USING RIGIDITY THEORY TO IDENTIFY HINGE JOINTS IN PROTEINS

Rittika Shamsuddin

This thesis was prepared under the guidance of Professor Audrey St. John

Presented to the faculty of Mount Holyoke College in partial fulfillment
of the requirements for the degree of Bachelor of Arts with Honors

Department of Computer Science South Hadley, Massachusetts

May 7, 2012

I give permission for public access to my thesis and for any copying to be done at discretion of the archives librarian and/or the College librarian.

May 9, 2012

....

Rittika Shamsuddin

Abstract

Computational biology uses tools from different fields, including computer science, to solve the challenges in the field of molecular biology and engages in the interpretation, classification and understanding of biological datasets. These datasets may involve RNA, DNA and protein sequences, comparison of genes and genomes of organisms, prediction of protein structures and creation of models for biological systems and molecules. Among these, building models for biological molecules, especially proteins, is of great importance.

Proteins have different functions as chemical receptors, messengers, catalysts, etc. Most of these tasks are achieved via conformational changes in the proteins. For some proteins, this change in conformation is compared to the opening and closing of doors. The proteins that undergo this open-close conformational change are called *hinge proteins* because they act as if they have relative *hinge joints* between the opening and closing bodies of the protein.

Since the conformational change in hinge proteins involves movement of different protein parts, we modeled proteins as structures understood by rigidity theory. While we hypothesized that this might lead to the identification of hinge proteins, we found that current rigidity theory models do not retain enough information to predict protein motion. In particular, rigidity theory does not model steric hindrance. Computational experiments indicate that the motions identified by the rigidity theory analysis may actually result in collisions between atoms.

This research was initially motivated by our successful analysis of mechanical models after which the proteins are modeled. Incorporating steric hindrance into our current protein model will most likely provide the additional information that we are missing.

Acknowledgment

I would like to thank my thesis advisor, Professor Audrey St. John, for giving me her time and bearing with me. I learned a lot, thanks to your guidance.

*

I want to thank Professor Craig Woodard for helping me find the right connections.

*

I want to thank Professor Margaret Robinson for teaching me the main concepts of linear algebra.

*

Thanks to Professor Harriet Pollatsek for lending me her book on Lie groups and algebra.

*

I also want to thank all my professors at the biology and computer science departments at Mount Holyoke College for their continuous support.

*

I also want to thank all Mount Holyoke College staff for taking care of all the tedious tasks (like taxes) for me.

*

I must also thank my elders for their great advice and support. Here I should also mention a small group of very special people for adding fun to my work .

*

Last, but not least, I thank all my friends for bearing with me, especially my two best friends and 'Vinsend Magiloa Oryeo' for their unconditional support and motivation.

Contents

Abstract **ii**

Acknowledgment **iii**

1 Introduction **1**

1.1 Problem Statement 2

1.2 Related Work 3

1.2.1 Hinge Identification: Experimental Approach 3

1.2.2 Hinge Identification: Computational Approach 4

1.2.3 Rigidity Theory and Proteins 7

1.3 Contribution and Result 8

1.4 Structure of this Thesis 8

2 Introduction to Protein Basics **10**

2.1 Structure of Proteins 11

2.1.1 Energy of a protein molecule 12

2.1.2 Levels of structure of a protein molecule 12

2.2 PDB Files: Textual representation of proteins 14

2.3 Hinge Proteins and Hinge Motion 14

3 Overview of Lie Groups and Lie Algebra 17

3.1 Why are Lie groups studied? 17

3.2 General Definition of a Group 18

3.3 Matrix Lie Groups 18

3.4 Sets of Matrices 19

3.5 Elements of O(n, $\mathbb{R}$) Preserves Distance 20

3.6 Calculus, Vector Space, Groups 21

3.7 Tangent Space and Lie Algebra 24

4 Rigid Body Motion 25

4.1 Dimension of $SO(n, \mathbb{R})$ and $SE(n, \mathbb{R})$ 26

4.2 Screws and Chasles's Theorem 28

4.2.1 Linking Elements of $se(3)$ and Screws 28

4.2.2 Linking $se(n, \mathbb{R})$ and Instantaneous Screws with 6-vectors 29

5 Infinitesimal Motion Space 30

5.1 Homogeneous Coordinate System 30

5.2 The Grassmann-Cayley Algebra 31

5.3 Calculating $Plücker\ Coordinates$ 31

5.3.1 $Plücker\ Coordinates$ of Lines in 3D 32

5.4 2-Tensors, Screws and $\mathbb{R}^6$ 32

6 Introduction to Rigidity Theory 34

6.1 Bar-and-Joint Structure 34

6.2 Frameworks and the Realization Problem 35

6.3 Introduction to Rigidity 36

6.3.1 Rigidity Analysis of Bar-and-Joint Structure 36

7 Rigidity Theory: Body-and-Bar/Hinge 40

7.1 Algebraic Rigidity 41

7.2 Infinitesimal Rigidity 42

7.3 Combinatorial Rigidity 43

8 Methodology for Analyzing Infinitesimal Motion Space 45

8.1 Infinitesimal Motion Space of a Pinned Framework 45

8.2 Relative Infinitesimal Motion Space of body j with respect to body i 47

8.3 Case by Case Analysis of the Relative Motion of a Body 48

9 Analyzing Infinitesimal Motion Space of CAD Models 51

9.1 SolidWorks 51

9.2 Methodology for CAD 51

10 Analyzing Infinitesimal Motion Space of a Model for a Protein 57

10.1 Software Tools 57

10.2 Investigating the Affect of Steric Hindrance 61

11 Conclusion and Future Work 70

List of Figures

1.1 Characteristics division of protein motion to time and amplitude scales. This schematic division is convenient for motion analysis and simulation. In reality, the timescale and amplitude of many of the described phenomena are far more disperse. This chart and caption is reproduced from [15] Note that the horizontal axis represents the timescale. . 2

1.2 The elastic network model. Reproduced from [39] 4

1.3 A simple harmonic system. mg is the weight of the object. Reproduced from [7] . . . 5

1.4 A sample hydrogen-bond dilution graph for protein cytochrome C. Flexible regions are shown by horizontal black lines, whereas each color represents a different rigid cluster. E is the energy cutoff that determines the number of hydrogen bonds to include and $< r >$ is the mean number of neighbors per atom. Reproduced from [16] 7

2.1 An amino acid monomer and a polymer of amino acids. Reproduced from [24] 11

2.2 Amino acids join together by forming peptide bonds. So, a single chain of amino acid is also called a *polypeptide*. Reproduced from [24] 11

2.3 The primary structure of the protein glycophorin A. The numbers 110 and 130 label the 110^{th} and 130^{th} amino acids in the sequence. Reproduced from [24] 12

2.4 A) Showing cartoonish ribbon representation of the helical structure in a protein. B) Showing the only backbone atoms of the protein. C) Showing the hydrogen and oxygen atoms, which are directly attached to the backbone atoms and are the only ones involved in forming this structure [3]. 13

2.5 D) Atoms and hydrogen atoms involved in forming the β-sheets. E) and F) are showing a simplified view of β-sheets.[3] 14

2.6 Tertiary structure of calmodulin as viewed in Jmol[2], using 1CFD as the PDB id. . . . 15

2.7 The quaternary structure of hemoglobin showing the four polypeptides that associate with each other to form the final working protein. Reproduced from [3] 15

6.1 A 2D bar-and-joint structure. 34

6.2 Rigid and flexible 2D bar-and-joint structures. If we attached the triangle over the flexible rectangle, we would have a framework that has both flexible and rigid components. The flex is usually determined by holding down or 'fixing' the bar between any two green nodes. 35

6.3 A bar-and-joint structure, with assignment P and its corresponding rigidity matrix. . . 38

7.1 A body-and-bar/hinge structure. Reproduced from [17] 40

7.2 A row from a rigidity matrix of a body-and-bar/hinge framework. 43

8.1 A figure showing how to obtain M_{pinned} from M. The colored part represents the original rows of M. 47

8.2 Extraction of instantaneous screws, describing the infinitesimal motion space of body j, from σ_{pinned} We are extracting the screws represented in the highlighted region. 48

9.1 Outline of the methodology for identifying joints in CAD models. 52

9.2 An example of defining a revolute joint in SolidWorks. Making the two green vertices and the two blue vertices coincide with each other will create a revolute joint between the cubes. The axis of rotation will be the line between the green and blue vertices. . . 52

9.3 Another example of defining revolute joint in SolidWorks. Making the two green edges (or lines) and the two blue vertices coincide with each other will create a revolute joint between the cubes. The axis of rotation will be the line defined by the green lines and blue vertices. 52

9.4 An example of defining a prismatic joint in SolidWorks. Making the two green edges parallel and the two blue edges coincide with each other will create a prismatic joint between the cubes. The of motion is shown by the arrow. 53

9.5 A second example of defining a prismatic joint in SolidWorks. Making the black and green edges perpendicular and the black and blue edges coincide will create a prismatic joint between the cubes. The direction of motion will be along the line of coincidence between the black and blue edges. 53

9.6 A sample XML file. 54

9.7 The pegboard model.Reproduced from [22]. 55

9.8 The constraint graph for the pegboard model, showing the different constraints used between different bodies. Reproduced from [22]. 55

10.1 Outline of the methodology for identifying hinge joints in protein models. 58

10.2 A graph demonstrating the changes in total energy of the system (in this case, the two atoms) as the distance between the atoms increases. Reproduced from the online english translation of [35]. 62

10.3 Outline of the methodology for investigating steric hindrance in protein. 62

10.4 Overall energy comparison for the modified conformations. The black line shows the overall energy of native conformation. The horizontal axis is the overall energy axis. . 65

10.5 Van der Waals repulsion. The black line shows the repulsion energy for the native conformation. 66

10.6 Van der Waals attraction. The black line shows the attraction energy for the native conformation. Note that the horizontal axis is much more expanded than in Figure 10.5. 67

10.7 Showing variation in the van der Waals attraction and repulsion among the samples on a more comparable scale. 68

10.8 This figure shows two charts, each of which shows variation in the different energy components. Note that the bar lengths in both charts form bands of almost uniform width. 69

Chapter 1

Introduction

Computational motion analysis is of great significance to engineers, physicists and biologists. In mechanics, motion of macroscopic 3D objects is studied in order to build mechanical structures, like bridges, to study their behavior, to characterize their performance and to further create new structures. The standard tools used in mechanics for motion analysis are: dynamics, the study of moving bodies; statics, the study of bodies at rest or constant motion; and kinematics, the study of motion without consideration for the forces causing the motion. Kinematics is also heavily used in CAD (Computer Aided Design). Motion analysis in CAD is required for generating computer simulations of products to ensure correct behavior. For example, manufacture of laptops should be preceded by model simulation to ensure that the laptop lid stays upright and closes smoothly as required.

In biology, studying the structure and motion of proteins, macromolecules of great biological importances, provides a greater insight into their functions. This insight leads towards the creation of powerful tools to control biological processes and hence towards the development of drugs or other cures for diseases. *Hinge proteins* and *hinge motion* are particularly important because they have been associated with substrate binding, catalysis and recognition of biomolecules, processes which lie at the heart of biological mechanisms. More details about proteins and its relevance to our research is given in Chapter 2.

1.1 Problem Statement

Proteins usually function by switching between two or more structural *conformations*. A conformation of a protein is the 3-dimensional coordinates of its atoms. Depending on the protein's current conformation, the protein can be described to be active(inactive) or open(close). To understand proteins' functions, we must understand the motion they undergo when changing their conformation. The range of protein motion is given by Figure 1.1.

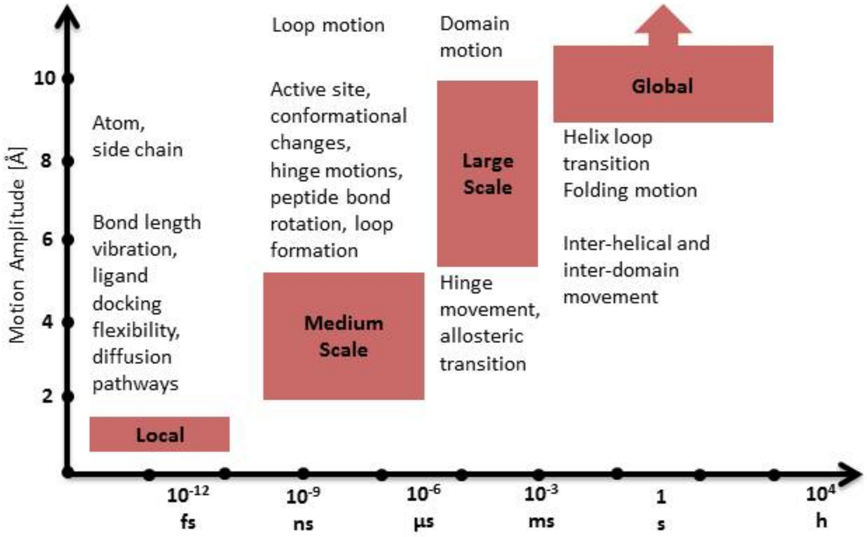


Figure 1.1: *Characteristics division of protein motion to time and amplitude scales. This schematic division is convenient for motion analysis and simulation. In reality, the timescale and amplitude of many of the described phenomena are far more disperse. This chart and caption is reproduced from [15] Note that the horizontal axis represents the timescale.*

Proteins can be classified according to their motions, which tend to correlate with particular functions. The class of proteins we focus on are called *hinge proteins*. These proteins contain *hinge joints*, which allow motion resembling the opening and closing of clams. Currently, there is no video capturing protein *hinge motion* because the timescale (usually in the scale of milliseconds), in which the motion occurs, is too small. Classical hinge protein identification happens through manual experimentation, which involves dedicating time (often in the order of years) to understand the protein structure. So we use computational approach to identify hinge proteins. Challenges arise when working with proteins, due to the complexity of the molecules, which are often composed of thousands of atoms. Proteins are very specific bio-chemicals and their functions usually depend on environmental factors. Standard motion simulation software that model a protein with accurate biochemistry are computationally too expensive for the simulation to be produced in a practical timescale. Thus, modeling a protein is an open research question.

The computational approach taken to identify hinge protein is defined in this paper as the **hinge protein identification problem**:

Given a text description of a protein, can we use only concepts from rigidity theory to computationally differentiate between hinge and non-hinge proteins?

1.2 Related Work

This section describes some of the experimental and computational works that have been done to identify hinge proteins. It also describes two applications of rigidity theory to proteins.

1.2.1 Hinge Identification: Experimental Approach

The structure of the periplasmic lysine/ariginine/arnithine-binding protein (which for brevity will be referred to as 2LAO) was studied by *Byung-Ha et al.* [4] via *X-ray crystallography*. X-ray crystallography is a mechanism that uses X-rays with very short wavelength (of order, 0.01 to 10 nm) to irradiate specially prepared crystalline protein. The spaces between atoms in the protein diffract the wave at different angles. X-ray sensitive photographic film is used to capture the X-ray diffraction pattern consisting of dots. This diffraction pattern gives rise to the electron density map, which when combined with relative position of the dots in the pattern provides information about arrangement of atoms in the crystal. Obtaining comprehensive information about the protein structure and motion often involves doing crystallography on multiple preparations of the protein.

Byung-Ha et al. looked mainly at the two different conformations of the 2LAO and compared them to analyze the motion that leads to the conformational change in the protein. They concluded that the motion is a “rigid body movement between one lobe with respect to the other” and is described by a rotation of 52° about a “virtual axis.” Note that the term lobe is used to describe each of two rigid clusters that make up the 2LAO structure.

1.2.2 Hinge Identification: Computational Approach

Since hinge proteins have been associated with important biological functions, they have been computationally studied before, and the two main systems that have been developed for identifying them are: HingeProt and StoneHinge. HingeProt uses both Gaussian Network Model (GNM) and Anisotropic Network models (ANM). StoneHinge uses a combination of constraint counting from rigidity theory and GNM. This section will give a brief overview of the work done by these two research groups to predict hinges in proteins.

HingeProt

HingeProt is a server that uses *elastic network models*, specifically GNM and ANM to predict hinges by analyzing vibration patterns of residues in a protein. GNM only produces scalar values that describe the fluctuations or vibrations and a special adaptation of GNM is required for the evaluation of the directions of these vibrations. This adaptation is called ANM. This thesis will only cover a brief overview of GNM.

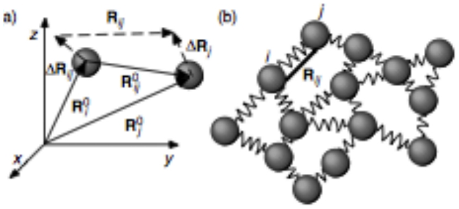


Figure 1.2: *The elastic network model. Reproduced from [39]*

Gaussian Network Model (GNM) In the elastic network model, as shown by Figure 1.2, a node is connected to its neighbors by springs. The nodes represent the residues of a protein, while the springs represent the covalent and non-covalent bonds. $\vec{R}^0_{ij}$ (shown by the arrow joining nodes i and j) represents the equilibrium position for nodes i and j (both are shown as metallic spheres in Figure 1.2a), while $\triangle\vec{R}_{ij}$ (shown by the horizontal dashed arrow) is the *instantaneous fluctuation vector* between the two nodes, and represents their displacement. In order to understand how to calculate the potential energy of this system, we need to know how to derive the potential energy of a *simple harmonic system*.

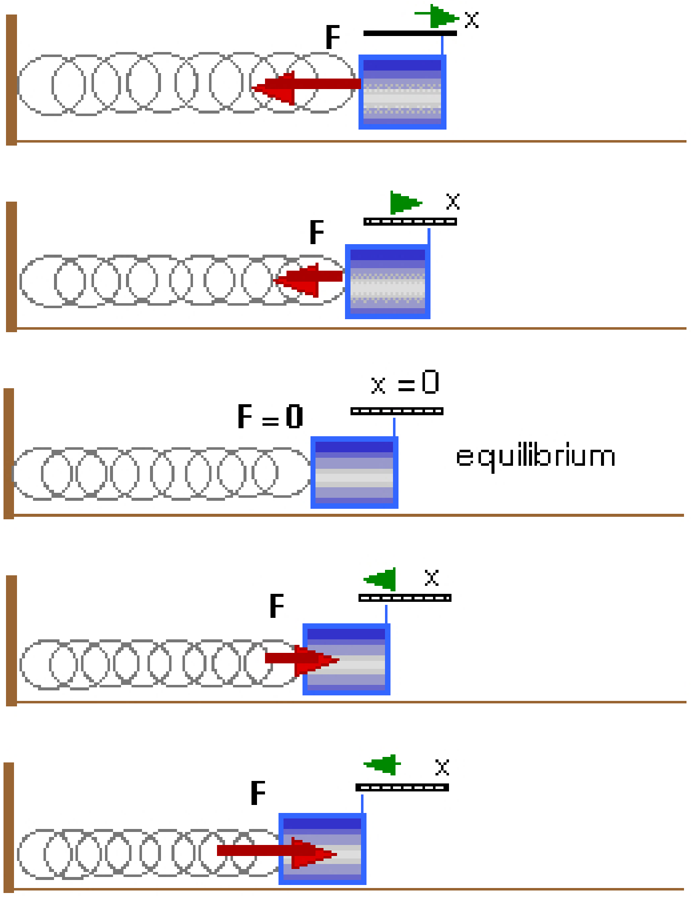


Figure 1.3: *A simple harmonic system. mg is the weight of the object. Reproduced from [7]*

A *simple harmonic system* is a system that undergoes *simple harmonic motion*. Simple harmonic motion is a type of periodic motion where the restoring force, F is directly proportional to the displacement. This force is given by $F = -kx$, where k is the spring constant and x is the displacement from equilibrium position. The potential energy, P_e of the system can be found by integrating the force function and so $P_e = \frac{1}{2}kx^2$ [7].

