Design of Flexure-based Motion Stages for Mechatronic Systems via Freedom, Actuation and Constraint Topologies (FACT)

by

Jonathan Brigham Hopkins

M.S. Mechanical Engineering
Massachusetts Institute of Technology, 2007

B.S. Mechanical Engineering
Massachusetts Institute of Technology, 2005

SUBMITTED TO THE DEPARTMENT OF MECHANICAL ENGINEERING
IN PARTIAL FULFILMENT OF THE REQUIRMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY IN MECHANICAL ENGINEERING
AT THE
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

SEPTEMBER 2010

Signature of Author: ……………………………………………………………………………
Department of Mechanical Engineering
August 24, 2010

Certified by: ………………………………………………………………………………………
Martin L. Culpepper
Associate Professor of Mechanical Engineering
Thesis Supervisor

Accepted by: ………………………………………………………………………………………
David E. Hardt
Professor of Mechanical Engineering
Chairman, Department Committee on Graduate Students

Design of Flexure-based Motion Stages for Mechatronic Systems via Freedom, Actuation and Constraint Topologies (FACT)

by

Jonathan Brigham Hopkins

Submitted to the Department of Mechanical Engineering on
August 24, 2010 in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy in
Mechanical Engineering

ABSTRACT

The aim of this thesis is to generate the knowledge required to *(i)* synthesize serial flexure systems and *(ii)* optimally place actuators using a comprehensive library of geometric shapes called freedom, actuation, and constraint spaces. These geometric shapes guide designers through the creative process of concept generation without compromising engineering rigor. Each shape rapidly conveys the mathematics of screw theory, projective geometry, and constraint-based design by visually depicting regions where constraints and actuators may be placed for synthesizing optimal flexure concepts. In this way, designers may consider every flexure concept that satisfies the desired functional requirements before selecting the final design. FACT was created to improve the design processes for small-scale flexure systems and precision machines. For instance, there is a need to create multi-axis nanopositioners for emerging three-dimensional nano-scale research/manufacturing. Through this work the following contributions were made: (1) the fifty freedom and constraint space types were found that may be used to synthesize both parallel and serial flexure concepts, (2) intermediate freedom spaces were created that help designers stack conjugated flexure elements to avoid or utilize underconstraint, (3) a twist-wrench stiffness matrix was created to model the elastomechanic behavior of flexure systems, (4) the twenty-six actuation spaces were found that help guide designers in placing actuators that minimize motion errors, and (5) a theory was created that determines the force and displacement actuator outputs for accessing a desired DOF once actuators have been placed. A serially conjugated lead screw flexure was designed using the FACT design process and a parallel flexure system was built to validate the theory of actuation described in this thesis.

Thesis Supervisor: Martin L. Culpepper
Title: Associate Professor of Mechanical Engineering

ACKNOWLEDGMENTS

I would like to thank all those who helped make this research possible. I first acknowledge God whose hand is in all discoveries and whose love makes all things possible. I am also forever grateful to my devoted parents, Barbara and James Hopkins, who have sacrificed greatly for my education. A special thanks to my advisor, Professor Martin Culpepper, who tirelessly motivated, guided and directed my efforts. This work would not have been possible without his patience and faith in me. I am grateful to the other members of my doctoral committee, Professors Sarma and Seering, who helped shape this research with their insights and suggestions. To Jan van Eijk, Jingjun Yu, and my dear friends in the Precision Compliant Systems Laboratory who offered their support, patience and correction, I am also in debt. And finally, I am grateful to NSF, Soliant Energy, and Novartis for providing funding during my time as a research assistant. Thank you all very much.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1:
Introduction

This chapter provides a discussion of the purpose, importance, and impact of this research. The background principles that are necessary to understand the contributions of this work are briefly reviewed and an overview of the content of this thesis is provided.

1.1 Research Objectives

The purpose of this thesis is to generate the principles required to *(i)* synthesize multi-degree of freedom (DOF) flexure systems and *(ii)* optimally place their actuators. These principles are embodied in the form of geometric shapes like those shown in Figure 1.1A-B that help designers visually identify all the possible regions where constraints and actuators may be placed in order to achieve desired motions. These shapes — called freedom, actuation, and constraint spaces — enable designers to rapidly consider all the possible flexure concepts before selecting the concept that best satisfies the system's functional requirements. The flexure design process that utilizes these shapes is called Freedom, Actuation, and Constraint Topologies (FACT). Examples of multi-DOF flexure systems designed using FACT are shown in Figure 1.1C-D.

Flexure systems have been used as precision machine elements for over a century [1-8] due to their exceptional resolution characteristics, their low–cost characteristics, and the ease with which they may be fabricated. Flexure systems continue to be important to conventional precision applications. For example they are commonly used within optical manipulation stages, precision motion stages and as general purpose flexure bearings. More recently, flexure systems have become attractive for use in motion stages for nanomanufacturing equipment [9-10], instruments that are used in nano–scale research/manufacturing [11-15], and micro– and nano–manipulators [16-18]. These instruments and devices typically require the capability to move in four to six axes with nanometer–level resolution.

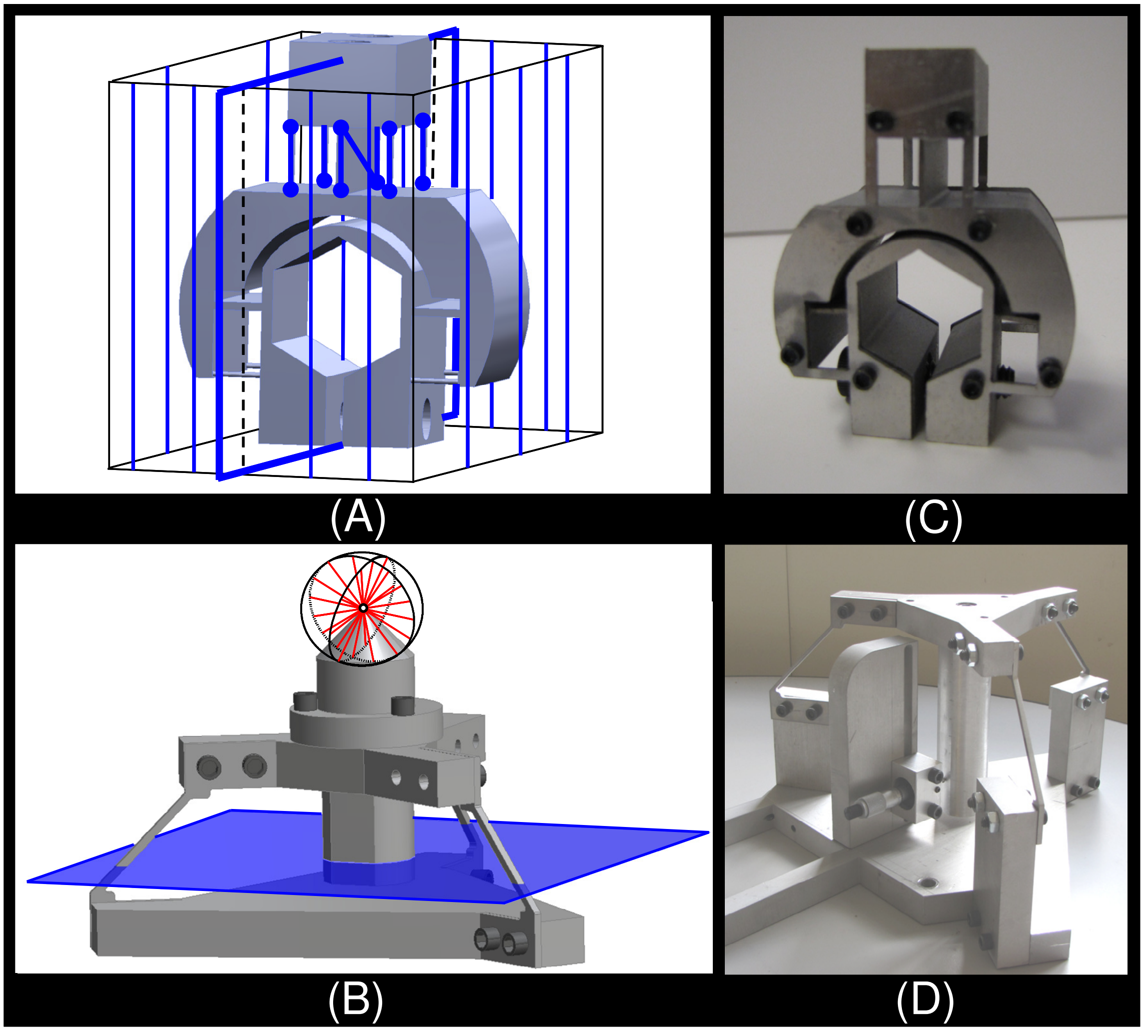

Figure 1.1: *Geometric shapes (A) and (B) may be used to synthesize and actuate flexure systems (C) and (D).*

1.1.1 Serial Flexure Synthesis

All flexure systems fall under two categories — parallel or serial. Parallel flexure systems consist of a single rigid stage that is connected directly to ground by non-conjugated flexible elements. Serial flexure systems consist of nested parallel flexure systems that are stacked together. Figure 1.2 shows a parallel and serial flexure system. The principles needed to synthesize parallel flexure systems are provided in Hopkins [19-21]. A major objective of this thesis is to generate the principles needed to synthesize serial flexure systems.

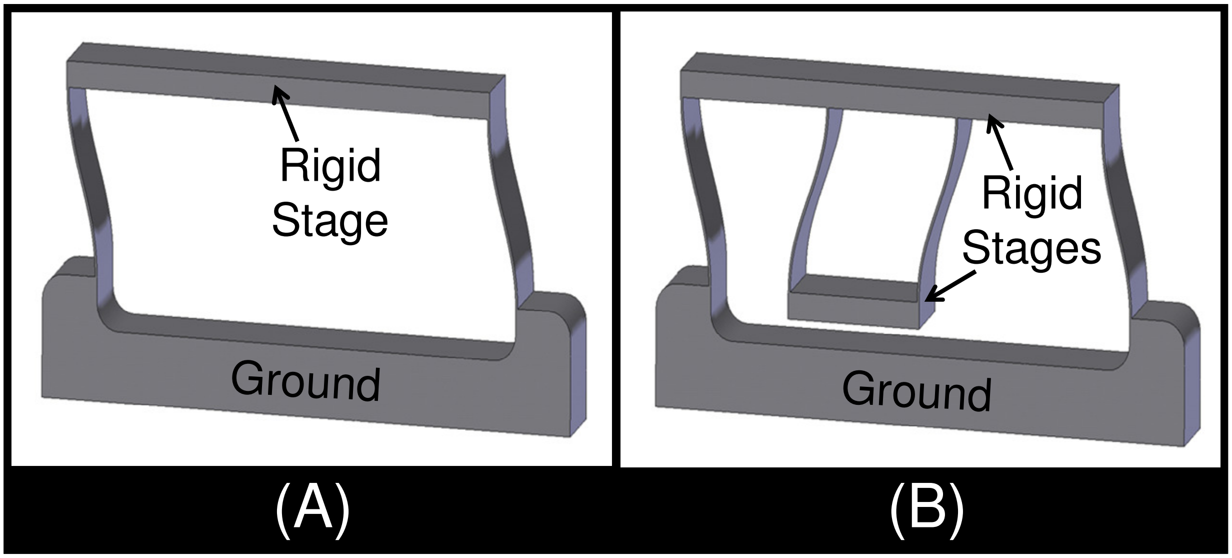

Figure 1.2: *Example of a parallel flexure system (A) and a serial flexure system (B).*

The synthesis of multi-DOF parallel flexure systems is difficult because there are typically many flexural components. It is difficult for designers to keep track of *(i)* the relative, three dimensional orientations of the flexural constraints, *(ii)* the orientation of the permitted motions, and *(iii)* the three dimensional relationships between each constraint and the permissible motions. Even if an expert designer is capable of the preceding, the task of synthesizing a multi-DOF serial flexure system becomes more difficult as extra stages are stacked on top of each other. The kinematics, elastomechanics, and dynamics of each stage are influenced by the kinematics, elastomechanics, and dynamics of the proceeding stages in the serial chain. The task of synthesizing most serial flexure systems is, therefore, very difficult for flexure system designers that rely on intuition and motion visualization capabilities alone.

The existing body of published flexible mechanism design knowledge provides little information or guidance as to how to synthesize three dimensional, multi-DOF, serial flexure systems. The three most common approaches for designing flexible mechanisms include constraint-based design (CBD), the pseudo-rigid body model (PRBM), and topological synthesis. Constraint-based designers have developed rules of thumb for synthesizing simple, planar, flexure systems using well known flexural elements as building blocks [22-24]. Pseudo–rigid body modeling models compliant mechanisms (CMs) as analogies to rigid–link mechanisms [25-27]. The rigid analog is then modeled using pre–existing rigid mechanism theory and the principle of virtual work to ascertain its kinematic and elastomechanic properties. The PRB model has been used to design precision elastic mechanisms [28-29] and many consumer products. The primary aim of PRBM is to model, rather than synthesize, and so it is not ideally suited to solve the problems that are the target of this thesis. Topological synthesis is based upon computer algorithms that examine a starting shape for a compliant mechanism and then determines how to add/subtract material in order to create concepts that satisfy performance specifications [30-32]. In this method, the computer makes the design decisions that determine the layout of the rigid and flexible elements. This approach is effective for rapid synthesis of non–precision compliant mechanisms for robotics, MEMS and aeronautics. Unfortunately, topology synthesis is not readily applied to solve most precision flexure design problems. The reason for this is that the knowledge needed to integrate the specialized precision design rules with topological synthesis does not currently exist.

The ability to synthesize serial flexure systems is important because of their unique capabilities and advantages over parallel flexure systems. First, serial flexure systems are capable of permitting DOFs that are not possible for parallel flexure systems. A flexure system, for instance, that must permit only three independent translations, i.e. x-y-z, is only possible for a serial flexure system. The reason for this fact is that the moment a single flexible constraint is connected directly from a stage to ground, the stage looses one of its three translations along the constraint's axis. Parallel flexure modules must therefore be stacked in series to achieve three translations. Second, serial flexure systems may be designed to possess larger ranges of motion than same-size parallel flexure systems. Note that the serial flexure system from Figure 1.2 is the same size as the parallel flexure system shown, but the serial flexure system's stage is capable of moving twice the distance of the parallel flexure system's stage. Third, serial flexure systems may be designed to reduce parasitic errors. The parallel flexure system's stage from Figure 1.2 does not possess a perfect translational DOF. The stage follows an arc-like path over its range of motion. The stage of the serial flexure system moves along a "straighter" path because the vertical components of the opposing arc motions of the two rigid stages cancel.

A number of applications require the advantages that are inherent to serial flexure systems. Compliant transmissions that amplify or attenuate forces or displacements are generally serial flexure systems because they consist of at least two stages—an input stage and an output stage. Six-axis, flexure-based nanopositioners must also be serial flexure systems if they are to achieve all six DOFs with comparable stiffness characteristics. Furthermore, some kinematically equivalent serial versions of parallel flexure systems may be designed with *(i)* better elastomechanic characteristics, *(ii)* better dynamic characteristics, *(iii)* greater load capacities, and *(iv)* increased resistance to buckling.

The results of this thesis will enable one to consider all the possible ways of synthesizing serial flexure systems that possess any desired DOFs. A comprehensive library of geometric shapes has been created that links serial flexure designs to desired motions. These "graphical" representations of flexure performance enable one to understand what a flexure should look like if one wishes to achieve certain motion characteristics. After selecting the right design via these shapes, designers may use screw theory to tune and optimize a concept for specific design requirements. This method helps designers ascertain when it is best to use a serial or a parallel flexure system for a given application. New design rules have been found for serial flexure systems. These rules enable designers to rapidly identify concepts that possess desired symmetry, load capacity, and constraint redundancy. They may also be used to optimize the layout of a flexure system so that one may achieve the best flexure system's stiffness and dynamic characteristics.

1.1.2 Optimal Actuator Placement

Precision engineers take great care to model, design, and fabricate flexures such that they exhibit low (often nanometer-level) error motions. It is equally important to place the actuators properly, as an improper layout may cause additional errors (100s to 1000s of nanometers). The optimal actuator placement depends upon coupling of many design parameters and requirements. This coupling presents a complex problem that is not suitably addressed by intuition. It is necessary to have tools/principles that link actuator placement and flexure performance, thereby enabling a designer to set the best layout.

This thesis contains these tools and principles. A mathematical basis is introduced that may be used to determine the best layout for actuators. This information is then encapsulated within geometric shapes that designers may use to visualize the best locations to place actuators. This quantitative/qualitative combination enables rapid actuator layout during concept design, and the modeling that may be used to perform concept optimization and refinement. This approach enables designers to *(i)* visualize the regions wherein actuators should be placed so as to minimize errors, *(ii)* guide designers in selecting these actuators to maximize the decoupling of actuator inputs, and *(iii)* determine actuator forces and displacements that actuate specific degrees of freedom.

The placement of actuators is a generic problem that is relevant to flexible mechanism design. This problem has been a topic of study in structure and robotic design [33-40], but little work has been directed towards the unique requirements of precision flexure systems, specifically those associated with thermal, vibration, fabrication, parasitic, and other error sources.

Parasitic errors are the deviations from a stage's intended motions where the magnitudes of these errors are linked with the magnitudes of the stage's intended motions. Parasitic errors that result from incorrectly actuating a flexure system are systematic, and may therefore be easily characterized by standard error mapping techniques [41-44]. For six DOF flexure systems, error mapping may be used to calibrate the machine via mechanical tuning or active control. For flexure systems with less than six DOFs, errors in non-actuated directions are not easily addressed. It is, therefore, important to know how to arrange the actuators to reduce these errors.

Proper actuator placement depends upon flexure geometry and material properties, therefore the process of specifying actuator placement should occur in early-stage design. In this approach, screw theory is used to generate three dimensional shapes, called actuation spaces, which may be used to visualize the best actuator placement.

One must consider the three dimensional geometries, motions, and deformations of the elastic elements in order to determine the best actuator placement. An experienced designer is not always guaranteed to find the best actuator placement via intuition. The thesis presents a tool that makes it easier for designers to consider all important aspects of the problem, and therefore select the best actuator placement.

The limitations of intuition are demonstrated via the following example. Suppose one wished to rotate the stage in Figure 1.3A about the dotted line that intersects a probe's tip. Engineers would typically intuit that the best actuator placement is that shown in Figure 1.3B. The detail in Figure 1.3B shows that this actuator placement yields a rotation about a line (solid) that is shifted from the desired line (dotted). This shifted rotation causes the probe's tip to displace with an error as shown in Figure 1.3B. If the flexure elements behaved as ideal constraints with infinite stiffness along their axes (this is often how they are visualized), this shift would not occur. In reality, the flexure elements exhibit deformations, e.g. stretching along their axis(es) of constraint, that yield non-ideal motions that manifest as parasitic errors. These deformations cannot be predicted via constraint-based design principles. Later in this thesis, the twist-wrench stiffness matrix (TWSM) will be provided. This matrix may be used to determine that the best actuator placement is that shown in Figure 1.3C.

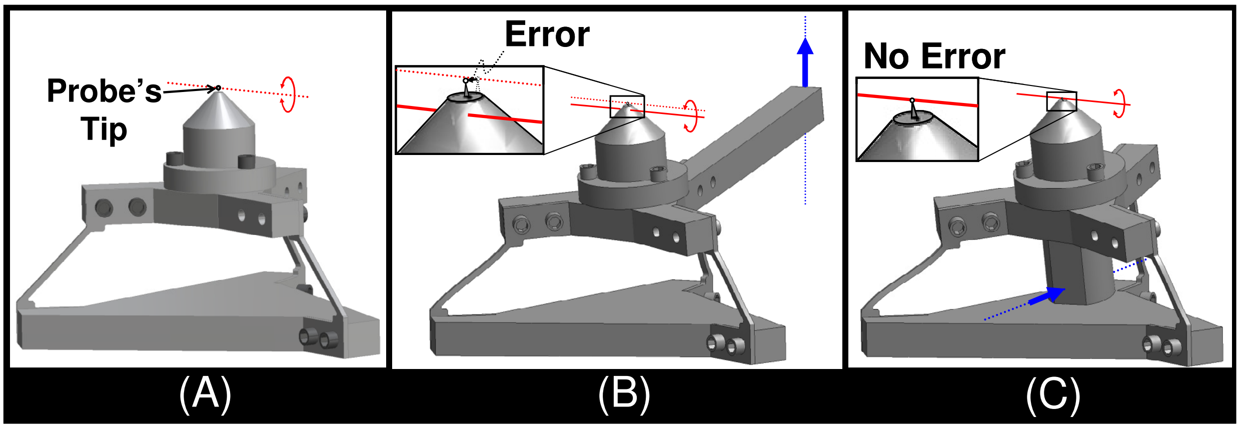

Figure 1.3: *Example flexure system with desired rotation (A), intuitive actuator placement with shifted rotation that yields error (B), correct actuator placement (C).*

1.2 Fundamental Principles

This section reviews the basic principles necessary to *(i)* synthesize parallel flexure modules using geometric shapes and *(ii)* model actuator placements. In following chapters these principles will be applied to *(i)* the synthesis of serial flexure systems and *(ii)* the optimal placement of actuators that minimize error.

1.2.1 Modeling Motions via Screw Theory

All small motions may be modeled as a 6x1 vector called a twist [45-48]. A twist may be visualized as a line about, or along, which a stage rotates and/or translates. The pitch of a twist is defined as the ratio of the distance a stage translates along the twist's axis to the coupled rotation about this axis. A general displacement twist, **T**, is defined as:

$$\mathbf{T} = [\boldsymbol{\Delta\theta} \quad \boldsymbol{\delta}]^T = [\boldsymbol{\Delta\theta} \quad ((\mathbf{c}\times\boldsymbol{\Delta\theta}) + p\cdot\boldsymbol{\Delta\theta})]^T, \tag{1.1}$$

where $\Delta\boldsymbol{\Theta}$ is a 1x3 vector that points along the twist's axis. The magnitude of $\Delta\boldsymbol{\Theta}$ represents the angle through which the stage rotates. The 1x3 vector, $\boldsymbol{\delta}$, is the linear displacement of the chosen coordinate system's origin. The location vector, **c**, is any 1x3 vector that points from the origin to any point along the twist's axis. The twist's pitch is p. These general twist parameters are depicted in Figure 1.4A. If the twist's pitch is zero or infinite, the twist describes a purely rotational or translational motion respectively. If its pitch is any other value, the twist describes a screw motion. The rotational, translational, and screw DOFs shown in Figure 1.4B-D may be modeled as twists with pitch values of zero, infinity, and a finite value respectively. In this thesis rotations are depicted as red lines, translations are depicted as black arrows, and screws are depicted as green lines.

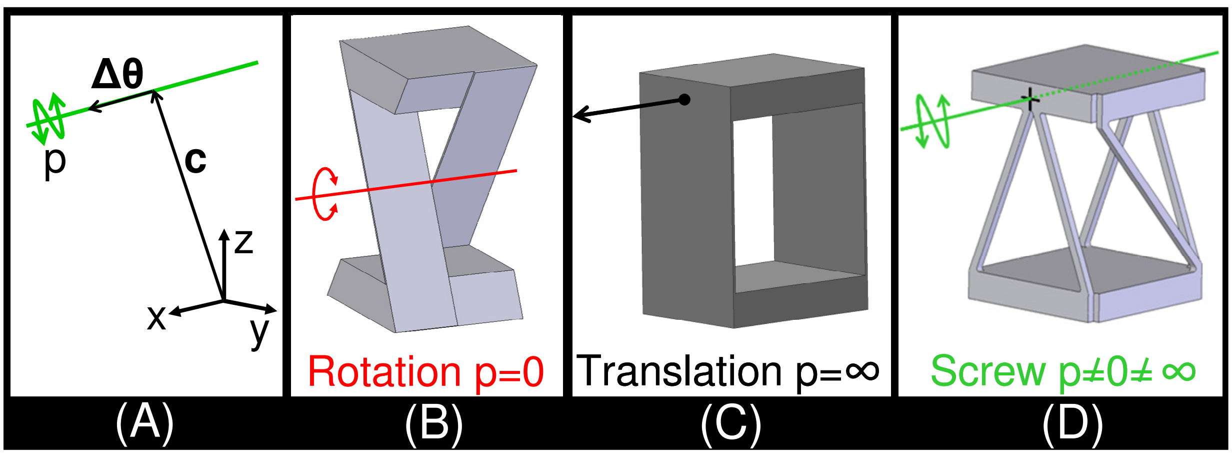

Figure 1.4: *Twist parameters defined (A), rotation twist (B), translation twist (C), screw twist (D).*

1.2.2 Freedom Space

The parallel flexure systems shown in Figure 1.4 are single DOF systems. Consider the 2-DOF parallel flexure system shown in Figure 1.5. It is capable of guiding two independent rotation DOFs as shown in Figure 1.5A-B. If these two independent rotations are simultaneously actuated and the relative ratio of their rotations are controlled, the resulting rotation line may be any of the lines within the disk or pencil shown in Figure 1.5C. This disk is the system's freedom space [19-21]. Freedom space is the geometric shape that visually represents a system's kinematics, i.e. all the twists the flexure permits. The freedom space of a system is modeled using a twist matrix, $[T]$, defined as

$$[T]=[\mathbf{T}_1 \quad \mathbf{T}_2 \quad \cdots \quad \mathbf{T}_n], \tag{1.2}$$

where n is the number of independent twists or DOFs that the system possesses. For the example flexure system from Figure 1.5, n is 2.

