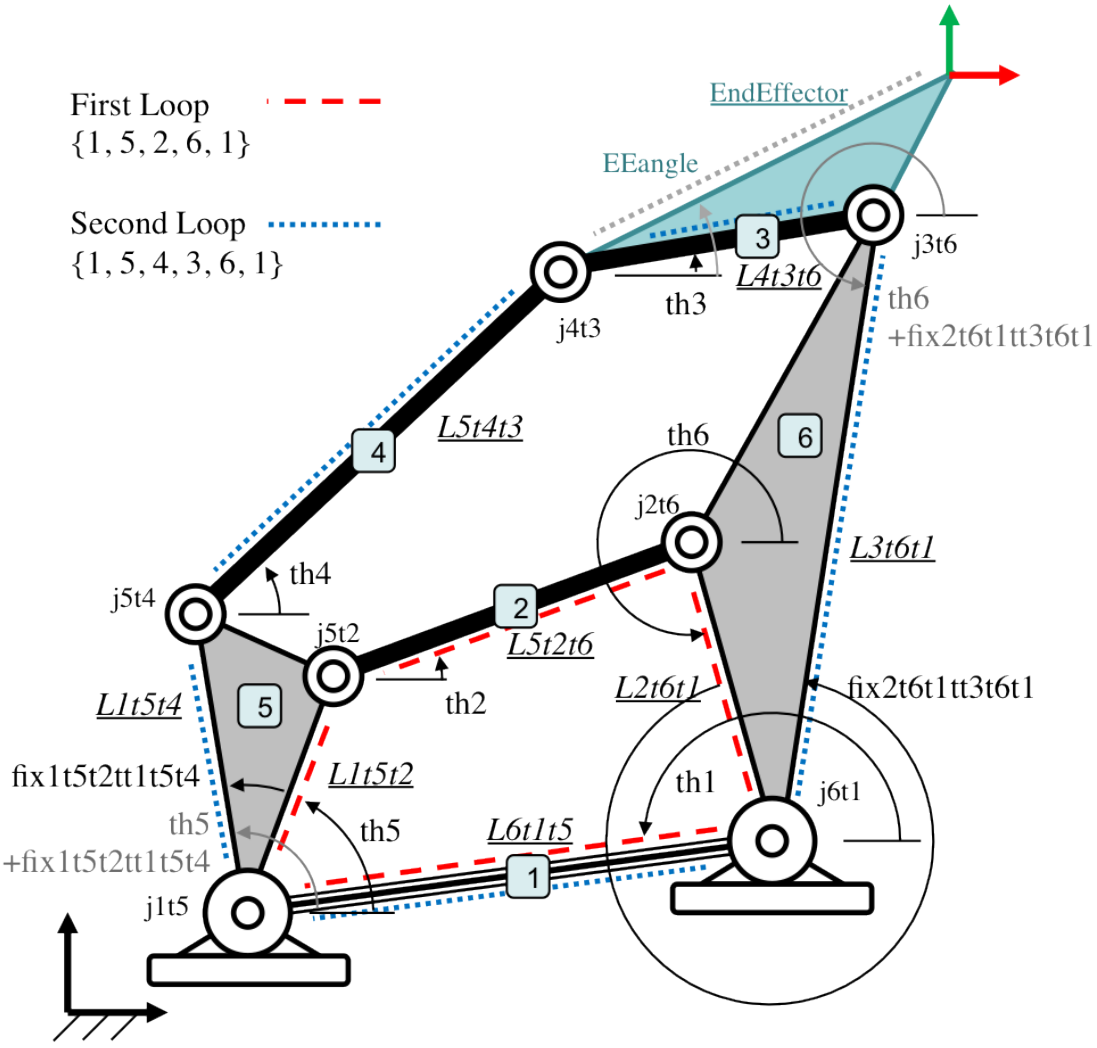


Figure 3.2: Automated naming convention applied to the Stephenson six-bar.

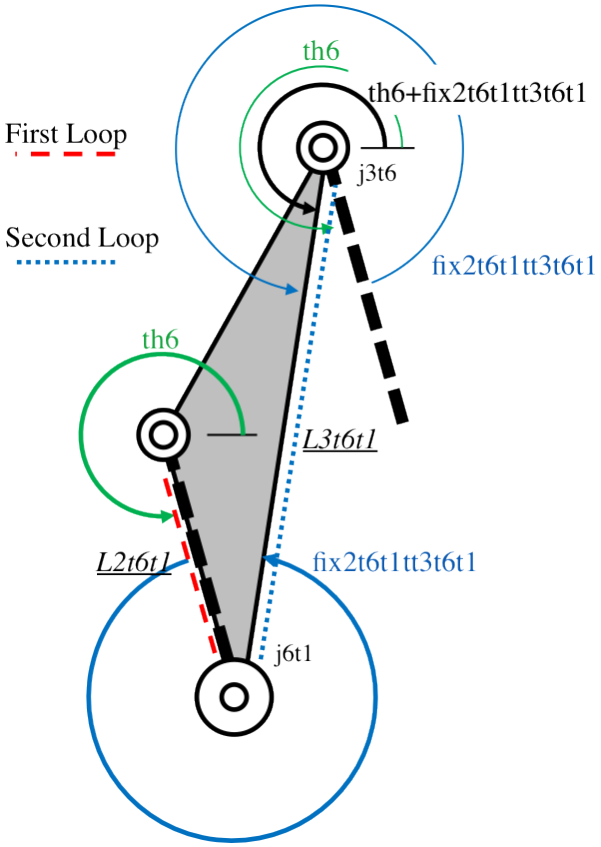


Figure 3.3: The summation of angles along convergent loops defines the angle of a link feature.

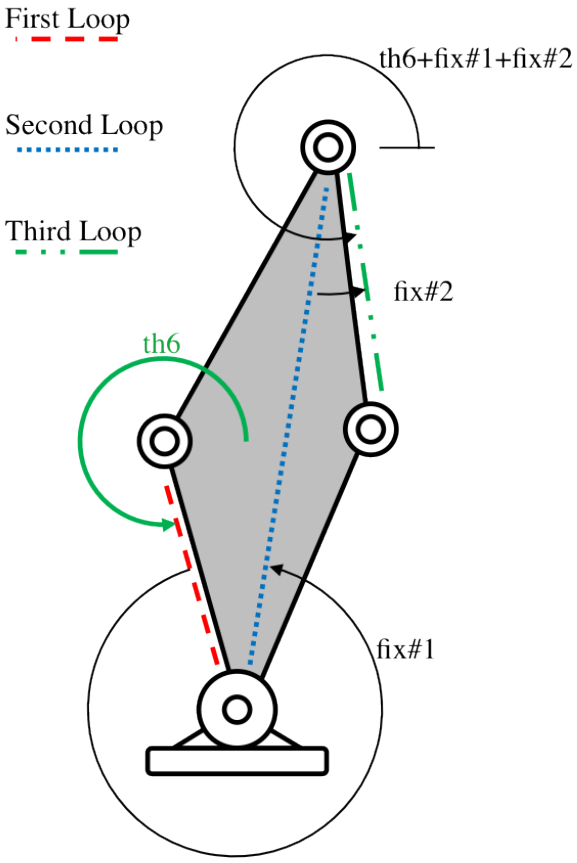


Figure 3.4: Quaternary links have two fixed angles that sum to define the true angle of a link feature.

necessitates a case-by-case linkage analysis approach. The non-standard naming requires a sketch of the linkage so that the analysis can accurately represent the meaning of each angle, represent the names of the link features between the joints, and assign the proper sign for the components.

3.2.2 FTLA Convention

To automate the process of developing the loop equations we apply the naming convention by automating the construction of a convention called FTLA. The FTLA convention describes the line on each link along each loop, e.g. $L1t5t2$, with four terms {From Joint, To Joint, Link Dimension, Angle}. Each loop is represented by a series of these four-term sets and the first set for each loop represents a 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 convenience. The only input needed to develop the FTLA is the cycle basis.

To create the FTLA for a linkage, first the cycle basis is converted to the series of joints it represents. The joints are named based on the links the joint connects. The joints are then paired in a {From Joint, To Joint} set to represent the end points of the lines on the links 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. The line name is appended to each of the joint pairs to form a trio representing the line and it's endpoints as {From Joint, To Joint, Link Dimension}.

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 link angle to properly define the line. All angles are positive counter clockwise.

The link angle origin is the first joint of the link encountered along the loop. The name assigned to this angle is "th" followed by the link number. Each of the {From Joint, To Joint, Link Dimension} trios represents a line along a particular link. That link number is the middle number of the name given to the line. This name is appended to each of the {From Joint, To Joint, Link Dimension} trios to form the initial form {From Joint, To Joint, Link Dimension, Link Angle}.

The second step generates the true angle for each link dimension by summing the fixed angle, e.g. $fix1t5t2tt1t5t4$, between the divergent or convergent loops with the angle of the reference 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 or diverges from a previous loop. These fit the form of Eqn. 3.1.

$$\begin{aligned} \text{Convergent Loop Form} &: \{Joint_2 - Joint_1, 0\}, \\ \text{Divergent Loop Form} &: \{0, Joint_2 - Joint_1\}. \end{aligned} \tag{3.1}$$

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.

When two or more loops use the same line along a link, the first loop to use the line establishes the appropriate fixed angle. This becomes important when determining the appropriate fixed angle for the third and higher loop. For example, if the second loop diverges from the first

loop but the third loop follows the same line as the first loop, the proper angle for the third loop is the same angle as defined by the first loop.

The FTLA convention provides a convenient interface with a synthesis algorithm because it provides all of the interconnect information necessary to draw the linkage in an easily decodable form. Once the physical dimensions are determined and the automated names are mapped to the names assigned in the synthesis algorithm, the synthesis algorithm can draw the physical linkage connections by drawing each line represented in the FTLA terms.

3.2.3 Derivation of the Loop Equations

The FTLA convention also provides a very convenient means of quickly writing the linkage loop equations. Using only the last two terms of each entry in the FTLA, we convert from the FTLA convention {From Joint, To Joint, Link Dimension, Angle} to the final loop equations by taking the summation along each loop of the product in Eqn. 3.2.

$$\begin{aligned} x &: (\text{Link Dimension}) * \cos(\text{Angle}) \\ y &: (\text{Link Dimension}) * \sin(\text{Angle}) \end{aligned} \tag{3.2}$$

Setting each of these summations equal to zero, the automated loop equations for the Stephenson six-bar linkage, Fig. 3.2, are shown in Eqn. 3.3.

Loop 1

$$x : L6t1t5\cos(th1) + L5t2t6\cos(th2) + L1t5t2\cos(th5) + L2t6t1\cos(th6) = 0$$

$$y : L6t1t5\sin(th1) + L5t2t6\sin(th2) + L1t5t2\sin(th5) + L2t6t1\sin(th6) = 0$$

Loop 2

$$x : L6t1t5\cos(th1) + L4t3t6\cos(th3) + L5t4t3\cos(th4)$$
$$+ L1t5t4\cos(fix1t5t2tt1t5t4 + th5) + L3t6t1\cos(fix2t6t1tt3t6t1 + th6) = 0$$

$$y : L6t1t5\sin(th1) + L4t3t6\sin(th3) + L5t4t3\sin(th4)$$
$$+ L1t5t4\sin(fix1t5t2tt1t5t4 + th5) + L3t6t1\sin(fix2t6t1tt3t6t1 + th6) = 0 \tag{3.3}$$

Dimensioned as shown in Fig. 2.13 the non-automated loop equations for the same Stephenson six-bar are repeated here in Eqn. 3.4. The non-automated loop equations have variation in the placement of angles that produces several instances of negative signs. The sense of exactly which element in the loop ties to which other element in the loop is also not apparent.

Loop 1

$$x : b_1\cos(\theta_1 - \gamma) + l_5\cos\theta_5 - b_2\cos(\theta_4 + \eta) - l_0\cos\theta_0 = 0$$

$$y : b_1\sin(\theta_1 - \gamma) + l_5\sin\theta_5 - b_2\sin(\theta_4 + \eta) - l_0\sin\theta_0 = 0$$

Loop 2

$$x : l_1\cos\theta_1 + l_2\cos\theta_2 + l_3\cos\theta_3 - l_4\cos\theta_4 - l_0\cos\theta_0 = 0$$

$$y : l_1\sin\theta_1 + l_2\sin\theta_2 + l_3\sin\theta_3 - l_4\sin\theta_4 - l_0\sin\theta_0 = 0. \tag{3.4}$$

3.3 Loop Equation Conclusion

This chapter showed the procedure to automate the linkage loop equations for any topology of eight-bar kinematic chain. The procedure used a shortest path algorithm in combination with a successive elimination of known short paths to ensure all paths are identified. The shortest paths are converted to cycles, also called loops, and the smallest set that forms a cycle basis is selected to represent the linkage.

To automate the derivation of the linkage loop equations a naming convention is applied algorithmically to the cycle basis. The naming convention uniquely defines the name of the geometric linkage features. The named features are the joints, the length of the lines between joints, the global angles, and the fixed angles.

A convenient convention called FTLA is defined that describes the link connections along each loop and creates a structured format for automating the linkage loop equations. The FTLA convention also conveniently organizes the link interconnection information so that a synthesis algorithm can draw the linkage.

Chapter 4

Automated Linkage Analysis

The linkage loop equations must be solved for a specific geometry and a specific selection of the ground and input links. This section describes the selected approach to solve for the angles of every link in the linkage through the Dixon determinant. The Jacobian is used to identify singular configurations.

4.1 Specific Linkage Dimensions

To analyze a specific linkage, the algorithm must evaluate a specific topology and a specific selection for the ground link and the ground connected input link. This information is provided as an input to this analysis.

To provide a numerical output, the linkage dimensions are also required. Rather than coordinating a feature by feature accounting of the linkage dimensions from a synthesis routine, the method used in this research to specify the physical geometry of the linkage is through an enhanced adjacency matrix. In a normal adjacency matrix each "1" represents a joint between two links. In the enhanced adjacency matrix each "1" is replaced with the {x,

y} coordinates of the joint when the linkage is in a single desired assembly configuration. When combined with the automatically derived linkage loops and the naming convention, this physical joint location information enables the calculation of all of the linkage geometric features necessary to perform the automated analysis.

4.2 Automated Dixon Determinant

To solve for the angles of all of the links we solve the Dixon determinant using the complex plane formulation as discussed in section 2.15. The automated equations using the naming convention do not use subscripts but the notation in this section retains the subscripts for clarity. The formulation uses the complex equation form of the linkage loop equations and provides all of the assembly configurations possible.

4.2.1 Automated Loop Equations in Complex Form

To create the loop equations in complex form, the list of complex variable names must be determined. The FTLA contains all of the variable names in rational form. The last term from each FTLA entry is the angle for each line along the loops of the linkage. These are collected and formed into a list of angles. The angle list is converted to the equivalent list of complex variable names by text manipulation replacing the "th" with "Θ". The conjugate variables are appended with the letter "c". The temporary variable list α is also produced using text manipulation.

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 transform the loop equations into complex form by applying trigonometric identities,

Eqn. 2.24, and exponential identities, Eqn. 2.25. The complex conjugates of the equations are required so that a sufficient quantity of independent equations is obtained.

Mathematica contains suitable functions to convert trigonometric expressions to exponential form but leaves the angle names in the form of $e^{\pm th_j}$. These exponential variable names are replaced with the complex variable names Θ_j and $\Theta_j c$. The fixed angles remain in exponential form. This forms the final vector loop equations.

Upon selection of an output angle for the eigenvalue Θ_n the vector loop equations are formed into the first row of the Dixon determinant. The second and subsequent rows are constructed by performing the substitutions of "Θ" to "α". Next the row to row subtractions are performed.

To factor out the extraneous roots where $\Theta_j = \alpha_j$, first the algorithm creates a substitution list using text manipulation. The substitution list is applied to convert all the terms from the form $\Theta_j - \alpha_j$ to the conjugate form $-\Theta_j\alpha_j(\Theta_j c - \alpha_j c)$ so that the conjugate terms can all be factored out to leave only monomials of Θ and α.

A few pre-existing functions in Mathematica are applied to automate the conversion to the generalized eigenvalue form. The determinant is taken using an existing function to produce the polynomial δ. The monomials and coefficients of that polynomial are extracted using existing functions to produce the vectors $\mathbf{a}$ and $\mathbf{t}$ and the matrix W that fit the form of Eqn. 2.27.

The matrix W is separated into two matrices M and N by collecting the terms containing the selected eigenvalue Θ_n. The terms that contain the eigenvalue conjugate $\Theta_n c$ can be eliminated by multiplying those rows by Θ_n. Since the quadrant that contains the conjugate terms is known in advance, the structure of W enables determination of the matrices M and N by quadrants.

4.2.2 Check for Proper Selection of the Eliminant

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. 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 the selection of the eigenvalue variable we simply test each output 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 this criterion is selected as the eigenvalue.

4.2.3 Flagging Linkages That Partition

The automation identifies linkages that partition into simpler linkages because these are the linkages where there is no eliminant, eigenvalue Θ_n, that produces a resulting eigenvector $\mathbf{t}$ such that there exists a monomial ratio that will produce every unknown angle. If every output angle is tested and none of them are a valid eigenvalue, the linkage is flagged as a partitioning linkage.

4.2.4 Solving the Loop Equations

Once a valid eigenvalue variable has been selected, the numerical values for the link dimensions and the angle of the ground link are substituted into the matrices M and N of Eqn. 2.30

to form a numerical generalized eigenvalue problem in one variable, the input angle and its conjugate. This equation is numerically solved for a given input angle. The complex valued solutions are converted back to real values. Some solutions will remain imaginary after applying the conversion back to real values. These do not represent real assembly configurations. The real valued solutions represent all of the possible linkage assembly configurations for that input angle.

4.3 Tracking Solutions

The Dixon determinant can be solved for several input angles to determine the output angles over a range of input angles. Because the Dixon determinant provides all of the possible real assembly configurations a method to identify a particular assembly configuration within the solutions is needed. McCarthy and Soh [23] show a numerical method using Newton's method to track a particular solution through the range of input angles. Instead of using that method, this research pursued an alternate technique using the determinant of the Jacobian.

4.3.1 Jacobian Automation

The automation of the Jacobian follows the development shown in section 2.14. We convert the Jacobian to block upper triangular form through a determinant preserving transform as shown by Silvester [36], Eqn. 2.17. This factors the determinant of the Jacobian into the determinant of the individual 2x2 block matrices along the diagonal.

Since the transform involves an inverse of sub-matrix 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. Mathematica has a pre-existing function that is used to find all of ways of rearranging the columns so that the diagonal elements are not zero.

The first step to order the columns is to produce a list of all the elements in each row that are non-zero. Then an ordering list is produced that shows all of the ways of picking non-zero elements for each column. For example, if the first row of the Jacobian has a non-zero term in the fourth and fifth columns, the columns could be rearranged so that either the fourth or the fifth column is moved to the first column. Because every two rows of the Jacobian have the same zero elements, either of the same two columns could also be moved to the second column. Moving these two columns to the first and second column locations makes the upper left 2x2 corner of the Jacobian full-rank. The ordering list is filtered to show only those orderings that do not duplicate any columns. Any of these orderings will put non-zero elements on the diagonal, so the first one is selected and converted to a permutation matrix which is applied to the Jacobian. Because every two rows of the Jacobian have zeros in the same columns, this reordering also produces a matrix where each of the 2x2 blocks on the diagonal are full rank.

For the eight-bar family the Jacobian is a 6x6 matrix and we apply the determinant preserving transform twice, the first transform treats the lower right 2x2 along the diagonal as D, the second transform treats the lower right 4x4 as D.

4.3.2 Jacobian Sign List

For many linkages the sign set of the factored Jacobian determinant uniquely identifies the configuration of interest among the solutions of the Dixon determinant. For linkages that contain a link that rotates more than 360 degrees before encountering a singularity, the factored Jacobian alone may not be sufficient to track the solutions uniquely because within certain ranges of input there will exist two assembly configurations with the same sign list. However, these cases may actually be adequately represented by the Jacobian determinant sign list since the two assembly configurations may be connected. If the output angle can

be tracked continuously over an input angle range that is greater than 360 degrees then the solutions should be able to be consistently tracked and distinguished.

As an output of the automation, we produce the Jacobian sign list symbolically and numerically. The numerical Jacobian sign list represents the assembly configuration specified by the enhanced adjacency matrix.

Future research is needed to determine if the sign list of the factored Jacobian can always identify the configuration of interest even when the linkage contains a link that rotates more than 360 degrees before reaching a singularity.

4.4 Analysis Results Outputs

There are several outputs that can be produced by the automation. One output provides a named adjacency matrix such that each "1" is replaced with the automated joint name. Another output is the FTLA for the linkage. The FTLA describes the connections in an easily decodable form. The Dixon determinant matrices M and N are produced in numerical form. The associated vector $\mathbf{t}$ is produced in symbolic form. The Jacobian sign list is produced symbolically and numerically. A substitution list relating the value of the geometric features to the feature names is also produced.

The automation is also capable of supplying the loop equations in rational and complex form.

4.5 Automated Linkage Analysis Conclusion

This chapter described how the linkage configuration analysis is automated. The chapter discussed the inputs required. It also showed how the Dixon determinant process is automated to solve the linkage loop equations.

The automation that identifies the valid eliminant for the Dixon determinant was shown along with the automation that identifies a linkage that partitions.

To distinguish each assembly configuration the automated derivation of the Jacobian was shown and the numerical output of the factored Jacobian determinant was discussed.

A list of the available algorithm outputs was provided.

Chapter 5

Examples

This chapter applies the automation to various examples.

Two examples in this chapter are from Parrish and McCarthy [31], used with permission.

5.1 Example: Automation of the Loop Equations

We apply the process to find the loop equations for the example eight-bar linkage shown in Fig. 5.1.

5.1.1 Example: Automation Input

The adjacency matrix, Eqn. 5.1, for the example linkage shows the link to link connections. The enhanced adjacency matrix, Eqn. 5.2, replaces the "1" with the physical location of the

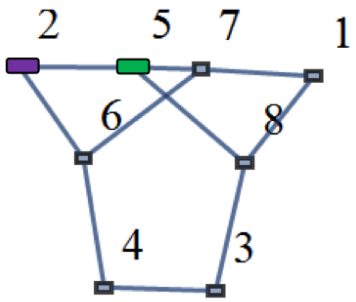
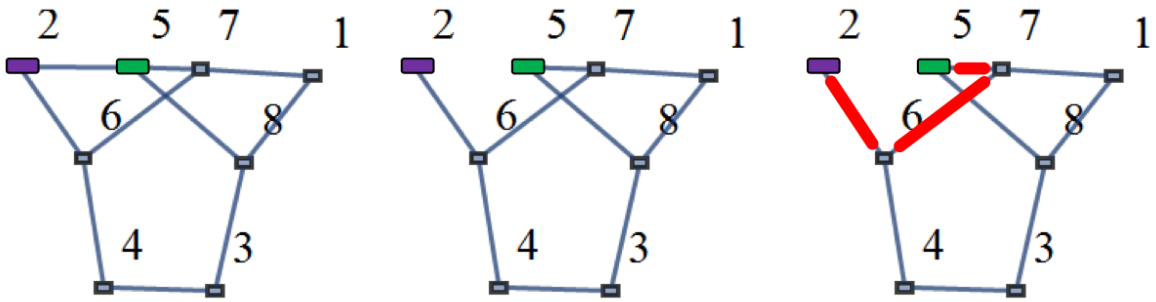


Figure 5.1: Example eight-bar linkage used to demonstrate the automation of the loop equations.

joints when the linkage is in one of the synthesized assembly configurations. The example is displayed here with only two significant figures.

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{5.1}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & \{0.1,\ 0.1\} & \{10.,\ 0.47\} \\ 0 & 0 & 0 & 0 & \{17.,\ 0.076\} & \{12.,\ 0\} & 0 & 0 \\ 0 & 0 & 0 & \{9.4,\ 10.\} & 0 & 0 & 0 & \{15.,\ -3.9\} \\ 0 & 0 & \{9.4,\ 10.\} & 0 & 0 & \{15.,\ -5.\} & 0 & 0 \\ 0 & \{17.,\ 0.076\} & 0 & 0 & 0 & 0 & \{12.,\ 6.4\} & \{13.,\ 5.4\} \\ 0 & \{12.,\ 0\} & 0 & \{15.,\ -5.\} & 0 & 0 & \{11.,\ 3.4\} & 0 \\ \{0.1,\ 0.1\} & 0 & 0 & 0 & \{12.,\ 6.4\} & \{11.,\ 3.4\} & 0 & 0 \\ \{10.,\ 0.47\} & 0 & \{15.,\ -3.9\} & 0 & \{13.,\ 5.4\} & 0 & 0 & 0 \end{bmatrix} \tag{5.2}$$



(a) Initial Graph (b) First Elimination (c) First Shortest Path

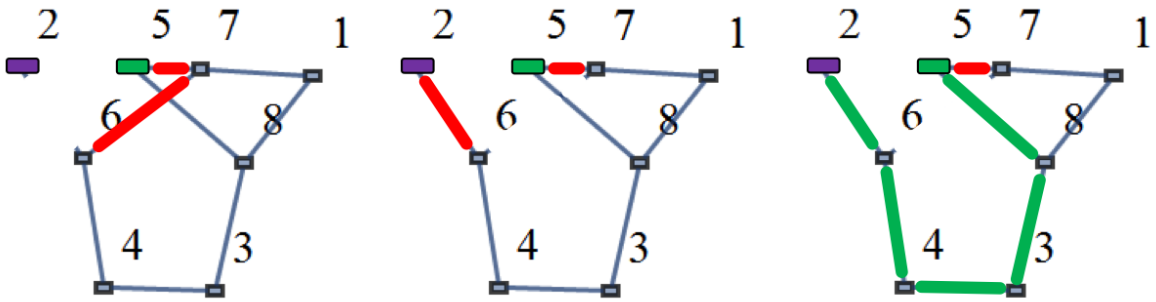
Figure 5.2: The first level of search provides the first shortest path.