Similarly, the potential energy of a elastic network can be calculated as follows:

$$\frac{\gamma}{2}\left[\sum_{i,j}^{N} \triangle\vec{R}_i \Gamma_{ij} \triangle\vec{R}_j\right]$$

In the above equation, γ is the spring constant and is uniform for all network springs; N is the total number of nodes or residues; Γ_{ij} is the ij-th element of the *Kirchhoff* (or connectivity) matrix. The

Kirchhoff matrix holds information about the distance between two nodes or residues. If the distance between different nodes i and j is less than 7 angstrom (700pm), then Γ_{ij} is set to -1; if the distance is greater than 7 angstrom, then Γ_{ij} is set to 0. The diagonal entries are set to $-\sum_{j,j\neq i}^{N} \Gamma_{ij}$. The standard assumption is that residues will spatially interact if they are less than 7 angstrom apart.

The GNM *normal mode* can be found by *diagonalization* of Γ, and is written as, $\Gamma = U\Lambda U^T$ [12]. U is matrix of eigenvectors of Γ and whose inverse is equal to its transpose. Λ is diagonal matrix of eigenvalues of Γ. These eigenvalues give the frequency of each mode. The normal mode is a mode of vibration and is a characteristic pattern with which a mechanical system is vibrating. A few of the slow or low frequency modes refer to protein motions that have been shown to be "collective" and global and potentially relevant to function of proteins [25, 34]. Fast and high frequency modes describe motions, which do not seem to correspond to "relevant" changes in the structure. Experiments have shown that the residue fluctuations obtained from Nuclear Magnetic Resonance (NMR) strongly correlate with the presence of specific modes obtained from GNM. NMR is another technique often used to study chemical structure [13].

StoneHinge

Keating et al.[18] developed StoneHinge analysis for predicting hinges between *structural domains* of a protein (for definition of domain, see Chapter 2). It combines the result from two complementary analyses, StoneHingeP and StoneHingeD for a consensus to classify a protein as a hinge protein or not.

StoneHingeP StoneHingeP uses constraint counting results from ProFlex [5], and is similar to FIRST [9], to identify the residues between two *rigid clusters* in the protein (see Chapter 2). After proper processing of the protein, which involves removing *ligands* (for a full definition of ligands see [3]) and adding hydrogen atoms via a molecular package to optimize their position of hydrogen atoms, ProFlex calculates a *hydrogen-bond dilution graph* (see Figure 1.4). A hydrogen-bond dilution graph is a way to graphically represent rigidity of a protein. The rigid clusters are defined by bars of different colors. The residues present in these clusters are provided along the horizontal axis, as a range of residue numbers. The vertical axis reflects the energy change as a function of hydrogen bonds present in the protein. This allows StoneHingeP to calculate the optimal number of hydrogen bonds that would have to be present, in order to divide the protein into two rigid clusters. After the identification of these two rigid clusters, they identify the residues between the two domains as hinges.

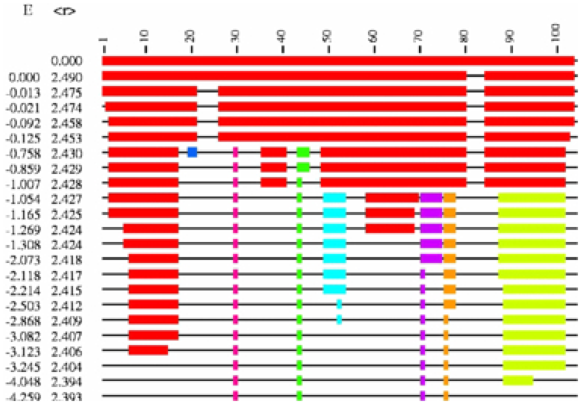


Figure 1.4: *A sample hydrogen-bond dilution graph for protein cytochrome C. Flexible regions are shown by horizontal black lines, whereas each color represents a different rigid cluster. E is the energy cutoff that determines the number of hydrogen bonds to include and* $< r >$ *is the mean number of neighbors per atom. Reproduced from [16]*

StoneHingeD StoneHingeD is similar to StoneHingeP but instead it uses a mechanism called DomDecomp [19]. DomDecomp is an adaptation of GNM to decompose proteins into structural domains. DomDecomp, like GNM, uses “modes” to analyse protein motion. StoneHingeD identifies residues that lie between residues with opening and closing modes. These identified residues are usually those that undergo the “least” motion, and the direction of their motion is the same as that of the large scale motion determined by the opening and closing modes. This effectively enables them to pinpoint a fixed point on what would be the *hinge axis*, between two consecutive residues.

StoneHinge identifies a protein as undergoing hinge motion only if the residues identified by StoneHingeD fall within five residues of a prediction made by StoneHingeP, because StoneHingeD has a tendency to over predict hinge residues.

1.2.3 Rigidity Theory and Proteins

Rigidity theory has been used by other research groups, which do not address the hinge identification problem, to study the rigidity of protein. Here we present two such softwares which are used in our research.

FIRST

FIRST [9] is an online program that uses rigidity theory to model proteins as abstract structures. Rigidity theory can then be directly applied to this abstract structure to identify rigid and flexible regions within the structure. FIRST can also be used to produce hydrogen-bond dilution graph (see Figure 1.4). More details about FIRST are given later in this paper.

KINARI

This is an online program and like FIRST, performs rigidity analysis on protein structures. However, KINARI is more user friendly because it allows the user to choose the model for the protein and also, provides greater control over input files. More information about KINARI is given in Chapter 10.

1.3 Contribution and Result

To the best of our knowledge, this is the first attempt to solve the identification of hinge proteins based solely on the analysis of protein structure via rigidity theory. Rigidity theory effectively captures geometric equality constraints. We began our analysis of identifying joints with CAD-models, because the geometric constraints in these models are well-defined. Using these models, we were able to develop an algorithm to computationally identify hinge joints[27]. Inspired by our success, we extended our analysis to proteins. Our results from analyzing protein models indicate that rigidity theory has limitations. We face the problem of accounting for steric hindrance in protein molecules, which involves accounting for collisions and unfortunately, classical rigidity theory allows collisions within a model. Note that collisions cannot be solved using equalities and represent a system of inequalities. We performed a case study on calmodulin[1], which is a hinge protein, to examine the significance of steric hindrance. The investigation with the case study concludes that collisions are important for analyzing protein motion.

1.4 Structure of this Thesis

The subsequent chapters describe the theory behind our analysis and provide the results with detailed discussion. Chapter 2 briefly discusses the protein structure providing foundations for the theoretical concepts used in this research. Chapters 3 through 7 focus on understanding the concepts behind rigidity theory. Chapter 3 gives a brief description of Lie groups and algebra. Chapter 4 uses Lie algebra and *screw theory* to describe motion in 3-space. Chapter 5 introduces the tools for describing the *infinitesimal motion space*, which is crucial for our analysis. Chapter 6 uses *2D bar-and-joint structures*, which are the simplest structures understood to explain the basic rigidity theory. Chapter 7 explains rigidity theory for *body-and-bar/hinge structures*, structures used to model proteins. Chapter 8 explains our general approach to identify hinges in both CAD and protein models. Chapters 9 and 10 focus on the specific methodology for CAD and protein models, respectively. Finally, Chapter 11 presents the conclusion of the thesis and the future directions for this research.

Chapter 2

Introduction to Protein Basics

Proteins can be seen as geometric structures. The geometry of proteins arises from the three dimensional structures they adopt and this structure is always unique to a protein. As such, proteins can be described to consist of three-dimensional *bodies* or domains. More importantly, their structure can be defined with a set of points described with three dimensions and it is important to look briefly at the biological aspects of proteins.

Three types of nutrients are essential as sources of energy for the human body: carbohydrates, fats and proteins. Carbohydrates are used as the main source of energy and fats as the main food storage by the human body. While all three are important for proper functioning, proteins are critical due to their wide range of functionalities. Proteins provide both structural and mechanical support to cells and tissues and are involved in repairing and building structures. As a matter of fact, proteins help to fight off germs and infections everyday, through the production of antibodies and are the main active constituents of the human immune system. All chemical reactions, occurring within cells to aid breathing, movement, sight, sense of smell and touch, require proteins to act as enzymes or hormones. Enzymes are biological catalysts which speed up or inhibit reactions and hormones are chemical messengers which coordinate bodily activities.

2.1 Structure of Proteins

Each cell in the human body has a specific function and a specific set of proteins to aid its function. Proteins are three dimensional macromolecules or large polymers of amino acids (see Figure 2.1 and Figure 2.2). The amino acids are usually small chemical molecules and join with one another via specific chemical reaction to form proteins. There are 20 different amino acids. Different proteins have different number of residues. For example, the TRP- Cage, extracted from the saliva of Gila monsters is 20 amino acids long, whereas Titin, responsible for passive elasticity in muscles, may constitute of more than 34 000 amino acids. The sequence of amino acids defines a protein and is very important for the protein function. The general structure of an amino acid is shown Figure 2.1.

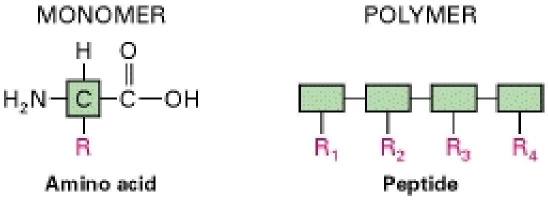


Figure 2.1: *An amino acid monomer and a polymer of amino acids. Reproduced from [24]*

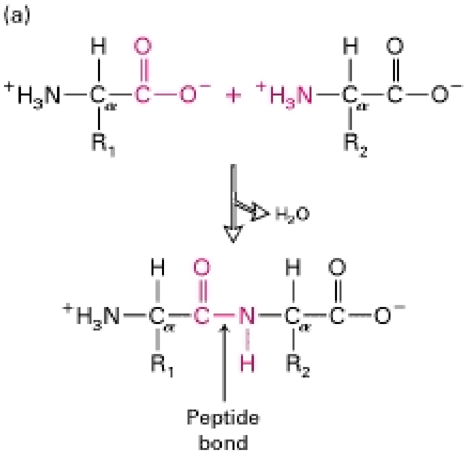


Figure 2.2: *Amino acids join together by forming peptide bonds. So, a single chain of amino acid is also called a polypeptide. Reproduced from [24]*

All 20 amino acids have this basic structure and differ from one another only in their identity of the R group or the side chain. The R group can be as simple as a hydrogen atom, H (as in glycine), or a methyl group, -CH3 (as in alanine), or can be more bulky. Based on the side chains, amino acids can be classified as being *hydrophobic* (water-fearing) or *hydrophilic* (water-loving). Thus, the unique side chains confer unique chemical properties on amino acids, and dictate how each amino acid interacts with the others within a protein.

2.1.1 Energy of a protein molecule

As with all other chemical molecules, proteins store *chemical energy* within them. This energy is a form of potential energy in chemicals, which changes every time a chemical reaction takes place. Since most chemical reactions take place to reduce the stored potential energy within the chemical reactants, chemical molecules have the tendency to transform to or assume a form that minimizes their potential energy because that makes them most stable. Thus in a normal healthy cell, a protein assumes a 3D structure or conformation that minimizes the energy. This conformation is called the *native conformation* of the protein.

2.1.2 Levels of structure of a protein molecule

The sequence of amino acid is called the *primary structure* of the protein (Figure 2.3). Even though the primary structure itself has no geometry to it, it is important for proper protein folding (through which the three dimensional structure arises). Even a change in a single amino acid can cause drastic changes to the three dimensional geometry of the protein. This is the reason why biologists refer to the amino acid sequence as a "structure."

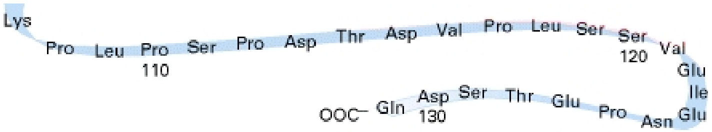


Figure 2.3: *The primary structure of the protein glycophorin A. The numbers 110 and 130 label the* 110^{th} *and* 130^{th} *amino acids in the sequence. Reproduced from [24]*

The *secondary structure* of the protein is defined by patterns of hydrogen bonds between the amine and carboxyl groups in the protein backbone. These patterns usually result in what is known as α-helices (Figure 2.4) and β-sheets (Figure 2.5). Usually the three dimensional structure of a protein will contain a combination of these two types of secondary structures. More importantly, these structures are usually regions of inflexibility i.e. they are usually rigid structures allowing very little movement, except for vibrations of atoms. However, a collection of these structures is not necessarily rigid.

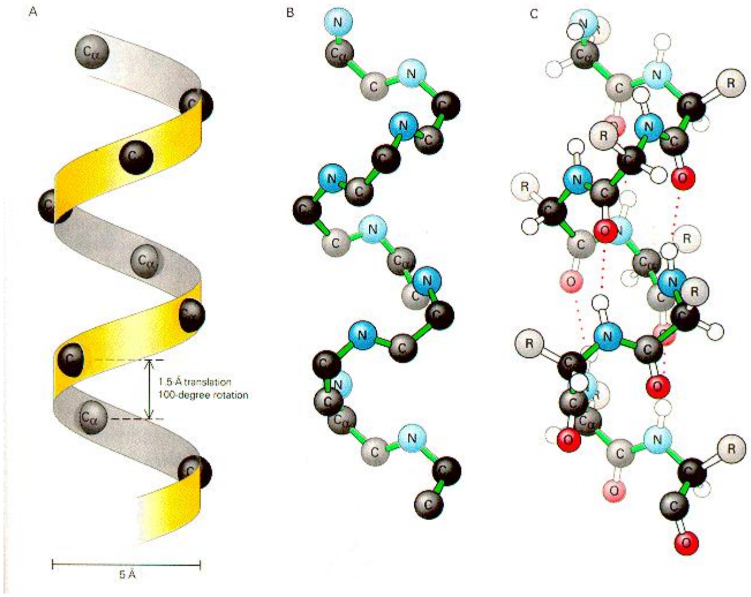


Figure 2.4: *A) Showing cartoonish ribbon representation of the helical structure in a protein. B) Showing the only backbone atoms of the protein. C) Showing the hydrogen and oxygen atoms, which are directly attached to the backbone atoms and are the only ones involved in forming this structure [3].*

The *tertiary structure* (shown in Figure 2.6) is the three-dimensional structure of a protein, which results from a large number of non-covalent interactions between the amino acids. The collective nature of these interactions, which usually occur between the side chains of the amino acids, is what stabilizes the protein. The tertiary structure allows the atoms in protein to be represented as a set of points with three dimensions.

The process of attaining the tertiary structure from the primary structure is referred to as *protein folding*. Sometimes, part of a protein sequence can fold by itself, independent of the rest of the protein, and assume a stable three-dimensional structure. This gives rise to *structural domains*. A protein can consist of one or multiple structural domains, which are linked together by covalent bonds.

Quaternary structure is not present in all proteins. Quaternary structure consists of multiple polypeptides (often referred to as *polypeptide chains*), held together by non-covalent interactions between them. Functional hemoglobin, depicted in Figure 2.7, is the protein that binds and carries oxygen in blood, and has a quaternary structure, which consists of two alpha globin and two beta globin polypeptides.

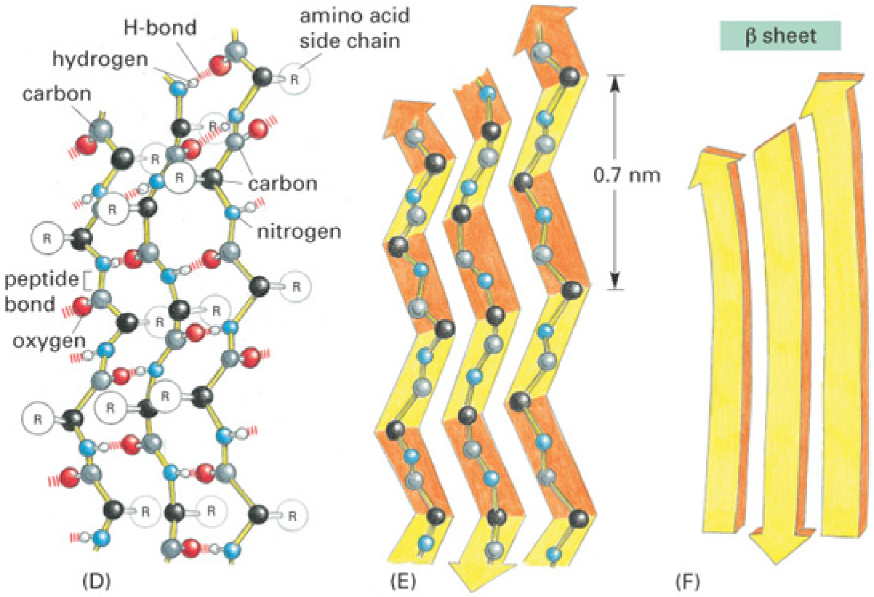


Figure 2.5: *D) Atoms and hydrogen atoms involved in forming the β-sheets. E) and F) are showing a simplified view of β-sheets.[3]*

2.2 PDB Files: Textual representation of proteins

The *Protein Data Bank (pdb) file format* is a textual file format describing the three dimensional structures of proteins and is in the *Protein Data Bank*. The Protein Data Bank (PDB) is a repository for the 3-D structural data of large biological molecules, such as proteins. Their information is typically obtained and submitted by biologists and biochemists using X-ray crystallography or NMR spectroscopy. The PDB files provide for description and annotation of proteins' structures including atomic coordinates, number of observed chains (or polypeptides), secondary structure assignments, and atomic connectivity. These files are often used as input files for computational programs.

2.3 Hinge Proteins and Hinge Motion

A flexible protein has two and sometimes more forms. These forms or *conformations* describe different geometric structures in terms of 3D coordinates for the atoms within the specified protein. A change from one conformation to another requires energy and is brought about by the movement of a structural domain of the protein in relation to another domain.