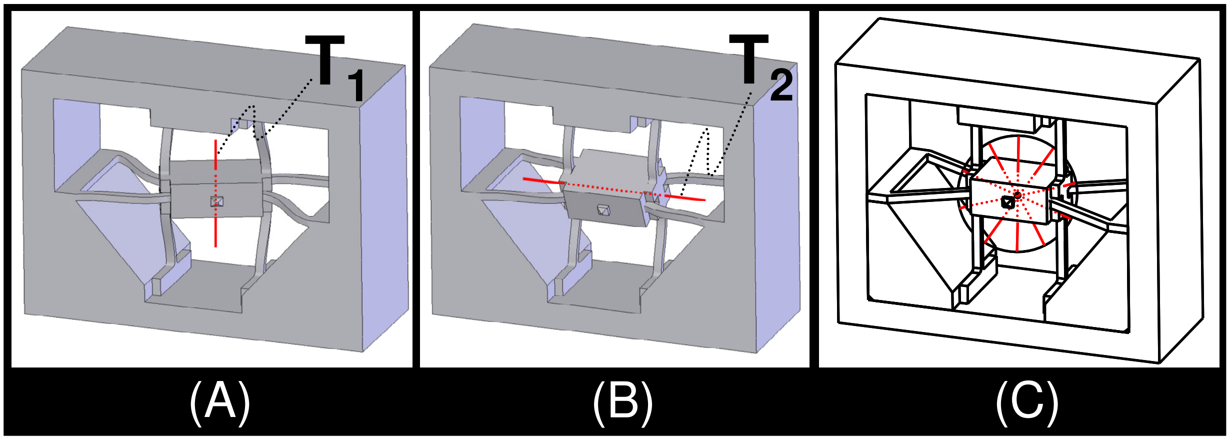

Figure 1.5: *Example flexure system with two rotational DOFs (A) and (B). The flexure system's freedom space is a disk of rotations (C).*

1.2.3 Modeling Flexible Constraints via Screw Theory

Any flexible constraint may be modeled using 6x1 vectors called wrenches [45-48]. A wrench may be visualized as a line along or about which a force and/or moment act. A general wrench, **W**, is defined as

$$\mathbf{W}=[\mathbf{f} \quad \boldsymbol{\tau}]^T=[\mathbf{f} \quad ((\mathbf{r}\times\mathbf{f})+q\cdot\mathbf{f})]^T, \tag{1.3}$$

where **f** is a 1x3 force vector that points along the wrench's axis, **τ** is a 1x3 vector that represents the moment about the coordinate system's origin due to the wrench, **r** is any 1x3 location vector that points from the origin to any point along the wrench's axis, and q is the ratio of the moment about the wrench's axis to the force along this axis. These general wrench parameters are depicted in Figure 1.6A. If the wrench's q-value is zero or infinite, the wrench describes a pure force or moment respectively. In this thesis pure forces are depicted as blue lines, pure moments are depicted as black lines with circular arrows, and wrenches with non-zero, finite q-values are depicted as orange lines. The differences between these lines are shown in Figure 1.6B. *It is important to note that only pure force wrenches model flexible constraints (q=0).* If a constraint is long and slender, as those shown in Figure 1.6C, a single pure force wrench oriented along the constraint's axis accurately models the constraint. If the flexible constraint is a thin blade flexure, as those shown in Figure 1.6D, the pure force wrenches that lie on the plane of the blade accurately model the constraint. Any three of these pure force wrenches that are not parallel, and do not intersect at a common point, will be independent and will also model the kinematics of the blade flexure. Note that if the three blue lines from one of the blade flexures shown in Figure 1.6D were replaced by long slender constraints, the kinematics of the stage would remain unchanged. If, however, one or more of those constraints was removed, the stage would possess unwanted DOFs.

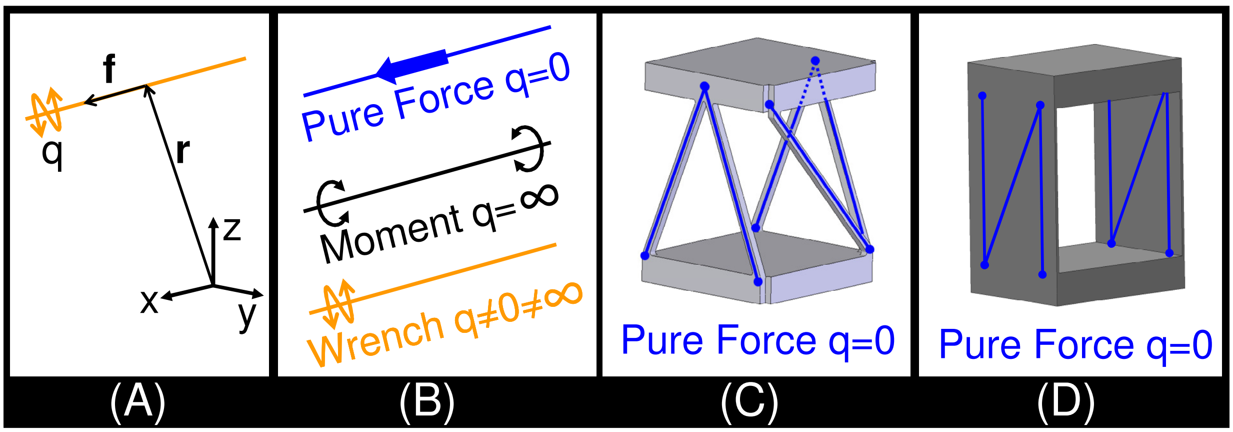

Figure 1.6: *Wrench parameters defined (A). Color coded wrench types (B). Pure force wrenches modeling long slender constraints (C). Pure force wrenches modeling thin blade flexures (D).*

1.2.4 Constraint Space

Consider again the parallel flexure system from Figure 1.5. The eight pure force wrenches shown in Figure 1.7A accurately model the system's flexible constraints. These eight constraints or wrenches are not all non-redundant or independent. Some of the constraints could be removed without affecting the system's kinematics. Such constraints are called redundant constraints. If a matrix were constructed using the eight wrenches from Figure 1.7A, Gaussian elimination would reveal that only four of these wrenches are independent. More redundant constraints could be added to the flexure system without changing the system's DOFs. All of these redundant constraints could be found by linearly combining the system's four independent wrenches. The wrenches of these redundant constraints would lie on, or within, a space called constraint space. Constraint space is the geometric shape that visually represents the regions where redundant constraints could be placed without altering a system's DOFs. The constraint space of the flexure system from Figure 1.7 is shown in Figure 1.7B. The constraint space of any system is modeled using a wrench matrix, $[W]$, defined as

$$[W]=[\mathbf{W}_1 \quad \mathbf{W}_2 \quad \cdots \quad \mathbf{W}_m], \tag{1.4}$$

where m is the number of independent wrenches or non-redundant constraints the system possesses. For the example flexure system from Figure 1.7, m is 4. Although mathematically it consists of wrenches of all q-values, the constraint space of interest to flexure designers consists of only pure force wrenches as only these wrenches model flexible constraints. The constraint space of use to flexure designers, therefore, consists of every pure force constraint line that lies on a plane outlined in blue shown in Figure 1.7B and every pure force constraint line that intersects a common point that lies on that plane represented by a sphere of blue lines. Note that wrenches $\mathbf{W_1}$, $\mathbf{W_4}$, $\mathbf{W_5}$, and $\mathbf{W_8}$ belong to this sphere shown in Figure 1.7C and $\mathbf{W_2}$, $\mathbf{W_3}$, $\mathbf{W_6}$, and $\mathbf{W_7}$ lie on the plane of the constraint space. Any other constraint selected from this plane or sphere would be redundant and would affect the system's elastomechanics, dynamics, and load capacity without changing the system DOFs.

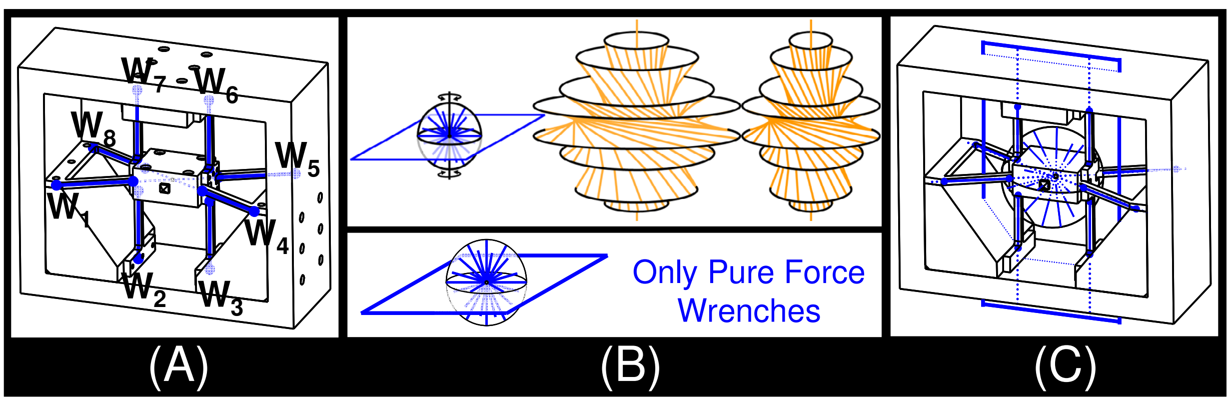

Figure 1.7: *Eight wrenches that model the system's constraints (A). Mathematical and practical system constraint space (B). System constraints lie within its constraint space (C).*

1.2.5 Constraint and DOF Relationship

The relationship between the number of non-redundant constraints, m, and DOFs, n, any given parallel flexure system possesses is given by

$$6 = m + n. \tag{1.5}$$

The relationship between a constraint, $\mathbf{W}$, and a DOF, $\mathbf{T}$, that the constraint permits is given by

$$\mathbf{W}^{\mathbf{T}} \cdot [\Phi] \cdot \mathbf{T} = 0\,, \tag{1.6}$$

where

$$[\Phi] = \begin{bmatrix} \mathbf{0}_{3\times 3} & \mathbf{I}_{3\times 3} \\ \mathbf{I}_{3\times 3} & \mathbf{0}_{3\times 3} \end{bmatrix}, \tag{1.7}$$

and where $\mathbf{0_{3x3}}$ is a 3x3 matrix of zeros and $\mathbf{I_{3x3}}$ is a 3x3 identity matrix. In essence, Equation 1.6 suggests that DOF twists are orthogonal to their complementary constraint wrenches. Or, in other words, for a given constraint wrench, **W**, the DOF twist, **T**, is a motion that produces no work. Intuitively this means that DOFs are the motions that produce the least resistance from the constraints.

Another useful form of Equation 1.6 is found by combining Equation 1.1, Equation 1.3, Equation 1.6 and Equation 1.7 and is given by

$$p + q = d \cdot \tan(\varphi)\,, \tag{1.8}$$

where p is the twist's pitch, q is the wrench's q-value, and d and φ are the shortest distance and skew angle between the twist's and wrench's lines of action respectively. These parameters are shown in Figure 1.8A.

The relationship between a system's constraint space, $[W]$, as defined in Equation 1.4 and freedom space, $[T]$, as defined in Equation 1.2 is given by

$$[W]^{\mathbf{T}} \cdot [\Phi] \cdot [T] = 0\,, \tag{1.9}$$

where $[\Phi]$ is defined in Equation 1.7. Equation 1.9 suggests that a parallel flexure system's freedom space is the null space of that system's constraint space. It is important to note, therefore, that there is a unique linking between a system's freedom space and its constraint space. The flexure system from Figure 1.5C possesses the disk-like freedom space because its constraints lie within this freedom space's complementary constraint space as shown in Figure 1.7C. Any parallel flexure system that possesses constraints, which lie within the constraint space shown in Figure 1.8B, will possess rotational DOFs that lie within the freedom space also shown in Figure 1.8B. The uniquely linked freedom and constraint space pair from Figure 1.8B helps designers recognize the region where constraints must be placed to synthesize a parallel flexure system that possesses two intersecting rotational DOFs.

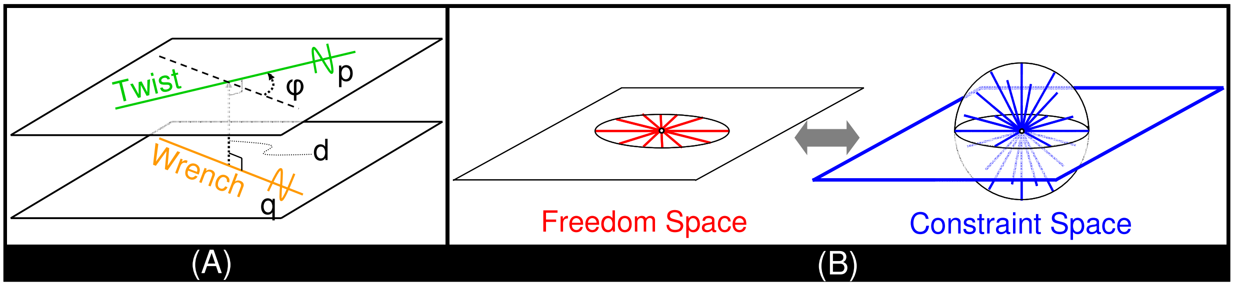

Figure 1.8: *Complementary twist and wrench parameters (A). Uniquely linked freedom and constraint space pair for the example flexure system (B).*

1.2.6 Modeling Actuators via Screw Theory

Any actuation force may also be modeled as a wrench vector as defined in Equation 1.3 where the wrench's line of action is synonymous with the actuator's line of action. This thesis is scoped to work with linear actuators therefore the q-values are generally zero. Pure moment actuators or screw-like actuators may, however, be modeled as wrenches with q-values equal to infinity or non-zero, finite values respectively. The linear actuators shown as a pure force lines with thick arrows in Figure 1.3B and Figure1.3C may be modeled as wrenches with q-values that are equal to zero.

1.2.7 Decomposing Wrench and Twist Vectors

The six components contained in **W** are not always easily determined given a wrench vector's line of constraint or actuation. The process of deconstructing a wrench vector into its location vector, **r**, orientation vector, **f**, and q-value is called wrench decomposition. The wrench's force vector, **f**, is simply the first three components of **W**. The wrench's q-value is

$$q = \frac{\mathbf{f} \bullet \boldsymbol{\tau}}{\mathbf{f} \bullet \mathbf{f}}, \tag{1.10}$$

where $\boldsymbol{\tau}$ contains the last three components of **W**. The wrench's location vector, **r**, may be any vector that satisfies this system of equations

$$\begin{bmatrix} q & -r_z & r_y \\ r_z & q & -r_x \\ -r_y & r_x & q \end{bmatrix} \mathbf{f}^T = \boldsymbol{\tau}^T. \tag{1.11}$$

Twist vectors may also be decomposed using Equation 1.10 and Equation 1.11 by substituting $\mathbf{\Delta\Theta}$ for $\mathbf{f}$, $\boldsymbol{\delta}$ for $\boldsymbol{\tau}$, p for q, and $\mathbf{c}$ for $\mathbf{r}$.

1.3 Thesis Overview

Chapter 2 discusses the principles necessary for synthesizing serial flexure systems using freedom and constraint space pairs. Chapter 3 describes and parameterizes every freedom and constraint space pair for any flexure system. Chapter 4 discusses the principles necessary for optimally placing actuators for parallel flexure systems. Chapter 5 concludes the thesis with a review of its major contributions and a discussion of future flexure research that is currently underway. Appendix A contains the proof of how all the spaces described in Chapter 3 were found and Appendix B contains the MATLAB code necessary for generating a parallel flexure system's best actuator scheme.

CHAPTER 2:
Serial Flexure Synthesis

This chapter demonstrates how serial flexure systems may be synthesized via FACT. The comprehensive chart of freedom and constraint space pairs is introduced and basic principles of serial synthesis are discussed in the context of these spaces. The steps of the FACT design process are described and a case study is presented.

2.1 FACT Chart

There are only 50 freedom and constraint space pairs called types. These types are described in detail in Chapter 3, derived in Appendix A, and are shown in Figure 2.1. The chart from this figure contains a great deal of information and the reader is not expected to understand its full significance at this point in the thesis. Notice, however, that all the types belong to one of seven columns. Each column pertains to the number of DOFs the type's freedom space possesses. Within each column, the freedom and constraint space pairs are labeled with type numbers. The freedom space of each type is shown to the left of a small, grey, double-sided arrow in the middle of each column and the constraint space of the same type is shown to the right of the same arrow. The freedom and constraint spaces in Figure 2.1 include all twists and wrenches of all pitch and q-values respectively. This chart is a visual representation of all screw systems [49-62] and their complementary spaces. Although much work has been done to classify screw systems [63-68], this chart provides a comprehensive classification of screw systems that is suited for flexure synthesis. Note that the spaces within the chart are symmetric about the 3 DOF column. This means that the freedom spaces within the *n* DOF column are identical to the constraint spaces within the *6-n* DOF column. The proof for this symmetry is found in Hopkins [21].

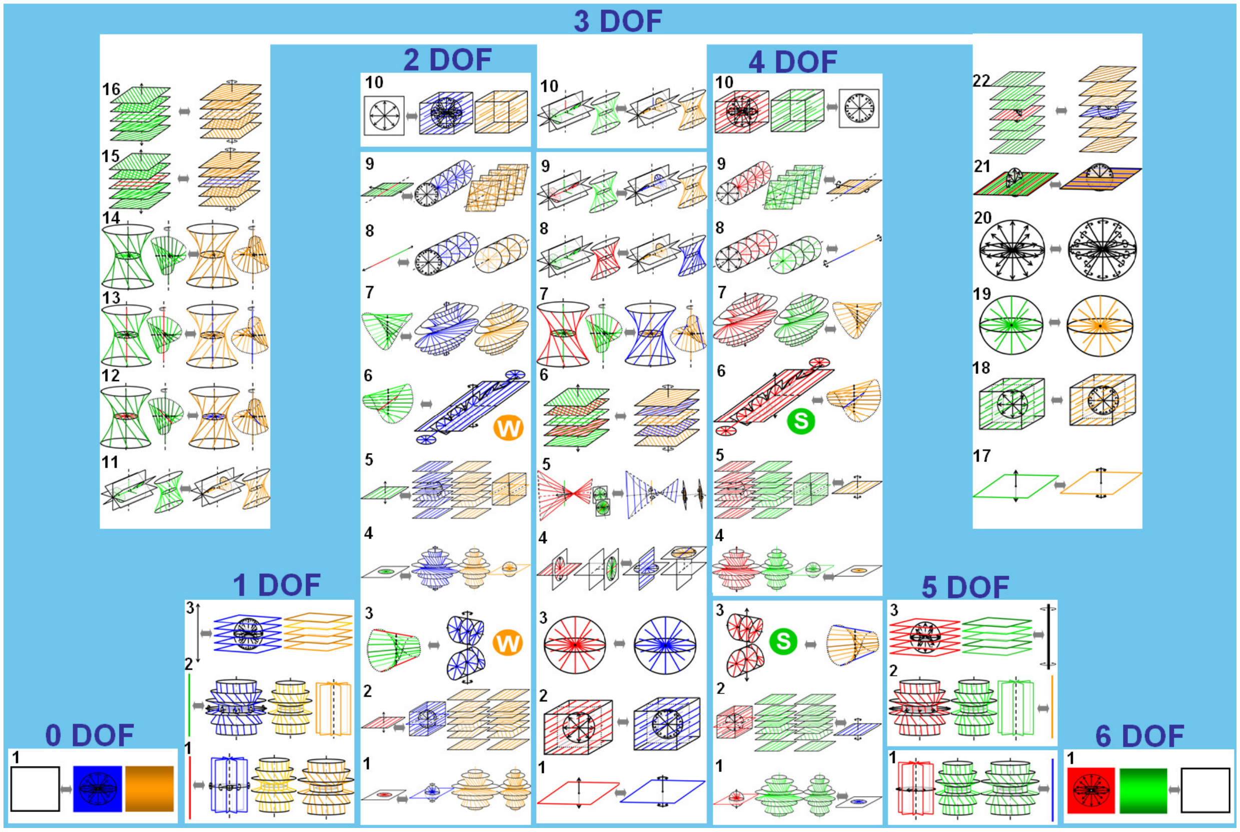

Figure 2.1: *Mathematically comprehensive FACT chart*

Recall from Section 1.2.3 that flexible constraints may only be modeled using pure force wrenches with q-values equal to zero. Thus the chart from Figure 2.1 is only useful to flexure designers if the constraint spaces contain only pure force wrenches (blue lines). Furthermore, practical constraint spaces must possess enough independent, pure force wrenches to produce the correct complementary freedom space. In other words, if a system's freedom space possesses n DOFs, its constraint space must possess 6-n independent pure force wrenches according to Equation 1.5 or it is not a useful constraint space. The chart, therefore, becomes useful to flexure designers if all constraint spaces that don't satisfy this criterion and all orange wrenches and pure torque lines (black lines with circular arrows) are removed from the chart of Figure 2.1. This chart is shown in Figure 2.2. The thick black line that separates the types with constraint spaces from the types without constraint spaces is called the "parallel pyramid". Parallel flexure systems may only possess freedom spaces that lie within the parallel pyramid. All flexure systems that possess freedom spaces that lie outside of this pyramid may only be synthesized by stacking parallel flexure models from within the parallel pyramid in series. The reason for this fact is that the freedom spaces that lie outside of the pyramid don't have constraint spaces from which the designer may select constraints. Note that the freedom and constraint space pair shown in Figure 1.8B of the 2 DOF flexure system from Chapter 1 is the first type at the bottom of the column labeled 2 DOF in Figure 2.2 and the pair lies within the parallel pyramid. Note also that the freedom space that contains only three pure translations (sphere of black arrows) is the 20th type in the column labeled 3 DOF and lies outside of this pyramid. This freedom space is only possible for a serial flexure system to possess.

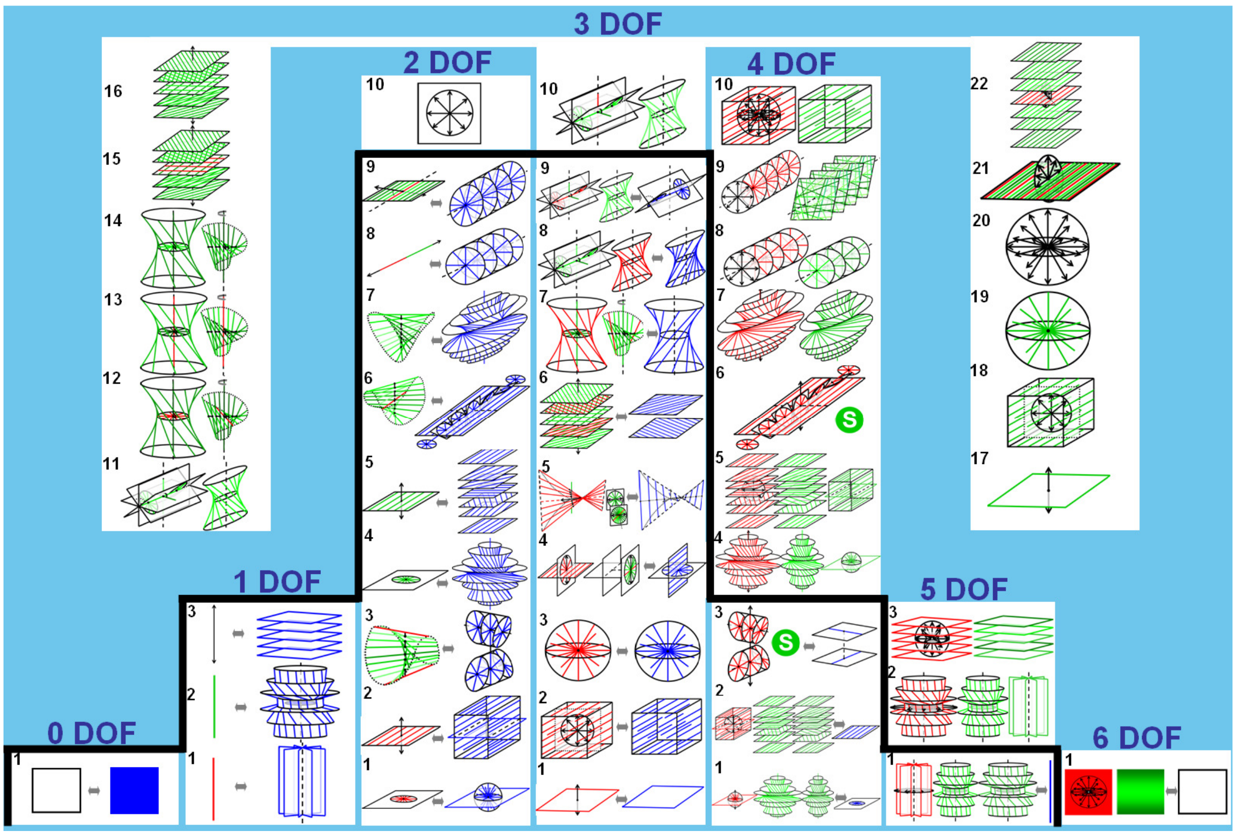

Figure 2.2: *Practical FACT chart for flexure synthesis*

2.2 Serial Synthesis Principles

This section discusses the principles necessary to synthesize serial flexure systems using geometric shapes as tools. The principles that govern underconstraint and kinematic equivalence are set forth.

2.2.1 Parallel and Serial Constraint Fundamentals

According to constraint-based design, flexures connected in parallel retain the DOFs that the individual constraints have in common, whereas flexures connected in series will possess the DOFs of the individual constraints combined [22]. Each constraint shown in the left and center images of Figure 2.3A permits the two-dimensional stage to possess a rotational DOF and a translational DOF that is perpendicular to the axis of the constraint. If these constraints are combined in parallel, the stage retains only the common rotational DOF shown on the right side of Figure 2.3A. If the translational DOF parallel flexure module shown in Figure 2.3B is stacked in series with itself, not only does the final stage inherit each module's translational DOF, but it also inherits every combination of those translational DOFs. The stage shown in Figure 2.3B may, therefore, move with any translation in the plane of the flexure as represented by the disk of arrows. It will later be demonstrated that these basic principles also apply to the design of flexure systems using geometric shapes.

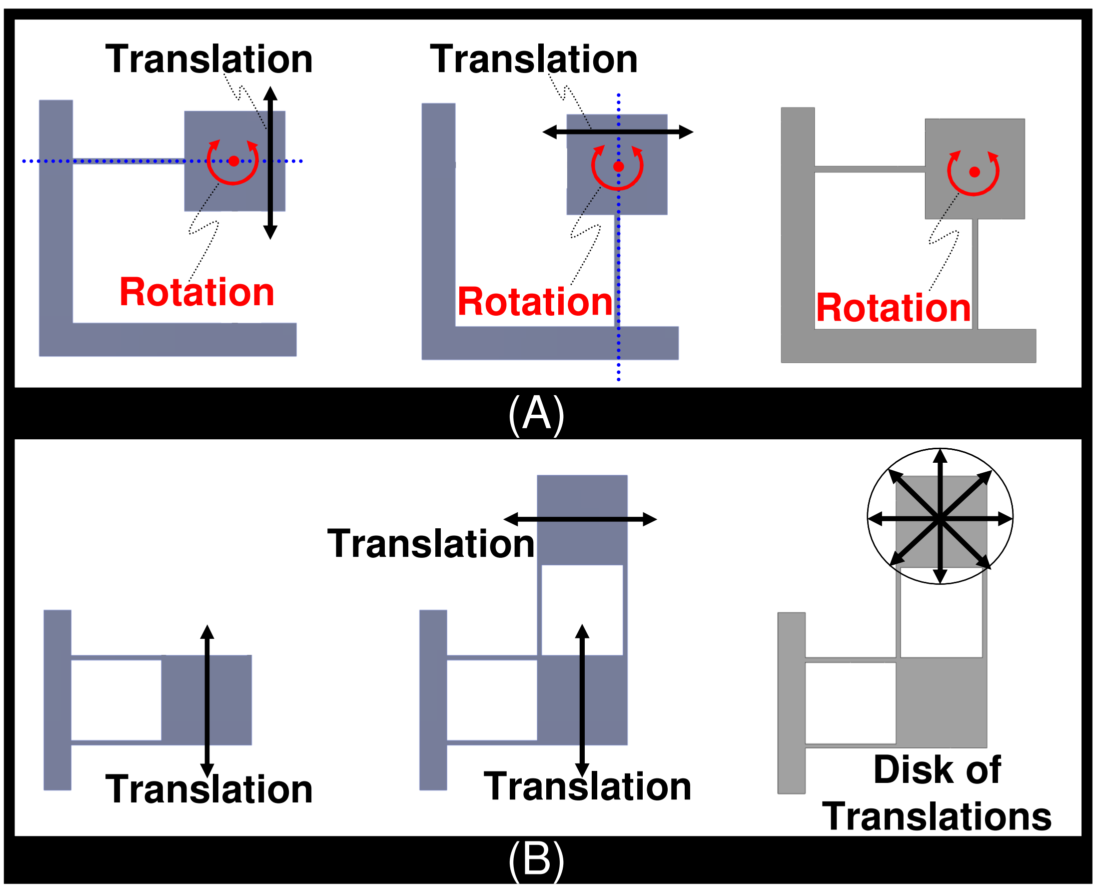

Figure 2.3: *Flexural elements in parallel (A) and in series (B)*

2.2.2 Underconstraint

Consider the 3 DOF system that is constrained by the flexure blade in Figure 2.4A. The system's stage is permitted to move with a translation and two rotations as depicted by the three twists that are shown in the figure. If this parallel flexure module were stacked on top of itself as shown in Figure 2.4B, one might expect the new stage to possess 6 DOFs—3 from each module. Although the stage inherits the DOFs of both parallel modules as shown in Figure 2.4B, the stage only possesses 5 DOFs because the rotations labeled $\mathbf{T_3}$ and $\mathbf{T_6}$ are redundant meaning that both parallel modules possess the same DOF. *When a serial flexure system possesses one or more redundant DOFs, the system is underconstrained.* When the system's stage and ground are held fixed, underconstrained systems possess one or more intermediate stages that are not fully constrained. Under such conditions these intermediate stages are free to move with the DOFs that are redundant. Consider, for example, holding the stage and ground of the serial flexure system from Figure 2.4B fixed. The intermediate stage labeled in the figure would remain free to rotate about the axis of the rotation labeled $\mathbf{T_3}$ and $\mathbf{T_6}$. It will later be demonstrated how geometric shapes may be used in conjunction with the principles of this section to avoid or utilize underconstraint in the design process of serial flexure systems.