5.1.2 Example: Finding Unique Loops

The process to automatically construct the linkage loops for a particular linkage begins with the linkage graph and the user defined selection for the ground link and input link. Applied to the example eight-bar Fig. 5.2 shows the original graph, the elimination of the edge between the ground and input links, and the identification of the first shortest path. This first shortest path, Fig. 5.2c, is the path at the first level of depth.

To find the second level of shortest paths that are independent of the currently known shortest path the algorithm must be forced to avoid the currently known shortest path. This is accomplished by eliminating an edge in the current path as well as eliminating the edge that produced the current path. The set of edges that perform this function are derived from the current path. Each edge of the current path must be attempted, therefore the list of eliminations that will be used to find loops at the second depth level are shown in Eqn. 5.3.

$$\begin{aligned} 1 &: \{\{5 \leftrightarrow 2\}, \{2 \leftrightarrow 6\}\} \\ 2 &: \{\{5 \leftrightarrow 2\}, \{6 \leftrightarrow 7\}\} \\ 3 &: \{\{5 \leftrightarrow 2\}, \{7 \leftrightarrow 5\}\} \end{aligned} \tag{5.3}$$



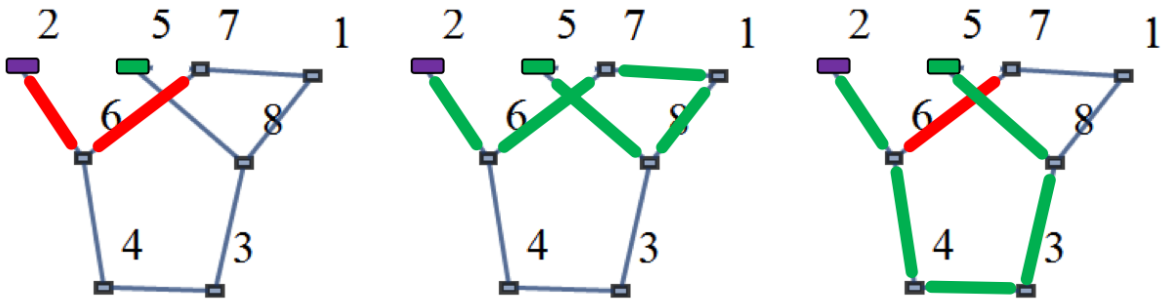
(a) First Failure Case (b) Valid Elimination (c) Second Shortest Path

Figure 5.3: The second level of search provides the second shortest path.

Removing an elimination set containing two or more edges could separate the graph into two pieces resulting in no paths from the input vertex back to ground. When this occurs the elimination set is not valid and the algorithm tests the next elimination set in the elimination list.

In Fig. 5.3a elimination of the first set, $\{\{5 \leftrightarrow 2\}, \{2 \leftrightarrow 6\}\}$, eliminates all connections to the input vertex. The first valid elimination set is $\{\{5 \leftrightarrow 2\}, \{6 \leftrightarrow 7\}\}$ and is shown in Fig. 5.3b. Finding the next shortest path on this reduced graph produces the second shortest path shown in Fig. 5.3c.

Next an elimination list is derived from the second shortest path that will eliminate one edge in this path and both edges that formed this current path. This elimination list, shown in Eqn. 5.4, will be used to find a path at the third level of depth.



(a) Valid Elimination (b) Alternate Path (c) Selected Third Path

Figure 5.4: The second level of search provides the third shortest path.

$$
\begin{aligned}
1 \;&:\; \{\{5 \leftrightarrow 2\}, \{6 \leftrightarrow 7\}, \{2 \leftrightarrow 6\}\},\\
2 \;&:\; \{\{5 \leftrightarrow 2\}, \{6 \leftrightarrow 7\}, \{6 \leftrightarrow 4\}\},\\
3 \;&:\; \{\{5 \leftrightarrow 2\}, \{6 \leftrightarrow 7\}, \{4 \leftrightarrow 3\}\},\\
4 \;&:\; \{\{5 \leftrightarrow 2\}, \{6 \leftrightarrow 7\}, \{3 \leftrightarrow 8\}\},\\
5 \;&:\; \{\{5 \leftrightarrow 2\}, \{6 \leftrightarrow 7\}, \{8 \leftrightarrow 5\}\}.
\end{aligned} \tag{5.4}
$$

Although we found a path at the second level, we are not yet done with the elimination sets derived from the first level. In Fig. 5.4a the final elimination set from the first shortest path is applied, $\{\{5 \leftrightarrow 2\}, \{7 \leftrightarrow 5\}\}$. This elimination produces two shortest paths of equal length, Fig. 5.4b and Fig. 5.4c. Either path can be selected for the next shortest path. In this example we select the last one, Fig. 5.4c, as the third shortest path. This is the same path as found previously. Although the path shown Fig. 5.4b was not selected, it will still be one of the shortest paths at the next level and will not get overlooked.

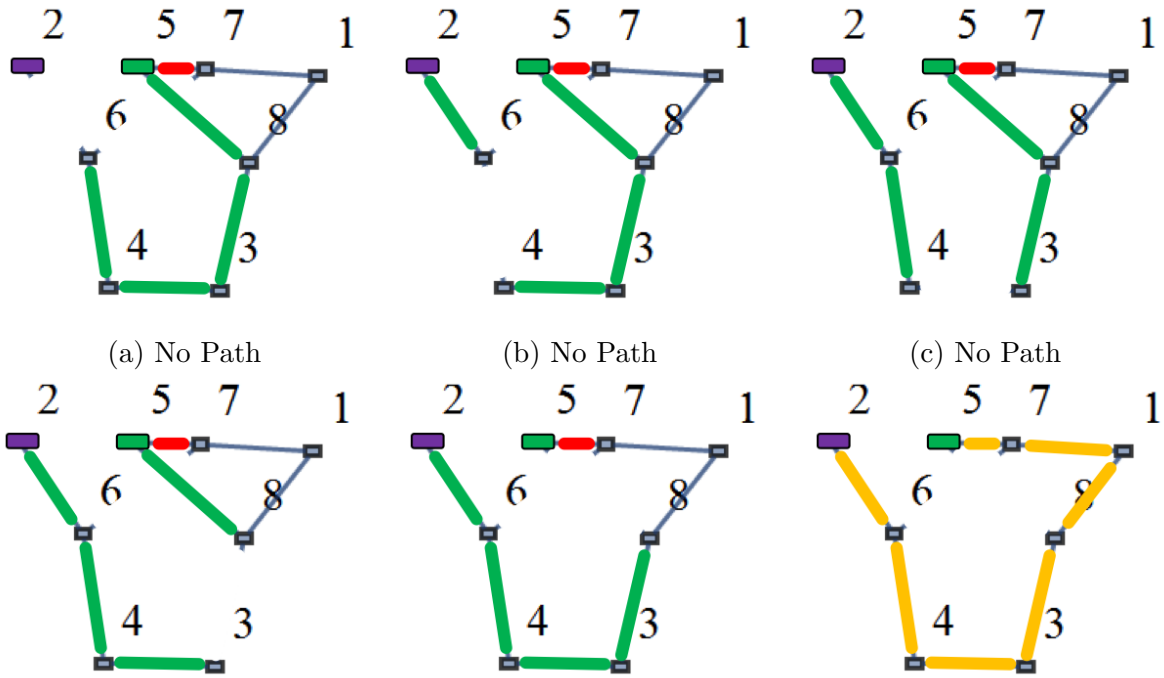
The elimination list derived from the third shortest path is shown in Eqn. 5.5.

$$
\begin{aligned}
1 \;&:\; \{\{5 \leftrightarrow 2\}, \{7 \leftrightarrow 5\}, \{2 \leftrightarrow 6\}\}, \\
2 \;&:\; \{\{5 \leftrightarrow 2\}, \{7 \leftrightarrow 5\}, \{6 \leftrightarrow 4\}\}, \\
3 \;&:\; \{\{5 \leftrightarrow 2\}, \{7 \leftrightarrow 5\}, \{4 \leftrightarrow 3\}\}, \\
4 \;&:\; \{\{5 \leftrightarrow 2\}, \{7 \leftrightarrow 5\}, \{3 \leftrightarrow 8\}\}, \\
5 \;&:\; \{\{5 \leftrightarrow 2\}, \{7 \leftrightarrow 5\}, \{8 \leftrightarrow 5\}\}.
\end{aligned} \tag{5.5}
$$

The last elimination set derived from the first level shortest path has now been evaluated and the second level of shortest paths have been found. From these second level paths two elimination lists have been formed, Eqn. 5.4 and Eqn. 5.5. These two elimination lists are now evaluated one element at a time to find the third level of paths.

Shown in Fig. 5.5 we eliminate each elimination set shown in Eqn. 5.4 and find that no valid paths are found until the last elimination set. Elimination set $\{5 \leftrightarrow 2\}, \{6 \leftrightarrow 7\}, \{8 \leftrightarrow 5\}$ produces the fourth shortest path shown Fig. 5.5f.

A fourth level elimination list is determined from this shortest path and is shown in Eqn. 5.6. We know that an eight-bar linkage will only have three independent loops therefore none of these eliminations will produce another path from input to ground. By inspection it is apparent that every one of these elimination sets will separate the linkage into two parts as expected. However, the automation will check each of these before stopping.

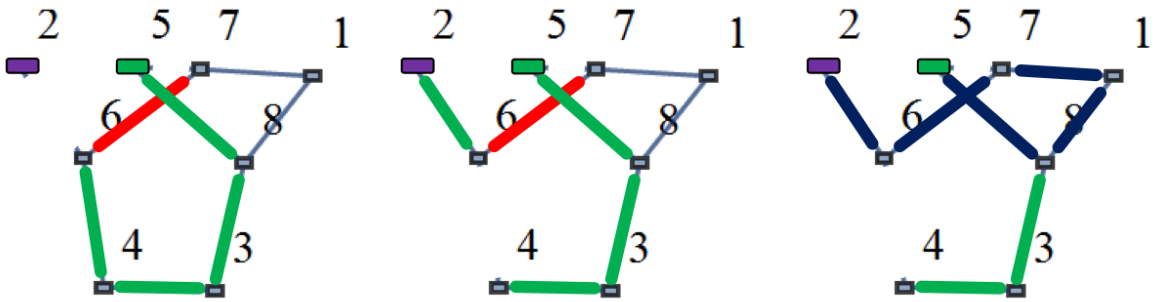


(a) No Path (b) No Path (c) No Path

(d) No Path (e) Valid Elimination (f) Fourth Shortest Path

Figure 5.5: The third level of search provides the fourth shortest path.

$$
\begin{aligned}
1 &: \{\{5 \leftrightarrow 2\}, \{6 \leftrightarrow 7\}, \{8 \leftrightarrow 5\}, \{2 \leftrightarrow 6\}\}, \\
2 &: \{\{5 \leftrightarrow 2\}, \{6 \leftrightarrow 7\}, \{8 \leftrightarrow 5\}, \{6 \leftrightarrow 4\}\}, \\
3 &: \{\{5 \leftrightarrow 2\}, \{6 \leftrightarrow 7\}, \{8 \leftrightarrow 5\}, \{4 \leftrightarrow 3\}\}, \\
4 &: \{\{5 \leftrightarrow 2\}, \{6 \leftrightarrow 7\}, \{8 \leftrightarrow 5\}, \{3 \leftrightarrow 8\}\}, \\
5 &: \{\{5 \leftrightarrow 2\}, \{6 \leftrightarrow 7\}, \{8 \leftrightarrow 5\}, \{8 \leftrightarrow 1\}\}, \\
6 &: \{\{5 \leftrightarrow 2\}, \{6 \leftrightarrow 7\}, \{8 \leftrightarrow 5\}, \{1 \leftrightarrow 7\}\}, \\
7 &: \{\{5 \leftrightarrow 2\}, \{6 \leftrightarrow 7\}, \{8 \leftrightarrow 5\}, \{7 \leftrightarrow 5\}\}.
\end{aligned} \tag{5.6}
$$



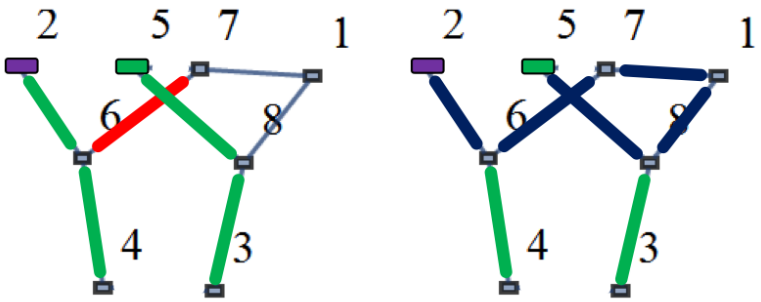
(a) No Path (b) Valid Elimination (c) Fifth Shortest Path

Figure 5.6: The third level of search provides the fifth shortest path.

Although we found the fourth shortest path we have not yet completed the evaluation of all paths found at the second level. Evaluating the elimination sets of Eqn. 5.5, the first valid elimination set is $\{\{5 \leftrightarrow 2\}, \{7 \leftrightarrow 5\}, \{6 \leftrightarrow 4\}\}$. This first valid elimination produces the fifth shortest path shown Fig. 5.6c.

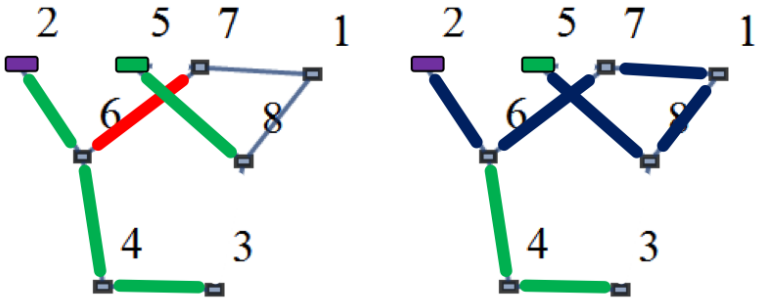
The elimination list determined from the fifth shortest path of the linkage shown in Fig. 5.6c is shown in Eqn. 5.7. By observation it is apparent that every one of these elimination sets will separate the linkage into two parts. However, the automation will check each of these before stopping.

$$
\begin{aligned}
1 &: \{\{5 \leftrightarrow 2\}, \{7 \leftrightarrow 5\}, \{6 \leftrightarrow 4\}, \{2 \leftrightarrow 6\}\}, \\
2 &: \{\{5 \leftrightarrow 2\}, \{7 \leftrightarrow 5\}, \{6 \leftrightarrow 4\}, \{6 \leftrightarrow 7\}\}, \\
3 &: \{\{5 \leftrightarrow 2\}, \{7 \leftrightarrow 5\}, \{6 \leftrightarrow 4\}, \{7 \leftrightarrow 1\}\}, \\
4 &: \{\{5 \leftrightarrow 2\}, \{7 \leftrightarrow 5\}, \{6 \leftrightarrow 4\}, \{1 \leftrightarrow 8\}\}, \\
5 &: \{\{5 \leftrightarrow 2\}, \{7 \leftrightarrow 5\}, \{6 \leftrightarrow 4\}, \{8 \leftrightarrow 5\}\}.
\end{aligned} \tag{5.7}
$$



(a) Valid Elimination (b) Sixth Shortest Path

Figure 5.7: The third level of search provides the sixth shortest path.



(a) Valid Elimination (b) Seventh Shortest Path

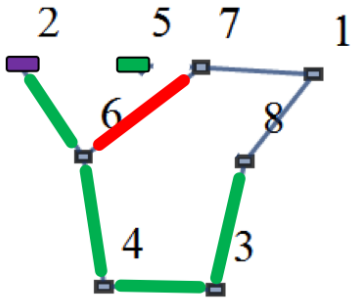
Figure 5.8: The third level of search provides the seventh shortest path.

Although we found the fifth shortest path we have not yet completed the evaluation of all paths found at the second level. Continuing through the elimination sets of Eqn. 5.5, the next valid elimination sets produce the same shortest path two more times. For completeness these are shown in Fig. 5.7 and Fig. 5.8. Their associated elimination lists are not shown.

The last elimination set in Eqn. 5.5 yields no paths, Fig. 5.9.

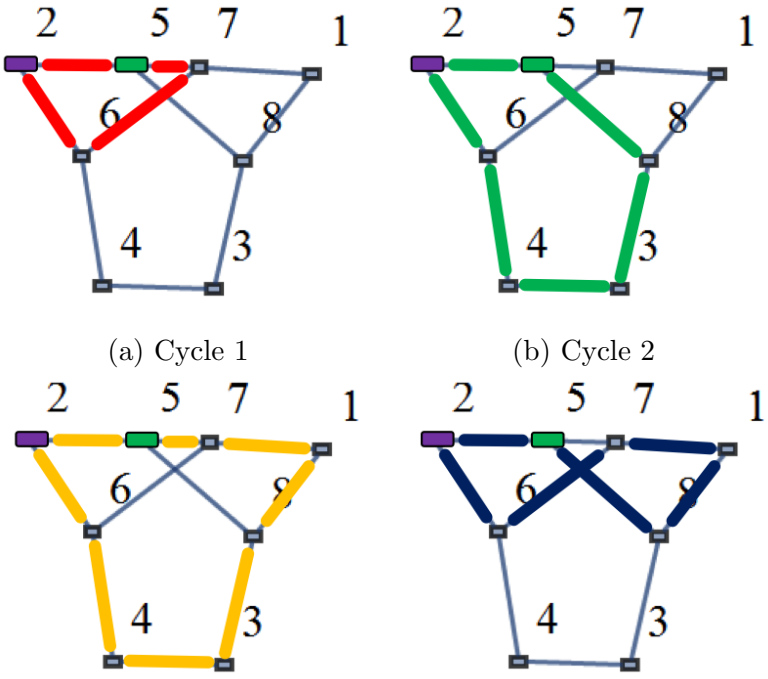
The third level of paths has now been determined. At this point the algorithm checks the elimination sets derived from each of these paths and finds that no new paths exist at the fourth depth level. The algorithm stops.

Among the seven shortest paths there are only four unique shortest paths. Each of these four paths is made into a cycle, or loop, by appending the edge from the ground vertex to the input vertex. The four unique cycles are shown in Fig. 5.10



(a) No Path

Figure 5.9: The final elimination at the third level of search produces no paths.



(a) Cycle 1

(b) Cycle 2

(c) Cycle 3

(d) Cycle 4

Figure 5.10: Four unique but not independent cycles are identified.

The cycles begin with the ground vertex. The second vertex is the input vertex. The rest of the vertices are as encountered in order along the loop, ending with ground. The cycles are shown in Eqn. 5.8

$$\begin{aligned}
&\text{Cycle 1} \;:\; \{5,2,6,7,5\},\\
&\text{Cycle 2} \;:\; \{5,2,6,4,3,8,5\},\\
&\text{Cycle 3} \;:\; \{5,2,6,4,3,8,1,7,5\},\\
&\text{Cycle 4} \;:\; \{5,2,6,7,1,8,5\}.
\end{aligned} \tag{5.8}$$

5.1.3 Example: Finding the Cycle Basis

To minimize the equations that represent a linkage the algorithm selects the smallest $n/2-1$ independent cycles to be the cycle basis. The cycles are sorted in the following order.

1. Loop Length
2. Vertex Number
3. Vertex Degree

Applying the sorting to the example linkage, the loops are sorted first by loop length in Eqn. 5.9. The order of the loops is $\{1,2,4,3\}$

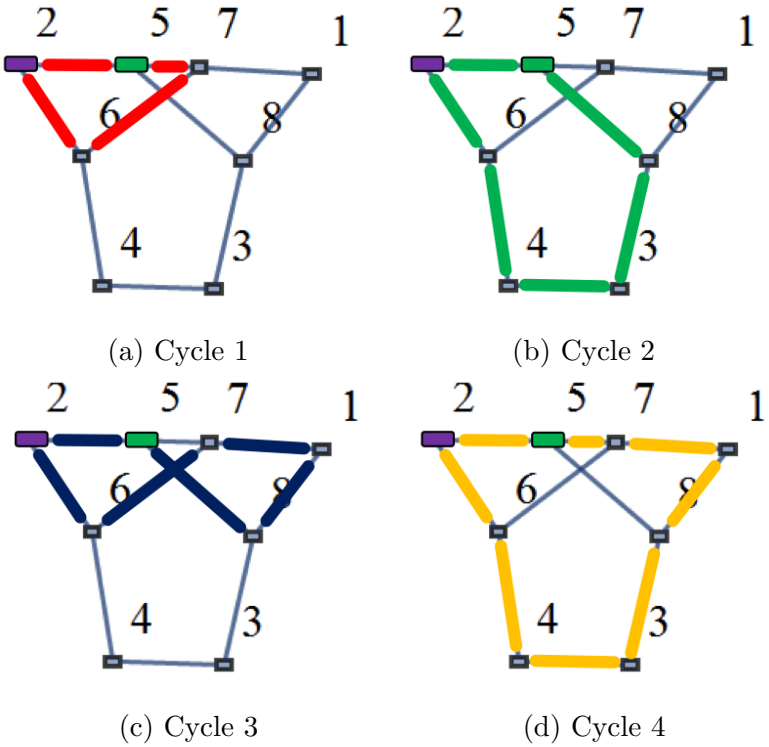
$$\begin{aligned}
\text{Cycle } 1 &: \{5,2,6,7,5\},\\
\text{Cycle } 2 &: \{5,2,6,4,3,8,5\},\\
\text{Cycle } 4 &: \{5,2,6,7,1,8,5\},\\
\text{Cycle } 3 &: \{5,2,6,4,3,8,1,7,5\}.
\end{aligned} \tag{5.9}$$

The loops are sorted next by loop vertex number. Among the loops of the same length, the loops $\{5,2,6,4,3,8,5\}$ and $\{5,2,6,7,1,8,5\}$ are already properly sorted therefore the order of the loops is unchanged, $\{1,2,4,3\}$.

The vertex degree list for each loop is determined, Eqn. 5.10.

$$\begin{aligned}
\text{Cycle } 1 &: \{3,2,3,3,3\},\\
\text{Cycle } 2 &: \{3,2,3,2,2,3,3\},\\
\text{Cycle } 4 &: \{3,2,3,3,2,3,3\},\\
\text{Cycle } 3 &: \{3,2,3,2,2,3,2,3,3\}.
\end{aligned} \tag{5.10}$$

The vertex degree lists are padded with zeros on the left, Eqn. 5.11, to equalize the length.



(a) Cycle 1 (b) Cycle 2

(c) Cycle 3 (d) Cycle 4

Figure 5.11: The four unique cycles sorted by size and vertex degree.

$$
\begin{aligned}
&\text{Cycle 1} \;:\; \{0,0,0,0,3,2,3,3,3\},\\
&\text{Cycle 2} \;:\; \{0,0,3,2,3,2,2,3,3\},\\
&\text{Cycle 4} \;:\; \{0,0,3,2,3,3,2,3,3\},\\
&\text{Cycle 3} \;:\; \{3,2,3,2,2,3,2,3,3\}.
\end{aligned}
\tag{5.11}
$$

The padded vertex degree lists are sorted in ascending order by depth. In this example, the order of the loops is unchanged, $\{1,2,4,3\}$. The sorting order is applied to the loops and the loops are renumbered as shown in Fig. 5.11.

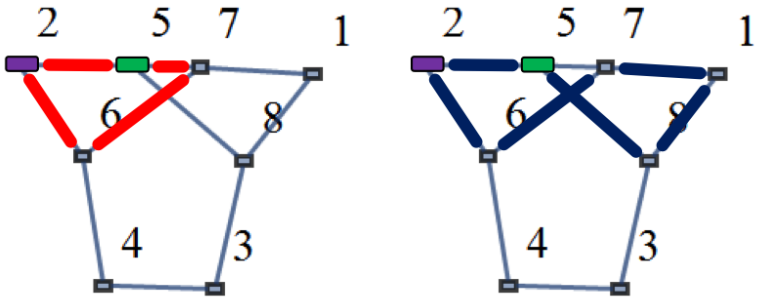
To select the smallest $n/2-1$ set of independent cycles for the cycle basis we start with the smallest cycle, Eqn. 5.12.

$$\text{Cycle 1 Contributes} \quad : \quad \{5 \leftrightarrow 2\}, \{2 \leftrightarrow 6\}, \{6 \leftrightarrow 7\}, \{7 \leftrightarrow 5\} \tag{5.12}$$

The next cycle that contains the fewest new edges is added to the cycle basis. Each cycle contributes the edges shown in Eqn. 5.13. In the example the third cycle contributes the fewest new edges and is added to the cycle basis, Fig. 5.12.