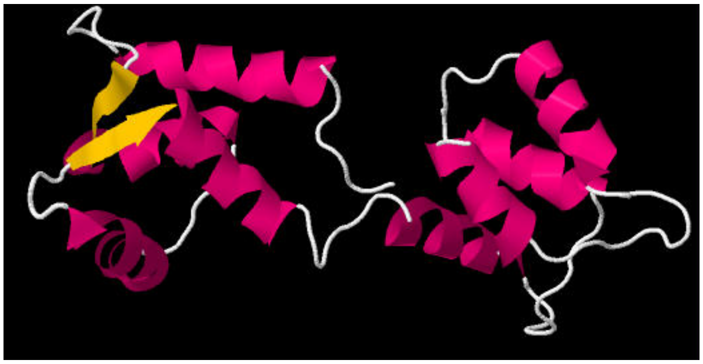


Figure 2.6: *Tertiary structure of calmodulin as viewed in Jmol[2], using 1CFD as the PDB id.*

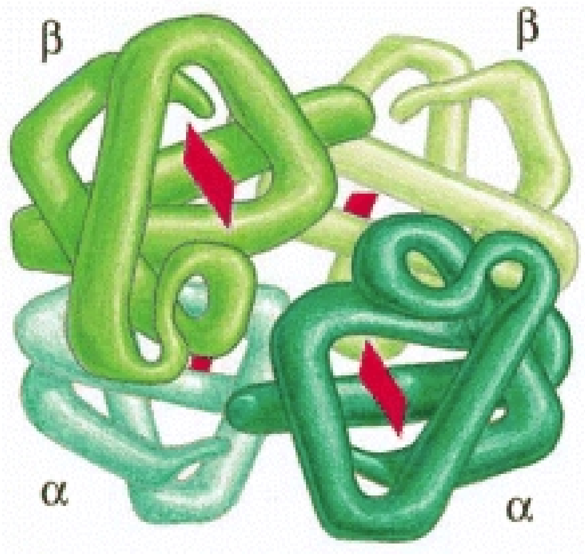


Figure 2.7: *The quaternary structure of hemoglobin showing the four polypeptides that associate with each other to form the final working protein. Reproduced from [3]*

Some proteins are known to undergo pivot-like motion. These proteins are called *hinge proteins* and contain rigid domains that undergo large-scale opening and closing. We define a *hinge axis* to be a fixed axis between two connected rigid domains, such that the pivot or hinge joint is found on this axis. In mechanical structures, most hinge joints allow rotation through a large (often obtuse) angle. However, a protein hardly undergoes such a huge change. A change in angle between domains by 52° is considered a large motion in a hinge protein.

In summary, the following analogy describes the behavior of hinge proteins: a laptop is composed of two rigid bodies, the lid attached to the keyboard. It can physically exist in two states: lid opened or lid closed. To switch between the two states, the lid (the first rigid body) needs to change the angle it makes with the keyboard (the second rigid body), but the angle is less than 360 degrees.

Chapter 3

Overview of Lie Groups and Lie Algebra

The next few chapters present background information on the theoretical concepts used for hinge protein identification problem. Lie groups and Lie algebra are important mathematical concepts, which are relevant for explaining rigidity theory. In this chapter, we give abbreviated materials found in [12] and [26] for containment and and closely follow these texts. Here we provide a general overview of Lie groups and Lie algebra, emphasizing areas relevant to this research and assuming familiarity with concepts from linear algebra.

3.1 Why are Lie groups studied?

Lie groups are complicated structures and are often useful in studying symmetries. Symmetry can be defined as invertible function, $f : X \to X$, such that X is a set and f preserves a certain feature of X. This feature of X can be shape, orientation, distance, interval etc. For example, take a square centered at the origin of $\mathbb{R}^2$ and apply a function, g to it, such that g causes the square to rotate through angles $0, \pi/2, \pi$ or $3\pi/2$; then g preserves the shape and orientation of the square.

When studying rigid bodies, we are interested in preserving the distance between two points within the rigid body because any change in the distance between those two points would mean the body is deforming.

3.2 General Definition of a Group

The following definition of a *group*, is taken directly from [26]:

A nonempty set G together with an operation * is a group provided,

1. The set is *closed* under the operation, i.e., if g and h belong to the set G, then so does $g * h$.
2. The operation is *associative*, i.e., if g, h, and k are any elements of G, then $g*(h*k) = (g*h)*k$.
3. There is an element e of G which is an *identity* for the operation, i.e., if g is any element of G, then $g * e = e * g = g$.
4. Every element of G has an *inverse*, which is also in G, i.e., if g is in G then there is an element of G denoted by g^{-1} satisfying $g * g^{-1} = g^{-1} * g = e$.

For a non-empty subset of G to be a **subgroup** of G, the subset needs to be a group itself.

3.3 Matrix Lie Groups

Even though not all Lie groups are matrix groups and not all matrix groups are Lie groups, matrix Lie groups come up often in scientific applications. This research uses one such application. The formal definition of *Lie group* is beyond the scope of this thesis but roughly speaking, a Lie group is a group or manifold [1] on which we can do calculus. A *matrix group* is a set of $n \times n$ matrices, where $n \epsilon \mathbb{Z}^+$ and the operation of the group is matrix multiplication. Since the identity element of the group is the identity matrix, I_n, the set of matrices must contain I_n. In addition, the set must also contain the inverse of all its elements and each element must be invertible. Thus a *matrix lie group* is a group that adheres to the properties of a Lie group and matrix group.

[1] A manifold is a topological space that is locally Euclidean. [28]

3.4 Sets of Matrices

Before proceeding further, note that this section focuses on *sets of matrices*, not *groups of matrices*. The following describes the notation used for sets of matrices.

- The set of *all* $n \times n$ matrices, where $n \epsilon \mathbb{Z}^+$, with real entries is denoted by $\mathcal{M}$(n, $\mathbb{R}$). This set is *not* a group because not all of its elements are invertible (i.e. these elements are matrices whose determinants are zero.)
- A proper subset of $\mathcal{M}$(n, $\mathbb{R}$), which is the set of all *invertible* $n \times n$ matrices is denoted by GL(n, $\mathbb{R}$) and is called the *general linear group*. It is defined as follows:

  $$GL(\text{n}, \mathbb{R}) = \{M \epsilon \mathcal{M}(\text{n}, \mathbb{R}) : \det(M) \neq 0 \}$$

- The **orthogonal group**, denoted by O(n, $\mathbb{R}$) is a subgroup of GL(n, $\mathbb{R}$) and is defined as follows:

  $$O(\text{n}, \mathbb{R}) = \{M \epsilon \mathcal{M}(\text{n}, \mathbb{R}) : M^T M = I_n \}$$

  Each matrix in O(n, $\mathbb{R}$) *preserves the distance between any pair of points*. The orthogonal group consists of two disconnected pieces: one piece contains matrices whose determinants are $+1$, including the identity matrix, and the second piece contains matrices whose determinants are -1 and does not include the identity matrix. O(2, $\mathbb{R}$), for example, is the group of all matrices that represent rotations and reflections in the xy- plane.

- The piece of O(n, $\mathbb{R}$), which contains the identity matrix is called the *special orthogonal group* and is defined as below:

  $$SO(\text{n}, \mathbb{R}) = \{M \epsilon O(\text{n}, \mathbb{R}) : \det(M) = 1 \}$$

  Thus SO(2, $\mathbb{R}$) is the part of O(2, $\mathbb{R}$) that contains rotational matrices only. We are interested in SO(n, $\mathbb{R}$) and the *special Euclidean group*, both of which are discussed in Chapter 4.

The matrix sets, described as groups in this section, are all matrix Lie groups.

3.5 Elements of O(n, $\mathbb{R}$) Preserves Distance

This section relates O(n, $\mathbb{R}$) to linear transformation matrix and starts by summarizing few concepts from linear algebra taken from [12].

Linear algebra reminders

- Every linear transformation, T, is associated with a matrix, M, such that multiplying M and a vector, $\vec{v}$, is equivalent to applying T to $\vec{v}$. Thus the language of matrices and language of linear transformations can be used interchangeably.

  This suggests that a set of linear transformations can form a group under **composition of functions**, and a particular group of linear transformation has a corresponding matrix group, such that the two groups are *isomorphic*.

- If $\vec{v}$ and $\vec{w}$ are column vectors, such that $\vec{v} = (v_1, v_2, ..., v_n)$ and $\vec{w} = (w_1, w_2, ...w_n)$, then their *dot product* is given by:

  $$\vec{v} \cdot \vec{w} = \vec{v}^T \vec{w} = (v_1 w_1 + v_2 w_2 + ... + v_n w_n)$$

- The *length* of $\vec{v}$, as measured from the origin is given by $\sqrt{\vec{v} \cdot \vec{v}}$.

- For any two points, V and W, in $\mathbb{R}^n$ and for any two vectors, $\vec{v}$ and $\vec{w}$, (also in $\mathbb{R}^n$), such that the vectors are directed along the line segments OV and OW respectively, where O is the origin,

  $$\text{distance between points } V \text{ and } W = \sqrt{(\vec{v} - \vec{w}) \cdot (\vec{v} - \vec{w})}$$

Using the rules stated above, it can be shown that a linear transformation, T, preserves distance if and only if T also preserves the dot product e.g.

$$T(\vec{v}) \cdot T(\vec{w}) = \vec{v} \cdot \vec{w} \text{ for all } \vec{v}, \vec{w} \,\epsilon\, \mathbb{R}^n$$

The group of linear transformations that preserves the dot product is denoted by $O(\mathbb{R}^n)$, which is isomorphic to O(n, $\mathbb{R}$) and hence is also called the orthogonal group. Thus, among the matrix groups, O(n, $\mathbb{R}$) and its subgroups (e.g. SO(n, $\mathbb{R}$)) preserve distance.

Why M^TM needs to equal to I_n for any matrix, M in O(n, $\mathbb{R}$)?

Suppose T is a linear transformation such that $T : \mathbb{R}^n \to \mathbb{R}^n$, preserves dot product and is represented by matrix M. Then, since for any two matrices $(AB)^T = B^TA^T$,

$$\begin{aligned} T(\vec{v}) \cdot T(\vec{w}) &= (M\vec{v}) \cdot (M\vec{w}) \\ &= (M\vec{v})^T(M\vec{w}) \\ &= \vec{v}^TM^TM\vec{w} \end{aligned}$$

Since T preserves dot product,

$$\begin{aligned} \vec{v} \cdot \vec{w} &= T(\vec{v}) \cdot T(\vec{w}) \\ &= \vec{v}^TM^TM\vec{w} \end{aligned}$$

However, since $\vec{v} \cdot \vec{w} = \vec{v}^T\vec{w}$,

$$\vec{v}^T\vec{w} = \vec{v}^TM^TM\vec{w}$$

Thus, if M is to preserve dot product and thus be an element of O(n, $\mathbb{R}$) , M^TM must be equal to I_n.

3.6 Calculus, Vector Space, Groups

What is a smooth function?

Previously we noted that by definition, we can do calculus on Lie groups and calculus can only be done on smooth functions. Generally speaking (definitions taken from [26]),

- A function $f : \mathbb{R} \to \mathbb{R}$ is smooth or infinitely differentiable if the nth derivative exists for all positive integers n.

- A function $f : \mathbb{R} \to \mathbb{R}^N$, with $f(x) = (f_1(x), ..., f_N(x))$ is smooth if all the component functions f_i are smooth.
- A function $F : \mathbb{R}^M \to \mathbb{R}^N$, with $F(x_1, ..., x_M) = (f_1(x_1, ..., x_M), ..., f_N(x_1, ..., x_M))$ is smooth if the partial derivatives of all orders exist for each component function f_i.

Realizing Matrices as Points in Euclidean Space Using Real Coordinates

The *Euclidean Space*, which is denoted as $\mathbb{R}^n$ in this thesis, is a n-dimensional *vector space*, where a point or vector is represented by n-tuple of real numbers and where the dot product gives distance information between points and angle information between two lines or vectors. A *vector space* is a collection of vectors, $\mathcal{V}$ such that for any $\vec{v}$ and $\vec{w}$ in $\mathcal{V}$,

- $\vec{v} + \vec{w} \,\epsilon\, \mathcal{V}$
- $r\vec{v} \,\epsilon\, \mathcal{V}$, where $r \,\epsilon\, \mathbb{R}$

Generally any $n \times n$ matrix can be seen as a point in $\mathbb{R}^N$, where $N = n^2$. For example a 3×3 matrix, M can be seen as a point in $\mathbb{R}^9$. By extending this idea we can view a pair of $n \times n$ matrices, (M, N) as a point in $\mathbb{R}^{2N}$, e.g. a pair of 3×3 matrices, (M, N) can be taken as a point in $\mathbb{R}^{18}$.

This means that every set of matrices described in section 3.4, can be seen as a *set of points*[2] in $\mathbb{R}^N$. Among the sets mentioned in section 3.4, only $\mathcal{M}(n, \mathbb{R})$ forms a vector space that is isomorphic to $\mathbb{R}^N$.

What does it mean for a matrix group operation to be differentiable?

Each group operation of a Lie group can be represented by a function (definitions taken from [26]). For the matrix Lie groups:

- **matrix multiplication** corresponds to a function, $f : \mathbb{R}^{2N} \to \mathbb{R}^N$ i.e., f takes in a pair of $n \times n$ (invertible) matrix as input and outputs one invertible $n \times n$ matrix.
- **matrix inversion** corresponds to a function $g : \mathbb{R}^N \to \mathbb{R}^N$ i.e., f takes in a single n x n invertible matrix as input and outputs another (invertible) $n \times n$ matrix.

[2] A set of points, where each point/vector is in a *vector space*, may not itself be a vector space if the set is not closed under vector addition and scalar multiplication.

For the matrix Lie group operations to be infinitely differentiable, these functions, f and g must be infinitely differentiable.

Parameterizing an element of a group

Suppose G is a group of $n \times n$ matrices described as follows:

$$G = \left\{ \begin{bmatrix} b_{11} & \dots & b_{1n} \\ \vdots & \ddots & \vdots \\ b_{n1} & \dots & b_{nn} \end{bmatrix} \quad : \quad b_{ij} \,\epsilon\, \mathbb{R} \right\}$$

The matrices in G can be generalized by writing each entry of $n \times n$ matrices in G as a function, $a_{ij} : \mathbb{R} \rightarrow \mathbb{R}$. This allows us to describe another function, $\alpha : \mathbb{R} \rightarrow G$ such that α maps real numbers to matrices in G, i.e.

$$\alpha(t) = \begin{bmatrix} a_{11}(t) & \dots & a_{1n}(t) \\ \vdots & \ddots & \vdots \\ a_{n1}(t) & \dots & a_{nn}(t) \end{bmatrix} \epsilon\, G$$

This means that for any real value of t, the matrix given by $\alpha(t)$ is in G. Therefore, α can be any one of infinite, different possible functions that outputs an element of G for each real value of t. In order to remember that α has infinite possibilities, it may be better to refer to a particular $\alpha(t)$ function as $\alpha_i(t)$ where i is a real number. In the study of Lie algebra, we are only interested in $\alpha_i(t)$ such that $\alpha_i(0) = I_N$, where I_N is the identity matrix (or the corresponding point in $\mathbb{R}^N$, which is referred to as the *identity point*).

G does not form a vector space in $\mathbb{R}^N$; it it forms a *s*mooth manifold in $\mathbb{R}^N$ and each function $\alpha_i(t)$, such that $\alpha_i(0) = I_N$, forms a curve, passing through the identity point on the surface G in $\mathbb{R}^N$. This curve is said to be a smooth curve, if each of its entries, $a_{ij}(t)$, is a smooth function.

3.7 Tangent Space and Lie Algebra

As described previously, G is a smooth manifold in $\mathbb{R}^N$, and we consider infinite number of curves, $\alpha_i(t)$ passing through the identity when $t = 0$. If we were to differentiate all $\alpha_i(t)$ at $t = 0$, such that

$$\alpha'(0) = \begin{bmatrix} a'_{11}(0) & \dots & a'_{1n}(0) \\ \vdots & \ddots & \vdots \\ a'_{n1}(0) & \dots & a'_{nn}(0) \end{bmatrix},$$

then each $\alpha_i(0)$, will give a vector in $\mathbb{R}^N$, and each one of these vectors will be tangent to surface G at the identity point. These vectors together form a subspace, which we call the **tangent space**, of G. Note this means that the tangent space of the group is a vector space. Thus a tangent space is a much simpler *mathematical structure* than a group and hence easier to study. However, this tangent space is a little too simple and does not retain enough information for referring back to the group. So mathematicians add a little more mathematical structure[3] to the tangent space, turning it into Lie algebra. The resulting Lie algebra encodes necessary information about the original Lie group. The most important concepts from Lie groups and Lie algebra that will be used in this thesis is: the dimension of a Lie Group is the same as the dimension on its associated Lie Algebra and every Lie group has an associated lie algebra, which forms a vector space.

[3] This is usually done by adding more operations to the vector space other than vector addition and scalar multiplication.

Chapter 4

Rigid Body Motion

This chapter uses concepts presented in the last chapter and lays the ground work for connecting a specific Lie algebra to instantaneous motion of rigid bodies. We closely follow [29] and include this material for containment.

A rigid body is a structure such that the distance between any two points, contained within the body, does not change when the body is in motion. Naturally, the only two types of motion a rigid body can undergo are: rotation, where the whole body moves around an axis and translation, where the whole body moves along the axis. For example, in $\mathbb{R}^2$, a rigid body can translate along the x and the y axes and rotate about the origin; in $\mathbb{R}^3$, a rigid body can rotate about or translate along the x, y or z axes. More technically, the group of rigid body transformations is the Lie group of linear transformations of $\mathbb{R}^n$ that preserve the distance between two points (in the rigid body) or rather preserve the dot product of two vectors and cause the points to either translate or rotate. If the points embedded in the rigid body are treated as vectors, then rigid body motion could be achieved by adding a constant vector, $\vec{x}$, to the rigid body vectors and by multiplying the vectors with orthogonal matrices. This can be expressed using the following matrix equation, where M is an orthogonal matrix and $\vec{x}$ is a constant vector, both acting on $\vec{w}$; $\vec{w'}$ is the transformed vector:

$$\vec{w'} = M\vec{w} + \vec{x}$$

Thus the above equation can be written as $(M, \vec{x})$,where $M \,\epsilon\, SO(n, \mathbb{R})$ and $x \,\epsilon\mathbb{R}^n$.

Thus, the group of rigid body motion in $\mathbb{R}^n$ is a Lie group and a *semi-direct product* of the special orthogonal group, $SO(n, \mathbb{R})$ and $\mathbb{R}^n$. *Semi-direct product* intuitively means combining two different subgroups to form another group, where each element of the group formed is an unique product of elements of the two subgroups. The semi-direct product of $SO(n, \mathbb{R})$ and $\mathbb{R}^n$ is called the **special Euclidean group** and is denoted by:

$$SE(n, \mathbb{R}) = SO(n, \mathbb{R}) \ltimes \mathbb{R}^n$$

For convenience, we write $(M, \vec{x}) \, \epsilon \, SE(n, \mathbb{R})$ as:

$$\begin{bmatrix} M & \vec{x} \\ 0 & 1 \end{bmatrix},$$

where $M \, \epsilon \, SO(n, \mathbb{R})$ and $\vec{x} \, \epsilon \, \mathbb{R}^n$. Since the rigid bodies are three dimensional, we are concerned with $SE(3, \mathbb{R})$and use 4×4 matrices. This representation is also called *homogeneous representation* (please refer to Chapter 5) and does not change the property of $(R, \vec{x})$ [29].