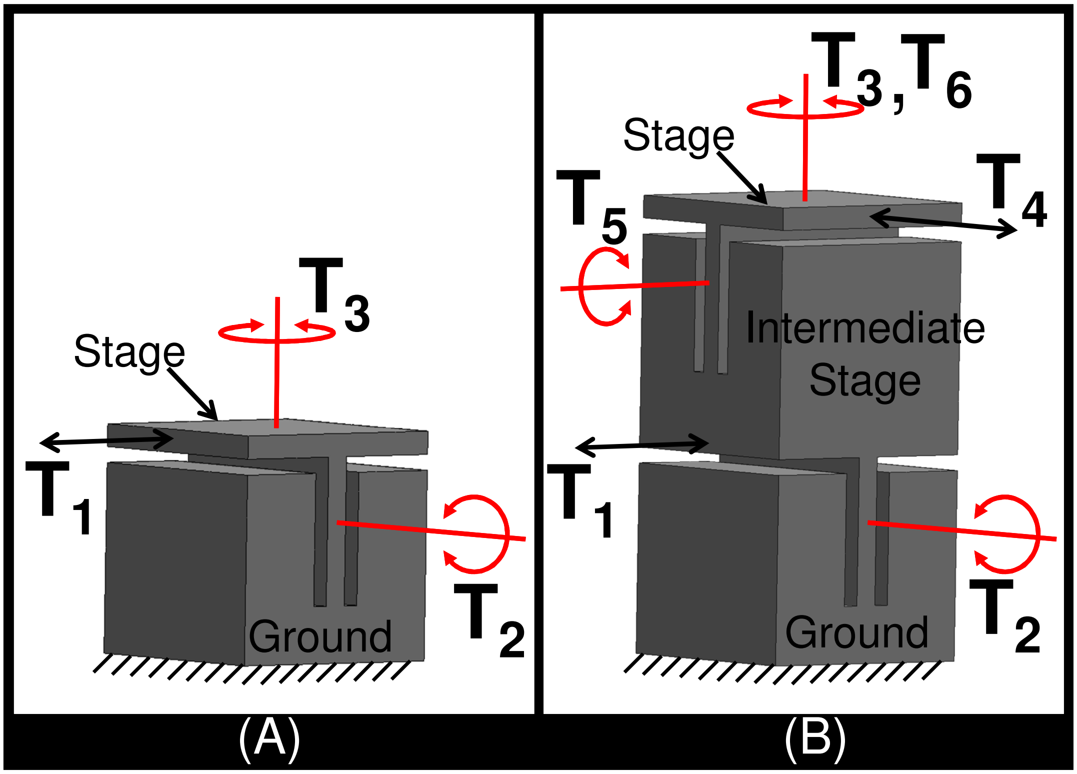

Figure 2.4: *3 DOF parallel flexure system (A). 5 DOF stacked serial flexure system (B).*

Advantages and challenges are associated with flexure systems that are underconstrained. Parallel modules stacked in series like those shown in Figure 1.2B and Figure 2.4B achieve a greater range of motion than the individual parallel modules alone. The reason for this increase in range is that the redundant motions of each module contribute to the full stroke of the final stage. A properly underconstrained system may, therefore, markedly increase a flexure's stroke to size ratio. Unfortunately, underconstrained flexure systems generally have poor dynamic characteristics. Reducing the mass of the intermediate stages and stiffening the flexible elements that connect them together helps mitigate this problem.

2.2.3 Intermediate Spaces

Consider the planar constraint space of the parallel flexure module from Figure 2.4A. This constraint space is shown with a different orientation in Figure 2.5A. Note that the constraint lines of the flexure blade shown in Figure 2.5B lay on the plane of the system's constraint space. The constraint space belongs to the first type in the 3 DOF column in Figure 2.2. The constraint space's complementary freedom space, shown in Figure 2.5A, consists of all rotational lines that lie on the plane of the constraint space as well as a translation perpendicular to this plane. Notice from Figure 2.5B that the three DOF twists shown in Figure 2.4A belong to the freedom space of the system. If the parallel flexure module is stacked in series with itself as shown in Figure 2.5C, the resulting freedom space of the serial stage not only possess twists from the planar freedom spaces of each individual parallel flexure module, but it also possesses the twists that result from combining the twists of these spaces. The freedom space of the serial flexure system's stage, therefore, contains *(i)* every rotational line that lies on the planes that intersect the dotted line shown in Figure 2.5D and *(ii)* every translation that is perpendicular to the same line. Note that the two planar freedom spaces of the two individual parallel modules both lie within this freedom space. The two freedom spaces that combine to form the serial system's freedom space are called intermediate freedom spaces. These spaces each contain three independent twists that are labeled in Figure 2.5C. These twists may be combined to generate the other twists within each space. Together, however, the six twists combine to generate all the twists shown in the freedom space of Figure 2.5D. If Gaussian elimination were performed with these six twists, only five of them would be shown to be independent. It is not surprising therefore, that the freedom space shown in Figure 2.5D belongs to the first type in the 5 DOF column of the chart shown in Figure 2.2 because the system possesses only five DOFs.

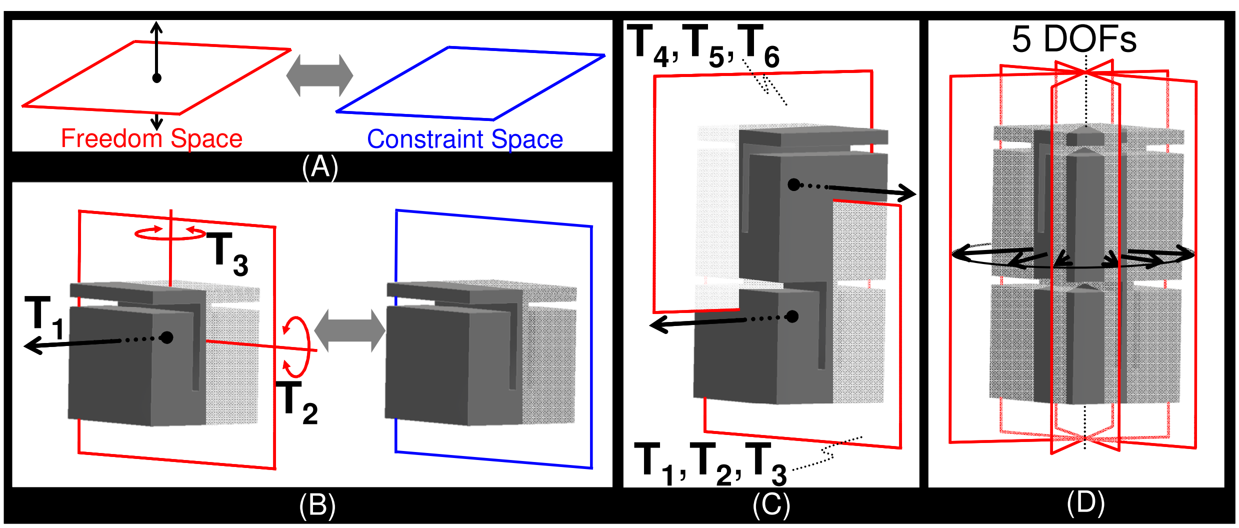

Figure 2.5: *3 DOF Type 1 freedom and constraint spaces (A). Spaces imposed on the 3 DOF parallel flexure module (B). Intermediate freedom spaces (C) sum together to produce the 5 DOF freedom space of the serial flexure system (D).*

Intermediate freedom spaces always exist within a system's freedom space. The total number of independent twists from all of the intermediate freedom spaces combined must equal the number of DOFs within the desired system freedom space. If this condition is not satisfied, the intermediate freedom spaces of the stacked parallel flexure modules will not produce the correct DOFs.

Intermediate freedom spaces help designers identifying whether or not a serial flexure system is underconstrained. If the sum of the number of DOFs of each intermediate freedom space is more than the number of DOFs of the system's freedom space, the serial flexure system will be underconstrained and will possess redundant DOFs. Note that the serial flexure chain from Figure 2.5 is underconstrained because the sum of the number of DOFs from each intermediate freedom space is six, and six is greater than the number of DOFs of the system's freedom space (5 DOFs).

2.2.4 Kinematic Equivalence

The freedom and constraint spaces of the serial flexure chain from Figure 2.5D are shown as Type 1 in the 5 DOF column of the chart of Figure 2.2 and again in Figure 2.6A. Note that the freedom space not only contains rotation lines and translations, it also contains screws. The shapes that describe the locations and orientations of these screws are described in detail in Chapter 3. The constraint space of the system is a constraint line that is collinear with the line of intersection of the planes of rotation from the freedom space. Note that a single flexure wire constraint also possesses the same freedom and constraint spaces as shown in Figure 2.6B. A flexure wire is a long, slender flexible element. The flexure wire and the serial flexure chain from Figure 2.6C are said to be kinematically equivalent because they possess the same freedom space. The parallel and serial flexure systems shown in Figure 2.6D are also kinematically equivalent because the flexure wires of the parallel flexure system were replaced with the kinematically equivalent flexure chains of the serial flexure system. Although kinematically equivalent flexure elements may be substituted without changing a system's DOFs, the system's elastomechanics, dynamics, and manufacturability may be altered to satisfy desired design requirements.

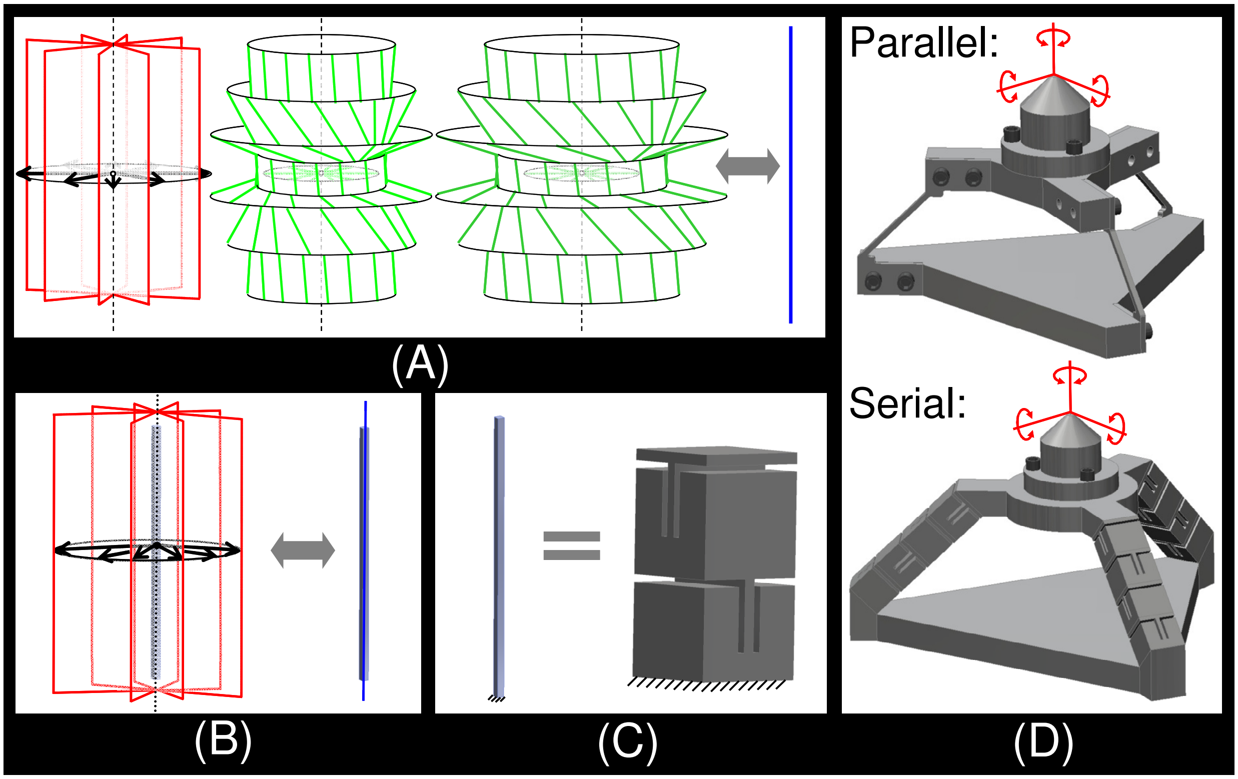

Figure 2.6: *5 DOF Type 1 freedom and constraint space (A). Spaces imposed on the 5 DOF wire flexure (B). Wire flexure and serial flexure chain are kinematically equivalent (C). Two kinematically equivalent flexure systems (D).*

2.3 FACT Design Process

This section describes the 6 steps of the FACT design process shown in Figure 2.7 that is used to synthesize precision flexure systems. A lead screw flexure will be designed using FACT.

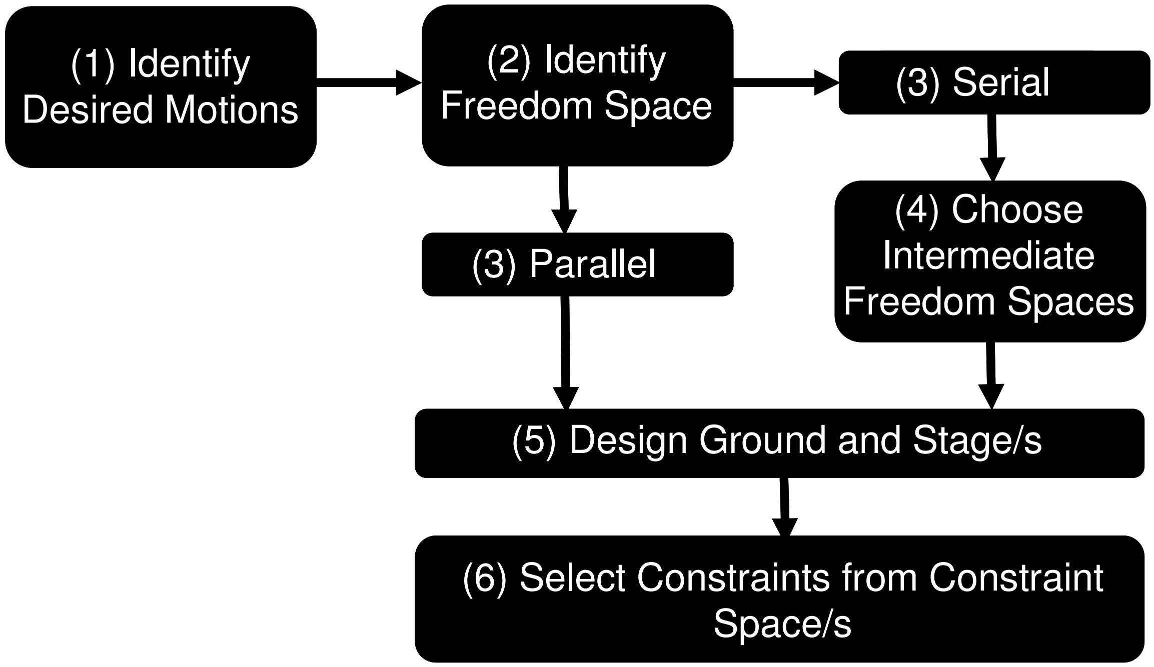

Figure 2.7: *Steps of the FACT design process*

Step (1): Identify Desired Motions

The designer must first recognize which DOFs the system should possess.

Step (2): Identify Freedom Space

The designer must then identify the freedom space that embodies the DOFs that were specified in Step (1). This freedom space will belong to the column from Figure 2.2 that pertains to the number of DOFs the system should possess. If the designer is not familiar enough with the spaces to recognize which space contains the desired DOFs, a computer code could be written using the materials from Appendix A and Hopkins [21] that identifies this freedom space from the appropriate column.

Step (3): Parallel or Serial?

The designer must then decide whether to synthesize a parallel or a serial flexure system. If the freedom space identified in Step (2) does not belong inside the parallel pyramid from Figure 2.2, then the designer must synthesize a serial flexure system to achieve the desired DOFs. If, however, the freedom space does belong inside the parallel pyramid, then parallel or serial concepts exist that achieve the desired DOFs. Designers are generally encouraged to first attempt generating parallel concepts before generating serial concepts because parallel concepts *(i)* do not suffer from stacked axis errors, *(ii)* are easier to design and fabricate, and *(iii)* generally possess better dynamic characteristics than serial flexure systems.

Step (4): Choose Intermediate Freedom Spaces

If the designer chooses to synthesize a serial flexure system, intermediate freedom spaces must be selected. The intermediate freedom spaces must exist in the chart of Figure 2.2 to the left of the column that contains the selected freedom space because the intermediate freedom space must exist within the freedom space. The intermediate freedom spaces must also belong within the parallel pyramid. If the designer has trouble recognizing feasible intermediate freedom spaces, a computer code could be written that identifies them. This code would rely on the materials from Appendix A and theory in Hopkins [21]. Designers may select any number of viable intermediate freedom spaces. An intermediate freedom spaces may be selected multiple times. The number of intermediate freedom spaces determines the number of rigid stages or conjugated elements the flexure system will possess. The fewer the stages, the less complex the design, the easier to fabricate and assemble, and the better will be the dynamic characteristics. Serial flexure systems require a minimum of two intermediate freedom spaces. Intermediate spaces that possess orthogonal features generally produce designs that are easily fabricated.

Step (5): Design Ground and Stage/s

The ground and rigid stages must then be designed. If the designer chose to synthesize a parallel flexure system, only one stage should be designed. If the designer chose to synthesize a serial flexure system, the number of stages should equal the number of intermediate freedom spaces that were selected from the previous step. The rigid stages should be far enough away from each other that they do not collide as they move.

Step (6): Select Constraints from Constraint Space/s

If the designer is synthesizing a parallel flexure system, he/she must select constraints from the constraint space of the system's freedom space. These constraints must connect the ground to the rigid stage. If the designer is synthesizing a serial flexure system, constraints from the constraints space of the first intermediate freedom space must be selected such that they connect the ground to the first intermediate rigid stage. Then constraints from the constraint space of the second intermediate freedom space must be selected such that they connect the first intermediate rigid stage to the second intermediate rigid stage. This process continues until constraints have been selected from the constraint spaces of every intermediate freedom space and all the stages have been stacked together to form the full serial flexure chain.

It is important that a suitable number of non-redundant constraints are selected from each freedom space's constraint space such that the rigid stage possesses the desired DOFs. If a freedom space contains n DOFs, one may apply Equation 1.5 to determine the number of non-redundant constraints, m, which should be selected from the freedom space's complementary constraint space. Not all constraints selected from the constraint space will be non-redundant. The designer could arbitrarily select m constraints from the constraint space and then apply Gaussian elimination to ensure that the wrenches of the constraints selected are all independent. The designer could also use the constraint space's sub-constraint spaces to ensure constraint independence. Every constraint space possesses a certain number of sub-constraint spaces that guide the designer in selecting independent constraints. Every sub-constraint space for any constraint space is derived and described in Hopkins [21].

2.3.1 Lead Screw Flexure Case Study

In this section, a serial flexure system will be designed using the steps of the FACT design process. Consider the desktop lathe shown in Figure 2.8A. A cross-section of the lathe is shown in Figure 2.8B. A mechanism must be designed that transforms the lead screw's rotation into the carriage's translation. If the hex nut on the lead screw were rigidly attached to the carriage, the carriage would stick and slip as it translates along the bearing rails because multiple elements of the system would be constraining the same DOFs. A flexure system should be designed, therefore, that stiffly links the hex nut to the carriage in certain directions while allowing for compliance in other directions. In this way, imperfections in the lead screw's geometry and alignment are corrected for and the bearing rails don't redundantly constrain the system.

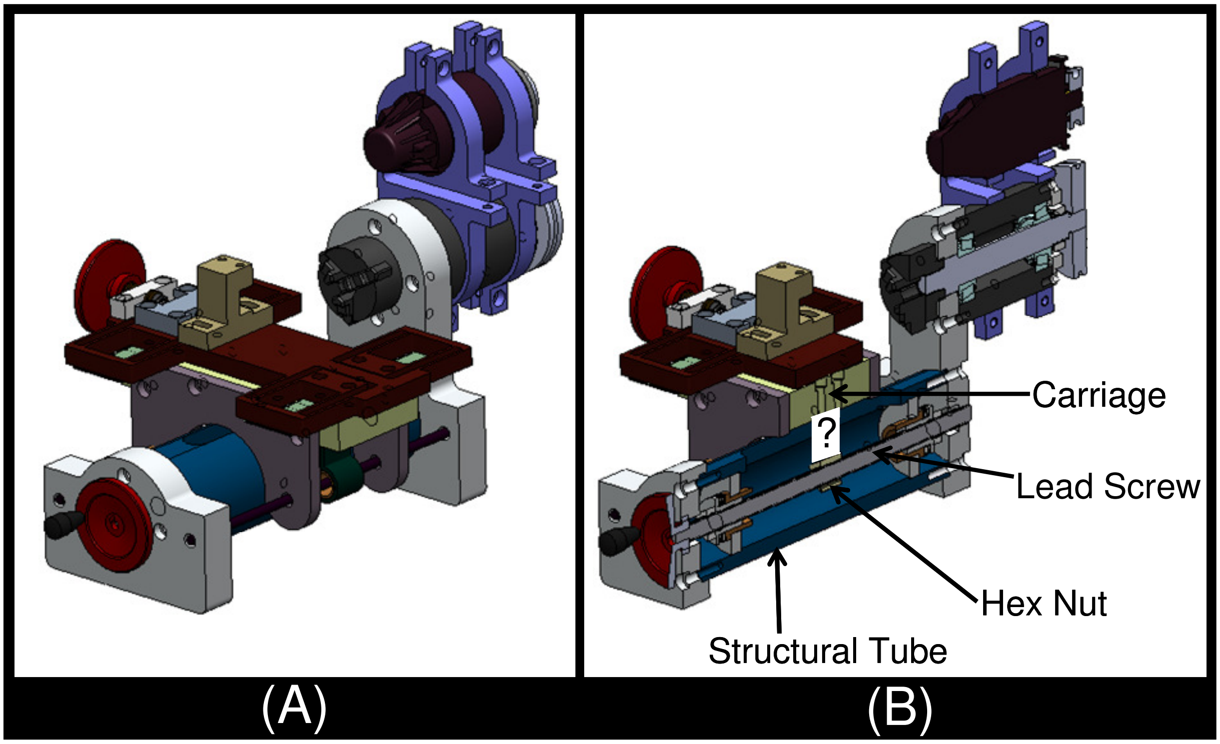

Figure 2.8: *Desktop lathe (A) and cross-section (B).*

Step (1): Identify Desired Motions

The flexure system should be stiff along the axis of the lead screw such that the carriage immediately translates with the hex nut. The rotational motion about the lead screw's axis should also be stiff such that the friction between the hex nut and threads of the lead screw is overcome as the lead screw rotates. This avoids errors that manifest as backlash-like behavior. The other two translations and two rotations perpendicular to the axis of the lead screw, shown in Figure 2.9A, should be as compliant as possible to accommodate for imperfections in the lead screw's straightness and alignment. The four DOFs shown in Figure 2.9A are, therefore, the desired motions of the flexure system.

Step (2): Identify Freedom Space

The freedom space that contains these desired motions belongs to the 4 DOF column of the chart from Figure 2.2 and is Type 8. The freedom space with the desired motions is shown in Figure 2.9B. This freedom space consists of all rotational lines that exist within pencils that lie on parallel planes and are intersected at their center point by the perpendicular axis of the lead screw, which is shown as a dashed line on the left side of Figure 2.9B. Screws of every pitch value exist in these disks as shown on the right side of the freedom space in Figure 2.9B. All translations that are perpendicular to the dashed line also exist and are represented by the thick arrows shown in the left side of the figure.

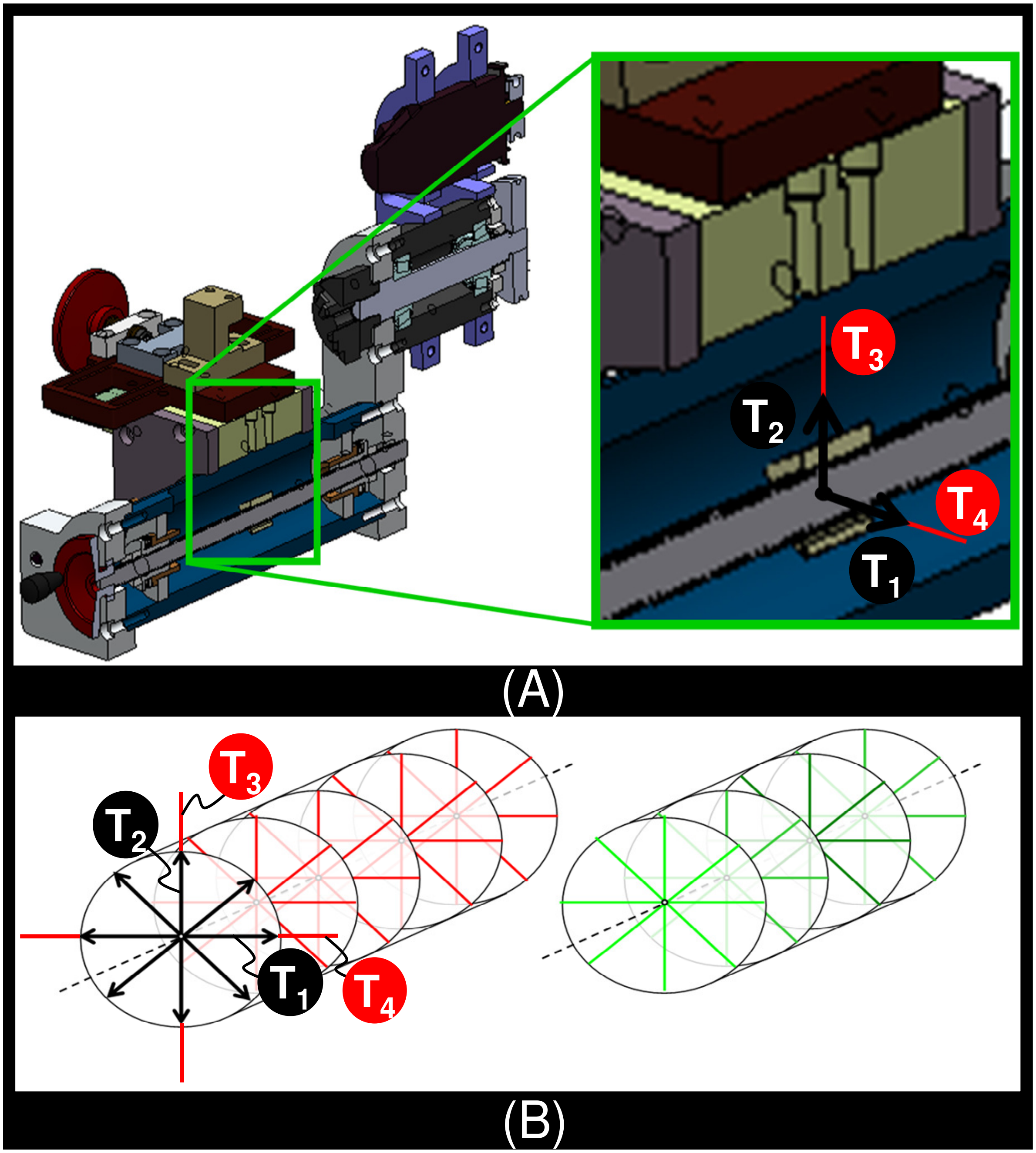

Figure 2.9: *The desired motions that should be imparted on the hex nut by the flexure system (A) and the freedom space of those desired motions (B).*

Step (3): Parallel or Serial?

