ABSTRACT

Title of dissertation: STRUCTURAL SYNTHESIS AND ANALYSIS OF PLANAR AND SPATIAL MECHANISMS SATISFYING GRUEBLER'S DEGREES OF FREEDOM EQUATION

Rajesh Pavan Sunkari
Doctor of Philosophy, 2006

Dissertation directed by: Dr. Linda Schmidt
Department of Mechanical Engineering

Design of mechanisms is an important branch of the theory of mechanical design. Kinematic structural studies play an important role in the design of mechanisms. These studies consider only the interconnectivity pattern of the individual links and hence, these studies are unaffected by the changes in the geometric properties of the mechanisms. The three classical problems in this area and the focus of this work are: synthesis of all non-isomorphic kinematic mechanisms; detection of all non-isomorphic pairs of mechanisms; and, classification of kinematic mechanisms based on type of mobility. Also, one of the important steps in the synthesis of kinematic mechanisms is the elimination of degenerate or rigid mechanisms. The computational complexity of these problems increases exponentially as the number of links in a mechanism increases. There is a need for efficient algorithms for solving these classical problems. This dissertation illustrates the successful use of techniques from graph theory and combinatorial optimization to solve structural kinematic problems.

An efficient algorithm is developed to synthesize all non-isomorphic planar kinematic mechanisms by adapting a McKay-type graph generation algorithm in combination with a degeneracy testing algorithm. This synthesis algorithm is about 13 times faster than the most recent synthesis algorithm reported in the literature.

There exist efficient approaches for detection of non-isomorphic mechanisms based on eigenvalues and eigenvectors of the adjacency or related matrices. However these approaches may fail to detect all cases. The reliability of these approaches is established in this work. It is shown, for the first time, that if the number of links is less than 15, the eigenvector approach detects all non-isomorphic mechanisms. A matrix is also proposed whose characteristic polynomial detects non-isomorphic mechanisms with a higher reliability than the adjacency or Laplace matrix.

An erroneous assumption often found in structural studies is that the graph of a planar kinematic chain is a planar graph. It is shown that all the existing algorithms for degeneracy testing and mobility type identification, except those by Lee and Yoon, have this error. Further, Lee and Yoon's algorithms are heuristic in nature and were not rigorously proved. Several structural results and implicit assumptions for planar kinematic chains are proved in this work without relying on the erroneous assumption. These new results provide the mathematical justification for Lee and Yoon's algorithms, thereby validating the adoption of the Lee and Yoon's algorithms for practical applications. A polynomial-time algorithm based on combinatorial optimization techniques is proposed for degeneracy testing. This polynomial-time algorithm is the first degeneracy testing algorithm that works for both planar and spatial kinematic mechanisms with different types of joints.

STRUCTURAL SYNTHESIS AND ANALYSIS OF PLANAR AND SPATIAL MECHANISMS SATISFYING GRUEBLER'S DEGREES OF FREEDOM EQUATION

by

Rajesh Pavan Sunkari

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2006

Advisory Committee:
Dr. Linda Schmidt, Chair/Advisor
Dr. Shapour Azarm
Dr. Peter Sandborn
Dr. Greg Schultz
Dr. Lawrence Washington

DEDICATION

To my parents, for their guidance, support, love and patience.

ACKNOWLEDGMENTS

First and foremost, I would like to thank my advisor, Dr. Linda Schmidt, for her guidance and support in both academic and nonacademic aspects during my graduate study. Over the past five years she has played the role of a mentor as well as a parent and a friend. Her motivational speeches helped me succeed at several fronts during my dissertation work.

I would like to thank the committee members, Dr. Shapour Azarm, Dr. Peter Sandborn, Dr. Greg Schultz and Dr. Lawrence Washington for their inputs and comments. I would also like to thank Dr. Sergey Novikov and Dr. Niranjan Ramachandran for their efforts spent explaining several intricate mathematical concepts.

I would like to thank the Department of Mechanical Engineering, University of Maryland, College Park, for providing the financial support during my graduate study. I would like to thank Dr. Sami Ainane for his advice on teaching and for giving me the valuable opportunity to be his teaching assistant. I wish to thank my colleagues, past and present, from the Designer Assistance and Tools Laboratory. I extend my thanks to the faculty and staff of the Department of Mechanical Engineering who over the years have helped me in many ways.

I would like to express my deepest gratitude to my parents for their guidance, support, love and patience. Without their support and encouragement throughout my life, I would not have achieved this important milestone in my career.

I would like to thank Gayatri Cuddalorepatta for patiently proof reading my technical drafts and providing critical commentary. I would also like to specially thank her for providing motivation and helping me stay focused on my dissertation during difficult times. I would like to thank Niranjan Balachandran for patiently explaining some of the subtle concepts of graph theory and also for his insightful comments on the dissertation.

I would like express my gratitude to my housemates Joe (Joseph Varghese), Mak (Anil Maddulapalli), Mungi (Kiran Panchamgam), Raju (Rajasekhar Jammalamadaka), Reddy (Prabhakar Gudla), Sasi (Sasidhar Karavadi) and my good friends Harish Dontineni, Krishna Kaipa, Nagendra Cuddalorepatta and Neeraja Balachander, for providing strong moral support during my graduate study.

Regrettably, it is impossible to list all those who deserve credit and I apologize to those that I might have inadvertently left out.

TABLE OF CONTENTS

List of Tables vii

List of Figures viii

1 Introduction 1
- 1.1 Motivation 1
- 1.2 Research Theme and Objectives 5
- 1.3 Methodology 6
- 1.4 Organization of the Dissertation 10

2 Background 11
- 2.1 Kinematic Chains and Mechanisms 11
- 2.2 Representations of Kinematic Chains 14
- 2.3 Mobility type of Kinematic Chains 16

3 Literature Review of Structural Studies 20
- 3.1 Introduction 20
- 3.2 Structural Synthesis 21
  - 3.2.1 Link Assortment 23
  - 3.2.2 Building up from Simpler Chains 26
  - 3.2.3 Other Methods 27
- 3.3 Structural Analysis 28
  - 3.3.1 Structural Isomorphism 32
  - 3.3.2 Degeneracy Testing 37

4 Structural Synthesis - Adapting a McKay-Type Algorithm 39
- 4.1 Introduction 39
- 4.2 Background on Generation of Non-Isomorphic Combinatorial Structures 40
  - 4.2.1 Homomorphism Principle-Type Algorithms 41
  - 4.2.2 Read/Faradzev-Type Algorithms 43
  - 4.2.3 McKay-Type Algorithms 45
- 4.3 Approach 50
- 4.4 Methodology and Results 51
- 4.5 Discussion 56
- 4.6 Summary 57

5 Efficiency and Reliability of Spectral Methods 59
- 5.1 Introduction 59
- 5.2 Reliability of the characteristic polynomial method for isomorphism detection 60
- 5.3 Reliability of the eigenvector method for isomorphism detection 62
- 5.4 Efficiencies of the spectral methods for isomorphism 67

5.5 Implementation 73
5.6 Discussion 75
5.7 Summary 77

6 Laplace and Extended Adjacency Matrices for Isomorphism Detection 78
6.1 Introduction 78
6.2 Characteristic polynomial methods 78
6.3 Laplace matrix 80
6.4 Extended adjacency matrix 82
6.5 Discussion 89
6.6 Summary 94

7 Reevaluation of degeneracy testing and mobility type identification algorithms 95
7.1 Introduction 95
7.2 Evaluation of existing degeneracy testing algorithms 96
7.2.1 Basic degeneracy testing algorithm 96
7.2.2 Hwang and Hwang's degeneracy testing algorithm 99
7.2.3 Lee and Yoon's degeneracy testing algorithm 105
7.2.4 Proposition 7.10 and its proof 106
7.3 Support for Lee and Yoon's mobility type identification algorithms . . 109
7.4 Future directions for generalizing structural analysis algorithms . . . 112
7.5 Summary 114

8 Polynomial-time Degeneracy Testing Algorithm 115
8.1 Introduction 115
8.2 Formulation of Degeneracy Testing as an Optimization Problem . . . 118
8.3 Network Flows 122
8.4 Minimum Cuts and 0-1 Optimization 125
8.5 Solution of Degeneracy Testing 127
8.6 Results and Discussion 130
8.7 Summary 134

9 Contributions and Future Work 136
9.1 Summary of Results 136
9.2 Relevance of Current Research 139
9.3 Contributions 140
9.4 Future Work 141

A Mathematical Background 145
A.1 Graph Theory 145
A.2 Some Concepts from Group Theory 149

Bibliography 150

LIST OF TABLES

4.1 Table of 1-DOF non-isomorphic kinematic chains. 53

4.2 Table of 2-DOF non-isomorphic kinematic chains. 54

4.3 Table of 3-DOF non-isomorphic kinematic chains. 55

4.4 Table of 4-DOF non-isomorphic kinematic chains. 55

5.1 Table of non-isomorphic adjacency isospectral kinematic chains. . . . 63

5.2 Table of non-isomorphic kinematic chains satisfying the hypothesis of converse of the Corollary 5.5. 68

5.3 Table of non-isomorphic kinematic chains with similar eigenvectors corresponding to the largest eigenvalue. 72

6.1 Table of Laplacian isospectral and non-isomorphic kinematic chains. . 83

6.2 Table of Laplacian and adjacency isospectral, and non-isomorphic kinematic chains. 84

6.3 Table of $A^{(d)}$ isospectral and non-isomorphic kinematic chains. 88

6.4 Table of isospectral and non-isomorphic kinematic chains for all extended matrices. 90

6.5 Table of all $A^{(d)}$ isospectral chains and $A^{(2)}$ isospectral and non-isomorphic chains. 91

6.6 Table of the Laplacian isospectral non-isomorphic chains with different degree sequence. 92

6.7 Table of the Laplacian isospectral non-isomorphic graphs with different degree sequence. 93

8.1 Table of 1 & 2 DOF non-isomorphic kinematic chains. 131

8.2 Table of 3 and 4 DOF non-isomorphic kinematic chains. 132

LIST OF FIGURES

1.1 Hierarchy of kinematic structural studies. 2

1.2 Research objectives in relation to the hierarchy of structural studies. . 7

2.1 Different types of links 12

2.2 Different types of joints 12

2.3 Binary joint equivalents of k-nary joint 13

2.4 Degenerate kinematic chains 14

2.5 Equivalence of Structural and Graph Representation 16

2.6 Graph and Franke's representation of mechanism 17

3.1 Kinematic Synthesis Process. 22

3.2 Detailed Kinematic Synthesis Process. 22

5.1 An example that requires multiple comparisons in eigenvector approach 71

7.1 Counterexample for Hwang and Hwang's algorithm. 104

7.2 Steps illustrating the Hwang and Hwang's algorithm. 104

8.1 Degenerate kinematic chains with identical joints 116

8.2 Degenerate kinematic chains with no binary links 117

8.3 Non-degenerate and degenerate spatial kinematic chains with same topology of graph 117

8.4 Non-degenerate and degenerate spatial kinematic chains with same topology of graph 118

8.5 Constraint graphs of spatial mechanisms 119

8.6 Non-planar spatial kinematic chain 133

A.1 Different kinds of graphs 147

A.2 Graph with 2 components 147
A.3 A pair of isomorphic graphs 148
A.4 K_5 and $K_{3,3}$ 148

Chapter 1

Introduction

1.1 Motivation

The study of mechanisms began as early as the Iron age when people started building simple machines. Today mechanisms are encountered everywhere from nanomechanical devices to the space shuttle. Due to the overwhelming need for novel mechanisms, automated design of mechanisms from a given set of functional requirements is advantageous. During the conceptual design phase some of the functional requirements can be transformed into structural requirements of the mechanisms.

The structural studies of kinematic mechanisms are broadly divided into structural synthesis and structural analysis. The structural synthesis of kinematic chains involves enumerating all possible kinematic chains having a specified number of links, degrees of freedom and types of joints. One of the important steps in this process is the detection of degenerate kinematic chains using Gruebler's degrees of freedom equation [1]. In structural analysis, the major problems are isomorphism detection and identification of type of mobility. Figure 1.1 shows the hierarchy of structural studies of kinematic mechanisms.

Structural synthesis usually involves generation of the list of several, possibly redundant, kinematic chains followed by explicit elimination of isomorphs. Due to the computational inefficiency of isomorph detection, efficient kinematic synthe-

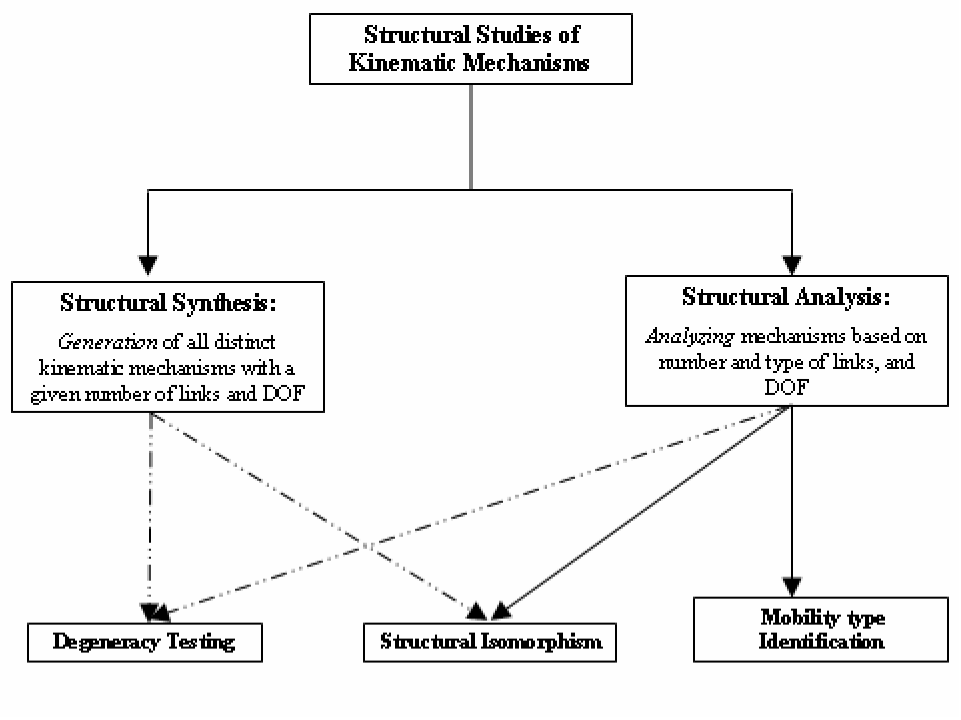


Figure 1.1: Hierarchy of kinematic structural studies.

sis algorithms must minimize the explicit isomorph detection step. Researchers in the mechanisms community follow different approaches to enumerate the non-isomorphic kinematic chains. However, most of the existing methods are not computationally efficient and hence only generation of non-isomorphic kinematic chains with fewer links was possible. Furthermore there are several discrepancies in the results obtained by researchers [2] in the mechanisms community.

Hence there is a need for an efficient and reliable algorithm for the synthesis of kinematic chains and a reexamination of the existing results on kinematic synthesis to validate the existing methods. This forms the first objective of this dissertation.

The structural isomorphism problem is an age old problem which has long been taunting researchers in the mechanisms community. Researchers developed different kinds of codes and indices that would indicate if a pair of kinematic chains is isomorphic. However most of these codes or indices are either computationally inefficient or unreliable. Much research has also been devoted in graph theory to the more general graph isomorphism problem. This problem was so popular that in 1977 it was named “The Graph Isomorphism Disease” [3]. In graph theory, researchers were able to find efficient solutions for certain classes of highly structured graphs. However, there is widespread skepticism about finding polynomial-time algorithms for all kinds of graphs [4, 5].

Since the graph of a kinematic chain has a well defined recursive structure, it can be hoped that one may develop a polynomial-time algorithm for kinematic chains. One approach would be to use spectral methods. Spectral methods use properties that are a function of the eigenvalues of graph matrices of a kinematic chain to identify isomorphism. These properties include the characteristic polynomial, eigenvalues and eigenvectors. Since the algorithms for finding spectral properties can be solved in polynomial-time, finding a spectral invariant which distinguishes non-isomorphs would result in finding a polynomial-time algorithm for the isomorphism problem. This motivated Objective 2 of this study to establish efficiency and reliability of spectral methods.

Researchers in graph theory have used another matrix, called the Laplace matrix, for similar structural results. This motivated the work of Objective 3, which is to evaluate the reliability of the characteristic polynomial of the Laplace matrix and also to develop a graph matrix of a kinematic chain that could be used to detect structural isomorphism with high reliability.

One of the common methods for structural classification of kinematic chains is based on the type of mobility a kinematic chain possesses. Broadly, a kinematic chain can possess partial or total mobility. A kinematic chain is called degenerate or rigid if any one of its closed subchains has less than one degrees of freedom (meaning the subchain cannot move). The problems of mobility type identification and degeneracy testing, when translated onto the graph of the kinematic chain, involves checking if a linear inequality involving the vertices and edges of the graph is valid for each induced closed subgraph. Several algorithms have been proposed for degeneracy testing but almost all of them are either computationally inefficient or valid for only certain types of chains.

In the structural analysis of planar kinematic chains one common error is to assume that the graph of a planar kinematic chain is a planar graph [6]. It was pointed out recently [7] that the main cause for this is the misuse of the word 'planar'. In mechanisms it means to *lie in one or more parallel planes* but in graph theory it means to *lie on a single plane.* Several algorithms for structural analysis, including the algorithms for degeneracy testing, isomorphism testing and mobility type identification, work under the assumption that the graph of a planar kinematic chain is a planar graph. In contrast Lee and Yoon's [8, 9] algorithms for degeneracy testing and mobility analysis are claimed to be valid for all planar kinematic chains with revolute joints but their published work does not include proper and complete mathematical proofs. *So the need exists to develop structural results for planar kinematic chains without using the planarity assumption.* This is taken up as Objective 4 and these structural results are used to provide justification for the algorithms of Lee and Yoon.

The extension of Lee and Yoon's algorithms to planar mechanisms with different types of joints results in certain problems. For the case of spatial mechanisms that satisfy Gruebler's degrees of freedom equation, none of Lee and Yoon's algorithms are applicable. This is because the graph structure of spatial mechanisms is different from that of planar mechanisms due to differences in the corresponding degrees of freedom equations. Also the theoretical complexity of the existing degeneracy testing and mobility type identification algorithms is not known. *So the need exists to consider the problem of finding efficient algorithms for degeneracy testing and mobility type identification that are valid for both planar and spatial kinematic mechanisms.* This is the last objective of this dissertation.

1.2 Research Theme and Objectives

Structural studies of kinematic mechanisms have not fully utilized the recent and more sophisticated techniques of graph theory. The overarching research theme of this work is to solve the structural problems of kinematic mechanisms using the techniques from graph theory and combinatorial optimization. This work illustrates the success of these approaches.

The following research objectives were formulated during the course of research.

Objective 1: To validate the results of structural synthesis of planar kinematic chains using McKay-type algorithms.

Objective 2: To evaluate the reliability and efficiency of spectral methods for isomorphism detection.

Objective 3: To evaluate the reliability of the characteristic polynomial of the Laplace matrix and to develop a graph matrix of a kinematic chain that detects structural isomorphism with high reliability.

Objective 4: To develop structural results for kinematic chains that do not rely on the planarity assumption and use these results to provide justification for the structural analysis algorithms of Lee and Yoon [8, 9].

Objective 5: To develop an algorithm for degeneracy testing that works for both spatial and planar kinematic chains, with planar or non-planar graphs and with different kinds of joints. (This algorithm will assume the validity of Gruebler's degrees of freedom equation for spatial kinematic chains.)

Figure 1.2 shows the relation of the research objectives to the hierarchy of structural studies of kinematic mechanisms.

1.3 Methodology

This section briefly describes the methodology applied to each of the research objectives stated above.

Objective 1: Structural synthesis of planar kinematic chains:

The discrepancies in the structural synthesis results in literature are either due to invalid isomorphism checking or invalid degeneracy testing. Tuttle [10], us-

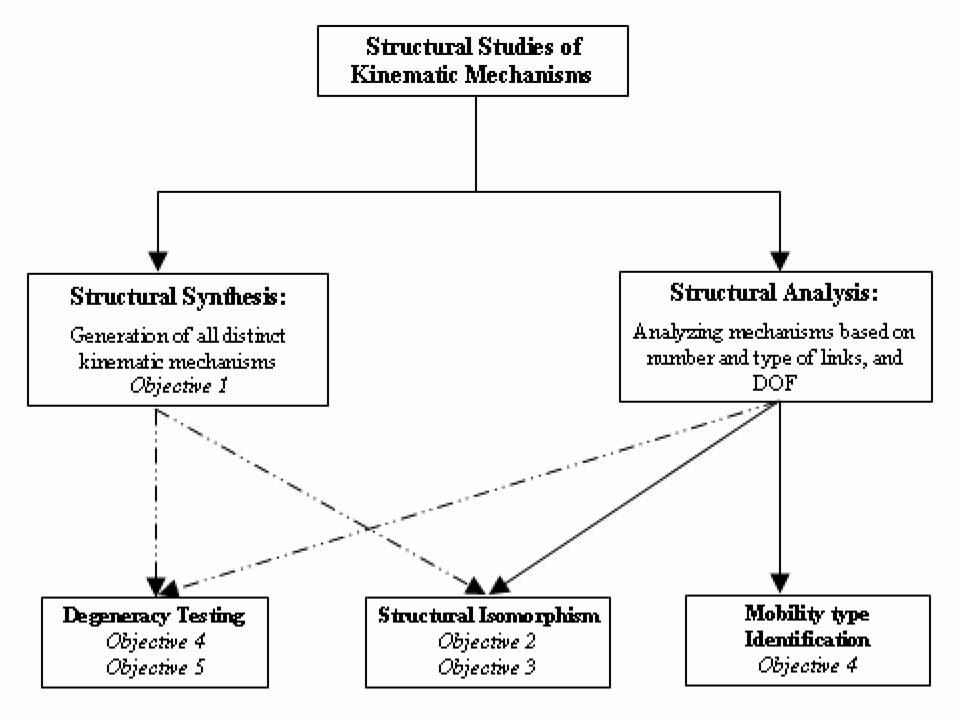


Figure 1.2: Research objectives in relation to the hierarchy of structural studies.

ing group theoretic techniques, was able to develop an efficient method for synthesis of kinematic chains. The success of Tuttle's application of group theoretic methods motivated the use of efficient exhaustive isomorph-free generation algorithms from the field of graph theory. These algorithms apply group theoretic techniques to minimize the explicit isomorphism checking. The exhaustive isomorph-free generation algorithms have never been used for the synthesis of kinematic mechanisms prior to this work. In this dissertation an algorithm belonging to one particular class of exhaustive isomorph-free generation algorithms, called McKay-type, in combination with an efficient degeneracy testing algorithm is used for the synthesis of planar mechanisms. The results are elaborated in Chapter 4.

Objective 2: Reliability and efficiency of spectral methods for isomorphism detection:

This work establishes, for the first time, the reliability of the existing spectral techniques—characteristic polynomial and eigenvector approaches—for isomorphism detection. The reliability of the use of the characteristic polynomial of the adjacency matrix is established by computationally determining the number of pairs of non-isomorphic chains with up to 14 links and 1, 2 and 3 degrees of freedom. The most recent eigenvector approach [11, 12, 13] is critically reviewed and the correct mathematical proof is provided for the statement which forms the basis for this approach. Chapter 5 contains the details of this objective.

Objective 3: Novel graph matrices for isomorphism detection:

The Laplace matrix of a graph is used extensively in the field of algebraic graph theory to characterize a graph by its spectral properties. This work establishes the reliability of the characteristic polynomial of the Laplace matrix for isomorphism detection of a kinematic chain through computational means. In the search for a matrix whose characteristic polynomial unfailingly detects isomorphism, novel matrices called the extended adjacency matrices are developed and examined. The results are discussed in Chapter 6.

Objective 4: Reevaluation of algorithms for degeneracy testing and mobility type identification:

Mobility analysis is one of the fundamental problems of structural studies of kinematic chains. Degeneracy testing, an important step in structural synthesis, can be considered as a part of the mobility analysis due to the similarity of the two problems. One common shortcoming in the algorithms for solving these two problems is the assumption that the graph of a planar kinematic chain is a planar graph. This dissertation shows that almost all the mobility type identification and degeneracy testing algorithms, except those of Lee and Yoon, are based on the assumption that the graph of a planar kinematic chain is a planar graph. The two most efficient algorithms on degeneracy testing (those by Hwang and Hwang, and Lee and Yoon) are investigated and reappraised with respect to this assumption. Further details on this objective are given in Chapter 7.

Objective 5: Polynomial-time algorithm for degeneracy testing:

This work develops an original, polynomial-time, algorithm for degeneracy testing that is applicable for both planar and spatial mechanisms with different types of joints. This is done by first reducing the degeneracy testing problem to a 0-1 quadratic optimization problem with a single constraint. Next the 0-1 quadratic optimization problem is further reduced to minimizing $|E|$ 0-1 quadratic functions, where $|E|$ is the number of edges of the graph. Chapter 8 contains the details of the research done to achieve this objective.

1.4 Organization of the Dissertation

Chapter 2 presents basic results from mechanisms literature to serve as a background for problems addressed in this work. Chapter 3 reviews the existing literature in the areas of kinematic synthesis and analysis, including the isomorphism and degeneracy detection problems. Chapter 4 describes the different types of exhaustive non-isomorphic generation methods for combinatorial structures. This chapter also describes the adaptation of a McKay-type algorithm for structural synthesis and discusses the results obtained. Chapter 5 establishes the efficiency and reliability of different existing spectral methods for isomorphism detection. Chapter 6 describes the methods and results of the use of the characteristic polynomial of the Laplace matrix for isomorphism detection. Chapter 6 chapter also presents a novel matrix whose characteristic polynomial detects isomorphism with high reliability. Chapter 7 presents the inadequacies of existing degeneracy testing algorithms and the justifications for the validity of Lee and Yoon's [8] algorithms. Chapter 8 describes a polynomial-time algorithm for degeneracy testing using combinatorial optimization methods. Lastly, Chapter 9 discusses the contributions and future directions of the work. The mathematical concepts from graph theory and group theory that are used in this work are briefly described in the Appendices.

Chapter 2

Background

This chapter presents basic definitions relevant to the study of kinematic chains and mechanisms. This chapter can be skipped by the readers with mechanisms background and is included for the sake of completeness of the work. Using the graph theoretic concepts, different equivalent representations of kinematic chains are presented. Finally, different types of mobility are defined for use in Chapters 7 and 8.

2.1 Kinematic Chains and Mechanisms

A kinematic chain consists of links which are connected pairwise by joints. Each link in a valid kinematic chain should have at least two distinct joints which are in turn connected to two other distinct links. If a link has k joints, it is called a k-nary link. Similarly, if a joint connects k links then it is called a k-nary joint. Figures 2.1 and 2.2 show different kinds of links and joints where the links are represented by a hatched polygon. (This representation is the structural representation as defined in the next section.) A k-nary link is also said to be a link with valency or degree equal to k. Any k-nary joint connecting k links can be equivalently represented by k links and $k-1$ binary joints as shown in Figure 2.3. A kinematic chain is said to closed if the degree of each link is at least two and is said to be open otherwise.

A mechanism is obtained from a kinematic chain when one of the links is fixed. All the mechanisms that can be obtained from a given kinematic chain form kinematic inversions of the chain. A basic kinematic chain is a kinematic chain such that each joint connects exactly two distinct links. Throughout the reminder of this dissertation, unless otherwise explicitly stated, a kinematic chain means a basic kinematic chain. A kinematic chain or mechanism is called planar if all the links of the chain or mechanism are in one plane or parallel planes.

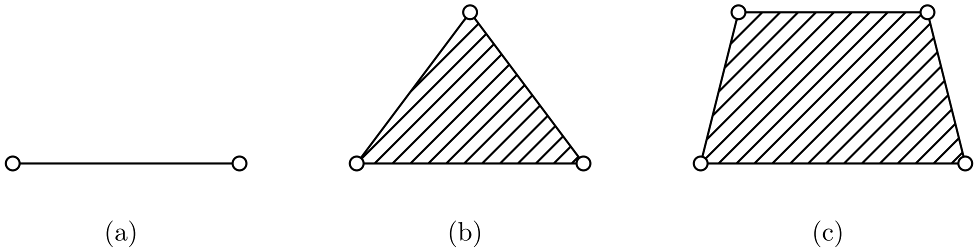

(a) (b) (c)

Figure 2.1: (a) Binary Link, (b) Ternary Link, and (c) Quaternary Link

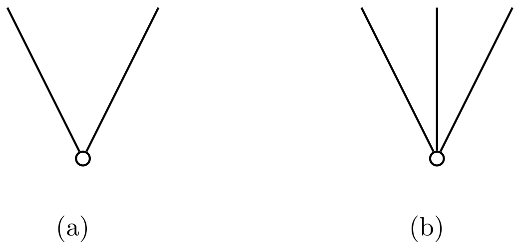

(a) (b)

Figure 2.2: (a) Binary joint and (b) Ternary joint

The degrees of freedom of a mechanism are the number of independent parameters required to completely specify the configuration of the mechanism in space.

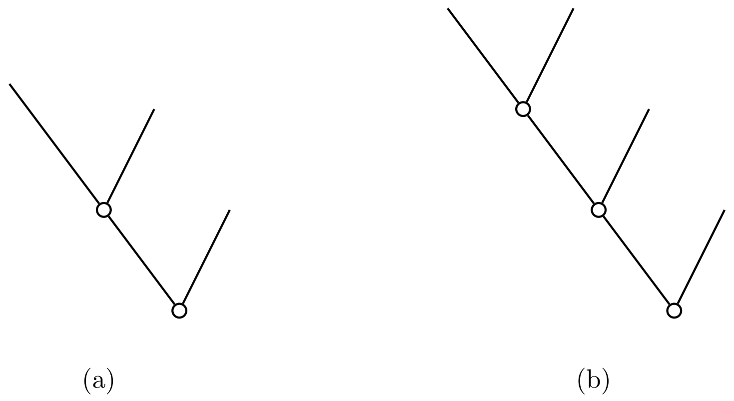


Figure 2.3: Binary joint equivalents of (a) Ternary joint as shown in Figure 2.2(b) and (b) Quaternary joint

The number of degrees of freedom, f, of a mechanism, M, is

$$\mathrm{DOF}(M) = \lambda(l-1) - \sum_{i=1}^{j} c_i \tag{2.1}$$

where l is the number of links, j is the number of joints, λ is the motion parameter and c_i is the degrees of constraints on relative motion imposed by a joint. λ is 3 (2 translations and 1 rotation) for planar mechanisms and is 6 (3 translations and 3 rotations) for spatial mechanisms. Considering only planar mechanisms with revolute joints, for which $c_i = 2$, we can rewrite the equation (2.1) as follows

$$3l - 2j = f + 3 \tag{2.2}$$

This is called the Gruebler equation [1]. The number of degrees of freedom (f) of a kinematic chain, K, is also called the mobility of K. This is different than the types of mobility defined later in Section 2.3. In this work, a higher pair for planar kinematic mechanism means a joint with more than one degrees of freedom.