4.1 Dimension of $SO(n, \mathbb{R})$ and $SE(n, \mathbb{R})$

Recall that $SO(n, \mathbb{R})$ and $SE(n, \mathbb{R})$ are both matrix Lie groups and have associated Lie algebras, denoted by $so(n, \mathbb{R})$ and $se(n, \mathbb{R})$ respectively. Since $SO(n, \mathbb{R})$ is a subgroup of $O(n, \mathbb{R})$, for all elements, M in $SO(n, \mathbb{R})$,

$$\tfrac{d}{dt} M(t)^T + \tfrac{d}{dt} M(t) = 0_n$$

(found by differentiating $M(t)^T M(t) = I_n$, at t = 0) [29]. Thus, if $so(n, \mathbb{R})$ is the Lie algebra associated with $SO(n, \mathbb{R})$ then we can define $so(n, \mathbb{R})$ as

$$so(n, \mathbb{R}) = \{A \, \epsilon \, \mathcal{M}(n) : A^T + A = 0_n\}$$

By definition, A is a *skew-symmetric* or equivalently, an *antisymmetric* matrix and is written as:

$$A = \begin{bmatrix} 0 & a_{12} & \dots & a_{1n} \\ -a_{12} & 0 & & \vdots \\ \vdots & & \ddots & a_{(n-1)n} \\ -a_{1n} & \dots & -a_{(n-1)n} & 0 \end{bmatrix}$$

The *dimension of a general* $n \times n$ *matrix* (which is not the same as the dimension of the row or column space associated with the matrix) is n^2, which is equal to the number of entries in the matrix. The dimension of a matrix can be seen as the total count of *empty slots that holds new information*.

Since A is a skew matrix, its dimensions can be found by:

1. Subtracting n from n^2 because there will be n zeros on the diagonal of the matrix.
2. Dividing $n^2 - n$ with 2 because the value of a_{12}, will be the negation of a_{21}.

Thus the dimension of $so(n, \mathbb{R})$ is

$$\text{dim}(so(n, \mathbb{R})) = \frac{n^2-n}{2} = \frac{n(n-1)}{2}$$

Since $SE(n, \mathbb{R})$ is also a matrix Lie group, it too can be parameterized using t, such that

$$\alpha(t) = \begin{bmatrix} M(t) & \vec{x}(t) \\ 0 & 1 \end{bmatrix}.$$

Differentiating this function at $t = 0$ we get

$$se(n, \mathbb{R}) = \left\{ \begin{bmatrix} M'(0) & \vec{x}'(0) \\ 0 & 0 \end{bmatrix} \quad : \quad M'(0) \,\epsilon\, so(n, \mathbb{R}), \vec{x}'(t) \,\epsilon\, \mathbb{R}^n \right\}$$

$M(t)$ has dimension $\frac{n(n-1)}{2}$ and $\vec{x}(t)$ has dimension n. Thus,

$$\dim(se(n,\mathbb{R})) = \tfrac{n^2-n}{2} + n = \tfrac{n(n+1)}{2} \ .$$

4.2 Screws and Chasles's Theorem

Screws

The theory of *screws* was introduced in [6] to describe helical motion. A helical motion is a rotation about a line, called the *screw axis*, together with a translation along the line. The mathematical representation of screw can be found in [29].

Chasles's Theorem. *All proper rigid body motions in 3-dimensional space, with the exception of pure translations, are equivalent to a screw motion.*

Even though this theorem does not state that a pure translation can be represented as a screw motion, it is known that screw motion can be used to represent rigid body displacement (including translation). The explanation and the proof of this theorem is outside the scope of this thesis but can be found in [29].

4.2.1 Linking Elements of $se(3)$ and Screws

A smooth curve in $SE(3)$ is a parameterized sequence of rigid transformations. Applying the sequence to some rigid body gives a smooth movement of the body through space. If the parameter t is taken to be time, then the derivative of the parameterized representation of $SE(3)$ will give the velocity. Velocities of rigid bodies are therefore, essentially, elements of Lie algebra $se(3,\mathbb{R})$ and for convenience is rewritten as:

$$se(3,\mathbb{R}) = \left\{ \begin{bmatrix} \Omega & \tau \\ 0 & 0 \end{bmatrix} \quad : \quad \Omega \,\epsilon\, so(3,\mathbb{R}), \tau \,\epsilon\, \mathbb{R}^n \right\}$$

Since Ω is skew-symmetric matrix, for any vector $\vec{\omega} \,\epsilon\, \mathbb{R}^3$, such that $\omega = (\omega_x, \omega_y, \omega_z)$, we can write,

$$\Omega = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$

so that, for any vector $\vec{x}$ in $\mathbb{R}^3$, $\Omega\vec{x} = \vec{\omega} \times \vec{x}$, where $\times$ represents the cross product. Also, for any point, p on a rigid body, such that $\vec{p} = (x, y, z)^T$, the instantaneous velocity of p, $\vec{v}$ is given by

$$\vec{v} = \Omega\vec{p} + \tau.$$

More importantly, it can be shown that the Lie algebra elements $se(3, \mathbb{R})$ are instantaneous screws and can be used to represent instantaneous screw motions.

4.2.2 Linking $se(n, \mathbb{R})$ and Instantaneous Screws with 6-vectors

According to the dimension formula of $se(n, \mathbb{R})$ obtained in this chapter,

$$\text{dim}(se(3, \mathbb{R})) = \tfrac{3(3+1)}{2} = 6.$$

This means that $se(3, \mathbb{R})$ elements can be represented using 6 parameters. Also, since screws are identifiable with elements of $(se(3, \mathbb{R}))$, we can associate a screw, s with a 6-vector, such that:

$$s = (-\vec{\omega}, \vec{\tau})^1 = (-\omega^x, -\omega^y, -\omega^z, \tau^x, \tau^y, \tau^z).$$

Thus $\mathbb{R}^6$ can be identified with the space of instantaneous screws and identified with $se(3, \mathbb{R})$.

[1] The negative sign in from of is necessary for interpretation of the result from the Grassmann-Cayley algebra, as we will see later.

Chapter 5

Infinitesimal Motion Space

This chapter presents another way to describe infinitesimal motion of rigid bodies using Grassmann-Cayley algebra. This approach is studied, in conjunction with screws and $se(3, \mathbb{R})$ elements, to define infinitesimal motion because it provides a convenient way to distinguish between instantaneous screws describing a pure rotation, pure translation or a combination of both.

Grassmann-Cayley algebra is a form of modeling algebra for use in projective geometry. The technique uses subspaces as basic elements of computation, which translates geometric statements into algebraic statements. The exact nature of the Grassmann-Cayley algebra is outside the scope of this thesis (see [36, 37]).

5.1 Homogeneous Coordinate System

This is a different coordinate system and is used in projective geometry. Its advantage over the Cartesian coordinates lies in the fact that it can represent a point in infinity with finite coordinates. The notation for writing a homogeneous coordinate is $[x; y; z]$. Instead of representing each point (x, y, z) in $\mathbb{R}^3$ with a single 3-vector, homogeneous coordinates allow each point (x, y, z) to be represented by any of one of the infinite number of 4-vectors: $(\varphi x, \varphi y, \varphi z, \varphi)$, where φ is a real number.

5.2 The Grassmann-Cayley Algebra

The Grassmann-Cayley algebra is defined by bracket ring. The 'bracket' in bracket ring actually refers to a determinant of a matrix. Grassmann-Cayley algebra [36, 37] has two operators: *join* and *meet*. Intuitively, join is analogous to union of sets and meet is analogous to intersection of sets. For example, the join of two points is a line and the meet of two intersecting non-identical lines is a point. These operators give rise to extensors and tensors. The join of k linearly independent points defines a k-extensor. The join is denoted by:

$$a_1 \vee a_2 \vee \cdots \vee a_k, \text{ where } \vee \text{ is the join operator and } a_i \text{ is a point.}$$

A *k-tensor* is the sum of k-extensors. If a k-tensor is itself a k-extensor, it is referred to as a *decomposable tensor*; otherwise, it is an *indecomposable tensor*.

5.3 Calculating $Plücker\ Coordinates$

To calculate the $Plücker$ coordinates of k points in $\mathbb{R}^d$:

i. We start with k points denoted by $a_1, a_2, ...a_k$.

ii. We then make a matrix, M, where each row is one of the k points in its homogeneous coordinate form. M is a $k \times (d+1)$ matrix.

iii. Computing all the $k \times k$ minors or determinants of M gives us the $Plücker$ coordinates. So the number of coordinates that arises from k points is $\binom{d+1}{k}$.

An example of this calculation can be found in [8]. The example shows that there are various ways of writing down or arranging the minors and we can choose the one that is most convenient for interpretation.

5.3.1 $Plücker\ Coordinates$ of Lines in 3D

A line in $\mathbb{R}^3$ is defined by two points. If we take the join of these two points, we get a 2-extensor, and their *span*[1] is described by $\binom{4}{2} = 6$ $Plücker$ coordinates. Hence, the $Plücker$ coordinates of a line in 3D is a 6-dimensional vector. For convenience, the order of minors is chosen to be $(M_{14}, M_{24}, M_{34}, M_{23}, -M_{13}, M_{12})$, where M_{ij} denotes a minor and i and j denotes column number in M.

However, not every 6-vector can be identified with the $Plücker$ coordinates of a line. Only those 6-vectors, which satisfy the $Grassmann$-$Plücker$ relation can be identified with lines in 3D. There are an infinite number of 6-vector representations of a specific line in 3D (since scalar multiples of the vector preserve the line). The $Grassmann$-$Plücker$ relation is described as follows:
Let $s = (a, b, c, d, e, f)$ and $s^* = (d, e, f, a, b, c)$. The relation is satisfied if and only if the dot product of s and s^* is zero, e.g., $s \cdot s^* = 0$.

The collection of these 6-vectors satisfying the $Grassmann$-$Plücker$ relation forms a surface in $\mathbb{R}^6$ (this is thought of as tracing a manifold in a higher dimension). This surface is referred to as the $Grassmannian$ and is very crucial to our analysis as explained in the next section.

5.4 2-Tensors, Screws and $\mathbb{R}^6$

The closure of 2-extensors under the addition operator is all of $\mathbb{R}^6$, i.e., 2-tensors are identified with $\mathbb{R}^6$. However, only the decomposable 2-tensors (which are actually 2-extensors) satisfy the $Grassmann$-$Plücker$ relation. Indecomposable 2-tensors form the rest of $\mathbb{R}^6$.

In Chapter 4, we identified $\mathbb{R}^6$ with instantaneous screws. Thus it is implied that instantaneous screws are identified with 2-tensors. All the decomposable 2-tensors represent screws which encode for either a pure rotation or a pure translation and the indecomposable 2-tensors are the sum of a pure rotational and a pure translational' 2-extensor. Since screws can be identified with 2-tensors, any instantaneous screw describing a pure rotation or translation will follow the $Grassmann$-$Plücker$ relation. Thus, the

[1] A span is a space.

$Grassmannian$ represents pure rotations and translations only. Also, recall that screws and $se(3, \mathbb{R})$ elements are used to describe $\mathbb{R}^6$. Hence, $\mathbb{R}^6$ can be identified with 2-tensors.

Chapter 6

Introduction to Rigidity Theory

Established mathematical structures are used to study the rigidity properties of physical structures. In this chapter we introduce the tools required to understand these mathematical structures and set the basis of rigidity theory.

6.1 Bar-and-Joint Structure

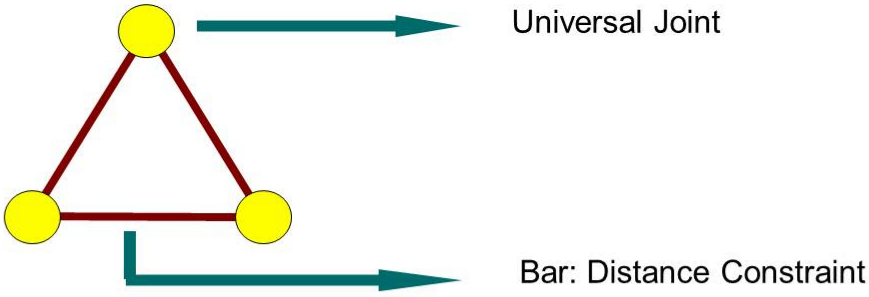


Figure 6.1: *A 2D bar-and-joint structure.*

Figure 6.1 shows an example of a 2D bar-and-joint structure. These are structures on which we can apply rigidity theory. Generally, bar-and-joint structures (in n-dimensional space) are composed of universal joints with distance constraints between them. The distance constraints are referred to as bars. Figure 6.2 gives an intuition about classifying a 2D bar-and-joint structure as either rigid or flexible.

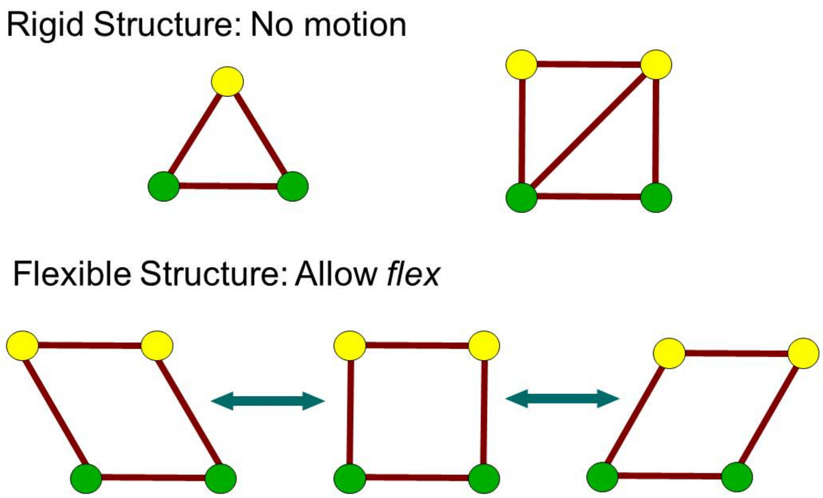


Figure 6.2: *Rigid and flexible 2D bar-and-joint structures. If we attached the triangle over the flexible rectangle, we would have a framework that has both flexible and rigid components. The flex is usually determined by holding down or 'fixing' the bar between any two green nodes.*

6.2 Frameworks and the Realization Problem

A bar-and-joint structure can be defined mathematically as a *framework*. Given an undirected finite graph, $G = (V, E)$, where V is the set of all vertices in G and E is the set of all edges in G, $|V|$ is the number of vertices, and $|E|$ is number of edges. Depending on the set of edges, a graph can be a *simple graph* (a graph that does not have any loops or parallel edges) or a *multi-graph* (a graph that can consist of parallel edges). Next consider L, which is a set of fixed lengths for edges in G. Adding L to G, gives a framework. Thus, a framework is denoted by (G, L). L is often defined as a distance function on the edges of G such that $L : E \longrightarrow \mathbb{R}$, where $\mathbb{R}$ is the set of real numbers.

Any given framework can raise the following question: 'Is the framework a valid structure?' In other words, we are asking whether the framework can be embedded in Euclidean space and assigned valid coordinates. This question about embedding addresses what is more commonly known as the *realization problem* [14]. $\mathbb{R}^n$ is used to denote n-dimensional Euclidean space and can be described as a set of points: $\Big\{(a_1, \ldots, a_n) : a_j \,\epsilon\, \mathbb{R}\Big\}$. The realization of a framework (G, L) in $\mathbb{R}^n$ is an assignment of points of $\mathbb{R}^n$ to the vertices of G. The assignment is denoted by P, where $P = (p_1, p_2, \ldots, p_{|V|})$. The points assign coordinates to the vertices of V, which are constrained by the edge lengths defined in L.

If the framework is defined as a single collection of real numbers, all taken from the assignment P, then the framework can be seen as a single point in $\mathbb{R}^{n|V|}$. The manifold of $\mathbb{R}^{n|V|}$ comprising all possible realizations of a framework forms the *configuration space* of the framework.

6.3 Introduction to Rigidity

Rigidity theory is used to study the configuration or motion space that a framework may have while preserving the constraints defined by its distance function. Rigidity analysis may identify a framework as rigid, flexible or flexible with rigid components. Note that a framework, rigid or not, can always preserve its distance constraints if the framework *rotates or translates as a whole about an axis*, i.e., every point in the framework is moved by the same amount. These movements are called the *trivial motions* of the framework because they are present in both flexible and rigid structures. A framework is said to be flexible if and only if it has *flex* (see Figure 6.2) or non-trivial motion i.e., it can be deformed without breaking the distance constraints.

6.3.1 Rigidity Analysis of Bar-and-Joint Structure

The development of the rigidity theory for any model, including the bar-and-joint structure in n-dimension, can be broken into three general steps: *algebraic*, *infinitesimal*, and *combinatorial*. This section presents brief descriptions of each, using 2D bar-and-joint structure as an example.

Algebraic Rigidity

Algebraic rigidity is closely associated with the realization problem and tries to find an embedding for the framework in $\mathbb{R}^n$. In $\mathbb{R}^2$, the distance between two points, p_i and p_j, assigned to two vertices of a framework, can be found using the following distance or length equation:

$$(x_i - x_j)^2 + (y_i - y_j)^2 = L_{ij}^2, \text{ where } p_i = (x_i, y_i), p_j = (x_j, y_j), \text{ and } L_{ij} \text{ is the fixed distance.}$$

A solution set of this system of quadratic equations, is an assignment of x and y values, which satisfies all the equations in the system. This solution set will contain the trivial motions.

Infinitesimal Rigidity

Since, the algebraic equation system is non-linear, it quickly becomes computationally expensive and infeasible. In order to obtain a linear equation system, we differentiate the equations in the algebraic system. The previous section, stated that for an assignment $P = (p_1, p_2, \ldots, p_{|V|})$ to framework (G, L), consisting of $|V|$ vertices in n-space, gives rise to an algebraic equation system. Using rudimentary linear algebra, the algebraic equation can be written as:

$$(p_i - p_j) \cdot (p_i - p_j) = L_{ij}^2,$$

where $(p_i - p_j) \cdot (p_i - p_j)$ is the dot product of the vector $(p_i - p_j)$ with itself. The motion of the framework can be represented as a continuous curve through the manifold, which is the configuration space. A particular solution can be obtained by parameterizing the equation system. In this case, t is the parameter and can be interpreted as time. So, at any particular moment t, P can be written as $P(t) = (p_1(t), p_2(t), ..., p_{|V|}(t))$. Thus, a more general form of the equation above is:

$$(p_i(t) - p_j(t)) \cdot (p_i(t) - p_j(t)) = L_{ij}^2$$

We use the product rule to differentiate the above equation, assuming that the configuration path of the embedding is smooth. It is simple to show that the result of the product rule can be written as the dot product. L_{ij}^2 does not change with t, because distance needs to be preserved. Thus the result is written as:

$$(p_i(t) - p_j(t)) \cdot (p_i'(t) - p_j'(t)) = 0$$

This gives a system of linear equations for *infinitesimal rigidity* of the framework. Infinitesimal rigidity defines a framework to be rigid or flexible depending on the type of its infinitesimal motion and can be viewed as an approximation for algebraic rigidity. Infinitesimal rigidity implies algebraic rigidity, but not vice versa [23].