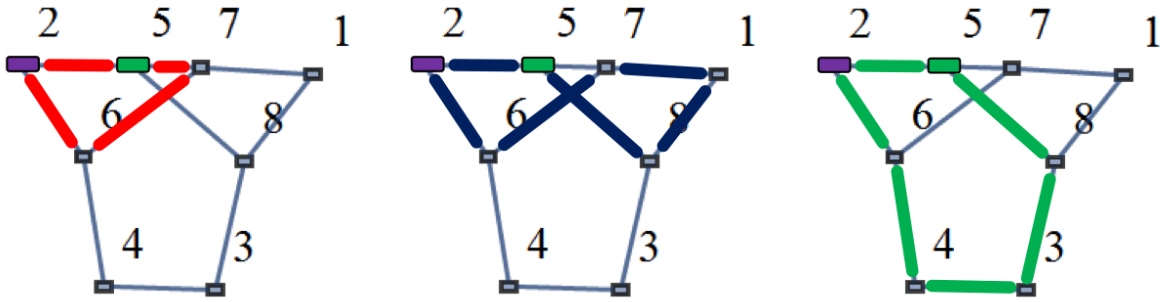
$$\begin{aligned}
&\text{Cycle 2 Contributes} \quad : \quad \{6 \leftrightarrow 4\}, \{4 \leftrightarrow 3\}, \{3 \leftrightarrow 8\}, \{8 \leftrightarrow 5\} \\
&\text{Cycle 3 Contributes} \quad : \quad \{7 \leftrightarrow 1\}, \{1 \leftrightarrow 8\}, \{8 \leftrightarrow 5\} \\
&\text{Cycle 4 Contributes} \quad : \quad \{6 \leftrightarrow 4\}, \{4 \leftrightarrow 3\}, \{3 \leftrightarrow 8\}\{8 \leftrightarrow 1\}, \{1 \leftrightarrow 7\}, \{7 \leftrightarrow 5\}.
\end{aligned} \tag{5.13}$$

At the next iteration, Cycle 2 and Cycle 4 both contribute three new edges, the same three new edges, shown in Eqn. 5.14. The first one, Cycle 2, is added to the cycle basis, Fig. 5.13.



(a) Cycle 1 (b) Cycle 3

Figure 5.12: The first two cycles in the cycle basis.

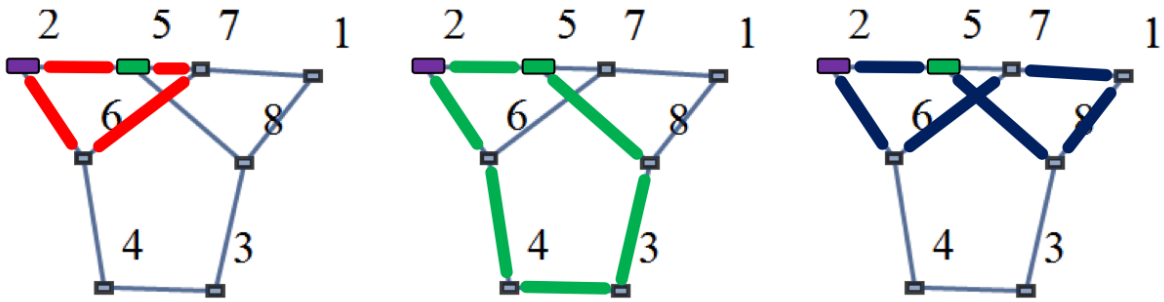


(a) Cycle 1 (b) Cycle 3 (c) Cycle 2

Figure 5.13: The three unsorted cycles in the cycle basis.

$$\begin{aligned} \text{Cycle 2 Contributes} &: \{6 \leftrightarrow 4\}, \{4 \leftrightarrow 3\}, \{3 \leftrightarrow 8\} \\ \text{Cycle 4 Contributes} &: \{6 \leftrightarrow 4\}, \{4 \leftrightarrow 3\}, \{3 \leftrightarrow 8\} \end{aligned} \tag{5.14}$$

The remaining cycle, Cycle 4, is dependent on the first three cycles as it can be formed by adding the previous cycles, modulo 2. Cycle 4 also does not contribute any new edges to the cycle basis, as expected.



(a) Cycle 1 (b) Cycle 2 (c) Cycle 3

Figure 5.14: Three smallest cycles selected for the cycle basis.

The cycle basis is then resorted by length, vertex, and degree. The final cycle basis for the example linkage is shown in Fig. 5.14.

The cycle basis is also expressed as loops in Eqn. 5.15.

$$\begin{aligned} \text{Cycle 1} &: \{5,2,6,7,5\}, \\ \text{Cycle 2} &: \{5,2,6,4,3,8,5\}, \\ \text{Cycle 3} &: \{5,2,6,7,1,8,5\}. \end{aligned} \tag{5.15}$$

5.1.4 Example: Applying the Naming Convention

The cycle basis provides the list of the links that are joined in order along each loop. Following the cycle basis along each loop we assign a unique name for the joints as "j", followed by the number of the first link being joined, followed by "t", and followed by the number of the second link being joined.

For the example eight-bar the cycle basis is shown in Eqn. 5.15 and the joint names along the loops are Eqn. 5.16.

$$\begin{aligned}
\text{Loop 1} &: \{j5t2, j2t6, j6t7, j7t5\} \\
\text{Loop 2} &: \{j5t2, j2t6, j6t4, j4t3, j3t8, j8t5\} \\
\text{Loop 3} &: \{j5t2, j2t6, j6t7, j7t1, j1t8, j8t5\}.
\end{aligned} \tag{5.16}$$

The joints are then paired to represent the end points of the lines on the links along the linkage loops. These are paired in a {From Joint, To Joint} set. Because the ground link is the first link in our formulation we start with the joints along the loops as they enter the ground link, meaning the first joint in the list of pairs is the last joint shown in each joint list. The {From Joint, To Joint} pairs are shown in Eqn. 5.17.

$$\begin{aligned}
\text{Loop 1} &: \{j7t5, j5t2\}, \{j5t2, j2t6\}, \{j2t6, j6t7\}, \{j6t7, j7t5\} \\
\text{Loop 2} &: \{j8t5, j5t2\}, \{j5t2, j2t6\}, \{j2t6, j6t4\}, \{j6t4, j4t3\}, \{j4t3, j3t8\}, \{j3t8, j8t5\} \\
\text{Loop 3} &: \{j8t5, j5t2\}, \{j5t2, j2t6\}, \{j2t6, j6t7\}, \{j6t7, j7t1\}, \{j7t1, j1t8\}, \{j1t8, j8t5\}
\end{aligned} \tag{5.17}$$

The line between these paired joints 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", and followed by the last digits of the ending joint. The {From Joint, To Joint, Link Dimension} trio is shown in Eqn. 5.18.

$$
\begin{aligned}
\text{Loop 1} \quad : \quad & \{j7t5, j5t2, L7t5t2\}, \{j5t2, j2t6, L5t2t6\}, \\
& \{j2t6, j6t7, L2t6t7\}, \{j6t7, j7t5, L6t7t5\}. \\
\text{Loop 2} \quad : \quad & \{j8t5, j5t2, L8t5t2\}, \{j5t2, j2t6, L5t2t6\}, \\
& \{j2t6, j6t4, L2t6t4\}, \{j6t4, j4t3, L6t4t3\}, \\
& \{j4t3, j3t8, L4t3t8\}, \{j3t8, j8t5, L3t8t5\}. \\
\text{Loop 3} \quad : \quad & \{j8t5, j5t2, L8t5t2\}, \{j5t2, j2t6, L5t2t6\}, \\
& \{j2t6, j6t7, L2t6t7\}, \{j6t7, j7t1, L6t7t1\}, \\
& \{j7t1, j1t8, L7t1t8\}, \{j1t8, j8t5, L1t8t5\}.
\end{aligned} \tag{5.18}
$$

For evaluating angles we use the convention that all angles are positive counter clockwise. The global angle of a link is defined from the 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. Each of the {From Joint, To Joint, Link Dimension} trios represents a line along a particular link. That link number is the middle number of the name given to the line. Initially we assign the global angle for the link to each of the {From Joint, To Joint, Link Dimension} trios. The initial {From Joint, To Joint, Link Dimension, Link Angle} form is shown in Eqn. 5.19.

$$
\begin{aligned}
\text{Loop 1} \;:\; & \{j7t5, j5t2, L7t5t2, th5\}, \{j5t2, j2t6, L5t2t6, th2\}, \\
& \{j2t6, j6t7, L2t6t7, th6\}, \{j6t7, j7t5, L6t7t5, th7\}. \\
\text{Loop 2} \;:\; & \{j8t5, j5t2, L8t5t2, th5\}, \{j5t2, j2t6, L5t2t6, th2\}, \\
& \{j2t6, j6t4, L2t6t4, th6\}, \{j6t4, j4t3, L6t4t3, th4\}, \\
& \{j4t3, j3t8, L4t3t8, th3\}, \{j3t8, j8t5, L3t8t5, th8\}. \\
\text{Loop 3} \;:\; & \{j8t5, j5t2, L8t5t2, th5\}, \{j5t2, j2t6, L5t2t6, th2\}, \\
& \{j2t6, j6t7, L2t6t7, th6\}, \{j6t7, j7t1, L6t7t1, th7\}, \\
& \{j7t1, j1t8, L7t1t8, th1\}, \{j1t8, j8t5, L1t8t5, th8\}.
\end{aligned} \tag{5.19}
$$

This is not the final FTLA form because the specific angle to each line needs to account for the loop divergences and convergences. The vertices where the loops diverge (or converge) represent ternary or higher links. We define 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 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).

Using Eqn. 5.17 we subtract each term in Loop 1 from each term in Loop 2 to reveal where the loops diverge and converge. When loops diverge the "From Joint" will match, when loops

converge the "To Joint" will match. Loop 2 and Loop 1 converge at $j5t2$ and diverge at $j2t6$. The difference between the two loops fits the desired pattern at these joints, Eqn. 5.20.

$$\begin{aligned} \{j8t5, j5t2\} - \{j7t5, j5t2\} &= \{j8t5 - j7t5, 0\} \\ \{j2t6, j6t4\} - \{j2t6, j6t7\} &= \{0, j6t4 - j6t7\}. \end{aligned} \tag{5.20}$$

These joint pairs represent the endpoints of a line. The two lines intersect at a common joint and define the start and end of a fixed angle located at the common joint. The first fixed angle goes from line $L7t5t2$ to line $L8t5t2$. The second fixed angle goes from line $L2t6t7$ to line $L2t6t4$. The angle between these lines is named based on the reference line and the divergent or convergent line. The fixed angle starts at the reference line and ends at the divergent or convergent line. The angle between $L7t5t2$ and $L8t5t2$ is called $fix7t5t2tt8t5t2$. The angle between $L2t6t7$ and $L2t6t4$ is called $fix2t6t7tt2t6t4$.

After adding these fixed angles to the second loop the {From Joint, To Joint, Link Dimension, Link Angle} is updated as shown in Eqn. 5.21.

Loop 1

$$\{j7t5, j5t2, L7t5t2, th5\}, \{j5t2, j2t6, L5t2t6, th2\},$$

$$\{j2t6, j6t7, L2t6t7, th6\}, \{j6t7, j7t5, L6t7t5, th7\},$$

Loop 2

$$\{j8t5, j5t2, L8t5t2, fix7t5t2tt8t5t2 + th5\}, \{j5t2, j2t6, L5t2t6, th2\},$$

$$\{j2t6, j6t4, L2t6t4, fix2t6t7tt2t6t4 + th6\}, \{j6t4, j4t3, L6t4t3, th4\},$$

$$\{j4t3, j3t8, L4t3t8, th3\}, \{j3t8, j8t5, L3t8t5, th8\},$$

Loop 3

$$\{j8t5, j5t2, L8t5t2, th5\}, \{j5t2, j2t6, L5t2t6, th2\},$$

$$\{j2t6, j6t7, L2t6t7, th6\}, \{j6t7, j7t1, L6t7t1, th7\},$$

$$\{j7t1, j1t8, L7t1t8, th1\}, \{j1t8, j8t5, L1t8t5, th8\}. \tag{5.21}$$

The same process is applied to Loop 3 relative to Loop 1 and Loop 2. Loop 3 converges to Loop 1 at $j5t2$ following the same path along $L8t5t2$ as Loop 2 therefore the fixed angle about $j5t2$ is the same for both Loop 2 and Loop 3. Loop 3 also converges to Loop 2 at $j8t5$ and diverges from Loop 1 at $j6t7$. This example also shows that although the line $L2t6t7$ along Loop 3 diverges from Loop 2 at $j2t6$ this line was first used as part of Loop 1. Therefore, the appropriate fixed angle for this line in the FTLA for Loop 3 is simply $th6$, same as Loop 1.

The final FTLA convention is shown in Eqn. 5.22.

$$
\begin{aligned}
&\text{Loop 1}\\
&\quad \{j7t5, j5t2, L7t5t2, th5\}, \{j5t2, j2t6, L5t2t6, th2\},\\
&\quad \{j2t6, j6t7, L2t6t7, th6\}, \{j6t7, j7t5, L6t7t5, th7\},\\
&\text{Loop 2}\\
&\quad \{j8t5, j5t2, L8t5t2, fix7t5t2tt8t5t2 + th5\}, \{j5t2, j2t6, L5t2t6, th2\},\\
&\quad \{j2t6, j6t4, L2t6t4, fix2t6t7tt2t6t4 + th6\}, \{j6t4, j4t3, L6t4t3, th4\},\\
&\quad \{j4t3, j3t8, L4t3t8, th3\}, \{j3t8, j8t5, L3t8t5, th8\},\\
&\text{Loop 3}\\
&\quad \{j8t5, j5t2, L8t5t2, fix7t5t2tt8t5t2 + th5\}, \{j5t2, j2t6, L5t2t6, th2\},\\
&\quad \{j2t6, j6t7, L2t6t7, th6\}, \{j6t7, j7t1, L6t7t1, fix6t7t5tt6t7t1 + th7\},\\
&\quad \{j7t1, j1t8, L7t1t8, th1\}, \{j1t8, j8t5, L1t8t5, fix3t8t5tt1t8t5 + th8\}. \qquad (5.22)
\end{aligned}
$$

5.1.5 Example: Derivation of the Loop Equations

We convert from the FTLA convention {From Joint, To Joint, Link Dimension, Angle} to the final loop equations by taking the sum along each loop of the product

$$
\begin{aligned}
&X : (\text{Link Dimension}) * cos(Angle)\\
&Y : (\text{Link Dimension}) * sin(Angle) \qquad (5.23)
\end{aligned}
$$

Setting each of these summations equal to zero, the final loop equations are shown in Eqn. 5.24.

Loop 1

$$X : L5t2t6\cos(th2) + L7t5t2\cos(th5) + L2t6t7\cos(th6) + L6t7t5\cos(th7) = 0$$

$$Y : L5t2t6\sin(th2) + L7t5t2\sin(th5) + L2t6t7\sin(th6) + L6t7t5\sin(th7) = 0$$

Loop 2

$$X : L5t2t6\cos(th2) + L4t3t8\cos(th3) + L6t4t3\cos(th4)$$
$$+ L8t5t2\cos(fix7t5t2tt8t5t2 + th5) + L2t6t4\cos(fix2t6t7tt2t6t4 + th6)$$
$$+ L3t8t5\cos(th8) = 0$$

$$Y : L5t2t6\sin(th2) + L4t3t8\sin(th3) + L6t4t3\sin(th4)$$
$$+ L8t5t2\sin(fix7t5t2tt8t5t2 + th5) + L2t6t4\sin(fix2t6t7tt2t6t4 + th6)$$
$$+ L3t8t5\sin(th8) = 0$$

Loop 3

$$X : L7t1t8\cos(th1) + L5t2t6\cos(th2) + L8t5t2\cos(fix7t5t2tt8t5t2 + th5)$$
$$+ L2t6t7\cos(th6) + L6t7t1\cos(fix6t7t5tt6t7t1 + th7)$$
$$+ L1t8t5\cos(fix3t8t5tt1t8t5 + th8) = 0$$

$$Y : L7t1t8\sin(th1) + L5t2t6\sin(th2) + L8t5t2\sin(fix7t5t2tt8t5t2 + th5)$$
$$+ L2t6t7\sin(th6) + L6t7t1\sin(fix6t7t5tt6t7t1 + th7)$$
$$+ L1t8t5\sin(fix3t8t5tt1t8t5 + th8) = 0 \qquad (5.24)$$

5.2 Example: Configuration Analysis

The Dixon determinant is applied to the example eight-bar of Fig. 5.1.

5.2.1 Example: Derivation of the Complex Loop Equations

For the example eight-bar of Fig. 5.1 we convert the loop equations Eqn. 5.24 to complex form, Eqn. 5.25.

Loop 1

$$F : e^{ith6}L2t6t7 + e^{ith2}L5t2t6 + e^{ith7}L6t7t5 + e^{ith5}L7t5t2 = 0$$

$$F^* : e^{-ith6}L2t6t7 + e^{-ith2}L5t2t6 + e^{-ith7}L6t7t5 + e^{-ith5}L7t5t2 = 0$$

Loop 2

$$F : e^{ifix2t6t7tt2t6t4+ith6}L2t6t4 + e^{ith8}L3t8t5 + e^{ith3}L4t3t8 + e^{ith2}L5t2t6$$
$$+ e^{ith4}L6t4t3 + e^{ifix7t5t2tt8t5t2+ith5}L8t5t2 = 0$$

$$F^* : e^{-ifix2t6t7tt2t6t4-ith6}L2t6t4 + e^{-ith8}L3t8t5 + e^{-ith3}L4t3t8 + e^{-ith2}L5t2t6$$
$$+ e^{-ith4}L6t4t3 + e^{-ifix7t5t2tt8t5t2-ith5}L8t5t2 = 0$$

Loop 3

$$F : e^{ifix3t8t5tt1t8t5+ith8}L1t8t5 + e^{ith6}L2t6t7 + e^{ith2}L5t2t6$$
$$+ e^{ifix6t7t5tt6t7t1+ith7}L6t7t1 + e^{ith1}L7t1t8 + e^{ifix7t5t2tt8t5t2+ith5}L8t5t2 = 0$$

$$F^* : e^{-ifix3t8t5tt1t8t5-ith8}L1t8t5 + e^{-ith6}L2t6t7 + e^{-ith2}L5t2t6$$
$$+ e^{-ifix6t7t5tt6t7t1-ith7}L6t7t1 + e^{-ith1}L7t1t8 + e^{-ifix7t5t2tt8t5t2-ith5}L8t5t2 = 0 \tag{5.25}$$

Substituting the complex variable names produces the final form of the loop equations shown Eqn. 5.26.

Loop 1

$$F : L5t2t6\theta 2 + L7t5t2\theta 5 + L2t6t7\theta 6 + L6t7t5\theta 7 = 0$$

$$F^* : L5t2t6\theta 2c + L7t5t2\theta 5c + L2t6t7\theta 6c + L6t7t5\theta 7c = 0$$

Loop 2

$$F : L5t2t6\theta 2 + L4t3t8\theta 3 + L6t4t3\theta 4 + e^{ifix7t5t2tt8t5t2} L8t5t2\theta 5$$
$$+ e^{ifix2t6t7tt2t6t4} L2t6t4\theta 6 + L3t8t5\theta 8 = 0$$

$$F^* : L5t2t6\theta 2c + L4t3t8\theta 3c + L6t4t3\theta 4c + e^{-ifix7t5t2tt8t5t2} L8t5t2\theta 5c$$
$$+ e^{-ifix2t6t7tt2t6t4} L2t6t4\theta 6c + L3t8t5\theta 8c = 0$$

Loop 3

$$F : L7t1t8\theta 1 + L5t2t6\theta 2 + e^{ifix7t5t2tt8t5t2} L8t5t2\theta 5 + L2t6t7\theta 6$$
$$+ e^{ifix6t7t5tt6t7t1} L6t7t1\theta 7 + e^{ifix3t8t5tt1t8t5} L1t8t5\theta 8 = 0$$

$$F^* : L7t1t8\theta 1c + L5t2t6\theta 2c + e^{-ifix7t5t2tt8t5t2} L8t5t2\theta 5c + L2t6t7\theta 6c$$
$$+ e^{-ifix6t7t5tt6t7t1} L6t7t1\theta 7c + e^{-ifix3t8t5tt1t8t5} L1t8t5\theta 8c = 0 \tag{5.26}$$

5.2.2 Example: Checking the Dixon Determinant Solution

The complex loop equations are solved using the Dixon determinant procedure. The selected eigenvalue for this example is $\theta 3$.

To verify the results of the Dixon determinant an independent calculation is performed. It is known that the enhanced adjacency matrix represents a real assembly configuration and this exact assembly configuration must exist among the solutions of the Dixon determinant. The numerical value of the angles representing this configuration are independently calculated

from the FTLA and compared to each of the solutions of the Dixon determinant. Among the solutions to the Dixon determinant the exact numerical solution represented by the enhanced adjacency matrix must exist.

For this example, the results are shown in Eqn. 5.27 to two significant figures. This verifies that the proper Dixon determinant has been produced by the automation.

$$\begin{aligned}
\theta 3 = &\quad 0.35 - 0.94i, \text{ real angle } -1.2 &&\text{matches the eigenvalue number 13.}\\
\theta 1 = &\quad 1.0 + 0.037i, \text{ real angle } 0.037 &&\text{matches the eigenvector number 13.}\\
\theta 4 = &\quad -0.34 + 0.94i, \text{ real angle } 1.9 &&\text{matches the eigenvector number 13.}\\
\theta 6 = &\quad -0.14 + 0.99i, \text{ real angle } 1.7 &&\text{matches the eigenvector number 13.}\\
\theta 7 = &\quad 0.32 + 0.95i, \text{ real angle } 1.2 &&\text{matches the eigenvector number 13.}\\
\theta 8 = &\quad -0.14 + 0.99i, \text{ real angle } 1.7 &&\text{matches the eigenvector number 13.}
\end{aligned} \tag{5.27}$$

5.2.3 Example: Automated Jacobian

To automate the Jacobian the algorithm first calculates the Jacobian from Eqn. 5.24 and rearranges the columns so that the diagonal 2x2 blocks are full rank and then transforms the Jacobian into block upper triangular form.

Our example linkage has the input link and the ground link within the four-bar sub-linkage therefore singularities for each loop are independent. We expect singularities when the following features are collinear: $L2t6t7$ and $L6t7t5$, $L7t1t8$ and $L1t8t5$, or $L6t4t3$ and $L4t3t8$.