Note that the freedom space of 4 DOF Type 8 from the chart of Figure 2.2 lies outside the parallel pyramid and has no complementary constraint space. In this case, therefore, the designer has no choice but to synthesize a serial flexure system as only a serial flexure system is capable of achieving the desired DOFs.

Step (4): Choose Intermediate Freedom Spaces

The designer must now identify which freedom spaces lie within the parallel pyramid and within the freedom space shown in Figure 2.9B. These freedom spaces will be the intermediate freedom spaces that the designer may use to synthesize the serial flexure system. They include Types 4 and 5 in the 3 DOF column, Types 1 through 9 in the 2 DOF column, and Types 1 through 3 in the 1 DOF column. Ideally, only two intermediate freedom spaces should be selected so that the serial flexure system will possess the fewest number of possible stages/conjugated elements. If the designer wishes to avoid underconstraint, the sum total of the number of DOFs from each intermediate freedom space should equal four as the system's freedom space consists of four DOFs. Many options would satisfy these requirements and would generate viable flexure concepts, but for the purposes of this case study, the Type 2 freedom space from the 2 DOF column will be selected twice because it is less complex than most of the other options. This intermediate freedom space and its complementary constraint space are shown in Figure 2.10A. The intermediate freedom space consists of every parallel rotation line on a plane and a translation that is perpendicular to that plane. The constraint space consists of every constraint line that lies on the same plane and every constraint line that is parallel to the rotation lines represented by the box of parallel lines. The planes of the two intermediate freedom spaces are oriented at 90 degree angles with respect to each other as shown in Figure 2.10B. These intermediate freedom spaces combined generate the freedom space of the system because the total number of independent twists from both intermediate freedom spaces is four, which is the number of DOFs of the system's freedom space. Note also that the system will not be underconstrained because the sum of the DOFs of both intermediate freedom spaces is also four.

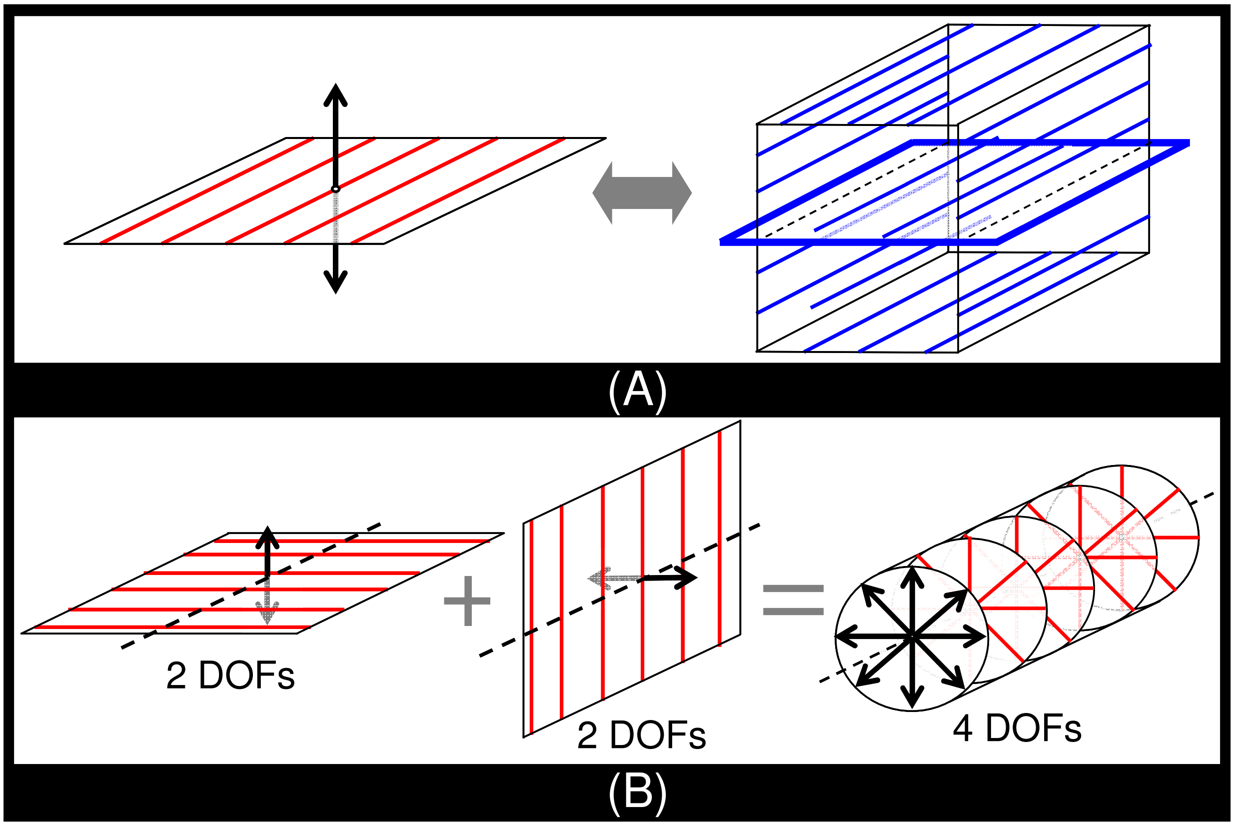

Figure 2.10: Freedom and constraint space of 2 DOF Type 2 (A). Two intermediate freedom spaces sum together to generate the freedom space of the system (B).

Step (5): Design Ground and Stage/s

The flexure system should be grounded to the carriage. The system should consist of two rigid stages because two intermediate freedom spaces were selected. The main stage that possesses the desired DOFs should clamp around the hex nut. Finally, the entire system should fit within the structural tube of the lathe shown in Figure 2.8 and again in Figure 2.11A. Some possible ground and stage designs that satisfy these conditions are shown in Figure 2.11B.

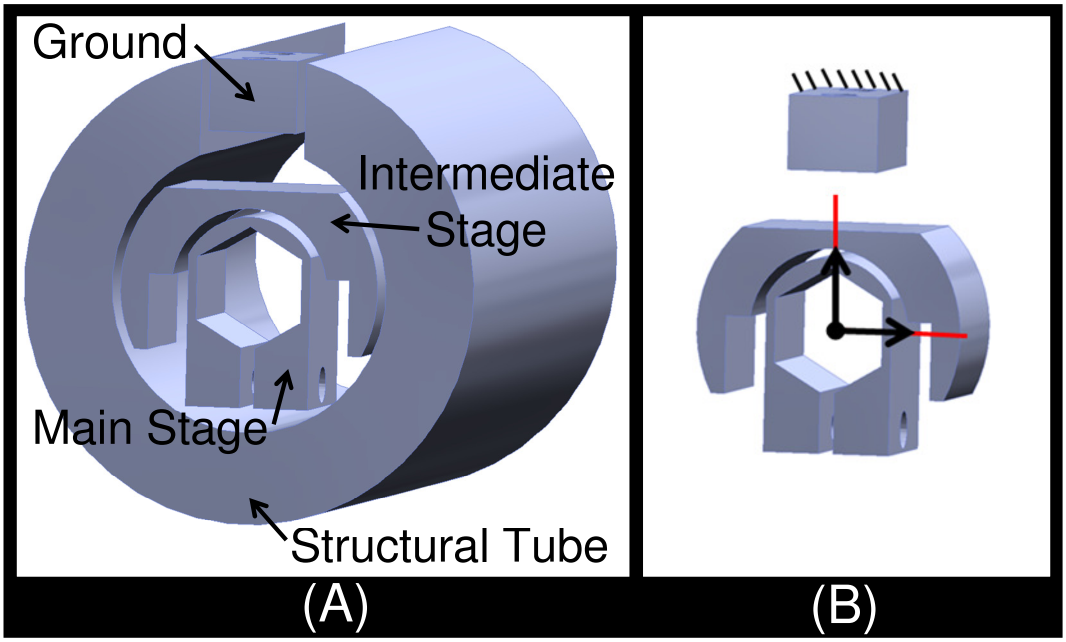

Figure 2.11: Geometric constraints shape the ground and stage designs (A). The ground and stages shown with the desired DOFs (B).

Step (6): Select Constraints from Constraint Space/s

The first intermediate freedom space from Figure 2.10B is shown superimposed on the system in Figure 2.12A. The complementary constraint space of this intermediate freedom space is shown in Figure 2.12B. Two flexure blades and four wire constraints are selected from the constraint space. These flexible elements shown in Figure 2.12C connect the main stage to the intermediate stage.

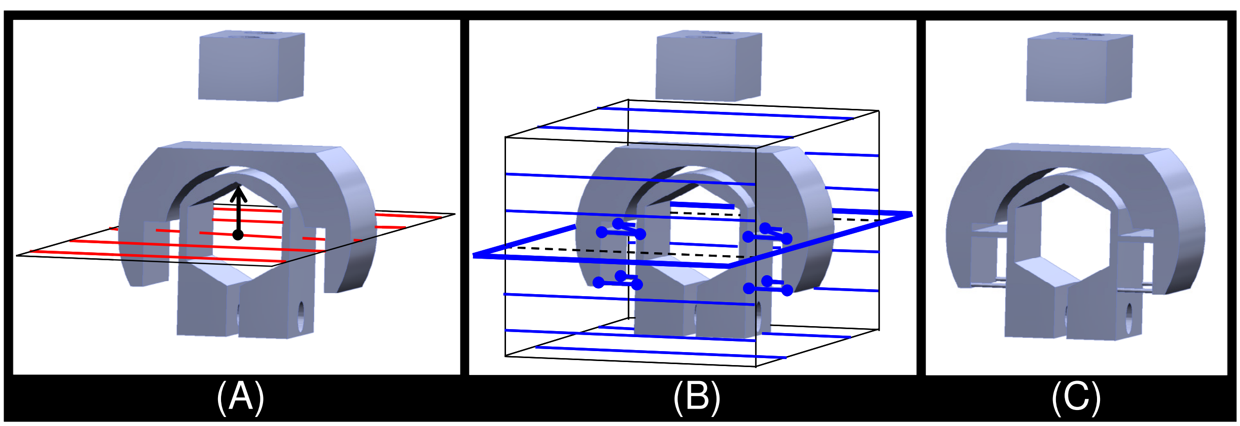

Figure 2.12: Orientation of the first intermediate freedom space (A) and its complementary constraint space (B). Flexible constraints selected from the constraint space connect the main stage to the intermediate stage (C).

The second intermediate freedom space from Figure 2.10B is shown superimposed on the system in Figure 2.13A. The complementary constraint space of this intermediate freedom space is shown in Figure 2.13B. A flexure blade and four wire constraints are selected from the constraint space. These flexible elements, shown in Figure 2.13C, connect the intermediate stage to the ground.

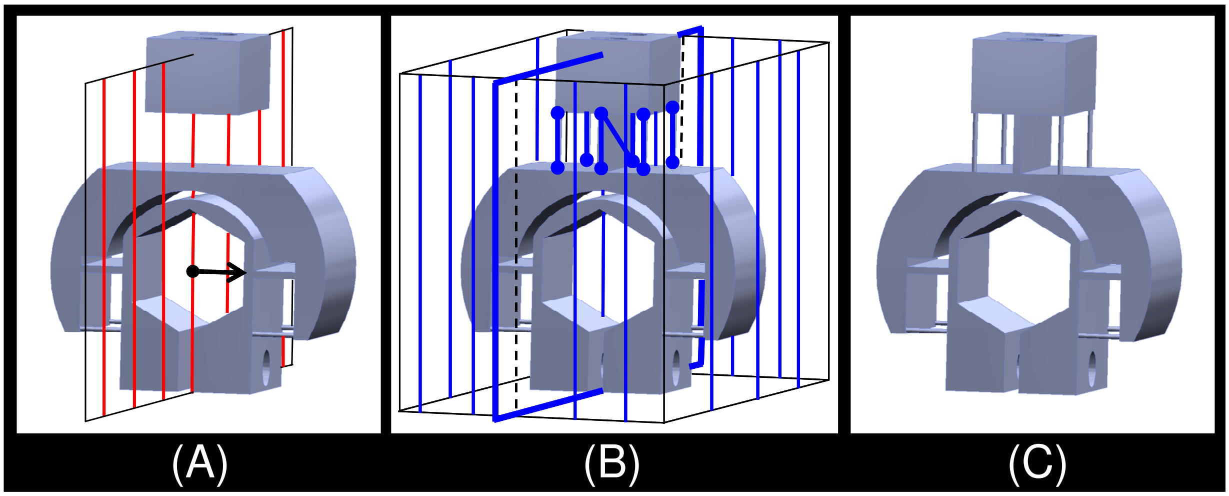

Figure 2.13: *Orientation of the second intermediate freedom space (A) and its complementary constraint space (B). Flexible constraints selected from the constraint space connect the intermediate stage to the ground (C).*

The final lead screw flexure shown in Figure 2.14 may be fabricated by cutting three planar pieces with a waterjet or wire EDM machine.

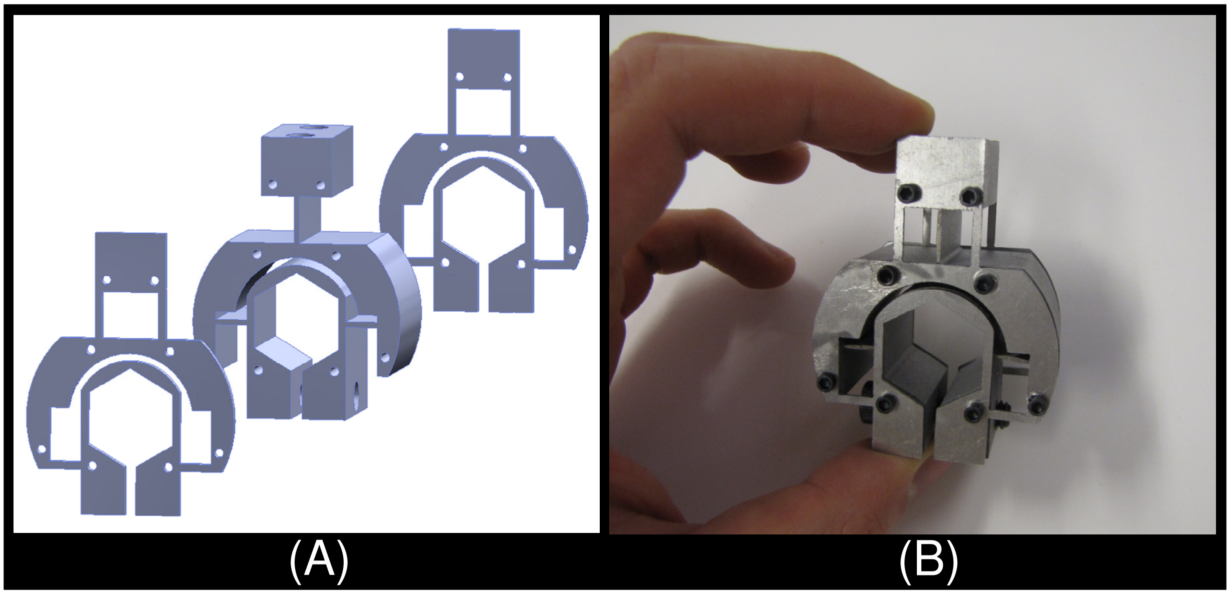

Figure 2.14: *The flexure consists of three planar pieces (A). The final lead screw flexure (B)*

Using FEA, images of the four desired DOFs are shown in Figure 2.15. These motions are orders of magnitude more compliant than the translation and rotation about the lead screw's axis. Note also, if the ground and main stage are held fixed, the intermediate stage will be fully constrained because the system is not underconstrained.

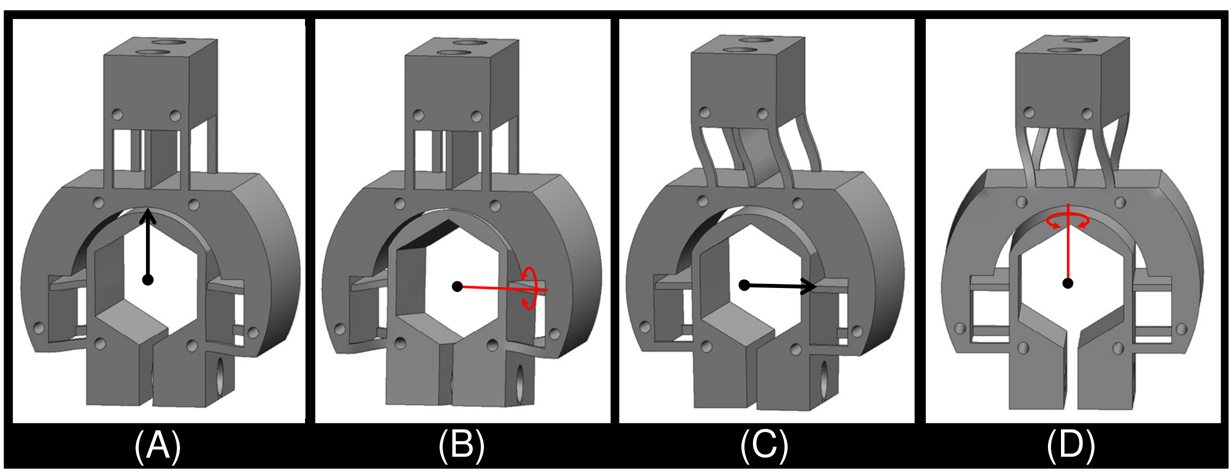

Figure 2.15: *The four DOFs of the lead screw flexure (A), (B), (C), and (D)*

CHAPTER 3:
Spaces of the FACT Chart

This chapter describes the geometry of every freedom and constraint space pair or type that belongs to the comprehensive FACT chart shown in Figure 2.1. The derivation of these spaces is found in Appendix A. Freedom spaces are depicted with (i) red lines that represent rotation lines, (ii) thick black arrows that represent translations, and (iii) green lines that represent screws. The darker the shade of green, the larger the absolute value of the pitch of the screw depicted. This information is summarized in Table 3.1.

Table 3.1: *Freedom space color code for various twist types*

Name	Pitch	Line Color
Rotation	$p = 0$	
Translation	$p = \infty$	
Screw	$p \neq 0 \neq \infty$	

Constraint spaces are depicted with *(i)* blue lines that represent pure forces or constraint lines, *(ii)* thick black lines with circular arrows that represent pure torques, and *(iii)* orange lines that represent wrenches with coupled forces and torques. The darker the shade of orange, the larger the absolute value of the q-value of the wrench depicted. This information is summarized in Table 3.2.

Table 3.2: *Constraint space color code for various wrench types*

Name	q-value	Line Color
Constraint Line	$q = 0$	
Pure Torque	$q = \infty$	
Wrench	$q \neq 0 \neq \infty$	

3.1 Types within the 0 DOF Column

This section describes the freedom and constraint space type that possesses zero DOFs

3.1.1 0 DOF Type 1

The freedom space of this type is shown in Figure 3.1. It consists of no rotations, translations, or screws and is thus depicted as an empty space.

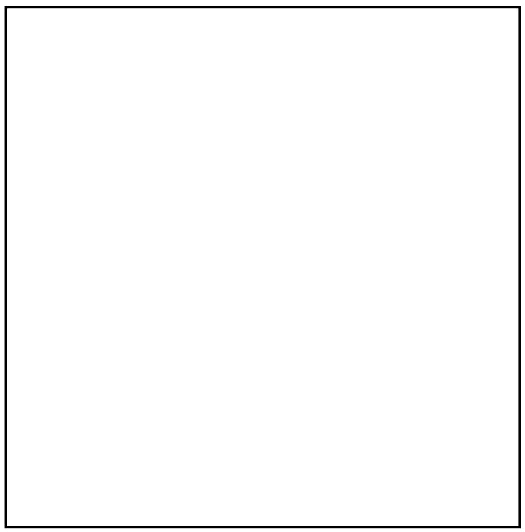

Figure 3.1: *Freedom space of 0 DOF Type 1*

The constraint space of this type is shown in Figure 3.2. It consists of every constraint line, pure torque, and wrench of every q-value that exists in all space. The constraint lines are therefore depicted as a space that is filled with blue lines. The pure torque lines are depicted as a sphere of black lines with circular arrows that point in all directions. The reason they are depicted in this way is that pure torques are directional only; they have no location. The wrench lines are depicted as a space filled with every shade of orange to represent a space filled with wrenches of every q-value.

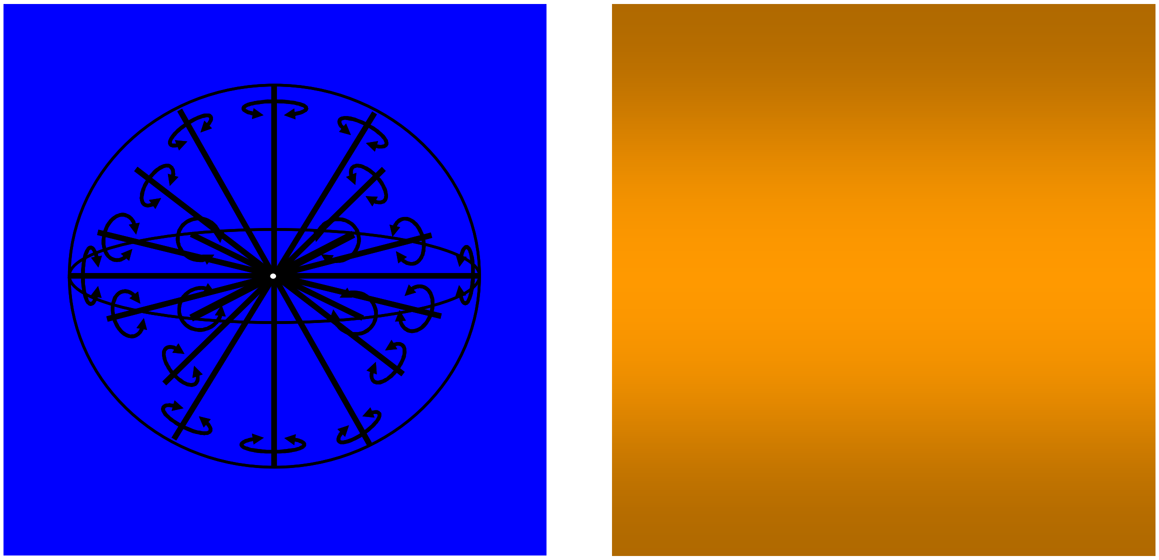

Figure 3.2: *Constraint space of 0 DOF Type 1*

3.2 Types within the 1 DOF Column

This section describes the three freedom and constraint space types that possess one DOF.

3.2.1 1 DOF Type 1

The freedom space of this type is shown in Figure 3.3. It consists of a single rotation line.

Figure 3.3: *Freedom space of 1 DOF Type 1*

The constraint space of this type is shown in Figure 3.4. The three vertical dashed lines in the figure are all collinear with the rotation line of Figure 3.3. The constraint space consists of every constraint line that lies on every plane that intersects the vertical dashed line shown on the left side of the figure. The constraint space also consists of every pure torque that points in a direction that is perpendicular to the same dashed line.

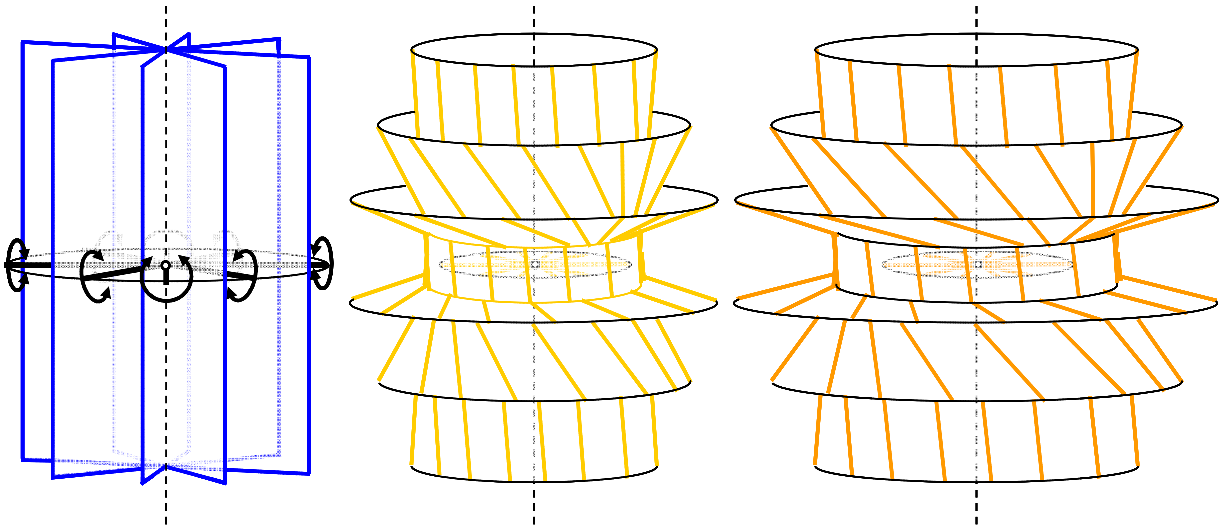

Figure 3.4: *Constraint space of 1 DOF Type 1*

The constraint space also consists of every wrench that satisfies Equation 1.8 where p equals zero, q is the q-value of the wrench and d and φ are parameters defined in Figure 3.5.

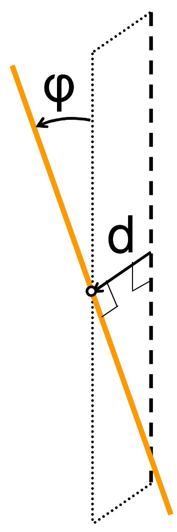

Figure 3.5: *Parameters that define the wrenches within the constraint space of 1 DOF Type 1*

The space shown in the middle of Figure 3.4 represents a set of wrenches that share the same q-value. This set contains a disk of wrenches that intersect and are perpendicular to the dashed vertical line. The set also contains wrenches that lie on the surfaces of circular hyperboloids with central circular cross-sections that share the same plane as the disk and have radii equal to *d*. These circular hyperboloids are formed when the orange line in Figure 3.5 is rotated about the dashed vertical line. The surfaces of the hyperboloids with larger and larger central circular cross-sections become more and more cylindrical. The reason for this fact is that as *d* in Equation 1.8 increases, φ must decrease for the q-value to remain the same.