A kinematic chain, K', contained within or equal to a given kinematic chain, K, is called a subchain of K. If the containment of K' in K is proper ($K' \neq K$), then K' is called a proper subchain of K. A kinematic chain, K, is called degenerate or rigid if any one of its subchains has less than one degrees of freedom. Hence a degenerate kinematic chain contains a subchain that cannot move. The kinematic chain shown in Figure 2.4(a) is an example of a degenerate chain, where links a, b and c form a structure or subchain of zero degrees of freedom. A kinematic chain, K, is called a basic rigid chain if the degrees of freedom of K are equal to 0 and the degrees of freedom of every proper subchain are greater than zero. In other words, a basic rigid chain is a degenerate kinematic chain such that all of its proper subchains are non-degenerate. Figure 2.4(b) shows an example of a basic rigid chain.

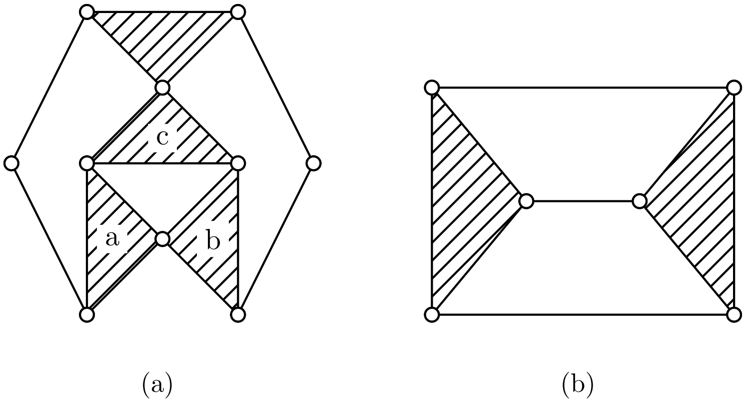


Figure 2.4: (a) 8-link degenerate mechanism and (b) 5-link basic rigid chain

2.2 Representations of Kinematic Chains

There are three main representations of kinematic chains: Structural Representation; Graph Representation; and Contracted Graph Representation.

1. Structural Representation: In structural representation, each link of a kinematic chain is denoted by a polygon whose vertices represent the joints. In the case of a binary link the polygon reduces to a line between two joints. Figure 2.1 shows different types of links (represented by cross-hatched polygons) and Figure 2.6(a) shows a structural representation of a kinematic chain. This is the most intuitive representation.

2. Graph Representation: In graph representation, any kinematic chain can be uniquely represented by a simple graph, called the graph of the kinematic chain. The links of the chain correspond to the nodes of the graph and the joints correspond to the edges. Figure 2.5 shows the correspondence of the links and joints in the structural representation to the vertices and edges, respectively, in the graph representation of a 5-link kinematic chain. Figure 2.6(a) and 2.6(b) show the structural and graph representation respectively of another kinematic chain.

3. Contracted Graph Representation or Franke's Notation: Given a graph, G, one can form a contracted graph, CG. CG along with the lengths of the binary strings corresponding to the each edge of CG uniquely represents G. Hence, the kinematic chain can be uniquely represented by CG. Figure 2.6 shows the contracted graph representation of a kinematic chain. This contracted graph representation is historically called Franke's notation for kinematic chains. Classically, in Franke's notation each polygonal link is represented by a circle and the string of binary links by lines. The degree of the link is placed within

the circle and the numbers corresponding to the number of consecutive binary links (including zero) in each binary string is placed next to the corresponding lines [14].

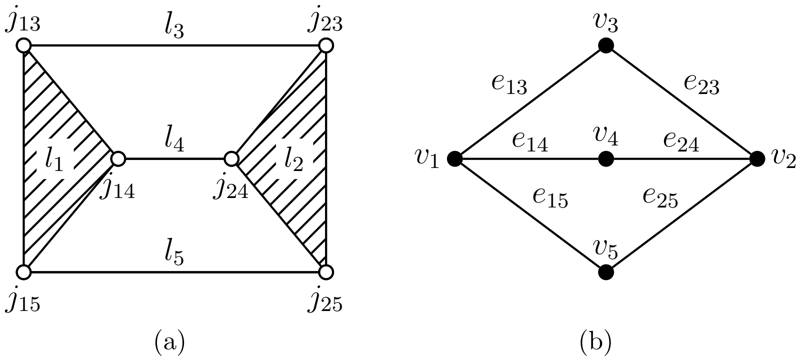


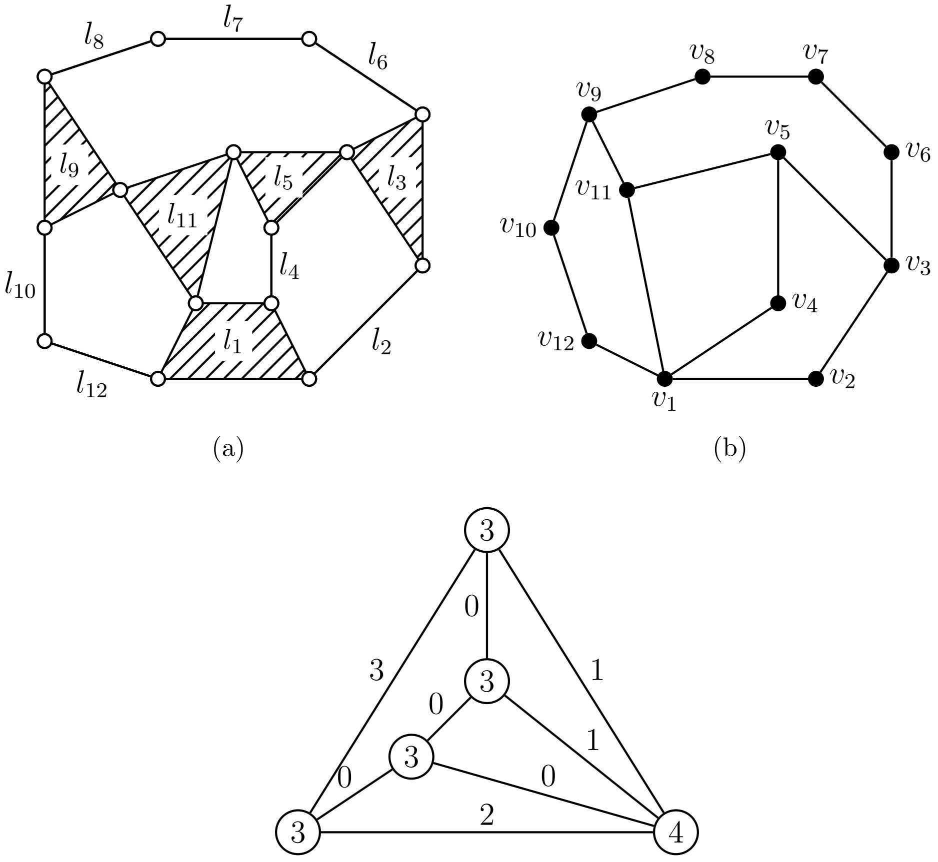
Figure 2.5: One-one correspondence between the links and joints of the structural representation to the vertices and edges of the graph representation (e.g. $l_1 \leftrightarrow v_1$ and $j_{23} \leftrightarrow e_{23}$) (a) Structural Representation and (b) Graph Representation

Since most of the analysis done in this work is based on graph theory, from this point forward, a kinematic chain will be represented by its graph. The words link and vertex and joint and edge are used interchangeably. All the invariants or functions of a given kinematic chain, like degrees of freedom, are considered to be the invariants or functions of the corresponding graph.

2.3 Mobility type of Kinematic Chains

In structural analysis the mechanisms are analyzed based on functional requirements. This mainly involves determining:

1. The type of freedom or mobility of the kinematic chain.



(a)

(b)

(c)

Figure 2.6: (a) Structural Representation, (b) Graph Representation and (c) Contracted Graph or Franke's Representation of a 12-link planar mechanism

2. The structurally distinct mechanisms that can be obtained from the chain.

3. The structurally distinct driving mechanisms that can be obtained from the chain.

Of the three, analysis based on the type of mobility is most common. The other two are not addressed in this work and hence only types of mobility are defined in this section.

A kinematic chain can possess one or more of the four types of mobility.

1. Non-degenerate mobility: A kinematic chain is said to posses non-degenerate mobility if it is not degenerate, that is, if all its subchains have more than zero degrees of freedom.

2. Fractionated mobility: A kinematic chain is said to posses fractionated mobility if it has a separation link or joint which, when cut into two splits the chain into separate (closed) kinematic chains. The graph of a non-fractionated kinematic chain is a block, as the term is defined in graph theory.

3. Partial mobility: A kinematic chain with $f > 0$ degrees of freedom is said to posses partial mobility if it has at least one subchain with degrees of freedom, $f^{'}$, such that $0 \leq f^{'} < f$.

4. Total mobility: A kinematic chain with $f > 0$ degrees of freedom is said to posses total mobility if all its subchains have degrees of freedom $f^{'} \geq f$.

In Chapter 7 a more general mobility type, called α-mobility, is introduced. α-mobility encompasses the definitions of non-degenerate, partial and total mobility.

Further details on structural studies of kinematic mechanisms can be found in [1].

Chapter 3

Literature Review of Structural Studies

This chapter presents the review of the literature in methods of structural synthesis and analysis of kinematic mechanisms. Studies on structural synthesis and analysis of kinematic mechanisms are together called structural studies of kinematic mechanisms.

3.1 Introduction

First, kinematic synthesis means obtaining a mechanism satisfying certain requirements by applying a systematic procedure. In general, kinematic synthesis is performed in three steps [14]: problem definition, type synthesis and dimension synthesis.

1. Problem definition: The design problem is formulated in terms of topological, functional and geometric requirements or constraints that must be achieved by the mechanism. Topological requirements mainly include a desired type of motion and degrees of freedom. Functional requirements include number of outputs and the functions to be performed by each of these outputs. Geometric requirements include dimensional constraints, and force and time response constraints.

2. Type synthesis: Type synthesis is the determination of the structure of the

mechanism from its desired kinematic performance as defined by Freudenstein and Dobrjanskyj [15]. The structure of a mechanism is generally specified by the number and type of links including input links and ground links, the number and type of joints and their connectivity. Type synthesis is generally done in two steps: structural synthesis and structural analysis which are discussed in more detail in the next section.

3. Dimension synthesis: This type of synthesis involves finding the necessary physical dimensions for all parts of the mechanism. The mechanisms obtained after type synthesis are further refined by imposing the geometric constraints.

Figure 3.1 displays the broad schematic of the steps of kinematic synthesis. Figure 3.2 shows the detailed schematic of the kinematic synthesis. The following sections will present a review of existing literature in structural synthesis and analysis of kinematic mechanisms.

3.2 Structural Synthesis

In structural synthesis all possible mechanical structures satisfying a given set of requirements like degrees of freedom, number of links, are determined. More precisely, mechanisms are enumerated based on topological requirements. Classically, structural synthesis has concentrated on planar revolute jointed chains [16, 17, 18, 19, 20, 21, 22]. The earliest attempt at synthesis by enumeration was performed by Gruebler [23, 24]. The approaches for structural synthesis of kinematic chains usually fall into one of two classes: by link assortments or by building up from simpler

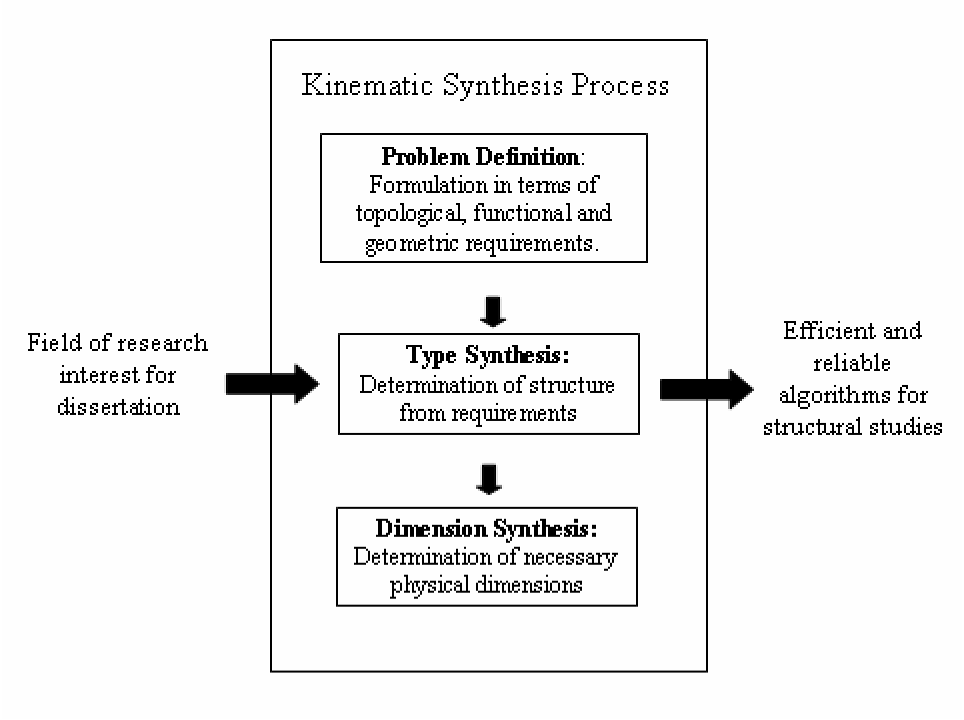


Figure 3.1: Kinematic Synthesis Process.

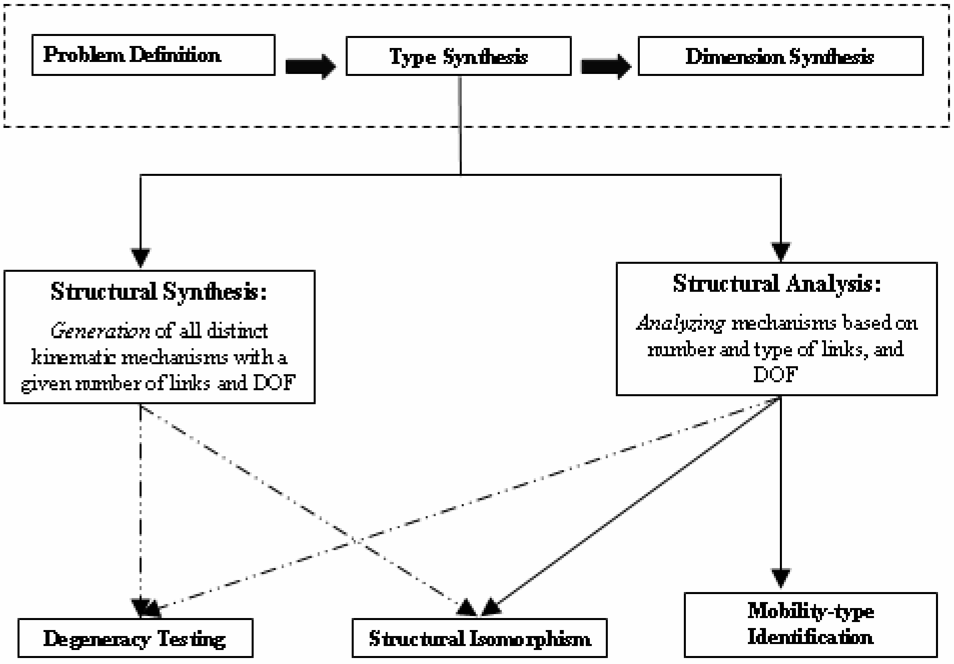


Figure 3.2: Detailed Kinematic Synthesis Process.

chains [14, 2].

3.2.1 Link Assortment

In the link assortment approach, kinematic chains are enumerated by first determining different assortments of (polygonal) links subject to the Gruebler's degree of freedom (DOF) equation (2.2). A kinematic chain can be uniquely represented by the graph whose vertices correspond to the links of the chain and whose edges correspond to the joints of the chain. In graph theoretic terms, the structural synthesis of kinematic chains corresponds to enumeration of graphs subjected to a linear constraint (DOF equation) between vertices and edges of each sub-graph.

Many researchers have applied the link assortment approach to synthesize kinematic chains. The first step in the link assortment synthesis is to generate all the possible contracted graphs for a given number of links and degrees of freedom from the link assortments. Then for each such contracted graph, binary links are inserted in all possible ways to generate the desired chains. Finally the graphs of the chains obtained are checked for the presence of rigid (i.e., degenerate) chains and isomorphs.

Using Franke's notation, Davies and Crossley [25] enumerated the 230 one DOF chains with 10 links, and 40 two DOF chains with 9 links. Freudenstein and Dobrjanskyj [15] and then Crossley [16] using a graph-theoretic approach verified the 16 one DOF chains with 8 links. Woo [17] generated the adjacency matrices of the contacted graphs using the interchange method from each of the degree sequences satisfying the DOF requirements. In his paper, he used permutation group techniques involving the Burnside lemma to enumerate the number of possible ways of adding consecutive binary vertices to the contracted graphs. Woo also sketched all the 230 single DOF chains with 10 links and substantiated Davies and Crossley's [25] count.

Freudenstein [26] emphasized the usage of Polya's theory of enumeration of colored graphs in structural synthesis of kinematic chains. Using Polya's theory one transforms the problem of enumerating distinct colorings with a group action to finding the cyclic index of that group. A typical application in structural synthesis would be to count the number of non-isomorphic kinematic chains that can be obtained from a contracted graph as this can be treated as a standard labelling problem in graph theory. This would involve finding the cyclic index of the automorphism group of the contracted graph which can be equally difficult for graphs with a large number of vertices. Using graph theory and Polya's theory of counting Huang and Soni [20] enumerated kinematic chains with kinematic elements (joints) such as revolute pairs, cam pairs, springs, belt-pulley, piston-cylinder and gears. Huang and Soni represented a chain with different kinds of joints as an edge-n-colored graph, where n represents the different kinds of joints, and hence they reduced the structural synthesis problem to the problem of enumeration of edge-n-colored graphs.

Using the same approach, Tuttle et al. [21, 27] enumerated the kinematic chains systematically, which reduced the need for isomorphism testing to a large extent. Up until now, the work of Tuttle and coworkers, is the best synthesis approach in terms of computational speed. In the spirit of Davies and Crossley's [25] representation, the contracted graphs in Tuttle's work are called bases and edges are called bonds. Tuttle's procedure can be summarized in graph theoretic terms as follows:

1. Determine all possible partitions (numbers and types) of higher links.
2. Find all non-isomorphic bases corresponding to a partition.
3. Determine the symmetry group of the base.
4. Determine all possible combinations of obtaining kinematic chains from a base.
5. Find the list of non-isomorphic chains for each combination by applying the symmetric group of the base.
6. Eliminate all degenerate chains.

Using this approach Tuttle et al. [21, 27] also enumerated distinct inversions of kinematic chains. Tuttle [10] further refined and automated the approach for finding non-isomorphic bases to enumerate the kinematic chains with 4 through 12 links and 1 through 3 DOF. Up until now, this is the most comprehensive enumeration found in mechanisms literature. Classifying the bases into distinct non-isomorphic classes and using the symmetric group of each base considerably reduces the need for isomorphism testing. One drawback to Tuttle's approach is that the synthesis is done as a function of the number of loops, which is a characteristic that is not clearly defined for mechanisms with non-planar graphs.

3.2.2 Building up from Simpler Chains

A different approach in synthesizing kinematic chains is to start building them from simpler kinematic chains. An element of an Assur group is a kinematic chain with free or unpaired joints on the links that, when connected to a stationary link, will have zero DOF. A basic rigid chain is a chain of zero DOF and all of whose subchains have DOF greater than zero. So in other words, an element of an Assur group is a basic rigid chain with one of its links deleted. In the approach of building from simpler chains an Assur group is added to a given chain and the resulting DOF is the same as that of the starting chain. One should observe that degenerate chains will not result from this process if the starting chain is non-degenerate and if the free joints of the Assur group are not all added to a single link. Manolescu and others [28, 29, 30, 18, 19] have used this method extensively in enumerating kinematic chains.

Tischler et al. [31, 32], use a modified version of an orderly generation algorithm of Colbourn and Read [33], called the Melbourne method, for the synthesis of kinematic chains. Tischler et al., first obtain all possible degree partitions for a given number of links and DOF. For each degree partition, starting from a link with highest degree, links are added in all possible ways. This imposes a tree-like structure where a branch exists when there is more than one choice of connection to an intermediate structure. The isomorphs are eliminated by defining a canonical connection using a canonical labelling of the kinematic chains. It has been claimed that this procedure results in very few isomorphs [31, 32]. Even though the method of Tischler et al., claims to minimize explicit isomorphism testing they were only able to generate kinematic chains with a lower number of links compared to other approaches. One of the possible reasons might be the complicated but not canonical rules used to produce canonical connection.

3.2.3 Other Methods

There are few other synthesis methods besides link assortment and building up methods. Mruthyunjaya [34] has used the approach of transformation of binary chains for the synthesis of chains. The first step in this approach is to enumerate the chains with binary links and with single and multiple joints. Then the multiple joints are successively replaced by two simple joints and the binary link is replaced with a polygonal link resulting in different types of kinematic chains with simple and multiple joints. Thus all simple kinematic chains are enumerated.

Hwang and Hwang [35] uses a contracted link adjacency matrix (CLAM) of the kinematic chain for enumeration. CLAM is the modified adjacency matrix of a specific contracted chain where the binary strings of length greater than zero are contracted into one binary string. The off-diagonal entries of CLAM are the same as the corresponding entries of the adjacency matrix. The principle diagonal elements of CLAM corresponding to the polygonal link are the valencies and those corresponding to the contracted link are the negative of the size of the binary chains before contraction. All possible CLAMs are generated by solving linear equations, the degenerate chains are eliminated and finally the isomorphs are detected.

Schmidt et al. [36] use a general graph grammar methodology for the synthesis of kinematic mechanisms. The grammar rules add vertices and loops to intermediate graphs to obtain a graph with desired structural properties. A grammar adaptation of the linear time algorithm of Hopcroft and Wong [37] is used to eliminate the isomorphic copies. The graphs are contracted for isomorphism testing to improve the efficiency. As the grammar rules are based on planar graphs their synthesis only generates kinematic chains with planar graphs. This method was proven to be accurate but was not practical for generation of kinematic chains with larger number of links as all the structures are stored in memory.

Recently Rao and Deshmukh [38] developed a method based on loop formation for structural synthesis which eliminates explicit isomorphism detection. The loop formation techniques are valid only for kinematic chains whose graphs are planar and hence are seriously restricted.

3.3 Structural Analysis

In structural analysis the mechanisms are analyzed based on functional requirements. This mainly involves determining:

1. The type of freedom or mobility of the kinematic chain.
2. The structurally distinct mechanisms that can be obtained from the chain.
3. The structurally distinct driving mechanisms that can be obtained from the chain.

The concepts of total, partial and fractional mobility were introduced by Manolescu [28]. Davies [39] used graph theory to generalize the Manolescu's ideas on total, partial and fractional mobility. Davies also stated several theorems relating the types of mobility to the size of simple loops of the mechanism. These results were used extensively in most of the subsequent papers on structural analysis. However most of Davies' theorems are only valid for planar mechanisms whose graphs are planar.

Sen and Mruthyunjaya [6] presented several counterexamples to Davies' theorems. They, in fact, analyzed the causes for failure of Davies' theorems, one of which being the assumption that the graphs of planar mechanisms are always planar graphs. Surprisingly, even after Sen and Mruthunjaya's work in 1996 many researchers continued to use results based on incorrect planarity assumptions in their analysis. Earlier, Mruthyunjaya and Raghavan [40] used the Davies' theorems and obtained algebraic procedures based on a link-link adjacency matrix representation of kinematic chains for the detection of fractionated and partial mobility of chains. Mruthyunjaya and Raghavan [40] also used the graph representation to characterize fractionated mobility by the presence of a cut-vertex. They developed a criterion for partial mobility by finding the minimum sized loops from the powers of the adjacency matrix. Unfortunately, these results have the same drawbacks as those in Davies' theorems. Mruthyunjaya and Raghavan [41] also presented computer implementations for detecting the type of mobility and for deriving distinct mechanisms and driving mechanisms, from a kinematic chain. Detection of type of mobility relied on the procedures that were developed earlier by Mruthyunjaya and

Raghavan [40]. So the work done by Mruthyunjaya and Raghavan [40, 41] is based on the planarity assumption.

Agrawal and Rao [42] proposed methods for detection of fractionated mobility using the path loop connectivity matrix (Path LOCM). They presented the properties of the loop freedom matrix for fractionated chains. In a subsequent paper Agrawal and Rao [43], presented a method for the analysis of the mobility properties of a kinematic chain by its loop freedom matrix and its permanent function. The DOF equation for a general mechanism in terms of its independent loops and connecting mechanisms was also derived by Agarwal and Rao [44]. Next, they developed a hierarchical classifying scheme of kinematic chains and mechanisms using loop connectivity properties of multi-loop kinematic chains [44]. All the work done by Agarwal and Rao is valid only for the kinematic chains whose graphs are planar as they also use loop based arguments.

Harary and Yan [45] gave a precise definition for a kinematic chain in terms of hypergraphs satisfying certain axioms. These hypergraphs of kinematic chains possess a unique dual which is a simple graph. This simple graph is the conventional graph of a kinematic chain. Harary and Yan stated that the planar block graphs and the planar kinematic chains with simple joints are in a one-to-one correspondence. As presented this statement is incorrect but will be correct if one imposes two more restrictions on the kinds of graphs considered. First, the rigid sub-structures should be eliminated, so each induced sub-graph satisfies the DOF > 0 criteria. Second, the graph need not be planar as it is well known that a graph of planar mechanism can be non-planar [39, 17, 46]. Hence, the correct statement would be, all block graphs satisfying the DOF >0 for all its induced sub-graphs are in one to one correspondence with the planar kinematic chains with simple joints.

Liu and Yu [47] presented a procedure for identifying and classifying multi-DOF and multiple loop mechanisms. They introduced the basic loop as an independent loop with certain properties. To determine the type of mobility, they used the information obtained by calculating the basic loops and their DOF. Rao and Pathapati [48] presented a loop based detection of isomorphism of chains. Their method is reported to give the information on mobility without extra computational effort.

Tischler et al. [32] introduced the notion of variety of a kinematic chain. A kinematic chain with f degrees of freedom is said to be of variety, V, if it does not contain any loop, or subsets of loops, with degrees of freedom less than $f-V$, but contains at least a loop, or a subset of loops, that has $f-V$ degrees of freedom. As such Tischler et al. formulated the concepts of total, partial and fractionated mobility in terms of the variety of a chain. The previous three works [47, 48, 32] are based on the loop analysis which is valid only in the case when the graph of the chain is planar.

Belfiore [7] reviewed the concept of planarity and pointed out that the lack of a correct and unambiguous meaning of the concept of a planar kinematic chain has caused serious misunderstandings. He suggested the usage of the adjective 'plane' when the plane motion is involved and the adjective 'planar' to mean embedability of a structure on a two dimensional plane. This usage conflicts with the usage of the adjective 'plane' in graph theory, where it means a particular embedding of a planar graph on plane. Since a planar graph can have multiple embeddings in a plane, a plane graph is one such embedding.

Interestingly, in 1996 Lee and Yoon [9] developed an algorithm for identifying the mobility type of a planar mechanism that is valid even when the graph of the mechanism is non-planar. Their algorithm first reduces the given chain by removing all pendant links and binary strings of length > 1 until only the isolated binary vertices remain. The same procedure is applied to all the subchains obtained by deleting the isolated binary vertices. To date, Lee and Yoon's is the only algorithm which can identify the mobility type of kinematic chains with both planar and non-planar graphs. While Lee and Yoon provide a working approach, the underlying mathematical reasoning for the algorithm's validity was not published in their work [9], limiting the credibility of the algorithm. The missing reasoning will be presented in Chapter 7, leaving no doubt as to the correctness of the steps of the algorithm.

3.3.1 Structural Isomorphism

One of the most important and extremely difficult problems in structural studies of kinematic chains is to check if two given kinematic chains are isomorphic. Two kinematic chains K_1 and K_2 are said to be isomorphic if there exists a one-to-one correspondence between the links of K_1 and K_2 such that a pair of links of K_1 are jointed if and only if the corresponding pair of links of K_2 are jointed. A function defined on a kinematic chain is called an index of isomorphism if any given pair of kinematic chains are isomorphic if and only if the corresponding values of the function are identical. In literature many attempts have been made to find an accurate and computationally efficient test for isomorphism. The methods for isomorphism detection can be broadly classified into spectral methods and canonical code-based methods.

Spectral Methods

The eigenvalues of a matrix are called its spectrum. Spectral methods use the properties that are a function of the eigenvalues of graph matrices of a kinematic chain to identify isomorphism. These properties include the characteristic polynomial, eigenvalues and the eigenvectors. Since the algorithms for finding spectral properties can be solved in polynomial-time, finding a spectral invariant which distinguishes non-isomorphs would amount to finding a polynomial-time algorithm for the isomorphism problem.

Uicker and Raicu [49] first used the characteristic polynomial of the adjacency matrix of a kinematic chain to distinguish non-isomorphic kinematic chains. Yan and Hwang [50] defined a new matrix called the structural matrix of a kinematic chain with n links and j joints as an $(n+j) \times (n+j)$ matrix and proposed to use the characteristic polynomial of this matrix as an index for isomorphism. Mruthyunjaya [51, 52, 53] used the characteristic polynomial of the adjacency matrix as the index for isomorphism in structural synthesis of 10 link, 1 degrees of freedom kinematic chains. He synthesized 229 chains instead of 230 as determined previously in literature. This finding resulted in the discovery of a pair of non-isomorphic kinematic chains with the same characteristic polynomial of the adjacency matrix [54, 55].