Conceptually, infinitesimal rigidity assigns an instantaneous or infinitesimal velocity vector to each point $p(t)$ in the assignment $P(t)$ of the framework and is denoted by the first derivative, $p'(t)$ or $v(t)$. More formally, we define $v(t)$ as the infinitesimal velocity vector of the framework at time t, where $\vec{v}$ is a n-vector in $\mathbb{R}^n$. So the general equation for the system becomes:

$$[p_i(t) - p_j(t)] \cdot [v_i(t) - v_j(t)] = 0 \text{ for all } i, j \; \epsilon \text{ E}$$

If the infinitesimal motions correspond only to the trivial ones (translations and rotations), the framework is *infinitesimally rigid*; otherwise, it is *infinitesimally flexible*.

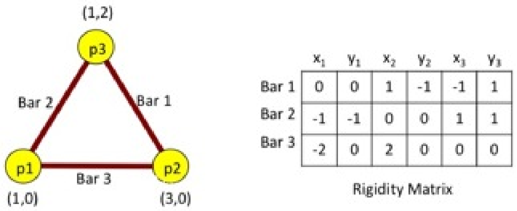
Representing the infinitesimal equation system Since we now have a system of linear equations, the system can be treated as a matrix system. If the framework is embedded in $\mathbb{R}^n$ and M is a matrix with $|E|$ rows and $n|V|$ columns (one column for each vertex coordinate), then the matrix system, $M\vec{v} = 0$ represents the linear equation system, where $\vec{v}$ is the collection of velocity vectors and the system holds true for all $\vec{v}$. The characteristics that must be true for M are:

i. For every edge $ij \; \epsilon$ E, there is a row in M.

ii. Each point $p_i(t)$, assigned to the framework, M has n column entries.

iii. For any edge ij, the n column entries associated with $p_i(t)$ contains the coordinate difference given by $p_i(t) - p_j(t)$, and the n column entries associated with $p_j(t)$ contains the coordinate difference given by $p_j(t) - p_i(t)$.

If all of the above holds true, then the matrix M is called the *rigidity matrix* for a bar-and-joint structure and $\vec{v}$ describes infinitesimal motion space of the structure. An example is shown in the Figure 6.3:



	x1	y1	x2	y2	x3	y3
Bar 1	0	0	1	-1	-1	1
Bar 2	-1	-1	0	0	1	1
Bar 3	-2	0	2	0	0	0

Rigidity Matrix

Figure 6.3: *A bar-and-joint structure, with assignment P and its corresponding rigidity matrix.*

Thus, the kernel or null space of the rigidity matrix of the framework defines the *infinitesimal motion space* of the framework. Note that the infinitesimal motion space contains all possible motion of the framework, including the trivial ones. Generally, the trivial motions relate to the elements of the Lie algebra $se(n, \mathbb{R})$, which has dimension $n(n + 1)/2$ (see Chapter 3). Thus, if the dimension of the kernel is exactly $n(n + 1)/2$, then the only possible transformations of the system are the trivial ones, resulting in a infinitesimally rigid framework. The framework is infinitesimally flexible if and only if the dimension of the kernel is greater than $n(n+1)/2$, implying it has more than the trivial motions. In Figure 6.3, the kernel of the rigidity matrix turns out to be 3, which is exactly equal to 2(2 + 1)/2 and hence, the structure is infinitesimally rigid.

Combinatorial Rigidity

Sometimes for a particular type of framework or model, it is possible that the pattern of the rigidity matrix may lead to the characterization of graphs that underly the structures. This combinatorial characterization gives information about infinitesimal rigidity of the structure. Laman's famous theorem is as follows [14]:

Laman's Theorem [20]. *A graph is generically*[1] *minimally rigid as a 2D bar-and-joint framework if and only if it has* $2n - 3$ *edges and any subset of* n' *vertices spans at most* $2n' - 3$ *edges.*

This kind of characterization can lead to combinatorial algorithms, e.g. Jacobs and Hendrickson′s 2D pebble game [10], for decomposing a given graph into rigid components as well as computing the degrees of freedom for the graph. The biggest advantage of combinatorial rigidity, over the other two types of rigidity constraint systems, is that it gives the most efficient algorithm for determining the rigidity of a structure.

However, note that the theorem applies only to 2D bar-and-joint structures. While algebraic and infinitesimal rigidity analysis exist for 3D bar-and-joint structures, there is no combinatorial characterization for these structures in 3D. This means that for analyzing a 3D bar-and-joint structure, the best choice is to use the linear equation system. If a protein is modeled using 3D bar-and-joint structure (with the atoms as nodes and the bonds as bars), then the rigidity matrix will have $3x$ columns, where x is the number of atoms in the protein and which can easily exceed a value of ten thousands.

Instead we use another class of framework, called the *body-and-bar/hinge structure*, to model protein. This class of structure is well understood in rigidity theory and has a combinatorial characterization, whose importance is described in section 7.3.

[1] The concept of generic rigidity is very complex and outside the scope of this thesis.

Chapter 7

Rigidity Theory: Body-and-Bar/Hinge

As stated in the last chapter, performing rigidity analysis on a protein requires the protein to be modeled as a body-and-bar/hinge structure. Example of such a structure is shown in Figure 5.1.

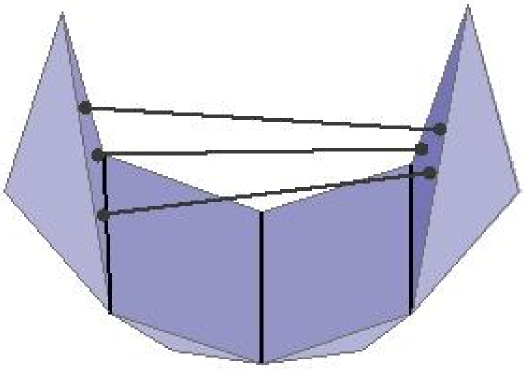


Figure 7.1: *A body-and-bar/hinge structure. Reproduced from [17]*

The structure in Figure 7.1 is composed of 4 rigid bodies. The three bars shown impose three different distance constraints between the two rigid bodies. The bars are attached to the bodies via universal joints. Each universal joint is located at a specific position in one of the bodies, defined by a point. A hinge, defined by a rotation axis, allows a rotation with a single degree of freedom.

Once again, the understanding of rigidity theory of 3D body-bar/hinge is obtained in three different steps. This foundation for understanding the rigid theory was presented in [38, 32]. First we lay the rigidity theory foundation for body-and-bar structures, then show how that is adapted to include hinges.

7.1 Algebraic Rigidity

The framework for a body-bar structure is described by $(G, \mathbf{p}, \mathbf{q}, L)$, where $G = (V, E)$ is a multi-graph with $|V|$ vertices and $|E|$ edges. Each vertex represents a body and each edge represents a bar. L is the distance function, $L : E \longrightarrow \mathbb{R}$ for each bar; **p** and **q** are each an m-tuple of points. The points in these tuples correspond to the universal joints of the bars, such that a bar e between two bodies would attach to one body at point $\mathbf{p}_e$ in **p** and to the second body at point $\mathbf{q}_e$ in **q**.

Just as before, algebraic rigidity is related to the realization problem and tries to find an embedding for the framework $(G, \mathbf{p}, \mathbf{q}, L)$ that satisfies L. However, for this model, the vertices no longer represent points in space and before proceeding, we need to define what it means to be a 3D rigid body and find its valid representation.

A body is an abstract structure or frame. We can define a single object or a collection of objects as a body. Mathematically, we use a transformation matrix to define a body. These transformation matrices in 3D are elements of the special Euclidean group $SE(3, \mathbb{R})$ (see Chapter 3). Recall that a point, acted upon by an element of $SE(3, \mathbb{R})$, undergoes a rotation, which is defined by the $SO(3, \mathbb{R})$ component of the element, and a translation, defined by the translation component of the element.

Now let $\mathbf{T} \epsilon (SE(3))^{|V|}$ be an assignment of frames to all $|V|$ bodies in V, with $\mathbf{T}_u$ being the transformation matrix for the frame assigned to body u [1] . Consider a bar $e = uv \ \epsilon$ E with attachment points $\mathbf{p}_e \ \epsilon \ \mathbb{R}^3$ and $\mathbf{q}_e \ \epsilon \ \mathbb{R}^3$ in bodies u and v, respectively. Then, the equation system for algebraic rigidity of body-and-bar structure has the following form:

$$\left\| \mathbf{T}_u(t) \begin{pmatrix} \mathbf{p}_e \\ 1 \end{pmatrix} - \mathbf{T}_v(t) \begin{pmatrix} \mathbf{q}_e \\ 1 \end{pmatrix} \right\|^2 = (L(e))^2$$

Note that the points **p** and **q** are represented using their homogenous coordinates, mainly to match the dimensions of matrices in **T**, which are all 4×4 matrices. According to [29], adding the extra

[1] $SE(3, \mathbb{R})$ is written as $SE(3)$ for brevity.

dimension does not change the position or geometric properties of the points. Observe that for this equation system, the unknowns are the elements of **T**, not the points.

Similarly as with bar-and-joint frameworks, a body-and-bar structure is flexible only if its motion space, which defined by the solution set of the equation system, results in a non-trivial configuration; otherwise the structure is rigid.

7.2 Infinitesimal Rigidity

The infinitesimal theory for body-and-bar frameworks is a study of the deformations of these frameworks over time, where a deformation at time t can also be thought of as the framework assuming a different configuration. If $\mathbf{T} \epsilon (SE(3))^{|V|}$ is the assignment of transformation matrices to the vertices of a framework, then $\mathbf{T}(t)$ represents the set of deformations of the structure over time. Since $\mathbf{T}_i(t)$ is an element of $SE(3,\mathbb{R})$, at $t = 0$, all $\mathbf{T}_i(0) = I_4$. Differentiating the algebraic equation we get:

$$\left\langle \mathbf{T}_u(t)\begin{pmatrix}\mathbf{p}_e\\1\end{pmatrix} - \mathbf{T}_v(t)\begin{pmatrix}\mathbf{q}_e\\1\end{pmatrix}, \mathbf{T}'_u(t)\begin{pmatrix}\mathbf{p}_e\\1\end{pmatrix} - \mathbf{T}'_v(t)\begin{pmatrix}\mathbf{q}_e\\1\end{pmatrix}\right\rangle = 0$$

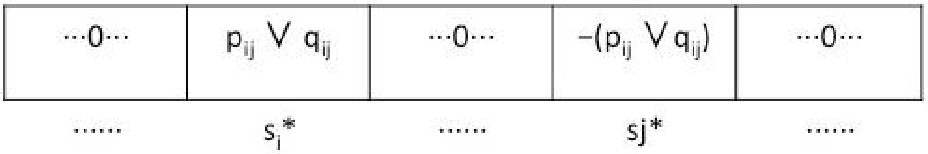
The first order derivative $\mathbf{T}'(t)$ represents the infinitesimal motions for the bodies at time t. Since, for each body u, $\mathbf{T}'_u(0) \epsilon se(3,\mathbb{R})$ (refer to Chapters 3 and 4), an infinitesimal motion for a body is described by an element of $se(3,\mathbb{R})$. Thus, $\mathbf{T}'(t) \epsilon (se(3,\mathbb{R}))^{|V|}$.

The infinitesimal motion of a framework is defined as an assignment $\vec{v} \epsilon (\mathbb{R}^6)^{|V|}$, such that $M\vec{v} = 0$, where M is the rigidity matrix of the framework (Figure 7.2). The set of values of $\vec{v}$ which satisfies $M\vec{v} = 0$, is the kernel of the M and the *infinitesimal motion space* of the entire framework. The kernel is denoted by σ. A trivial infinitesimal motion assigns the same screw to all bodies.

It can be shown [31, 21] that the bar constraints in the body-and-bar/hinge structure will be maintained infinitesimally if and only if the following equation holds true:

$$\langle s_i^*, p_{ij} \vee q_{ij} \rangle - \langle s_j^*, p_{ij} \vee q_{ij} \rangle = 0$$

In the above equation, p_{ij} and q_{ij} are attachment points of the bar $p_{ij}q_{ij}$ in bodies i and j respectively; $p_{ij} \vee q_{ij}$ are the $Plücker$ coordinates; and the operation $< \quad >$, refers to the vector dot product. This equation is then used to set up the rigidity matrix for the structure as shown in Figure 7.2. If $|E|$ is the number of bars and $|V|$ the number of bodies in one such framework, then the rigidity matrix for the framework is an $|E| \times 6|V|$ matrix, in which each bar is defined by one row, and each body is associated with 6 columns or 6 entries from the $Plücker$ coordinates describing an instantaneous screw. The kernel of the rigidity matrix is the space of infinitesimal motions and is described by the set of screws, $\{s_1, \ldots, s_i, s_j, \ldots, s_{|V|}\}$. Note that the maximum rank of the rigidity matrix is restricted to $6|V| - 6$, because the kernel or null space of the matrix will always contain the 6 trivial motions, which are the 3 rotations and 3 translations. Thus, if the rigidity matrix achieves its maximum rank of $6|V| - 6$, then the structure is *infinitesimally rigid* and it can only undergo the trivial motions.



···0···	$p_{ij} \vee q_{ij}$	···0···	$-(p_{ij} \vee q_{ij})$	···0···
······	s_i^*	······	sj*	······

Figure 7.2: *A row from a rigidity matrix of a body-and-bar/hinge framework.*

Incorporating a hinge to the infinitesimal equation system Hinge joints can easily be modeled into the infinitesimal equation system. A hinge can be represented by adding five rows in the rigidity matrix of the body-and-bar structure. The process of finding the five rows for a particular hinge is shown in [8]. The only requirement for the hinge constraint to be preserved is that the hinge axis must be the same as the screw axis.

7.3 Combinatorial Rigidity

Combinatorial rigidity for body-and-bar structure has been characterized by Tay.

Tay's Theorem [32]. *A 3-dimensional body-and-bar structure is generically rigid if and only if the associated multigraph with vertices instead of bodies and edges instead of bars is composed of 6 edge-disjoint spanning trees.*

Tays theorem has lead to a 3D pebble game algorithm, used by FIRST [9]. Initially, FIRST models the proteins by identifying each atom as a body. This would actually result in a rigidity matrix with $6x$ rows, where x is the total number of atoms in the protein. In comparison, using a 3D bar-and-joint structure, to model the protein, would require $3x$ columns. However, using the combinatorial characterization of body-and-bar structures, FIRST can group the atoms together into rigid clusters and reduce the number of bodies by identifying the clusters as bodies. (For more work on combinatorial rigidity, see [8].)

Chapter 8

Methodology for Analyzing Infinitesimal Motion Space

In order to obtain information about the infinitesimal of a framework, we need to analyze its infinitesimal motion space (see Chapter 7). This chapter gives the general procedure that we developed to analyze the infinitesimal motion space of both CAD and protein models.

8.1 Infinitesimal Motion Space of a Pinned Framework

As stated in Chapters 6 and 7, rigidity theory can be applied only to certain types of structures and we are interested in 3D body-and-bar/hinge structures. When working with a particular body-and-bar/hinge structure, we first need to convert it into a framework or rigidity model, which is denoted by $F = (G, \mathbf{p}, \mathbf{q}, L)$ (see Chapter 7). Using infinitesimal rigidity theory, we obtain the rigidity matrix, M for the framework. Recall that M is a $|E| \times 6|V|$ matrix, in which each row represents a constraint within F and that M has 6 columns for every body in F. Thus, solving the constraint system for F is equivalent to solving the matrix system $M\vec{v} = 0$. The set of all vectors, $\vec{v}$, that satisfy $M\vec{v} = 0$, is the kernel, σ of M, .i.e., $\sigma = \{\vec{v} \mid M\vec{v} = 0\}$. Since σ is a vector space, it can be described by the span of a set of k vectors in $\mathbb{R}^{6|V|}$. These k vectors are referred to as the basis vectors[1] of σ. The basis is denoted

[1] The set of basis vectors is called the basis of the vector space. Basis is a set of linearly independent vectors. For a detailed explanation of basis and linear combination, please refer to [12]

by, $\{\vec{r_1}, \vec{r_2}, \ldots \vec{r_k}\}$, where $\vec{r_i}$ is in $(\mathbb{R}^6)^{|V|}$. This vector space is called the *infinitesimal motion space* of F.

Since σ is a vector space and the kernel of M, each of its element can be described as a linear combination[2] of the k basis vectors. Thus, $\vec{v} \, \epsilon \, \sigma$ can be defined as follows:

$$\vec{v} = c_1\vec{r_1} + c_2\vec{r_2} \ldots + c_k\vec{r_k}$$

where $c_i \, \epsilon \, \mathbb{R}$. The dimension of σ will be denoted by k, where k is the number of *degrees of freedom* for F. Degrees of freedom is the measure of independent motion. As noted in Chapter 6, every framework in 3-space will have at least 6 degrees of freedom, which correspond to the 6 trivial motions (3 rotations and 3 translations). Thus, $k \geq 6$.

However, we are interested only in infinitesimal motions that are not the trivial motions of F. We can effectively remove the trivial infinitesimal motions from the infinitesimal motion space of F, by introducing six more rows to M, such that they remove 6 degrees of freedom from F. This is done by first placing the following identity matrix at the bottom of the columns for body i in M, and then filling in the remaining row entries with zeros (see Figure 8.1). This literally *pins* down or fixes body i and gives us a second rigidity matrix for F, M_{pinned}, which is a $(|E| + 6) \times 6|V|$ matrix [38].

$$I_6 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Solving the constraint system of F, using M_{pinned} instead of M gives us a new kernel, σ_{pinned} and hence, a new infinitesimal motion space that excludes trivial motion. Thus the dimension, k_{pinned}, of σ_{pinned} is, $k_{pinned} = k - 6$ and will take on a value of zero if F is completely rigid.

[2] For a complete definition of linear combination, please refer to any standard linear algebra textbook e.g. [12]

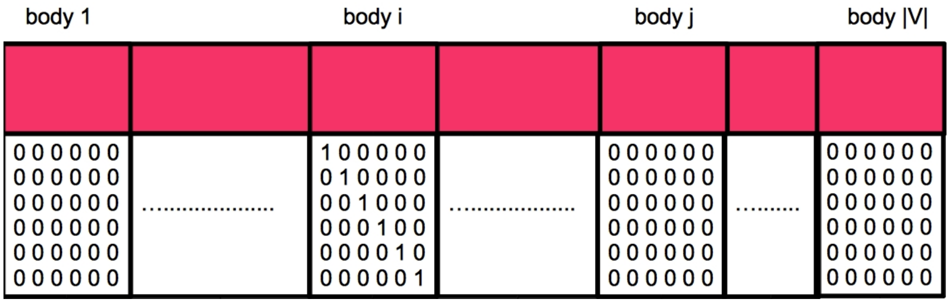


Figure 8.1: *A figure showing how to obtain M_{pinned} from M. The colored part represents the original rows of M.*

8.2 Relative Infinitesimal Motion Space of body j with respect to body i

Now that we have a framework, F_{pinned}, which is pinned at body i, we set out to analyze the infinitesimal motion space of a particular body, j. Doing so gives us the infinitesimal motion of body j, *relative* to the pinned body. But before proceeding, to avoid confusion between σ and σ_{pinned}, we define the basis of σ_{pinned} to be a set, β, such that $\beta = \{\beta_1, \beta_2, \ldots, \beta_{k-6}\}$, where β_i is a row vector of the kernel of M_{pinned} in $(\mathbb{R}^6)^{|V|}$. Thus, $\vec{v}_{pinned}$ is defined as:

$$\vec{v}_{pinned} = c_1\vec{\beta_1} + c_2\vec{\beta_2} + \ldots + c_{k-6}\vec{\beta_{k-6}}.$$

Each β_l is defined as follows:

$$\beta_l = [s^*_{l,1}, s^*_{l,2}, \ldots, s^*_{l,|V|}]$$

where, if $s^*_{l,1} = [e, f, g, a, b, c]$ and $s_{l,1} = [a, b, c, e, f, g]$, then $s_{l,1}$ is the instantaneous screw assigned to body 1, obtained from the l^{th} row of σ_{pinned}.