The set in the middle of Figure 3.4 is not the only set with that same q-value. Imagine the same set displaced upwards along the vertical line. That set also exists within the constraint space of the system and all of its wrenches possess the same q-value as the set shown in the middle of Figure 3.4. In fact that same set exists within the constraint space anywhere along the vertical line.

Now consider the space on the far right of Figure 3.4. This space also consists of a set of wrenches that belongs to the system's constraint space, but its wrenches possess a q-value that is different than the set shown in the middle of the figure. Hence its wrenches are depicted with a different shade of orange. This set also consists of wrenches that lie within a disk and on the surface of circular hyperboloids that become more cylindrical the larger their central circular cross-sectional radius becomes. There is a similar set for every value of *q* according to Equation 1.8.

3.2.2 1 DOF Type 2

The freedom space of this type is shown in Figure 3.6. It consists of a single screw with a pitch of *p*.

Figure 3.6: *Freedom space of 1 DOF Type 2*

The constraint space of this type is shown in Figure 3.7. The three dashed vertical lines in the figure are all collinear with the line of the screw from Figure 3.6. The constraint space consists of every pure torque that is perpendicular to this line.

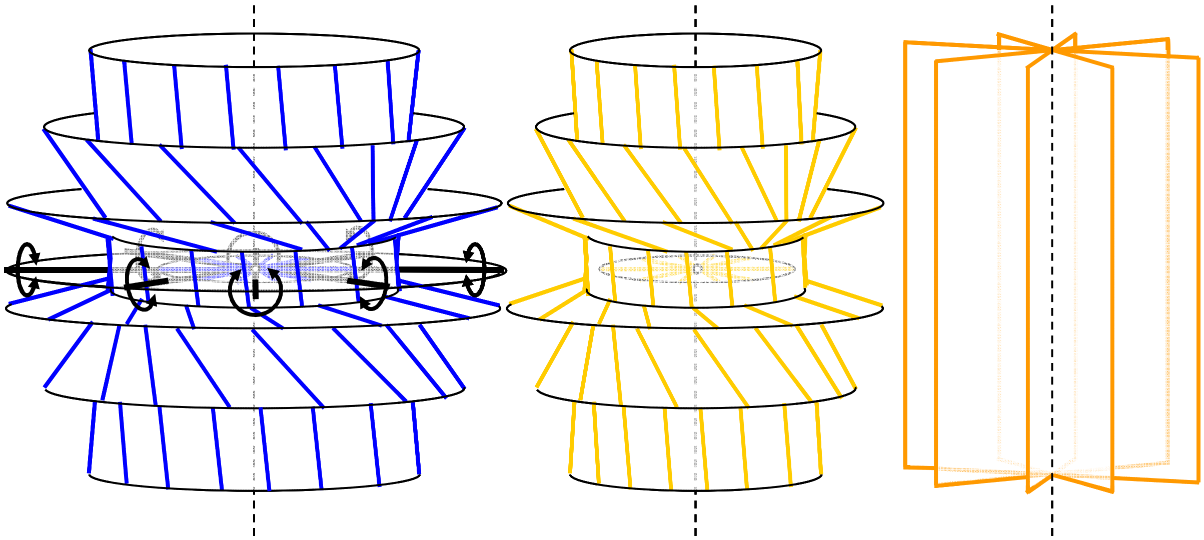

Figure 3.7: *Constraint space of 1 DOF Type 2*

The constraint space also consists of every constraint line that satisfies Equation 1.8 where p is the pitch of the freedom space's screw, q equals zero, and d and φ are parameters defined in Figure 3.8. These constraint lines are depicted in the same way as the wrench lines were depicted in the constraint space of Figure 3.4 because they too are governed by a similar equation. The constraint lines also lie within a disk and on the surface of circular hyperboloids that become more cylindrical the larger their central circular cross-sectional radius becomes. Not only does the constraint space contain the constraint lines depicted on the left side of Figure 3.7, it also contains many other constraint lines that belong to other spaces that look identical to the space shown in the figure but are displaced vertically along the dashed line.

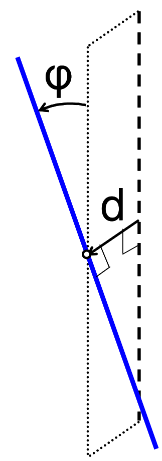

Figure 3.8: *Parameters that define the constraint lines within the constraint space of 1 DOF Type 1*

The constraint space also consists of every wrench line that satisfies Equation 1.8 where p is the pitch of the freedom space's screw, q is the q-value of the wrench, and d and φ are parameters defined in Figure 3.5. These wrench lines are shown in the middle of Figure 3.7 and are depicted in the same way as the wrench lines in Figure 3.4. There is also a set of wrenches that lie on every plane that intersects the vertical line shown on the right of Figure 3.7. The q-value of these wrenches is equal to the negative of the pitch value of the screw of the freedom space.

3.2.3 1 DOF Type 3

The freedom space of this type is shown in Figure 3.9. It consists of a single translation.

Figure 3.9: *Freedom space of 1 DOF Type 3*

The constraint space of this type is shown in Figure 3.10. It consists of every constraint line that lies on any plane that is perpendicular to the direction of the translation. These constraint lines are depicted as blue planes on the left side of Figure 3.10. The constraint space also consists of all pure torques that point in all directions. These torques are depicted as a sphere of pure torque lines on the left side of Figure 3.10. Finally, the system's constraint space consists of all wrench lines that lies on any plane that is perpendicular to the direction of the translation. There are an infinite number of wrenches of the same q-value on every plane. Furthermore, wrenches of all q-values lie on every plane. These wrench lines are depicted as orange shaded planes on the right side of Figure 3.10.

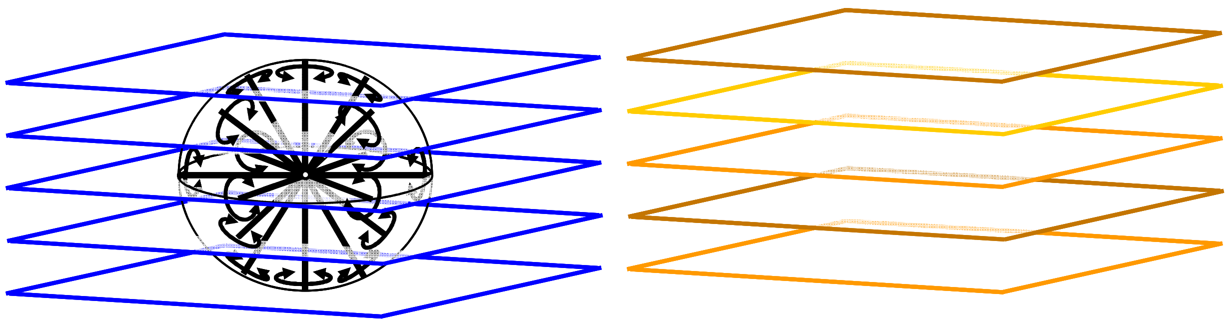

Figure 3.10: *Constraint space of 1 DOF Type 3*

3.3 Types within the 2 DOF Column

This section describes the ten freedom and constraint space types that possess two DOFs.

3.3.1 2 DOF Type 1

The freedom space of this type is shown in Figure 3.11. It consists of all coplanar rotation lines that intersect a single point. These rotation lines are represented by a red disk.

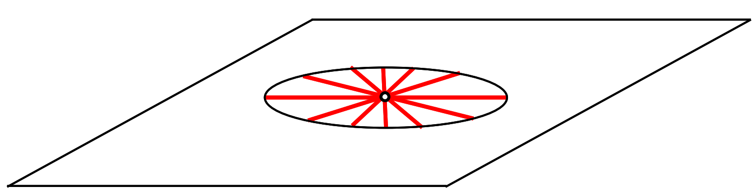

Figure 3.11: *Freedom space of 2 DOF Type 1*

The constraint space of this type is shown in Figure 3.12. It consists of every constraint line that lies on the plane of the disk of rotation lines. These constraint lines are represented by the blue plane in Figure 3.12. The constraint space also consists of every constraint line that intersects the same point at the center of the disk of rotation lines. These constraint lines are represented by the sphere of blue lines shown in Figure 3.12. The constraint space also consists of a pure torque line also shown in Figure 3.12 that points in a direction perpendicular to the plane of the disk of rotation lines.

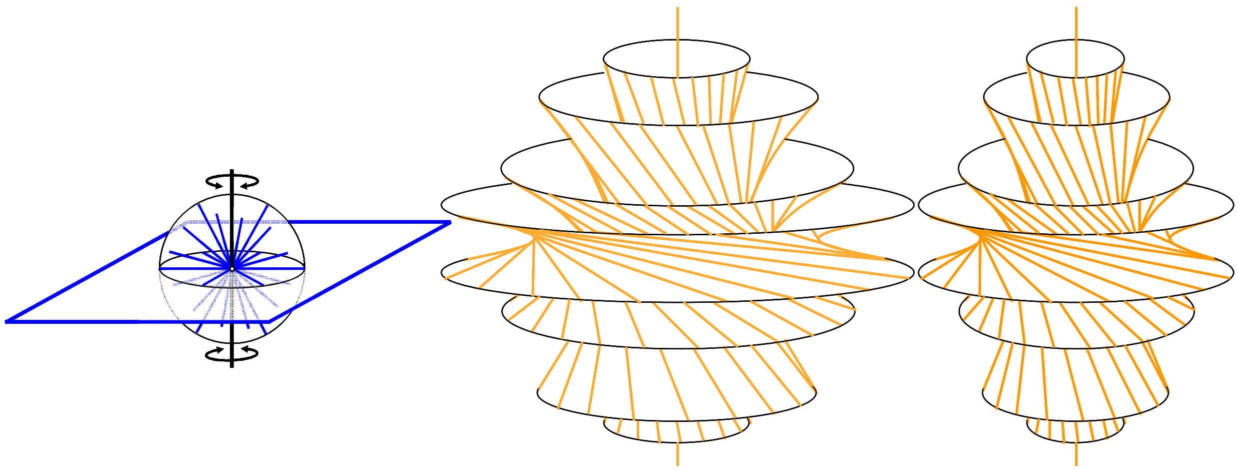

Figure 3.12: *Constraint space of 2 DOF Type 1*

The constraint space also consists of every wrench line that satisfies Equation 1.8 where p equals zero, q is the q-value of the wrench, and d and φ are parameters defined in Figure 3.13. The plane of Figure 3.13 is coplanar with the plane of Figure 3.11. The center point of the disk of Figure 3.11 corresponds with the intersection point of the vertical dashed line and the plane shown in Figure 3.13. A set of wrenches exists for every q-value. Each of these sets consists of a vertical wrench that is collinear with the vertical dashed line shown in Figure 3.13 and all the wrenches that lie one the surfaces of nested circular hyperboloids like the two sets shown on the right side of Figure 3.12. Note that all the wrenches within each set shown in Figure 3.12 are colored with the same shade of orange but that the two different sets are colored with a different shade of orange. This difference in shading indicates that each set consists of wrenches that possess a different q-value. The circular hyperboloids within each set are generated as the orange wrench shown in Figure 3.13 is revolved about the vertical dashed line. Note that as d increases, φ must decrease in order to keep the q-value of Equation 1.8 from changing. Note also that the wrenches from the sets shown in Figure 3.12 must possess negative q-values since the φ values defined in Figure 3.13 are shown as being larger than 90 degrees but smaller than 180 degrees.

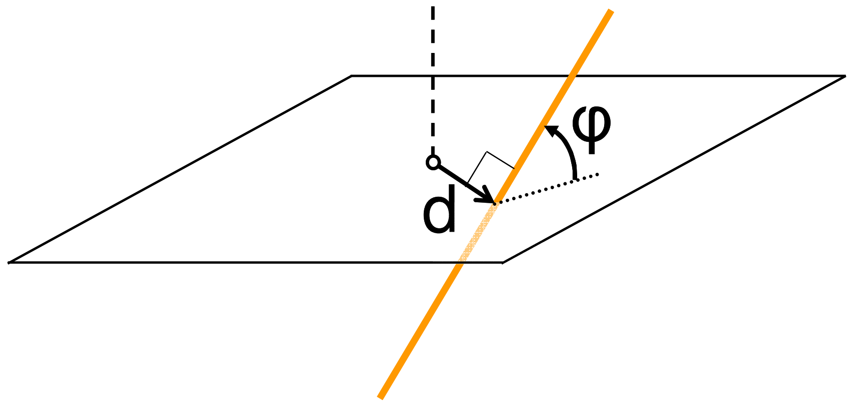

Figure 3.13: *Parameters that define the wrench lines within the constraint space of 2 DOF Type 1*

3.3.2 2 DOF Type 2

The freedom space of this type is shown in Figure 3.14. It consists of every parallel rotation line on a plane and a translation perpendicular to that plane.

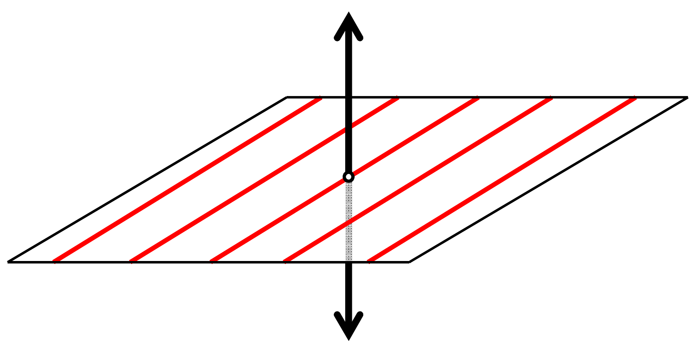

Figure 3.14: *Freedom space of 2 DOF Type 2*

The constraint space of this type is shown in Figure 3.15. It consists of every constraint line that lies on the plane of the freedom space and every constraint line that is parallel to the rotation lines of the freedom space. These constraint lines are represented by a blue plane and a box of parallel blue lines as shown in Figure 3.15. The constraint space also consists of every pure torque line that is perpendicular to the parallel rotation lines of the freedom space. These pure torque lines are represented by the disk of black lines with circular arrows.

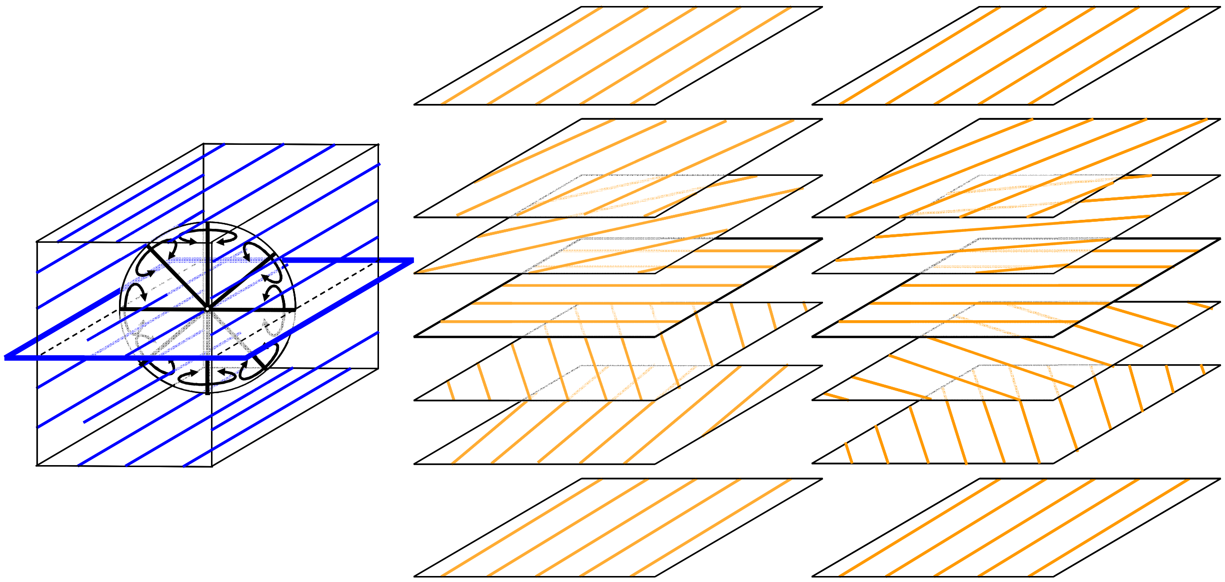

Figure 3.15: *Constraint space of 2 DOF Type 2*

The constraint space also consists of every wrench line that satisfies Equation 1.8 where p equals zero, q is the q-value of the wrench, and d and φ are parameters defined in Figure 3.16. The plane outlined with a thick black border shown in Figure 3.16 is coplanar with the plane of the freedom space from Figure 3.14. The dashed black lines shown in Figure 3.16 are parallel to the rotation lines from Figure 3.14. A set of wrenches exists for every q-value. Each of these sets consists of parallel planes of parallel wrench lines of the same q-value as shown in Figure 3.15. The wrenches on the plane of the freedom space are perpendicular to the rotation lines of the freedom space. The parallel wrenches on each successive plane rotate more and more until the wrenches on the planes infinitely far from the plane of the freedom space are parallel to the rotation lines of the freedom space. The amount the parallel wrench lines rotate on planes a distance d from the plane of the freedom space may be determined using Equation 1.8. Note from Figure 3.16 as d increases, φ must decrease for the q-value to remain the same. Note also that the wrenches from the sets shown in Figure 3.15 must possess negative q-values since the φ values defined in Figure 3.16 are shown as being larger than 90 degrees but smaller than 180 degrees.

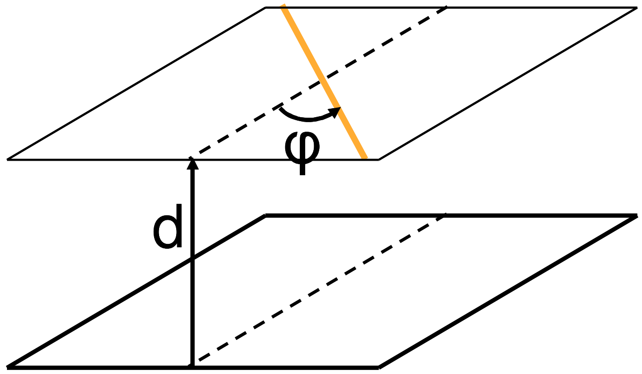

Figure 3.16: *Parameters that define the wrench lines within the constraint space of 2 DOF Type 2*

3.3.3 2 DOF Type 3

The freedom space of this type is shown in Figure 3.17. It consists of an infinite number of screws and two rotation lines that lie on the surface of a cylindroid. The orthogonal principle generators of the cylindroid must consist of a screw with a positive pitch value and a screw with a negative pitch value. Although the two rotation lines shown in Figure 3.17 are depicted as the extreme generators of the cylindroid, the two skew rotation lines don't necessarily have to be extreme generators to belong to this freedom space. The different shades of green shown in Figure 3.17 demonstrate how the pitch values of the screws vary continuously along the cylindroid's surface.

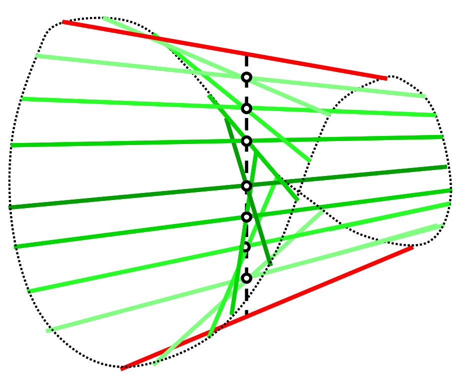

Figure 3.17: *Freedom space of 2 DOF Type 3*

The constraint space of this type is shown in Figure 3.18. It consists of all the constraint lines that intersect both rotation lines from the freedom space. These rotation lines are synonymous to the two skew dashed lines in Figure 3.18. The constraint lines are represented by an infinite number of blue disks that *(i)* lie on planes that intersect one of the dashed skew lines and *(ii)* possess center points that are intersected by the other dashed skew line. The constraint space also consists of a pure torque line that points in the direction of the cylindroid's axis. This axis is perpendicular to every twist line in the freedom space. The constraint space also consists of wrenches of all q-values that are collinear with this axis. Other sets of wrenches exist that are too complex to visually represent. As such, the system's wrenches are depicted using a "W" as shown in Figure 3.18. Every wrench may be mathematically determined using Equation 1.9.

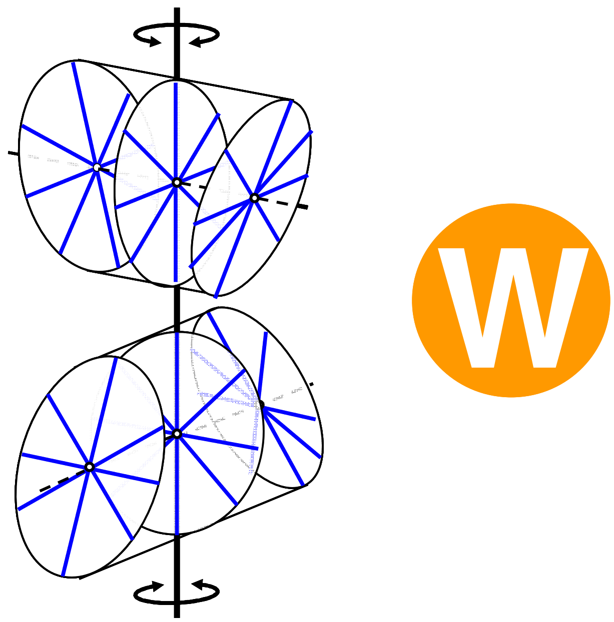

Figure 3.18: *Constraint space of 2 DOF Type 3*

3.3.4 2 DOF Type 4

The freedom space of this type is shown in Figure 3.19. It consists of every coplanar screw of the same pitch that intersects a common point. These screws are represented by a green disk in Figure 3.19. Notice that only one shade of green is used for the screws.

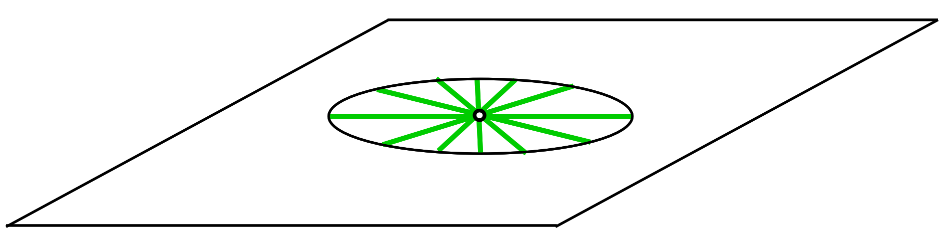

Figure 3.19: *Freedom space of 2 DOF Type 4*

The constraint space of this type is shown in Figure 3.20. It consists of a pure torque line that points in the direction perpendicular to the plane of the disk of screws. The constraint space also consists of a constraint line that is perpendicular to the disk of screws and intersects the disk's central point as well as all constraint lines that lie on the surfaces of nested circular hyperboloids like those shown in Figure 3.20.

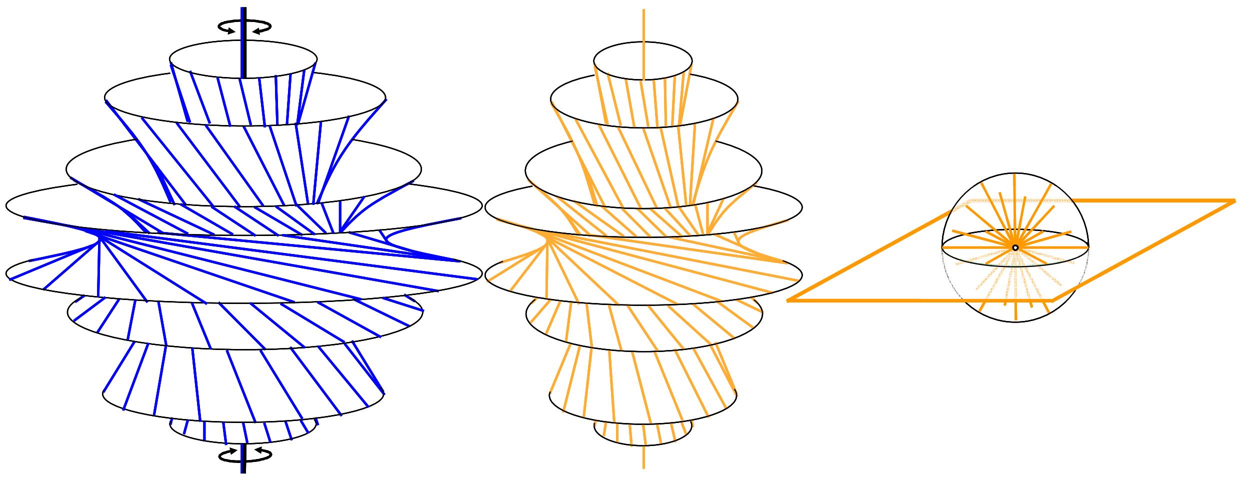

Figure 3.20: *Constraint space of 2 DOF Type 4*

The constraint lines that lie on the surfaces of these nested circular hyperboloids satisfy Equation 1.8 where p is the pitch value of the screws in the disk, q equals zero, and d and φ are parameters defined in Figure 3.21. The plane of Figure 3.21 is coplanar with the plane of Figure 3.19. The center point of the disk of Figure 3.19 corresponds with the intersection point of the vertical dashed line and the plane shown in Figure 3.21. The blue circular hyperboloids shown on the left side of Figure 3.20 are generated as the blue constraint line shown in Figure 3.21 is revolved about the vertical dashed line. Note that as d increases, φ must decrease in order to keep the pitch of Equation 1.8 from changing. Note also that the pitch value of the screws in the freedom space must be negative according to the configuration of the constraint lines shown in Figure 3.20 since the constraint lines' φ values defined in Figure 3.21 are shown as being larger than 90 degrees but smaller than 180 degrees.

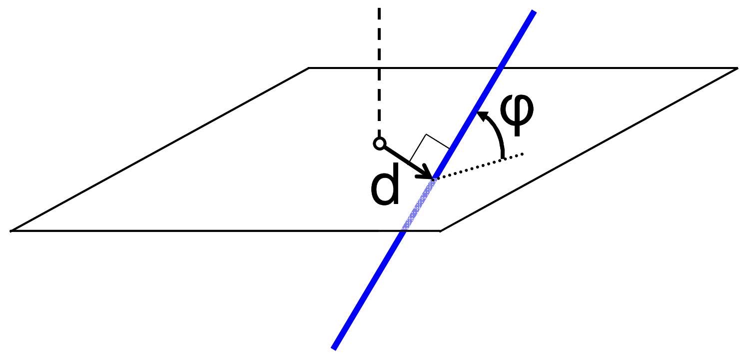

Figure 3.21: *Parameters that define the constraint lines within the constraint space of 2 DOF Type 4*

The constraint space also consists of all wrenches that satisfy Equation 1.8 where p is the pitch value of the screws in the disk, q is the q-value of the wrenches, and d and φ are parameters defined in Figure 3.13. A set of wrenches exists for every q-value. Each of these sets consists of a vertical wrench that is collinear with the vertical dashed line shown in Figure 3.13 and all the wrenches that lie one the surfaces of nested circular hyperboloids like the set shown in the middle of Figure 3.20. The circular hyperboloids within each set are generated as the orange wrench shown in Figure 3.13 is revolved about the vertical dashed line. Note that as d increases, φ must decrease in order to keep the pitch and q-value in Equation 1.8 from changing.