Discovery of counterexamples for the existing characteristic polynomial based indices motivated the creation of several new matrices whose characteristic polynomial could be used as an index for isomorphism. Mruthyunjaya and Balasubramanian [54] proposed a vertex-vertex degree matrix whose ij^{th} entry is the sum of degrees of links i and j if i and j are adjacent and is equal to 1 otherwise. The characteristic polynomial of this matrix successfully identified all the 10-link kinematic chains with up to 3 degrees of freedom. Dubey and Rao [56] defined a distance matrix for a kinematic chain whose ij^{th} entry corresponds to the path distance between the links i and j and zero if $i = j$.

Chang et al. [11] proposed a method already used in graph theory based on eigenvalues and eigenvectors to identify isomorphism of kinematic chains. The method works only for kinematic chains whose eigenvectors corresponding to the simple eigenvalues have sufficiently distinct coordinates. The proof given in their paper is incorrect. It is invalid for the case when the eigenvalues are repeated. He et al. [12] independently proposed a similar method for identifying isomorphism. Even though He et al. consider the case of repeated eigenvalues, testing the corresponding eigenvectors for equivalence can take exponential-time in the worst case and some simplification is needed. More recently, Cubillo and Wan [13] corrected some of the errors in the work by Chang et al. [11] and proposed results similar to He et al. [12] on using the eigenvector approach for isomorphism detection. Even when considered together these works do not clearly specify the inadequacies and the possible modes of failure of the eigenvector approach. These are described in Chapter 5.

Canonical Code-based Methods

In canonical code-based methods a kinematic chain is assigned a unique code. Among all the codes of the kinematic chains which are mutually isomorphic a unique code, called the canonical code, is chosen. Hence checking two given kinematic chains for isomorphism reduces to checking the corresponding canonical codes for equality and the canonical code can be an index of isomorphism. In most of the cases the codes will have an ordering which makes it easier to check for inequality.

Ambekar and Agarwal [57, 58] adopted an approach from graph theory to obtain a canonical code for kinematic chains called the MAX code. The 0-1 sequence obtained from the adjacency matrix of a kinematic chain by reading the upper triangular part row by row from left to right and from top to bottom can be considered its adjacency code. The MAX code of a kinematic chain can be defined as the code which is maximum among all the adjacency codes of kinematic chains obtained by relabelling the links. In the worst case it takes $n!$ computations to obtain a canonical code.

Tang and Lui [59] suggested the use of the degree code, which is the increasing sequence of the degrees of the links of kinematic chain together with the code which is maximum among all the adjacency codes of kinematic chains obtained by relabelling the links only with labels of the same degree [1]. If the degree codes are identical then this is similar to calculating the MAX code. Several others, in particular Kim and Kwak [60] and Shin and Krishnamurthy [61], extended the above approaches to obtain more computationally efficient codes.

Rao [62, 63] introduced the concept of Hamming distance to the structural studies of kinematic chains. The row of the adjacency matrix corresponding to a link is called the Hamming code of that link. The Hamming distance between two links is defined to be number of places where the Hamming codes for the two links differ. The Hamming matrix is the matrix of the same size as the adjacency matrix when the ij^{th} entry corresponds to the Hamming distance between the links i and j. Then the link Hamming number is the sum of the corresponding row of the Hamming matrix. Similarly the chain Hamming number is the sum of all the link Hamming numbers. Let the number of links of a chain be n. Rao and Varadaraju [63] defines the link Hamming string as the concatenation of the link Hamming number with the frequency of the occurrence of all the integers from n down to zero. The chain Hamming string is then defined as the concatenation of the chain Hamming value and all the link Hamming strings arranged in decreasing order. This chain Hamming string is proposed as a canonical code for testing isomorphisms. However, there are examples of non-isomorphic chains with identical chain Hamming strings [2].

There are many other codes for the detection of isomorphism. Of these, codes by Quist and Soni [64] and Rao and Rao [65] use loop based approaches that will be mostly applicable to chains with planar graphs. Other types of codes include linkage path codes by Yan and Hwang [66] and distance matrix based codes of Yadev et al [67].

3.3.2 Degeneracy Testing

The problem of identification of degenerate kinematic chains can be considered as part of structural analysis owing to its similarity to other problems. Degeneracy testing is most often performed during structural synthesis where one of the important steps is to eliminate the degenerate kinematic chains. Identification of degenerate kinematic chains involves verifying each subchain for rigidity. If one proceeds by brute force, 2^n calculations are needed where n is the number of links. However, using the structural results on planar kinematic chains, one can reduce the computation in most of the cases.

Hwang and Hwang [35] proved a set of theorems on structural results of kinematic chains, and used them to develop an algorithm for detection of basic rigid subchains in a kinematic chain. However, a step in their algorithm needs modification as it contains an incorrect usage of their theorems. The results of Hwang and Hwang's work are reviewed in detail later in this work (Chapter 7). Tuttle [10], generalizing some of the Hwang and Hwang's [22] theorems, reduced the problem of identifying degenerate kinematic chains with 10 or less links to that of detecting the basic rigid subchains of 7 links or less. Patterns of the basic rigid chains of 7 links or less were obtained and these patterns were used in the detection of basic rigid subchains. Tuttle's method will not be practical to apply to kinematic chains with greater than 10 links as the number of such patterns increases exponentially.

Lee and Yoon [8] independently developed an algorithm for degeneracy testing similar to Hwang and Hwang [35]. Lee and Yoon's work on degeneracy testing was done prior to their work on the identification of the mobility type of a planar mechanism. Lee and Yoon's algorithm for degeneracy testing performs reduction of binary chains recursively, similar to their algorithm on identification of mobility type [9]. To date, this is the most efficient algorithm for checking degeneracy. One drawback of this algorithm (and of their algorithm for mobility type identification) is that it is applicable only for planar mechanisms with revolute joints.

Hsu and Wu [68] developed an algorithm for the detection of a rigid structure in planetary gear trains. Using the notion of a fundamental circuit of the graph of planetary gear trains, they list the vertex sets formed by a collection of the fundamental circuits. A gear train was proposed to be degenerate if there exists a vertex set formed by k fundamental circuits with $k + 1$ vertices. The existence of fundamental circuits can be guaranteed only if the graph of the gear train is a planar graph. Hence this algorithm also has limited validity.

Chapter 4

Structural Synthesis - Adapting a McKay-Type Algorithm

4.1 Introduction

Researchers in the mechanisms community follow different approaches to efficiently enumerate the non-isomorphic kinematic chains during structural synthesis. But most of the existing methods are not computationally efficient hence only generation of non-isomorphic kinematic chains with a limited number of links was possible. There are also many discrepancies in the results obtained by researchers.

Synthesis of kinematic chains can be viewed as the enumeration of a certain class of graphs. Very efficient algorithms, using group theoretic techniques, exist for exhaustive isomorph-free generation of certain classes of combinatorial objects, which either eliminate or restrict the explicit isomorphism detection. An algorithm belonging to one particular class called McKay-type, in combination with an efficient degeneracy testing algorithm, is used for the synthesis of planar mechanisms. This chapter provides the details of this approach. The results from the literature are reexamined and the discrepancies are reconciled.

4.2 Background on Generation of Non-Isomorphic Combinatorial Structures

The difficulty of the exhaustive generation of non-isomorphic combinatorial objects or structures is characterized by the difficulty of the corresponding isomorphism problem. Hereinafter, ‘generation’ of objects will be assumed to mean generation of non-isomorphic objects. Some easier generation problems include generation of subsets, partitions and trees, but generation of graphs is difficult as the graph isomorphism problem is extremely challenging. Generation of kinematic chains is also difficult because of the corresponding isomorphism problem.

The most efficient generation algorithms either completely eliminate the explicit isomorph detection or restrict it to a small subset. Any generation algorithm generates a single representative from the set or class of all the objects (called the labelled objects) isomorphic to it. This set is called the isomorphism class of the representative. In terms of group theory, a set of combinatorial objects corresponds to a set along with an action of a group. The isomorphism classes would correspond to the orbits of the set under the group action, and the generation of non-isomorphic objects would correspond to generating an orbit representative for each of the orbits. The relevant group theoretic concepts are presented in Appendix A.2.

The generation algorithms, following the approach of McKay [69] and the presentation of Brinkmann [70], can be classified into three broad classes, namely, Homomorphism principle-type, Read/Faradev-type and McKay-type algorithms.

4.2.1 Homomorphism Principle-Type Algorithms

Using the Homomorphism principle method, the isomorphism class representatives are generated as follows [70]:

Step 1: A coarser structure is first identified such that any isomorphism between two structures induces an isomorphism on the corresponding coarser structures.

Step 2: All the non-isomorphic coarser structures are generated.

Step 3: All the finer structures of each of the non-isomorphic coarser structures are generated.

Example 4.1 (Colored Graph Enumeration)**.** A k-colored graph on n-vertices can be represented by (G, π) where G is the graph and the partition $\pi = (V_1, V_2, \ldots, V_k)$ is a k-tuple of non-empty disjoint subsets of V(G). An isomorphism between two r-colored graphs, (G_1, π_1) and (G_2, π_2) induces an isomorphism betweeen G_1 and G_2. Hence, the enumeration of the non-isomorphic k-colored graphs on n-vertices can be done in two steps as follows:

1. Enumerate all the non-isomorphic graphs on n vertices.
2. Enumerate all the non-isomorphic k-colored graphs for each of the graphs generated in the previous step.

A similar method can be used to enumerate distinct mechanisms and mechanisms with different joints.

Example 4.2 (Kinematic Chain Enumeration)**.** Any basic kinematic chain can be represented by Franke's (or contracted graph) notation as discussed earlier. Any isomorphism between two kinematic chains induces an isomorphism between the corresponding contracted graphs. Here the contracted graph will be a multi-graph. Any kinematic chain (in fact any simple graph) can be considered as a colored multi-graph. Hence the enumeration of the non-isomorphic kinematic chains with n-links and f-DOF can be done in two steps as follows:

1. Enumerate all the non-isomorphic contracted graphs of kinematic chains with n-links and f-DOF.
2. Enumerate all the non-isomorphic kinematic chains for each contracted graph generated in the previous step.

This approach or a slight variation of it has been used in almost all the kinematic synthesis algorithms which use the link assortment approach [17, 10, 1]. The main reason for high computational speed achieved by Tuttle [10] is the removal of all the isomorphic copies of the contracted graphs.

The main advantages of Homomorphism Principle-type algorithms are:

(a) The enumeration of complicated objects is split into enumeration of simpler objects. Existing efficient methods for the enumeration of these simpler objects can be used.

(b) Canonical representatives need not be calculated.

The main disadvantages of Homomorphism Principle-type algorithms are:

(a) It is not always possible to find a coarser structure.

(b) It is necessary to use other methods to enumerate the structures which cannot be coarsened further.

(c) Disk space is required to store intermediate objects.

4.2.2 Read/Faradzev-Type Algorithms

The Read/Faradzev method is an extension of the method of generating the canonical representatives of each isomorphism class. The following are the steps in the generation of a canonical representative for each isomorphism class:

Step 1: Each labelled object is assigned a unique code.

Step 2: Among all the codes of the elements of an isomorphism class a unique code, called the canonical code, is chosen. This canonical code is usually a maximal element in that class under some ordering.

Step 3: A generated structure is accepted only if its assigned code is a canonical code.

In the case of the Read/Faradzev-type enumeration algorithms the canonical code is chosen so that it imposes restrictions on the sub-structure of the canonical object. Read/Faradzev-type enumeration algorithms have been used to implement fast generators of cubic graphs and regular graphs [69].

Example 4.3 (Graph Enumeration)**.** Let G be a labelled graph with vertices v_α and suppose there exists a total ordering on $\{v_\alpha\}$. Then, the i^{th} row (and column) of the adjacency matrix of G corresponds to the i^{th} element in the ascending order of $\{v_\alpha\}$.

The 0-1 sequence obtained from the adjacency matrix of graph G by reading the upper triangular part row by row from left to right and from top to bottom can be considered as a code of G. The original graph can be reconstructed from this code in the obvious way. The canonical code of G can be defined as the code which is maximal among all the codes of the isomorphism class of G under the lexicographic ordering.

Let, G be a graph with vertices $V = \{v_1 \ldots v_n\}$ and let G_i be $G \setminus v_i$. If the code of G_i is not maximal then there exists a graph $G_i^{'}$ obtained by a permutation of the vertices of G_i such that the code of $G_i^{'}$ is greater than the code of G_i. This implies that there exists a $G^{'}$ belonging to the isomorphism class of G such that the code of $G^{'}$ is greater than the code of G. So if the code of G is canonical then the codes of G_i are all canonical. Hence this canonical code imposes restrictions on the sub-graphs from which it can be constructed.

Using this code for a labelled graph, all non-isomorphic graphs can be generated recursively in the Read/Faradzev sense by starting from a single vertex and adding a new vertex only to a graph with a canonical code.

Example 4.4 (Tree Enumeration)**.** Any labelled tree T with vertices $V = \{v_1 \ldots v_n\}$ is in one-one correspondence with some $n-2$ tuple of integers in $\{1, \ldots, n\}$ [71]. Such a correspondence can be obtained for instance through the Prufer code. Similar to the last example, the canonical code of T can be defined as the $n-2$ tuple which is maximal among all the tuples of the isomorphism class of T under lexicographic ordering.

Now, since every $n-2$ tuple represents a labelled tree a restriction can be imposed to check if a tree's code is canonical. Let the i^{th} element of the $n-2$ tuple t be $t(i)$. If $t(i) < (n-i-1)$ then we can permute the vertices of the tree corresponding to t to obtain an $n-2$ tuple, $t^{'}$, such that the $t^{'}(i) \geq n-i-1$. In fact, if t is a canonical code, $t(1) = n$ and for all $i > 1$, $t(i) \geq (min\{t(j), j < i\} - 1)$. If this criterion is not satisfied the code need not be further checked for canonicity and hence the corresponding tree can be rejected.

The main advantages of Read/Faradzev-type algorithms and any canonical representative-type algorithms, are [70]:

(a) Only the canonical code needs to be computed.

(b) Disk space is not required to store any intermediate objects.

However, identifying an early bounding criterion is not always possible as it depends on the construction process. Other disadvantages of any canonical representative-type algorithms are:

(a) The canonical form may be hard to compute.

(b) Construction process should be chosen to generate a structure corresponding to a canonical code.

4.2.3 McKay-Type Algorithms

Using McKay's method the isomorphism class representatives are generated by canonical construction path rather than canonical representation. Larger objects are constructed from smaller objects recursively by well-defined operations which eliminate isomorphs at each step. Every structure will be assigned a unique parent from which it must be generated. A structure is accepted if and only if it is generated from its parent. A structure is generated only once from its parent.

This work will use a McKay-type algorithm for the synthesis of kinematic chains. Therefore, the theoretical model for McKay-type algorithms is presented in greater detail below. This description adheres closely to the notation and presentation followed by McKay [69].

Let A be a group acting on a set $\mathcal{L}$. The elements of $\mathcal{L}$ are called the labelled objects, and the orbits of $\mathcal{L}$ under the action of A are called the unlabelled objects. The set of unlabelled objects is denoted by $\mathcal{U}$. Each labelled object $X \in \mathcal{L}$ has an order or size $o(X)$, which is constant on each orbit of unlabelled objects $\mathcal{L}$. Hence, $o(S)$ can be defined for $S \in \mathcal{U}$.

In the case of kinematic chain synthesis, $\mathcal{L}$ will be the set of all labelled kinematic chains and $\mathcal{U}$ will be a set of non-isomorphic chains. $o(X)$ (where $X \in \mathcal{L}$) can be defined to be the number of links of the chain X. The group A acting on labelled graphs can be taken to be $\prod S_i$, where S_i is the symmetric group on i elements.

Each labelled object $X \in \mathcal{L}$ is associated with a finite set $L(X)$ of lower objects and a finite set $U(X)$ of upper objects. A lower object contains the information needed to go backwards one step in the construction path. Similarly an upper object contains the information needed to go forward one step. Let $\check{\mathcal{L}} = \bigcup_{X \in \mathcal{L}} L(X)$ and $\hat{\mathcal{L}} = \bigcup_{X \in \mathcal{L}} U(X)$.

In the case of kinematic chains, $L(X)$ can be defined as a pair (X, v), where

X is a chain and v a link of X, and the obvious way to go backward is by removing v from X. So, the set $L(X)$ contains all such pairs, except when X has size 1 as in that case going backward is not possible. Similarly, for chains, $U(X)$ can be defined to be the set of all pairs (X, W), where W is a set of links of X that when joined to a new link will not form a degenerate structure. This gives a way to go one step forward.

The lower and upper objects are related by means of a binary relation $R \subseteq \check{\mathcal{L}} \times \hat{\mathcal{L}}$. Let $f_R : \check{\mathcal{L}} \longrightarrow 2^{\hat{\mathcal{L}}}$ and $f_R^{'} : \hat{\mathcal{L}} \longrightarrow 2^{\check{\mathcal{L}}}$ be defined as

$$f_R(\check{Y}) = \{\hat{X} \in \hat{\mathcal{L}} | (\check{Y}, \hat{X}) \in R\}$$

$$f_R^{'}(\hat{X}) = \{\check{Y} \in \check{\mathcal{L}} | (\check{Y}, \hat{X}) \in R\}$$

The group A is assumed to act on $\mathcal{L} \cup \check{\mathcal{L}} \cup \hat{\mathcal{L}}$ so that certain conditions are satisfied.

The relation between the upper and lower objects is straightforward in the case of kinematic chains: $(X, v) \in L(X)$ is related to $(X - v, W)$, where W is the set of links adjacent to v in X. In fact, $\{(X - v, W)^a | a \in A\}$ are all taken to be related to (X, v), which means the labelling is disregarded.

An unlabelled object $S \in \mathcal{U}$ is called irreducible if $L(X) = \emptyset$ for each $X \in S$, otherwise it is called reducible. The set of reducible labelled objects is denoted by $\mathcal{U}_1$. In case of kinematic chains $\mathcal{U}_1$ corresponds to all chains of size > 1.

There exists a function $m : \mathcal{L} \longrightarrow 2^{\check{\mathcal{L}}}$ satisfying the following conditions.

1. If $L(X) = \emptyset$, then $m(X) = \emptyset$.
2. If $L(X) \neq \emptyset$, then $m(X)$ is an orbit of $\mathrm{Aut}(X) = A$ on $L(X)$.

3. Each $X \in \mathcal{L}$ and $a \in A$ we have $m(X^a) = m(X)^a$.

This means that for a labelled object, X, $m(X)$ gives a *canonical* way of obtaining X from an unlabelled object of lower order. One way to construct $m(X)$ for kinematic chains would be to consider X as a graph of size n, and obtain a canonical representation of X as described in Example 4.3. Vertex $v^{'}$ corresponding to label 1 is selected. Then $m(X)$ can be defined to be $\{(X, v)^a | a \in \mathrm{Aut}(X)\}$.

The structure imposed on $\mathcal{L}$, $\check{\mathcal{L}}$ and $\hat{\mathcal{L}}$ and the function m ensure existence of a function p on reducible unlabelled objects called the parent function. For each $S \in \mathcal{U}_1$, $p(S)$ gives a unique parent in $\mathcal{U}$ from which S can be obtained. This gives rise to a structure of disjoint rooted trees on $\mathcal{U}$. These trees are traversed using an algorithm similar to a depth-first search algorithm. The algorithm shown below is a McKay-type traversal algorithm.

It should be pointed out that the Melbourne method used by Tischler et al. [31, 32] resembles the McKay-type generation algorithms. But their algorithm is reported to generate a few isomorphs unlike McKay-type algorithms.

The main advantages of McKay-type algorithms, are [70, 69]:

(a) Disk space is not required to store any intermediate objects.

(b) The canonical code used to assign a unique parent need not be compatible with the construction process.

(c) The tree structure on unlabelled objects enables parallelization of the code.

Algorithm 4.1 McKay-type algorithm

```
procedure SCAN($X$: labelled object, $n$: integer)
    for all orbits $A$ of the action of $\mathrm{Aut}(X)$ on $U(X)$ do
        select any $\hat{X} \in A$
        comment: Augment $X$ to obtain a child
        if $f^{'}(\hat{X}) \neq \emptyset$ then
            select any $\check{Y} \in f^{'}(\hat{X})$, and let $\check{Y} \in L(Y)$
            comment: $Y$ is an augmented object of $X$
            if $o(Y) \leq n$ and $\check{Y} \in m(Y)$ then
            comment: Check if $Y$ is a valid child of $X$
                SCAN($Y$, $n$)
            end if
        end if
    end for
end procedure
```

4.3 Approach

The discrepancies in the reported results on structural synthesis in the literature have been either due to invalid isomorphism checking or invalid degeneracy testing. Tuttle [10], using group theoretic techniques was able to find an efficient method for synthesis of kinematic chains. The success of Tuttle's application of group theoretic methods motivated the use of the efficient exhaustive isomorph-free generation algorithms from the field of graph theory. These algorithms apply group theoretic techniques to minimize the explicit isomorphism checking. Enormous gains in computational efficiency can be obtained by the use of group theoretic methods in generation of combinatorial structures, which include kinematic mechanisms. Many such exhaustive isomorph-free generation algorithms have been developed in the field of graph theory. These are being applied to fields like chemistry to generate non-isomorphic molecular structures. These exhaustive isomorph-free generation algorithms have never been used explicitly for the synthesis of kinematic mechanisms. This work adapts one particular class of exhaustive isomorph-free generation algorithms called the McKay-type algorithms.

The structural synthesis of basic kinematic chains is the generation of a certain subclass of simple graphs. This has been achieved by using a McKay-type algorithm in combination with an efficient degeneracy testing algorithms. Lee and Yoon's [8] algorithm for degeneracy testing and another similar degeneracy testing algorithm were used. Using this McKay-type kinematic synthesis algorithm, the non-isomorphic planar mechanisms with up to four degrees of freedom and with up to 14 links were generated. Since the number of non-isomorphic planar mechanisms with one degree of freedom and with six, eight and ten links are very well established, these numbers were used to test the basic validity of the algorithms.

4.4 Methodology and Results

Given a class of combinatorial objects $\mathcal{C}$, the extra structure required to use a McKay-type algorithm for generation of non-isomorphic objects is, the existence of the functions L, U, m and relation R on $\mathcal{C}$. Let $\mathcal{G}$ be the class of all simple graphs. Then, as described in the Section 4.2, one could define for $G \in \mathcal{G}$:

1. $L(G) = \{(G, v)\}$, for all possible $v \in V(G)$.
2. $U(G) = \{(G, W)\}$, for all possible subsets W of $V(G)$.
3. $R = \{((G, v), (G - v, W)^a) | a \in S_n\}$, for all $G \in \mathcal{G}$ where W is set of vertices joined to v in G and S_n is the symmetric group on n elements.

Let $\mathcal{C}$ be a class of graphs. $\mathcal{C}$ satisfies vertex hereditary property if for every $G \in \mathcal{C}$, every induced subgraph of G also belongs to $\mathcal{C}$. Now let $\mathcal{C}$ satisfy the vertex hereditary property then one could define for $G \in \mathcal{C}$:

1. $L(G) = \{(G, v)\}$, for all possible $v \in V(G)$.
2. $U(G) = \{(G, W)\}$, for all possible subsets W of $V(G)$ such that the graph obtained by joining the vertices of W to a new vertex belongs to $\mathcal{C}$.
3. $R = \{((G, v), (G - v, W)^a) | a \in S_n\}$, for all $G \in \mathcal{C}$ where W is set of vertices joined to v in G.

Due to the vertex hereditary property of $\mathcal{C}$, the definitions of the function L and relation R are the same as the corresponding definition for $\mathcal{G}$. Let m satisfying the properties listed in previous section be defined on $\mathcal{G}$. Now again, due to the vertex hereditary property of $\mathcal{C}$, m can be restricted to $\mathcal{C}$ like L and R.

Suppose that a McKay-type algorithm to generate non-isomorphic graphs is given with L, U and R as defined above for $\mathcal{G}$. From the discussion above, by just imposing an extra constraint on the definition of U, that same algorithm can be used to generate non-isomorphic objects of $\mathcal{C}$.

The class of graph representatives of kinematic chains satisfies the vertex hereditary property. Hence, by imposing a constraint on the definition of U in McKay's algorithm, it can be used to synthesize kinematic chains. This constraint would be an algorithm to check if a given graph represents a kinematic chain. This algorithm is the standard degeneracy testing algorithm, which checks if the DOF criterion is satisfied for all closed subchains of a kinematic chain.

McKay [69, 72] wrote an algorithm in C language for generation of graphs using the above techniques. McKay's program was adapted by using the degeneracy testing algorithms for synthesizing planar kinematic chains. This program has been implemented on an Intel Pentium III computer running the Linux operating system. The planar kinematic chains with up to 14 links and up to 4 degrees of freedom are enumerated using Lee and Yoon's [8] algorithm and also a similar reduction-based degeneracy testing algorithm detailed in Chapter 7. The latter degeneracy testing algorithm is computationally slower than Lee and Yoon's algorithm. More specifically, the latter algorithm removes all the links one at a time, after removing all the binary chains of length greater than 1. In contrast, Lee and Yoon's algorithm just removes the binary links one at a time, after removing all the binary chains of length greater than one [8]. In both cases the same results were obtained. The Tables 4.1 through 4.4 show the results. The computational efficiency of the algorithm can be seen from the fact that the 318,162 14-link and 1-DOF planar kinematic chains were generated in 37.28 seconds on a Pentium III 1.7GHz personal computer with 512MB RAM. This is approximately 13 times faster than the recent planar synthesis algorithm by Butcher and Hartman [73], even when considering the differences in CPU speeds.

Table 4.1: 1-DOF non-isomorphic kinematic chains.

Links	Chains	Confirms	Contradicts
6	2	Well established result	
8	16	Well established result	
10	230	Well established result	
12	6856	Tuttle Lee & Yoon	Hwang & Hwang (6862)
14	318,162		Tuttle (318,126) Lee & Yoon (275,255)
16	19,819,281	New result	

Table 4.2: 2-DOF non-isomorphic kinematic chains.

Links	Chains	Confirms	Contradicts
7	3	Tuttle Lee & Yoon	Hwang & Hwang (4)
9	35	Tuttle Lee & Yoon	Hwang & Hwang (40)
11	753	Tuttle Lee & Yoon	Hwang & Hwang (839)
13	27,496	Tuttle Lee & Yoon	Hwang & Hwang (29,704)
15	1,432,730		Tuttle (1,432,608)

Table 4.3: 3-DOF non-isomorphic kinematic chains.

Links	Chains	Confirms	Contradicts
8	5	Tuttle Lee & Yoon	Hwang & Hwang (7)
10	74	Tuttle Lee & Yoon	Hwang & Hwang (98)
12	1962	Tuttle Lee & Yoon	Hwang & Hwang (2442)
14	83,547	Tuttle Lee & Yoon	
16	4,805,764		Tuttle (4,805,382)

Table 4.4: 4-DOF non-isomorphic kinematic chains.

Links	Chains	Confirms	Contradicts
9	6	Lee & Yoon	Hwang & Hwang (10)
11	126	Lee & Yoon	Hwang & Hwang (189)
13	4356	Lee & Yoon	Hwang & Hwang (5951)
15	216,291	Lee & Yoon	
17	13,743,920	New result	

4.5 Discussion

It should be noted that the number of non-isomorphic chains reported by Hwang and Hwang [22] are consistently higher than those obtained here. In the case of Tuttle and Lee and Yoon the results matched for most of the cases. In the case of unmatched result their results were lower than those obtained here. The reasons for the discrepancies of the existing comprehensive results for structural synthesis by Tuttle [10], Lee and Yoon [74] and Hwang and Hwang [22], with the present work are proposed here.

All the three unmatched results of Tuttle were smaller than the results presented here. One possible explanation is that the degeneracy testing algorithm of Tuttle is eliminating some of the valid kinematic chains. The degeneracy testing algorithm of Tuttle [10] is based on eliminating certain structural patterns from the list of contracted graphs generated in the process. Also, degeneracy testing based on recognizing valid patterns of contracted graphs becomes computationally intensive as the number of links increases.

Lee and Yoon's list of results [74] was not as complete as Tuttle's [10] for larger numbers of links even though they followed Tuttle's enumeration approach. Since the present work obtained similar results using the degeneracy testing algorithm of Lee and Yoon, the only place for error, if any, would be in the generation algorithm. The generation algorithm might be eliminating some valid non-isomorphic chains. Since there is only a single unmatched result more information is needed for further analysis.

The main reason for descrepancy between the present results and Hwang and Hwang's [22] is their degeneracy testing algorithm. It can be seen that all the unmatched results of Hwang and Hwang are larger than those reported here. This might imply that the degeneracy testing algorithm sometimes includes a degenerate chain, leading to an overstatement of the number of possible kinematic chains. In fact, counterexamples to the degeneracy testing algorithm have been obtained as shown in Chapter 7.

Once the database of all the planar kinematic chains with a given number of links and degrees of freedom is obtained, the graph algorithms for finding the number of loops, degrees of vertices etc., can be used to further classify the planar mechanisms based on number and type of links, number of loops etc.

4.6 Summary

Synthesis of kinematic chains can be viewed as the enumeration of a certain class of graphs. An enormous gain in computational efficiency can be obtained by the use of group theoretic methods in generation of combinatorial structures of which kinematic mechanisms form a part. These algorithms need to be adapted for efficient synthesis of kinematic mechanisms, as shown here. An algorithm belonging to one particular class, called McKay-type, in combination with an efficient degeneracy testing algorithm is used for the synthesis of planar mechanisms. It appears that the McKay-type generation of planar kinematic chains is a very efficient and reliable method. The current generation algorithm is approximately 13 times faster than the recent planar synthesis algorithm by Butcher and Hartman [73] after considering the differences in CPU speeds. The computational speed at which the kinematic chains are generated can largely be attributed to the fact that the McKay-type algorithms greatly minimize the explicit isomorphism detection. Using the results obtained, the existing results are reexamined and the discrepancies are reconciled. Due to the broad applicability of McKay-type algorithms, they can be used in generation of other mechanical structures like epicyclic gear trains, spatial kinematic mechanisms and many types of truss structures.