Therefore, to analyze the infinitesimal motion space of body j, relative to body i, we extract from σ_{pinned}, the instantaneous screws defined by all $s^*_{l,j}$ for all values of l, where the value of l ranges from 1 to (k-6) and where j refers to a particular body (Figure 8.2).

After the extraction, we are left with a set, S of instantaneous screws, such that $S = \{s_{1,j}, s_{2,j}, s_{3,j}, ...s_{k-6,j}\}$, where each $s_{l,j}$ is represented as a 6-vector. Next, we look for dependence of these 6-vectors in S. Dependence within a vector set arises, when one vector in the set can

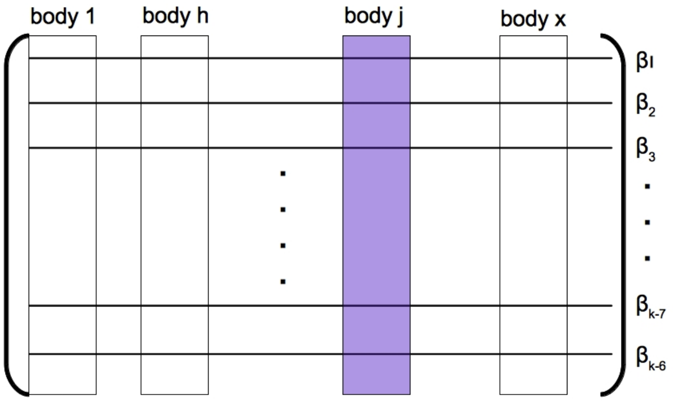


Figure 8.2: *Extraction of instantaneous screws, describing the infinitesimal motion space of body j, from* σ_{pinned} *We are extracting the screws represented in the highlighted region.*

be expressed as a linear combination of two other vectors in the set [12]. We use Mathematica code to filter out the dependent vectors and obtain a basis, S_B, that describes the infinitesimal motion space of body j, relative to body i. We say that S_B is the *relative infinitesimal motion space*.

Then, to examine how close the infinitesimal motion of body j is to the to a pure rotation or translation, we calculate a minimized perpendicular distance between the vector space in $\mathbb{R}^6$ formed by S_B and the Grassmannain. Recall from Chapter 5 that the Grassmannian is a structure in $\mathbb{R}^6$ that is described by all 2-extensors (which identifies with instantaneous screws), which define pure rotations or pure translations. A case analysis for interpreting the value of the minimum distance, combined with the dimension of the relative infinitesimal motion space (or the vector space) is given in the next section. Note that the dimension of the relative infinitesimal motion space is given by $dim(S_B)$.

8.3 Case by Case Analysis of the Relative Motion of a Body

Here, we first illustrate the possible outcomes from our analysis.

Case I: *The vector space is a proper subspace of* $\mathbb{R}^6$ *and the minimum distance is greater than but 'closer' to zero.*

The interpretation of result is as follows:

- The relative infinitesimal motion space does not overlap with the Grassmannian. Dimension of subspace is less than six.
- The relevant structure does not undergo pure rotation.
- The motion of the structure is close to that of a pure rotation.
- If the structure is a protein, then the protein is likely to be a hinge protein.

This case is non-trivial because important information, regarding the type of infinitesimal motion, is obtained from the relative infinitesimal motion space.

Case II: *The vector space is a proper subspace of* $\mathbb{R}^6$ *and the minimum distance is greater than but 'further' from zero.*

The interpretation of result is as follows:

- The relative infinitesimal motion space does not overlap with the $Grassmannian$. Dimension of subspace is less than six.
- The structure does not undergo pure rotation.
- The motion of the structure is not close to that of a pure rotation.
- If the structure is a protein, then the protein is likely to be a non-hinge protein.

This case is non-trivial because important information, regarding the type of infinitesimal motion, is obtained from the relative infinitesimal motion space.

Case III: *The vector space is a proper subspace of* $\mathbb{R}^6$ *and the minimum distance is zero.*

The interpretation of result is as follows:

- The relative infinitesimal motion space does overlap with the Grassmannian. Dimension of subspace is less than six.
- The structure does undergo pure rotation.
- The motion of the structure is a pure rotation.
- The result is unlikely for a protein.

This case is non-trivial because important information, regarding the type of infinitesimal motion, is obtained from the infinitesimal motion space.

Case IV: *The vector space is all of* $\mathbb{R}^6$ *and the minimum distance is zero.*

The interpretation of result is as follows:

- The relative infinitesimal motion space trivially contains the Grassmannian.
- No relevant information on the structure.
- The result is unlikely for CAD structures.
- If the structure is a protein, then it most likely that the infinitesimal motion space captures motion that is not usually allowed in a protein. The structure instantaneously behaves as if it is disconnected.

This case is trivial because no new information is obtained from the infinitesimal motion space.

Chapter 9

Analyzing Infinitesimal Motion Space of CAD Models

This chapter focuses on our research with CAD models and illustrates how we applied the general procedure (described in the Chapter 8) to those models. The first section of this chapter introduces the important software that were used.

9.1 SolidWorks

SolidWorks is a 3D mechanical CAD (computer-aided design) program that runs on Microsoft Windows. This allows engineers and designers to define a mechanical model by specifying constraints or parameters to the building blocks of the model. Building a model in SolidWorks usually starts with a 2D sketch, which is then used as the blueprint for the 3D model.

9.2 Methodology for CAD

The outline of the procedure we followed to analyze the mechanical models is shown in 9.1.

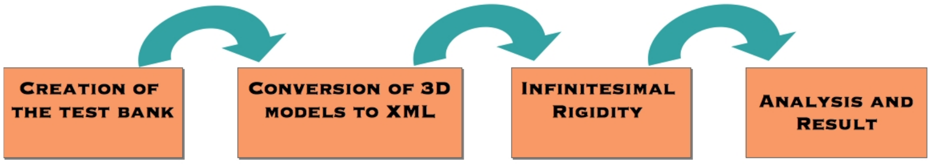


Figure 9.1: *Outline of the methodology for identifying joints in CAD models.*

Creation of the Test Bank

In SolidWorks, more than one set of constraints can be used to define a revolute (rotational with one degree of freedom) or a prismatic (translational with one degree of freedom) joint between two rigid bodies [31] (see Figures 9.2 through 9.5). A set of models with defined revolute joints and a second set of models with defined prismatic joints were made. All of these models in these sets, consist of two rigid bodies only and are thus collectively called the *simple test models*. A third set of test models were created, where each model had more than two rigid bodies and joints. Thus, the models in the third set are thought of as 'more complicated' due to the increased number of structures and joints. Also, for models in the third set, the connections between the structures are implied, i.e., even though there are no constraint between two bodies, their behavior suggests otherwise. These three sets of test models are collectively called the **test bank**.

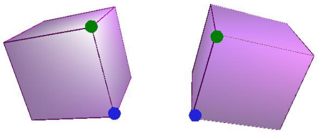


Figure 9.2: *An example of defining a revolute joint in SolidWorks. Making the two green vertices and the two blue vertices coincide with each other will create a revolute joint between the cubes. The axis of rotation will be the line between the green and blue vertices.*

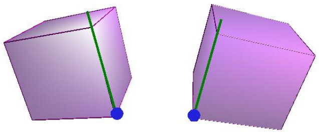


Figure 9.3: *Another example of defining revolute joint in SolidWorks. Making the two green edges (or lines) and the two blue vertices coincide with each other will create a revolute joint between the cubes. The axis of rotation will be the line defined by the green lines and blue vertices.*

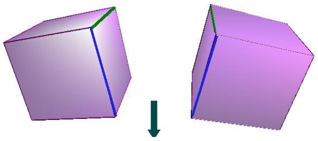


Figure 9.4: *An example of defining a prismatic joint in SolidWorks. Making the two green edges parallel and the two blue edges coincide with each other will create a prismatic joint between the cubes. The of motion is shown by the arrow.*

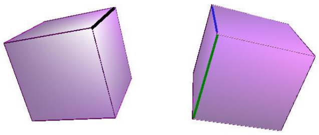


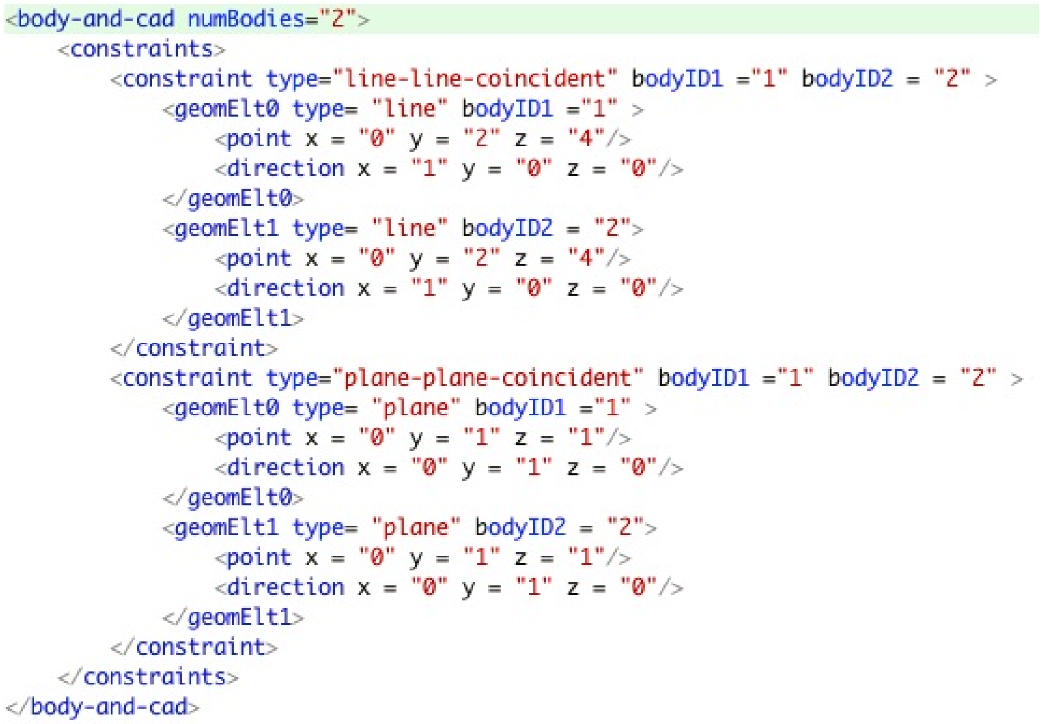
Figure 9.5: *A second example of defining a prismatic joint in SolidWorks. Making the black and green edges perpendicular and the black and blue edges coincide will create a prismatic joint between the cubes. The direction of motion will be along the line of coincidence between the black and blue edges.*

Conversion of 3D model to XML

XML file is a text document that uses mark-up language and specifies the format in which data should be entered, making it readable by both human and computers. To allow analysis of the models, a XML file was created for each model in the test bank. The specifications for data entry, based on the format used, are: the number of rigid bodies in the model, the geometric elements (lines/edges, points/vertices, planes) involved in making joints, the coordinates of the geometric elements and the type of constraints between the elements. Within the XML, a line or a plane is defined by a point on the line/plane and the direction of the line or the direction of the normal respectively. Figure 9.6 shows a snapshot of one of the XML files.

Infinitesimal Rigidity

The XML file for the model is then used to construct its corresponding rigidity matrix. The next step involves following the steps outlined in section 8.1, which describes how to manipulate the rigidity matrix to exclude trivial infinitesimal motions from the infinitesimal motion space of the model by pinning down a body.



```
<body-and-cad numBodies="2">
    <constraints>
        <constraint type="line-line-coincident" bodyID1 ="1" bodyID2 = "2" >
            <geomElt0 type= "line" bodyID1 ="1" >
                <point x = "0" y = "2" z = "4"/>
                <direction x = "1" y = "0" z = "0"/>
            </geomElt0>
            <geomElt1 type= "line" bodyID2 = "2">
                <point x = "0" y = "2" z = "4"/>
                <direction x = "1" y = "0" z = "0"/>
            </geomElt1>
        </constraint>
        <constraint type="plane-plane-coincident" bodyID1 ="1" bodyID2 = "2" >
            <geomElt0 type= "plane" bodyID1 ="1" >
                <point x = "0" y = "1" z = "1"/>
                <direction x = "0" y = "1" z = "0"/>
            </geomElt0>
            <geomElt1 type= "plane" bodyID2 = "2">
                <point x = "0" y = "1" z = "1"/>
                <direction x = "0" y = "1" z = "0"/>
            </geomElt1>
        </constraint>
    </constraints>
</body-and-cad>
```

Figure 9.6: *A sample XML file.*

Analysis and Result

As described in section 8.1, the kernel of the rigidity matrix forms the infinitesimal motion space of the model. Once the required body in the CAD model is pinned down, we investigate the relative infinitesimal motion space of the other bodies (see section 8.2).

Since the simple test models consist of only two bodies and undergoes either a rotation or a translation with 1 degree of freedom, the dimension of the infinitesimal motion space of the whole model is 1 and hence, the dimension of the infinitesimal motion space for studying relative infinitesimal motion is also 1.

The pegboard, shown in Figure 9.7, is an example of a more complicated model. It consists of three rotating pegs, all of which appear to be attached to the board. Also, for each peg, its axis of rotation passes through its center and is held perpendicular to the board. However, Figure 9.8 shows that their

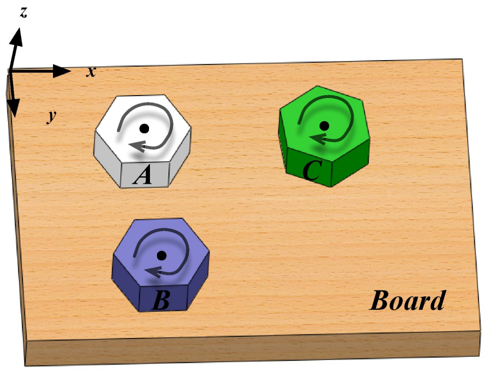
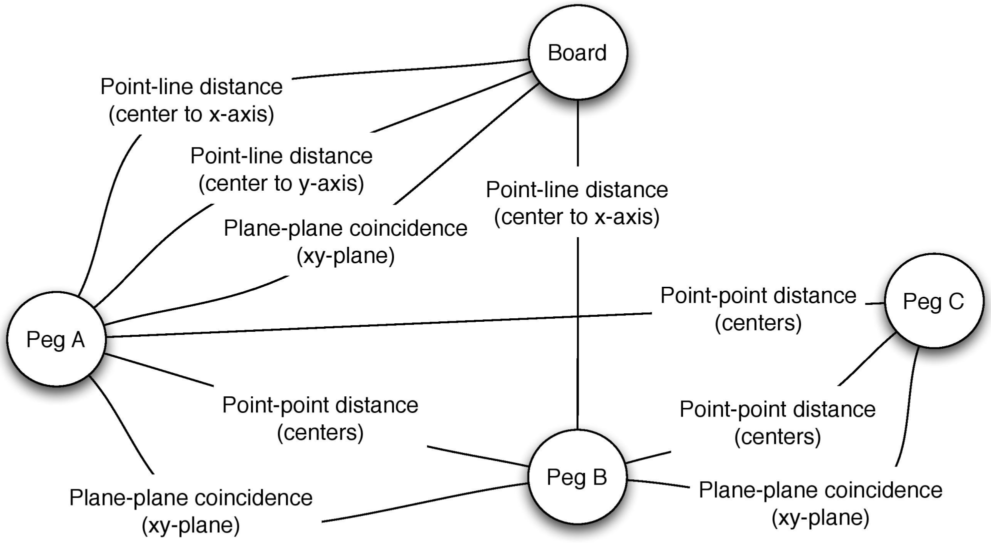


Figure 9.7: *The pegboard model.Reproduced from [22].*

is no constraints between peg C and the board. In this case, the dimension of the infinitesimal motion space of the whole model is 3. However, the dimension of the relative infinitesimal motion space, S_B, turns out to be 1 for each peg (see section 8.2). This means that any particular peg is allowed to undergo a motion with one degree of freedom, relative to the board. We found similar results for other models with implied connections.



Board
Point-line distance (center to x-axis)
Point-line distance (center to y-axis)
Plane-plane coincidence (xy-plane)
Point-line distance (center to x-axis)
Point-point distance (centers)
Peg C
Peg A
Point-point distance (centers)
Point-point distance (centers)
Peg B
Plane-plane coincidence (xy-plane)
Plane-plane coincidence (xy-plane)

Figure 9.8: *The constraint graph for the pegboard model, showing the different constraints used between different bodies. Reproduced from [22].*

For each of these models, the minimum distance calculated was 0 and based on the dimensions of the relative infinitesimal motion space, the outcome of their analysis coincided with Case III as described in section 8.3.

We were able to distinguish between rotational and translational degrees of freedom by looking at the instantaneous screws that were extracted from σ_{pinned} in each case. Recall that a screw, $s = (-\omega^x, -\omega^y, -\omega^z, \tau^x, \tau^y, \tau^z)$. The ω entries of s are all zeros if the screw describes a translation only; otherwise, the screw is either a decomposable 2-tensor and describes a rotation or it is an indecomposable 2-tensor. Since, for CAD models, the revolute joint is the same as a hinge joint, our approach successfully identified hinge motion in CAD models.

Chapter 10

Analyzing Infinitesimal Motion Space of a Model for a Protein

Inspired by our work with the CAD models, we adapted our procedure to identify hinges in the CAD models so that it can be applied to proteins in general. We did our case study with calmodulin, which is a known hinge protein.

10.1 Software Tools

This section describes the important software we used to analyze calmodulin.

FIRST

FIRST (Floppy Inclusions and Rigid Substructure Topography) is a program that identifies rigidity (flexibility) in hinge frameworks. FIRST models a protein structure as a body-bar graph. Rigid clusters of atoms, which often tend to correspond to structural domains (Chapter 2) within the protein, are modeled as bodies (to see the definition of body, see Chapter 7); hydrogen bonds and hydrophobic interactions are modeled as bars; and covalent bonds as hinges. The program removes different degrees of freedom[1] to model the different bonds as follows:

[1] Degrees of freedom is the measure of independent motion. A rigid body can at most have 6 degrees of freedom.

i. non-rotatable covalent bonds, like the peptide bonds, remove 6 degrees of freedom,

ii. rotatable covalent bonds remove 5 degrees of freedom,

iii. hydrogen bonds remove 4 degrees of freedom, and

iv. hydrophobic tethers remove 2 degrees of freedom.