The Jacobian for the example linkage does not have full rank 2x2 submatrices along the diagonal, Eqn. 5.28.

$$\begin{bmatrix}
0 & 0 & 0 & \text{-L2t6t7 Sin[th6]} & \text{-L6t7t5 Sin[th7]} & 0 \\
0 & 0 & 0 & \text{L2t6t7 Cos[th6]} & \text{L6t7t5 Cos[th7]} & 0 \\
0 & \text{-L4t3t8 Sin[th3]} & \text{-L6t4t3 Sin[th4]} & \text{-L2t6t4 Sin[fix2t6t7tt2t6t4 +th6]} & 0 & \text{-L3t8t5 Sin[th8]} \\
0 & \text{L4t3t8 Cos[th3]} & \text{L6t4t3 Cos[th4]} & \text{L2t6t4 Cos[fix2t6t7tt2t6t4 +th6]} & 0 & \text{L3t8t5 Cos[th8]} \\
\text{-L7t1t8 Sin[th1]} & 0 & 0 & \text{-L2t6t7 Sin[th6]} & \text{-L6t7t1 Sin[fix6t7t5tt6t7t1 +th7]} & \text{-L1t8t5 Sin[fix3t8t5tt1t8t5 +th8]} \\
\text{L7t1t8 Cos[th1]} & 0 & 0 & \text{L2t6t7 Cos[th6]} & \text{L6t7t1 Cos[fix6t7t5tt6t7t1 +th7]} & \text{L1t8t5 Cos[fix3t8t5tt1t8t5 +th8]}
\end{bmatrix} \tag{5.28}$$

Rearranging the columns so that the diagonal is non-zero makes the diagonal 2x2 sub matrices full rank. There are several valid column arrangements that can be used. The algorithm chooses the first valid sorting which for this example is $\{4, 5, 2, 3, 1, 6\}$. The sorted Jacobian for the example linkage now has full rank diagonal 2x2 blocks, Eqn. 5.29.

$$\begin{bmatrix}
\text{-L2t6t7 Sin[th6]} & \text{-L6t7t5 Sin[th7]} & 0 & 0 & 0 & 0 \\
\text{L2t6t7 Cos[th6]} & \text{L6t7t5 Cos[th7]} & 0 & 0 & 0 & 0 \\
\text{-L2t6t4 Sin[fix2t6t7tt2t6t4 +th6]} & 0 & \text{-L4t3t8 Sin[th3]} & \text{-L6t4t3 Sin[th4]} & 0 & \text{-L3t8t5 Sin[th8]} \\
\text{L2t6t4 Cos[fix2t6t7tt2t6t4 +th6]} & 0 & \text{L4t3t8 Cos[th3]} & \text{L6t4t3 Cos[th4]} & 0 & \text{L3t8t5 Cos[th8]} \\
\text{-L2t6t7 Sin[th6]} & \text{-L6t7t1 Sin[fix6t7t5tt6t7t1 +th7]} & 0 & 0 & \text{-L7t1t8 Sin[th1]} & \text{-L1t8t5 Sin[fix3t8t5tt1t8t5 +th8]} \\
\text{L2t6t7 Cos[th6]} & \text{L6t7t1 Cos[fix6t7t5tt6t7t1 +th7]} & 0 & 0 & \text{L7t1t8 Cos[th1]} & \text{L1t8t5 Cos[fix3t8t5tt1t8t5 +th8]}
\end{bmatrix} \tag{5.29}$$

Applying the transform twice produces the block upper triangular form of the Jacobian shown in Eqn. 5.30.

$$\begin{bmatrix}
\text{-L2t6t7 Sin[th6]} & \text{-L6t7t5 Sin[th7]} & 0 & 0 & 0 & 0 \\
\text{L2t6t7 Cos[th6]} & \text{L6t7t5 Cos[th7]} & 0 & 0 & 0 & 0 \\
0 & 0 & \text{-L4t3t8 Sin[th3]} & \text{-L6t4t3 Sin[th4]} & 0 & \text{-L3t8t5 Sin[th8]} \\
0 & 0 & \text{L4t3t8 Cos[th3]} & \text{L6t4t3 Cos[th4]} & 0 & \text{L3t8t5 Cos[th8]} \\
0 & 0 & 0 & 0 & \text{-L7t1t8 Sin[th1]} & \text{-L1t8t5 Sin[fix3t8t5tt1t8t5 +th8]} \\
0 & 0 & 0 & 0 & \text{L7t1t8 Cos[th1]} & \text{L1t8t5 Cos[fix3t8t5tt1t8t5 +th8]}
\end{bmatrix} \tag{5.30}$$

Taking the determinant of each 2x2 block along the diagonal produces the three Jacobian factors shown in Eqn. 5.31. One of these factors is zero when one of the expected link pairs is collinear.

$$\begin{aligned}
J_1 :& - L2t6t7L6t7t5\cos(th7)\sin(th6) \\
&+ L2t6t7L6t7t5\cos(th6)\sin(th7) \\
J_2 :& - L4t3t8L6t4t3\cos(th4)\sin(th3) \\
&+ L4t3t8L6t4t3\cos(th3)\sin(th4) \\
J_3 :& - L1t8t5L7t1t8\cos(fix3t8t5tt1t8t5 + th8)\sin(th1) \\
&+ L1t8t5L7t1t8\cos(th1)\sin(fix3t8t5tt1t8t5 + th8).
\end{aligned} \tag{5.31}$$

5.3 Example: Tracking a Six-bar Linkage

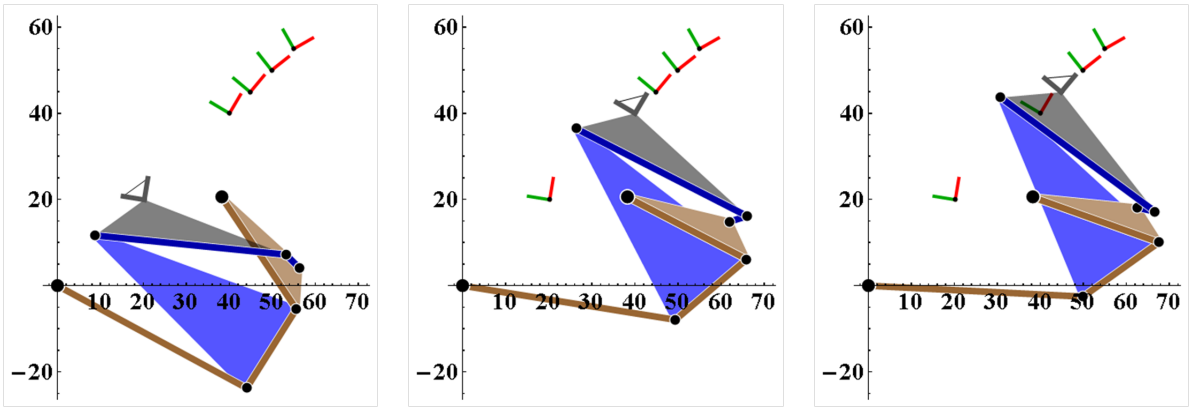
Portions of this section are from Parrish and McCarthy [31], used with permission.

This section shows how the signs of the Jacobian determinant factors are used to consistently track the assembly configuration of two six-bar linkages. The example linkages in this section do not use the automated naming convention.

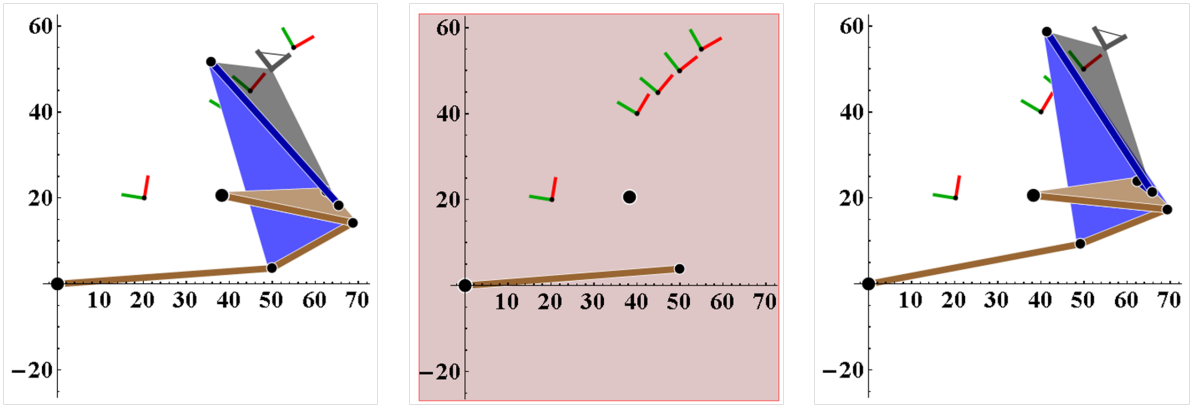
We start our evaluation with a linkage of a known configuration and a known input angle which reaches a desired task position. Because we use an input angle contained within the four-bar loop the branch and circuit for this linkage position is uniquely identified by the combination of three things, the input angle and the sign of the determinant of the Jacobians for each linkage loop, specifically the signs of J_1 and J_2.

Smooth movement occurs when a linkage remains on the same branch of the same circuit over the desired range of input angles therefore the signs of the determinants of the Jacobians must remain consistent throughout a desired range of input angles. For example, smooth motion occurs when J_1 is positive for all input angles and J_2 is negative for all input angles. For either J_1 or J_2 to change sign the linkage must have passed through a singularity such that J_1 or J_2 equates to zero.

A numerical approach is utilized such that we parameterize the input angle θ_1 contained within the four-bar loop and incrementally advance θ_1 over the range of desired input angles. At each incrementally advanced input angle we verify that there exists an assembly configuration with the desired set of Jacobian signs. If the desired set of signs does not exist for one of the incrementally advanced input angles the linkage cannot be assembled in the desired configuration for this input angle, therefore, the linkage is no longer on the same branch of the same circuit. We do not use the value of J_1 and J_2 for this evaluation, just the sign. To do this evaluation we use the Dixon determinant to provide all of the possible



(a) Task Position 1 (b) Task Position 2 (c) Task Position 3



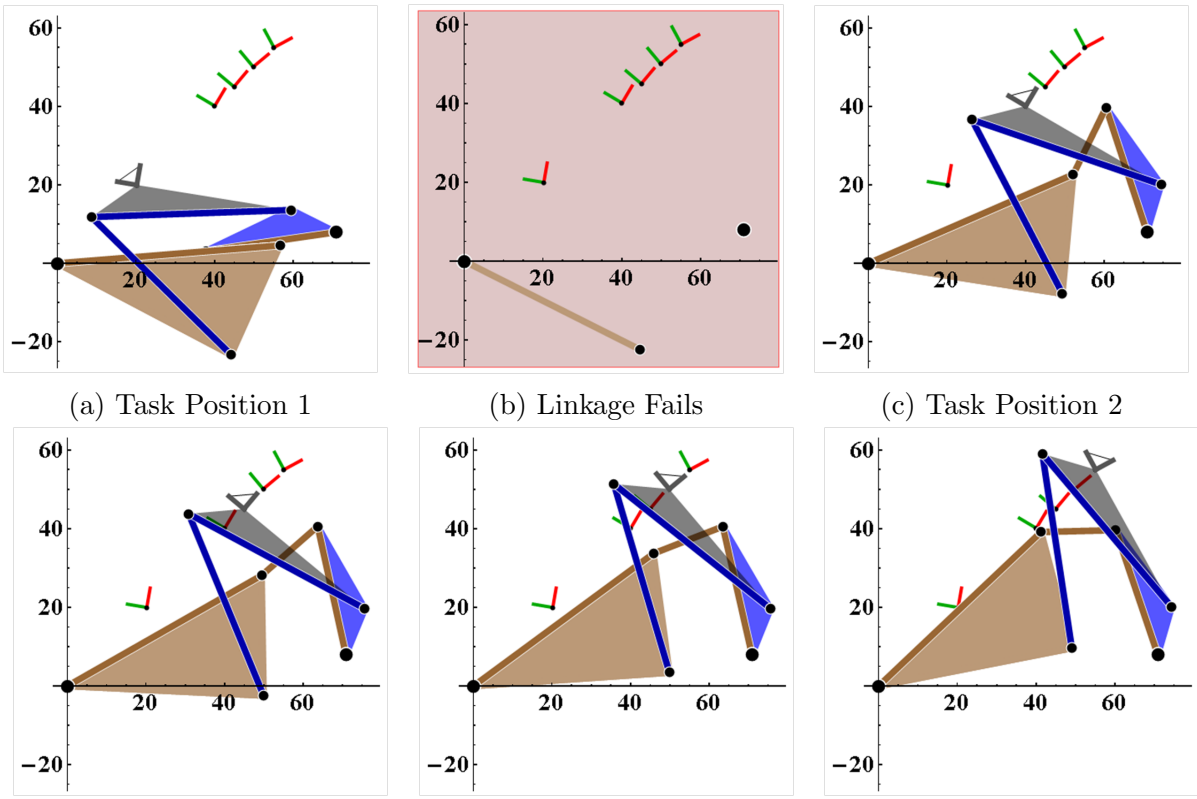
(d) Task Position 4 (e) Linkage Fails (f) Task Position 5

Figure 5.15: Example Watt I, candidate 57, that fails branch consistency check.

linkage assembly configurations available at each parametric input angle and determine the signs of the Jacobian determinants for each real solution.

Figure 5.15 shows an example Watt I linkage which does not contain the desired set of Jacobian determinant signs at one of the incremental input angles. For this synthesized linkage the two links in the second loop located by θ_3 and θ_5, as defined in Figure 2.12, cannot be assembled at the input angle in Fig. 5.15e therefore the linkage is no longer on the same branch of the same circuit.

Figure 5.16 shows an example Stephenson I linkage which does not contain the desired set of Jacobian determinant signs at three of the incremental input angles. For this synthesized linkage, the two links in the first loop located by θ_4 and θ_5, as defined in Figure 2.13, cannot



(a) Task Position 1 (b) Linkage Fails (c) Task Position 2

(d) Task Position 3 (e) Task Position 4 (f) Task Position 5

Figure 5.16: Example Stephenson I, candidate 120, that fails branch consistency check.

be assembled at the input angle in Fig. 5.16b therefore the linkage is no longer on the same branch of the same circuit.

This evaluation of branch consistency may not be sufficient to conclude that the linkage movement is smooth. A special case may exist such that the desired signs of J_1 and J_2 do not exist at a point between the incrementally evaluated input angles. A procedure for evaluating this special case is provided by Parrish and McCarthy [31] that uses numerical methods to find the input angle that minimizes the absolute value of the Jacobian.

5.4 Examples Conclusion

This chapter demonstrated the procedure to automate the linkage loop equations on an example eight-bar kinematic chain. The shortest path and edge elimination algorithm was demonstrated and the cycle basis established.

Also demonstrated was the automated derivation of the linkage loop equations by algorithmically applying a naming convention to the cycle basis. A convenient convention called FTLA was demonstrated that describes the link connections along each loop.

From the FTLA the automated linkage loop equations were derived for use in the Dixon determinant and the final Dixon determinant solution was verified by an independent dimensional check based on the enhanced adjacency matrix.

The automation to derive the factored Jacobian was demonstrated.

An example using the factored Jacobian determinant to track a particular linkage over a range of input angles is shown. The example linkages do not move smoothly over the entire input angle range.

Chapter 6

Classification of Linkages

This research enabled the establishment of a linkage classification convention called NATML that defines a linkage classification, or name, for every unique linkage with a ground connected input. Each linkage is uniquely identified by a specific five number index based on the smallest cycle basis through the common edge connecting the input and ground vertices of the adjacency graph.

Starting with the full set of adjacency matrices for the four-bar, six-bar, and eight-bar families, this research identified all of the unique linkages for each family. The NATML names correlate with the traditional names for the six-bar topologies, mechanisms and linkages.

6.1 Linkage Classification Basis

The NATML classification is five numbers $\{n, a, t, m, l\}$ that distinguish unique instances of kinematic chains at the logical levels ranging from the number of bars to the specific linkage. The index definitions are shown in Table 6.1.

Table 6.1: NATML definition.

Index	Title	Description
n	N-bar	Quantity of Links
a	Assortment	Quantity of Binary, Ternary, Quaternary, etc. Links
t	Topology	Connectivity Between the Links
m	Mechanism	Specific Ground Link
l	Linkage	Specific Input Link

The lowest level represents the unique linkages. The unique linkages are identified by the set of loop vertex degree lists based on the smallest cycle basis through the common edge between ground and input. This unique set of loop vertex degree lists is used as a basis for sorting the linkages. The unique linkages are sorted by

1. Lowest degree of the grounded vertex (binary first)
2. Shortest loop length of the first loop
3. Lowest vertex degree of the first loop, depth first
4. When equal, sort by the next loop

If two linkages are equal at every loop then the two linkages are not distinct because one can be obtained from the other by renumbering the links.

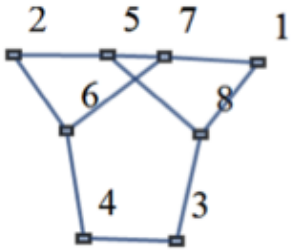
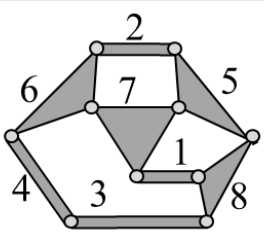
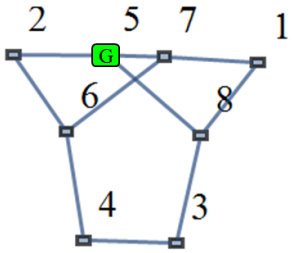
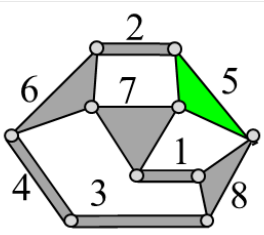
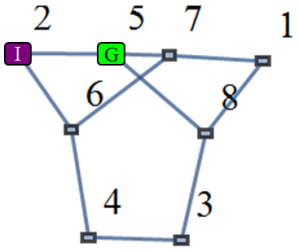
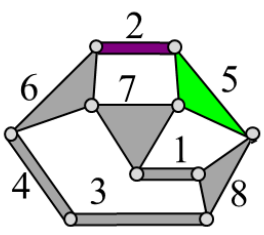
The linkages are sorted per the criteria within each mechanism group so that the first linkage within the group has the "smallest" set of loop vertex degree lists. The mechanism groups are sorted within the topologies based on the first linkage within each mechanism group. The topologies are sorted within the link assortments based on the first linkage within each topology. The link assortments are sorted within the N-bar family by the smallest quantity of links of each degree, lowest degree first. For example, the quantity of links of each degree in sorted order for the eight-bar family is shown in Table 6.2.

An example of the classification convention is applied to a linkage in Table 6.3.

Table 6.2: Ordered eight-bar link assortments.

Link Assortment	Quantity of Links by Type
4400	4 Binary, 4 Ternary, 0 Quaternary, 0 Quintenary
5210	5 Binary, 2 Ternary, 1 Quaternary, 0 Quintenary
6020	6 Binary, 0 Ternary, 2 Quaternary, 0 Quintenary

Table 6.3: Classification convention for an example linkage, NATML $\{8, 1, 5, 3, 1\}$.

Level	NATML	Example	Description	Detail/Sketch
N-bar	{n}	{8}	8-bar family	
Assortment	{n,a}	{8,1}	4400	4 Binary, 4 Ternary
Topology	{n,a,t}	{8,1,5}		
Mechanism	{n,a,t,m}	{8,1,5,3}		
Linkage	{n,a,t,m,l}	{8,1,5,3,1}		

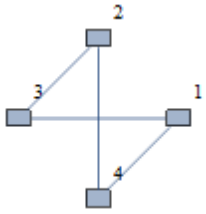
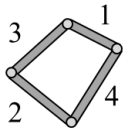
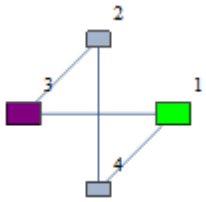
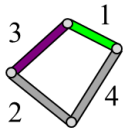
6.2 Linkage Classifications

6.2.1 Unique Four-bar Linkages

The four-bar family contains one unique link assortment, one unique topology, one unique mechanism, and one unique linkage with a ground-connected input.

Table 6.4 shows the link assortment, topology and linkage for the four-bar family. The one unique four-bar topology is represented by an adjacency matrix, an adjacency graph, and a topology sketch. Within that one topology the one unique linkage is listed and represented by NATML, vertex degree list, cycle basis, adjacency graph, and a linkage sketch. In both the adjacency graph and the linkage sketch the ground is shown in green and the input is shown in purple.

Table 6.4: Unique four-bar linkages by NATML

#	NATML	Adjacency Matrix, Vertex Degree List, Cycle Basis	Adjacency Graph	Topology and Linkage Sketch
		Link Assortment 4000		
	Topology {4,1,1} Four Bar	$\begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}$		
1	{4,1,1,1,1}	2 2 2 2 2 1 3 2 4 1		

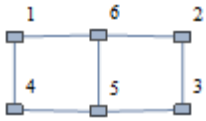
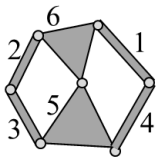
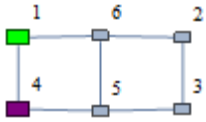
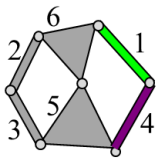
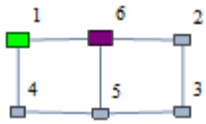
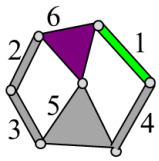
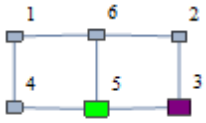
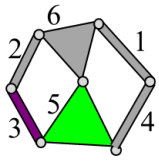
6.2.2 Unique Six-bar Linkages

The six-bar family contains one unique link assortment, two unique topologies, five unique mechanisms, and nine unique linkages with a ground-connected input.

Table 6.5 shows the link assortment, the two topologies and the nine linkages for the six-bar family. The two unique six-bar topologies are represented by an adjacency matrix, an adjacency graph, and a topology sketch. Within each topology the unique linkages are listed and represented by NATML, vertex degree list, cycle basis, adjacency graph, and a linkage

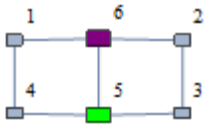
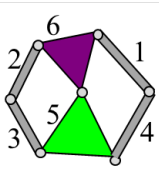
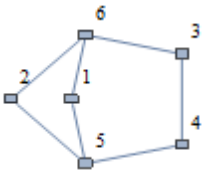
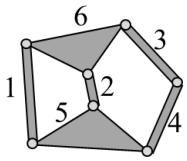
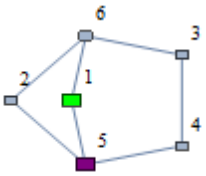
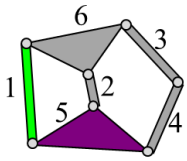
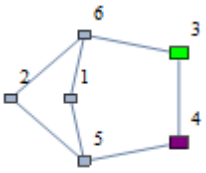
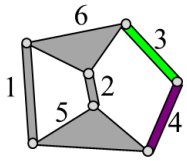
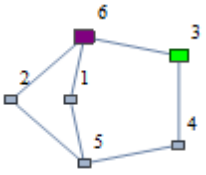
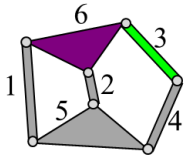
sketch. In both the adjacency graph and the linkage sketch the ground is shown in green and the input is shown in purple. The traditional names for these topologies and linkages are also shown for reference. Under the NATML name, the table notes the one six-bar linkage that partitions into simpler one-DoF linkages, the Watt IIb {6,1,1,2,2}.

Table 6.5: Unique six-bar linkages by NATML

#	NATML	Adjacency Matrix, Vertex Degree List, Cycle Basis	Adjacency Graph	Topology and Linkage Sketch
	Topology {6,1,1} Watt	Link Assortment 4200 $\begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 \end{pmatrix}$		
1	{6,1,1,1,1} Watt Ia	2 2 3 3 2 2 2 3 2 2 3 2 1 4 5 6 1 1 4 5 3 2 6 1		
2	{6,1,1,1,2} Watt Ib	2 3 3 2 2 2 3 2 2 3 2 2 1 6 5 4 1 1 6 2 3 5 4 1		
3	{6,1,1,2,1} Watt IIa	3 2 2 3 3 3 2 2 3 2 2 3 5 3 2 6 5 5 3 2 6 1 4 5		

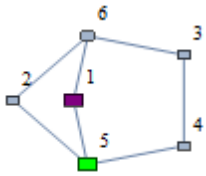
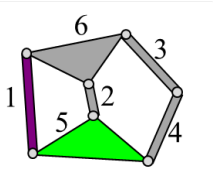
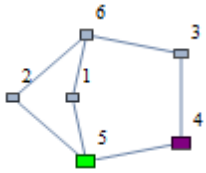
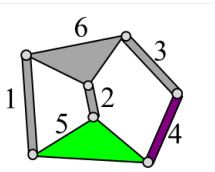
Continued on next page

Table 6.5 – continued from previous page

#	NATML	Adjacency Matrix, Vertex Degree List, Cycle Basis	Adjacency Graph	Topology and Linkage Sketch
4	{6,1,1,2,2} Watt IIb Partitions	3 3 2 2 3 3 3 2 2 3 5 6 1 4 5 5 6 2 3 5		
	Topology {6,1,2} Stephenson	$\begin{pmatrix} 0&0&0&0&1&1\\ 0&0&0&0&1&1\\ 0&0&0&1&0&1\\ 0&0&1&0&1&0\\ 1&1&0&1&0&0\\ 1&1&1&0&0&0 \end{pmatrix}$		
5	{6,1,2,1,1} Steph. Ia	2 3 2 3 2 2 3 2 2 3 2 1 5 2 6 1 1 5 4 3 6 1		
6	{6,1,2,2,1} Steph. IIa	2 2 3 2 3 2 2 2 3 2 3 2 3 4 5 1 6 3 3 4 5 2 6 3		
7	{6,1,2,2,2} Steph. IIb	2 3 2 3 2 2 2 3 2 3 2 2 3 6 1 5 4 3 3 6 2 5 4 3		

Continued on next page

Table 6.5 – continued from previous page

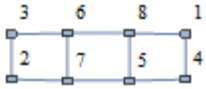
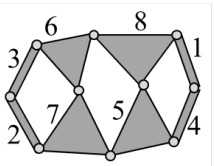
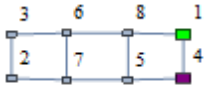
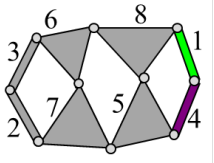
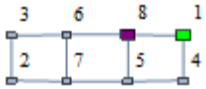
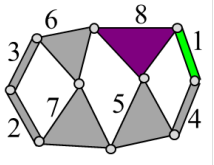
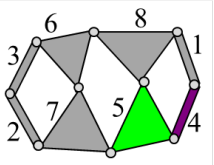
#	NATML	Adjacency Matrix, Vertex Degree List, Cycle Basis	Adjacency Graph	Topology and Linkage Sketch
8	{6,1,2,3,1} Steph. IIIa	3 2 3 2 3 3 2 3 2 2 3 5 1 6 2 5 5 1 6 3 4 5		
9	{6,1,2,3,2} Steph. IIIb	3 2 2 3 2 3 3 2 2 3 2 3 5 4 3 6 1 5 5 4 3 6 2 5		

6.2.3 Unique Eight-bar Linkages

The eight-bar family contains three unique link assortments, 16 unique topologies, 71 unique mechanisms, and 153 unique linkages with a ground-connected input.

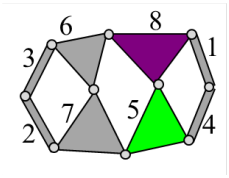
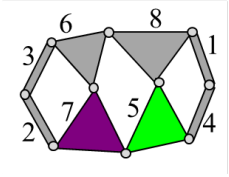
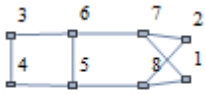
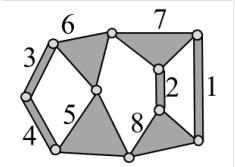
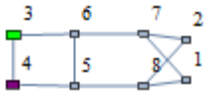
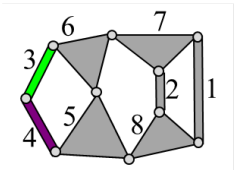
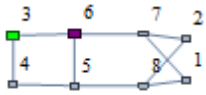
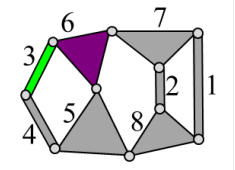
Table 6.6 shows the three link assortments, the 16 topologies and the 153 linkages for the eight-bar family. The 16 unique eight-bar topologies are represented by an adjacency matrix, an adjacency graph, and a topology sketch. Within each topology the unique linkages are listed and represented by NATML, vertex degree list, cycle basis, adjacency graph, and a linkage sketch. In both the adjacency graph and the linkage sketch the ground is shown in green and the input is shown in purple. Under the NATML name, the table notes the 24 eight-bar linkages that partition into simpler one-DoF linkages.