Chapter 5

Efficiency and Reliability of Spectral Methods

5.1 Introduction

Reliability of an index or method for isomorphism detection of kinematic chains with a given number of links and degrees of freedom can be defined, in the usual sense. That is, defining reliability as the percentage of the number of distinct pairs of non-isomorphic chains identified by the index or the method out of the total number of distinct pairs of non-isomorphic chains. But, since the total number of pairs of non-isomorphic chains increases exponentially, reliability defined this way will be a number approaching 100% and will be impractical to use. A more practical indicator of reliability for isomorphism detection would be the number of distinct pairs of non-isomorphic chains that are *not* identified by the index or the method. Hence, the smaller the number of distinct pairs of non-isomorphic chains that are not detected by a method, the greater will be the reliability of that method, and vice versa. Given two methods or indices, A and B, we say that the reliability of A is *higher* than B, if the number of non-isomorphic chains not identified by the index A is *lower* than that of B.

The efficiency of an index or method for isomorphism detection is the computational efficiency of the index or the method. Computational efficiency can be quantified by the number of computations being performed. Similarly, given two methods or indices, A and B, we say that the efficiency of A is *higher* than B, if the number of computations taken by the index A is *lower* than that of B. There exist many methods or indices for isomorphism detection that are completely reliable, but these indices can be computationally very inefficient. One such example is the binary string obtained by concatenating the rows of the upper triangular part of an adjacency matrix. Many attempts have been made in the literature to find a reliable and computationally efficient index. The methods for isomorphism detection can be broadly classified into canonical code-based methods and spectral methods.

This chapter critically reviews the existing spectral methods in the mechanisms literature for the isomorphism detection of kinematic chains. The reliability of these methods has been established for kinematic chains with a given number of links and degrees of freedom, by determining the number of pairs of non-isomorphic chains with similar spectral properties. Kinematic chains with as many as 14 links and one, two and three degrees of freedom are considered.

5.2 Reliability of the characteristic polynomial method for isomorphism detection

For the remainder of this chapter, a kinematic chain refers to a non-degenerate planar kinematic chain, i.e., a planar kinematic chain with no rigid or immobile subchains. Given a graph G, $A(G)$ denotes the adjacency matrix of G. When distinction between $A(G)$ and G is immaterial $A(G)$ and G are used interchangeably. Graphs G_1 and G_2 are isomorphic, if and only if there exists a permutation matrix

P, such that $A(G_2) = PA(G_1)P^T$. This is usually written as

$$G_2 = PG_1P^T \tag{5.1}$$

Matrices M_1 and M_2 are called similar matrices if there exists an invertible matrix Q, such that $M_2 = QM_1Q^{-1}$. Since the determinant is a multiplicative function, we have the following proposition.

Proposition 5.1. *The characteristic polynomials of similar matrices are identical. Equivalently, the sets of eigenvalues of similar matrices are identical.*

Since $P^T = P^{-1}$ we have the following corollary.

Corollary 5.2. *If two graphs are isomorphic then their characteristic polynomials are identical.*

Corollary 5.2 gives only a necessary condition for the graphs to be isomorphic. Taken together, the eigenvalues of a matrix are called its spectrum. Hence a pair of graphs with the same adjacency characteristic polynomial are called adjacency isospectral graphs. There are several pairs of adjacency isospectral and non-isomorphic graphs. For the graphs of kinematic chains it is also well known that there exists pairs of adjacency isospectral and non-isomorphic graphs [54]. The number of such pairs of kinematic chains for a given number of links and degrees of freedom was unknown prior to this work. Table 5.1 lists the number of adjacency isospectral and non-isomorphic kinematic chains for a given number of links and degrees of freedom. The number of *pairs* of non-isomorphic chains is obtained by choosing two distinct non-isomorphic chains at a time, which is the same as the

2-combinations of the number of non-isomorphic chains. The total number of pairs of non-isomorphic chains are also listed to show the impracticality of applying the standard definition of reliability, and to show the computational overhead of the method. Table 5.1 shows that the reliability of adjacency characteristic polynomial decreases drastically as the number of links increases.

5.3 Reliability of the eigenvector method for isomorphism detection

Given a pair of graphs, if the eigenvalues of the adjacency matrix are distinct, then, most of the time, the eigenvectors can be used to detect if the graphs are non-isomorphic [11, 12, 13]. The statement and its proof are incorrect as published in [11, 12, 13], and hence is proved here rigourously. First, a preliminary lemma is proved.

Lemma 5.3. *Let A_1 and A_2 be the adjacency matrices of the isomorphic graphs G_1 and G_2 respectively. Let λ be a simple (non-repeated) eigenvalue of A_1 (and hence of A_2). Let x_1^λ and x_2^λ be a unit eigenvector corresponding to λ for A_1 and A_2, respectively. Then there exists a permutation matrix P such that $Px_1^\lambda = \pm x_2^\lambda$.*

Proof. Since G_1 is isomorphic to G_2,

$$A_2 P = P A_1 \tag{5.2}$$

for some permutation matrix P. From the hypothesis,

$$A_2 P x_1^\lambda = P A_1 x_1^\lambda = P \lambda x_1^\lambda = \lambda P x_1^\lambda \tag{5.3}$$

Table 5.1: Table of non-isomorphic adjacency isospectral kinematic chains.

DOF	Links	Number of non-isomorphic chains generated by McKay-type synthesis	Number of *pairs* of non-isomorphic chains	Number of pairs of adjacency isospectral non-isomorphic chains
1	6	2	1	0
	8	16	120	0
	10	230	26,335	2
	12	6856	23,498,940	225
	14	318,162	50,613,370,041	10,451
2	7	3	2	0
	9	35	595	0
	11	753	283,128	17
	13	27,496	378,001,260	746
3	8	5	10	0
	10	74	2701	2
	12	1962	1,923,741	30
	14	83,547	3,490,008,831	1916

Since the eigenvalue, λ, is simple, the dimension of the eigenspace corresponding to λ is exactly 1. Hence, the unit eigenvectors corresponding to λ can be chosen uniquely up to a sign. This implies that Px_1^λ is also a unit eigenvector corresponding to λ for A_2. Hence

$$Px_1^\lambda = \pm x_2^\lambda \tag{5.4}$$

□

Based on this lemma, one can derive a corollary similar to the main claim made by Chang et al. [11], He et al. [12], and Cubillo and Wan [13].

Corollary 5.4. *Let A_1 and A_2 be the adjacency matrices of the graphs G_1 and G_2, respectively, and let O_1 and O_2 be the matrices whose columns are the orthonormal eigenvectors of A_1 and A_2, respectively, arranged in the increasing order of the corresponding eigenvalues. If all the eigenvalues of G_1 and G_2 are simple, then there exists a permutation matrix P such that $PO_1 = O_2Z$, if and only if G_1 is isomorphic to G_2, where Z is a diagonal matrix with entries ± 1.*

In their claims and/or proofs Chang et al. [11] do not mention the fact that all the eigenvalues must be simple. Chang et al. [11], He et al. [12], and Cubillo and Wan [13] do not state the correct inference that $PO_1 = O_2Z$, instead they claim only that $PO_1 = O_2$. It should be noted that in their subsequent discussion of the same work He et al. [12] and Cubillo and Wan [13] mentioned this possibility.

Proof. Suppose G_1 is isomorphic to G_2 and hence $A_2 = PA_1P^T$. Since all the eigenvalues are simple, Lemma 5.3 can be applied. Hence, $Po_1^{\lambda_i} = \pm o_2^{\lambda_i}$ for all eigenvalues λ_i of A_1 (and hence of A_2) and some permutation matrix P. This can be rewritten as $PO_1 = O_2Z$.

The converse just follows by definition. $\square$

A stronger corollary whose proof is similar to the Corollary 5.4 is as follows,

Corollary 5.5. *Let A_1 and A_2 be the adjacency matrices of a pair of isomorphic graphs G_1 and G_2, respectively. Let X_1 and X_2 be the matrices whose columns are the unit eigenvectors of A_1 and A_2 corresponding to the simple eigenvalues. Let these vectors be arranged in the increasing order of the corresponding eigenvalues. Then there exists a permutation matrix P such that $PX_1 = X_2Z$ if G_1 is isomorphic to G_2, where Z is square diagonal matrix with entries ± 1.*

Even though the Corollary 5.4 appears stronger than Corollary 5.5 due the 'if and only if' condition, it is, in fact, weaker than Corollary 5.5 as the hypothesis of Corollary 5.4 is very restrictive. Hence, Corollary 5.5 should be used for isomorphism detection. It should be noted that Corollary 5.5 gives *only* a *necessary* condition for isomorphism (similar to the Corollary 5.2 involving the characteristic polynomial). This fact was not emphasized completely in the previous works [11, 13]. In fact, there exist a class of graphs called strongly regular graphs for which the eigenvalues and eigenvectors of the adjacency matrix provide no information for isomorphism detection. In other words, for the strongly regular graphs the converse of the Corollary 5.5 is not true.

A graph G is said to be k-regular if every vertex has degree exactly equal to k. A graph G on n vertices is said to be strongly regular with parameters (n, k, a, c) if it is k-regular, every pair of adjacent vertices has a common neighbors and every pair of distinct non-adjacent vertices has c common neighbors [75]. A simple example is a 4-cycle (a simple loop with 4 vertices), which is a (4,2,0,1) strongly regular graph as it is a 2-regular graph such that adjacent vertices have no common neighbor and distinct nonadjacent vertices have exactly 1 common neighbor.

For strongly regular graphs, there exist exactly 3 distinct eigenvalues. The distinct eigenvalues of a strongly regular graph, say, $\lambda_1 < \lambda_2 < \lambda_3$ can be expressed in terms of the parameters (n, k, a, c). $\lambda_3 = k$ and a corresponding eigenvector is the column vector consisting of all ones. The repeated eigenvalues λ_1 and λ_2 are the roots of $\lambda^2 - (a-c)\lambda - (k-c) = 0$. This implies that a pair of non-isomorphic strongly regular graphs with the same parameters will have the same eigenvalues and also the same unit eigenvectors (up to a sign) corresponding to the simple eigenvalue. There exist two non-isomorphic strongly regular graphs with parameters (16,6,2,2). Hence, this implies that these two non-isomorphic graphs satisfy the hypothesis of the converse of the Corollary 5.5. For further details, readers are referred to books on algebraic graph theory like Godsil and Royle [75], Biggs [76] and Cvetkovic et al. [77].

To determine if the converse of the Corollary 5.5 is false for kinematic chains, all the pairs of non-isomorphic kinematic chains with a given number of links and degrees of freedom can be checked to determine if they satisfy the hypothesis of the converse. Table 5.2 lists the number of pairs of non-isomorphic kinematic chains satisfying the hypothesis of the converse of Corollary 5.5. The pairs of adjacency isospectral and non-isomorphic chains are also listed for comparison. No pair of non-isomorphic kinematic chains with 14 links or less and with 1, 2 and 3 degrees of freedom satisfies the hypothesis of the converse of Corollary 5.5. Hence for the kinematic chains with not more than 14 links and with not more than 3 degrees of freedom, the eigenvector approach can be used to detect isomorphism with complete reliability. The results in the Table 5.2 give us confidence on the converse of Corollary 5.5. As a cautionary note the same confidence in the converse of Corollary 5.2 would remain if Table 5.1 was constructed for only up to 9 links and 1, 2 and 3 degrees of freedom. Hence, the converse of the Corollary 5.5 *need not* be true in general for all kinematic chains.

5.4 Efficiencies of the spectral methods for isomorphism

The biggest advantage of using spectral properties for isomorphism detection is that they can be computed in polynomial-time. If a characteristic polynomial could identify isomorphism with complete reliability, then it implies that a polynomial-time algorithm exists for isomorphism detection. Since eigenvalues are the roots of the characteristic polynomial, verifying the equality of characteristic polynomials for a pair of adjacency matrices is equivalent to verifying the equality of the set of eigenvalues of that pair. Computing eigenvalues involves fewer computations than computing the characteristic polynomial. Hence using eigenvalues instead of a characteristic polynomial for isomorphism detection is computationally more efficient.

Eigenvectors can be computed in polynomial-time even though just computing the eigenvalues takes fewer computations. However, in some cases, the eigenvector

Table 5.2: Table of non-isomorphic kinematic chains satisfying the hypothesis of converse of the Corollary 5.5.

DOF	Links	Non-isomorphic chains generated by McKay-type synthesis	Pairs of adjacency isospectral non-isomorphic chains	Pairs of non-isomorphic kinematic chains satisfying the converse of Corollary 5.5.
1	6	2	0	0
	8	16	0	0
	10	230	2	0
	12	6856	225	0
	14	318,162	10,451	0
2	7	3	0	0
	9	35	0	0
	11	753	17	0
	13	27,496	746	0
3	8	5	0	0
	10	74	2	0
	12	1962	30	0
	14	83,547	1916	0

approach for isomorphism detection requires exponential time since one not only needs to compute eigenvectors but also needs to verify if $PX_1 = X_2Z$ where P, X_1 and X_2 be as defined in Corollary 5.5. Consider the following example,

$$X_1 = \begin{pmatrix} a & d \\ b & d \\ c & d \\ a & e \\ b & e \\ c & e \\ a & f \\ b & f \\ c & f \end{pmatrix} \text{ and } X_2 = \begin{pmatrix} b & d \\ c & d \\ a & d \\ c & e \\ a & e \\ b & e \\ b & f \\ a & f \\ c & f \end{pmatrix} \text{ where } a,\ b \text{ and } c \text{ are some real numbers.}$$

There exists a unique permutation, namely (2, 3, 1, 6, 4, 5, 8, 7, 9), which takes each column of X_1 to X_2. But each column of X_1 can be individually permuted to the corresponding column of X_2 in $(3!)^3$ ways. Hence at least $(3!)^3$ permutations must be examined. Similarly, for a given n, one could construct two matrices X_1 and X_2 consisting of two columns of length n^2 and which would require $(n!)^n$ permutations for comparison. The underlying reason for this is that the stabilizers of each column, under the action of the symmetric group on n^2 elements, S_{n^2}, have a trivial intersection. Each of the stabilizers has $(n!)^n$ elements. More general examples can be constructed such that the individual column stabilizers have a large size and that their combined intersection is very small. However, in general, we do not have a theorem that guarantees the existence of such eigenvectors. Figure 5.1 shows the eigenvalue (Figure 5.1(a)) and eigenvector matrices (Figures 5.1(b) and 5.1(c)) of a pair of non-isomorphic kinematic chains with 12 links and 1 degrees of freedom. This is an example of a case that requires a lot of comparisons.

To improve the computational efficiency of the eigenvector approach, one could only verify if the individual eigenvectors corresponding to the simple eigenvalues can be permuted into one another. In other words, instead of verifying $PX_1 = X_2Z$ one could check if there exists permutation matrices P^i such that $P^i x_1^i = \pm x_2^i$ for each pair of i^{th} columns, (x_1^i, x_2^i), of X_1 and X_2 respectively. If $Qx = \pm y$ then x and y are said to be *similar*.

The efficiency can be further improved by only checking if the eigenvectors corresponding to the largest eigenvalue are similar. The mathematical justification for this comes from Theorem 5.6 which guarantees that the largest eigenvalue is simple. Table 5.3 lists the number of pairs of non-isomorphic kinematic chains with similar eigenvectors corresponding to the largest eigenvalue and also the number of pairs of non-isomorphic kinematic chains with similar eigenvectors corresponding to all simple eigenvalues.

Theorem 5.6 (Weak Perron-Frobenius)**.** *Suppose A is a real symmetric nonnegative $n \times n$ matrix whose underlying graph X is connected. Then, the spectral radius, $\rho(A)$, is a simple eigenvalue of A. If x is its corresponding eigenvector, then none of the entries of x are zero and all the entries have the same sign.*

For more details on Theorem 5.6, readers are referred to Godsil and Royle [75] or any other book on algebraic graph theory.

$$\begin{pmatrix} -2.7321 & -2.0000 & -1.4142 & -1.4142 & -0.7321 & +0.0000 & +0.0000 & +0.7321 & +1.4142 & +1.4142 & +2.0000 & +2.7321 \end{pmatrix}$$

(a)

$$\begin{pmatrix}
0.3140 & 0.3536 & 0.2945 & -0.1957 & -0.1625 & -0.0008 & -0.5000 & 0.1625 & -0.0073 & 0.3535 & -0.3536 & -0.3140 \\
0.3140 & 0.3536 & -0.2945 & 0.1957 & -0.1625 & 0.0008 & 0.5000 & 0.1625 & 0.0073 & -0.3535 & -0.3536 & -0.3140 \\
0.2299 & 0.0000 & -0.2768 & -0.4164 & 0.4440 & 0.0000 & 0.0000 & -0.4440 & 0.4999 & 0.0103 & 0.0000 & -0.2299 \\
0.2299 & 0.0000 & 0.2768 & 0.4164 & 0.4440 & 0.0000 & 0.0000 & -0.4440 & -0.4999 & -0.0103 & 0.0000 & -0.2299 \\
-0.2299 & 0.0000 & -0.4164 & 0.2768 & 0.4440 & 0.0000 & 0.0000 & 0.4440 & -0.0103 & 0.4999 & 0.0000 & -0.2299 \\
-0.2299 & 0.0000 & 0.4164 & -0.2768 & 0.4440 & 0.0000 & 0.0000 & 0.4440 & 0.0103 & -0.4999 & 0.0000 & -0.2299 \\
-0.3140 & 0.3536 & 0.1957 & 0.2945 & -0.1625 & -0.5000 & 0.0008 & -0.1625 & 0.3535 & 0.0073 & 0.3536 & -0.3140 \\
-0.3140 & 0.3536 & -0.1957 & -0.2945 & -0.1625 & 0.5000 & -0.0008 & -0.1625 & -0.3535 & -0.0073 & 0.3536 & -0.3140 \\
-0.3140 & -0.3536 & 0.1957 & 0.2945 & -0.1625 & 0.5000 & -0.0008 & -0.1625 & 0.3535 & 0.0073 & -0.3536 & -0.3140 \\
-0.3140 & -0.3536 & -0.1957 & -0.2945 & -0.1625 & -0.5000 & 0.0008 & -0.1625 & -0.3535 & -0.0073 & -0.3536 & -0.3140 \\
0.3140 & -0.3536 & 0.2945 & -0.1957 & -0.1625 & 0.0008 & 0.5000 & 0.1625 & -0.0073 & 0.3535 & 0.3536 & -0.3140 \\
0.3140 & -0.3536 & -0.2945 & 0.1957 & -0.1625 & -0.0008 & -0.5000 & 0.1625 & 0.0073 & -0.3535 & 0.3536 & -0.3140
\end{pmatrix}$$

(b)

$$\begin{pmatrix}
0.3140 & 0.3536 & -0.3536 & 0.0014 & 0.1625 & 0.0964 & -0.4906 & 0.1625 & -0.3534 & -0.0105 & -0.3536 & -0.3140 \\
0.3140 & 0.3536 & 0.3536 & -0.0014 & 0.1625 & -0.0964 & 0.4906 & 0.1625 & 0.3534 & 0.0105 & -0.3536 & -0.3140 \\
0.3140 & -0.3536 & -0.3536 & 0.0014 & 0.1625 & -0.0964 & 0.4906 & 0.1625 & -0.3534 & -0.0105 & 0.3536 & -0.3140 \\
0.3140 & -0.3536 & 0.3536 & -0.0014 & 0.1625 & 0.0964 & -0.4906 & 0.1625 & 0.3534 & 0.0105 & 0.3536 & -0.3140 \\
0.2299 & 0.0000 & -0.0020 & -0.5000 & -0.4440 & 0.0000 & 0.0000 & -0.4440 & -0.0149 & 0.4998 & 0.0000 & -0.2299 \\
0.2299 & 0.0000 & 0.0020 & 0.5000 & -0.4440 & 0.0000 & 0.0000 & -0.4440 & 0.0149 & -0.4998 & 0.0000 & -0.2299 \\
-0.2299 & -0.3536 & 0.0000 & 0.0000 & -0.4440 & -0.4906 & -0.0964 & 0.4440 & 0.0000 & 0.0000 & -0.3536 & -0.2299 \\
-0.2299 & 0.3536 & 0.0000 & 0.0000 & -0.4440 & 0.4906 & 0.0964 & 0.4440 & 0.0000 & 0.0000 & 0.3536 & -0.2299 \\
-0.3140 & 0.0000 & 0.5014 & 0.3515 & 0.1625 & 0.0000 & 0.0000 & -0.1625 & -0.5103 & 0.3385 & 0.0000 & -0.3140 \\
-0.3140 & -0.3536 & -0.0014 & -0.3536 & 0.1625 & 0.4906 & 0.0964 & -0.1625 & 0.0105 & -0.3534 & -0.3536 & -0.3140 \\
-0.3140 & 0.0000 & -0.4986 & 0.3556 & 0.1625 & 0.0000 & 0.0000 & -0.1625 & 0.4892 & 0.3683 & 0.0000 & -0.3140 \\
-0.3140 & 0.3536 & -0.0014 & -0.3536 & 0.1625 & -0.4906 & -0.0964 & -0.1625 & 0.0105 & -0.3534 & 0.3536 & -0.3140
\end{pmatrix}$$

(c)

Figure 5.1: Eigenvalues (a) and eigenvector matrices (b) and (c) of a pair of non-isomorphic 12 link and 1 degrees of freedom obtained after Step 3

Table 5.3: Table of non-isomorphic kinematic chains with similar eigenvectors corresponding to the largest eigenvalue.

DOF	Links	Non-isomorphic chains	Pairs of non-isomorphic kinematic chains with similar eigenvectors corresponding to the largest eigenvalue	Pairs of non-isomorphic kinematic chains with similar eigenvectors corresponding to all simple eigenvalues
1	6	2	0	0
	8	16	0	0
	10	230	0	0
	12	6856	4	4
	14	318,162	41	34
2	7	3	0	0
	9	35	0	0
	11	753	1	1
	13	27,496	6	6
3	8	5	0	0
	10	74	0	0
	12	1962	0	0
	14	83,547	11	9

5.5 Implementation

For this study, all the non-isomorphic kinematic chains for a given number of links and degrees of freedom were synthesized using a McKay-type exhaustive isomorph-free generation algorithm in conjunction with Lee and Yoon's degeneracy testing algorithm. Using McKay's method the isomorphism class representatives are generated by canonical construction path where the larger kinematic chains are constructed from smaller kinematic chains recursively by well-defined operations which eliminate isomorphs at each step. More specifically, every kinematic chain will be assigned a unique parent from which it must be generated and a kinematic chain is accepted if and only if it is generated from its parent. A kinematic chain is generated only once from its parent which ensures that only non-isomorphic kinematic chains are generated in this procedure. The basic algorithm for synthesis of planar kinematic chains is presented in Chapter 4.

Databases of the adjacency matrices of all the possible non-isomorphic chains consisting of a given number of links and degrees of freedom for 6 to 14 links and 1 to 3 degrees of freedom were created. To establish the reliability of the characteristic polynomial method, all possible pairs of non-isomorphic kinematic chains from each database were tested to verify if they have the same set of eigenvalues. Algorithm 5.1 is an implementation of isomorphism detection using the modified eigenvector approach. The reliability of the modified eigenvector approach is established by checking if all possible pairs of non-isomorphic kinematic chains are detected by the Algorithm 5.1.

Algorithm 5.1 Modified eigenvector isomorphism detection algorithm

```
procedure isIsoEigVec($A_1$: adjacency matrix, $A_2$: adjacency matrix)
    if the sets of eigenvalues of $A_1$ and $A_2$ are equal then
        select the eigenvectors corresponding to the largest eigenvalue
        if the corresponding eigenvectors are similar then
            if all the eigenvectors corresponding to simple eigenvalues are similar
                then
                if $PX_1 = X_2Z$ for some permutation matrix $P$ then
                comment: $X_1$, $X_2$ and $Z$ are as defined in Corollary 5.5
                    calculate all possible $P$ such that $PX_1 = X_2Z$
                    if $PA_1P^T = A_2$ for some $P$ calculated above then return true
                    end if
                end if
            end if
        end if
    end ifreturn false
end procedure
```

Verifying two eigenvectors x, y for similarity, i.e., verifying the existence of a permutation matrix Q such that $Qx = \pm y$, is done efficiently by using a sort function. First, x, y, $-y$ are sorted and then the norms of $x_s - y_s$ and $x_s - (-y)_s$ (where subscript s indicates that the vectors are sorted) are tested to see if they are simultaneously zero. Verifying the existence of a permutation matrix P such that $PX_1 = X_2Z$ (where X_1, X_2 and Z are defined as in Corollary 5.5) is done as follows:

1. A column vector of X_1 with maximum number of distinct coordinates is first selected and the corresponding column vector of X_2 is also picked.

2. These column vectors are verified for similarity and a corresponding permutation, if exists, is calculated as described earlier.

3. The stabilizer of selected column vector of X_1 under the action of S_n is calculated. This is used to generate all the possible permutations which make the corresponding selected columns similar.

4. Each of the permutations generated are acted on the remaining columns to verify the feasibility of the permutation.

All the algorithms were programmed in MATLAB and were implemented on personal computers with Intel Pentium 4 processor and 1GB of RAM.

5.6 Discussion

The results in Table 5.1 show that the reliability of the adjacency characteristic polynomial for isomorphism detection decreases as the number of links increases. It should be noted that this is the most computationally time consuming step even for the eigenvector approach, as all the possible pairs of non-isomorphic chains are tested for equality of the eigenvalues. This work shows that the eigenvector approach is completely reliable (Table 5.2) for kinematic chain with up to 14 links and having 1, 2 and 3 degrees of freedom but is computationally less efficient in certain cases. Furthermore, one could conclude from Tables 5.2, and 5.3 that Algorithm 5.2, which is more efficient than Algorithm 5.1, detects isomorphism with complete reliability for kinematic chains with up to 14 links.

Algorithm 5.2 Modified eigenvector isomorphism detection algorithm with up to 14 links

```
procedure isIsoEigVec($A_1$: adjacency matrix, $A_2$: adjacency matrix)
    if the sets of eigenvalues of $A_1$ and $A_2$ are equal then
        select the eigenvectors corresponding to the largest eigenvalue
        if the corresponding eigenvectors are similar then
            if $PX_1 = X_2Z$ for some permutation matrix $P$ then
            comment: $X_1$, $X_2$ and $Z$ are as defined in Corollary 5.5 return true
            end if
        end if
    end ifreturn false
end procedure
```

It should be noted that the computationally expensive step of Algorithms 5.1 and 5.2 is finding the permutation matrix P such that $PX_1 = X_2Z$. In fact, this step can take exponential time to execute. An example is the eigenvector matrices of a pair of non-isomorphic 12 link and 1 degrees of freedom obtained after Step 3 shown in Figure 5.1(b) and 5.1(c).

5.7 Summary

There exist many techniques to detect if a pair of kinematic chains are isomorphic. However, most of these techniques are either computationally inefficient or unreliable. In particular, spectral methods such as the traditional characteristic polynomial method and the more recently proposed eigenvector approach, have attracted a lot of attention as there exist polynomial-time algorithms to compute the spectral properties of a matrix. The reliability of the characteristic polynomial of adjacency matrix is established and, as expected, the reliability decreased as number of links increased. Unlike the characteristic polynomial approach, the eigenvector approach can take exponential-time in the worst case. The eigenvector approach can be refined for computational efficiency by first verifying if the eigenvectors corresponding to the largest eigenvalue are similar. The eigenvector approach has detected all non-isomorphic chains with up to 14 links and 1, 2 and 3 degrees of freedom with complete reliability. It remains to be established if the eigenvector approach fails to identify a pair of non-isomorphic kinematic chains, as in the case of simple graphs.

Chapter 6

Laplace and Extended Adjacency Matrices for Isomorphism Detection

6.1 Introduction

In the graph theory community, researchers have used the spectra of the Laplace matrix of a graph for isomorphism detection. This chapter establishes the reliability of using the eigenvalues of the Laplace matrix of a kinematic chain for isomorphism detection. Novel matrices, called extended adjacency matrices, similar to the Laplace matrices are developed in this chapter for isomorphism detection. The reliability of the eigenvalues of the extended adjacency matrices of a kinematic chain for isomorphism detection is also established in this chapter.

6.2 Characteristic polynomial methods

As in the previous chapter, given a graph, G, $A(G)$ denotes the adjacency matrix of G. Graphs G_1 and G_2 are isomorphic if and only if there exists a permutation matrix P such that $A(G_2) = PA(G_1)P^T$. In the previous chapter the exact number of adjacency isospectral and non-isomorphic kinematic chains for a given number of links and degrees of freedom are tabulated.

The goal remains to find a matrix, $M(G)$, such that the a pair of graphs are isomorphic, if and only if the characteristic polynomials with respect to the matrix M of the graph are equal. A pair of graphs with the same characteristic polynomial with respect to $M(G)$ are called isospectral graphs with respect to $M(G)$ or $M(G)$-isospectral graphs. Identifying such a matrix, $M(G)$, would amount to finding a polynomial-time algorithm for the graph isomorphism problem. The graph theory community is highly skeptical about finding such a matrix for general graphs.

Polynomial-time isomorphism detection algorithms exist for certain classes of graphs with inherent structure such as, planar graphs, trees and graphs with bounded degree. Unfortunately, the graphs of planar kinematic chains do not fall into any of the classes stated above, but they are closely related to planar graphs. The graphs of planar kinematic chains have a rich recursive structure due to the constraints imposed by the degree of freedom equation. Hence one can hope to develop a polynomial-time algorithm for isomorphism detection by exploiting the recursive structure.

Since the characteristic polynomial of the adjacency matrix is not a completely reliable isomorphism index, researchers have proposed several other graph matrices. In the mechanisms community the examples include the structural matrix [50], the vertex-vertex degree matrix [54] and the distance matrix [56]. However, in the graph theory community a different matrix, called the Laplace matrix, was chosen to characterize graphs using their spectral properties [78, 79].