The constraint space also consists of a set of wrenches with a q-value equal to the negative of the pitch value of the screws in the disk of the freedom space. This set consists of every wrench that lies on the plane of the disk and every wrench that intersects the disk's center point. These wrenches are depicted on the right side of Figure 3.20 as an orange plane and sphere.

3.3.5 2 DOF Type 5

The freedom space of this type is shown in Figure 3.22. It consists of every coplanar parallel screw with a common pitch value and a translation perpendicular to the plane of the parallel screws.

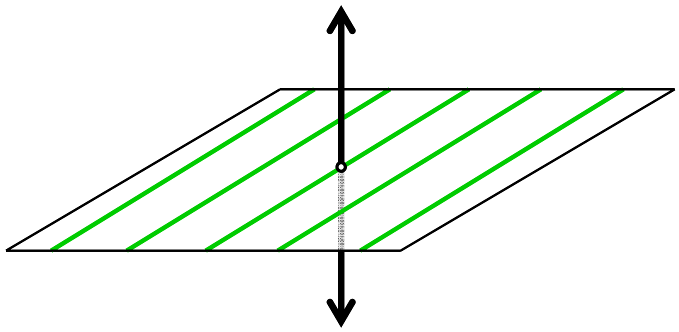

Figure 3.22: *Freedom space of 2 DOF Type 5.*

The constraint space of this type is shown in Figure 3.23. It consists of every pure torque line that is perpendicular to the parallel screw lines of the freedom space. These pure torque lines are represented by the disk of black lines with circular arrows shown on the left side of Figure 3.23.

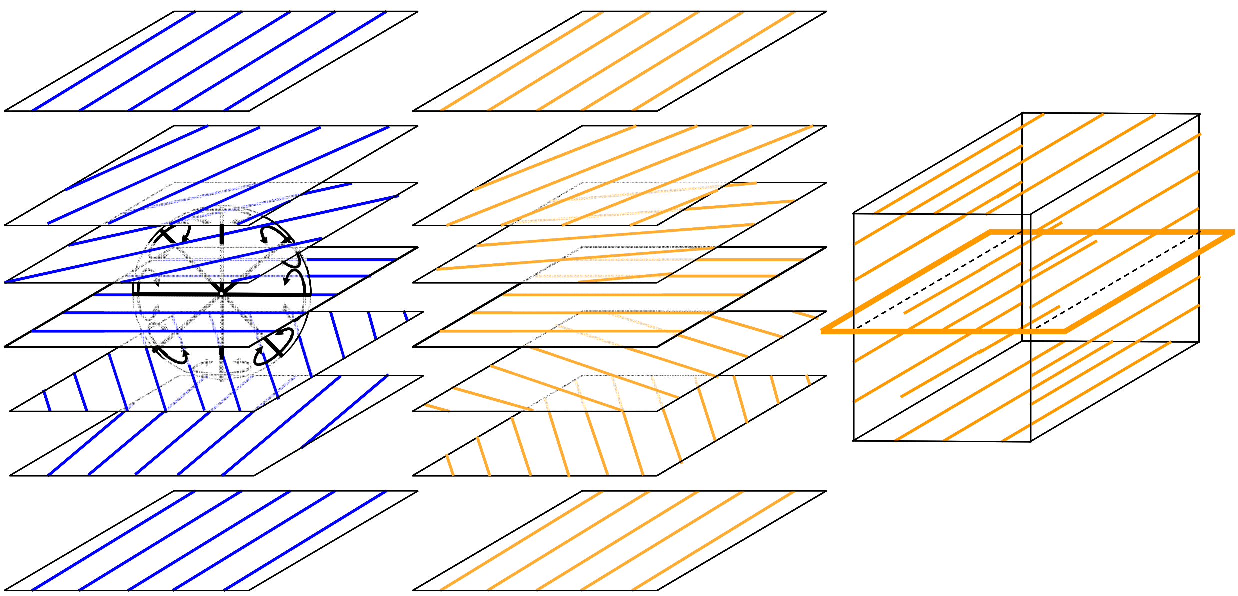

Figure 3.23: *Constraint space of 2 DOF Type 5.*

The constraint space also consists of every constraint line that satisfies Equation 1.8 where *p* is the pitch of the parallel screws, *q* equals zero, and *d* and *φ* are parameters defined in Figure 3.24. The plane outlined with a thick black border shown in Figure 3.24 is coplanar with the plane of the freedom space from Figure 3.22. The dashed black lines shown in Figure 3.24 are parallel to the screw lines from Figure 3.22. The constraint space, therefore, consists of parallel planes of parallel constraint lines as shown on the left side of Figure 3.23. The constraint lines on the plane of the freedom space are perpendicular to the screw lines of the freedom space. The parallel constraint lines on each successive plane rotate more and more until the constraint lines on the planes infinitely far from the plane of the freedom space are parallel to the screw lines of the freedom space. The amount the parallel constraint lines rotate on planes a distance *d* from the plane of the freedom space may be determined using Equation 1.8. Note from Figure 3.24 as *d* increases, φ must decrease for the pitch of the screws to remain the same. Note also that the pitch value of the screws in the freedom space must be negative according to the configuration of the constraint lines shown in Figure 3.23 since the constraint lines' φ values defined in Figure 3.24 are shown as being larger than 90 degrees but smaller than 180 degrees.

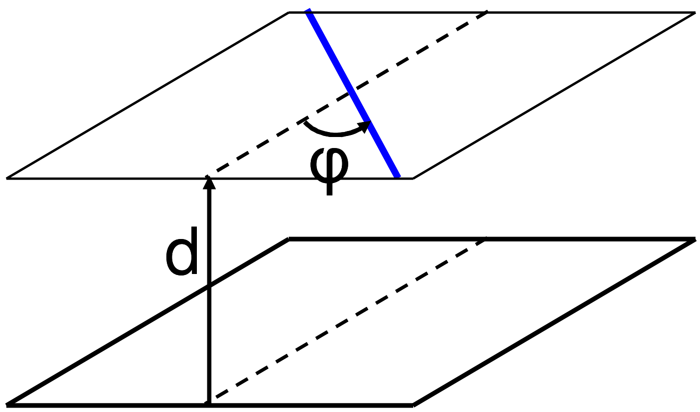

Figure 3.24: *Parameters that define the constraint lines within the constraint space of 2 DOF Type 5*

The constraint space also consists of all wrenches that satisfy Equation 1.8 where *p* is the pitch of the parallel screws, *q* is the q-value of the wrenches, and *d* and φ are parameters defined in Figure 3.16. A set of wrenches exists for every q-value. Each of these sets consists of parallel planes of parallel wrench lines of the same q-value as shown in the middle of Figure 3.23. The wrenches on the plane of the freedom space are perpendicular to the screw lines of the freedom space. The parallel wrenches on each successive plane rotate more and more until the wrenches on the planes infinitely far from the plane of the freedom space are parallel to the screw lines of the freedom space. The amount the parallel wrench lines rotate on planes a distance *d* from the plane of the freedom space may be determined using Equation 1.8. Note from Figure 3.16 as *d* increases, φ must decrease for the q-value to remain the same.

The constraint space also consists of a set of wrenches with a q-value equal to the negative of the pitch value of the parallel screws in the freedom space. This set consists of every wrench that lies on the plane of the parallel screws and every wrench that is parallel to these screws. These wrenches are depicted on the right side of Figure 3.23 as an orange plane and a box of orange lines.

3.3.6 2 DOF Type 6

The freedom space of this type is shown in Figure 3.25. It consists of a single rotation line and an infinite number of screws that lie on the surface of a cylindroid. The rotation line must be one of the cylindroid's two principle generators. The different shades of green shown in Figure 3.25 demonstrate how the pitch values of the screws vary continuously along the cylindroid's surface.

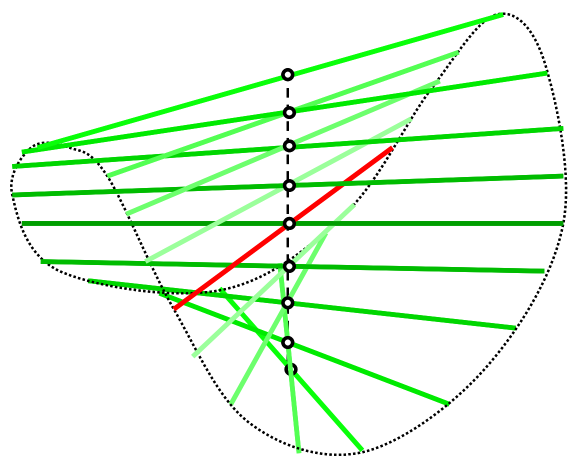

Figure 3.25: *Freedom space of 2 DOF Type 6.*

The constraint space of this type is shown in Figure 3.26. It consists of a pure torque line that points in the direction of the cylindroid's axis. This axis is perpendicular to every twist line in the freedom space. The constraint space also consists of every constraint line that is parallel to the single rotation line and lies on the plane of the cylindroid's principle generators. The constraint space also consists of every constraint line that lies within disks shown in Figure 3.26 that rotate as they translate along the axis of the freedom space's rotation line. The other principle generator, colored with the darkest shade of green in Figure 3.25, is collinear with the dashed line shown in Figure 3.26.

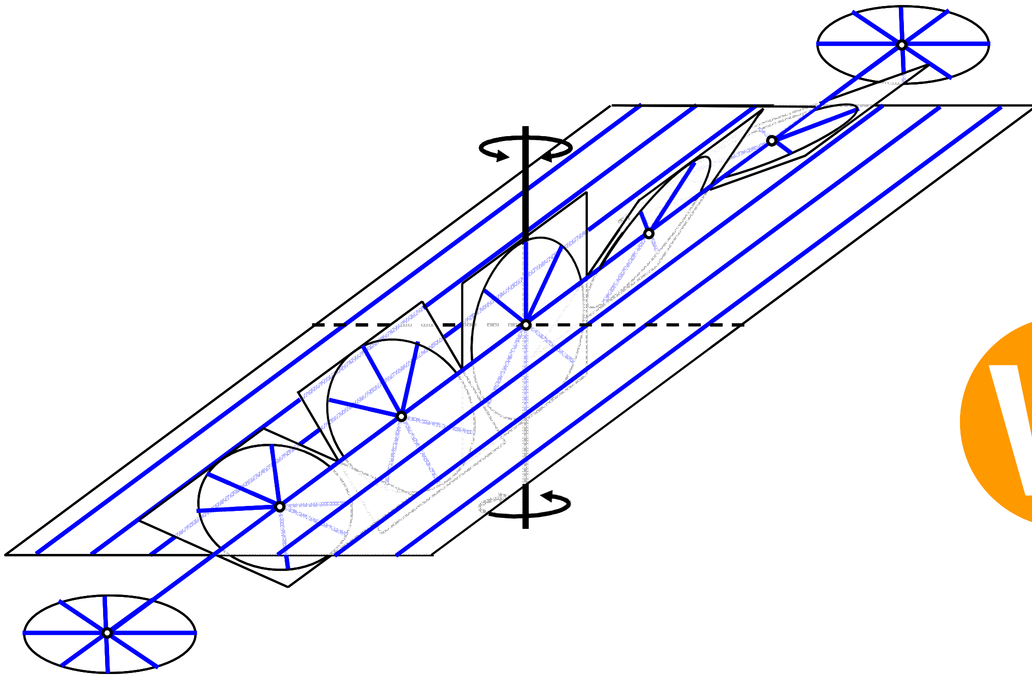

Figure 3.26: *Constraint space of 2 DOF Type 6.*

The constraint lines that lie within the disks satisfy Equation 1.8 where p is the pitch of the dark green principle generator shown in Figure 3.25, q equals zero, and d and φ are parameters defined in Figure 3.27. The screws within the freedom space of Figure 3.25 must have negative pitch values according to the orientations of the disks shown in Figure 3.26. Note from Figure 3.27, as d increases, φ must also increase to maintain the negative pitch value of the cylindroid's principle generator because φ is larger than 90 degrees but smaller than 180 degrees. Note also that the disks that are infinitely far from the dashed line are coplanar with the plane of parallel constraint lines.

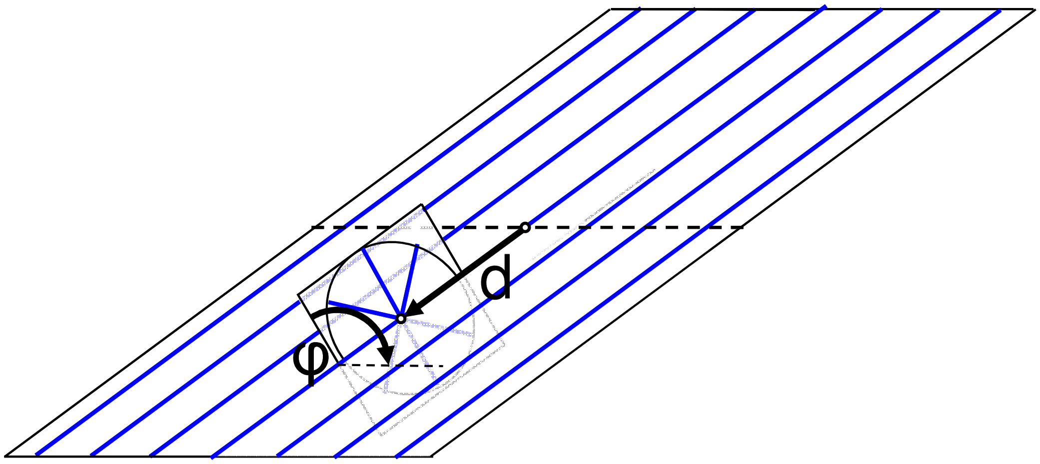

Figure 3.27: *Parameters that define the blue disks within the constraint space of 2 DOF Type 6*

The constraint space also consists of wrenches of all q-values that are too complex to visually represent. As such, the system's wrenches are depicted using a "W" as shown in Figure 3.26. Every wrench may be mathematically determined using Equation 1.9.

3.3.7 2 DOF Type 7

The freedom space of this type is shown in Figure 3.28. It consists of an infinite number of screws that lie on the surface of a cylindroid. The cylindroid's two principle generators must either both have positive pitch values or negative pitch values. The pitch values cannot be equivalent or equal to zero. The different shades of green shown in Figure 3.28 demonstrate how the pitch values of the screws vary continuously along the cylindroid's surface

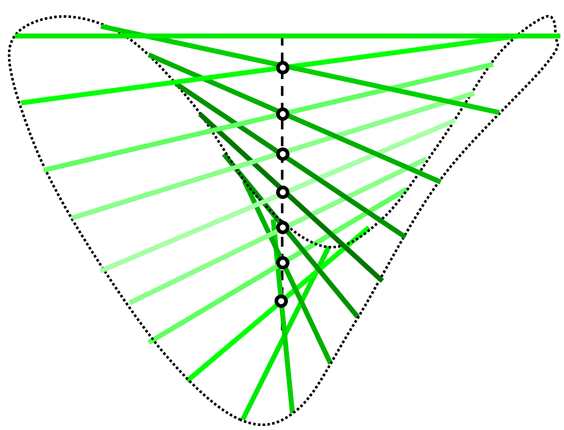

Figure 3.28: *Freedom space of 2 DOF Type 7*

The constraint space of this type is shown in Figure 3.29. It consists of a pure torque line that points in the direction of the cylindroid's axis. This axis, shown as a dashed line in Figure 3.28, is perpendicular to every screw in the freedom space. The constraint space also consists of a constraint line that is collinear with the cylindroid's axis and every constraint line that lies on the surfaces of nested elliptical hyperboloids as shown on the left side of Figure 3.29. The major and minor axes of the central elliptical cross-sections of the hyperboloids are collinear with the orthogonal principle generators of the freedom space shown in Figure 3.28.

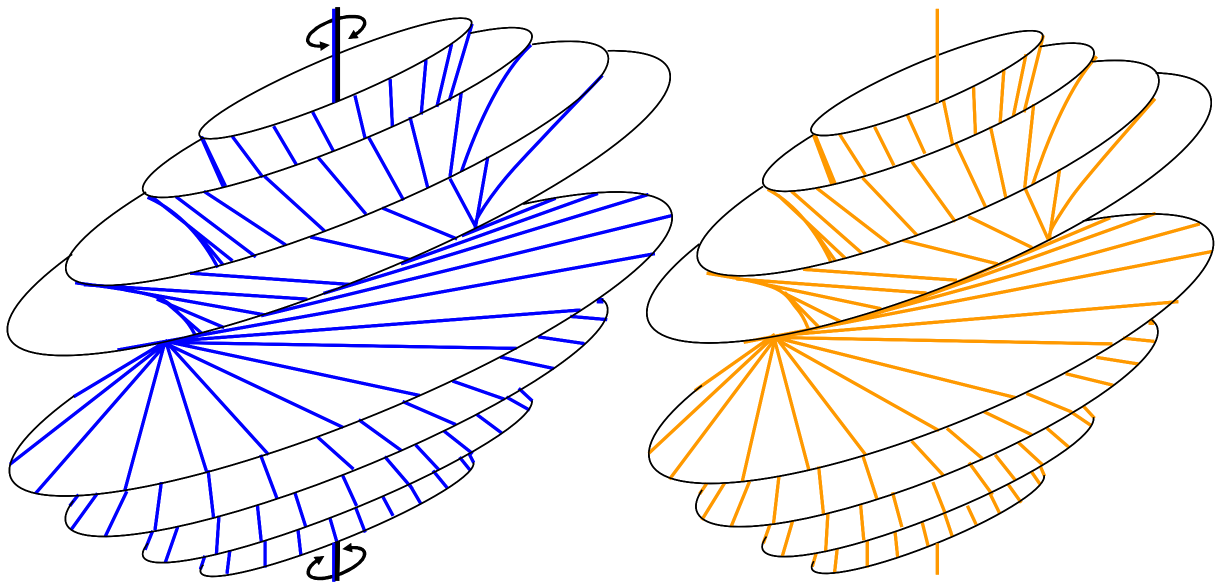

Figure 3.29: *Constraint space of 2 DOF Type 7*

The elliptical hyperboloids on which the constraint lines lie may be determined by

$$p_i + q_i = d_i \cdot \tan(\varphi_i), \tag{3.1}$$

for i equal to1 or 2 where p_1 is the pitch of the principle generator that is collinear with the major axis of the hyperboloids' central elliptical cross-section, p_2 is the pitch of the principle generator that is collinear with the major axis of the hyperboloids' central elliptical cross-section, q_1 and q_2 are equal to zero, and d_1, d_2, φ_1 and φ_2 are defined in Figure 3.30A. Note that two other constraint lines that intersect and are perpendicular to the minor and major axes of the ellipse would also exist on the other sides of the hyperboloid for negative d_1 and d_2 values. The elliptical hyperboloid is generated by linearly combining any three of these four constraint lines. Note that in Figure 3.29, φ_1 and φ_2 are both greater than 90 degrees but less than 180 degrees.

The pitch values of the screws in the freedom space must, therefore, be negative according to the orientation of the constraint lines shown in Figure 3.29.

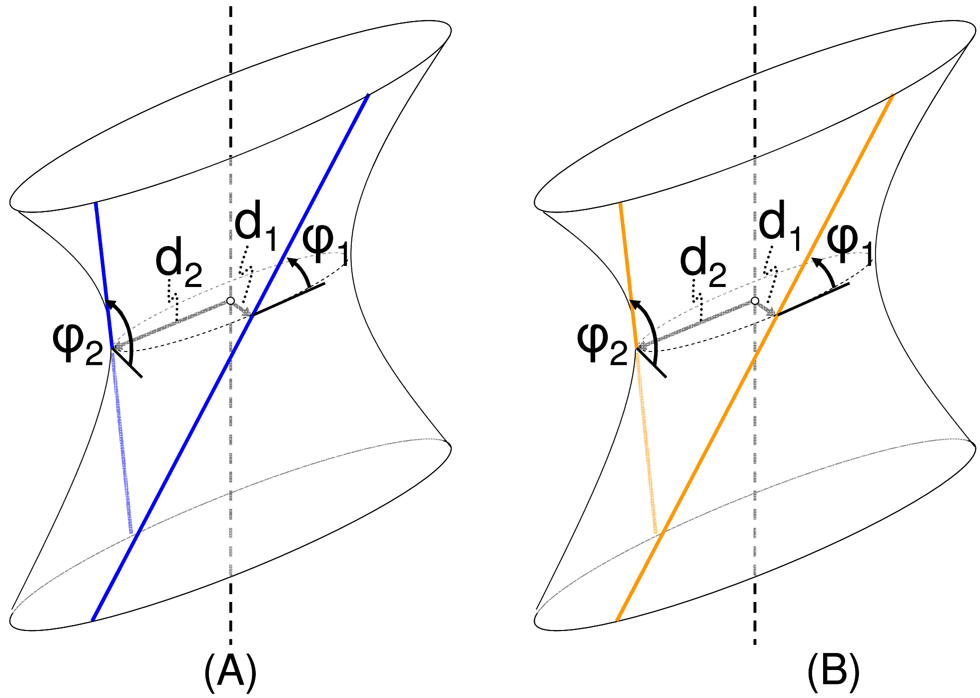

Figure 3.30: *Parameters that define the blue constraint lines (A) and orange wrench lines (B) within the constraint space of 2 DOF Type 7.*

The constraint space also consists of different sets of wrenches. A set of wrenches exists for every q-value. These sets consist of a wrench line that is collinear with the cylindroid's axis and every wrench line that lies on the surfaces of nested elliptical hyperboloids as shown on the right side of Figure 3.29. The elliptical hyperboloids on which the wrench lines lie may be determined by Equation (3.1) for i equal to1 or 2 where p_1 is the pitch of the principle generator that is collinear with the major axis of the hyperboloids' central elliptical cross-section, p_2 is the pitch of the principle generator that is collinear with the major axis of the hyperboloids' central elliptical cross-section, q_1 is the q-value of the wrench intersected by the minor axis, q_2 is the q-value of the wrench intersected by the major axis , and d_1, d_2, φ_1 and φ_2 are defined on the right side of Figure 3.30B.

3.3.8 2 DOF Type 8

The freedom space of this type is shown in Figure 3.31. It consists of a rotation line, all screws of every pitch value that are collinear with the rotation line, and a translation that points along the axis of the rotation line.

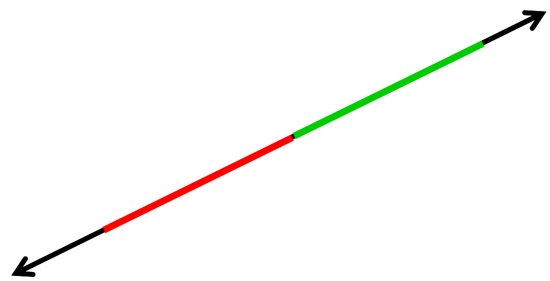

Figure 3.31: *Freedom space of 2 DOF Type 8*

The constraint space of this type is shown in Figure 3.32. The dashed lines in the figure are collinear with the twist lines of the freedom space. The constraint space consists of all pure torque lines that are perpendicular to the dashed line as depicted by the disk of back lines with circular arrows in Figure 3.32. The constraint space also consists of all constraint lines that exist within disks that lie on parallel planes. The dashed line intersects these disks through their central points at orthogonal angles. The constraint space also consists of wrench lines of every q-value that are collinear with every constraint line. These wrenches are shown on the right side of Figure 3.32 as disks that are colored different shades of orange.

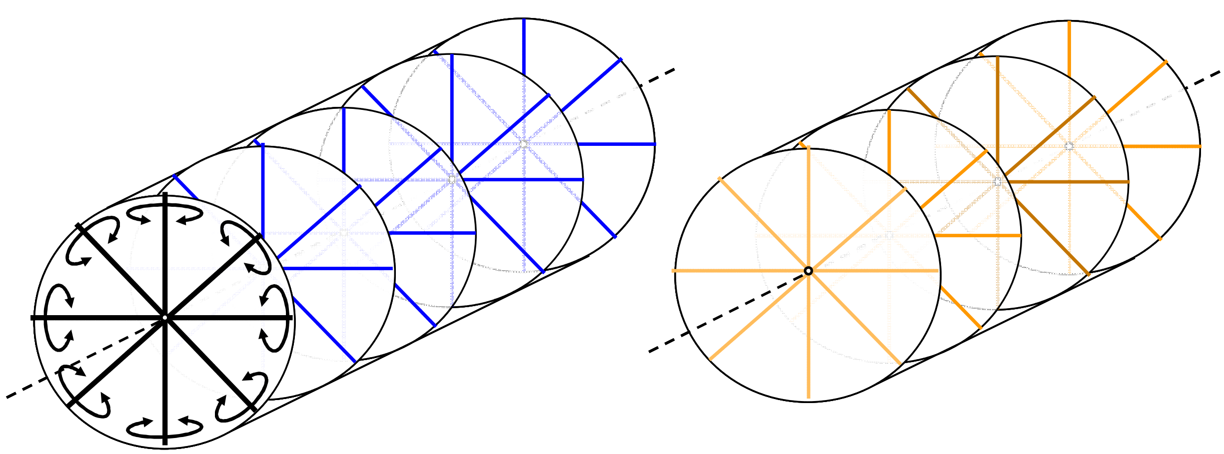

Figure 3.32: *Constraint space of 2 DOF Type 8*

3.3.9 2 DOF Type 9

The freedom space of this type is shown in Figure 3.33. It consists of a rotation line and coplanar screws that are parallel to the rotation line as shown in the figure. The freedom space also consists of a single translation that points in a direction that is not parallel or perpendicular to the rotation line. The plane of the parallel screws is orthogonal to the plane that is normal to the cross product of the direction of the rotation line and the translation. The screws on one side of the plane have pitch values that are negative and the screws on the other side of the plane have pitch values that are positive. The magnitude of a screw's pitch value increases linearly the farther the screw is from the rotation line according to

$$p = d \cdot \tan(\beta + 90^{\circ}), \tag{3.2}$$

where p is the pitch of the screw that is a distance d from the rotation line. The angle β is defined in Figure 3.33.

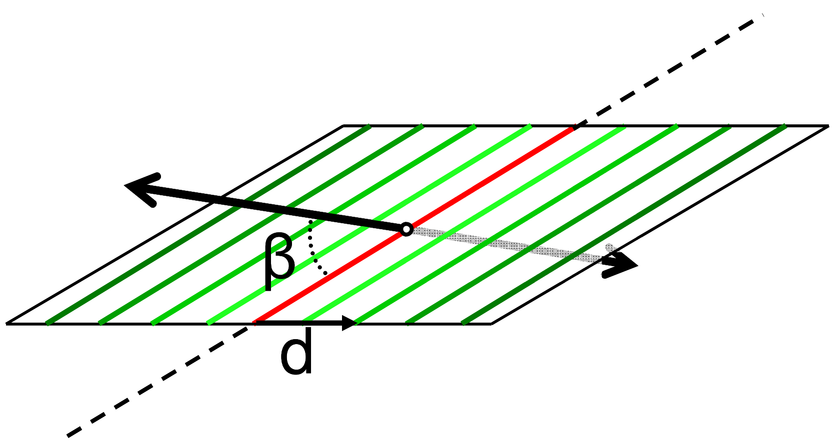

Figure 3.33: *Freedom space of 2 DOF Type 9*

The constraint space of this type is shown in Figure 3.34. The dashed lines in the figure are collinear with the rotation line of the freedom space. The constraint space consists of all pure torque lines that are perpendicular to the dashed line as depicted by the disk of back lines with circular arrows in Figure 3.34. The constraint space also consists of all constraint lines that exist within disks that lie on parallel planes that are normal to the direction of the translation of the freedom space. The dashed line intersects these disks through their central points. The constraint space also consists of wrench lines of various q-values and locations that lie on the planes of the blue constraint disks as shown on the right side of Figure 3.34. Every plane contains a wrench of every q-value.