Table 6.6: Unique eight-bar linkages by NATML

#	NATML	Adjacency Matrix, Vertex Degree List, Cycle Basis	Adjacency Graph	Topology and Linkage Sketch
	Topology {8,1,1}	Link Assortment 4400 $\begin{pmatrix} 0&0&0&1&0&0&0&1\\ 0&0&1&0&0&0&1&0\\ 0&1&0&0&0&1&0&0\\ 1&0&0&0&1&0&0&0\\ 0&0&0&1&0&0&1&1\\ 0&0&1&0&0&0&1&1\\ 0&1&0&0&1&1&0&0\\ 1&0&0&0&1&1&0&0 \end{pmatrix}$		
1	{8,1,1,1,1}	2 2 3 3 2 2 2 3 3 3 3 2 2 2 3 3 2 2 3 3 2 1 4 5 8 1 1 4 5 7 6 8 1 1 4 5 7 2 3 6 8 1		
2	{8,1,1,1,2}	2 3 3 2 2 2 3 3 3 3 2 2 2 3 3 2 2 3 3 2 2 1 8 5 4 1 1 8 6 7 5 4 1 1 8 6 3 2 7 5 4 1		
3	{8,1,1,2,1}	3 2 2 3 3 3 2 2 3 3 3 3 3 2 2 3 3 2 2 3 3 5 4 1 8 5 5 4 1 8 6 7 5 5 4 1 8 6 3 2 7 5		

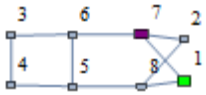
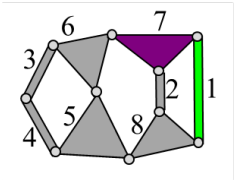
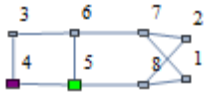
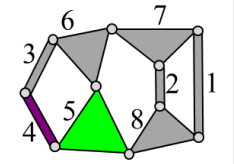
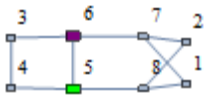
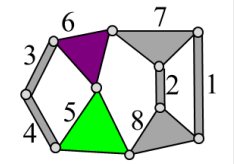
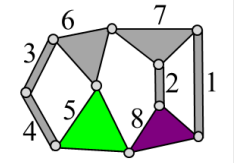
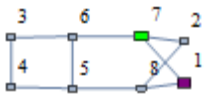
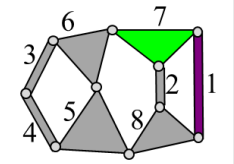
Continued on next page

Table 6.6 – continued from previous page

#	NATML	Adjacency Matrix, Vertex Degree List, Cycle Basis	Adjacency Graph	Topology and Linkage Sketch
4	{8,1,1,2,2} Partitions	3 3 2 2 3 3 3 3 3 3 3 3 3 2 2 3 3 5 8 1 4 5 5 8 6 7 5 5 8 6 3 2 7 5		
5	{8,1,1,2,3} Partitions	3 3 3 3 3 3 3 2 2 3 3 3 3 3 3 3 2 2 3 5 7 6 8 5 5 7 2 3 6 8 5 5 7 6 8 1 4 5		
	Topology {8,1,2}	$\begin{pmatrix} 0&0&0&0&0&0&1&1\\ 0&0&0&0&0&0&1&1\\ 0&0&0&1&0&1&0&0\\ 0&0&1&0&1&0&0&0\\ 0&0&0&1&0&1&0&1\\ 0&0&1&0&1&0&1&0\\ 1&1&0&0&0&1&0&0\\ 1&1&0&0&1&0&0&0 \end{pmatrix}$		
6	{8,1,2,1,1}	2 2 3 3 2 2 2 3 3 2 3 3 2 2 2 3 3 2 3 3 2 3 4 5 6 3 3 4 5 8 1 7 6 3 3 4 5 8 2 7 6 3		
7	{8,1,2,1,2}	2 3 3 2 2 2 3 3 2 3 3 2 2 2 3 3 2 3 3 2 2 3 6 5 4 3 3 6 7 1 8 5 4 3 3 6 7 2 8 5 4 3		

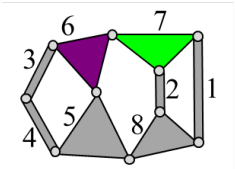
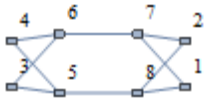
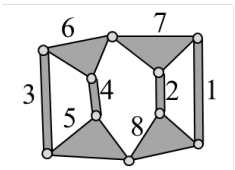
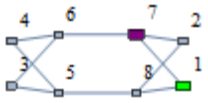
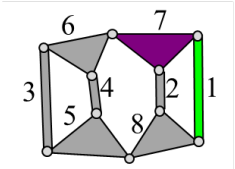
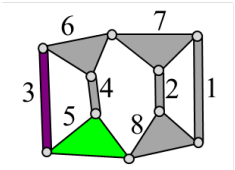
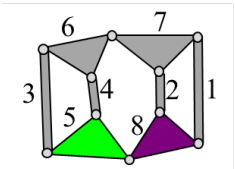
Continued on next page

Table 6.6 – continued from previous page

#	NATML	Adjacency Matrix, Vertex Degree List, Cycle Basis	Adjacency Graph	Topology and Linkage Sketch
8	{8,1,2,2,1}	2 3 2 3 2 2 3 3 3 3 2 2 3 3 2 2 3 3 2 1 7 2 8 1 1 7 6 5 8 1 1 7 6 3 4 5 8 1		
9	{8,1,2,3,1}	3 2 2 3 3 3 2 2 3 3 2 3 3 3 2 2 3 3 2 3 3 5 4 3 6 5 5 4 3 6 7 1 8 5 5 4 3 6 7 2 8 5		
10	{8,1,2,3,2} Partitions	3 3 2 2 3 3 3 3 2 3 3 3 3 3 2 3 3 5 6 3 4 5 5 6 7 1 8 5 5 6 7 2 8 5		
11	{8,1,2,3,3}	3 3 2 3 3 3 3 3 2 3 3 3 3 3 2 3 3 2 2 3 5 8 1 7 6 5 5 8 2 7 6 5 5 8 1 7 6 3 4 5		
12	{8,1,2,4,1}	3 2 3 2 3 3 2 3 3 3 3 3 2 3 3 2 2 3 3 7 1 8 2 7 7 1 8 5 6 7 7 1 8 5 4 3 6 7		

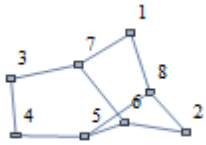
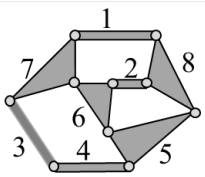
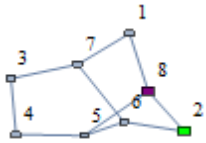
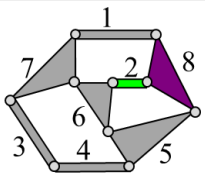
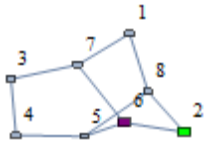
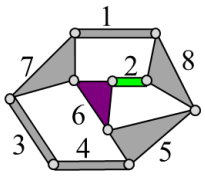
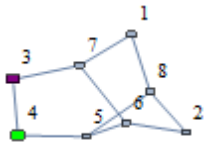
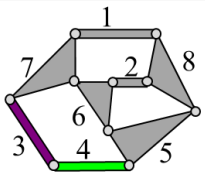
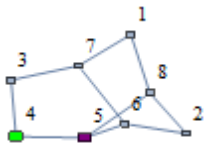
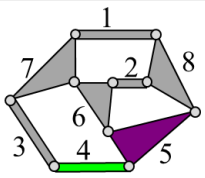
Continued on next page

Table 6.6 – continued from previous page

#	NATML	Adjacency Matrix, Vertex Degree List, Cycle Basis	Adjacency Graph	Topology and Linkage Sketch
13	{8,1,2,4,2}	3 3 3 3 2 3 3 3 3 3 2 3 3 3 2 2 3 3 2 3 7 6 5 8 1 7 7 6 5 8 2 7 7 6 3 4 5 8 1 7		
	Topology {8,1,3}	$\begin{pmatrix} 0&0&0&0&0&0&1&1\\ 0&0&0&0&0&0&1&1\\ 0&0&0&0&1&1&0&0\\ 0&0&0&0&1&1&0&0\\ 0&0&1&1&0&0&0&1\\ 0&0&1&1&0&0&1&0\\ 1&1&0&0&0&1&0&0\\ 1&1&0&0&1&0&0&0 \end{pmatrix}$		
14	{8,1,3,1,1}	2 3 2 3 2 2 3 3 2 3 3 2 2 3 3 2 3 3 2 1 7 2 8 1 1 7 6 3 5 8 1 1 7 6 4 5 8 1		
15	{8,1,3,2,1}	3 2 3 2 3 3 2 3 3 2 3 3 3 2 3 3 2 3 3 5 3 6 4 5 5 3 6 7 1 8 5 5 3 6 7 2 8 5		
16	{8,1,3,2,2}	3 3 2 3 3 2 3 3 3 2 3 3 2 3 3 3 2 3 3 2 3 5 8 1 7 6 3 5 5 8 1 7 6 4 5 5 8 2 7 6 3 5		

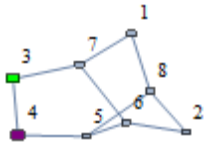
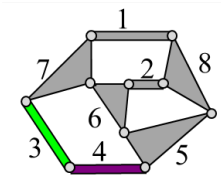
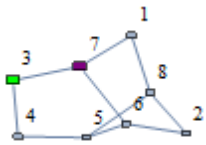
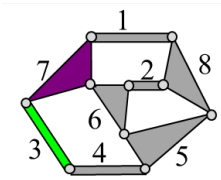
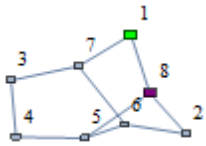
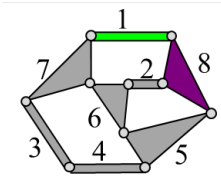
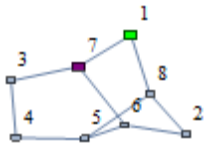
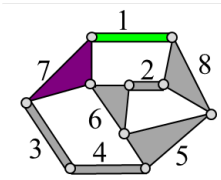
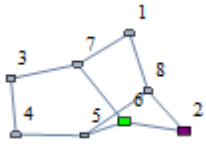
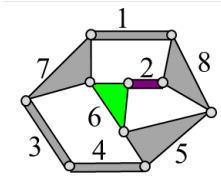
Continued on next page

Table 6.6 – continued from previous page

#	NATML	Adjacency Matrix, Vertex Degree List, Cycle Basis	Adjacency Graph	Topology and Linkage Sketch
	Topology {8,1,4}	$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$		
17	{8,1,4,1,1}	2 3 3 3 2 2 3 2 3 3 2 2 3 3 2 2 3 3 2 2 8 5 6 2 2 8 1 7 6 2 2 8 5 4 3 7 6 2		
18	{8,1,4,1,2}	2 3 3 3 2 2 3 3 2 3 2 2 3 3 2 2 3 3 2 2 6 5 8 2 2 6 7 1 8 2 2 6 7 3 4 5 8 2		
19	{8,1,4,2,1}	2 2 3 3 3 2 2 2 3 2 3 3 2 2 2 3 3 2 3 3 2 4 3 7 6 5 4 4 3 7 1 8 5 4 4 3 7 6 2 8 5 4		
20	{8,1,4,2,2}	2 3 3 3 2 2 2 3 3 2 3 2 2 2 3 3 2 3 3 2 2 4 5 6 7 3 4 4 5 8 1 7 3 4 4 5 8 2 6 7 3 4		

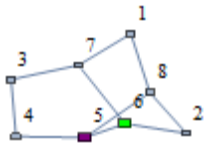
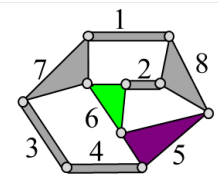
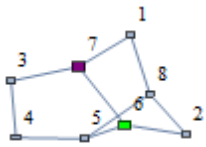
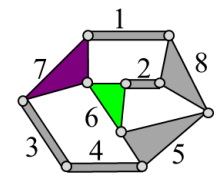
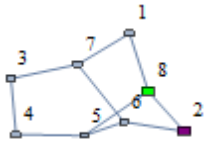
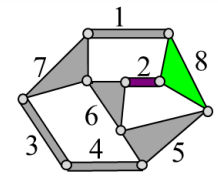
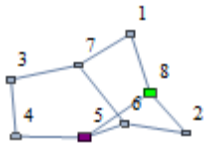
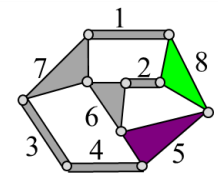
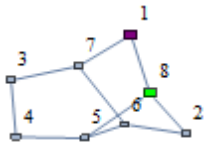
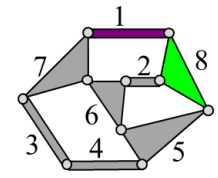
Continued on next page

Table 6.6 – continued from previous page

#	NATML	Adjacency Matrix, Vertex Degree List, Cycle Basis	Adjacency Graph	Topology and Linkage Sketch
21	{8,1,4,3,1}	2 2 3 3 3 2 2 2 3 3 2 3 2 2 2 3 3 2 3 3 2 3 4 5 6 7 3 3 4 5 8 1 7 3 3 4 5 8 2 6 7 3		
22	{8,1,4,3,2}	2 3 3 3 2 2 2 3 2 3 3 2 2 2 3 3 2 3 3 2 2 3 7 6 5 4 3 3 7 1 8 5 4 3 3 7 6 2 8 5 4 3		
23	{8,1,4,4,1}	2 3 2 3 3 2 2 3 3 3 3 2 2 3 3 2 2 3 2 1 8 2 6 7 1 1 8 5 6 7 1 1 8 5 4 3 7 1		
24	{8,1,4,4,2}	2 3 3 2 3 2 2 3 3 3 3 2 2 3 2 2 3 3 2 1 7 6 2 8 1 1 7 6 5 8 1 1 7 3 4 5 8 1		
25	{8,1,4,5,1}	3 2 3 3 3 3 2 3 2 3 3 3 2 3 3 2 2 3 3 6 2 8 5 6 6 2 8 1 7 6 6 2 8 5 4 3 7 6		

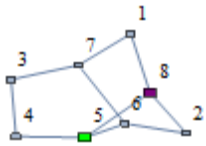
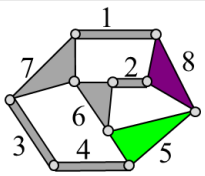
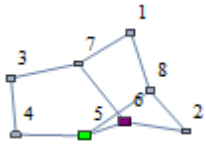
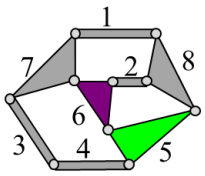
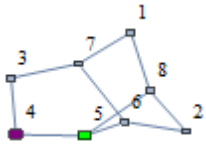
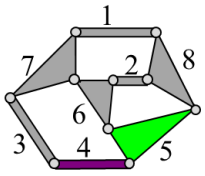
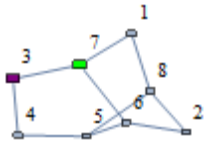
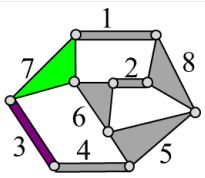
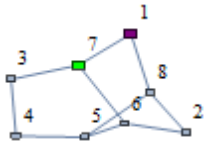
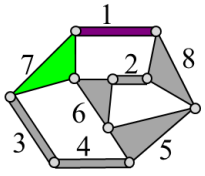
Continued on next page

Table 6.6 – continued from previous page

#	NATML	Adjacency Matrix, Vertex Degree List, Cycle Basis	Adjacency Graph	Topology and Linkage Sketch
26	{8,1,4,5,2}	3 3 3 2 3 3 3 2 2 3 3 3 3 3 2 3 3 6 5 8 2 6 6 5 4 3 7 6 6 5 8 1 7 6		
27	{8,1,4,5,3}	3 3 2 2 3 3 3 3 2 3 2 3 3 3 2 3 3 3 6 7 3 4 5 6 6 7 1 8 2 6 6 7 1 8 5 6		
28	{8,1,4,6,1}	3 2 3 3 3 3 2 3 3 2 3 3 2 3 3 2 2 3 3 8 2 6 5 8 8 2 6 7 1 8 8 2 6 7 3 4 5 8		
29	{8,1,4,6,2}	3 3 3 2 3 3 3 3 3 2 3 3 3 2 2 3 2 3 8 5 6 2 8 8 5 6 7 1 8 8 5 4 3 7 1 8		
30	{8,1,4,6,3}	3 2 3 3 2 3 3 2 3 3 3 3 3 2 3 2 2 3 3 8 1 7 6 2 8 8 1 7 6 5 8 8 1 7 3 4 5 8		

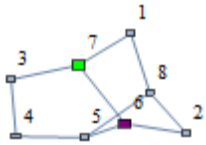
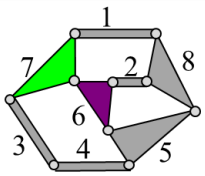
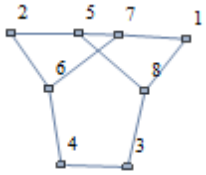
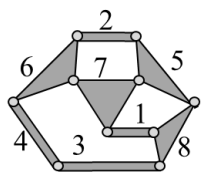
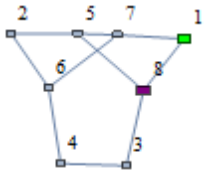
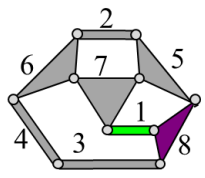
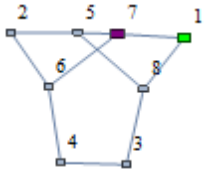
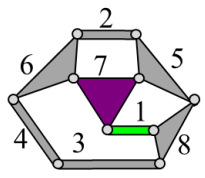
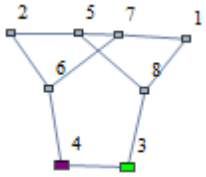
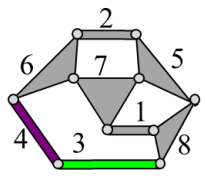
Continued on next page

Table 6.6 – continued from previous page

#	NATML	Adjacency Matrix, Vertex Degree List, Cycle Basis	Adjacency Graph	Topology and Linkage Sketch
31	{8,1,4,7,1}	3 3 2 3 3 3 3 2 3 3 3 3 3 2 3 2 2 3 5 8 2 6 5 5 8 1 7 6 5 5 8 1 7 3 4 5		
32	{8,1,4,7,2}	3 3 2 3 3 3 3 3 2 2 3 3 3 3 2 3 3 5 6 2 8 5 5 6 7 3 4 5 5 6 7 1 8 5		
33	{8,1,4,7,3}	3 2 2 3 3 3 3 2 2 3 2 3 3 3 2 2 3 3 2 3 3 5 4 3 7 6 5 5 4 3 7 1 8 5 5 4 3 7 6 2 8 5		
34	{8,1,4,8,1}	3 2 2 3 3 3 3 2 2 3 3 2 3 3 2 2 3 3 2 3 3 7 3 4 5 6 7 7 3 4 5 8 1 7 7 3 4 5 8 2 6 7		
35	{8,1,4,8,2}	3 2 3 2 3 3 3 2 3 3 3 3 3 2 3 3 2 2 3 7 1 8 2 6 7 7 1 8 5 6 7 7 1 8 5 4 3 7		

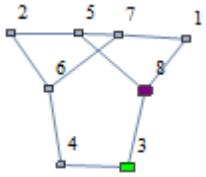
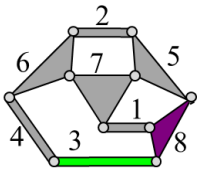
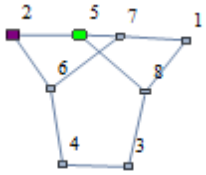
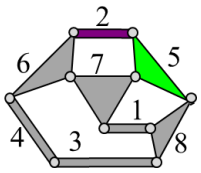
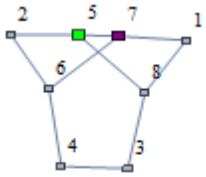
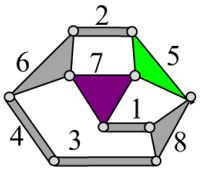
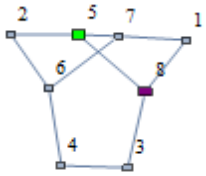
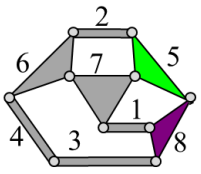
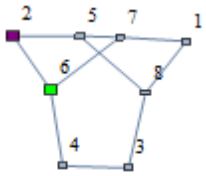
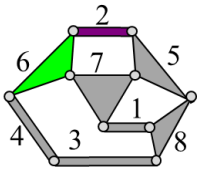
Continued on next page

Table 6.6 – continued from previous page

#	NATML	Adjacency Matrix, Vertex Degree List, Cycle Basis	Adjacency Graph	Topology and Linkage Sketch
36	{8,1,4,8,3}	3 3 2 3 2 3 3 3 3 2 2 3 3 3 3 3 2 3 7 6 2 8 1 7 7 6 5 4 3 7 7 6 5 8 1 7		
	Topology {8,1,5}	$\begin{pmatrix} 0&0&0&0&0&0&1&1\\ 0&0&0&0&1&1&0&0\\ 0&0&0&1&0&0&0&1\\ 0&0&1&0&0&1&0&0\\ 0&1&0&0&0&0&1&1\\ 0&1&0&1&0&0&1&0\\ 1&0&0&0&1&1&0&0\\ 1&0&1&0&1&0&0&0 \end{pmatrix}$		
37	{8,1,5,1,1}	2 3 3 3 2 2 3 2 2 3 3 2 2 3 3 2 3 3 2 1 8 5 7 1 1 8 3 4 6 7 1 1 8 5 2 6 7 1		
38	{8,1,5,1,2}	2 3 3 3 2 2 3 3 2 2 3 2 2 3 3 2 3 3 2 1 7 5 8 1 1 7 6 4 3 8 1 1 7 6 2 5 8 1		
39	{8,1,5,2,1}	2 2 3 2 3 3 2 2 2 3 3 2 3 2 2 2 3 3 3 3 2 3 4 6 2 5 8 3 3 4 6 7 1 8 3 3 4 6 7 5 8 3		

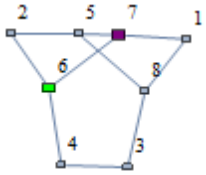
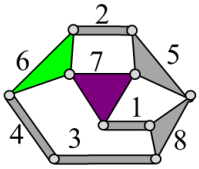
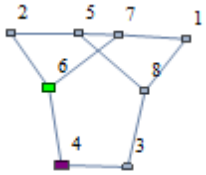
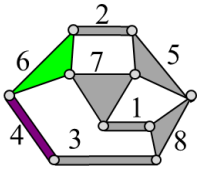
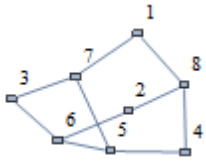
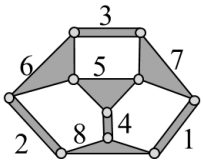
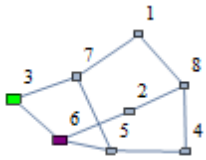
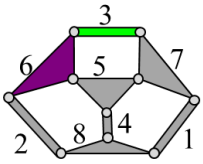
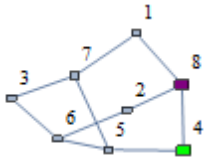
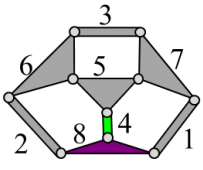
Continued on next page

Table 6.6 – continued from previous page

#	NATML	Adjacency Matrix, Vertex Degree List, Cycle Basis	Adjacency Graph	Topology and Linkage Sketch
40	{8,1,5,2,2}	2 3 2 3 3 2 2 2 3 3 2 3 2 2 2 3 3 3 3 2 2 3 8 1 7 6 4 3 3 8 5 2 6 4 3 3 8 5 7 6 4 3		
41	{8,1,5,3,1}	3 2 3 3 3 3 2 3 2 2 3 3 3 2 3 3 2 3 3 5 2 6 7 5 5 2 6 4 3 8 5 5 2 6 7 1 8 5		
42	{8,1,5,3,2}	3 3 2 3 3 3 3 3 2 3 3 3 3 2 2 3 3 5 7 1 8 5 5 7 6 2 5 5 7 6 4 3 8 5		
43	{8,1,5,3,3}	3 3 2 3 3 3 3 2 2 3 2 3 3 3 2 3 3 2 3 5 8 1 7 5 5 8 3 4 6 2 5 5 8 1 7 6 2 5		
44	{8,1,5,4,1}	3 2 3 3 3 3 2 3 3 2 2 3 3 2 3 3 2 3 3 6 2 5 7 6 6 2 5 8 3 4 6 6 2 5 8 1 7 6		