6.3 Laplace matrix

The Laplace matrix or the Laplacian of a graph G, $L(G)$ is defined to be $A(G) - D(G)$ where $D(G)$ is the diagonal matrix made of the degrees of G. From the definition of the Laplacian, it is easy to see that, if $A = (a_{ij})$ then L can be written as

$$L = \begin{pmatrix} -(a_{11} + \cdots + a_{1n}) & a_{12} & \cdots & a_{1n} \\ a_{21} & -(a_{21} + \cdots + a_{2n}) & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & -(a_{n1} + \cdots + a_{nn}) \end{pmatrix}$$

The following claim for the Laplace matrix holds true.

Claim 6.1. *Graphs G_1 and G_2 are isomorphic if and only if there exists a permutation matrix P such that $L(G_2) = PL(G_1)P^T$.*

Proof. G_1 and G_2 are isomorphic if an only if there exists a permutation matrix P such that $A(G_2) = PA(G_1)P^T$. Let π be the permutation corresponding to the permutation matrix P. Then

$$\begin{aligned} PL(G_1)P^T &= P(A(G_1) - D(G_1))P^T \\ &= PA(G_1)P^T - PD(G_1)P^T \\ &= A(G_2) - PD(G_1)P^T \end{aligned} \tag{6.1}$$

Let $A_1 = (a_{1,ij})$ and $A_2 = (a_{2,ij})$ then,

$$
\begin{aligned}
&PD(G_1)P^T = \\
&\quad P\begin{pmatrix} -(a_{1,11} + \cdots + a_{1,1n}) & & \\ & \ddots & \\ & & -(a_{1,n1} + \cdots + a_{1,nn}) \end{pmatrix} P^T = \\
&\quad \begin{pmatrix} -(a_{1,\pi(1)1} + \cdots + a_{1,\pi(1)n}) & & \\ & \ddots & \\ & & -(a_{1,\pi(n)1} + \cdots + a_{1,\pi(n)n}) \end{pmatrix} = \\
&\quad \begin{pmatrix} (a_{2,1\pi^{-1}(1)} + \cdots + a_{2,2\pi^{-1}(n)}) & & \\ & \ddots & \\ & & -(a_{2,n\pi^{-1}(1)} + \cdots + a_{2,n\pi^{-1}(n)}) \end{pmatrix} = \\
&\quad \begin{pmatrix} -(a_{2,11} + \cdots + a_{2,2n}) & & \\ & \ddots & \\ & & -(a_{2,n1} + \cdots + a_{2,nn}) \end{pmatrix} = \\
&\quad PD(G_2)P^T
\end{aligned}
\tag{6.2}
$$

Equations 6.1 and 6.2 combined together give the required result. □

The Laplacian was not used in mechanisms literature prior to this work for isomorphism detection. A list of all Laplacian isospectral and non-isomorphic kinematic chains for different number of links and degrees of freedom, was obtained by comparing the Laplacian eigenvalues for every possible non-isomorphic pair of chains. The results obtained are listed in Table 6.1. In this and subsequent tables the results of the reliability of the characteristic polynomial of the adjacency matrix are included for comparison.

These results infer that the Laplacian characteristic polynomial is a better index than the adjacency characteristic polynomial, in most of the cases where the number of links are large. However, the opposite was observed for most of the cases where the number of links was small.

Since the reliability of the characteristic polynomial of the Laplace matrix was comparable to that of the adjacency matrix, the combination of verifying isomorphism using both the characteristic polynomials was considered. All the pairs of non-isomorphic chains were compared to enumerate the number of pairs which were isospectral with respect to *both* the adjacency and the Laplacian matrices. The results obtained are tabulated in Table 6.2.

Results shown in Table 6.2 infer that for kinematic chains with links less than or equal to 11, the characteristic polynomials of both the adjacency and the Laplace matrix can simultaneously be used as the isomorphism index. In the case of kinematic chains with 12 or more links, both these polynomials together eliminated more than 96% of the adjacency isospectral and non-isomorphic chains. Unfortunately, these results also show that these polynomials cannot be used as a ideal isomorphism index.

6.4 Extended adjacency matrix

This section deals with the problem of finding such a matrix $M(G)$ such that a pair of kinematic chains are isomorphic, if and only if the characteristic polynomials

Table 6.1: Table of Laplacian isospectral and non-isomorphic kinematic chains.

DOF	Links	Non-isomorphic chains	Adjacency isospectral chains	Laplacian isospectral chains
1	6	2	0	0
	8	16	0	1
	10	230	2	8
	12	6,856	225	114
	14	318,162	10,451	2,324
2	7	3	0	0
	9	35	0	0
	11	753	17	6
	13	27,496	746	259
3	8	5	0	0
	10	74	2	1
	12	1,962	30	33
	14	83,547	1,916	676

Table 6.2: Table of Laplacian and adjacency isospectral, and non-isomorphic kinematic chains.

DOF	Links	Non-isomorphic chains	Adjacency isospectral chains	Adjacency and Laplacian isospectral chains
1	6	2	0	0
	8	16	0	0
	10	230	2	0
	12	6,856	225	9
	14	318,162	10,451	310
2	7	3	0	0
	9	35	0	0
	11	753	17	0
	13	27,496	746	18
3	8	5	0	0
	10	74	2	0
	12	1,962	30	0
	14	83,547	1,916	38

with respect to the matrix M are equal.

Given the graphs G_1 and G_2, the corresponding matrices $M(G_1)$ and $M(G_2)$ should satisfy the following conditions:

M1: If G_1 and G_2 are isomorphic, then $M(G_2) = PM(G_1)P^T$ for some permutation matrix P.

M2: If the characteristic polynomials of $M(G_1)$ and $M(G_2)$ are identical, then G_1 and G_2 are isomorphic.

Laplacian and adjacency matrices satisfy the first condition. The Laplacian matrix, L, by definition is $A(G) - D(G)$ and hence has $(a_{k1} + \cdots + a_{kn})$ as diagonal terms which are invariant under the action of the permutation group. If the diagonal elements of Laplace matrix are replaced by any symmetric functions, $sym(a_{k1}, \ldots, a_{kn})$, of the corresponding elements of the row, the resulting matrix will also satisfy the Condition M1.

A symmetric function on n elements is a function which is invariant under the action of the permutation group on the variables. Given variables $(u_1, \ldots, u_n) = u$, the elementary symmetric polynomials, $s_i(u)$, are defined as

$$\begin{aligned}
s_0(u) &= 0 \\
s_1(u) &= u_1 + u_2 \cdots + u_n \\
s_2(u) &= u_1u_2 + u_1u_3 + \cdots u_{n-1}u_n = \sum_{i<j} u_iu_j \\
s_3(u) &= \sum_{i<j<k} u_iu_ju_k \\
&\vdots \\
s_n(u) &= u_1u_2 \cdots u_n
\end{aligned}$$

Given a permutation π of first n natural numbers, let the action of π on u be $\pi(u) = (u_{\pi(1)}, \ldots, u_{\pi(n)})$. Then, from the definition, $s_k(\pi(u)) = s_k(u)$. Hence the elementary symmetric polynomials are invariant under the action of the permutation group. Since any symmetric polynomial can be written as a polynomial of elementary symmetric polynomials, one could define an extended adjacency matrix of a graph G for each such elementary symmetric polynomial. An extended adjacency matrix of order d of a graph G can be defined as, given adjacency matrix $A = (a_{ij})$,

$$A^{(d)} = \begin{pmatrix} s_d(-a_{11}, \ldots, -a_{1n}) & a_{12} & \cdots & a_{1n} \\ a_{21} & s_d(-a_{21}, \ldots, -a_{2n}) & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & s_(- a_{n1}, \ldots, -a_{nn}) \end{pmatrix}$$

Clearly from the definition of $A^{(d)}(G)$, $A^{(0)}(G) = A(G)$, the conventional adjacency matrix, and $A^{(1)}(G) = L(G)$, Laplace matrix. The Proposition shows that the extended adjacency matrices satisfy the Condition M1.

Proposition 6.2. *Graphs G_1 and G_2 are isomorphic if and only if there exists a permutation matrix P such that $A^{(d)}(G_2) = PA^{(d)}(G_1)P^T$.*

Proof. Proof is similar to the proof of the Claim 6.1. When all the diagonal entries in the proof of Claim 6.1 of form $-(a_{p,k1}+\cdots+a_{p,kn})$ are replaced by $s_d(a_{p,k1},\ldots,a_{p,kn})$ we obtain the proof of this proposition. □

The reliability of the characteristic polynomials of the extended adjacency matrices for isomorphism detection was computed for a given number of links and degrees of freedom. The results are listed in Table 6.3. From the results it can be seen that the reliability of the characteristic polynomials of $A^{(2)}$ and $A^{(3)}$ ranked first and second respectively, and the reliability of the characteristic polynomials of other matrices was comparable to the adjacency matrix ($A^{(0)}$). Notably, the characteristic polynomial of the $A^{(2)}$ showed unusually high reliability compared to others.

It was shown that the characteristic polynomials of the adjacency matrix ($A^{(0)}$)and that of the Laplace matrix ($A^{(1)}$) together served as a better isomorphism index than the individual polynomials. This provided motivation to check if the characteristic polynomials of all the extended adjacency matrices together can serve as a completely reliable index for isomorphism. All the pairs of non-isomorphic chains were compared to enumerate the pairs which were isospectral with respect to *all* the extended adjacency matrices. These results are listed in Table 6.4. From the results, it can be seen that the reliability of the characteristic polynomials of all the extended matrices was slightly better than those using the characteristic polynomials of adjacency and Laplace matrices. Surprisingly, the reliability of the

Table 6.3: Table of $A^{(d)}$ isospectral and non-isomorphic kinematic chains.

DOF		1				2				3		
Links		6	8	10	12	7	9	11	13	8	10	12
Non-isomorphic chains		2	16	230	6856	3	35	753	27,496	5	74	1962
Isospectral & non-isomorphic chains of matrix	$A^{(0)}$	0	0	2	225	0	0	17	746	0	2	30
	$A^{(1)}$	0	1	8	114	0	0	6	259	0	1	33
	$A^{(2)}$	0	0	0	7	0	0	0	18	0	0	0
	$A^{(3)}$	0	1	0	16	0	0	2	101	0	1	16
	$A^{(4)}$	0	0	1	109	0	0	9	459	0	1	23
	$A^{(5)}$	0	0	2	208	0	0	15	727	0	2	29
	$A^{(6)}$		0	2	224	0	0	17	746	0	2	30
	$A^{(7)}$		0	2	225		0	17	746	0	2	30
	$A^{(8)}$			2	225		0	17	746		2	30
	$A^{(9)}$			2	225			17	746		2	30
	$A^{(10)}$				225			17	746			30
	$A^{(11)}$				225				746			30
	$A^{(12)}$								746			

characteristic polynomials of all the extended matrices was exactly equal to the reliability of the characteristic polynomial of the $A^{(2)}$ matrix.

Next, all the pairs of $A^{(2)}$-isospectral and non-isomorphic kinematic chains were verified to assess if they were isospectral for all other extended adjacency matrices when number of links is 14 and 1 and 3 degrees of freedom. The results are shown in the Table 6.5. This shows that the pairs of non-isomorphic chains with up to 14 links and 1 to 3 degrees of freedom that are isospectral for all extended matrices are exactly the pairs of $A^{(2)}$-isospectral chains.

6.5 Discussion

The reliability of the characteristic polynomial of the $A^{(2)}$ matrix was a lot higher than that of the standard adjacency matrix. The $A^{(2)}$ matrix must be used for the characteristic polynomial based isomorphism detection in kinematic chains. It remains to be seen if the pairs of non-isomorphic chains that are isospectral for all extended matrices are exactly the pairs of $A^{(2)}$ isospectral chains for *any* number of links and degrees of freedom. It should be noted that similar extended matrices can be constructed for the vertex-vertex degree matrix [54] and the distance matrix [56] that were defined in the mechanisms literature. It also remains to be seen if the corresponding $A^{(2)}$ extended matrices have the same characteristics.

While analyzing the data of the Laplacian isospectral and non-isomorphic graphs it was observed that all the pairs of Laplacian isospectral and non-isomorphic graphs had the same degree sequence. The results are tabulated in Table 6.6.

Table 6.4: Table of isospectral and non-isomorphic kinematic chains for all extended matrices.

DOF	Links	Non-isomorphic chains from Chapter 5	Adjacency ($A^{(0)}$) isospectral chains	Isospectral chains for all extended matrices
1	6	2	0	0
	8	16	0	0
	10	230	2	0
	12	6,856	225	7
2	7	3	0	0
	9	35	0	0
	11	753	17	0
	13	27,496	746	18
3	8	5	0	0
	10	74	2	0
	12	1,962	30	0

Table 6.5: Table of all $A^{(d)}$ isospectral chains and $A^{(2)}$ isospectral and non-isomorphic chains.

DOF	Links	$A^{(2)}$ isospectral chains	Isospectral chains for all extended matrices
1	6	2	0
	8	0	0
	10	0	0
	12	7	7
	14	304	304
2	7	0	0
	9	0	0
	11	0	0
	13	18	18
3	8	0	0
	10	0	0
	12	0	0
	14	36	36

Table 6.6: Table of the Laplacian isospectral non-isomorphic chains with different degree sequence.

DOF	Links	Laplacian Isospectral chains	Laplacian Isospectral chains with different Degree Sequence
1	6	0	0
	8	1	0
	10	8	0
	12	114	0
2	7	0	0
	9	0	0
	11	6	0
	13	259	0
3	8	0	0
	10	1	0
	12	33	0

It is well known that the Laplacian isospectral graphs have the same number of edges because the trace of the Laplacian is the negative number of edges [76]. To verify if there exist graphs with non-isomorphic Laplacian isospectral graphs that have a different degree sequence, all non-isomorphic graphs pairs on 7 and 8 vertices were tested. The results are listed in the Table 6.7

Table 6.7: Table of the Laplacian isospectral non-isomorphic graphs with different degree sequence.

Vertices	Laplacian Isospectral graphs	Laplacian Isospectral chains with different Degree Sequence
7	74	36
8	1112	400

From the above results, it can be observed that there exist non-isomorphic graphs that are Laplacian isospectral and have different degree sequence. This phenomenon, which seems to be unique to kinematic chains, needs further research to decide if it is valid for all kinematic chains. If this phenomenon is true in general for kinematic chains, one can hope to find more structural invariants which might eventually help in isomorphism detection of kinematic chains.

6.6 Summary

Kinematic chain isomorphism detection based on the characteristic polynomial of the graph matrices would imply a polynomial-time algorithm for isomorphism detection. To date, there does not a exist a graph matrix whose characteristic polynomial detects isomorphism with complete reliability. In an attempt to find such a matrix for kinematic chains, the Laplace matrix was first tested. The reliability of the eigenvalues of Laplace matrix for isomorphism detection was similar to the eigenvalue of adjacency matrix but when both were used simultaneously there was a significant increase in reliability.

Extended adjacency matrices similar to the Laplace matrix were developed. The reliability of the eigenvalues of almost all of the extended adjacency matrices was similar to that of the standard adjacency matrix. Interestingly, the eigenvalues of $A^{(2)}$ had unusually high reliability. It was also observed that the pairs of non-isomorphic chains that are isospectral for all extended matrices, are exactly the pairs of $A^{(2)}$-isospectral chains. This makes the $A^{(2)}$ matrix the best candidate for isomorphism detection of chains using the characteristic polynomial approach and also suggests the need for more research.

Chapter 7

Reevaluation of degeneracy testing and mobility type identification algorithms

7.1 Introduction

Structural properties derived from the graph representation of planar kinematic mechanisms have enabled the development of efficient algorithms for structural analysis tasks, such as degeneracy testing, mobility type identification, and isomorphism detection. This chapter investigates and reappraises the most effective algorithms for degeneracy testing, namely those by Hwang and Hwang [35] and Lee and Yoon [8], and Lee and Yoon's [9] successful algorithm for mobility type identification. The examination of the algorithms will include a full description of underlying assumptions that are either explicitly stated in previous work or have been assumed or implied by subsequent implementations. The limitations of Hwang and Hwang's degeneracy testing algorithm are detailed and the missing mathematical justifications for Lee and Yoon's algorithms for degeneracy testing and mobility type identification are provided.

7.2 Evaluation of existing degeneracy testing algorithms

To test if a given kinematic chain G is degenerate one needs to check if $\mathrm{DOF}(G') > 0$ for all sub chains G'. If the order of G is n, 2^n calculations are needed. Instead of performing 2^n calculations, one can reduce the number of calculations significantly by performing reductions based on the following theorems providing structural results about planar kinematic chains. For the rest of chapter, the distinction between a kinematic chain and its graph representation is ignored. Hence, the terms vertices and links, and edges and joints are used interchangeably.

7.2.1 Basic degeneracy testing algorithm

A degeneracy testing algorithm based on the structural results that do not require the planarity assumption is presented in this section. The required structural results are stated as Propositions 7.1 and 7.2.

Proposition 7.1. *Let G be a kinematic chain, H be a subchain of G, $G' = G \setminus H$ and $E_{G'H} = E(G) \setminus (E(H) \cup E(G'))$ i.e., the external edges between G' and H. Then* $\mathrm{DOF}(G') = \mathrm{DOF}(G) - \mathrm{DOF}(H) + 2|E_{G'H}| - 3$.

Proof. By definition,

$$\mathrm{DOF}(G') = 3|V(G')| - 2|E(G')| - 3 \tag{7.1}$$

From the hypothesis of the proposition it follows that $|V(G')| = |V(G)| - |V(H)|$ and $|E(G')| = |E(G)| - |E(H)| - |E_{G'H}|$. Substituting these results in Equation

(7.1) and rearranging

$$\mathrm{DOF}(G^{'}) = (3|V(G)|-2|E(G)|-3)-(3|V(H)|-2|E(H)|-3)+(2|E_{G'H}|-3) \quad (7.2)$$

which proves the proposition. □

Proposition 7.2. *Let G be a kinematic chain and B be a binary chain of length greater than 1, which is not part of a triangle. Then G is degenerate if and only if $G \setminus B$ is degenerate.*

Proof. If G is non-degenerate, then by definition $G \setminus B$ is non-degenerate. Hence $G \setminus B$ degenerate implies G is degenerate. Suppose G is degenerate, then there exists a subchain H such that $\mathrm{DOF}(H) \leq 0$ and without loss of generality H can be assumed to be connected. If $H \cap B = \emptyset$ then $H \subset (G \setminus B)$ which implies $G \setminus B$ is degenerate.

If $H \cap B = B_H \neq \emptyset$, let $H^{'} = H \setminus B$ and $E_{H'B_H} = E(H) \setminus (E(H^{'}) \cup E(B_H))$. $H^{'} = \emptyset$ implies that $H = B$ and hence B is degenerate. However, a binary chain is degenerate if and only if it is a triangle and hence it can be assumed that $H^{'} \neq \emptyset$.

The binary chain B can either be a path or a cycle, but a cycle would mean G is disconnected and hence, B can only be a path. Let x_B and y_B be the vertices to which B is connected. Then, $\{x_B, y_B\} \cap H = \emptyset$ implies H is disconnected. Hence, B_H can only be of one of the following types

1. A path with exactly one end connected to H.
2. A path with both ends connected to H.
3. Union of two paths with exactly one end of each path connected to H.

By Lemma 7.1,

$$\mathrm{DOF}(H^{'}) = \mathrm{DOF}(H) - \mathrm{DOF}(B_H) + 2|E_{H'B_H}| - 3 \tag{7.3}$$

In all the described types of binary chains, B_H, either $|E_{H'B_H}| = 1$ or $|E_{H'B_H}| = 2$ along with $\mathrm{DOF}(B_H) \geq 1$ (since $|V(B_H)| \geq 2$ and the length of $B \geq 2$).

In both cases, it follows from Equation 7.3 that $\mathrm{DOF}(H) \geq \mathrm{DOF}(H^{'})$. Now $\emptyset \neq H^{'} \subset (G \setminus B)$ which implies $G \setminus B$ is degenerate. □

Using Proposition 7.2 all the binary chains of length ≥ 2 of a kinematic chain can be removed before degeneracy testing. As a consequence, the following algorithm for degeneracy testing of a kinematic chain G, by reducing all the binary chains of length ≥ 2, can be obtained. It should be noted that the proofs of the previous propositions didn't assume planarity of the graph of kinematic chain. Based on the two propositions the following degeneracy testing algorithm is proposed:

Basic degeneracy testing algorithm (Binary chain reduction):

1. Store the given kinematic chain G in a stack.
2. If stack is empty Stop procedure and G is non-degenerate; otherwise pick a chain and proceed to next step.
3. If there does not exist a binary chain of length ≥ 2 Go to Step 6; otherwise proceed to next step.
4. Remove all binary chains of length ≥ 2 and that are not triangles and Go to Step 3. (Justified by the Proposition 7.2).

5. If the number of links of the reduced kinematic chain, H, is > 3 Go to Step 6; otherwise,
    (a) If DOF(H) > 0 then Go to Step 2.
    (b) If DOF(H) ≤ 0 then G is degenerate, Stop procedure.
6. Remove one vertex at a time and store the resulting chains in the stack and Go to Step 2.

This algorithm is a good baseline for comparing other algorithms as it is based on the results that do not require planarity assumption.

7.2.2 Hwang and Hwang's degeneracy testing algorithm

Now the algorithm of Hwang and Hwang [35] is reappraised. Hwang and Hwang [35] presented the following six theorems providing some structural results for planar kinematic chains. However, three of them inherently assume that the graph of a planar kinematic chain is a planar graph. Hence, Theorems 7.3, 7.5 and 7.6 of Hwang and Hwang are applicable only to kinematic chains whose graphs are planar graphs. Theorems 7.7 and 7.8 are both crucial for justifying the steps of Hwang and Hwang's degeneracy algorithm. The proofs of these theorems contain some gaps or depend on the Theorems 7.3, 7.5 or 7.6 that are valid only for kinematic chains with planar graphs. The proofs of Propositions 7.2 and 7.10 presented in this work fill these gaps and do not rely on the planarity assumption.

Theorem 7.3 (Hwang and Hwang, 1991). *If a binary chain of length m is removed from a kinematic chain with l independent loops and f degrees of freedom, the new kinematic chain has $(l-1)$ independent loops and $(f+2-m)$ degrees of freedom.*

Theorem 7.4 (Hwang and Hwang, 1991). *A kinematic chain with n links and f degrees of freedom is a degenerate kinematic chain if it contains a binary chain of length $(f+2)$.*

Theorem 7.5 (Hwang and Hwang, 1991). *A kinematic chain with l independent loops and without binary chains of length $(f+2)$ cannot contain a basic rigid subchain with $(l-1)$ independent loops.*

Theorem 7.6 (Hwang and Hwang, 1991). *A kinematic chain with n links and f degrees of freedom and without binary chains of length $(f+2)$ cannot contain a basic rigid subchain with n_b links if $n < f + n_b + 4$* (An improved version of this theorem is proved by Tuttle [10] and it is stated as Theorem 7.9 later not to affect the flow of presentation).

Theorem 7.7 (Hwang and Hwang, 1991). *Removing any binary chain of length ≥ 2 from a kinematic chain will not affect the result of the detection of a degenerate kinematic chain.*

This theorem is exactly the same as the Proposition 7.2, but the proof of Hwang and Hwang [35] implicitly assumes that if a kinematic chain is degenerate then there exists a subchain of the given chain which is a basic rigid chain or at least a chain with zero degrees of freedom. This is not an obvious fact and is proved in the upcoming Proposition 7.10. The proof is presented later in order to preserve the flow of the section. Note that this theorem of Hwang and Hwang also depends on Theorems 7.3, 7.5 or 7.6 and that Proposition 7.2 does not depend on the planarity assumption.

The next theorem justifies the removal of the binary links (not binary chains) one at a time.

Theorem 7.8 (Hwang and Hwang, 1991)**.** *A kinematic chain with f degrees of freedom must have at least $(f+2)$ binary links in a kinematic chain which do not belong to a basic rigid subchain.*

The proof of this theorem also assumes that a degenerate chains contains basic rigid subchain and hence requires the Proposition 7.10 provided in this work.

Based on the above theorems, Hwang and Hwang [35] present an algorithm for the detection of basic rigid subchains with n_b links in a given kinematic chain. This algorithm, described below, might have to be used several times to check if a given chain is degenerate.

Hwang and Hwang's conditionally valid n_b-link basic rigid subchain detection algorithm:

1. For a kinematic chain with n links and f degrees of freedom, if it contains an $(f+2)$-contracted-link, then it is a degenerate kinematic chain (Theorem 7.4), Stop the procedure; otherwise proceed to the next step.

2. If $n < f + n_b + 4$, then the kinematic chain is not a degenerate kinematic chain

(Theorem 7.6), Stop the procedure; otherwise proceed to the next step.

3. Store the original kinematic chain in the stack and proceed to the next step.
4. If the stack is not empty, pick a chain from the stack and Go to Step 5; otherwise, Stop the procedure and the original chain is not a degenerate kinematic chain.
5. Remove all binary links of length ≥ 2 (Theorem 7.7) and identify the number of links as $n^{'}$ and the degrees of freedom as $f^{'}$ for the new kinematic chain.
    (a) If $f^{'} \leq 0$, then the original chain is a degenerate kinematic chain, Stop the procedure.
    (b) If $f^{'} > 0$ and $n < f^{'} + n_b + 4$, Go to Step 4.
    (c) If $f^{'} > 0$ and $n \geq f^{'} + n_b + 4$, Go to Step 6.
6. For the new chain containing only multiple links and binary chains of length 1 formed in Step 5, calculate the number of binary links $n_2^{'}$, arbitrarily select $n_2^{'} - f^{'} - 1$ binary links to be removed (Theorem 7.8), and remove only one of them each time to generate $n_2^{'} - f^{'} - 1$ subchains. Store these chains in the stack and Go to Step 4.

Unfortunately, this algorithm does not detect all degenerate kinematic chains. The degenerate 8-link 1-DOF kinematic chain, containing the 3-link triangular rigid structure, shown in Figure 7.1 is an example that will not be identified as a degenerate chain by the Hwang and Hwang's degeneracy testing algorithm. This incorrect degeneracy testing algorithm of Hwang and Hwang could be the main reason for the discrepancy in their structural synthesis results [22], as all the unmatched results of Hwang and Hwang [22] are larger than those reported by Tuttle [10] and the results of Chapter 4.

Tuttle's [10] work on degeneracy testing yields the following structural result that can be used to improve Hwang and Hwang's degeneracy testing algorithm.

Theorem 7.9 (Tuttle, 1996). *A basic rigid chain with n_b links can be present only in kinematic chains of at least $f + n_b + 2$ links and f degrees of freedom.*

Theorem 7.9 implies that Step 2 and Step 5 in the above algorithm can be modified by replacing $f+n_b+4$ with $f+n_b+2$ and then completely eliminating Step 1. After modifying Step 1, Step 2 and Step 5 based on Theorem 7.9 the reduction process ends after the graph shown in Figure 7.2(a) is obtained and falsely identifies the original kinematic chain as non-degenerate. Even replacing the constraints in Step 5 by Gruebler's DOF equation yields incorrect results. The step by step reduction process after making all the above modifications to Hwang and Hwang's algorithm is shown in Figure 7.2. The final graphs as shown in Figures 7.2(b) and 7.2(c) are non-degenerate and hence the original graph would be falsely identified as non-degenerate by the modified Hwang and Hwang's algorithm.

The reason that Hwang and Hwang's algorithm fails in this case is that Step 6 of Hwang and Hwang algorithm is based on Theorem 7.8. Theorem 7.8 is valid only for closed kinematic chains. However, at an intermediate step in the reduction process there is no guarantee that the substructure obtained is a closed kinematic

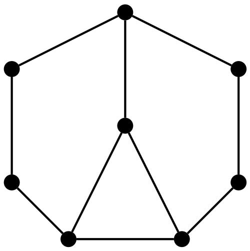
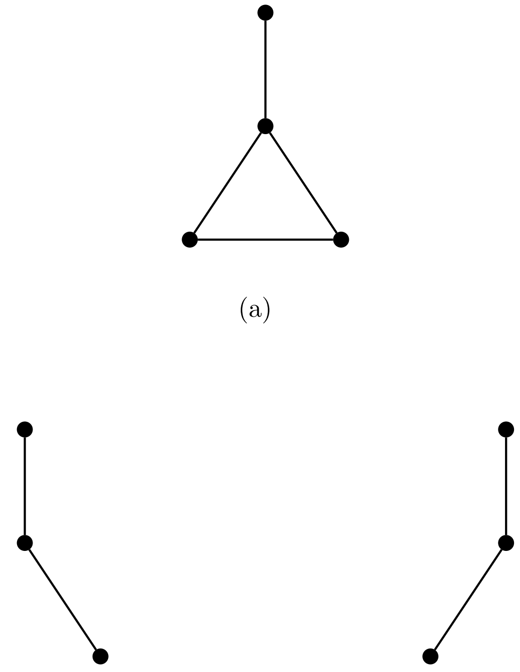


Figure 7.1: Counterexample for Hwang and Hwang's algorithm.



(a)

(b)

(c)

Figure 7.2: (a) This is obtained after Step 5 from the chain in Figure 7.1 (b), (c) are obtained from Figure 7.2(a) after applying Step 6

chain. More specifically, the substructure might not have all vertices of degree ≥ 2. This is clearly seen in the example shown in Figures 7.1 and 7.2. Also, when selecting binary chains of length exactly equal to 2, the selected chains should be checked for existence of triangles, as triangles should not be removed from the kinematic chain.

7.2.3 Lee and Yoon's degeneracy testing algorithm

Lee and Yoon's degeneracy testing algorithm [8], is similar to Hwang and Hwang's [35] algorithm. It rectifies the deficiencies pointed out in the previous section by adding two simple steps. The first step is to verify if the selected binary chain is a loop. The other step is to remove the maximal binary chain, called the pendant chain, attached to a vertex of degree less than two from intermediate structures before removing other binary chains. Removal of such chains will not affect the degeneracy testing because a basic degenerate binary chain cannot contain such links as they are not closed. Also, this guarantees that the intermediate chains that are obtained have all vertices of degree ≥ 2. Lee and Yoon's algorithm also discards the unnecessary Steps 1 and 2 in Hwang and Hwang's algorithm. Removing these steps eliminates the necessity of testing the presence of all possible basic rigid chains. The validity of the most crucial step, Step 7, was missing from Lee and Yoon's work. Also proofs of some of the steps depended on the assumption that a degenerate kinematic chain contains basic rigid subchain.