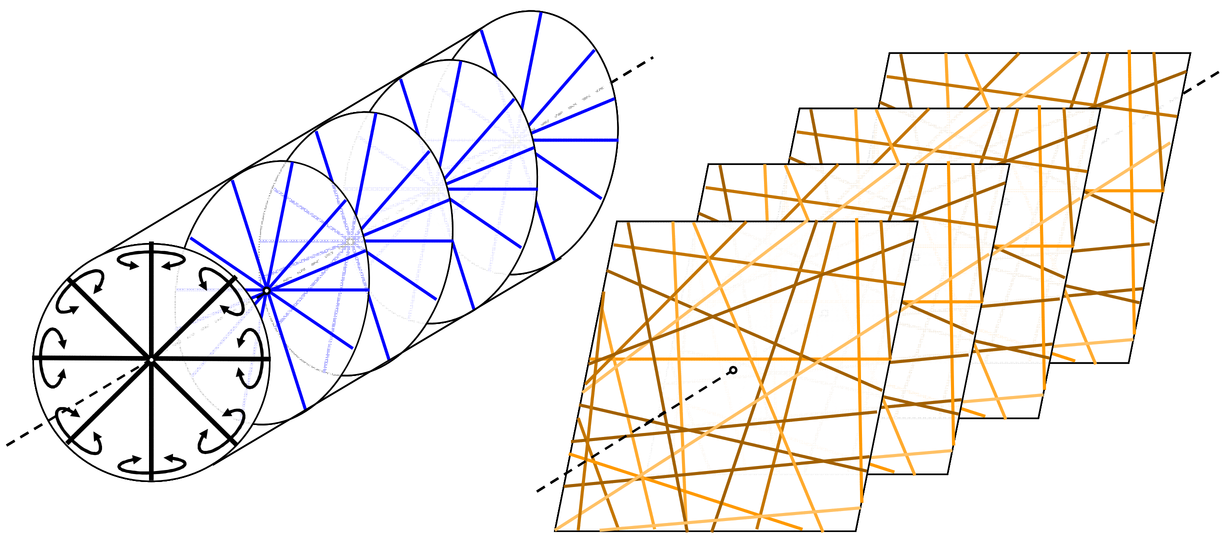

Figure 3.34: *Constraint space of 2 DOF Type 9*

3.3.10 2 DOF Type 10

The freedom space of this type is shown in Figure 3.35. It consists of every translation in a plane. These translations are depicted by the disk of arrows shown in the figure.

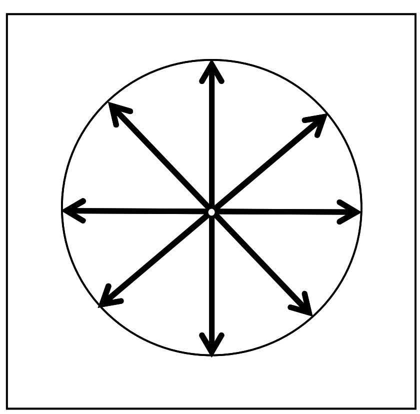

Figure 3.35: *Freedom space of 2 DOF Type 10*

The constraint space of this type is shown in Figure 3.36. It consists of every pure torque line that points in all directions. These pure torque lines are depicted as a sphere of black lines with circular arrows. The constraint space also consists of every parallel constraint line that is orthogonal to the direction of the translations of the freedom space. These constraint lines are depicted as a box of blue parallel lines on the left side of Figure 3.36. The constraint space also consists of wrenches of every q-value that are collinear with every blue constraint line. These wrenches are depicted as a box of orange shaded parallel lines on the right side of Figure 3.36

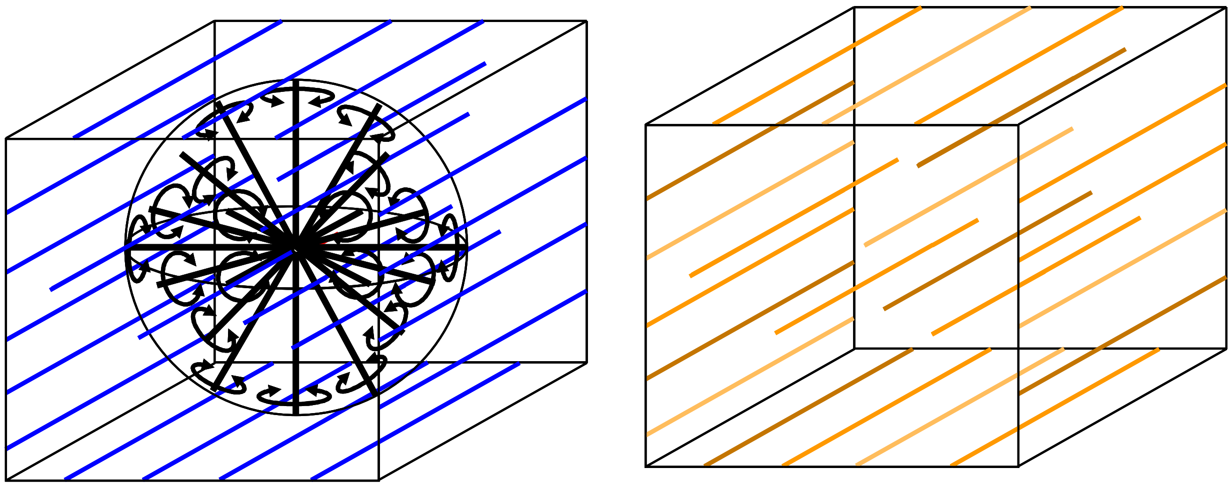

Figure 3.36: *Constraint space of 2 DOF Type 10*

3.4 Types within the 3 DOF Column

This section describes the twenty-two freedom and constraint space types that possess three DOFs.

3.4.1 3 DOF Type 1

The freedom space of this type is shown in Figure 3.37. It consists of all the rotation lines that lie on a plane and a single translation that points in a direction perpendicular to this plane.

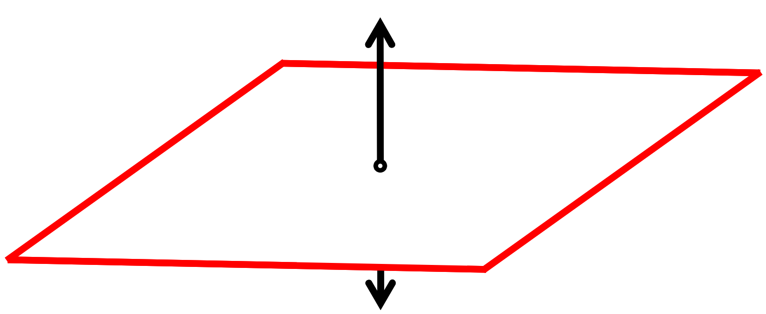

Figure 3.37: *Freedom space of 3 DOF Type 1*

The constraint space of this type is shown in Figure 3.38. It consists of all the constraint lines that lie on the same plane as the rotation lines and a single pure torque line that points in a direction perpendicular to this plane.

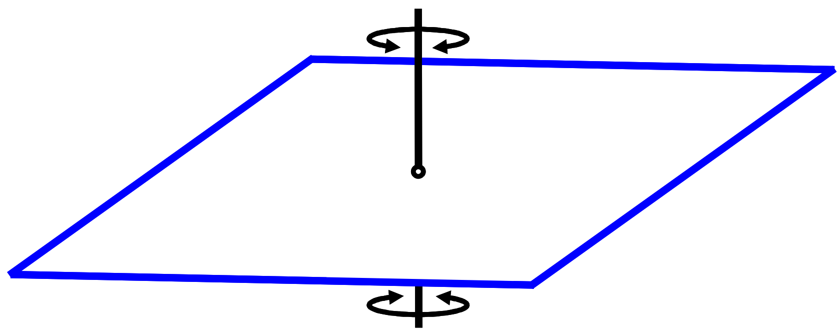

Figure 3.38: *Constraint space of 3 DOF Type 1*

3.4.2 3 DOF Type 2

The freedom space of this type is shown in Figure 3.39. It consists of all parallel rotation lines that point in a specific direction and all translations that point in directions perpendicular to these rotation lines.

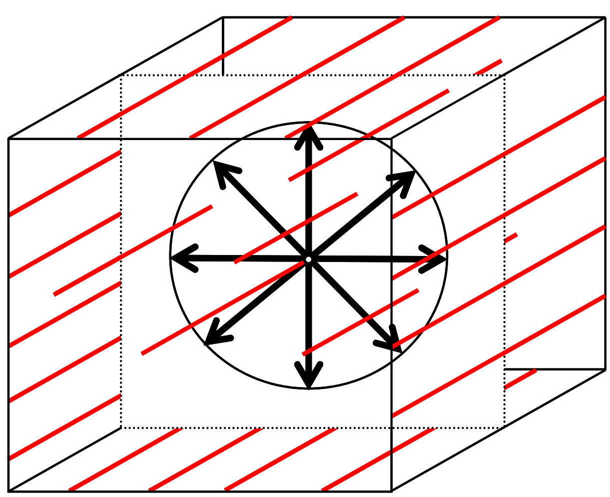

Figure 3.39: *Freedom space of 3 DOF Type 2*

The constraint space of this type is shown in Figure 3.40. It consists of all parallel constraint lines that point in the direction of the rotation lines and all pure torque lines that point in directions perpendicular to these constraint lines.

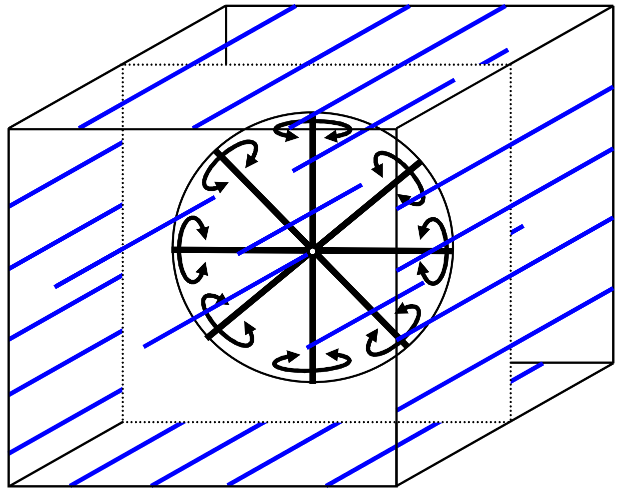

Figure 3.40: *Constraint space of 3 DOF Type 2*

3.4.3 3 DOF Type 3

The freedom space of this type is shown in Figure 3.41. It consists of all rotation lines that intersect a single point.

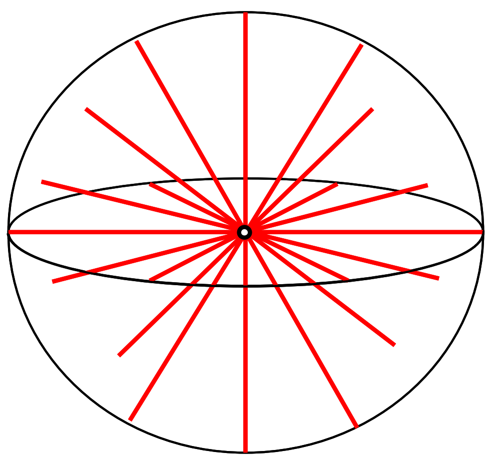

Figure 3.41: *Freedom space of 3 DOF Type 3*

The constraint space of this type is shown in Figure 3.42. It consists of all constraint lines that intersect the same point as the rotation lines of the freedom space.

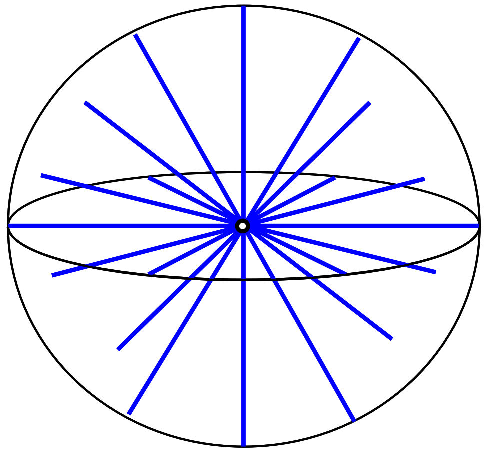

Figure 3.42: *Constraint space of 3 DOF Type 3*

3.4.4 3 DOF Type 4

The freedom space of this type is divided into two forms—Type4A and Type4B. The freedom space of Type4A is shown in Figure 3.43. It consists of all parallel rotation lines that lie on a plane. This plane is perpendicular to another plane that contains all rotation lines in a disk. The disk's center point lies along the intersection of the two planes. The freedom space also consists of a translation that points in a direction perpendicular to the plane of parallel rotation lines.

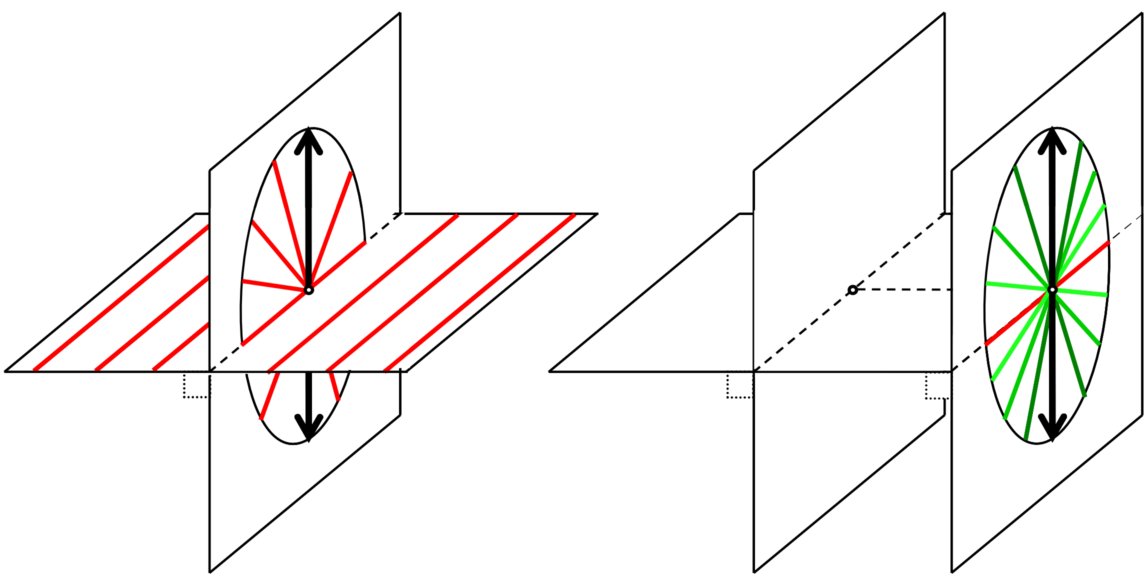

Figure 3.43: *Freedom space of 3 DOF Type 4A*

The freedom space also consists of disks of screws with various pitch values. These disks lie on planes that are perpendicular to the plane of parallel rotation lines. The center point of each disk lies on the intersection line of the disk's plane and the plane of the parallel rotation lines as shown on the right side of Figure 3.43. The vertical plane to the left of the disk of screws shown on the right side of Figure 3.43 is coplanar with the plane of the disk of rotation lines shown on the left side of the figure. A single translation and rotation line belong to each disk of screws. The rotation line always lies on the horizontal plane and the translation always points in the direction normal to this plane. Screws of all pitch values exist within each of these disks. No two screws of the same pitch value exist within any one disk. Note the three vertical planes shown in Figure 3.44. The vertical plane in the middle of this figure is coplanar with the disk of rotation lines from Figure 3.43. If all the screws of a particular pitch value were depicted by themselves, they would look similar to the green screws shown in Figure 3.44. The screws of this and all other sets of screws within the freedom space of Type 4A lie on the surfaces of hyperbolic paraboloids. More specifically, the sets of screws exist within orthogonal ribbons defined and described in Hopkins [21].

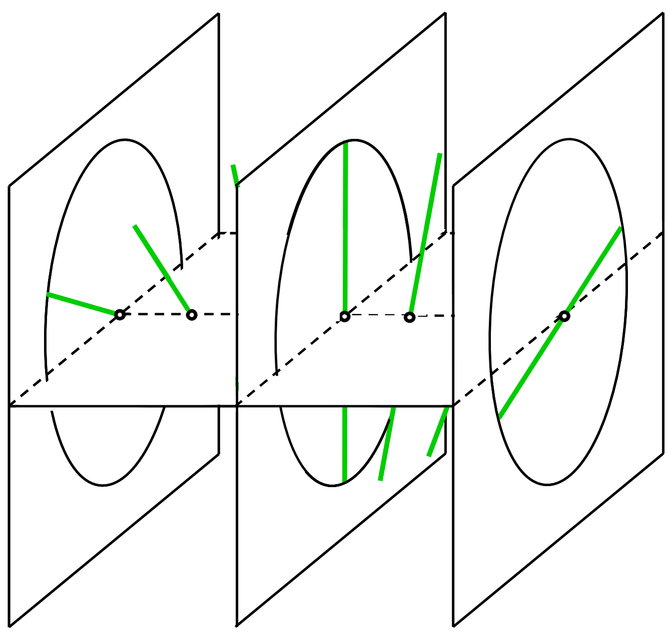

Figure 3.44: *A set of screws of the same pitch value that lie on the surface of a hyperbolic paraboloid and belong to the freedom space of 3 DOF Type 4A.*

The constraint space of this type is shown in Figure 3.45. If the translation, rotation lines, and screw lines of the freedom space from Figure 3.43 were replaced with a pure torque line, constraint lines, and wrench lines respectively, and the entire space where rotated 90 degrees, the resulting space would represent Type 4A's constraint space. The plane of parallel constraint lines from the constraint space is coplanar with the disk of rotation lines from the freedom space. The disk of constraint lines from the constraint space is coplanar with the plane of parallel rotation lines from the freedom space.

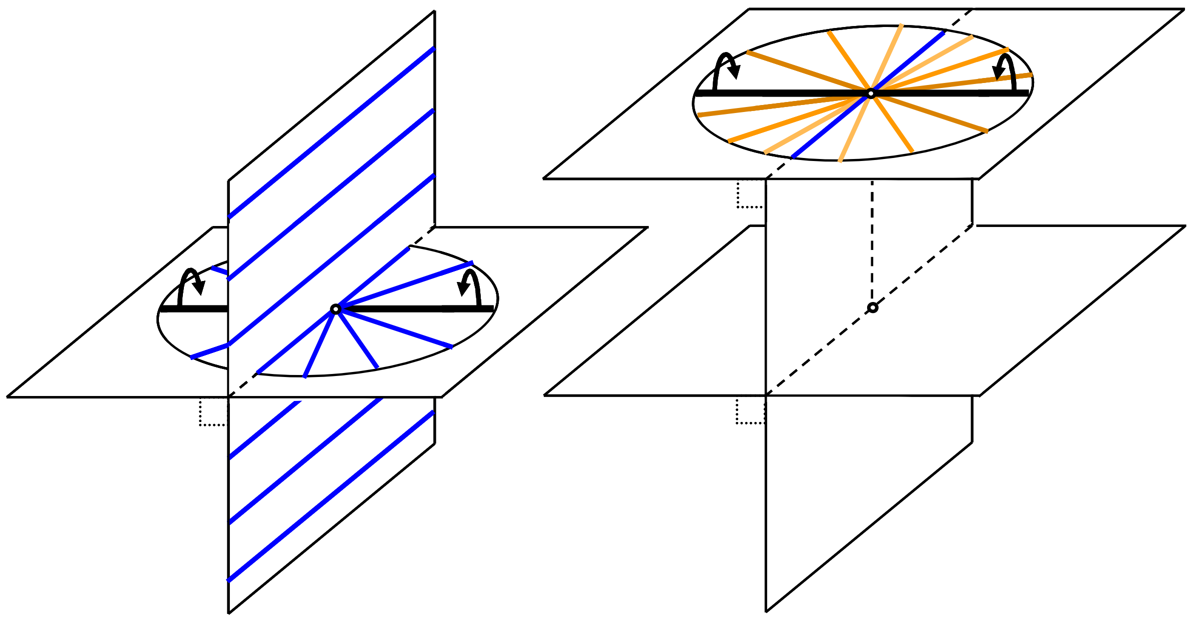

Figure 3.45: *Constraint space of 3 DOF Type 4A*

The freedom space of Type4B is shown in Figure 3.46. It consists of all parallel rotation lines that lie on a plane. This plane is intersected by another non-orthogonal plane that contains all rotation lines in a disk. The disk's center point lies along the intersection of the two planes. The freedom space also consists of a translation that points in a direction perpendicular to the plane of parallel rotation lines.

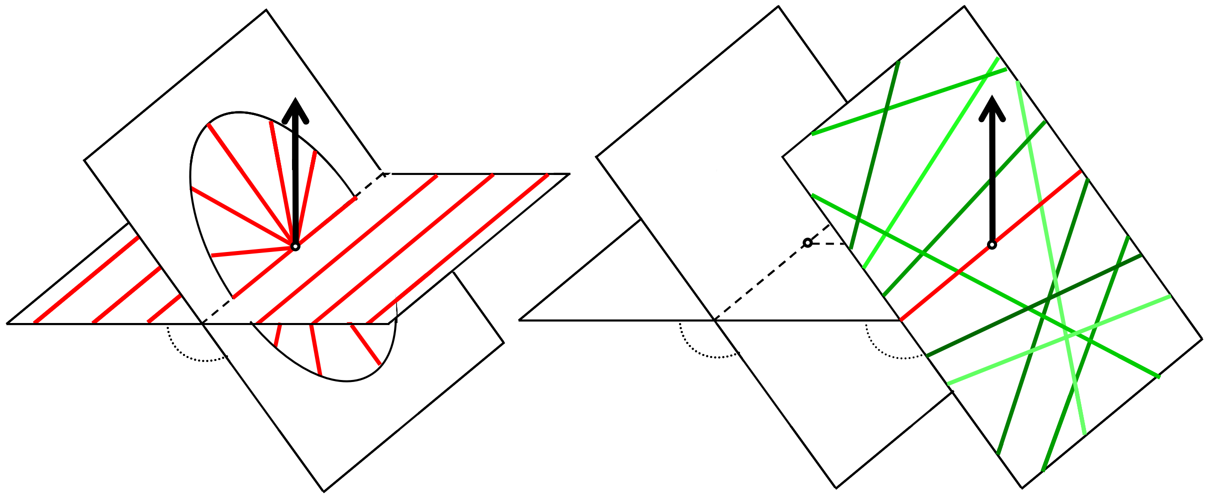

Figure 3.46: *Freedom space of 3 DOF Type 4B*

The freedom space also consists of screws of various pitch values that lie on planes that are parallel to the plane of the disk of rotation lines as shown on the right side of Figure 3.46. The slanted plane to the left of the plane of screws shown on the right side of Figure 3.43 is coplanar with the plane of the disk of rotation lines shown on the left side of the figure. A single rotation line belongs to each plane of screws. The rotation line always lies on the horizontal plane. Screws of all pitch values exist within each of these planes. Sets of screws of common pitch values exist within the freedom space of Type 4B. Each of these sets consists of screws that lie on the surface of a hyperbolic paraboloid. The hyperbolic paraboloids of screws within the freedom space of Type 4B are, however, non-orthogonal ribbons. Non-orthogonal ribbons are defined and described in Hopkins [21].

The constraint space of this type is shown in Figure 3.47. If the translation, rotation lines, and screw lines of the freedom space from Figure 3.46 where replaced with a pure torque line, constraint lines, and wrench lines respectively, and the entire space were reoriented, the resulting space would represent Type 4B's constraint space. The plane of parallel constraint lines from the constraint space is coplanar with the disk of rotation lines from the freedom space. The disk of constraint lines from the constraint space is coplanar with the plane of parallel rotation lines from the freedom space.

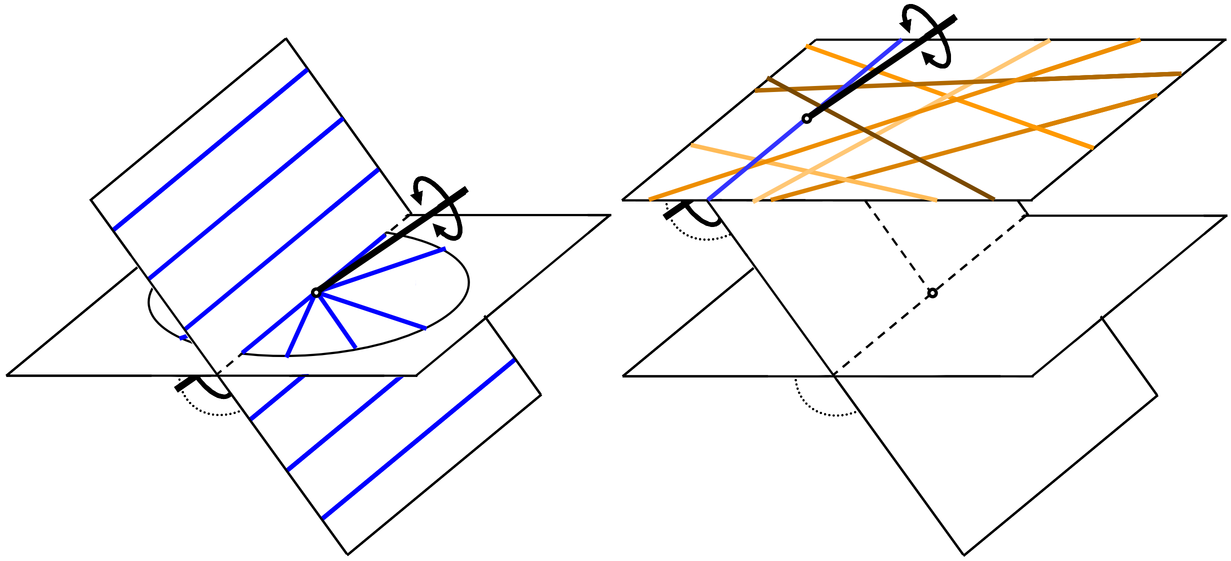

Figure 3.47: *Constraint space of 3 DOF Type 4B*

3.4.5 3 DOF Type 5

The freedom space of this type consists of all rotation lines that lie on the surface of a hyperbolic paraboloid. The freedom space of this type is divided into two forms—Type5A and Type5B. The freedom space of Type5A is shown in Figure 3.48. It consists of all the rotation lines that lie on the surface of an orthogonal ribbon. The freedom space of Type 5B consists of all the rotation lines that lie on the surface of a non-orthogonal ribbon. Both types possess one translation and one characteristic screw. The system's characteristic screw is depicted as a green line on the left side of Figure 3.48.