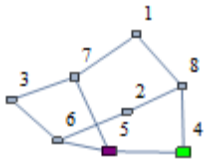
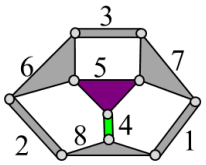
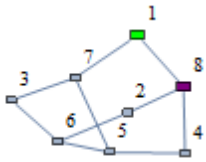
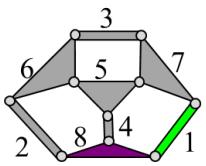
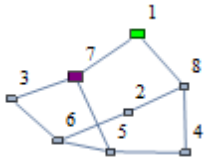
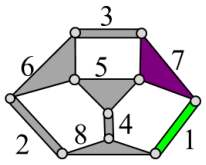
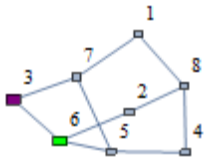
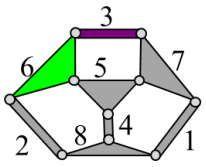
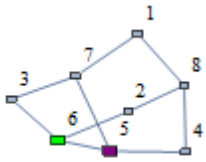
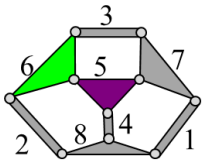
Continued on next page

Table 6.6 – continued from previous page

#	NATML	Adjacency Matrix, Vertex Degree List, Cycle Basis	Adjacency Graph	Topology and Linkage Sketch
45	{8,1,5,4,2}	3 3 3 2 3 3 3 2 3 2 2 3 3 3 2 3 3 2 3 6 7 5 2 6 6 7 1 8 3 4 6 6 7 1 8 5 2 6		
46	{8,1,5,4,3}	3 2 2 3 2 3 3 3 2 2 3 3 2 3 3 2 2 3 3 3 3 6 4 3 8 1 7 6 6 4 3 8 5 2 6 6 4 3 8 5 7 6		
	Topology {8,1,6}	$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$		
47	{8,1,6,1,1}	2 3 3 3 2 2 3 2 3 2 3 2 2 3 2 3 2 3 3 2 3 6 5 7 3 3 6 2 8 1 7 3 3 6 2 8 4 5 7 3		
48	{8,1,6,2,1}	2 3 2 3 3 2 2 3 2 3 3 2 2 3 2 3 2 3 3 2 4 8 1 7 5 4 4 8 2 6 5 4 4 8 1 7 3 6 5 4		

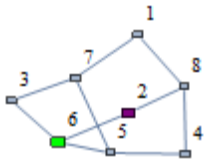
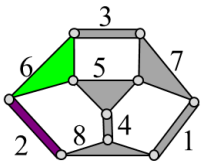
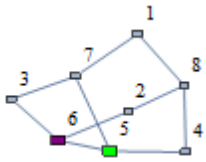
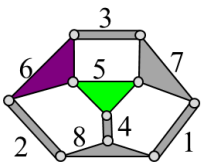
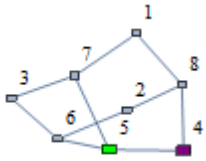
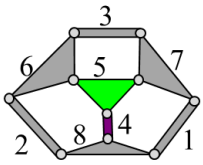
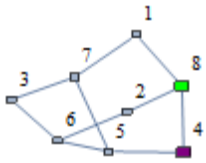
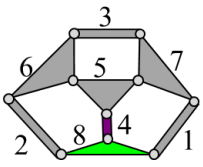
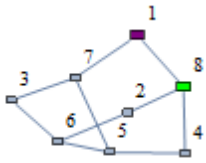
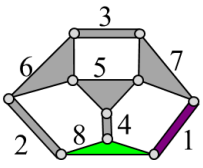
Continued on next page

Table 6.6 – continued from previous page

#	NATML	Adjacency Matrix, Vertex Degree List, Cycle Basis	Adjacency Graph	Topology and Linkage Sketch
49	{8,1,6,2,2}	2 3 3 2 3 2 2 3 3 2 3 2 2 3 3 2 3 2 3 2 4 5 6 2 8 4 4 5 7 1 8 4 4 5 6 3 7 1 8 4		
50	{8,1,6,3,1}	2 3 2 3 3 2 2 3 2 3 2 3 2 2 3 2 3 3 3 2 1 8 4 5 7 1 1 8 2 6 3 7 1 1 8 2 6 5 7 1		
51	{8,1,6,3,2}	2 3 3 2 3 2 2 3 2 3 2 3 2 2 3 3 3 2 3 2 1 7 5 4 8 1 1 7 3 6 2 8 1 1 7 5 6 2 8 1		
52	{8,1,6,4,1}	3 2 3 3 3 3 2 3 2 3 2 3 3 2 3 2 3 2 3 3 6 3 7 5 6 6 3 7 1 8 2 6 6 3 7 1 8 4 5 6		
53	{8,1,6,4,2}	3 3 3 2 3 3 3 2 3 2 3 3 3 3 2 3 2 3 6 5 7 3 6 6 5 4 8 2 6 6 5 7 1 8 2 6		

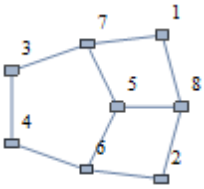
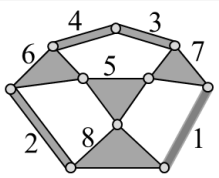
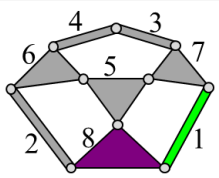
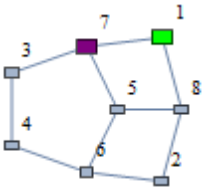
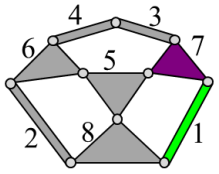
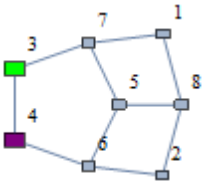
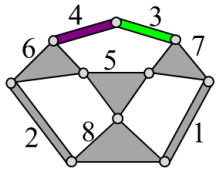
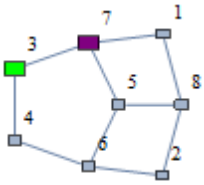
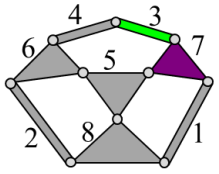
Continued on next page

Table 6.6 – continued from previous page

#	NATML	Adjacency Matrix, Vertex Degree List, Cycle Basis	Adjacency Graph	Topology and Linkage Sketch
54	{8,1,6,4,3}	3 2 3 2 3 3 3 2 3 2 3 2 3 3 2 3 2 3 3 3 6 2 8 4 5 6 6 2 8 1 7 3 6 6 2 8 1 7 5 6		
55	{8,1,6,5,1}	3 3 2 3 3 3 3 2 3 2 3 3 3 2 3 2 3 3 5 6 3 7 5 5 6 2 8 4 5 5 6 2 8 1 7 5		
56	{8,1,6,5,2}	3 2 3 2 3 3 3 2 3 2 3 3 3 2 3 2 3 2 3 3 5 4 8 1 7 5 5 4 8 2 6 5 5 4 8 1 7 3 6 5		
57	{8,1,6,6,1}	3 2 3 3 2 3 3 2 3 3 2 3 3 2 3 3 2 3 2 3 8 4 5 6 2 8 8 4 5 7 1 8 8 4 5 6 3 7 1 8		
58	{8,1,6,6,2}	3 2 3 3 2 3 3 2 3 2 3 2 3 3 2 3 3 3 2 3 8 1 7 5 4 8 8 1 7 3 6 2 8 8 1 7 5 6 2 8		

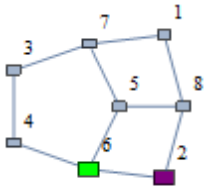
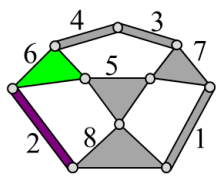
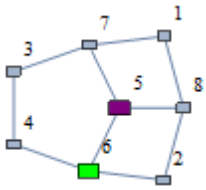
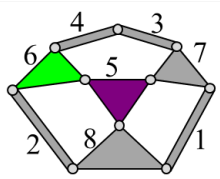
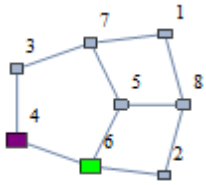
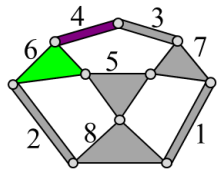
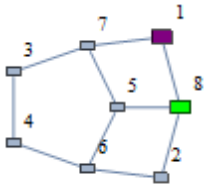
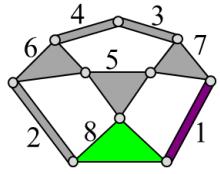
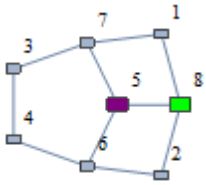
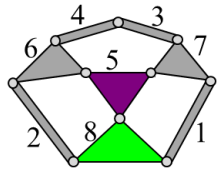
Continued on next page

Table 6.6 – continued from previous page

#	NATML	Adjacency Matrix, Vertex Degree List, Cycle Basis	Adjacency Graph	Topology and Linkage Sketch
	Topology {8,1,7}	$\begin{pmatrix} 0&0&0&0&0&0&1&1\\ 0&0&0&0&0&1&0&1\\ 0&0&0&1&0&0&1&0\\ 0&0&1&0&0&1&0&0\\ 0&0&0&0&0&1&1&1\\ 0&1&0&1&1&0&0&0\\ 1&0&1&0&1&0&0&0\\ 1&1&0&0&1&0&0&0 \end{pmatrix}$		
59	{8,1,7,1,1}	2 3 3 3 2 2 3 2 3 3 3 2 2 3 2 3 2 2 3 2 1 8 5 7 1 1 8 2 6 5 7 1 1 8 2 6 4 3 7 1		
60	{8,1,7,1,2}	2 3 3 3 2 2 3 3 3 2 3 2 2 3 2 2 3 2 3 2 1 7 5 8 1 1 7 5 6 2 8 1 1 7 3 4 6 2 8 1		
61	{8,1,7,2,1}	2 2 3 3 3 2 2 2 3 2 3 2 3 2 2 2 3 2 3 3 3 2 3 4 6 5 7 3 3 4 6 2 8 1 7 3 3 4 6 2 8 5 7 3		
62	{8,1,7,2,2}	2 3 3 3 2 2 2 3 2 3 2 3 2 2 2 3 2 3 3 3 2 2 3 7 5 6 4 3 3 7 1 8 2 6 4 3 3 7 1 8 5 6 4 3		

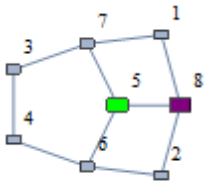
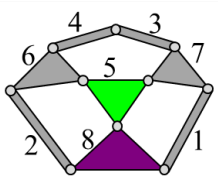
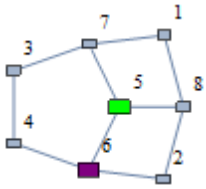
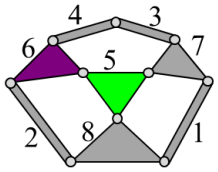
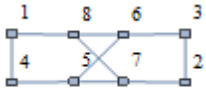
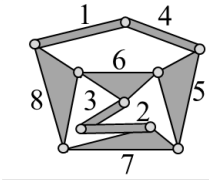
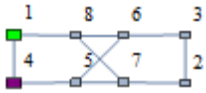
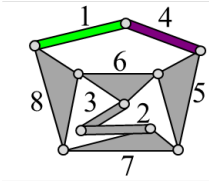
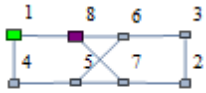
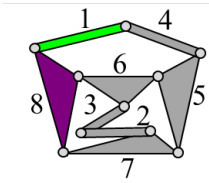
Continued on next page

Table 6.6 – continued from previous page

#	NATML	Adjacency Matrix, Vertex Degree List, Cycle Basis	Adjacency Graph	Topology and Linkage Sketch
63	{8,1,7,3,1}	3 2 3 3 3 3 2 3 2 3 3 3 3 2 3 2 3 2 2 3 6 2 8 5 6 6 2 8 1 7 5 6 6 2 8 1 7 3 4 6		
64	{8,1,7,3,2}	3 3 3 2 3 3 3 3 2 2 3 3 3 3 2 3 2 3 6 5 8 2 6 6 5 7 3 4 6 6 5 7 1 8 2 6		
65	{8,1,7,3,3}	3 2 2 3 3 3 3 2 2 3 2 3 2 3 3 2 2 3 2 3 3 3 6 4 3 7 5 6 6 4 3 7 1 8 2 6 6 4 3 7 1 8 5 6		
66	{8,1,7,4,1}	3 2 3 3 3 3 2 3 3 3 2 3 3 2 3 2 2 3 2 3 8 1 7 5 8 8 1 7 5 6 2 8 8 1 7 3 4 6 2 8		
67	{8,1,7,4,2}	3 3 3 2 3 3 3 3 2 3 3 3 3 2 2 3 2 3 8 5 6 2 8 8 5 7 1 8 8 5 6 4 3 7 1 8		

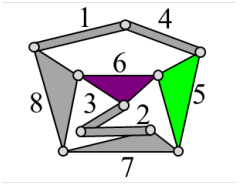
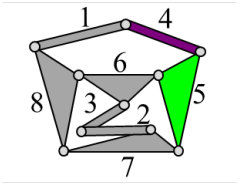
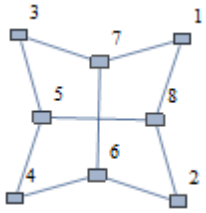
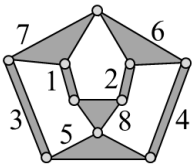
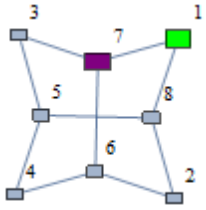
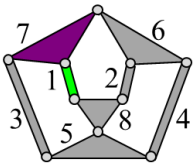
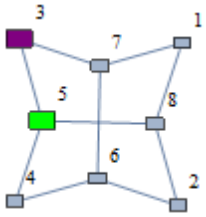
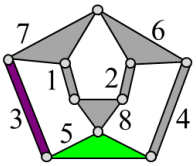
Continued on next page

Table 6.6 – continued from previous page

#	NATML	Adjacency Matrix, Vertex Degree List, Cycle Basis	Adjacency Graph	Topology and Linkage Sketch
68	{8,1,7,5,1}	3 3 2 3 3 3 3 2 3 3 3 3 2 3 2 2 3 3 5 8 1 7 5 5 8 2 6 5 5 8 1 7 3 4 6 5		
69	{8,1,7,5,2}	3 3 2 3 3 3 3 2 2 3 3 3 3 2 3 2 3 3 5 6 2 8 5 5 6 4 3 7 5 5 6 2 8 1 7 5		
	Topology {8,1,8}	$\begin{pmatrix} 0&0&0&1&0&0&0&1\\ 0&0&1&0&0&0&1&0\\ 0&1&0&0&0&1&0&0\\ 1&0&0&0&1&0&0&0\\ 0&0&0&1&0&1&1&0\\ 0&0&1&0&1&0&0&1\\ 0&1&0&0&1&0&0&1\\ 1&0&0&0&0&1&1&0 \end{pmatrix}$		
70	{8,1,8,1,1}	2 2 3 3 3 2 2 2 3 3 3 2 2 2 3 3 2 2 3 3 2 1 4 5 6 8 1 1 4 5 7 8 1 1 4 5 6 3 2 7 8 1		
71	{8,1,8,1,2}	2 3 3 3 2 2 2 3 3 3 2 2 2 3 3 2 2 3 3 2 2 1 8 6 5 4 1 1 8 7 5 4 1 1 8 6 3 2 7 5 4 1		

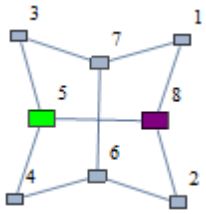
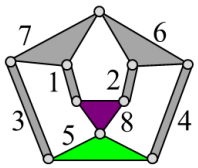
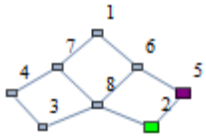
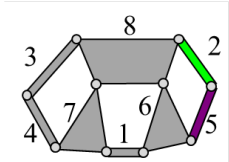
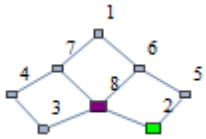
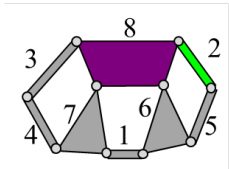
Continued on next page

Table 6.6 – continued from previous page

#	NATML	Adjacency Matrix, Vertex Degree List, Cycle Basis	Adjacency Graph	Topology and Linkage Sketch
72	{8,1,8,2,1} Partitions	3 3 3 3 3 3 3 2 2 3 3 3 3 3 2 2 3 5 6 8 7 5 5 6 3 2 7 5 5 6 8 1 4 5		
73	{8,1,8,2,2}	3 2 2 3 3 3 3 2 2 3 3 3 3 2 2 3 3 2 2 3 3 5 4 1 8 6 5 5 4 1 8 7 5 5 4 1 8 6 3 2 7 5		
	Topology {8,1,9} Double Butterfly	$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$		
74	{8,1,9,1,1}	2 3 2 3 3 2 2 3 3 2 3 2 2 3 3 2 3 3 2 1 7 3 5 8 1 1 7 6 2 8 1 1 7 6 4 5 8 1		
75	{8,1,9,2,1}	3 2 3 2 3 3 3 2 3 3 2 3 3 2 3 3 2 3 3 5 3 7 1 8 5 5 3 7 6 4 5 5 3 7 6 2 8 5		

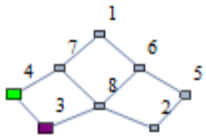
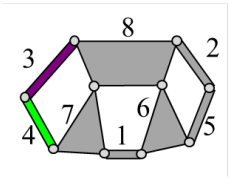
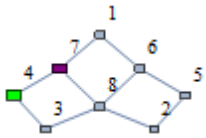
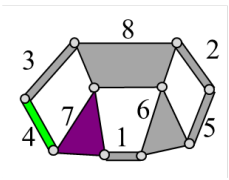
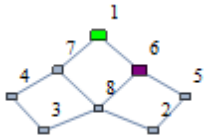
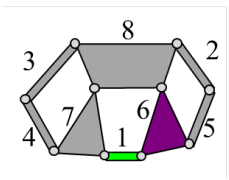
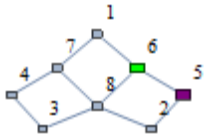
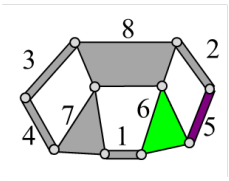
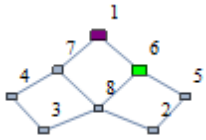
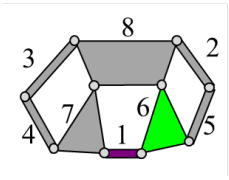
Continued on next page

Table 6.6 – continued from previous page

#	NATML	Adjacency Matrix, Vertex Degree List, Cycle Basis	Adjacency Graph	Topology and Linkage Sketch
76	{8,1,9,2,2}	3 3 2 3 2 3 3 3 2 3 2 3 3 3 2 3 3 2 3 5 8 1 7 3 5 5 8 2 6 4 5 5 8 1 7 6 4 5 Link Assortment 5210 $\begin{pmatrix} 0&0&0&0&0&1&1&0\\ 0&0&0&0&1&0&0&1\\ 0&0&0&1&0&0&0&1\\ 0&0&1&0&0&0&1&0\\ 0&1&0&0&0&1&0&0\\ 1&0&0&0&1&0&0&1\\ 1&0&0&1&0&0&0&1\\ 0&1&1&0&0&1&1&0 \end{pmatrix}$		
77	{8,2,1,1,1}	2 2 3 4 2 2 2 3 2 3 4 2 2 2 3 2 3 2 2 4 2 2 5 6 8 2 2 5 6 1 7 8 2 2 5 6 1 7 4 3 8 2		
78	{8,2,1,1,2}	2 4 3 2 2 2 4 3 2 3 2 2 2 4 2 2 3 2 3 2 2 2 8 6 5 2 2 8 7 1 6 5 2 2 8 3 4 7 1 6 5 2		

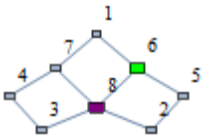
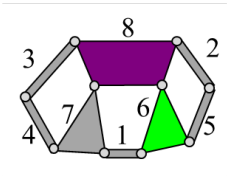
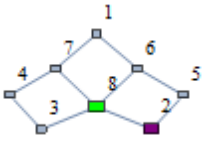
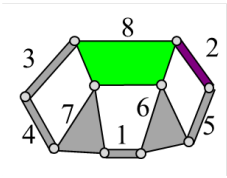
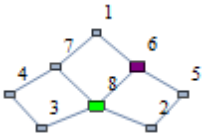
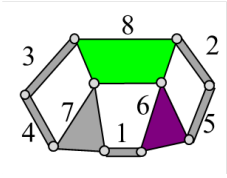
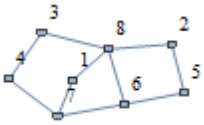
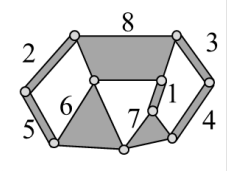
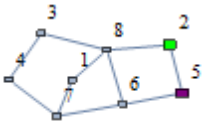
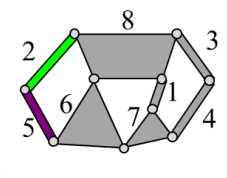
Continued on next page

Table 6.6 – continued from previous page

#	NATML	Adjacency Matrix, Vertex Degree List, Cycle Basis	Adjacency Graph	Topology and Linkage Sketch
79	{8,2,1,2,1}	2 2 4 3 2 2 2 4 3 2 3 2 2 2 4 2 2 3 2 3 2 4 3 8 7 4 4 3 8 6 1 7 4 4 3 8 2 5 6 1 7 4		
80	{8,2,1,2,2}	2 3 4 2 2 2 3 2 3 4 2 2 2 3 2 3 2 2 4 2 2 4 7 8 3 4 4 7 1 6 8 3 4 4 7 1 6 5 2 8 3 4		
81	{8,2,1,3,1} Partitions	2 3 4 3 2 2 3 2 2 4 3 2 2 3 4 2 2 3 2 1 6 8 7 1 1 6 5 2 8 7 1 1 6 8 3 4 7 1		
82	{8,2,1,4,1}	3 2 2 4 3 3 2 2 4 3 2 3 3 2 2 4 2 2 3 2 3 6 5 2 8 6 6 5 2 8 7 1 6 6 5 2 8 3 4 7 1 6		
83	{8,2,1,4,2} Partitions	3 2 3 4 3 3 2 3 2 2 4 3 3 2 3 4 2 2 3 6 1 7 8 6 6 1 7 4 3 8 6 6 1 7 8 2 5 6		

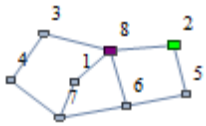
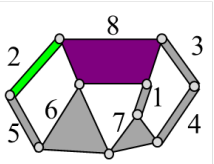
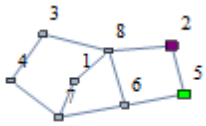
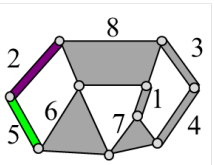
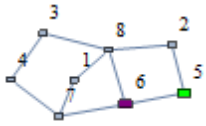
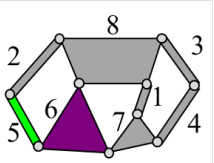
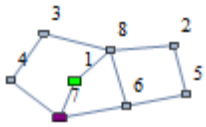
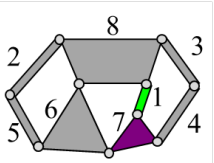
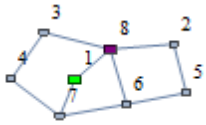
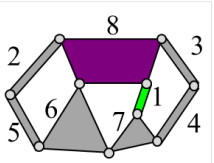
Continued on next page

Table 6.6 – continued from previous page

#	NATML	Adjacency Matrix, Vertex Degree List, Cycle Basis	Adjacency Graph	Topology and Linkage Sketch
84	{8,2,1,4,3} Partitions	3 4 2 2 3 3 4 3 2 3 3 4 2 2 3 2 3 6 8 2 5 6 6 8 7 1 6 6 8 3 4 7 1 6		
85	{8,2,1,5,1}	4 2 2 3 4 4 2 2 3 2 3 4 4 2 2 3 2 3 2 2 4 8 2 5 6 8 8 2 5 6 1 7 8 8 2 5 6 1 7 4 3 8		
86	{8,2,1,5,2} Partitions	4 3 2 2 4 4 3 2 3 4 4 3 2 3 2 2 4 8 6 5 2 8 8 6 1 7 8 8 6 1 7 4 3 8		
	Topology {8,2,2}	$\begin{pmatrix} 0&0&0&0&0&0&1&1\\ 0&0&0&0&1&0&0&1\\ 0&0&0&1&0&0&0&1\\ 0&0&1&0&0&0&1&0\\ 0&1&0&0&0&1&0&0\\ 0&0&0&0&1&0&1&1\\ 1&0&0&1&0&1&0&0\\ 1&1&1&0&0&1&0&0 \end{pmatrix}$		
87	{8,2,2,1,1}	2 2 3 4 2 2 2 3 3 2 4 2 2 2 3 3 2 2 4 2 2 5 6 8 2 2 5 6 7 1 8 2 2 5 6 7 4 3 8 2		