Hydrogen bonds, covalent bonds and hydrophobic interactions are attractions (or repulsions) between atoms or group of atoms that are found in proteins. More information about the bonds can be found in [3] and more information about FIRST at [9, 33].

KINARI

KINematic And RIgidity analysis of proteins is a second-generation protein rigidity analysis software, which is similar to FIRST but provides an interactive web server for performing the analysis and visually exploring rigidity properties of proteins. It also provides tools for pre-processing the input data, such as selecting relevant chains from PDB files, adding hydrogen atoms and identifying stabilizing interactions. [30]

PyRosetta

PyRosetta is an interactive Python-based interface for the Rosetta molecular modeling software. Rostta is a program suite for predicting and designing protein structures, protein folding mechanisms, and protein-protein interactions. PyRosetta provides sampling methods and energy functions, which can then be used to analyze the quality of a given protein conformation among various other applications.

In order to identify hinge proteins solely based on rigidity analysis, we started down the path shown in Figure 10.1:

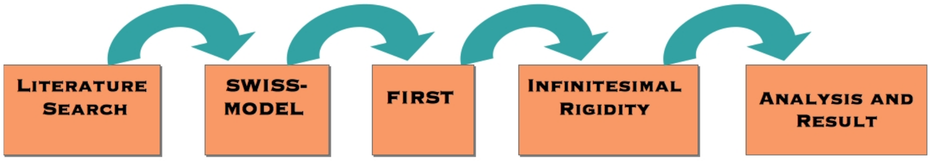


Figure 10.1: *Outline of the methodology for identifying hinge joints in protein models.*

Literature Search

First, we need to find candidate proteins for our analysis. The process of identifying candidate hinge proteins involves searching for publications that propose new ways to determine hinge proteins with acceptable success rate. The publications should also list the proteins that have been identified as hinge proteins. Moreover, due to the limitations of the software we had to use, the protein should consist of a single polypeptide chain (refer to Chapter 2). For our research, we used the hinge proteins identified by StoneHinge [18]. Once a candidate protein is identified, we can find its PDB file at the Protein Data Bank (*http://www.pdb.org*).

SWISS-MODEL

SWISS-MODEL is an automated protein structure homology-modeling server, which is used to fix downloaded PDB files. Often, PDB files have missing residues due to experimental limitations faced when studying the protein structure. Residues are the parts of amino acids that become incorporated into the protein during protein synthesis. Missing residues are important for rigidity analysis and becomes problematic. Running the downloaded PDB file through SWISS-MODEL leaves us with a complete polypeptide chain with no missing residues [11].

FIRST

The fixed PDB file is run through online FIRST, which produces a number of output files describing the rigidity of the protein structure. In particular, it identifies the rigid components (or bodies) in the protein structure. To make the situation analogous to the CAD models, we experiment with the *energy cutoff* value, which is a numeric parameter that has to be passed on to online FIRST, along with the fixed PDB file. The energy cutoff value determines the number of hydrogen bonds to include in the protein structure. A greater number of hydrogen bonds results in fewer and larger rigid bodies or clusters. To have two rigid clusters joined by the known hinge region or hinge residues, we manually perform the same operation as StoneHingeP (refer to [18]), determining the energy cutoff value by visually examining the rigid clusters, using Jmol [2].

Once the two reasonable rigid clusters have been identified, we create a residue text file for each rigid cluster in the protein structure. The text files are made by visually reading the ID number of residues contained within the corresponding rigid cluster using the FIRST molecule viewer, which uses Jmol. These residue files, along with the output files produced by online FIRST, are fed into a Perl script. The script reads the PDB file for the protein and finds the ID numbers of all the atoms that make up the residues included in the residue file. It then uses the files, generated from the output of online FIRST (describing the rigid clusters in the protein), and finds all bodies, which consist of these atoms. The output file of the script gives a list of pairs of (i, j) values, where i represents the ID of the body that contains j number of atoms from the ones present in the provided residues. Thus, the body ID of the two rigid clusters can be obtained, e.g. the body ID is i, which is associated with the largest j. One of these bodies is fixed, so that we explore the relative infinitesimal motion space for the second body.

Next we need to run the original fixed PDB file through the command-line FIRST to get required output files that are not produced by online FIRST. These files are given as parameter to another Perl script. This script produces the required XML file describing the body-bar-hinge graph for the protein structure. We then use a MATLAB code to calculate the kernel of the protein model.

Analysis and Result

We propose to analyze the relative infinitesimal motion space of one of two bodies (corresponding to the rigid clusters). Using concepts from rigidity theory and different areas of mathematics, we have obtained the kernel of the rigidity matrix of the framework representing the protein. Once we have the protein modeled as a body-and-bar/hinge structure, we can apply the same procedure, described in Chapter 8, on the protein model.

Since we are looking at two rigid clusters that are connected within a protein that has no missing residues, we expect the relative infinitesimal motion space of the protein model to be a proper subspace of $\mathbb{R}^6$. In other words, we expect the relative infinitesimal motion space to have a dimension less than 6 *and* the minimum distance to be greater than but close to 0, if the protein has a hinge between the two clusters. If the protein is a non-hinge protein, then the minimum distance should be 'more' positive. Thus, we expected out result to coincide with Case II.

Our test set of hinge proteins included: LAO binding protein, Bence-Jones protein, DNA polymerase beta protein, inorganic pyrophosphatase and calmodulin. For all of these proteins, we performed our analysis of relative infinitesimal motion on their corresponding models. For all the protein models, we find that the minimum distance, between the Grassmannian and the body under investigation, is zero and that the dimensions of the relative infinitesimal motion space is 6. This coincides with Case IV of our analysis procedure and gives us a trivial result.

10.2 Investigating the Affect of Steric Hindrance

Since the results did not support our hypothesis, we undertook further investigation to figure out the challenges presented by a protein. We hypothesize that the information about the 'feasible' motions of the protein is masked by the presence of 'infeasible' motions in the infinitesimal motion space. This possibility arises because, while rigidity theory validly captures the geometric equality constraints, it does not account for collisions within the structure that is being analyzed. This means that rigidity theory, solely based on constraint analysis, will identify motion even if it leads to collisions.

Since proteins are naturally occurring chemical molecules, like all chemical molecules, they prefer to exist in a state, where they have the minimum potential energy. However, collisions between atoms in a protein have an unfavorable effect on the potential energy of the protein. Two atoms are said to 'collide' with one another when the distance between the two nuclei goes below a certain value. This threshold value is the sum of the *Van der Waals radii* of the two atoms. The van der Waals radius of an atom is the radius of an imaginary sphere, which is often used to model an atom.

In Figure 10.2, the two circles represent two atoms and r is the distance between the two atoms. The energy of the whole system decreases as r increases and reaches a minimum at some particular value of r. This value will be different for different atoms involved and is taken as the internuclear separation equilibrium. After the equilibrium point, the energy rises again but stays less than zero. Two atoms are colliding if their energy is in the repulsive region (as shown in Figure 10.2).

We developed the procedure shown in Figure 10.3 to test our hypothesis that the infinitesimal motion

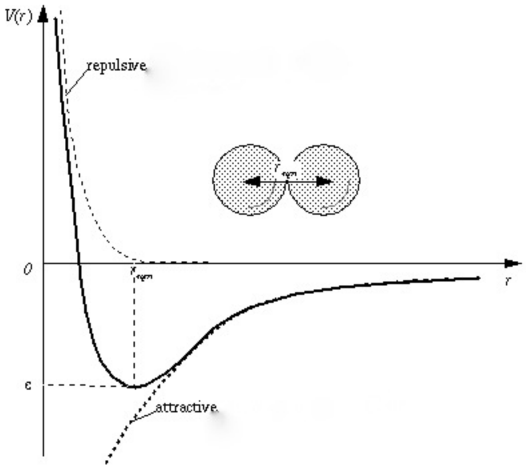


Figure 10.2: *A graph demonstrating the changes in total energy of the system (in this case, the two atoms) as the distance between the atoms increases. Reproduced from the online english translation of [35].*

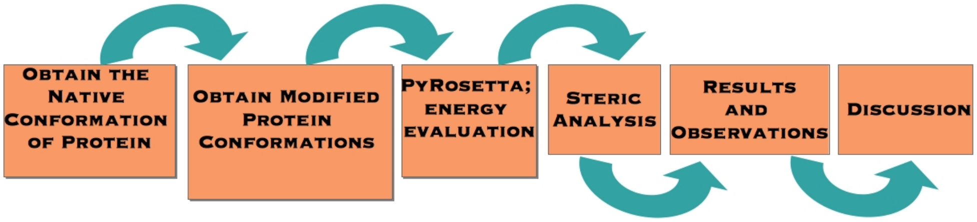

Obtain the Native Conformation of Protein
Obtain Modified Protein Conformations
PyRosetta; Energy Evaluation
Steric Analysis
Results and Observations
Discussion

Figure 10.3: *Outline of the methodology for investigating steric hindrance in protein.*

space of the protein model includes motion that is not biologically feasible due to *steric hindrance* (hindrance due to physical position).

Obtain the Native Conformation of Protein

The initial PDB file is obtained from the Protein Data Bank. Using Pymol, an open-source molecular visualization system, two conformations (usually the naturally occurring closed and open conformations) of the protein are aligned to correct residues and mutations. KINARI is then used to add hydrogen atoms back to the protein PDB file because they initially need to be removed for aligning. Among the two conformations, we choose the one that is not bound to ligands[2] (because ligand bound state has altered stability) and call the corresponding cleaned PDB file the native conformation file. We need this

[2] Ligands are usually small chemical molecules that bind to protein, affecting protein stability and hence, motion.

file for the subsequent steps.

Obtain Modified Protein Conformations

The XML file produced by FIRST is a representation of the protein as a body-and-bar/hinge structure. Basically, FIRST takes the atoms in the protein and assigns a collection of atoms to a particular body. Each body has an unique numerical ID. Note that bodies can overlap with each other e.g. two bodies are allowed to have a subset of atoms that is common between them.

In Chapter 8, we have seen that a solution to the infinitesimal rigidity constraint system of a model, obtained using the rigidity matrix, M_{pinned}, can be written as a linear combination of vectors, $\vec{\beta_i} \epsilon \beta$, where the linear combination is represented as follows:

$$\begin{aligned}\sigma_{pinned} &= c_1\vec{\beta_1} + c_2\vec{\beta_2} + \ldots + c_{k-6}\vec{\beta_{k-6}} \\ &= c_1\sum_{l=1}^{k-6} s^*_{l,1} + c_2\sum_{l=1}^{k-6} s^*_{l,2} + \ldots + c_{|V|}\sum_{l=1}^{k-6} s^*_{l,|V|}\end{aligned}$$

Recall that this allows σ_{pinned} to be represented as a vector in $(\mathbb{R}^6)^{|V|}$. Thus for each body j, such that $1 \leq j \leq |V|$, we can extract its corresponding $c_1\sum_{l=1}^{k-6} s^*_{l,j}$ and thus, extract the instantaneous screw assigned to body j. Also, recall from section 4.2 that the equation:
$instantaneous\ velocity, \vec{v} = \Omega\vec{p} + \tau,$
translates the instantaneous screw to the instantaneous velocity of a point within a rigid body.
So, for each $c_j\sum_{l=1}^{k-6} s^*_{l,j}$, we will get a $\vec{u_j}$, which is the *instantaneous velocity vector* for all points contained in body j.

The coordinates of the atoms of the protein are essentially the points within the rigid bodies, used to model the protein. Thus, by adding $\vec{u_j}$, for all values of j, to the coordinates of the atoms (which make up the rigid cluster that corresponds to body j), we can displace the atoms from their positions in the native conformation of the protein. This results in a modified conformation of a protein, which is achieved by an infinitesimal motion described by the infinitesimal motion space of the protein model.

Note that c_j is a scalar quantity and does not actually change the direction of $\vec{u_j}$, but does change its magnitude. Hence, if we randomly choose the coefficients values, then we are actually randomly sampling the infinitesimal motion space of the protein model. Each time we randomize, we obtain a different modified conformation of the protein.

PyRosetta

The scoring function in PyRosetta calculates different energy components, including van der Waals repulsion between atoms, van der Waals attraction, solvation energy, and the energy in various hydrogen (long and short range, between main chain and side chain) and disulfide bonds. Using these components, the function then outputs a weighted score for the overall energy of the protein. This score is a measure of the stability of the protein molecule and is directly proportional to the energy of the protein. For our purpose, we use the default weight set in PyRosetta. The output of PyRosetta is saved in a text file, which is then read into Excel for statistical analysis.

Steric Analysis

For steric analysis, we used a protein called calmodulin. Using the native conformation state of calmodulin, we obtained two sets of samples[1]: one set has 100 samples, while the other has 1000. Using these two data sets we look for evidence that might support or reject our hypothesis about steric hindrance.

Results and Observations

The statistics presented in this section are from the set with 1000 samples. For clarity, the graphs presented here are the ones that were obtained from set with 100 samples. The change in sample size caused insignificant variation in the statistics obtained from the data.

First, we obtained a chart, shown in Figure 10.4 that compared the overall scored energy of all the protein conformations. A majority of the modified conformations seemed to have overall energies that are much higher than the overall energy of the native conformation. However, a few modified

[1] Samples refer to the different conformations of the protein.

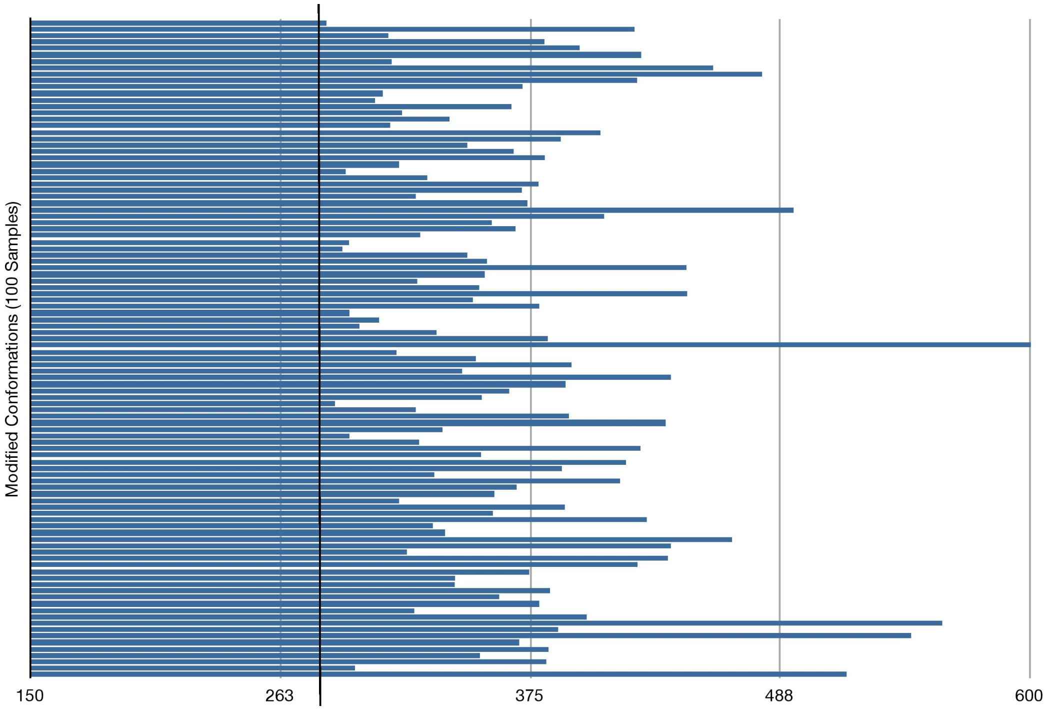


Figure 10.4: *Overall energy comparison for the modified conformations. The black line shows the overall energy of native conformation. The horizontal axis is the overall energy axis.*

conformation, seemed to have overall energies that are very close to the overall energy value of the native conformation. Next, to determine what caused the variation in the overall energy between the modified protein conformations, we obtained charts to examine the energy components that are involved in the calculation of the overall energy. Figures 10.5 through 10.8 show the variation observed in the different energy components between the modified conformations. Figure 10.5 shows that there is great variation in the van der Waals repulsion; the range of variation is between 400 to above 1300 score units. Figure 10.6 shows that there is some variation in the van der Waals attraction between the modified conformations; the range of variation is between -614 to -630 score units. Note that it is standard practice to use negative value to represent attraction and as always, lower energy (i.e. more negative energy value) implies increased stability. When looking only at van der Waals attraction, we find that some of the modified conformations actually have more negative energy values than the energy value of native conformation. However, this increase in stability is countered by a greater increase in the energy for van der Waals repulsion (shown in Figure 10.7) and it can be seen that very few conformations have repulsion energy that is similar to or as low as that of the native conformation. Figure 10.8 shows little variations in other energy components, which include solvation energy and energies from different

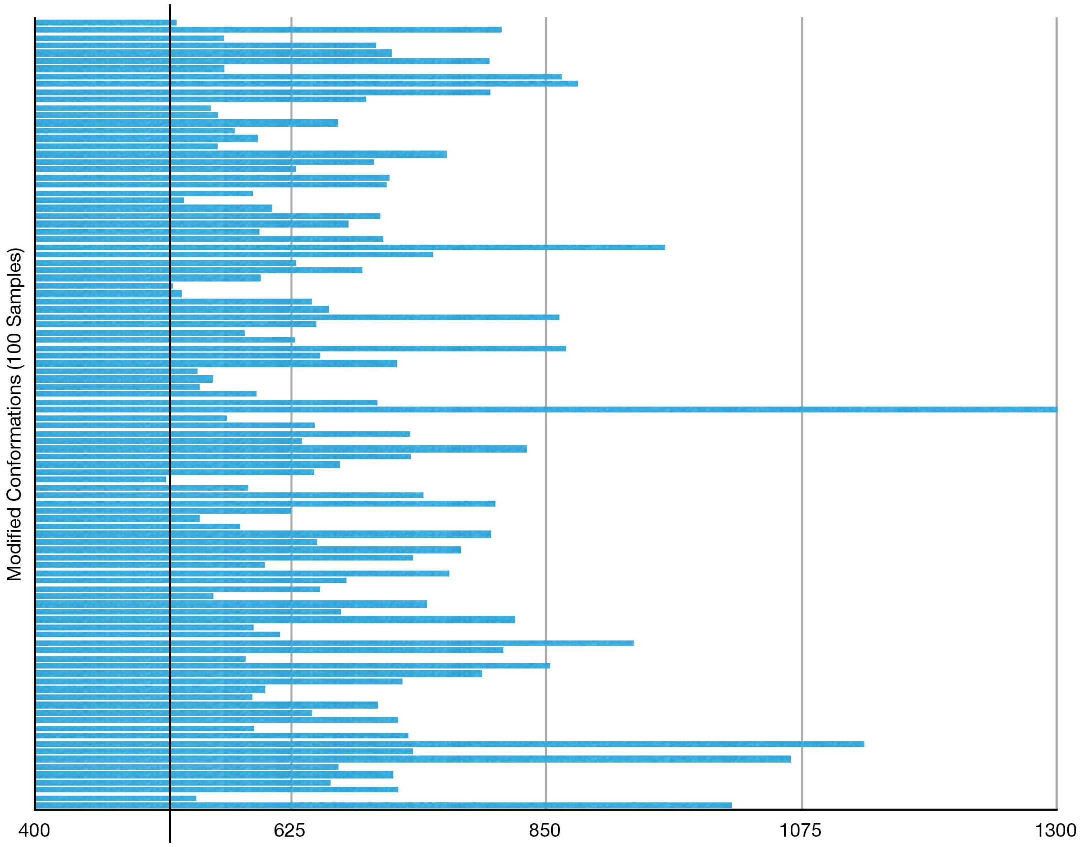


Figure 10.5: *Van der Waals repulsion. The black line shows the repulsion energy for the native conformation.*

hydrogen bonds. Note, for each chart presented in Figures 10.5 through 10.8, one bar on the chart represents a modified protein conformation. Also, each chart is represented with a different scale for the horizontal axis and different color shows the different number of energy components being compared in that particular chart.