Lee and Yoon's consistently valid degeneracy testing algorithm:

1. Store the given kinematic chain G in a stack.

2. If stack is empty G is non-degenerate and Stop procedure; otherwise pick a chain and proceed to next step.
3. Select a pendant chain, if it exists remove this subchain and Repeat this step; otherwise proceed to next step. (Justified by the discussion above)
4. Select a binary chains of length ≥ 2. If none exists Go to Step 7; otherwise proceed to next step.
5. If the selected subchain is loop proceed to next step; otherwise remove this subchain and Go to Step 3. (Justified by the Proposition 7.2).
6. If the DOF of the loop is < 1 then G is degenerate Stop procedure; otherwise remove the loop and Go to Step 3 (Justified by the Proposition 7.2).
7. Select all the binary links, and remove only one of them each time to generate set of subchains (Theorem 7.8 and Proposition 7.10). Store these chains into the stack and Go to Step 2.

Propositions 7.2 and 7.10 along with Theorem 7.8 justifies the key steps.

7.2.4 Proposition 7.10 and its proof

The original analysis of Hwang and Hwang [35] and Lee and Yoon [8] assumes that a degenerate kinematic chain contains a basic rigid subchain. This assumption requires a rigorous proof. Proposition 7.10 provides that proof. It should be noted that the proof of Proposition 7.10 does not depend on planarity assumption.

Proposition 7.10. *A planar kinematic chain, with degree of freedom ≥ 0, is degenerate if and only if there exists a basic rigid subchain.*

Proof. Let G be the given degenerate planar kinematic chain. By definition, there exists a closed subchain such that its DOF ≤ 1. Let H be one such subchain with a minimum number of links. If DOF(H) $= 0$, then, by definition of H, due to the minimality of links, all its closed subchains must have their DOF > 0 and hence, H is a basic rigid chain.

Suppose that DOF(H) < 0 then similarly, by definition H, all its closed subchains must have their DOF > 0. Also, the number of links of H is greater than three.

Any (closed or non-closed) subchain of a kinematic chain is either a union of trees or can be constructed from a closed subchain by successively adding a vertex of degree one.

If a chain is union of trees, clearly, its DOF > 0. By adding a vertex of degree one to a chain, the resulting new chain will have one DOF more.

Hence, if all the closed subchains have their DOF > 0, it implies that all the subchains with 3 links or more have their DOF > 0. In particular all the subchains obtained by removing a single link have their DOF > 0. However, by Lemma 7.11 DOF(H) ≥ 0 and hence gives a contradiction. This implies that DOF(H) $= 0$ and that H is a basic rigid subchain. □

Lemma 7.11. *Let G be a planar kinematic chain. If all the subchains of G with exactly one link removed have degrees of freedom greater than zero, then the degrees* *of freedom of G is greater than or equal to zero.*

Proof. Let V, E be the set of vertices and edges of G. Also let d_i be the degree of vertex $v_i \in V$ and let $G_i = G \setminus v_i$.

$$\begin{aligned} \mathrm{DOF}(G) &= 3|V| - 2|E| - 3 \\ \mathrm{DOF}(G_i) &= 3(|V| - 1) - 2(|E| - d_i) - 3 \end{aligned} \tag{7.4}$$

Let $\mathrm{DOF}(G_i) \geq \alpha$, using Equation 7.4,

$$\begin{aligned} \sum_{1}^{|V|}(3(|V| - 1) - 2(|E| - d_i) - 3) &\geq \sum_{1}^{|V|}\alpha \\ \sum_{1}^{|V|}((3|V| - 2|E|) - 2(3 - d_i)) &\geq \alpha|V| \\ (|V|(3|V| - 2|E|) - 2(3|V| - 2|E|)) &\geq \alpha|V| \\ (3|V| - 2|E| - 3) &\geq (\frac{|V|}{|V| - 2}\alpha) - 3 \\ \mathrm{DOF}(G) &> \alpha - 3 \end{aligned} \tag{7.5}$$

From Equation 7.4 it follows that if $|V|$ is odd then $\mathrm{DOF}(G_i)$ is odd and $\mathrm{DOF}(G)$ is even. Similarly, if $|V|$ is even then $\mathrm{DOF}(G_i)$ is even and $\mathrm{DOF}(G)$ is odd.

If $|V|$ is odd, we have the following three statements

$\mathrm{DOF}(G_i) \geq 1$

$\mathrm{DOF}(G) > -2$ (using Equation 7.5)

$\mathrm{DOF}(G)$ is even

Hence, it follows that $\mathrm{DOF}(G) \geq 0$

If $|V|$ is even, we similarly have the following three statements

$\text{DOF}(G_i) \geq 2$

$\text{DOF}(G) > -1$ (using Equation 7.5)

$\text{DOF}(G)$ is odd

Hence, it follows $\text{DOF}(G) \geq 1 \geq 0$

□

7.3 Support for Lee and Yoon's mobility type identification algorithms

Lee and Yoon's algorithm [9] for mobility type identification is similar to their degeneracy testing algorithm. This former algorithm determines if a given closed non-fractionated kinematic chain has total or partial mobility.

Lee and Yoon's mobility type identification algorithm:

1. Store the given kinematic chain G in a stack.
2. If stack is empty G has total mobility and Stop procedure; otherwise pick a chain and proceed to the next step.
3. Select a pendant chain, if it exists remove this subchain and Repeat this step; otherwise proceed to the next step. (Pendant chains are not closed and hence can be eliminated from the analysis).
4. Select a binary chain of longest length. If its length < 2 Go to Step 8; otherwise proceed to the next step.

5. If the selected binary subchain is loop proceed to the next step; otherwise Go to Step 7.

6. If the DOF of the loop is < DOF(G) then G has partial mobility Stop procedure; otherwise remove the loop and Go to Step 3.

7. If the length of the binary chain is > 2 then G has partial mobility Stop procedure; otherwise remove the selected subchain and Go to Step 3. (Justified by the Proposition 7.2).

8. Select all the binary links, and remove only one of them each of time to generate set of subchains. Store these chains into the stack and Go to Step 2. (*This step needs justification*)

All the steps in the above algorithm can be justified by previous propositions or remarks except the last step. Proposition 7.12, stated below, is required for the justification of the last step. This proposition is similar to the Theorem 7.8 by Hwang and Hwang. But the proof of this proposition cannot be similar to the proof of Theorem 7.8 because, unlike a basic rigid chain, a partial mobility chain cannot be replaced by a single link or some unique subchain.

Proposition 7.12. *Let G be a closed, non-fractionated kinematic chain with partial mobility. Then there exists a subchain H of G which is obtained by removing a single binary link of G and which also has partial mobility.*

The proof of Proposition 7.12 follows easily from the Lemma 7.13. This lemma states a more general claim than the proposition above. Lemma 7.13 can be used to extend the Lee and Yoon's mobility type identification algorithm to the more general α-mobility identification algorithm for a given α. This lemma can also be used give an alternate proof for Theorem 7.8 without requiring the assumption that a degenerate kinematic chain contains a basic rigid subchain. It should be noted that the proof of Lemma 7.13 uses the fact that the minimum degree of links in a kinematic chain is ≥ 2.

Lemma 7.13. *Let G be a closed, non-fractionated kinematic chain with f (> 0) degrees of freedom. Suppose if G contains a subchain R of the degrees of freedom $< f$, then there exists a subchain H of G containing R which is obtained by removing a single binary link of G.*

Proof. Let $G(V_G, E_G) = G$. Since the $f = \text{DOF}(G) > 0$ the number of binary vertices is greater than one. By hypothesis, there exists a subchain, R, such that $\text{DOF}(R) < \text{DOF}(G)$. If there exists a binary vertex not belonging to R, then that vertex can be removed and the result follows. Now suppose that all the binary vertices belong to R. We can then decompose V_G into two disjoint sets, V_R and V_S. Let $R(V_R, E_R)$ and $S(V_S, E_S)$ be the induced subgraphs of V_R and V_S respectively. By the previous assumption all the vertices of G belonging to V_S do not have degree 2. Let $E_G \setminus (E_R \cup E_S) = E_{RS}$. By the definition of R and S, $|V_G| = |V_R| + |V_S|$ and $|E_G| = |E_R| + |E_S| + |E_{RS}|$. Hence,

$$\begin{aligned} \text{DOF}(G) &= 3|V_G| - 2|E_G| - 3 \\ &= 3(|V_R| + |V_S|) - 2(|E_R| + |E_S| + |E_{RS}|) - 3 \qquad (7.6) \end{aligned}$$

Equation (7.6) can be rewritten as

$$\mathrm{DOF}(G) \quad = \quad (3|V_R| - 2|E_R| - 3) + (3|V_S| - 2|E_S| - |E_{RS}|) - |E_{RS}| \tag{7.7}$$

This can be simplified as

$$\mathrm{DOF}(G) = \mathrm{DOF}(R) + \sum_S (3 - d_i^G) - |E_{RS}| \tag{7.8}$$

where d_i^G is the degree of the vertex i of the graph G. From the definition of S, $d_i^G \neq 2$ for all vertices i of S. Since G is 2-connected, $d_i^G \geq 2$, it implies $d_i^G \geq 3$ and due to connectivity $|E_{RS}| \geq 1$. Hence, $\mathrm{DOF}(G) < \mathrm{DOF}(R) < f$. This is a contradiction and hence our assumption that all the binary vertices belong to R is false. This completes the proof. □

7.4 Future directions for generalizing structural analysis algorithms

With the justification provided in this work, the degeneracy testing and mobility type identification algorithms of Lee and Yoon [8, 9] are the only efficient algorithms which are applicable for planar kinematic chains with planar or non-planar graphs. Lee and Yoon's algorithms are valid only for planar kinematic chains with revolute joints. The extension of Lee and Yoon's [8] degeneracy testing algorithm to planar mechanisms with different types of joints results in certain problems.

The joints in planar mechanisms can have either 1 or 2 degrees of freedom. If a planar mechanism consists of joints with only 1 degrees of freedom then Lee and Yoon's [8] procedure can be applied to remove binary links. If a planar mechanism contains both types of joints then all the binary chains which contain joints with only 1 degrees of freedom and at least one joint with 2 degrees of freedom can be removed. In the case of the planar mechanisms with only 1 DOF joints, sequential removal of binary chains eventually results in mechanisms with 3 or fewer links. However, for planar mechanisms with different joints, there is no guarantee that the sequential removal of binary chains will eventually result in mechanisms with 3 or fewer links.

There exist degenerate planar kinematic chains with different types of joints, which contain no binary links. Figure 8.2(a) in the next chapter shows such chains. The method of removing binary chains will not be applicable to these chains. Lee and Yoon's [9] algorithm for mobility type identification, similar to their degeneracy testing algorithm, uses sequential removal of binary chains and hence, cannot be extended to planar mechanisms with different types of joints.

None of the results stated earlier are applicable for spatial mechanisms satisfying Gruebler's degrees of freedom equation. This is because the graph structure of spatial mechanisms is different from planar mechanisms due to the differences in the corresponding degrees of freedom equations. Figure 8.2(b) of next chapter shows a chain to which the method of removing binary chains will not be applicable. In order to develop degeneracy testing algorithms for spatial mechanisms structural results similar to the theorems stated for planar mechanisms must be obtained. Structural results of spatial kinematic mechanisms, similar to that of planar mechanisms, hold the key to success of efficient algorithms for structural analysis.

7.5 Summary

Several algorithms exist for the mobility type identification of kinematic chains. It is shown that almost all the algorithms for mobility type identification assume that the graph of a planar kinematic chain is a planar graph and must be used with this limitation in mind. Degeneracy testing of kinematic chains was most often addressed in structural synthesis, but it can be considered as part of mobility analysis as it is equivalent to identification of non-degenerate mobility. This work proves several structural results and justifies implicit assumptions for planar kinematic chains, used in literature, without the assumption that the graph of a planar kinematic chain is a planar graph.

The two existing algorithms on degeneracy testing by Hwang and Hwang [35] and Lee and Yoon [8] were investigated. The limitations of Hwang and Hwang's theorems and potential implementation errors of their degeneracy testing algorithm are detailed. These issues may account for the discrepancies between structural synthesis results of Hwang and Hwang [22] and later works [10]. The justification for Lee and Yoon's [8, 9] degeneracy testing and mobility analysis algorithms is provided, making Lee and Yoon's algorithms the preferred choice. However, these algorithms cannot be extended to planar mechanisms with different types of joints or to spatial mechanisms that satisfy the Gruebler's degrees of freedom equation. Additional structural results, similar to that of planar mechanisms with revolute joints, are needed for this purpose.

Chapter 8

Polynomial-time Degeneracy Testing Algorithm

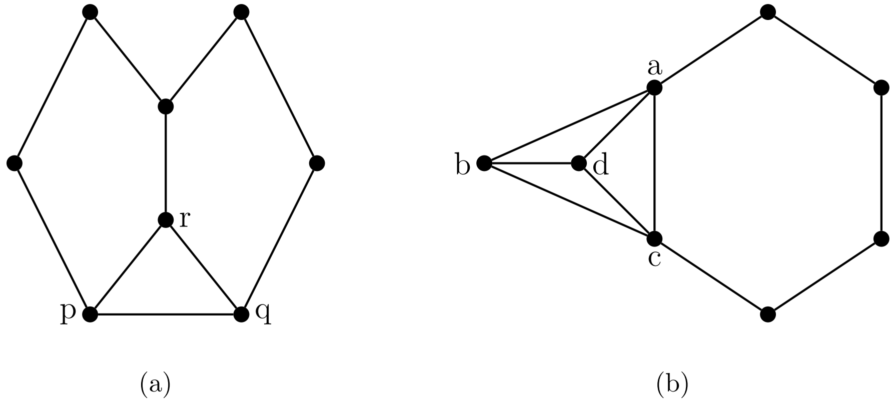
This chapter presents a polynomial-time algorithm for degeneracy testing. Unlike the degeneracy testing algorithms discussed in previous chapter, this degeneracy testing algorithm will be applicable for both planar and spatial mechanisms with different types of joints that have planar or non-planar graphs.

8.1 Introduction

The Figure 8.1(a) shows a degenerate planar kinematic chain with revolute joints. The induced subgraph generated by vertices p, q and r is the subgraph of a rigid subchain. Similarly, the Figure 8.1(b) shows a degenerate spatial kinematic chain with spherical joints. The induced subgraph generated by a, b, c and d is the subgraph of a rigid subchain. The method of removing binary chains was the basis of the degeneracy testing algorithms for planar kinematic chains with revolute joints discussed in Chapter 7. However, this approach will not be applicable to more general kinematic chains as there can exist degenerate planar and spatial kinematic chains with different types of joints, that contain no binary links. Figure 8.2 shows such chains.

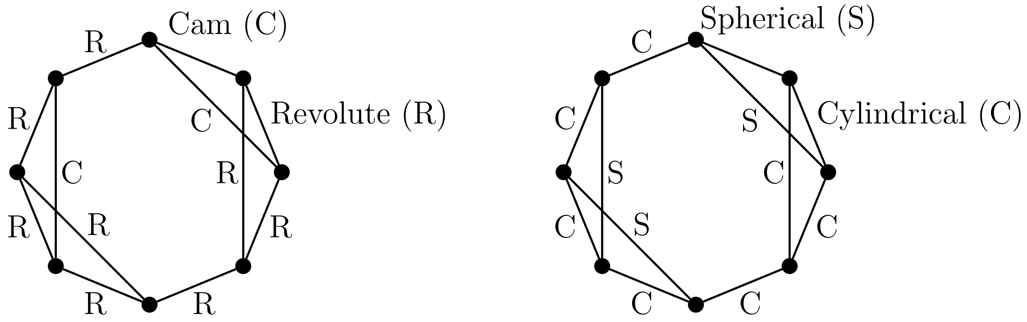
Unlike kinematic chains with identical joints, the topological structure of the graph alone may not determine whether a given chain is degenerate. Spatial kinematic chains consisting of revolute (R) and cylindrical (C) joints are shown in Figure 8.3. The kinematic chain depicted in Figure 8.3(a) is not degenerate, however the spatial kinematic chain in Figure 8.3(b) is degenerate as the outer loop forms a rigid structure. The topological structure of the graphs of the chains in Figure 8.3 are the same, but that fact alone is not sufficient to determine if the chains are degenerate. Similarly, Figure 8.4 shows graphs of two spatial kinematic chain with R and S joints with same topological structure but one is degenerate and other is not. For spatial kinematic chains joint types play a major role in degeneracy testing.

In terms of graph theory the degeneracy testing must be performed on graphs with labelled edges so that joint types can be indicated. Typical approaches are to use weighted or colored graphs. This motivates the use of combinatorial optimization in solving the degeneracy testing problem, as the underlying graph is a weighted graph.



(a) (b)

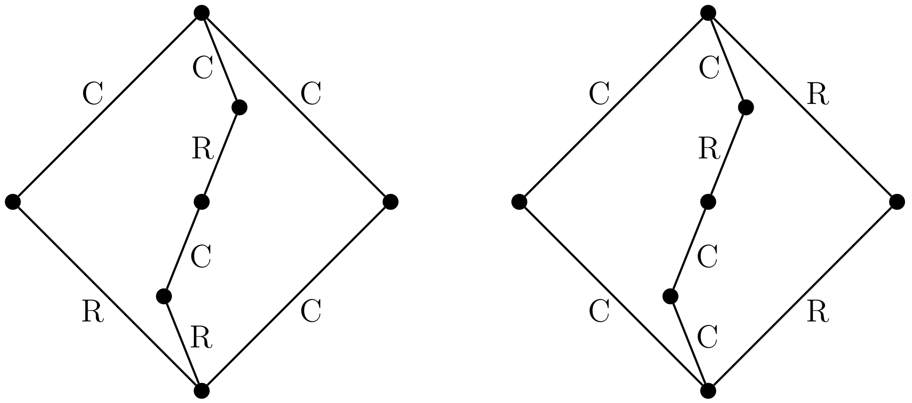
Figure 8.1: Graphs of (a) 8-link degenerate planar mechanism with revolute joints and (b) 8-link degenerate spatial mechanism with spherical joints



(a)

(b)

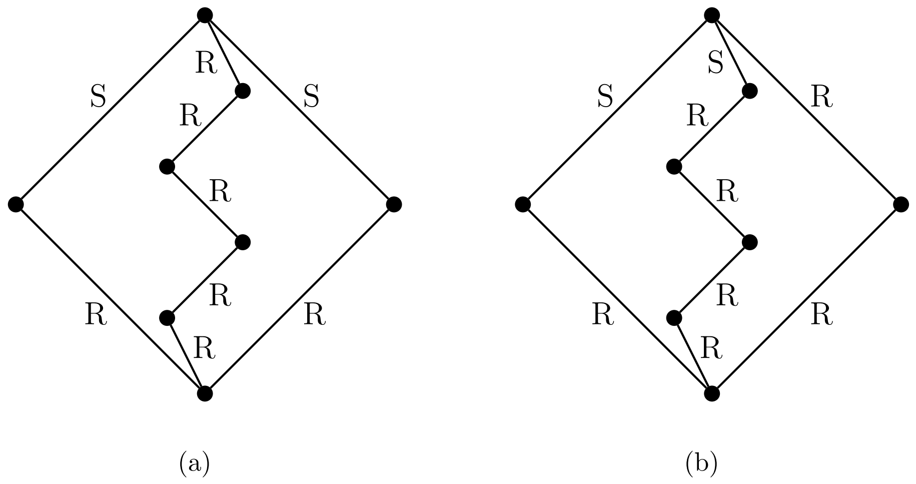
Figure 8.2: Graphs of (a) degenerate planar chain without binary links and (b) degenerate spatial chain without binary links



(a)

(b)

Figure 8.3: Graphs of spatial mechanism with R and C joints with same topological structure such that (a) is non-degenerate and (b) is degenerate



(a)

(b)

Figure 8.4: Graphs of spatial mechanisms with R and S joints with same topological structure such that (a) is non-degenerate and (b) is degenerate

8.2 Formulation of Degeneracy Testing as an Optimization Problem

The degeneracy testing problem is equivalent to verifying that the degree of freedom equation is greater than zero for the each induced closed subgraphs. Recall that the DOF equation is a linear function of the vertices and edges of a graph. For any planar kinematic chain, G, with revolute joints this amounts to checking if

$$\mathrm{DOF}(G^{'}) = 3|V(G^{'})| - 2|E(G^{'})| - 3 > 0 \tag{8.1}$$

for all closed induced subgraphs $G^{'}$ of G. In the case of spatial kinematic chains the problem is complicated because the type of joint impacts the DOF of any subchain that includes it. Hence, in this case it is necessary to use an indicator (e.g. weights) of each edge to recognize joint type.

An edge-weighted graph can be considered as an ordered pair (G, w) where $G = (V, E)$ and w is the weight function on E. Given an induced subgraph $G^{'}$ of G, let $W(G^{'})$ denote the sum of all the weights of the edges of $G^{'}$. Given the graph representation of a kinematic chain one can construct an edge-weighted graph from it called the constraint graph. In the constraint graph the weight of each edges is set as the number of constraints imposed by the corresponding joint. For a planar kinematic chain with revolute joints weight of each edge will be 2 (as revolute joint allows only one DOF) and $W(G^{'}) = 2|E(G^{'})|$. Figure 8.5 shows the constraint graphs corresponding to the kinematic chains in Figure 8.4.

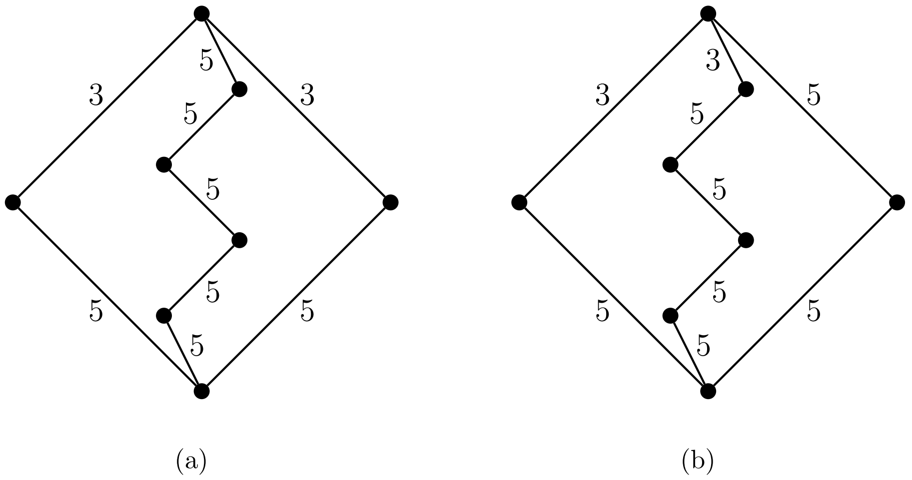


Figure 8.5: Constraint graphs of spatial mechanisms in (a) Figure 8.4(a) and (b) Figure 8.4(b)

Using this terminology the Equation 8.1 can be written as

$$\mathrm{DOF}(G^{'}) = 3|V(G^{'})| - W(G^{'}) - 3 > 0 \tag{8.2}$$

Thus the equation reduces to the Equation 8.1.

In general, given a kinematic chain (planar or spatial), let (G, w) denote the corresponding constraint graph. Now, checking the kinematic chain for degeneracy is equivalent to checking if

$$\mathrm{DOF}(G^{'}) = \lambda|V(G^{'})| - W(G^{'}) - \lambda > 0 \tag{8.3}$$

for all closed induced subgraphs $G^{'}$ of G, where λ is the motion parameter (3 for planar mechanisms and 6 for spatial mechanisms). Each induced subgraph $G^{'}$ of G can be represented by a vector, $X(G^{'}) = (x_1(G^{'}) \ldots x_n(G^{'}))$, such that n is equal to the vertex set of G and each $x_i(G^{'})$ is 1 if and only if the vertex v_i of G belongs to $G^{'}$ and 0 otherwise. Let $A = (a_{ij})$ be the weighted adjacency matrix of (G, w). Then

$$\begin{aligned} |V(G^{'})| &= \sum_{i=1}^{n} x_i(G^{'}) \\ W(G^{'}) &= \frac{1}{2} X^T(G^{'}) A X(G^{'}) \\ &= \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_i(G^{'}) x_j(G^{'}) \qquad (8.4) \end{aligned}$$

Using Equation 8.4 degeneracy testing is equivalent to the problem of verifying if

$$\mathrm{DOF}(G^{'}) = \lambda \sum_{i=1}^{n} x_i(G^{'}) - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_i(G^{'}) x_j(G^{'}) - \lambda > 0 \qquad (8.5)$$

for all closed induced subgraphs $G^{'}$ of G.

One way to solve this problem is solve the *harder* problem of finding the minimum of $\mathrm{DOF}(G^{'})$ over all closed induced subgraphs $G^{'}$ of G. The goal is to use combinatorial optimization techniques to solve this minimization problem but there is no easy way to represent the space of all closed induced subgraphs a G. However, the space of all induced subgraphs can be represented by a 0-1 vector of dimension equal to the size of G. The following lemma states that that minimum of $\mathrm{DOF}(G^{'})$ can, in fact, be taken over all induced subgraphs of size > 1.

Lemma 8.1. *Let G be a kinematic chain then* $\mathrm{DOF}(G^{'}) > 0$ *for all closed induced subgraphs $G^{'}$ of G is equivalent to* $\mathrm{DOF}(G^{'}) > 0$ *for all induced subgraphs $G^{'}$ of G*

with size greater than 1.

Proof. Since all the closed induced subgraphs of G are contained in the induced subgraphs of G with size greater than 1, $\mathrm{DOF}(G^{'}) > 0$ for all induced subgraphs with size greater than 1 implies $\mathrm{DOF}(G^{'}) > 0$ for all closed induced subgraphs. Suppose if $\mathrm{DOF}(G^{'}) \leq 0$ for some open induced subgraph $G^{'}$ of size greater than 1, then, since $G^{'}$ is open there exists a vertex of degree ≤ 1. By removing that vertex and the attached edge (if any) a new induced subgraph $G^{''}$ is formed and the $\mathrm{DOF}(G^{''}) = \mathrm{DOF}(G^{'}) - (\lambda - w)$, where w is the weight of the edge if some edge is removed or zero otherwise. This implies $\mathrm{DOF}(G^{''}) < \mathrm{DOF}(G^{'})$. Continuing this process results in either a closed induced subgraph or an induced subgraph on two vertices (as the initial induced subgraph had > 1 vertices).

Suppose if an induced subgraph on 2 vertices is obtained, the assumption that $\mathrm{DOF}(G^{'}) \leq 0$ implies that the degrees of freedom the induced subgraph on 2 vertices is ≤ 0. This is a contradiction, since any induced subgraph on 2 vertices has > 0 degrees of freedom. Hence a closed induced subgraph, say H, is obtained. Since, $\mathrm{DOF}(G^{'}) \leq 0$ this implies $\mathrm{DOF}(H) \leq 0$. Hence $\mathrm{DOF}(G^{'}) > 0$ for all closed induced subgraphs implies $\mathrm{DOF}(G^{'}) > 0$ for all induced subgraphs with size greater than 1. □

Using Lemma 8.1 the degeneracy testing of kinematic chains is equivalent to verifying if

$$\mathrm{DOF}(G^{'}) = \lambda \sum_{i=1}^{n} x_i(G^{'}) - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_i(G^{'}) x_j(G^{'}) - \lambda > 0 \qquad (8.6)$$

for all induced subgraphs $G^{'}$ of G with size greater than 1. This in turn is equivalent to checking if

$$\mathrm{DOF}(G^{'}) = \lambda \sum_{i=1}^{n} x_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_i x_j - \lambda > 0 \tag{8.7}$$

subject to the constraints $\sum_{i=1}^{n} x_i > 1$ and $x_i \in \{0, 1\}$. One way to solve this problem, as mentioned before, is to minimize

$$\mathrm{DOF}(X) = \lambda \sum_{i=1}^{n} x_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_i x_j - \lambda \tag{8.8}$$

subjected to $\sum_{i=1}^{n} x_i > 1$ and $x_i \in \{0, 1\}$. Hence it is a non-linear 0-1 optimization problem. Some concepts from network flows are needed in order to discuss the solution of this problems. These are presented briefly in the next section.

8.3 Network Flows

A network $\mathcal{N} = (G, c, s, t)$ consists of

1. A directed graph $G = (V, E)$, with finite vertex set V and directed edge set E,
2. A source vertex s and a sink vertex t,
3. A nonnegative capacity $c : E \to \mathbb{R}^{+} \cup \{0\}$.

By defining $c(u, v) = 0$ if $(u, v) \notin E$, c can be extended as a function on $V \times V$. As V is finite c can be thought of as a matrix $C = (c_{ij})$ where $c_{ij} = c(i, j)$. Since G is directed it should be noted that C need not be symmetric.

A flow of $\mathcal{N}$ is a function $f : V \times V \to \mathbb{R}$ satisfying:

1. Capacity Constraint: $f(u,v) \le c(u,v)$ for $(u,v) \in V \times V$,

2. Skew Symmetry: $f(u,v) = -f(v,u)$ for $(u,v) \in V \times V$,

3. Conservation: $\sum_{v \in V} f(u,v) = 0$ for $u \in V - \{s,t\}$.

The value of the flow f is $|f| = \sum_{v \in V} f(s,v)$. A maximum flow is a flow of maximum value. The residual capacity of $(u,v) \in V \times V$, induced by f, is $c_f(u,v) = c(u,v) - f(u,v)$. Note that $c_f(u,v) \ge 0$.

The residual graph of $\mathcal{N}$, induced by f, is $G_f = (V, E_f)$ where $E_f = \{(u,v) \in E | c_f(u,v) > 0\}$. Hence the flow gives rise to the residual network $\mathcal{N}_f = (G_f, c_f, s, t)$. An augmenting path α with respect to f is a path from s to t in G_f. Given an augmenting path α with respect to f, $f_\alpha : V \times V \to \mathbb{R}^{\ge 0}$ is defined as:

$$f_\alpha(u,v) = \begin{cases} \quad min\{c_f(x,y)|(x,y) \in \alpha\} & \text{if } (u,v) \in \alpha \\ - \; min\{c_f(x,y)|(x,y) \in \alpha\} & \text{if } (v,u) \in \alpha \\ \quad 0 & \text{otherwise} \end{cases}$$

A cut (S,T) of $\mathcal{N}$ is a partition of V into S and $T = V \setminus S$ such that $s \in S$ and $t \in T$. The capacity of the cut (S,T) is defined by $c(S,T) = \sum_{(x,y) \in S \times T} c(x,y)$. The flow across (S,T) is $f(S,T) = \sum_{(x,y) \in S \times T} f(x,y)$. A minimum cut is a cut of minimum capacity. The following two lemmas present some easy consequences of the definitions [80, 81].