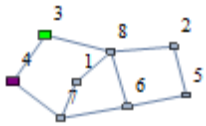
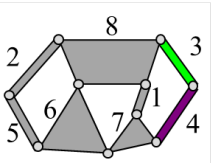
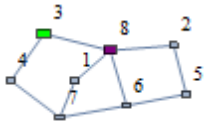
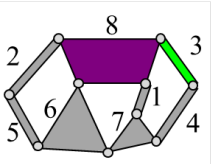
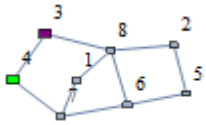
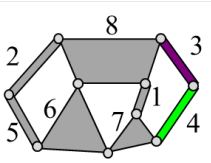
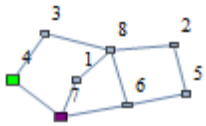
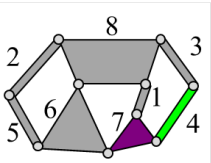
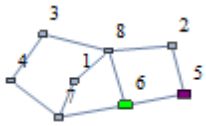
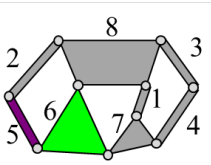
Continued on next page

Table 6.6 – continued from previous page

#	NATML	Adjacency Matrix, Vertex Degree List, Cycle Basis	Adjacency Graph	Topology and Linkage Sketch
88	{8,2,2,1,2}	2 4 3 2 2 2 4 2 3 3 2 2 2 4 2 2 3 3 2 2 2 8 6 5 2 2 8 1 7 6 5 2 2 8 3 4 7 6 5 2		
89	{8,2,2,2,1}	2 2 4 3 2 2 2 4 2 3 3 2 2 2 4 2 2 3 3 2 5 2 8 6 5 5 2 8 1 7 6 5 5 2 8 3 4 7 6 5		
90	{8,2,2,2,2}	2 3 4 2 2 2 3 3 2 4 2 2 2 3 3 2 2 4 2 2 5 6 8 2 5 5 6 7 1 8 2 5 5 6 7 4 3 8 2 5		
91	{8,2,2,3,1} Partitions	2 3 3 4 2 2 3 2 2 4 2 2 3 3 2 2 4 2 1 7 6 8 1 1 7 4 3 8 1 1 7 6 5 2 8 1		
92	{8,2,2,3,2} Partitions	2 4 3 3 2 2 4 2 2 3 2 2 4 2 2 3 3 2 1 8 6 7 1 1 8 3 4 7 1 1 8 2 5 6 7 1		

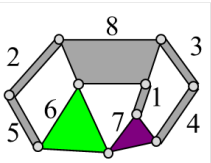
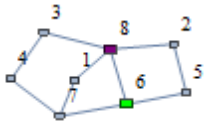
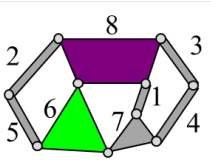
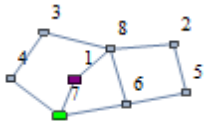
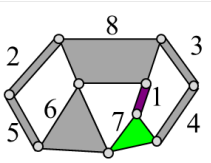
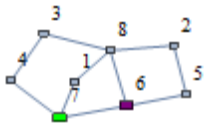
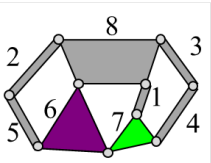
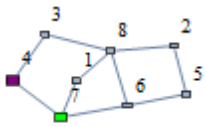
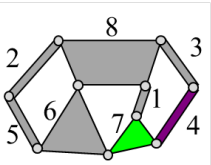
Continued on next page

Table 6.6 – continued from previous page

#	NATML	Adjacency Matrix, Vertex Degree List, Cycle Basis	Adjacency Graph	Topology and Linkage Sketch
93	{8,2,2,4,1}	2 2 3 2 4 2 2 2 3 3 4 2 2 2 3 3 2 2 4 2 3 4 7 1 8 3 3 4 7 6 8 3 3 4 7 6 5 2 8 3		
94	{8,2,2,4,2}	2 4 2 3 2 2 2 4 3 3 2 2 2 4 2 2 3 3 2 2 3 8 1 7 4 3 3 8 6 7 4 3 3 8 2 5 6 7 4 3		
95	{8,2,2,5,1}	2 2 4 2 3 2 2 2 4 3 3 2 2 2 4 2 2 3 3 2 4 3 8 1 7 4 4 3 8 6 7 4 4 3 8 2 5 6 7 4		
96	{8,2,2,5,2}	2 3 2 4 2 2 2 3 3 4 2 2 2 3 3 2 2 4 2 2 4 7 1 8 3 4 4 7 6 8 3 4 4 7 6 5 2 8 3 4		
97	{8,2,2,6,1}	3 2 2 4 3 3 2 2 4 2 3 3 3 2 2 4 2 2 3 3 6 5 2 8 6 6 5 2 8 1 7 6 6 5 2 8 3 4 7 6		

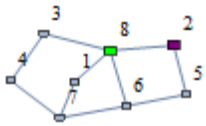
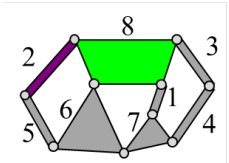
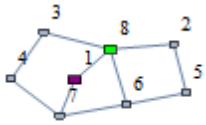
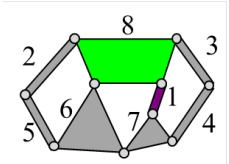
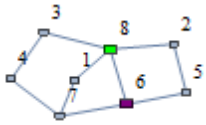
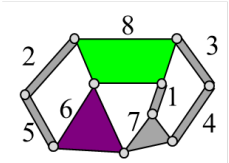
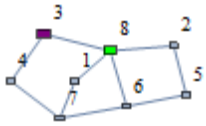
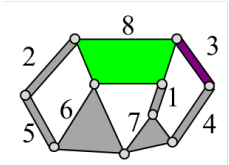
Continued on next page

Table 6.6 – continued from previous page

#	NATML	Adjacency Matrix, Vertex Degree List, Cycle Basis	Adjacency Graph	Topology and Linkage Sketch
98	{8,2,2,6,2} Partitions	3 3 2 4 3 3 3 2 2 4 3 3 3 2 4 2 2 3 6 7 1 8 6 6 7 4 3 8 6 6 7 1 8 2 5 6		
99	{8,2,2,6,3} Partitions	3 4 2 2 3 3 4 2 3 3 3 4 2 2 3 3 6 8 2 5 6 6 8 1 7 6 6 8 3 4 7 6		
100	{8,2,2,7,1} Partitions	3 2 4 3 3 3 2 4 2 2 3 3 2 4 2 2 3 3 7 1 8 6 7 7 1 8 3 4 7 7 1 8 2 5 6 7		
101	{8,2,2,7,2} Partitions	3 3 4 2 3 3 3 4 2 2 3 3 3 2 2 4 2 3 7 6 8 1 7 7 6 8 3 4 7 7 6 5 2 8 1 7		
102	{8,2,2,7,3}	3 2 2 4 2 3 3 2 2 4 3 3 3 2 2 4 2 2 3 3 7 4 3 8 1 7 7 4 3 8 6 7 7 4 3 8 2 5 6 7		

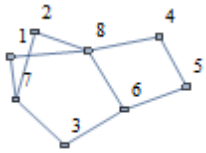
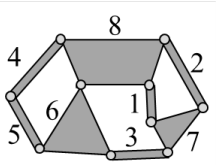
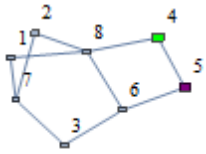
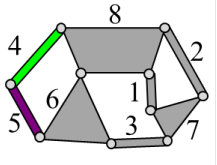
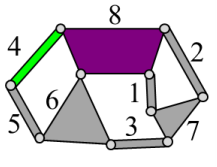
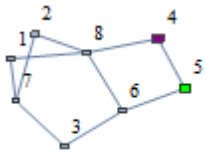
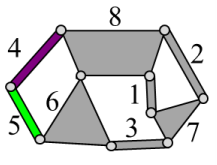
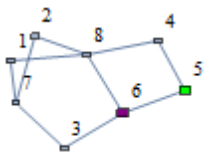
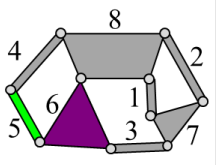
Continued on next page

Table 6.6 – continued from previous page

#	NATML	Adjacency Matrix, Vertex Degree List, Cycle Basis	Adjacency Graph	Topology and Linkage Sketch
103	{8,2,2,8,1}	4 2 2 3 4 4 2 2 3 3 2 4 4 2 2 3 3 2 2 4 8 2 5 6 8 8 2 5 6 7 1 8 8 2 5 6 7 4 3 8		
104	{8,2,2,8,2} Partitions	4 2 3 3 4 4 2 3 2 2 4 4 2 3 3 2 2 4 8 1 7 6 8 8 1 7 4 3 8 8 1 7 6 5 2 8		
105	{8,2,2,8,3} Partitions	4 3 2 2 4 4 3 3 2 4 4 3 3 2 2 4 8 6 5 2 8 8 6 7 1 8 8 6 7 4 3 8		
106	{8,2,2,8,4}	4 2 2 3 2 4 4 2 2 3 3 4 4 2 2 3 3 2 2 4 8 3 4 7 1 8 8 3 4 7 6 8 8 3 4 7 6 5 2 8		

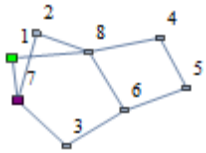
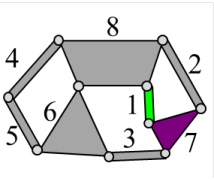
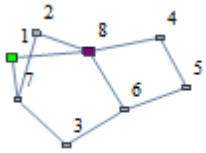
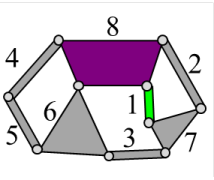
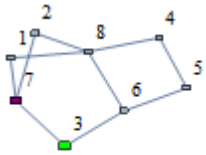
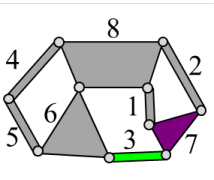
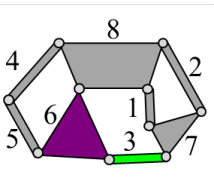
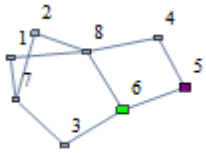
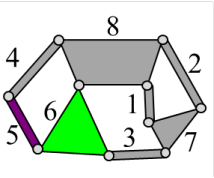
Continued on next page

Table 6.6 – continued from previous page

#	NATML	Adjacency Matrix, Vertex Degree List, Cycle Basis	Adjacency Graph	Topology and Linkage Sketch
	Topology {8,2,3}	$\begin{pmatrix} 0&0&0&0&0&0&1&1\\ 0&0&0&0&0&0&1&1\\ 0&0&0&0&0&1&1&0\\ 0&0&0&0&1&0&0&1\\ 0&0&0&1&0&1&0&0\\ 0&0&1&0&1&0&0&1\\ 1&1&1&0&0&0&0&0\\ 1&1&0&1&0&1&0&0 \end{pmatrix}$		
107	{8,2,3,1,1}	2 2 3 4 2 2 2 3 2 3 2 4 2 2 2 3 2 3 2 4 2 4 5 6 8 4 4 5 6 3 7 1 8 4 4 5 6 3 7 2 8 4		
108	{8,2,3,1,2}	2 4 3 2 2 2 4 2 3 2 3 2 2 2 4 2 3 2 3 2 2 4 8 6 5 4 4 8 1 7 3 6 5 4 4 8 2 7 3 6 5 4		
109	{8,2,3,2,1}	2 2 4 3 2 2 2 4 2 3 2 3 2 2 2 4 2 3 2 3 2 5 4 8 6 5 5 4 8 1 7 3 6 5 5 4 8 2 7 3 6 5		
110	{8,2,3,2,2}	2 3 4 2 2 2 3 2 3 2 4 2 2 2 3 2 3 2 4 2 2 5 6 8 4 5 5 6 3 7 1 8 4 5 5 6 3 7 2 8 4 5		

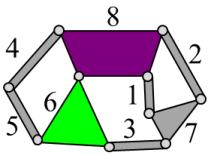
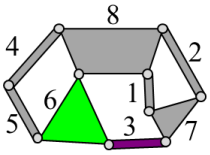
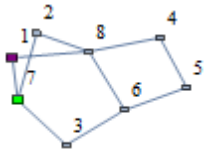
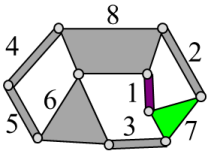
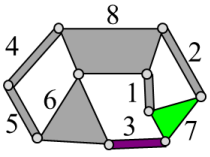
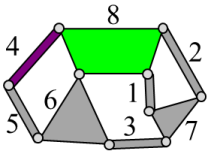
Continued on next page

Table 6.6 – continued from previous page

#	NATML	Adjacency Matrix, Vertex Degree List, Cycle Basis	Adjacency Graph	Topology and Linkage Sketch
111	{8,2,3,3,1}	2 3 2 4 2 2 3 2 3 4 2 2 3 2 3 2 2 4 2 1 7 2 8 1 1 7 3 6 8 1 1 7 3 6 5 4 8 1		
112	{8,2,3,3,2}	2 4 2 3 2 2 4 3 2 3 2 2 4 2 2 3 2 3 2 1 8 2 7 1 1 8 6 3 7 1 1 8 4 5 6 3 7 1		
113	{8,2,3,4,1}	2 3 2 4 3 2 2 3 2 4 3 2 2 3 2 4 2 2 3 2 3 7 1 8 6 3 3 7 2 8 6 3 3 7 1 8 4 5 6 3		
114	{8,2,3,4,2}	2 3 4 2 3 2 2 3 4 2 3 2 2 3 2 2 4 2 3 2 3 6 8 1 7 3 3 6 8 2 7 3 3 6 5 4 8 1 7 3		
115	{8,2,3,5,1}	3 2 2 4 3 3 2 2 4 2 3 2 3 3 2 2 4 2 3 2 3 6 5 4 8 6 6 5 4 8 1 7 3 6 6 5 4 8 2 7 3 6		

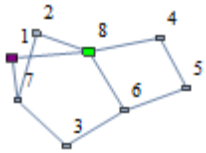
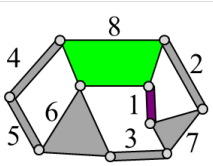
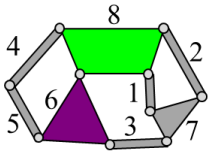
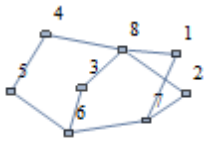
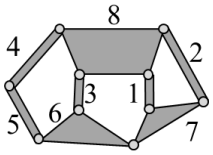
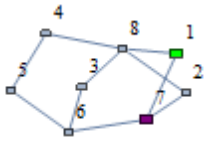
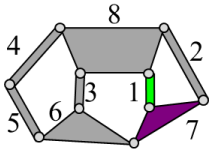
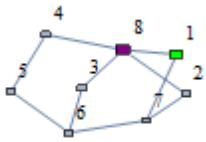
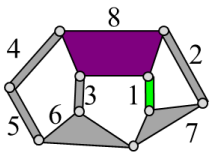
Continued on next page

Table 6.6 – continued from previous page

#	NATML	Adjacency Matrix, Vertex Degree List, Cycle Basis	Adjacency Graph	Topology and Linkage Sketch
116	{8,2,3,5,2} Partitions	3 4 2 2 3 3 4 2 3 2 3 3 4 2 3 2 3 6 8 4 5 6 6 8 1 7 3 6 6 8 2 7 3 6		
117	{8,2,3,5,3}	3 2 3 2 4 3 3 2 3 2 4 3 3 2 3 2 4 2 2 3 6 3 7 1 8 6 6 3 7 2 8 6 6 3 7 1 8 4 5 6		
118	{8,2,3,6,1}	3 2 4 2 3 3 2 4 3 2 3 3 2 4 2 2 3 2 3 7 1 8 2 7 7 1 8 6 3 7 7 1 8 4 5 6 3 7		
119	{8,2,3,6,2}	3 2 3 4 2 3 3 2 3 4 2 3 3 2 3 2 2 4 2 3 7 3 6 8 1 7 7 3 6 8 2 7 7 3 6 5 4 8 1 7		
120	{8,2,3,7,1}	4 2 2 3 4 4 2 2 3 2 3 2 4 4 2 2 3 2 3 2 4 8 4 5 6 8 8 4 5 6 3 7 1 8 8 4 5 6 3 7 2 8		

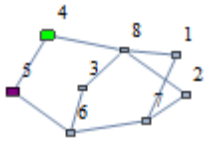
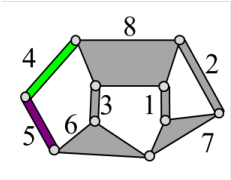
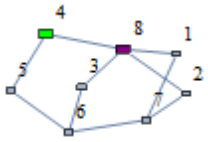
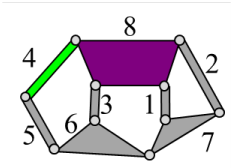
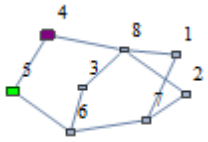
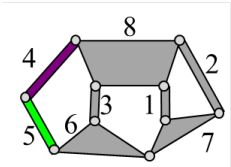
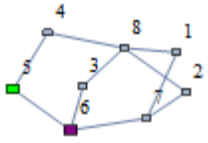
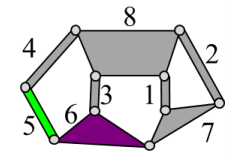
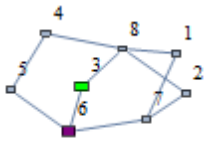
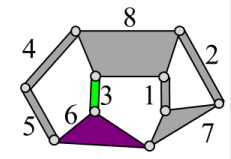
Continued on next page

Table 6.6 – continued from previous page

#	NATML	Adjacency Matrix, Vertex Degree List, Cycle Basis	Adjacency Graph	Topology and Linkage Sketch
121	{8,2,3,7,2}	4 2 3 2 4 4 2 3 2 3 4 4 2 3 2 3 2 2 4 8 1 7 2 8 8 1 7 3 6 8 8 1 7 3 6 5 4 8		
122	{8,2,3,7,3} Partitions	4 3 2 2 4 4 3 2 3 2 4 4 3 2 3 2 4 8 6 5 4 8 8 6 3 7 1 8 8 6 3 7 2 8		
	Topology {8,2,4}	$\begin{pmatrix} 0&0&0&0&0&0&1&1\\ 0&0&0&0&0&0&1&1\\ 0&0&0&0&0&1&0&1\\ 0&0&0&0&1&0&0&1\\ 0&0&0&1&0&1&0&0\\ 0&0&1&0&1&0&1&0\\ 1&1&0&0&0&1&0&0\\ 1&1&1&1&0&0&0&0 \end{pmatrix}$		
123	{8,2,4,1,1}	2 3 2 4 2 2 3 3 2 4 2 2 3 3 2 2 4 2 1 7 2 8 1 1 7 6 3 8 1 1 7 6 5 4 8 1		
124	{8,2,4,1,2}	2 4 2 3 2 2 4 2 3 3 2 2 4 2 2 3 3 2 1 8 2 7 1 1 8 3 6 7 1 1 8 4 5 6 7 1		

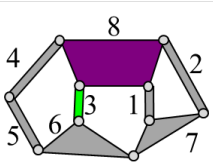
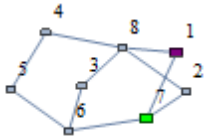
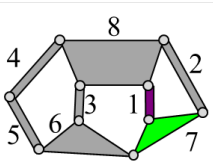
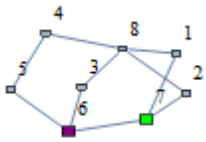
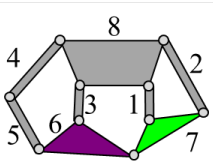
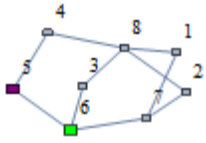
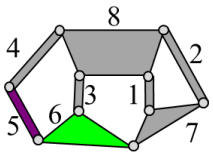
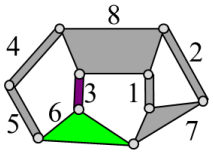
Continued on next page

Table 6.6 – continued from previous page

#	NATML	Adjacency Matrix, Vertex Degree List, Cycle Basis	Adjacency Graph	Topology and Linkage Sketch
125	{8,2,4,2,1}	2 2 3 2 4 2 2 2 3 3 2 4 2 2 2 3 3 2 4 2 4 5 6 3 8 4 4 5 6 7 1 8 4 4 5 6 7 2 8 4		
126	{8,2,4,2,2}	2 4 2 3 2 2 2 4 2 3 3 2 2 2 4 2 3 3 2 2 4 8 3 6 5 4 4 8 1 7 6 5 4 4 8 2 7 6 5 4		
127	{8,2,4,3,1}	2 2 4 2 3 2 2 2 4 2 3 3 2 2 2 4 2 3 3 2 5 4 8 3 6 5 5 4 8 1 7 6 5 5 4 8 2 7 6 5		
128	{8,2,4,3,2}	2 3 2 4 2 2 2 3 3 2 4 2 2 2 3 3 2 4 2 2 5 6 3 8 4 5 5 6 7 1 8 4 5 5 6 7 2 8 4 5		
129	{8,2,4,4,1}	2 3 2 2 4 2 2 3 3 2 4 2 2 3 3 2 4 2 3 6 5 4 8 3 3 6 7 1 8 3 3 6 7 2 8 3		

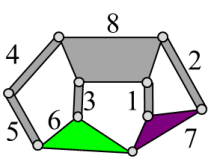
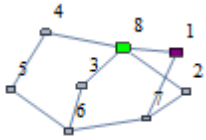
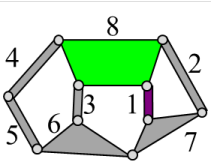
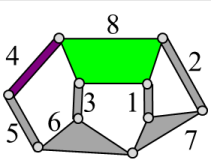
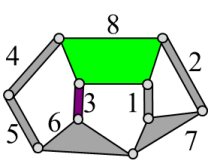
Continued on next page

Table 6.6 – continued from previous page

#	NATML	Adjacency Matrix, Vertex Degree List, Cycle Basis	Adjacency Graph	Topology and Linkage Sketch
130	{8,2,4,4,2}	2 4 2 2 3 2 2 4 2 3 3 2 2 4 2 3 3 2 3 8 4 5 6 3 3 8 1 7 6 3 3 8 2 7 6 3		
131	{8,2,4,5,1}	3 2 4 2 3 3 2 4 2 3 3 3 2 4 2 2 3 3 7 1 8 2 7 7 1 8 3 6 7 7 1 8 4 5 6 7		
132	{8,2,4,5,2}	3 3 2 4 2 3 3 3 2 4 2 3 3 3 2 2 4 2 3 7 6 3 8 1 7 7 6 3 8 2 7 7 6 5 4 8 1 7		
133	{8,2,4,6,1}	3 2 2 4 2 3 3 2 2 4 2 3 3 3 2 2 4 2 3 3 6 5 4 8 3 6 6 5 4 8 1 7 6 6 6 5 4 8 2 7 6		
134	{8,2,4,6,2}	3 2 4 2 2 3 3 2 4 2 3 3 3 2 4 2 3 3 6 3 8 4 5 6 6 3 8 1 7 6 6 3 8 2 7 6		

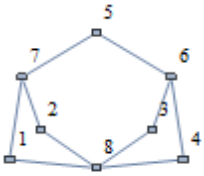
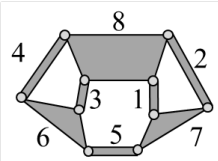
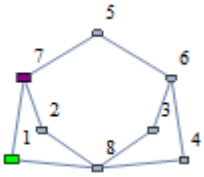
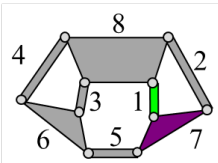
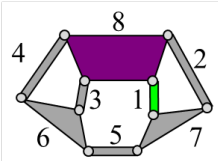
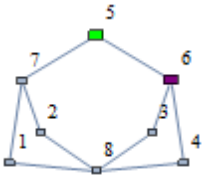
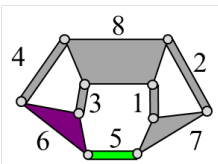
Continued on next page

Table 6.6 – continued from previous page

#	NATML	Adjacency Matrix, Vertex Degree List, Cycle Basis	Adjacency Graph	Topology and Linkage Sketch
135	{8,2,4,6,3}	3 3 2 4 2 3 3 3 2 4 2 3 3 3 2 4 2 2 3 6 7 1 8 3 6 6 7 2 8 3 6 6 7 1 8 4 5 6		
136	{8,2,4,7,1}	4 2 3 2 4 4 2 3 3 2 4 4 2 3 3 2 2 4 8 1 7 2 8 8 1 7 6 3 8 8 1 7 6 5 4 8		
137	{8,2,4,7,2}	4 2 2 3 2 4 4 2 2 3 3 2 4 4 2 2 3 3 2 4 8 4 5 6 3 8 8 4 5 6 7 1 8 8 4 5 6 7 2 8		
138	{8,2,4,7,3}	4 2 3 2 2 4 4 2 3 3 2 4 4 2 3 3 2 4 8 3 6 5 4 8 8 3 6 7 1 8 8 3 6 7 2 8		