Now, we present the statistics obtained from the two data sets. The average of the overall energy is 372.4 score units with a standard deviation of 61. The average of all other energy component, except van der Waals repulsion, is very similar to that of the native conformation with little variation. The average van der Waals repulsion is 691.4 score units with a standard deviation of 135.7, whereas the repulsion energy in the native conformation is 524.3 score units.

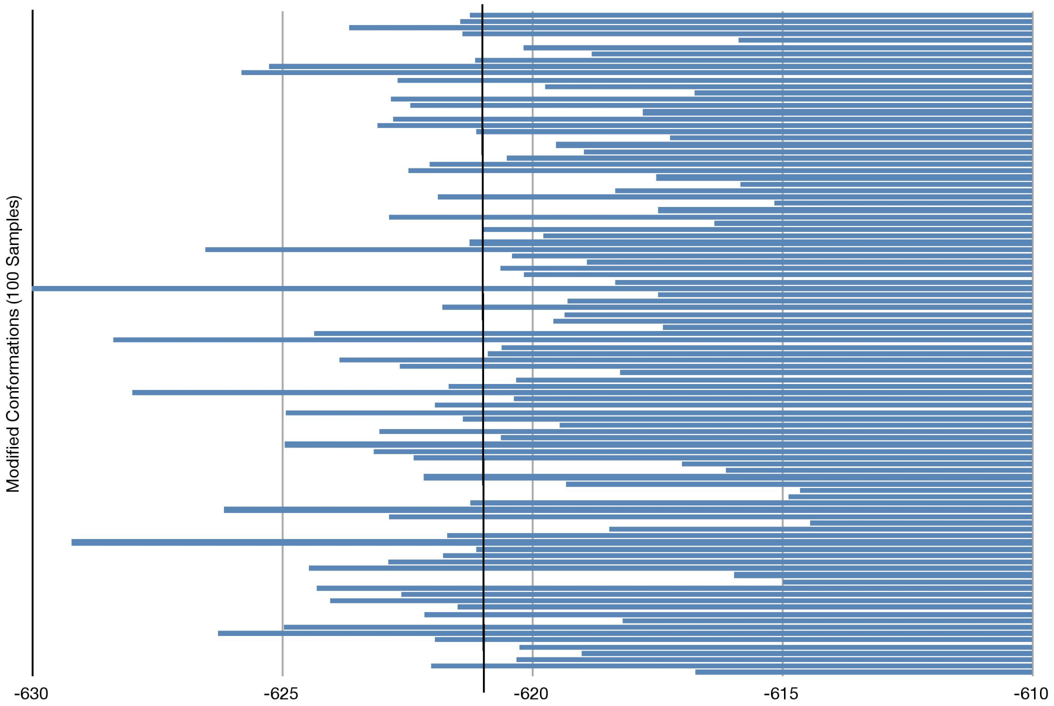


Figure 10.6: *Van der Waals attraction. The black line shows the attraction energy for the native conformation. Note that the horizontal axis is much more expanded than in Figure 10.5.*

Discussion

Once we had all the data, we subtracted the overall energy of the native conformation of the protein from the overall energy of each of the modified conformation. Based on this calculation, we define “feasible motion” as an infinitesimal motion that leads to a modified protein conformation, which has an overall energy, E such that E is not higher than the overall energy of the native conformation by more than 20 score units; any other infinitesimal motion is defined as “infeasible.” In our data set of 1000 samples, 43 modified conformations seemed to arise from feasible motion, i.e., 4% of the relative infinitesimal motion space of the protein model seems to consist of feasible motion. Note that since we are looking at the relative infinitesimal motion space (which factors out the trivial motions), the 43 conformations mentioned here cannot possibly be the same as the native conformation of the protein. This seems to support our conjecture that the relative infinitesimal motion space of the protein model contains infinitesimal motion that is physically possible and encourages further work.

The large variation in the van der Waals repulsion, among our samples, shows that it is an important

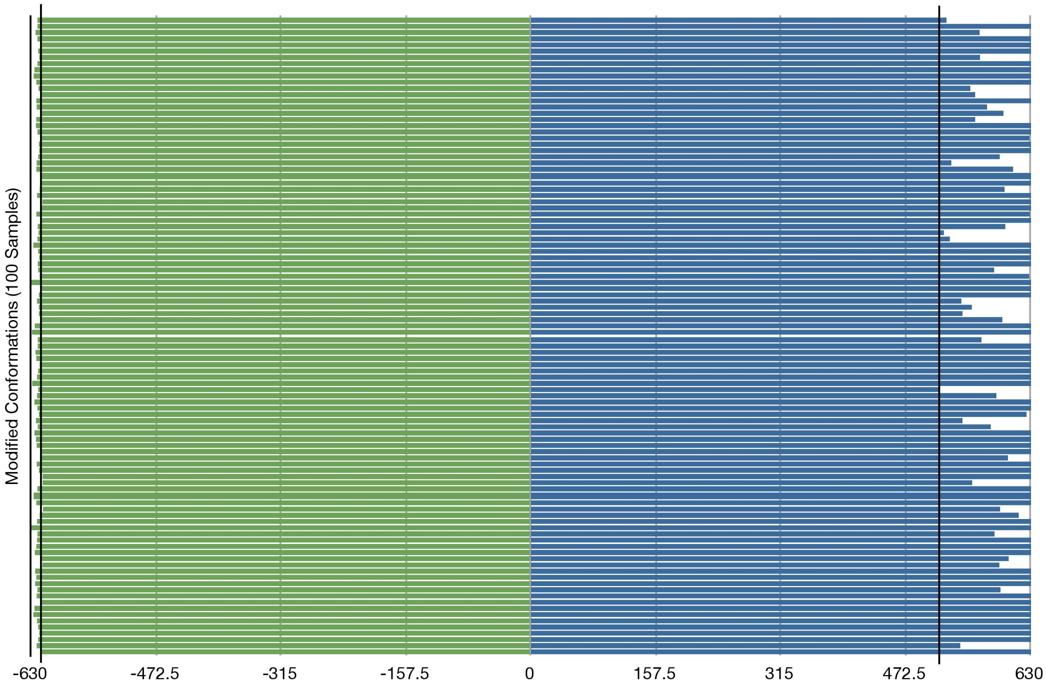


Figure 10.7: *Showing variation in the van der Waals attraction and repulsion among the samples on a more comparable scale.*

determinant of the overall energy. As the van der Waals attraction, the solvation energy and hydrogen-bond energies do not vary as much as the van der Waals repulsion, they play a minor role in determining the overall energy of a protein conformation. Thus, our findings show that repulsion is the only energy component that varies significantly. This supports our hypothesis that steric hindrance within protein structure plays an important role in protein motion in the physical world.

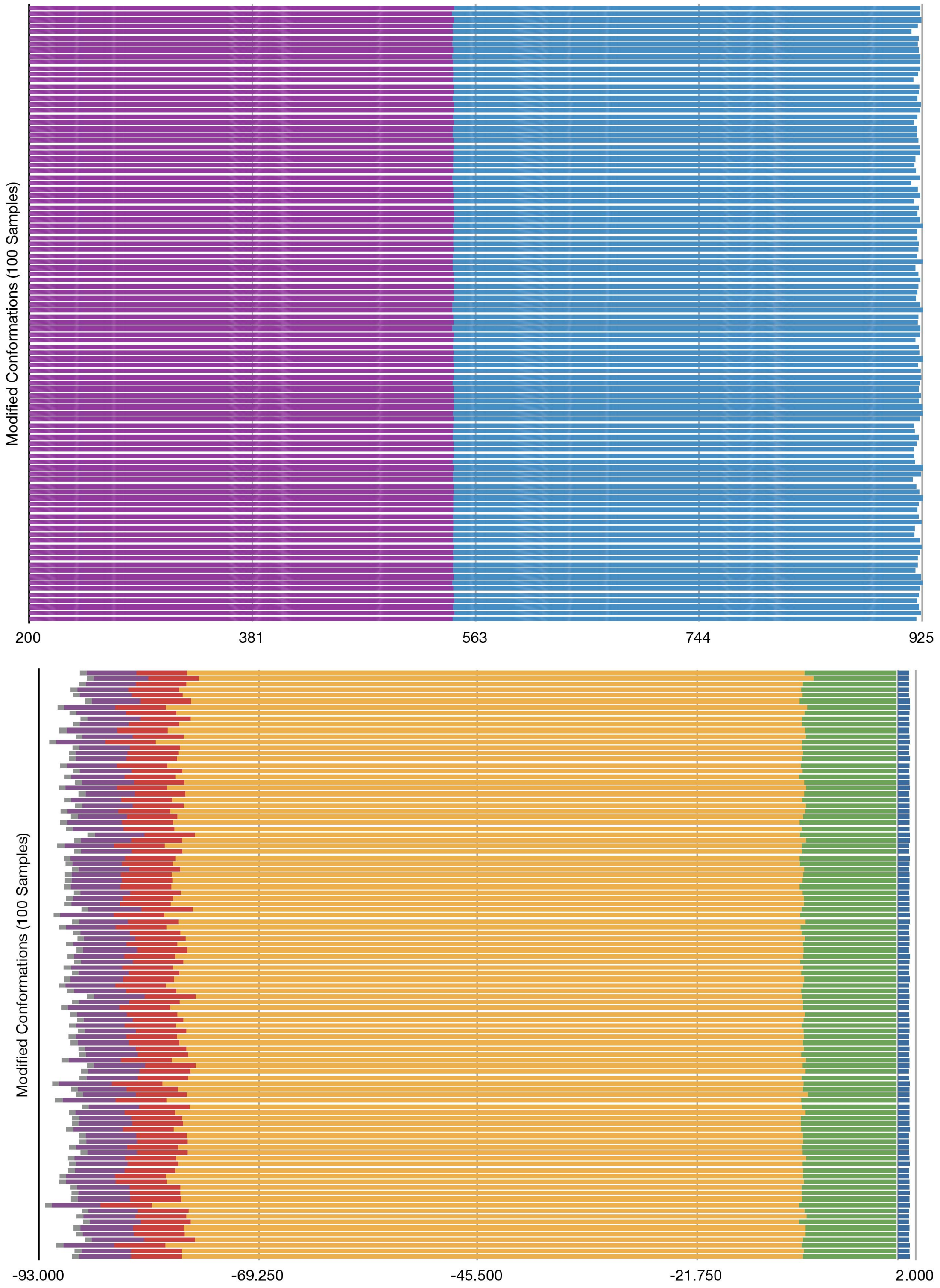


Figure 10.8: *This figure shows two charts, each of which shows variation in the different energy components. Note that the bar lengths in both charts form bands of almost uniform width.*

Chapter 11

Conclusion and Future Work

The question that we consider in this thesis is: can we identify hinge proteins using rigidity theory only. We conclude that it is challenging to address this problem, without taking collisions into account. To account for collisions, we have to incorporate inequality constraints within the constraint system obtained from infinitesimal rigidity.

The problem investigated in this research has important applications in understanding protein function. We applied concepts that are often used in computer science and mathematics to examine protein structure and its potential for motion. We modeled proteins as specific structures whose movements are restricted by specific geometric constraints. This allowed us to conduct rigidity analysis on several protein structures: LAO binding protein, Bence-Jones protein, DNA polymerase beta protein, inorganic pyrophosphatase and calmodulin. In order to explore the infinitesimal motion space of the protein models, we used concepts from matrix Lie groups, instantaneous screws and Grassmann-Cayley algebra and had to rely on software like SWISS-MODEL, FIRST, KINARI, PyRosetta, SolidWorks and Mathematica. We then identified steric hindrance as one of the most important factors that plays a role in determining protein motion apart from equality constraints in protein structures. Our analysis shows that infinitesimal motion space has the potential to be used to identify hinge motion in proteins but requires further work of incorporating steric hindrance into our analysis to account for collisions between atoms in proteins.

Future Work on this Project

The immediate next step should be to try the steric analysis on multiple proteins. Doing so will provide additional evidence to support our claim that steric hindrance plays an important role in determining protein motion. Then we must develop a mathematical approach to solve this problem of considering a system of inequalities because collisions are represented as distance inequality constraints. Solutions to this new equation system might be used to find the feasible motion in the infinitesimal motion space of the protein model. However, this will introduce the challenge of solving inequalities within an acceptable timescale. Beside collisions between atoms, ligands also have considerable effect on protein stability, it is essential to integrate its effect into our analysis to make the analysis more robust. Identifying other factors, like steric hindrance, might also improve robustness. Finally, we hope to develop a user interface that allows easy access to the analysis.

Bibliography

[1] The cabp data library for calmodulin. website.

[2] Jmol: An open soruce java 3d molecular viewer.

[3] Bruce Alberts. *Essential Cell Biology*. Garland Science, 2009.

[4] Byung Ha Oh Jayvardhan Pandit Chul-Hee Kang Kishiko Nikaido Sabiha Gokcen Giovanna Ferro-Luzzi Ames and Sung-Hou Kim. Three-dimensional structures of the periplasmic lysine/ariginine/ornithine-binding protein with and without a ligand. *The Journal of Biological Chemistry*, 268(15), 1993.

[5] Protein Structural Analysis and Design Lab. Protein structural analysis laboratory.

[6] Robert Stawell Sir Ball. *A Treatise On The Theory of Screws*. At the University Press, 1900.

[7] Doug Davis. Periodic motion.

[8] Milka Doktorova. *Computational Analysis of Statics and Dynamics of Macromolecules*. PhD thesis, Mount Holyoke College, May 2010.

[9] Michigan State University Don Jacobs. Flexweb: Analysis of flexibility in biomolecules and networks. web server.

[10] Bruce Hendrickson Donald Jacobs. An algorithm for dimensional perolation: The pebble game. *Journal of Computational Physics*, 1997.

[11] M. Kunzli L. Bordoli T. Schwede F. Kiefer K. Arnold. The swiss-model repository and associated resources. *Nucleic Acids Research*, 37, 2009.

[12] Fraleigh and Raymond A. Beauregard. *Linear Algebra*. Addison-Wesley, 1995.

[13] A. S. Zektzer G. E. Martin. *Two-Dimensional NMR Methods for Establishing Molecular Connectivity*. Wiley-VCH, 1988.

[14] Herman Servatius Jack E. Graver, Brigitte Servatius. *Combinatorial rigidity*, volume 2. American Mathematical Society.

[15] Philip E. Bourne Jenny Gu, editor. *Structural Bioinformatics*. Wiley-Blackwell, second edition, 2009.

[16] S.A Wells J.E Jimenez-Roldan and R A Romer. Comparative analysis of rigidity across protein families. *Phys. Biol*, 6:1–11, 2009.

[17] Audrey St. John. Rigidity.

[18] Mark B. Gerstein Kevin S. Keating, Samuel C. Flores and Leslie A. Kuhn. Stonehinge: Hinge prediction by network analysis of individual protein structures. *proteinscience.org*, pages 1–12, 2008.

[19] Phillips GN Jr. Kundu S, Sorensen DC. Automatic domain decomposition of proteins by a gaussian network model. *Proteins*, Dec 1 2004.

[20] G. Laman. On graphs and rigidity of plane skeletal structures. *Journal of Engineering Mathematics*, 4(331-340), 1970.

[21] Audrey Lee. *Geometric Constraint Systems With Application In CAD Aad Biology*. PhD thesis, University of Massachusetts Amherst, May 2008.

[22] Audrey Lee-St.John. Kinematic joint recognition in cad constraint systems. preprint, 2012.

[23] Ben Roth Leonard Asimow. The rigidity of graphs ii. *Journal of Mathematical Analysis and Applications*, 68:171–190, March 1979.

[24] et al. Lodish H Berk A Zipursky Sl. *Molecular Cell Biology*. New York: W. H. Freeman, 4th edition, 200.

[25] Anisotropy of fluctuation dynamics of proteins with an elastic network model. A r atilgan, s r durell, r l jernigan, m c demirel, o keskin, i bahar. *Biophysical Journal*, 80(1):505–515, 2001.

[26] Harriet Pollatsek. *Lie Groups: A Problem-Oriented Introduction Via Matrix Groups Lie Groups: A Problem-Oriented Introduction Via Matrix Groups Lie Groups: A Problem-Oriented Introduction Via Matrix Groups*. MAA TEXTBOOKS, 2009.

[27] Audrey Lee-St.John Rittika Shamsuddin. The joint recognition problem: from cad constraints to kinematic joints. poster, October 2010.

[28] Todd Rowland. Manifold. From MathWorld- A Wolfram Web Resource, created by Eric W. Weisstein.

[29] J. M. Selig. *Geometric Fundamentals of Robotics*. Springer, 1996.

[30] Naomi Fox Filip Jagodzinski Yang Li Ileana Streinu. Kinari-web: A server for protein rigidity analysis. *Nucleic Acids Research*, 39, 2011.

[31] Neil White Kirk Haller Audrey Lee-St.John Ileana Streinu and Meera Sitharam. Body-and-cad geometric constraint systems*. *Computational Geometry: Theory and Applications*, pages 1–33, 2010.

[32] Tiong-Seng Tay. Rigidity of multi-graphs: Linking rigid bodies in n-space. *Combinatorial Theory, Series B*, 1984.

[33] Donald Jacobs A.J. Rader Leslie A. Kuhn M.F. Thorpe. Protein flexibility predictions using graph theory. *PROTEINS: Structure, Function, and Genetics*, 44:150–165, 2001.

[34] * Turkan Haliloglu, Ivet Bahar and Burak Erman. Gaussian dynamics of folded proteins. *Physical Review Letters*, 79(16), October 1997.

[35] J. D. van der Waals and Ph. Kohnstamm. *Lehrbuch der Thermodynamic*, volume 1. Mass and van Suchtelen, Leipzig, online edition edition, 1908.

[36] Neil White. Grassmann-cayley algebra and robotics, 1994.

[37] Neil White. Geometric applications of the grassmann-cayley algebra. *Handbook of Discrete and Computational Geometry*, 1997.

[38] Neil White and Walter Whiteley. The algebraic geometry of motions of bar-and-body frameworks*. *SIAM Journal of Algebraic Discrete Methods*, pages 1–32, 1987.

[39] A.J. Rader Chakra Chennubhotla Lee-Wei Yang and Ivet Bahar. The gaussian network model: Theory and applications, October 2005.