Lemma 8.2. *Given a network $\mathcal{N} = (G,c,s,t)$ and a flow f. For every cut (S,T), $f(S,T) = |f|$.*

Lemma 8.3. *Given a network $\mathcal{N} = (G,c,s,t)$ and a flow f. For every cut (S,T), $f(S,T) \le c(S,T)$.*

Lemma 8.3 implies that the value of a maximum flow is not greater than the capacity of the minimum cut. The max-flow min-cut theorem of Ford and Fulkerson [80, 81] states that the two quantities are, in fact, equal.

Theorem 8.4 (Max-Flow Min-Cut Theorem)**.** *Given a network $\mathcal{N} = (G, c, s, t)$ and a flow f. The following are equivalent*

1. *f is a maximum flow of $\mathcal{N}$.*
2. *There are no augmenting paths with respect to f.*
3. *$|f| = c(S, T)$ for some cut (S, T) and, hence, the minimum cut of $\mathcal{N}$.*

Maximum Flow Algorithm($\mathcal{N} = (G, c, s, t)$**)**:

```
1. initialize: $f = 0$
2. while(there is an augmenting path with respect to $f$) do
    (a) pick an augmenting path $\alpha$ using breath-first search in $G_f$
    (b) $f = f + f_\alpha$
3. end while
4. return($f$)
```

The maximum flow algorithm of Ford-Fulkerson & Edmonds-Karp described above takes $O(|V||E|^2)$ time [80, 81]. Given a maximum flow f of $\mathcal{N}$, let $S = \{v \in V | s$ and v are connected in $G_f\}$ and let $T = V \setminus S$ then (S, T) forms a minimum cut of $\mathcal{N}$.

Minimum Cut Algorithm($\mathcal{N} = (G, c, s, t)$):

```
1. initialize: $f = 0$
2. while(there is an augmenting path with respect to $f$) do
    (a) pick an augmenting path $\alpha$ using breath-first search in $G_f$
    (b) $f = f + f_\alpha$
3. end while
4. $S = \{v \in V | s$ and $v$ are connected in $G_f\}$ and $T = V \setminus S$
5. return($(S, T)$)
```

8.4 Minimum Cuts and 0-1 Optimization

Certain classes of quadratic 0-1 optimization problems can be solved as minimum cut problems [82, 83, 84]. It will turn out that the optimization problem formulated at the end of the Section 8.2, after some modifications, can also be solved as a minimum cut problem. The approach described in the section closely follows Picard and Ratliff [82]. Given a network $\mathcal{N} = (G, c, s, t)$ let $V = \{v_0, v_1, \dots v_n, v_{n+1}\}$ such that $v_0 = s$ and $v_{n+1} = t$. Considering c as a matrix, for a cut (S, T) of $\mathcal{N}$, $c(S, T)$ can be rewritten as:

$$c(S, T) = \sum_{i \in S} \sum_{j \in T} c_{ij} \tag{8.9}$$

Any cut (S,T) can be represented by a vector $X = (1, x_1, \ldots, x_n, 0)$ where $x_i \in \{0,1\}$, $S = \{v_i | x_i = 1\}$ and $T = \{v_i | x_i = 0\}$. Let $c(X) = c(S,T)$. Then

$$\begin{aligned} c(X) &= \sum_{i=0}^{n+1}\sum_{j=0}^{n+1} c_{ij}x_i(1-x_j) \\ &= \sum_{i=0}^{n+1}(\sum_{j=0}^{n+1} c_{ij})x_i - \sum_{i=0}^{n+1}\sum_{j=0}^{n+1} c_{ij}x_ix_j \end{aligned} \tag{8.10}$$

by substituting $x_0 = 1$ and $x_{n+1} = 0$ and expanding,

$$\begin{aligned} c(X) &= \sum_{i=1}^{n}(\sum_{j=0}^{n+1} c_{ij})x_i + \sum_{j=0}^{n+1} c_{0j} - \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_ix_j - \sum_{i=1}^{n} c_{i0}x_i - \sum_{j=1}^{n} c_{0j}x_j \\ &= \sum_{i=1}^{n}(\sum_{j=1}^{n} c_{ij})x_i + \sum_{i=1}^{n} c_{i,n+1}x_i + \sum_{i=1}^{n} c_{i0}x_i + \sum_{j=0}^{n+1} c_{0j} - \\ &\quad - \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_ix_j - \sum_{i=1}^{n} c_{i0}x_i - \sum_{j=1}^{n} c_{0j}x_j \\ &= \sum_{i=1}^{n}(\sum_{j=1}^{n} c_{ij})x_i + \sum_{i=1}^{n} c_{i,n+1}x_i + \sum_{j=0}^{n+1} c_{0j} - \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_ix_j - \sum_{j=1}^{n} c_{0j}x_j \\ &= \sum_{i=1}^{n}(\sum_{j=1}^{n} c_{ij} + c_{i,n+1} - c_{0i})x_i - \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_ix_j + \sum_{j=0}^{n} c_{0j} + c_{0,n+1} \end{aligned} \tag{8.11}$$

Considering any function of the form

$$f(X) = \sum_{i=1}^{n} p_ix_i - \sum_{i=1}^{n}\sum_{j=1}^{n} q_{ij}x_ix_j + r \tag{8.12}$$

where $x_i \in \{0,1\}$. Comparing Equation 8.12 with the Equation 8.11 we obtain the following relations:

$$c_{ij} + c_{ji} = q_{ij} + q_{ji} \tag{8.13}$$

$$c_{i,n+1} - c_{0i} = p_i - \sum_{j=1}^{n} c_{ij} \tag{8.14}$$

$$c_{0,n+1} = r - \sum_{j=0}^{n} c_{0j} \tag{8.15}$$

for $i, j \in 1, \ldots, n$. Suppose if there exists a network $\mathcal{N} = (G, c, s, t)$ such that the Equations 8.13, 8.14 and 8.15 are satisfied then $f(X)$ can minimized by finding a minimum cut of $\mathcal{N}$. Let $\hat{X}$ be vector representation of a minimum cut of $\mathcal{N}$, then $\hat{X} = (x_1, \ldots, x_n)$. The value of X that minimizes f does not depend on the constant term r of f. Hence, in the Equation 8.15, $c_{0,n+1}$ can be taken to be zero. Suppose if $q_{ij} \geq 0$, using Equations 8.13, 8.14 and 8.15, c can be defined as follows:

$$\begin{array}{lcll}
c_{ij} & = & \frac{1}{2}(q_{ij} + q_{ji}) & \text{if } i \leq j \text{ and for } 1 \leq i, j \leq n \\
c_{ij} & = & 0 & \text{if } i > j \text{ and for } 1 \leq i, j \leq n \\
c_{i,n+1} & = & \frac{1}{2}(|p_i - \sum_{j=1}^{n} c_{ij}| + p_i - \sum_{j=1}^{n} c_{ij}) & \text{for } 1 \leq i \leq n \\
c_{0i} & = & \frac{1}{2}(|p_i - \sum_{j=1}^{n} c_{ij}| + \sum_{j=1}^{n} c_{ij} - p_i) & \text{for } 1 \leq i \leq n \\
c_{0,n+1} & = & 0 &
\end{array}$$

8.5 Solution of Degeneracy Testing

The objective function (Equation 8.8) for testing degeneracy of a graph $G = (V, E)$ (of kinematic chain) is similar to the function f in Equation 8.12. However, the degeneracy testing problem has an additional constraint that $\sum_{i=1}^{n} x_i > 1$. Hence, the method described in Section 8.4 cannot be applied directly. However, the degeneracy testing problem can be reduced to minimizing m functions of the form in Equation 8.12, where $m = |E|$.

Suppose that for all pairs of vertices there exists no degenerate subgraphs of G containing that pair, then there cannot exist a degenerate subgraph with more than 1 vertex. This implies that minimizing the $n(n-1)/2$ functions obtained by substituting $x_k = x_l = 1$ for all combinations $\{k, l\}$ gives the minimum of the objective function (Equation 8.8) for testing degeneracy, where $n = |V|$. The $n(n-1)/2$ functions can be reduced to m by observing that if there exists a degenerate graph containing a selected pair of non-adjacent vertices it must contain some edge, say e, (otherwise that subgraph is completely disconnected and its DOF will be greater than any subgraph consisting of single edge) but then the same graph will be a degenerate graph containing the endpoints of e. Hence it is enough to consider the pairs of vertices that are adjacent. This implies that degeneracy testing problem can be reduced to minimizing m functions of the form in Equation 8.12.

Given $\{k,l\} \in E$, substituting $x_k = x_l = 1$ in the objective function, shown in Equation 8.8, can be rewritten as follows:

$$
\begin{aligned}
\mathrm{DOF}_{kl}(X) &= \sum_{i=1}^{n} \lambda x_i - \sum_{i=1}^{n}\sum_{j=1}^{n} \frac{a_{ij}}{2} x_i x_j - \lambda \\
&= \sum_{\substack{i=1\\ i\notin\{k,l\}}}^{n} \lambda x_i - \sum_{\substack{i=1\\ i\notin\{k,l\}}}^{n} \sum_{j=1}^{n} \frac{a_{ij}}{2} x_i x_j - \sum_{j=1}^{n} \frac{a_{kj}}{2} x_j - \sum_{j=1}^{n} \frac{a_{lj}}{2} x_j + \lambda \\
&= \sum_{\substack{i=1\\ i\notin\{k,l\}}}^{n} \lambda x_i - \sum_{\substack{i=1\\ i\notin\{k,l\}}}^{n} \sum_{\substack{j=1\\ j\notin\{k,l\}}}^{n} \frac{a_{ij}}{2} x_i x_j - \sum_{\substack{i=1\\ i\notin\{k,l\}}}^{n} \frac{a_{ik}}{2} x_i - \sum_{\substack{i=1\\ i\notin\{k,l\}}}^{n} \frac{a_{il}}{2} x_i - \\
&\quad - \sum_{\substack{j=1\\ j\notin\{k,l\}}}^{n} \frac{a_{kj}}{2} x_j - \sum_{\substack{j=1\\ j\notin\{k,l\}}}^{n} \frac{a_{lj}}{2} x_j + (\lambda - a_{kl}) \\
&= \sum_{\substack{i=1\\ i\notin\{k,l\}}}^{n} (\lambda - a_{ik} - a_{il}) x_i - \sum_{\substack{i=1\\ i\notin\{k,l\}}}^{n} \sum_{\substack{j=1\\ j\notin\{k,l\}}}^{n} \frac{a_{ij}}{2} x_i x_j + (\lambda - 1)
\end{aligned}
\tag{8.16}
$$

Let $\pi : \{1, \ldots, n\} \rightarrow \{0, \ldots, n-1\}$ be a one-to-one function such that $\pi(\{1, \ldots, n\} \setminus \{k, l\}) = \{1, \ldots, n-2\}$, $\pi(k) = 0$ and $\pi(l) = n-1$. π is nothing but a relabelling function. Let its inverse be denoted by π^{-1}. Let $b_{ij} = a_{\pi^{-1}(i),\pi^{-1}(j)}$,

$y_i = x_{\pi^{-1}(i)}$ and $Y = (y_1, \ldots, y_{n-2})$ then Equation 8.16 is equivalent to

$$\mathrm{DOF}(Y) = \sum_{i=1}^{n-2} (\lambda - b_{i0} - b_{i,n-1}) y_i - \sum_{i=1}^{n-2} \sum_{j=1}^{n-2} \frac{b_{ij}}{2} y_i y_j + (\lambda - 1) \tag{8.17}$$

Since (a_{ij}) is symmetric, (b_{ij}) is also symmetric. Equation 8.17 has a similar form as the Equation 8.12 and $b_{ij} \geq 0$, hence there exists a network $\mathcal{N}' = (G', c', s', t')$ with $G' = ((v'_0, v'_1, \ldots, v'_{n-2}, v'_{n-1}), E')$ such that $s' = v'_0$, $t' = v'_{n-1}$ and

$$\begin{array}{lcll}
c'_{ij} & = & \frac{1}{2} b_{ij} & \text{if } i \leq j \text{ and for } 1 \leq i,j \leq n-2 \\
c'_{ij} & = & 0 & \text{if } i > j \text{ and for } 1 \leq i,j \leq n-2 \\
c'_{i,n-1} & = & \frac{1}{2}(|(\lambda - b_{i0} - b_{i,n-1}) - \sum_{j=1}^{n-2} c'_{ij}| + & \\
 & & +(\lambda - b_{i0} - b_{i,n-1}) - \sum_{j=1}^{n-2} c'_{ij}) & \text{for } 1 \leq i \leq n-2 \\
c'_{0i} & = & \frac{1}{2}(|(\lambda - b_{i0} - b_{i,n-1}) - \sum_{j=1}^{n-2} c'_{ij}| + & \\
 & & +\sum_{j=1}^{n-2} c'_{ij} - (\lambda - b_{i0} - b_{i,n-1})) & \text{for } 1 \leq i \leq n-2 \\
c'_{0,n-1} & = & 0 &
\end{array}$$

whose minimum cut vector, $\hat{Y}$, minimizes $\mathrm{DOF}(Y)$ (Equation 8.17). Expressing these results in terms of a_{ij} by using the function π, we obtain that there exists a network $\mathcal{N} = (G, c, s, t)$ with $G = ((v_1, \ldots, v_n), E)$ such that $s = v_k$, $t = v_l$ and

$$\begin{array}{lcll}
c_{ij} & = & \frac{1}{2} a_{ij} & \text{if } i \leq j \text{ and for } i,j \in \{1, \ldots, n\} \setminus \{k,l\} \\
c_{ij} & = & 0 & \text{if } i > j \text{ and for } i,j \in \{1, \ldots, n\} \setminus \{k,l\} \\
c_{il} & = & \frac{1}{2}(|[\lambda - a_{ik} - a_{il}] - \sum^{n}_{\substack{j=1 \\ j \notin \{k,l\}}} c_{ij}| + & \\
 & & +[\lambda - a_{ik} - a_{il}] - \sum^{n}_{\substack{j=1 \\ j \notin \{k,l\}}} c_{ij}) & \text{for } i \in \{1, \ldots, n\} \setminus \{k,l\} \\
c_{ki} & = & \frac{1}{2}(|[\lambda - a_{ik} - a_{il}] - \sum^{n}_{\substack{j=1 \\ j \notin \{k,l\}}} c_{ij}| + & \\
 & & +\sum^{n}_{\substack{j=1 \\ j \notin \{k,l\}}} c_{ij} - [\lambda - a_{ik} - a_{il}]) & \text{for } i \in \{1, \ldots, n\} \setminus \{k,l\} \\
c_{kl} & = & 0 &
\end{array}$$

whose minimum cut vector, X minimizes $\mathrm{DOF}_{kl}(X)$ (Equation 8.16) equivalently $\mathrm{DOF}(X)$ (Equation 8.8) such that $x_k = x_l = 1$. Using these results the algorithm for finding minimum of $\mathrm{DOF}(X)$ such that $\sum_{i=1}^{n} x_i > 1$ is as follows:

```
Minimum($\mathrm{DOF}(X) \ni \sum_{i=1}^{n} x_i > 1$):
initialize: $mDOF = \mathrm{DOF}(G)$
for (all $(k, l) \in E$)
    form the network $\mathcal{N}$ corresponding to $\mathrm{DOF}_{kl}(X)$.
    $X = (S, T) =$ minimum cut($\mathcal{N}$)
    $mDOF = min\{mDOF, \mathrm{DOF}(X)\}$
end for
return($mDOF$)
```

If the $mDOF$ returned by the above algorithm is less than 1 then the kinematic chain with graph G is degenerate. Since the *for-loop* executes E times, this algorithm takes time $O(|V||E|^3)$ as the minimum cut algorithm used here takes $O(|V||E|^2)$ time. Also since we have, by the degree of freedom equation, that $O(|E|) = O(|V|)$, the time taken by the algorithm is of $O(|V|^4)$.

8.6 Results and Discussion

The accuracy of the algorithm is tested by performing structural synthesis of planar kinematic chains with revolute joints with up to 16 links and 1-4 degrees of freedom. Exactly same number of chains were synthesized as compared to the structural synthesis using Lee and Yoon's [8] degeneracy testing algorithm as described in Chapter 4. The actual processor time taken by using the two degeneracy testing algorithms were compared and the results are shown in Tables 8.1 and 8.2. All the programs are written in C language and are run on a PC with 1.7 GHz processor.

Table 8.1: Non-isomorphic kinematic chains with 1 and 2 DOF.

DOF	Links	Chains	Processor Time	
			Lee and Yoon	Current
1	8	16	0.00 sec	0.00 sec
	10	230	0.01 sec	0.03 sec
	12	6856	0.64 sec	1.58 sec
	14	318,162	39.09 sec	101.84 sec
	16	19,819,281	65.08 min	156.33 min
2	9	35	0.00 sec	0.00 sec
	11	753	0.06 sec	0.12 sec
	13	27,496	2.28 sec	6.08 sec
	15	1,432,730	174.09 sec	445.17 sec

Considering only the results with 1 sec or more running time, the polynomial-time algorithm was on an average 2.3 times slower than Lee and Yoon's degeneracy testing algorithm. This might suggest the reduction type degeneracy testing algorithms also are of polynomial order. The time taken by the current degeneracy test-

Table 8.2: Non-isomorphic kinematic chains with 3 & 4 DOF.

DOF	Links	Chains	Processor Time	
			Lee and Yoon	Current
3	10	74	0.00 sec	0.00 sec
	12	1962	0.19 sec	0.41 sec
	14	83,547	8.72 sec	21.14 sec
	16	4,805,764	12.47 min	28.51 min
4	11	126	0.01 sec	0.03 sec
	13	4356	0.69 sec	1.27 sec
	15	216,291	43.67 sec	78.88 sec
	17	13,743,920	62.40 min	114.68 min

ing algorithm depends mainly on the time taken by the maximum flow (minimum cut) algorithm used. The Ford-Fulkerson & Edmonds-Karp maximum flow algorithm is used in the current work and this algorithm takes $O(|V||E|^2)$ time. A more recent maximum flow algorithm by Gallo et al. [85] takes $O(|V||E|\log(|V|^2/|E|))$ time. Hence, using more efficient maximum flow algorithms increases efficiency. Still the total synthesis time for planar mechanisms using the current degeneracy testing algorithm is approximately 5.6 times faster than the recent planar synthesis results by Butcher and Hartman [73], even considering the differences in CPU speeds.

Since there are no other algorithms for degeneracy testing of planar kinematic chains with non-revolute joints (joints with more than one DOF) or for spatial mechanisms the accuracy of the algorithm was established by checking degeneracy of a random spatial kinematic chain by the current algorithm and verifying it by brute force. The current algorithm correctly identified the degenerate spatial kinematic chains with R and C joints in Figures 8.3, with R and S joints in Figure 8.4 and with all S joints in Figure 8.1(b). In the process of degeneracy testing, a spatial kinematic chain with a non-planar graph is identified, it is shown in Figure 8.6.

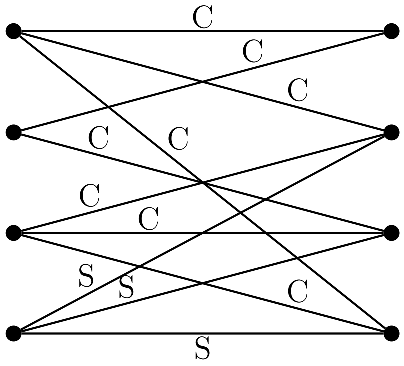


Figure 8.6: Non-planar spatial kinematic chain

Extending the current degeneracy testing algorithm to mobility type identification or checking if a chain has α-mobility ($\alpha > 0$) will be possible only if there exists a *nice* representation of closed induced subgraphs. In the case when $\alpha = 1$, because of the Lemma 8.1, it was sufficient to consider all the induced subgraphs of size > 1. Doing the same thing for mobility type identification will result in an incorrect algorithm. For example, consider a single loop planar mechanism with 6 revolute joints, since it has just 1 closed subchain, it has total mobility. However a subchain with just 2 links attached has fewer degrees of freedom than the total degrees of freedom of the chain. Hence the analogous approach will not result in a polynomial-time mobility type identification algorithm.

In the case of planar graphs, a polynomial-time algorithm for planarity testing led the way to a polynomial-time isomorphism detection algorithm [86, 37]. Analogously, one can also hope that the polynomial-time degeneracy testing algorithm for a kinematic chains may also aid in development of a polynomial-time isomorphism detection algorithm for kinematic chains.

8.7 Summary

A polynomial-time algorithm for degeneracy testing that will apply for both planar and spatial mechanisms with different types of joints that have planar or non-planar graphs was developed. This was done by first reducing the degeneracy testing problem to a 0-1 quadratic optimization problem with a single constraint. Next the 0-1 quadratic optimization problem was further reduced to minimizing

$|E|$ 0-1 quadratic functions, where $|E|$ is the number of edges of the graph. For each of the 0-1 quadratic optimization problems, a network was constructed such that the minimum cut vector of the network provides the minimum value. Finally the minimum cut problem was solved using the Ford-Fulkerson & Edmonds-Karp maximum flow algorithm.

The current degeneracy testing algorithm presented in Section 8.5 identifies all the planar kinematic chains with revolute joints correctly but it appears to be slower than Lee and Yoon's degeneracy testing algorithm for planar kinematic chains. However, the speed of current algorithm can be increased by using more efficient maximum flow algorithms. The current algorithm also correctly identifies degenerate planar kinematic chains with non-revolute joints and spatial kinematic chains. Since there are no other algorithms for degeneracy testing of planar kinematic chains with non-revolute joints or for spatial mechanisms, the comparison was done by brute force.

Chapter 9

Contributions and Future Work

9.1 Summary of Results

The research theme of this work was to address the structural problems of kinematic mechanisms using techniques from graph theory and combinatorial optimization. This work illustrated the power of these techniques as applied to structural kinematic studies. The rest of the section summarizes the results obtained for the research objectives described in the Section 1.2. The objectives satisfied here represent a range of challenges across the broad category of kinematic synthesis.

Objective 1: Structural synthesis of planar kinematic chains:

> A McKay-type algorithm, in combination with an efficient degeneracy testing algorithm, was used for the synthesis of planar mechanisms. This generation algorithm is approximately 13 times faster than the recent planar synthesis algorithm by Butcher and Hartman [73], even after considering the differences in CPU speeds. Lee and Yoon's [8], and Hwang and Hwang's [35] algorithms for degeneracy testing were validated by testing them on different planar mechanisms. The degeneracy testing algorithms were programmed and implemented with a McKay-type algorithm for the enumeration of non-isomorphic planar mechanisms with up to 4 degrees of freedom and 14 links. Since the number

of non-isomorphic planar mechanisms with one degree of freedom and with 6, 8 and 10 links are very well established, these numbers were used to further test the validity of the existing algorithms.

Objective 2: Reliability and efficiency of spectral methods for isomorphism detection:

The reliability of the characteristic polynomial of the adjacency matrix was established. It is shown, for the first time, that the eigenvector approach can identify all non-isomorphic chains, with up to 14 links and 1, 2 and 3 degrees of freedom. It is also shown that, unlike the characteristic polynomial method, the eigenvector approach in the worst case might take exponential computation time and, hence, may not be as efficient as the characteristic polynomial method. Finally, the Perron-Frobenius theorem was used to suggest more efficient methods to the classical eigenvector approach.

Objective 3: Novel graph matrices for isomorphism detection:

The reliability of the characteristic polynomial of the Laplace matrix for isomorphism detection of a kinematic chain was established and determined to be comparable with that of the adjacency matrix. However, using the characteristic polynomials of both the matrices is superior to using either alone. In the search for a single matrix whose characteristic polynomial unfailingly detects isomorphism, novel matrices called the extended adjacency matrices were developed. The reliability of the characteristic polynomial of almost all of the extended adjacency matrices is similar to that of the standard adjacency matrix. Interestingly, the characteristic polynomial of one of the extended adjacency matrices, $A^{(2)}$, had unusually high reliability.

Objective 4: Reevaluation of algorithms for degeneracy testing and mobility type identification:

It was shown that almost all the mobility type identification and degeneracy testing algorithms are based on the assumption that the graph of a planar kinematic chain is a planar graph. This work mathematically proves several structural results and implicit assumptions about planar kinematic chains *without* using the planarity assumption. It is also shown that the implementation of Hwang and Hwang's degeneracy testing algorithm fails to identify some of the degenerate chains, and this could be the main reason for the discrepancy in the structural synthesis results of Hwang and Hwang. Furthermore, accuracy of Lee and Yoon's algorithms for both mobility analysis and degeneracy testing is proved by providing the mathematical justification of the individual steps. It was also shown that Lee and Yoon's algorithms for degeneracy testing and mobility analysis cannot be extended to spatial mechanisms (that satisfy the Gruebler's degrees of freedom equation) without additional structural results.

Objective 5: Polynomial-time algorithm for degeneracy testing:

This work developed an original polynomial-time algorithm for degeneracy testing by first reducing the problem to a 0-1 quadratic optimization problem with a single constraint. Next, the 0-1 quadratic optimization problem was further reduced to minimizing $|E|$ 0-1 quadratic functions. For each of the

0-1 quadratic optimization problems, a network was constructed such that the minimum cut vector of the network provided the minimum value. Finally the minimum cut problem was solved using Ford-Fulkerson & Edmonds-Karp maximum flow algorithm. The current degeneracy testing algorithm identifies all the planar kinematic chains with revolute joints correctly but it appears to be slower than Lee and Yoon's degeneracy testing algorithm for planar kinematic chains. The current algorithm also correctly identified degenerate planar kinematic chains with higher pairs and spatial kinematic chains. Since there are no other algorithms for degeneracy testing of planar kinematic chains with higher pairs or for spatial mechanisms, the comparison was done by brute force.

9.2 Relevance of Current Research

In mechanical engineering planar and spatial kinematic chains with up to 10 links have been used for various applications [1]. Automotive industry has used epicyclic gear trains with up to 10 links in automatic transmission mechanisms. The applicability of mechanisms with more than 10 links to traditional mechanical engineering applications has always been a debate. However, recent research in parallel robots [87], reconfigurable robots [88, 89] and multiple robotics [90] involve spatial kinematic chains with up to 14 links. Kinematic chains also appear in many applications outside of robotics, such as virtual prototyping, computer graphics, computational chemistry and biology [91, 92].

One significant application is in designing robots, which are modelled as closed spatial kinematic chains, for navigation in environment with obstacles [93, 94]. In this application, to avoid obstacles, the robots must be able to reconfigure in several different ways. This requires more than 12 links to give the necessary degrees of freedom. Here the synthesis techniques discussed in this work can be applied to generate different non-isomorphic chains. It should also be noted that degeneracy testing is extremely important in this scenario and hence the degeneracy testing algorithm developed in this work, for spatial kinematic chains, would be very helpful.

9.3 Contributions

The successful application of graph theory and combinatorial optimization to kinematic structural studies yielded many interesting results. The research contributions of this work are listed in this section along with the references to the derived publications.

1: This work proposes a new method for the synthesis of planar mechanisms, adapting a McKay-type algorithm [95].

2: This work establishes the reliability of the eigenvector approach for isomorphism detection [96].

3: This work develops a novel matrix of a kinematic chain whose characteristic polynomial detects structural isomorphism more reliably than existing methods [97].

4: This work provides rigorous proofs of structural results without using the planarity assumption [98].

5: This work provides for use the best existing degeneracy testing and mobility type identification algorithms [99].

6: This work proposes a novel polynomial-time algorithm for degeneracy testing of both spatial and planar kinematic chains, with planar or non-planar graphs, and with different kinds of joints [100].

This work is an example of the benefits achievable by applying recent mathematical techniques to classical engineering problems.

9.4 Future Work

The research described herein not only led to a broad set of results but also identified a number of specific questions for additional research.

1: Enumeration of non-isomorphic mechanisms with different types of joints.

Enumeration of non-isomorphic planar and spatial mechanisms with different types of joints parallels the enumeration of non-isomorphic edge colored graphs satisfying the non-degeneracy criterion. By using the homomorphism principle, generation of non-isomorphic edge colored graphs can be split into generation of non-isomorphic uncolored graphs and then generating the non-isomorphic edge colorings to each of the uncolored graphs.

2: Establishing the reliability and efficiency of spectral methods for isomorphism detection of mechanisms with different types of joints.

   Establishing the reliability and efficiency of spectral methods for isomorphism detection of planar and spatial mechanism with different types of joints will parallel the approach done in Chapter 5. As in Chapter 5 this would require a priori a database of non-isomorphic planar and spatial mechanism with different types of joints.

3: Developing structural results for spatial kinematic chains.

   Structural results of spatial kinematic mechanisms, similar to that of planar mechanisms, hold the key to success of efficient algorithms for structural analysis. However, two main difficulties exist. First, not all spatial mechanisms satisfy Gruebler's degrees of freedom equation, hence there need not be a linear relationship between the vertices and edges of the corresponding graph. Second, due to different types of joints that need to be considered, the corresponding graph would have to be an edge-colored graph. This would mean the best approach would be to use techniques from both graph theory and combinatorial optimization.

4: Developing degeneracy testing algorithm for spatial mechanisms not satisfying Gruebler's DOF equation.

   Spatial mechanisms exist that do not satisfy the Gruebler's DOF equation. Finding a formula for DOF for any spatial mechanism, similar to that of Gruebler's DOF equation, is a very active research area. Using the recent results from this area and combinatorial optimization approaches one may hope to find a polynomial-time degeneracy testing algorithm for a general mechanism.

5: Developing polynomial-time mobility type identification algorithm.