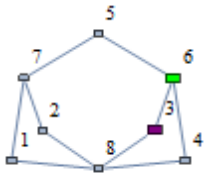
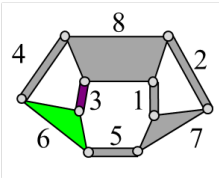
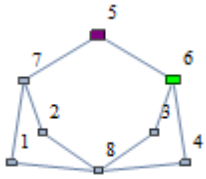
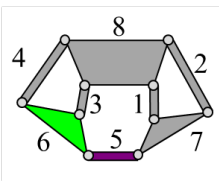
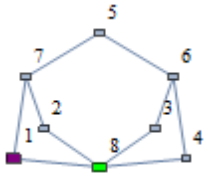
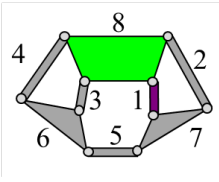
Continued on next page

Table 6.6 – continued from previous page

#	NATML	Adjacency Matrix, Vertex Degree List, Cycle Basis	Adjacency Graph	Topology and Linkage Sketch
	Topology {8,2,5}	$\begin{pmatrix} 0&0&0&0&0&0&1&1\\ 0&0&0&0&0&0&1&1\\ 0&0&0&0&0&1&0&1\\ 0&0&0&0&0&1&0&1\\ 0&0&0&0&0&1&1&0\\ 0&0&1&1&1&0&0&0\\ 1&1&0&0&1&0&0&0\\ 1&1&1&1&0&0&0&0 \end{pmatrix}$		
139	{8,2,5,1,1}	2 3 2 4 2 2 3 2 3 2 4 2 2 3 2 3 2 4 2 1 7 2 8 1 1 7 5 6 3 8 1 1 7 5 6 4 8 1		
140	{8,2,5,1,2}	2 4 2 3 2 2 4 2 3 2 3 2 2 4 2 3 2 3 2 1 8 2 7 1 1 8 3 6 5 7 1 1 8 4 6 5 7 1		
141	{8,2,5,2,1}	2 3 2 4 2 3 2 2 3 2 4 2 3 2 2 3 2 4 2 3 2 5 6 3 8 1 7 5 5 6 3 8 2 7 5 5 6 4 8 1 7 5		

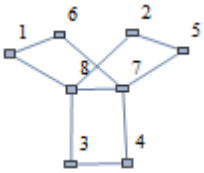
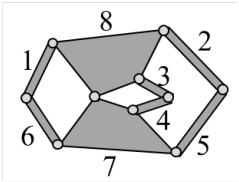
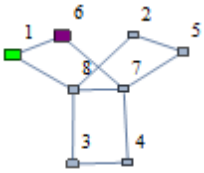
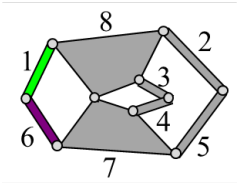
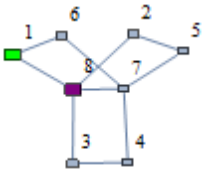
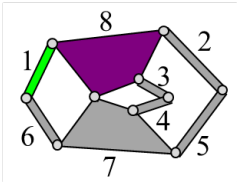
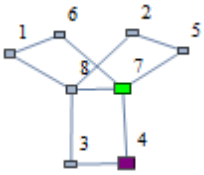
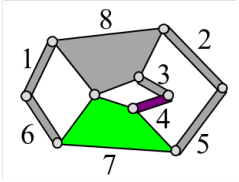
Continued on next page

Table 6.6 – continued from previous page

#	NATML	Adjacency Matrix, Vertex Degree List, Cycle Basis	Adjacency Graph	Topology and Linkage Sketch
142	{8,2,5,3,1}	3 2 4 2 3 3 2 4 2 3 2 3 3 2 4 2 3 2 3 6 3 8 4 6 6 3 8 1 7 5 6 6 3 8 2 7 5 6		
143	{8,2,5,3,2}	3 2 3 2 4 2 3 3 2 3 2 4 2 3 3 2 3 2 4 2 3 6 5 7 1 8 3 6 6 5 7 1 8 4 6 6 5 7 2 8 3 6		
144	{8,2,5,4,1}	4 2 3 2 4 4 2 3 2 3 2 4 4 2 3 2 3 2 4 8 1 7 2 8 8 1 7 5 6 3 8 8 1 7 5 6 4 8		

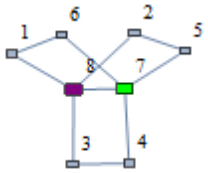
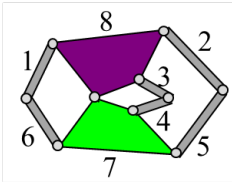
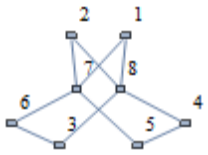
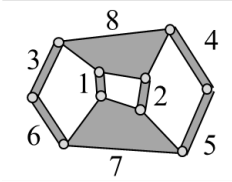
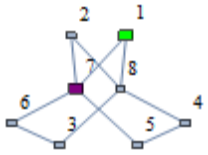
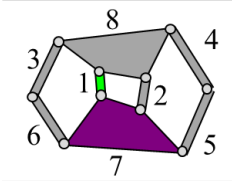
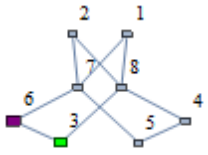
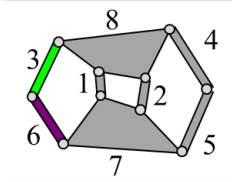
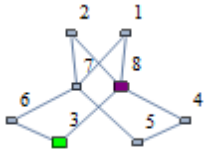
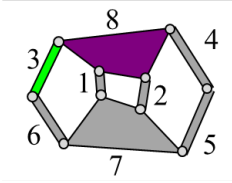
Continued on next page

Table 6.6 – continued from previous page

#	NATML	Adjacency Matrix, Vertex Degree List, Cycle Basis	Adjacency Graph	Topology and Linkage Sketch
		Link Assortment 6020		
	Topology {8,3,1}	$\begin{pmatrix} 0&0&0&0&0&1&0&1 \\ 0&0&0&0&1&0&0&1 \\ 0&0&0&1&0&0&0&1 \\ 0&0&1&0&0&0&1&0 \\ 0&1&0&0&0&0&1&0 \\ 1&0&0&0&0&0&1&0 \\ 0&0&0&1&1&1&0&1 \\ 1&1&1&0&0&0&1&0 \end{pmatrix}$		
145	{8,3,1,1,1} Partitions	2 2 4 4 2 2 2 4 2 2 4 2 2 2 4 2 2 4 2 1 6 7 8 1 1 6 7 4 3 8 1 1 6 7 5 2 8 1		
146	{8,3,1,1,2} Partitions	2 4 4 2 2 2 4 2 2 4 2 2 2 4 2 2 4 2 2 1 8 7 6 1 1 8 2 5 7 6 1 1 8 3 4 7 6 1		
147	{8,3,1,2,1} Partitions	4 2 2 4 4 4 2 2 4 2 2 4 4 2 2 4 2 2 4 7 4 3 8 7 7 4 3 8 1 6 7 7 4 3 8 2 5 7		

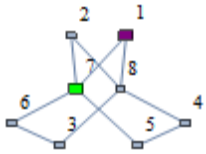
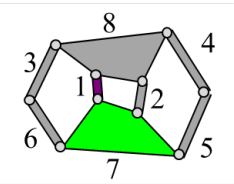
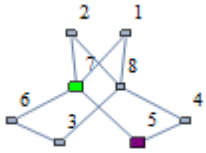
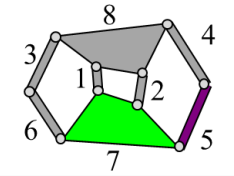
Continued on next page

Table 6.6 – continued from previous page

#	NATML	Adjacency Matrix, Vertex Degree List, Cycle Basis	Adjacency Graph	Topology and Linkage Sketch
148	{8,3,1,2,2} Partitions	4 4 2 2 4 4 4 2 2 4 4 4 2 2 4 7 8 1 6 7 7 8 2 5 7 7 8 3 4 7		
	Topology {8,3,2}	$\begin{pmatrix} 0&0&0&0&0&0&1&1\\ 0&0&0&0&0&0&1&1\\ 0&0&0&0&0&1&0&1\\ 0&0&0&0&1&0&0&1\\ 0&0&0&1&0&0&1&0\\ 0&0&1&0&0&0&1&0\\ 1&1&0&0&1&1&0&0\\ 1&1&1&1&0&0&0&0 \end{pmatrix}$		
149	{8,3,2,1,1} Partitions	2 4 2 4 2 2 4 2 2 4 2 2 4 2 2 4 2 1 7 2 8 1 1 7 5 4 8 1 1 7 6 3 8 1		
150	{8,3,2,2,1}	2 2 4 2 4 2 2 2 4 2 4 2 2 2 4 2 2 4 2 3 6 7 1 8 3 3 6 7 2 8 3 3 6 7 5 4 8 3		
151	{8,3,2,2,2}	2 4 2 4 2 2 2 4 2 4 2 2 2 4 2 2 4 2 2 3 8 1 7 6 3 3 8 2 7 6 3 3 8 4 5 7 6 3		

Continued on next page

Table 6.6 – continued from previous page

#	NATML	Adjacency Matrix, Vertex Degree List, Cycle Basis	Adjacency Graph	Topology and Linkage Sketch
152	{8,3,2,3,1} Partitions	4 2 4 2 4 4 2 4 2 2 4 4 2 4 2 2 4 7 1 8 2 7 7 1 8 3 6 7 7 1 8 4 5 7		
153	{8,3,2,3,2}	4 2 2 4 2 4 4 2 2 4 2 4 4 2 2 4 2 2 4 7 5 4 8 1 7 7 5 4 8 2 7 7 7 5 4 8 3 6 7		

6.3 Linkage Classification Conclusion

This section provides the complete classification of four, six and eight-bar families. The linkages within the families are uniquely identified by a five number index $\{n, a, t, m, l\}$.

The identification scheme also uniquely represents all five levels of categorization that are commonly used: family (number of bars), link assortment (number of links by type), topology (adjacency matrix), mechanism (selected ground link), and linkage (selected driving link). A planar one-DoF linkage with a ground connected input has a specific NATML classification.

This method should be extendable to linkages with higher bar counts but requires that all unique topologies for a particular N-bar family are known and used as a starting condition.

Chapter 7

Enumeration Results

This section summarizes the enumeration results of the research.

7.1 Enumeration of Unique Mechanisms and Linkages

The quantity of unique six-bar linkages derived by this process matches the expected value published by McCarthy and Soh [23]. Grouped by the two unique six-bar topologies, Table 7.1 shows the breakdown of the six-bar mechanisms and linkages. The process identified the five unique six-bar mechanisms and the nine unique six-bar linkages with a ground connected input, matching the known Watt and Stephenson families. Among those nine linkages the algorithm also properly identified the existence of the Watt IIb, a linkage not shown by Soh and McCarthy because it is not suitable for the linkage synthesis process they used.

Table 7.1: Distinct six-bar mechanisms and linkages, total by topology.

Topology	Name	Mechanisms	Linkages
{6,1,1}	Watt	2	4
{6,1,2}	Stephenson	3	5
Total		5	9

Table 7.2: Distinct eight-bar mechanisms and linkages, total by topology.

Topology	Name	Mechanisms	Linkages
{8,1,1}		2	5
{8,1,2}		4	8
{8,1,3}		2	3
{8,1,4}		8	20
{8,1,5}		4	10
{8,1,6}		6	12
{8,1,7}		5	11
{8,1,8}		2	4
{8,1,9}	Double Butterfly	2	3
{8,2,1}		5	10
{8,2,2}		8	20
{8,2,3}		7	16
{8,2,4}		7	16
{8,2,5}		4	6
{8,3,1}		2	4
{8,3,2}		3	5
Total		71	153

The quantity of unique mechanisms for the eight-bar family derived by this process matches the expected quantity of 71 published by Tuttle [49]. Grouped by the 16 unique eight-bar topologies Table 7.2 provides the breakdown of the 71 unique eight-bar mechanisms and the new result showing 153 unique linkages with a ground-connected input. Each of these linkages has been assigned a unique classification by the NATML convention in chapter 6.

7.2 Identification of Linkages that Partition

The algorithm found the Watt IIb linkage and identified it as the one six-bar linkage that partitions. This linkage is identified by the algorithm as a partitioning linkage since there does not exist an acceptable selection for the Dixon determinant eigenvalue.

In the eight-bar family the algorithm found 24 linkages that do not have an acceptable selection for the eigenvalue and cannot be solved as a whole linkage using the Dixon determinant.

Inspection of these 24 linkages show that the ground and input links are driving two independent one-DoF sub-linkages. For the {8,3,1} topology, no matter which ground link and input link is selected, none of the linkages can be solved with the Dixon determinant as a whole linkage because they all partition. One of the linkages, number {8,3,1,2,2}, has a partition of three independent four-bars with a common ground and input link.

7.3 Enumeration Results Conclusion

The research enumerated the unique mechanisms and linkages obtained from the 16 eight-bar topologies and the results correlate with published results and extend the published results to show 153 unique linkages with a ground connected input.

The research also enumerated the linkages that partition into simpler linkages. There is one six-bar linkage that partitions and there are 24 eight-bar linkages that partition.

Chapter 8

Conclusion

This dissertation presents research into the automation of the configuration analysis of eight-bar linkages based on the known 16 linkage topologies. The contribution of this research is an automated analysis of eight-bar linkages that applies to four-bar and six-bar linkages, and has been useful in the analysis of 10-bar linkages. The automation process can formulate the loop equations and Jacobian conditions for all 153 cases, as well as for all four-bar and six-bar cases.

Automated configuration analysis for eight-bar linkages provides an important tool for evaluating the range of movement of linkages obtained in mechanism synthesis algorithms, allowing identification of linkages that achieve a required task.

8.1 Contribution Summary

This research established a method to automatically derive loop equations for any linkage in the four-bar, six-bar, and eight-bar one-DoF linkages with revolute joints. Loop equations

for four 10-bar linkages have also been automatically produced, including linkages with non-planar graphs and one with a quintenary link.

A repeatable method for determining a specific cycle basis through a common edge has been established. This enabled the construction of a linkage specific loop vertex degree list that has been used to uniquely identify every four-bar, six-bar, and eight-bar one-DoF linkage with revolute joints.

The unique identification of the linkages has been leveraged to sort every linkage and establish a linkage classification convention where every linkage is uniquely identified by a specific five number index, NATML. This convention represents a linkage by family (number of bars), link assortment (number of links by type), topology (adjacency matrix), mechanism (selected ground link), and linkage (selected driving link). The classification is applicable to four-bar, six-bar, and eight-bar linkages.

The linkage topologies do not distinguish a ground link or an input link, and this research shows that the selection of a ground link yields five unique six-bar mechanisms and 71 unique eight-bar mechanisms, in agreement with published results. Similarly the selection of an input link connected to the ground link yields nine unique six-bar linkages, in agreement with published results, and 153 unique eight-bar linkages with distinct sets of loop equations.

The Dixon determinant is automatically derived from the loop equations and the configuration analysis is accomplished by solving the Dixon determinant for a given input angle to determine all possible assembly configurations. Using the eigenvalue, a method for identifying linkages that partition into simpler linkages has been established.

The interface from a synthesis algorithm to the automated analysis algorithm has been defined through the use of an enhanced adjacency matrix. This format enables a compact description of the linkage physical geometry and the interconnections between the links.

The interface from the automated analysis algorithm to the synthesis algorithm has been defined through the use of the FTLA format which describes the joints, link dimensions and angles of every link feature. The FTLA provides an unambiguous description of the features used in the automated analysis and provides a compact description for drawing the linkage.

8.2 Future Research

The automation approach is general therefore it should form a basis for the automation of more complicated linkages. Extensions of the work are expected to include planar multi-degree of freedom linkages, the entire family of 10-bar linkages, prismatic joints, and an extension to spherical linkages.

The classification, NATML, is applicable to four-bar, six-bar, and eight-bar linkages but is also general such that it should extend without modification to the 10-bar and higher planar linkages with revolute joints. The classification scheme may be able to be enhanced in the future to include unambiguous identification of linkages with prismatic joints, multiple degrees of freedom, and spherical linkages.

When extending the automation to 10-bar and higher linkages some of the graphs cannot be embedded in a plane. Since this automation methodology depends on finding independent cycles, regardless of how the graph itself must be depicted, it is believed that the methodology will work for linkages with non-planar graphs in general. Two example 10-bar linkages with non-planar graphs have been successfully demonstrated with the present algorithm, however, more research is needed to ensure that this algorithm will sufficiently develop the linkage loop equations for non-planar graphs.

Also when extending the work to 10-bar and higher linkages some of the linkages contain quintenary links. The loops in the quintenary links will have more complicated combinations

of fixed angles. Since this automation methodology builds the fixed angles upon a reference line whose angle is already defined, the methodology is believed to extend to the quintenary and higher links. However, more research is needed to ensure that it will always work for these links.

An alternate method to identify the loops in a graph called the Surballe algorithm applies a weighting of edges to identify unique paths. Future research using the Surballe algorithm may provide a method for identifying the smallest possible cycle basis that is computationally more efficient.

Bibliography

[1] N. P. Belfiore. A brief note on the concept of planarity for kinematic chains. *Mechanism and Machine Theory*, 2000.

[2] C. Bona, C. Galletti, and A. Lucifredi. Computer-aided automatic design. *Mechanism and Machine Theory*, 1973.

[3] T. R. Chase and J. A. Mirth. Circuits and branches of single-degree-of-freedom planar linkages. *ASME Journal of Mechanical Design*, 1993.

[4] T. H. Davies. An extension of Manolescu's classification of planar kinematic chains and mechanisms of mobility M$\geq$ 1, using graph theory. *Journal of Mechanisms*, 1968.

[5] T. H. Davies and F. E. Crossley. Structural analysis of plane linkages by Franke's condensed notation. *Journal of Mechanisms*, 1966.

[6] A. K. Dhingra, A. N. Almadi, and D. Kohli. A Gröbner-Sylvester hybrid method for closed-form displacement analysis of mechanisms. *ASME Journal of Mechanical Design*, 2000.

[7] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1959.

[8] H. Ding, F. Hou, A. Kecskeméthy, and Z. Huang. Synthesis of the whole family of planar 1-DOF kinematic chains and creation of their atlas database. *Mechanism and Machine Theory*, 2012.

[9] H. Ding and Z. Huang. The establishment of the canonical perimeter topological graph of kinematic chains and isomorphism identification. *ASME Journal of Mechanical Design*, 2007.

[10] H. Ding and Z. Huang. Isomorphism identification of graphs: especially for the graphs of kinematic chains. *Mechanism and Machine Theory*, 2009.

[11] H. Ding, P. Yang, W. Huang, and A. Kecskeméthy. Automatic structural synthesis of planar multiple joint kinematic chains. *ASME Journal of Mechanical Design*, 2013.

[12] A. Dixon. The eliminant of three quantics in two independent variables. *Proceedings of the London Mathematical Society*, 1909.

[13] A. Dixon. The eliminant of three quantics in two independent variables: (second paper). *Proceedings of the London Mathematical Society*, 1909.

[14] X. Dou and K.-L. Ting. Module approach for branch analysis of multiloop linkages/manipulators. *Mechanism and Machine Theory*, 1998.

[15] R. W. Floyd. Algorithm 97: Shortest path. *Communications of the ACM*, 1962.

[16] R. Franke. *Vom Aufbau der Getriebe*. VDI-Verlag, 1958.

[17] A. Kecskeméthy, T. Krupp, and M. Hiller. Symbolic processing of multiloop mechanism dynamics using closed-form kinematics solutions. *Multibody System Dynamics*, 1997.

[18] H. Kim, S. Hamid, and A. Soni. Synthesis of six-link mechanisms for point path generation. *Journal of Mechanisms*, 1971.

[19] K. Kutzbach. Mechanische leitungsverzweigung, ihre gesetze und anwendungen. *Maschinenbau. Betrieb*, 1929.

[20] C. Liebchen and R. Rizzi. Classes of cycle bases. *Discrete Applied Mathematics*, 2007.

[21] N. Manolescu. A method based on Baranov trusses, and using graph theory to find the set of planar jointed kinematic chains and mechanisms. *Mechanism and Machine Theory*, 1973.

[22] J. M. McCarthy. Mechanism synthesis theory and the design of robots. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, 2000.

[23] J. M. McCarthy and G. S. Soh. *Geometric Design of Linkages*. Springer, 2010.

[24] B. D. McKay. Isomorph-free exhaustive generation. *Journal of Algorithms*, 1998.

[25] C. McLarnan. Synthesis of six-link plane mechanisms by numerical analysis. *Journal of Engineering for Industry*, 1963.

[26] T. Mruthyunjaya. Kinematic structure of mechanisms revisited. *Mechanism and Machine Theory*, 2003.

[27] T. Mruthyunjaya and M. Raghavan. Computer-aided analysis of the structure of kinematic chains. *Mechanism and Machine Theory*, 1984.

[28] D. H. Myszka, A. P. Murray, and C. W. Wampler. Mechanism branches, turning curves and critical points. In *ASME 2012 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE2011), 36th Mechanisms and Robotics Conference*, 2012.

[29] J. Nielsen and B. Roth. Solving the input/output problem for planar mechanisms. *ASME Journal of Mechanical Design*, 1999.

[30] B. Parrish, A. Perez, and J. M. McCarthy. Identification of a usable six-bar linkage for dimensional synthesis. In *Proceedings of EuCoMeS the 4th European Conference on Mechanism Science, Santander Spain*, 2012.

[31] B. E. Parrish and J. M. McCarthy. Use of the Jacobian to verify smooth movement in Watt I and Stephenson I six-bar linkages. In *ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2013.

[32] A. Perez and J. M. McCarthy. Clifford algebra exponentials and planar linkage synthesis equations. *ASME Journal of Mechanical Design*, 2005.

[33] J. M. Porta, L. Ros, and F. Thomas. A linear relaxation technique for the position analysis of multiloop linkages. *IEEE Transactions on*, 2009.

[34] S. A. Shamsudin, A. P. Murray, D. H. Myszka, and J. P. Schmiedeler. Kinematic synthesis of planar, shape-changing, rigid body mechanisms for design profiles with significant differences in arc length. *Mechanism and Machine Theory*, 2013.

[35] W.-B. Shieh. *Design and optimization of planar leg mechanisms featuring symmetrical foot-point paths*. PhD thesis, University of Maryland, 1996.

[36] J. R. Silvester. Determinants of block matrices. *The Mathematical Gazette*, 2000.

[37] R. Simoni, A. Carboni, H. Simas, and D. Martins. Enumeration of kinematic chains and mechanisms review. In *13th World Congress in Mechanism and Machine Science, Guanajuato, Mexico*, 2011.

[38] R. Simoni, A. P. Carboni, and D. Martins. Enumeration of parallel manipulators. *Robotica*, 2009.

[39] G. S. Soh and J. M. McCarthy. Synthesis of eight-bar linkages as mechanically constrained parallel robots. In *12th IFToMM world congress A*, 2007.

[40] G. S. Soh and J. M. McCarthy. The synthesis of six-bar linkages as constrained planar 3R chains. *Mechanism and Machine Theory*, 2008.

[41] G. S. Soh, A. Perez, and J. M. McCarthy. The kinematic synthesis of mechanically constrained planar 3R chains. *Proceedings of EuCoMeS, the first European Conference on Mechanism Science, Obergurgl Austria*, 2006.

[42] R. P. Sunkari and L. C. Schmidt. Critical review of existing degeneracy testing and mobility type identification algorithms for kinematic chains. In *ASME 2005 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2005.

[43] R. P. Sunkari and L. C. Schmidt. Structural synthesis of planar kinematic chains by adapting a McKay-type algorithm. *Mechanism and Machine Theory*, 2006.

[44] K. Ting, C. Xue, J. Wang, and K. R. Currie. General mobility identification and rectification of Watt six-bar linkages. In *ASME 2007 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE2007), 31st Mechanisms and Robotics Conference*, 2007.

[45] K. Ting, C. Xue, J. Wang, and K. R. Currie. Stretch rotation and complete mobility identification of Watt six-bar chains. *Mechanism and Machine Theory*, 2009.

[46] K.-L. Ting and X. Dou. Classification and branch identification of Stephenson six-bar chains. *Mechanism and Machine Theory*, 1996.

[47] K.-L. Ting, J. Wang, and C. Xue. Unified mobility identification and rectification of six-bar linkages. In *ASME 2009 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE2009), 33rd Mechanisms and Robotics Conference*, 2009.

[48] L.-W. Tsai. *Mechanism design: enumeration of kinematic structures according to function*. CRC press, 2000.

[49] E. R. Tuttle. Generation of planar kinematic chains. *Mechanism and Machine Theory*, 1996.

[50] A. Verho. An extension of the concept of the group. *Mechanism and Machine Theory*, 1973.

[51] C. W. Wampler. Solving the kinematics of planar mechanisms by Dixon determinant and a complex plane formulation. *ASME Journal of Mechanical Design*, 2001.

[52] J. Wang, K. Ting, and C. Xue. Discriminant method for the mobility identification of single degree-of-freedom double-loop linkages. *Mechanism and Machine Theory*, 2010.

[53] S. Warshall. A theorem on boolean matrices. *Journal of the Association for Computing Machinery (JACM)*, 1962.

[54] H. Whitney. Non-separable and planar graphs. *Transactions of the American Mathematical Society*, 1932.

[55] E. T. Wolbrecht, D. J. Reinkensmeyer, and A. Perez-Gracia. Single degree-of-freedom exoskeleton mechanism design for finger rehabilitation. In *Rehabilitation Robotics (ICORR), 2011 IEEE International Conference on*, 2011.

[56] L. Woo. Type synthesis of plane linkages. *Journal of Engineering for Industry*, 1967.