The polynomial-time degeneracy testing algorithm cannot be extended directly to mobility type identification algorithm (for kinematic chains satisfying Gruebler's DOF equation) as the minimum of degrees of freedom of the closed induced subgraphs need not be the same as minimum of degrees of freedom of all the induced subgraphs. Hence to develop a polynomial-time algorithm for mobility-type identification using optimization techniques, a result similar to Lemma 8.1 is required.

6: Determining the reason for the unusually high reliability of the $A^{(2)}$ matrix in isomorphism detection.

The characteristic polynomial of $A^{(2)}$ matrix showed unusually high reliability not only for kinematic chains but also for general graphs. So one can hope that there is a mathematical reason behind this unusual behavior. However, it should also be noted that this can be just a mere coincidence and the trend might not continue for kinematic chains with higher number of links.

7: Determining a pair of non-isomorphic kinematic chains that cannot be distinguished by the eigenvector approach.

This work shows that all the kinematic chains with 14 links or less can be identified by the eigenvector approach. However, there exist pairs of graphs that cannot be distinguished by the eigenvector approach as described in Chapter 5. Hence there is good chance that there also exists a pair of kinematic mechanisms that cannot be distinguished by the eigenvector approach.

8: Parallelizing the structural synthesis algorithms.

Due to the exponential-time nature of the synthesis algorithms the best way to yield new results is to parallelize the algorithms. Active research is being done in developing parallel algorithms for exhaustive generation of non-isomorphic structures. These new parallel algorithms can be adapted as in the case of this work to yield parallel structural synthesis algorithms.

It is hoped that researchers undertaking these challenges in future achieve the same level of intellectual satisfactions as the author of this dissertation and his advisor.

Appendix A

Mathematical Background

A.1 Graph Theory

This section describes basic graph theoretic concepts which would be used in structural analysis of kinematic mechanisms. A comprehensive treatment of graph theory can found in [101, 102].

A directed graph, $G = (V, E)$, is a pair of disjoint sets of vertices and edges together with the maps init: $E \to V$ and ter: $E \to V$ assigning to every edge an initial vertex init(e) and a terminal vertex ter(e) as shown in Figure A.1(b). The edge e is said to be directed from init(e) to ter(e). Two vertices a and b of graph, G are called adjacent if there exists an edge, e, such that init(e) $= a$ and ter(e) $= b$. A directed graph may have several edges between two given vertices and such edges are called multiple edges and if they are in same direction they are called parallel. If init(e) = ter(e) then the edge e is called a loop. A weighted directed graph, $G = (V, E)$, is a pair of disjoint sets of vertices and edges together with the maps init: $E \to V$, ter: $E \to V$ and wt: $E \to \mathbb{R}$ assigning to every edge an initial vertex init(e), a terminal vertex ter(e) and a weight wt(e) as shown in Figure A.1(a).

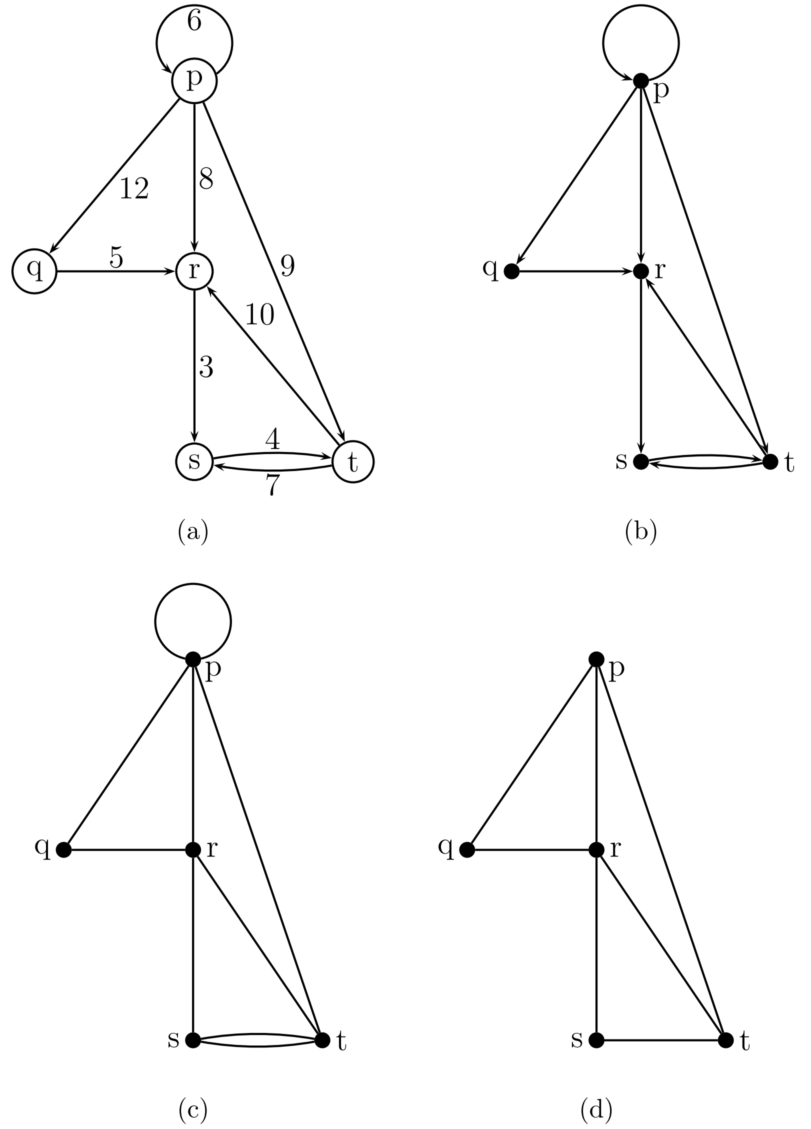
Let $[V]^k$ denote the set of all k-element subsets of V. A multigraph, $G = (V, E)$, is a pair of disjoint sets of vertices and edges together with the map $E \to V \cup [V]^2$ assigning to each edge either one or two vertices, its ends as shown in figure

A.1(c). Mulitgraph is a directed graph with no directions. Note that a multigraph can contain multiple edges and loops. A weighted multigraph can be similarly defined as above. A simple graph is a multigraph without loops or multiple edges as shown in Figure A.1(d). In other words a simple graph, $G = (V, E)$, is pair of disjoint sets of vertices and edges satisfying $E \subseteq [V]^2$.

Given a graph $G = (V, E)$ (belonging to one of the classes—weighted directed, directed, multi and simple—defined above), the vertex set of the graph, V, is denoted by $V(G)$ and similarly the edge set, E, by $E(G)$. There will not be strict distinctions between the graph and its vertex set or edge set.

A graph, G, is said to be connected if every vertex of G is connected to every other vertex of G by a path. A maximal connected subgraph of graph, G, is called a component of G. The graph shown in Figure A.2 is not connected and has 2 components, the induced subgraph on the vertices v_1, v_2, v_3, v_4 and v_5 form one component and the rest forms another component. An articulation point or cut vertex of graph is a vertex whose removal results in increase of the number of components. A bridge is an edge whose removal results in an increase of number of components. In Figure A.2 the vertices v_8, v_9 are the cut vertices and the edge $\{v_a, v_b\}$ is a bridge. A connected graph with no cut points is called a block.

Two graphs G_1 and G_2 are said to be isomorphic if there exists a one-to-one correspondence between their vertices such that the adjacency of the vertices is preserved. The graphs shown in Figure A.3 are isomorphic. Graph isomorphism is a classical problem in complexity theory.



(a) (b)

(c) (d)

Figure A.1: Graphs shown in the decreasing order of complexity (a) Weighted directed graph, (b) Directed graph, (c) Multigraph and (d) Simple graph

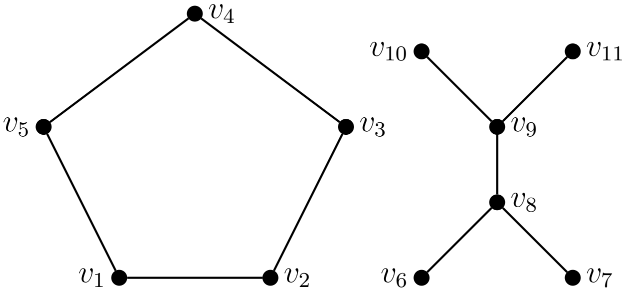
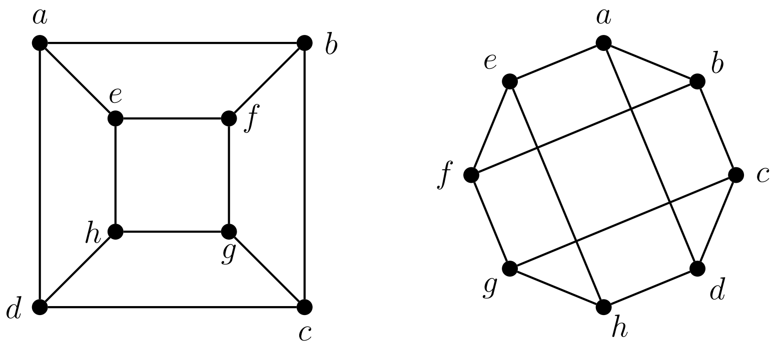


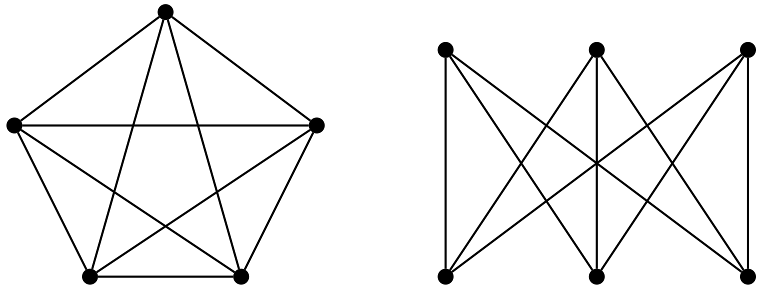
Figure A.2: Graph with 2 components

A graph, G, is said to be planar if it can be drawn on a plane surface such that no two edges intersect each other. A graph H is called a minor of the graph G if there exists a graph K such that both G and H can be obtained from K by a sequence of an edge sub-divisions (by inserting a vertex). A famous theorem in graph theory, Kuratowski's theorem, states that a graph G is planar if and only if it is not a minor of the complete graph on 5 vertices, K_5, or a complete bipartite graph on two sets of 3 vertices, $K_{3,3}$. The graphs, K_5 and $K_{3,3}$ are shown in Figure A.4.



(a) (b)

Figure A.3: A pair of isomorphic graphs



(a) (b)

Figure A.4: (a) K_5 (b) $K_{3,3}$

A.2 Some Concepts from Group Theory

A group $(G, *)$ is a nonempty set G together with a binary operation $* : G \times G \longrightarrow G$, satisfying the following group axioms. For notational convenience $*(a, b)$ is denoted by $(a * b)$.

1. Associativity: For all a, b and c in G, $(a * b) * c = a * (b * c)$.
2. Identity element: There is an element e in G such that for all a in G, $e * a = a * e = a$.
3. Inverse element: For all a in G, there is an element b in G such that $a * b = b * a = e$, where e is the identity element.

Given a group G and a set X, then a group action of G on X is a binary function $G \times X \longrightarrow X$ which satisfies the following two axioms. For notational convenience $.(g, x)$ is denoted by $(g.x)$.

1. $g.(h.x) = (g * h).x$ for all g, h in G and x in X.
2. $e.x = x$ for every x in X, where e is the identity element of G.

An orbit of X containing $x \in X$ under the action of group G is the set $Gx = \{g.x | g \in G\}$. A group G acting on X partitions X into disjoint orbits of X. The stabilizer of $x \in X$ is the subgroup $Stab_G(x) = \{g \in G | g.x = x\}$. Stabilizer of x is sometimes called the Automorphism group of x. For more information on group theory the reader is advised to consult texts by Lang [103] and Artin [104].

BIBLIOGRAPHY

[1] L.W. Tsai. *Mechanism Design: Enumeration of kinematic structures according to function.* CRC Press LLC, Boca Raton, 2000.

[2] T.S. Mruthyunjaya. Kinematic structure of mechanisms revisited. *Mechanism and Machine Theory*, 38:279–320, 2003.

[3] R. Read and D. Corneil. The graph isomorphism disease. *Journal of Graph Theory*, 1:339–363, 1977.

[4] J. Kobler, U. Schoning, and J.Toran. *The Graph Isomorphism Problem: Its Structural Complexity.* Birkhauser, Boston, 1993.

[5] S. Fortin. The graph isomorphism problem. Technical Report TR 96-20, Department of Computer Science, University of Alberta, Canada, 1996.

[6] D. Sen and T.S. Mruthyunjaya. The question of minimax loop-size in planar homogeneous kinematic chains. *Mechanism and Machine Theory*, 31:821–829, 1996.

[7] N.P. Belfiore. A breif note on the concept of planarity for kinematic chains. *Mechanism and Machine Theory*, 35:1745–1750, 2000.

[8] H.J. Lee and Y.S. Yoon. Detection of rigid structure in enumerating basic kinematic chain by sequential removal of binary link string. *JSME International Journal*, 35(4):647–651, 1992.

[9] H.J. Lee and Y.S. Yoon. Algorithm to identify the types of degrees of freedom in kinematic chains. *JSME International Journal*, 39(1):144–148, 1996.

[10] E.R. Tuttle. Generation of planar kinematic chains. *Mechanism and Machine Theory*, 31:729–748, 1996.

[11] Z.Y. Chang, C. Zhang, Y.H. Yang, and Y.X. Wang. A new method to mechanism kinematic chain isomorphism identification. *Mechanism and Machine Theory*, 37:411–417, 2002.

[12] P.R. He, W.J. Zhang, Q. Li, and F.X. Wu. A new method for detection of graph isomorphism based on the quadratic form. *Journal of Mechanical Design, Transactions of the ASME*, 125:640–642, 2003.

[13] J.P. Cubillo and J. Wan. Comments on mechanism kinematic chain isomorphism identification using adjacent matrices. *Mechanism and Machine Theory*, 40:131–139, 2005.

[14] D.G. Olson, A.G. Erdman, and D.R. Riley. A systematic procedure for type synthesis of mechanisms with literature review. *Mechanism and Machine Theory*, 20(4):285–295, 1985.

[15] F. Freudenstein and L. Dobrjanskyj. On a theory of type synthesis of mechansims. In *Proceedings of 11th International Congress of Applied Mechanics*, pages 420–428. Springer, Berlin, 1964.

[16] F.R.E. Crossley. *Developments in Theoretical and Applied Mechanics*, volume 2, chapter Permutations of kinematic chains with eight members or less from graph-theoretic view point, pages 467–486. Pregamon Press, Oxford, 1965.

[17] L.S. Woo. Type synthesis of plane linkages. *Journal of Engineering for Industry, Transactions of the ASME, Series B*, 89:158–172, 1967.

[18] N.I. Manolescu and I.Tempea. Structural synthesis of plane kinematic chains with multiple joints and five degree of freedom. *Revue Roumaine des Sciences Techniques, Serie de Mecanique Appliquee*, 12:1117–1140, 1967.

[19] N.I. Manolescu and I. Tempea. Structural and kinematic synthesis of plane driving mechanisms with multiple joints and two degrees of mobility. *Revue Roumaine des Sciences Techniques, Serie de Mecanique Appliquee*, 12:1251–1275, 1967.

[20] M. Huang and A.H. Soni. Application of linear and nonlinear graphs in structural synthesis of kinematic chains. *Journal of Engineering for Industry, Transactions of the ASME, Series B*, 95:525–532, 1973.

[21] E.R. Tuttle, S.W. Peterson, and J.E. Titus. Enumeration of basic kinematic chains using the theory of finite groups. *Journal of Mechanisms, Transmissions and Automation in Design, Transactions of the ASME*, 111:498–503, 1989.

[22] W.M. Hwang and Y.W. Hwang. Computer aided structural synthesis of planar kinematic chains with simple joints. *Mechanism and Machine Theory*, 27:189–199, 1992.

[23] M. Gruebler. Allgemeine eigenschaften der zwanglaufigen ebenen inematischen ketten: Part I. *Civilingenieur*, 29:167–200, 1885.

[24] M. Gruebler. Allgemeine eigenschaften der zwanglaufigen ebenen inematischen ketten: Part II. *Verh. Ver. Bef. Gew.*, 64:179–223, 1885.

[25] T.H. Davies and F.R.E. Crossley. Structural analysis of plane linkages by franke's condensed notation. *Journal of Mechanisms*, 1:171–184, 1966.

[26] F. Freudenstein. The basic concepts of polya's theory of enumeration with applications to structural classification of mechanisms. *Journal of Mechanisms*, 3:275–290, 1967.

[27] E.R. Tuttle, S.W. Peterson, and J.E. Titus. Further applications of group theory to the enumeration and structural analysis of basic kinematic chains. *Journal of Mechanisms, Transmissions and Automation in Design, Transactions of the ASME*, 111:494–497, 1989.

[28] N.I. Manolescu. A unitary method for construction of the kinematic plane chains and plane mechanisms with different degrees of mobility. *Revue Roumaine des Sciences Techniques, Serie de Mecanique Appliquee*, 9:1263–1313, 1964.

[29] N.I. Manolescu. Systematization and classification of plane jointed driving mechanisms with degree of mobility 'total' $M_3 = 2$ and 'partial' $M_{3p} = 2$. *Revue Roumaine des Sciences Techniques, Serie de Mecanique Appliquee*, 10:999–1042, 1965.

[30] N.I. Manolescu. For a united point of view in the study of the structural analysis of kinematic chains and mechanisms. *Journal of Mechanisms*, 3:149–169, 1968.

[31] C.R. Tischler, A.E. Samuel, and K.H. Hunt. Kinematic chains for robot hands–I: Orderly number synthesis. *Mechanism and Machine Theory*, 30:1193–1215, 1995.

[32] C.R. Tischler, A.E. Samuel, and K.H. Hunt. Kinematic chains for robot hands–II: Kinematic constraints, classification, connectivity and actuation. *Mechanism and Machine Theory*, 30:1217–1239, 1995.

[33] C.J. Colburn and R.C. Read. Orderly algorithms for graph generation. *International Journal of Computer Mathematics*, pages 167–172, 1979.

[34] T.S. Mruthyunjaya. Structural synthesis by transformation of binary chains. *Mechanism and Machine Theory*, 14:221–231, 1979.

[35] W.M. Hwang and Y.W. Hwang. An algorithm for the detection of degenerate kinematic chains. *Mathematical and Computer Modelling*, 15(11):9–15, 1991.

[36] L.C. Schmidt, H. Shetty, and S.C. Chase. A graph grammar approach for structure synthesis of mechanisms. *Journal of Mechanical Design, Transactions of the ASME*, 122:371–376, 2000.

[37] J.E. Hopcroft and J.K. Wong. A linear time algorithm for the isomorphism of planar graphs. In *Proceedings of the 6th Annual ACM Symposium On Theory of Computing*, pages 172–184. ACM Press, New York, 1974.

[38] A.C. Rao and P.B. Deshmukh. Computer aided structural synthesis of planar kinematic chains obviating the test for isomorphism. *Mechanism and Machine Theory*, 36:489–506, 2001.

[39] 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*, 3:87–100, 1968.

[40] T.S. Mruthyunjaya and M.R. Raghavan. Structural analysis of kinematic hains and mechanisms based on matrix representation. *Journal of Mechanical Design, Transactions of the ASME*, 101:488–494, 1979.

[41] T.S. Mruthyunjaya and M.R. Raghavan. Computer aided analysis of the structure of kinematic chains. *Mechanism and Machine Theory*, 19:357–368, 1984.

[42] V.P. Agrawal and J.S. Rao. Fractionated freedom kinematic chains and mechanisms. *Mechanism and Machine Theory*, 22:125–130, 1987.

[43] V.P. Agrawal and J.S. Rao. Structural classification of kinematic chains. *Mechanism and Machine Theory*, 22:489–496, 1987.

[44] V.P. Agrawal and J.S. Rao. Mobility properties of kinematic chains. *Mechanism and Machine Theory*, 22:497–504, 1987.

[45] F. Harary and H.S. Yan. Logical foundations of kinematic chains: graphs, line graphs, and hyper graphs. *Journal of Mechanical Design, Transactions of the ASME*, 112:79–83, 1990.

[46] F. Freudenstein and L.S. Woo. *Basic Questions of Design Theory*, pages 241–264. North Holland/American Elsevier, New York, 1974.

[47] T. Liu and C.H. Yu. Identification and classification of multi degree-of-freedom and multi-loop mechanisms. *Journal of Mechanical Design, Transactions of the ASME*, 117:104–111, 1995.

[48] A.C. Rao and V.V.N.R. Prasad Raju Pathapti. Loop based detection of isomorphism among chains, inversions and type of freedom in multi degree of freedom chain. *Journal of Mechanical Design, Transactions of the ASME*, 122:31–42, 2000.

[49] J.J. Uicker and A. Raicu. A method for identification and recognition of equivalence of kinematic chains. *Mechanism and Machine Theory*, 10:375–383, 1975.

[50] H.S. Yan and W.M. Hwang. A method for the identification of planar linkage chains. *Journal of Mechanisms*, 105:658–662, 1983.

[51] T.S. Mruthyunjaya. A computerized methodology for structural synthesis of kinematic chains: Part 1: Formulation. *Mechanism and Machine Theory*, 19:487–495, 1984.

[52] T.S. Mruthyunjaya. A computerized methodology for structural synthesis of kinematic chains: Part 2: Application to several fully or partially known cases. *Mechanism and Machine Theory*, 19:497–505, 1984.

[53] T.S. Mruthyunjaya. A computerized methodology for structural synthesis of kinematic chains: Part 3: Application to new case of 10-link three freedom chains. *Mechanism and Machine Theory*, 19:507–530, 1984.

[54] T.S. Mruthyunjaya and H.R. Balasubramanian. In quest of a reliable and efficient computational test for detection of isomorphism in kinematic chains. *Mechanism and Machine Theory*, 22:131–139, 1987.

[55] L.W. Tsai. An application of linkage characteristic polynomial to the topological synthesis of epicyclic gear trains. *Journal of Mechanisms, Transmissions and Automation in Design, Transactions of the ASME*, 109:329–336, 1987.

[56] R.K. Dubey and A.C. Rao. New characteristic polynomial: A reliable index to detect isomorphism between kinematic chains. In *Proceedings of the National Conference on Machine and Mechanism*, pages 36–40, IISC, Banglore, 1985.

[57] A.G. Ambekar and V.P. Agrawal. On canonical numbering of kinematic chains and isomorphisim problem: Max code. In *ASME Mechanisms Conference*, 1986.

[58] A.G. Ambekar and V.P. Agrawal. Canonical numbering of kinematic chains and isomorphism problem: Min code. *Mechanism and Machine Theory*, 22:453–461, 1987.

[59] C.S. Tang and T. Liu. The degree codea new mechanism identifier. *Journal of Mechanical Design, Transactions of the ASME*, 115:627–630, 1993.

[60] J.T. Kim and B.M. Kwak. An algorithm of topological ordering for unique representation of graphs. *Journal of Mechanical Design, Transactions of the ASME*, 114:103–108, 1992.

[61] J.K. Shin and S. Krishnamurty. On identification and canonical numbering of pin jointed kinematic chains. *Journal of Mechanical Design, Transactions of the ASME*, 116:182–188, 1994.

[62] A.C. Rao. Comparison of plane and spatial kinematic chains for structural error performance using pseudo- hamming distances. *Indian Journal of Technology*, 26:155–160, 1988.

[63] A.C. Rao and D. Varada Raju. Application of the hamming number technique to detect isomorphism among kinematic chains and inversions. *Mechanism and Machine Theory*, 26:55–75, 1991.

[64] F.F. Quist and A.H. Soni. Structural synthesis and analysis of kinematic chains using path matrices. In *Proceedings of the 3rd World Congress for Theory of Machines and Mechanisms*, pages D213–D222, 1971.

[65] A.C. Rao and C.N. Rao. Loop-based pseudo-hamming values. I: Testing isomorphism and rating kinematic chains. *Mechanism and Machine Theory*, 28:113–127, 1993.

[66] H.S. Yan and W.M. Hwang. Atlas of basic rigid chains. In *Proceedings of 9th Applied Mechanisms Conference*, pages 1.1–1.8, 1985.

[67] J.N. Yadav, C.R. Pratap, and V.P. Agrawal. Detection of isomorphism among kinematic chains using the distance concept. *Journal of Mechanical Design, Transactions of the ASME*, 117:607–611, 1995.

[68] C.H. Hsu and Y.C. Wu. Automatic detection of embedded structure in planetary gear trains. *Journal of Mechanical Design, Transactions of the ASME*, 117:117–122, 1995.

[69] B.D. McKay. Isomorph-free exhaustive generation. *Journal of Algorithms*, 26:306–324, 1998.

[70] G. Brinkmann. *Discrete Mathematical Chemistry*, volume 51 of *DIMACS Series on Discrete Mathematics and Theoretical Computer Science*, chapter Isomorphism rejection in structure generation programs, pages 25–38. AMS, Providence, RI, 2000.

[71] L. Lovasz. *Combinatorial Problems and Exercises.* North-Holland Publishing Co., Amsterdam, second edition, 1993.

[72] B.D. McKay. Nauty user's guide (version 1.5). Technical Report TR-CS-90-02, Australian National University, 1990.

[73] E.A. Butcher and C. Hartman. Efficient enumeration and hierarchical classification of planar simple-jointed kinematic chains: Application to 12- and 14-bar single degree-of-freedom chains. *Mechanism and Machine Theory*, 40:1030–1050, 2005.

[74] H.J. Lee and Y.S. Yoon. Automatic method for enumeration of complete set of kinematic chains. *JSME International Journal*, 37(4):812–818, 1994.

[75] C. Godsil and G. Royle. *Algebraic Graph theory*, volume 207 of *Graduate Texts in Mathematics.* Springer-Verlag, New York, 2001.

[76] N.L. Biggs. *Algebraic Graph theory*, volume 173 of *Cambridge Mathematical Library.* Cambridge University Press, Cambridge, second edition, 1993.

[77] D. Cvetkovic, P. Rowlinson, and S. Simic. *Eigenspaces of graphs*, volume 66 of *Encyclopedia of Mathematics and its applications.* Cambridge University Press, Cambridge, 1996.

[78] F.R.K. Chung. *Spectral Graph Theory.* American Mathematical Society, Providence, RI, 1997.

[79] R. Merris. Laplacian matrices of graphs: a survey. *Linear Algebra and its Applications*, 197-198:143–176, 1994.

[80] B. Korte and J. Vygen. *Combinatorial Optimization : Theory and Algorithms*, volume 21 of *Algorithms and Combinatorics.* Springer-Verlag, Berlin, 2002.

[81] C.H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization : Algorithms and Complexity*. Dover, 1998.

[82] J.C. Picard and H.D. Ratliff. Minimum cuts and related problems. *Networks*, 5:357–370, 1973.

[83] J.C. Picard and M. Queyranne. A network flow solution to some non-linear 0-1 programming problems, with application to graph theory. *Networks*, 12:141–159, 1982.

[84] J.C. Picard and M. Queyranne. Selected applications of minimum cuts in networks. *INFOR*, 20:394–422, 1982.

[85] G. Gallo, M.D. Grigoriadis, and R.E. Tarjan. A fast parametric maximum flow algorithm and applications. *SIAM Journal on Computing*, 18:30–55, 1989.

[86] J.E. Hopcroft and R. Tarjan. Efficient planarity testing. *Journal of the Association for Computing Machinery*, 21:549–568, 1974.

[87] J.P. Merlet. Still a long way to go on the road for parallel mechanisms. In *Proceedings of ASME International Mechchanical Engineering Congress and Exhibition*, 2002.

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

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

[90] K. Chang D. Ruspini R. Holmberg O. Khatib, K. Yokoi and A. Casal. Vehicle/arm coordination and multiple mobile manipulator decentralized cooperation. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 546–553, 1996.

[91] M. Kallmann, A. Aubel, T. Abaci, and D. Thalmann. Planning collition-free reaching motion for interactive object manipulation and grasping. In *Eurographics*, 2003.

[92] A.P. Singh, J.C. Latombe, and D.L. Brutlag. A motion planning approach to exible ligand binding. In *Proceedings of 7th International Conference on Intelligent Systems for Molecular Biology (ISMB)*, pages 252–261, 1999.

[93] D. Xie and N.M. Amato. A kinematics-based probabilistic roadmap method for high dof closed chain systems. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2004.

[94] A. Yershova and S.M. LaValle. Planning for closed chains without inverse kinematics. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2007.

[95] R.P. Sunkari and L. Schmidt. Structural synthesis of kinematic chains by adapting a McKay-type algorithm. To appear in *Mechanism and Machine Theory*, 2006.

[96] R.P. Sunkari and L. Schmidt. Reliability and efficiency of the existing spectral methods for isomorphism detection. To appear in *Journal of Mechanical Design, Transactions of the ASME*, 2006.

[97] R.P. Sunkari and L. Schmidt. Laplace and extended adjacency matrices for isomorphism detection of kinematic chains using the characteristic polynomial approach. In *CD-ROM proceedings of the ASME 2005 29th Mechanisms and Robotics Conference, Long Beach, CA*, 2005.

[98] R.P. Sunkari and L. Schmidt. Reevaluation of degeneracy testing and mobility type identification algorithms for planar kinematic chains with nonplanar graphs. Submitted to *Journal of Mechanical Design, Transactions of the ASME*, 2006.

[99] R.P. Sunkari and L. Schmidt. Critical review of existing degeneracy testing and mobility type identification algorithms for kinematic chains. In *CD-ROM proceedings of the ASME 2005 29th Mechanisms and Robotics Conference, Long Beach, CA*, 2005.

[100] R.P. Sunkari and L. Schmidt. A polynomial-time degeneracy testing algorithm for planar kinematic chains with higher pairs and spatial kinematic chains. Submitted to *Mechanism and Machine Theory*, 2006.

[101] F. Harary. *Graph theory.* Addison-Wesley, Reading, MA, 1969.

[102] R. Diestel. *Graph theory*, volume 173 of *Graduate Texts in Mathematics.* Springer-Verlag, New York, second edition, 2000.

[103] S. Lang. *Algebra*, volume 211 of *Graduate Texts in Mathematics.* Springer-Verlag, New York, third edition, 2002.

[104] M. Artin. *Algebra.* Prentice Hall, 1991.