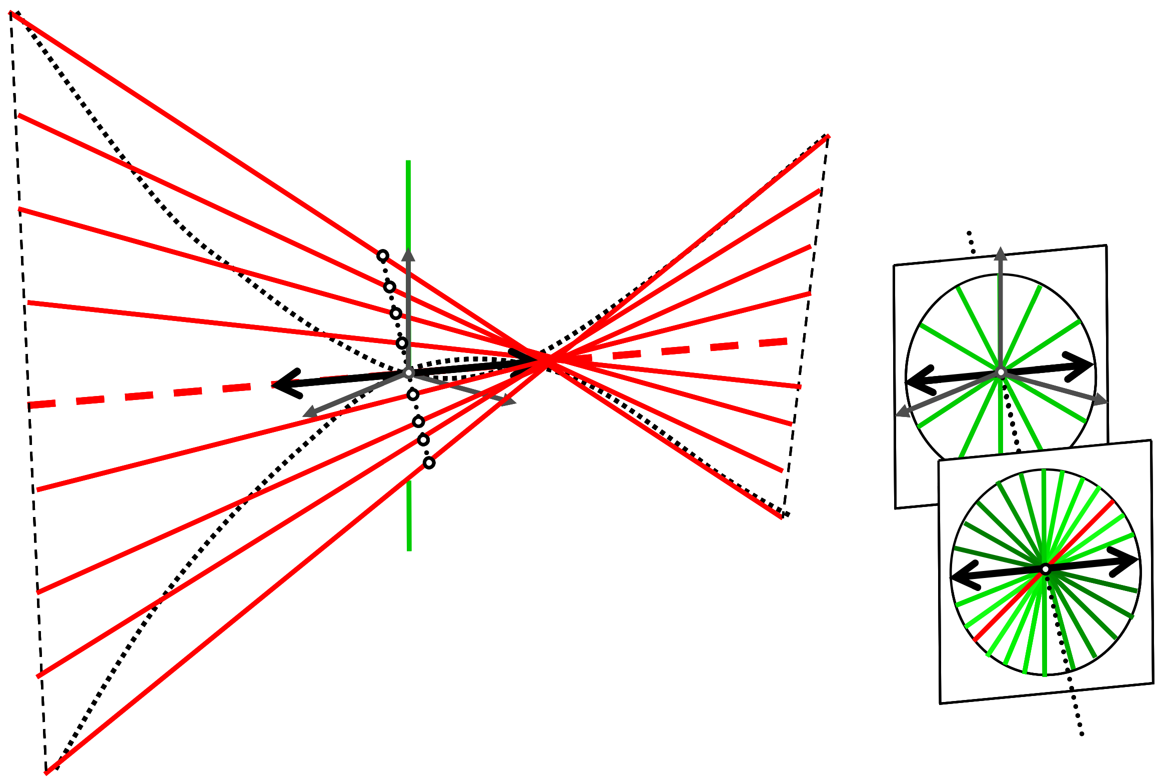

Figure 3.48: *Freedom space of 3 DOF Type 5A*

Type 5A also consists of disks of screws of various pitch values. These disks lie on parallel planes. Each disk contains one of the rotation lines from the red orthogonal ribbon. These disks are shown on the right side of Figure 3.48. Screws of all pitch values exist within each of these disks. No two screws of the same pitch value exist within any one disk. If all the screws of a particular pitch value were depicted by themselves, they would look similar to the green screws shown in Figure 3.44. Many such sets of screws within the freedom space of Type 5A, lie on the surfaces of various orthogonal ribbons. The only exception is the set of screws that possesses the pitch value of the characteristic screw. This set of screws looks like the space shown in Figure 3.49. It consists of a plane of parallel screws that intersects a disk of screws at its center point. The characteristic screw belongs to this disk. The angle θ equals 90 degrees for the freedom space of Type 5A.

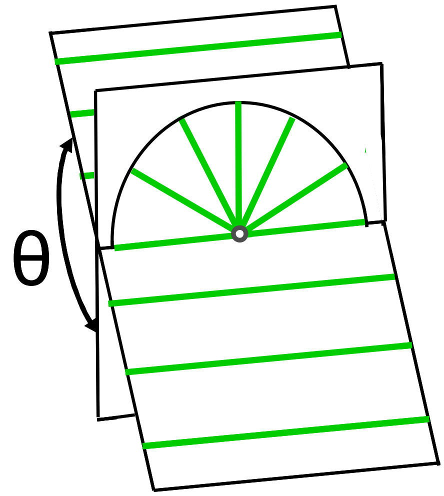

Figure 3.49: *A set of screws within the freedom space of 3 DOF Type 5. The screws within this set possess the characteristic screw's pitch value.*

Type 5B consists of screws of various pitch values that lie on parallel planes. Each plane of screws contains one of the rotation lines from the red non-orthogonal ribbon. Screws of all pitch values exist on each of these planes. Many sets of screws of common pitch values lie on the surfaces of non-orthogonal ribbons. The only exception is the set of screws that possesses the pitch value of the characteristic screw. This set of screws also looks like the space shown in Figure 3.49 where the angle θ does not equal 90 degrees for the freedom space of Type 5B.

The constraint space of this type is shown in Figure 3.50. It consists of all constraint lines that lie on the same hyperbolic paraboloid as the rotation lines of the freedom space. The constraint space also consists of a pure torque line and a characteristic wrench with a q-value equal to the negative pitch value of the freedom space's characteristic screw. Both the characteristic wrench and the characteristic screw are collinear. Similar to the system's freedom space, the constraint space consists of wrenches of various q-values that lie *(i)* within parallel disks for Type 5A or *(ii)* on parallel planes for Type 5B. The orange characteristic wrench shown on the left side of Figure 3.50 exists within the far right disk on the right side of the same figure.

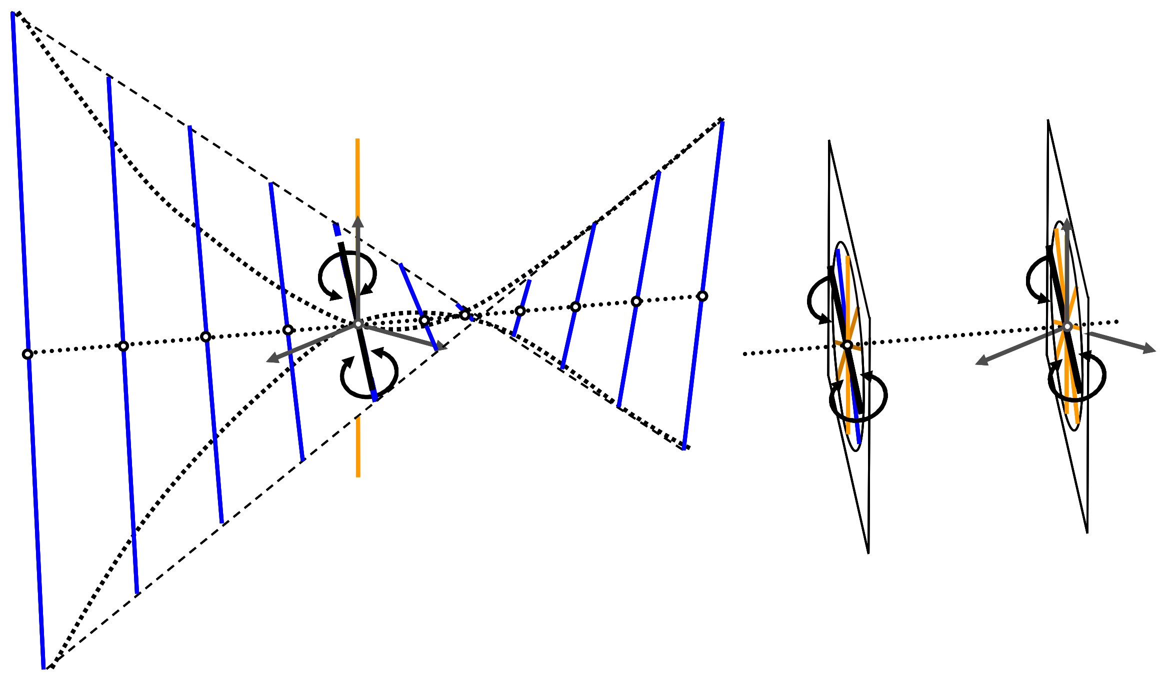

Figure 3.50: *Constraint space of 3 DOF Type 5A*

See Hopkins [21] for a more detailed description of this type's freedom and constraint space pair.

3.4.6 3 DOF Type 6

The freedom space of this type is shown in Figure 3.51. It consists of an infinite number of consecutive cylindroids from the freedom space of 2 DOF Type 3. The axes of these cylindroids are parallel. The central plane in the middle of Figure 3.51 consists of two orthogonal sets of parallel screw lines. One set of parallel lines consists of screws with a negative pitch value and the other set consists of screws with a positive pitch value. Each line within each set represents a principle generator of one of the consecutive cylindroids that make up the freedom space of Figure 3.51. The sets of parallel screw lines on the top and bottom planes of Figure 3.51 represent the extreme generators of the consecutive cylindroids. Two sets of parallel rotation lines lie on two different parallel planes. The freedom space also consists of a pure translation that is normal to these planes.

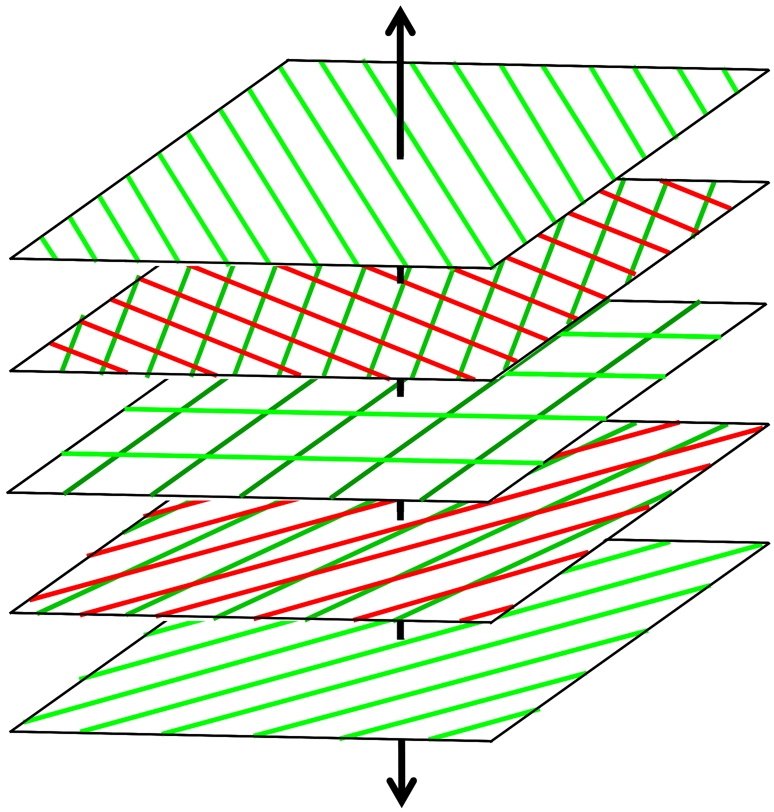

Figure 3.51: *Freedom space of 3 DOF Type 6*

The constraint space of this type is shown in Figure 3.52. If the translation, rotation lines, and screw lines of the freedom space from Figure 3.51 where replaced with a pure torque line, constraint lines, and wrench lines with q-values equal to the negative pitch values of their corresponding screw lines respectively, and all the planes above the middle plane were replaced with the planes below the middle plane and all the planes below the middle plane were replaced with the planes above the middle plane, the resulting space would represent the constraint space shown in Figure 3.52.

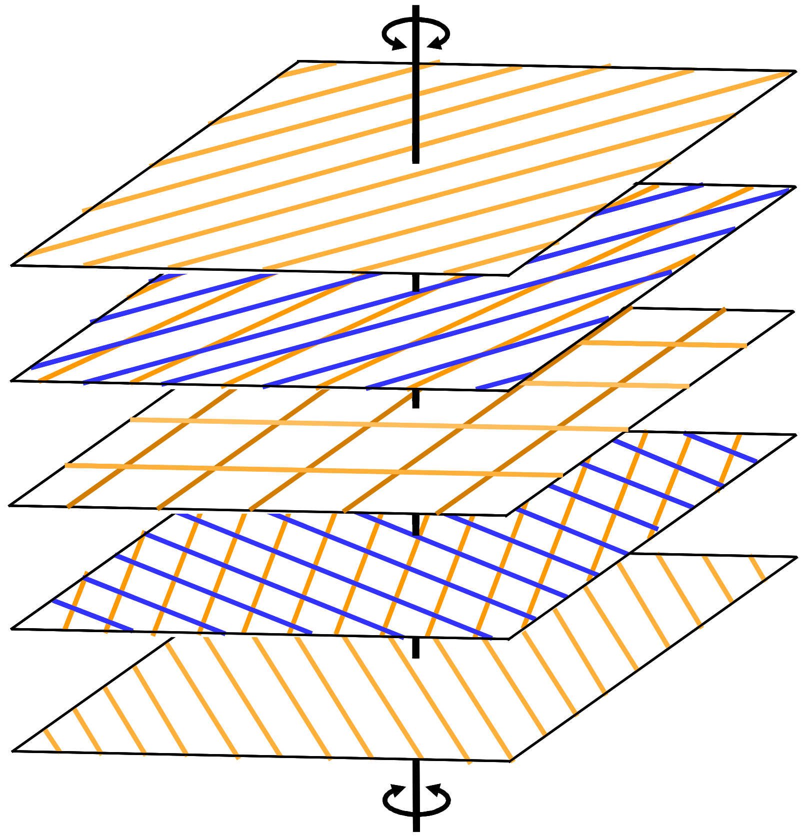

Figure 3.52: *Constraint space of 3 DOF Type 6*

3.4.7 3 DOF Type 7

The freedom space of this type is shown in Figure 3.53. The twists within this space may be visualized by rotating the cylindroid from the freedom space of 2 DOF Type 3 about one of its principle generators as shown on the right side of Figure 3.53. The vertical dashed lines shown in the figure are collinear with this principle generator. Rotation lines lie on the surface of a circular hyperboloid as shown on the left side of Figure 3.53. The screws in the disk shown on the plane of the central circular cross-section of this hyperboloid all have the same pitch value. Each of the screws in the disk represents one of the principle generators of the cylindroids that make up the space. The pitch value of these screws has an opposite sign from the pitch value of the screw that is collinear with the vertical dashed line.

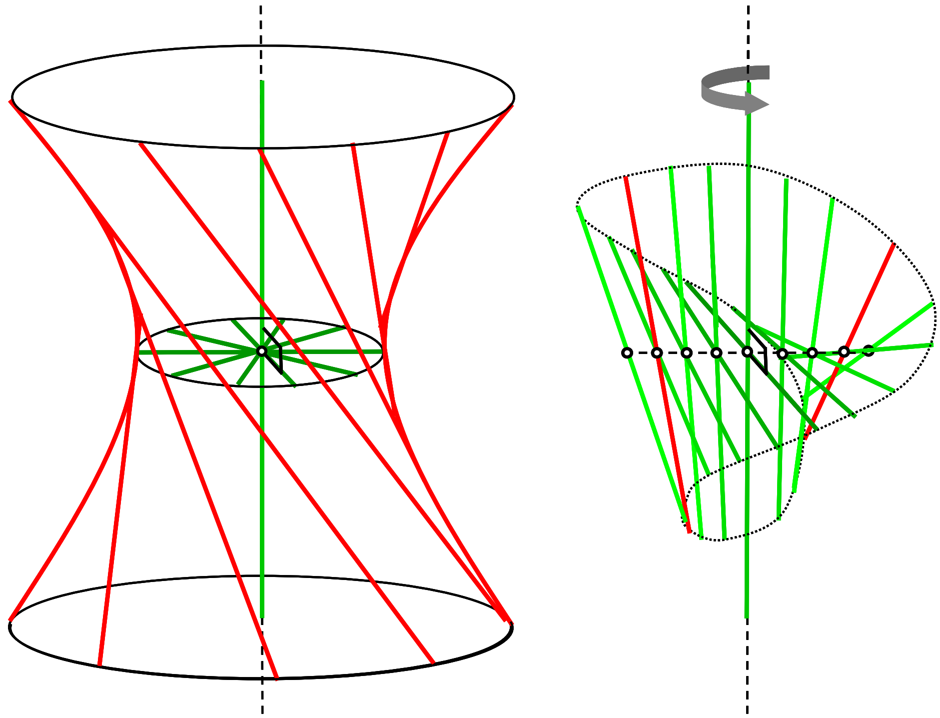

Figure 3.53: *Freedom space of 3 DOF Type 7*

The constraint space of this type is shown in Figure 3.54. The wrenches within this space may be visualized by rotating the cylindroid shown on the right side of the figure about one of its principle generators. The vertical dashed lines shown in Figure 3.53 and Figure 3.54 are collinear with this principle generator. Constraint lines lie on the surface of the same circular hyperboloid as the rotation lines from the freedom space. The wrenches in the disk shown on the plane of the central circular cross-section of this hyperboloid all have a q-value that is equal to the negative of the pitch value of the screws in the disk of the freedom space. Each of the wrenches in the disk represents one of the principle generators of the cylindroids that make up the space. The q-value of these wrenches has an opposite sign from the q-value of the wrench that is collinear with the vertical dashed line.

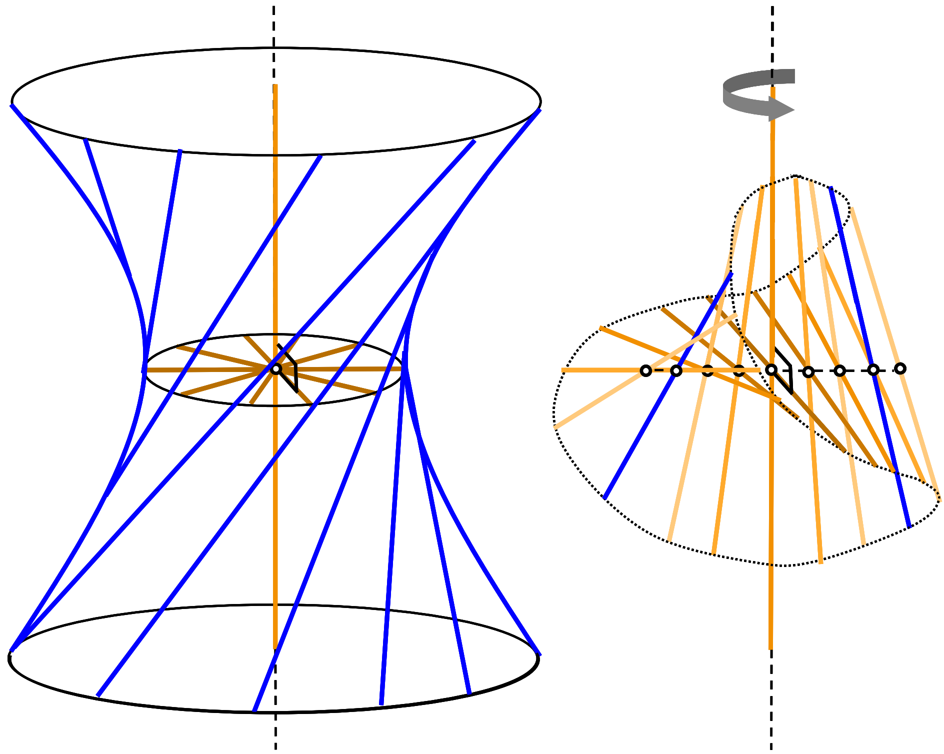

Figure 3.54: *Constraint space of 3 DOF Type 7*

3.4.8 3 DOF Type 8

The freedom space of this type is shown in Figure 3.55. It consists of three orthogonal twists that intersect at a common point as shown on the left side of Figure 3.55. The pitch values of all three of these twists must be different. The twist that is collinear with the vertical dashed line is a screw. If the pitch value of this screw is a positive value, the pitch values of the other two orthogonal screws that intersect it are negative. If the pitch value of the vertical screw is a negative value, the pitch values of the other two orthogonal screws that intersect it are positive. The pitch value of the screw that lies on the intersection line of the planes shown on the left side of Figure 3.55 is between the pitch values of the other two orthogonal screws. Two disks of screws that have a pitch value equal to the pitch value of the screw that lies on the intersection line of the planes exist on two of these planes as shown on the left side of Figure 3.55. The angles between the planes of each disk and the vertical plane shown on the left side of Figure 3.55 are equal. The center points of each disk are equally spaced from the intersection of the orthogonal twists. An elliptical hyperboloid of rotation lines also exists that is shown on the right side of Figure 3.55. The central elliptical cross-section of this hyperboloid is normal to the vertical dashed line. Many other sets of screws also exist in the freedom space. These screws lie on the surfaces of elliptical hyperboloids. These hyperboloids are oriented in such a way that they are not practical to visually depict.

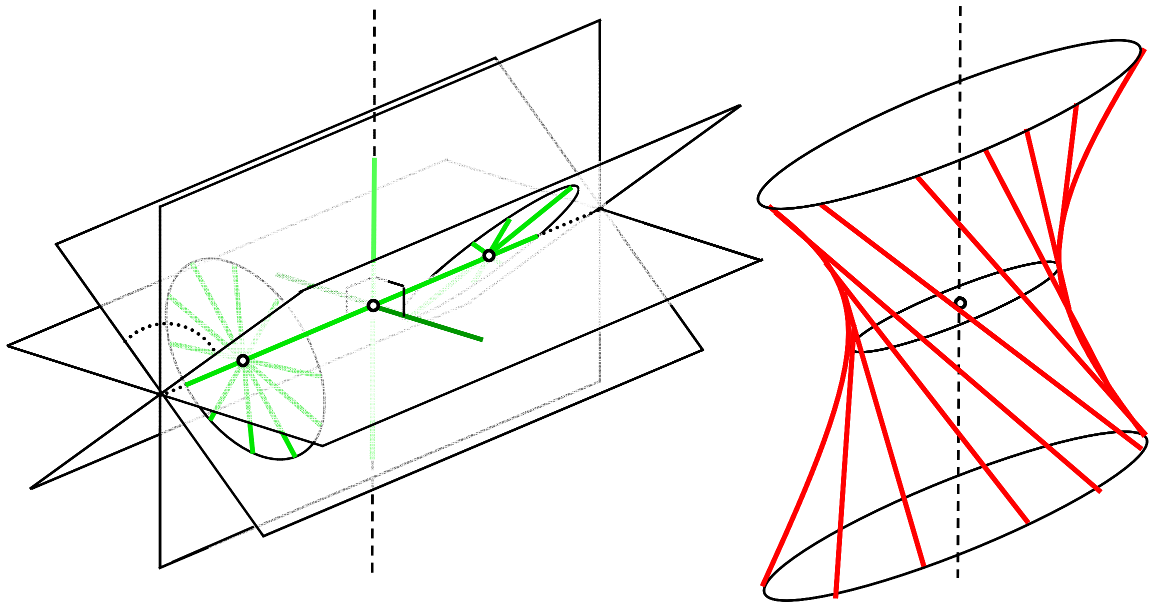

Figure 3.55: *Freedom space of 3 DOF Type 8*

The constraint space of this type is shown in Figure 3.56. It consists of three orthogonal wrenches that intersect at a common point as shown on the left side of Figure 3.56. These wrenches are collinear with the three orthogonal screws of the freedom space. The q-value of each of these wrenches equals the negative of the pitch value of its corresponding screw from the freedom space. Two disks of wrenches that have a q-value equal to the q-value of the wrench that lies on the intersection line of the intersecting planes shown on the left side of Figure 3.56 exist in the constraint space. These disks lie on the same two planes on which the disks of screws from the freedom space lie. Each disk of wrenches shares the same center point as one of the disks of screws. Constraint lines also exist that lie on the surface of the same elliptical hyperboloid as the rotation lines of the freedom space as shown on the right side of Figure 3.56. Many other sets of wrenches also exist in the constraint space. These wrenches lie on the surfaces of elliptical hyperboloids. These hyperboloids are oriented in such a way that they are not practical to visually depict.

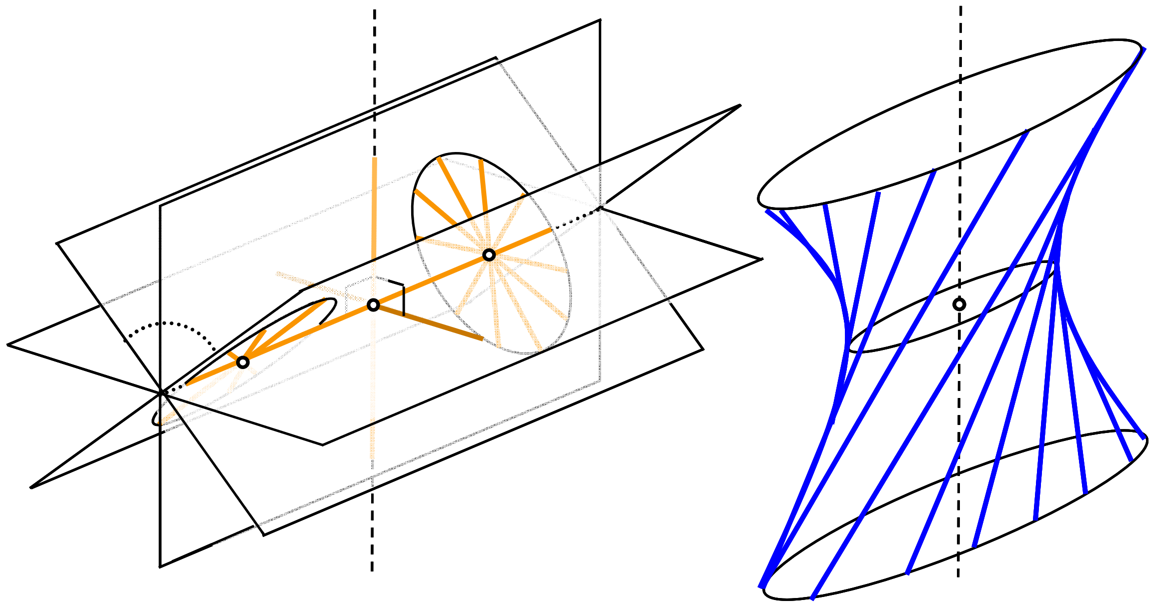

Figure 3.56: *Constraint space of 3 DOF Type 8*

3.4.9 3 DOF Type 9

The freedom space of this type is shown in Figure 3.57. It consists of three orthogonal twists that intersect at a common point as shown on the left side of Figure 3.57. The pitch values of all three of these twists must be different. Two of these twists are screws and are shown as green lines on the left side of Figure 3.57. The pitch values of these screws have opposite signs. The twist that lies on the intersection line of the planes shown on the left side of Figure 3.57 is a rotation line. Two disks of rotation lines lie on two of these planes. The angles between the planes of each disk and the vertical plane shown on the left side of Figure 3.57 are equal. The center points of each disk are equally spaced from the intersection of the orthogonal twists. An elliptical hyperboloid of screws also exists that is shown on the right side of Figure 3.57. The central elliptical cross-section of this hyperboloid is normal to the vertical dashed line. Many other sets of screws also exist in the freedom space. These screws lie on the surfaces of elliptical hyperboloids. These hyperboloids are oriented in such a way that they are not practical to visually depict.

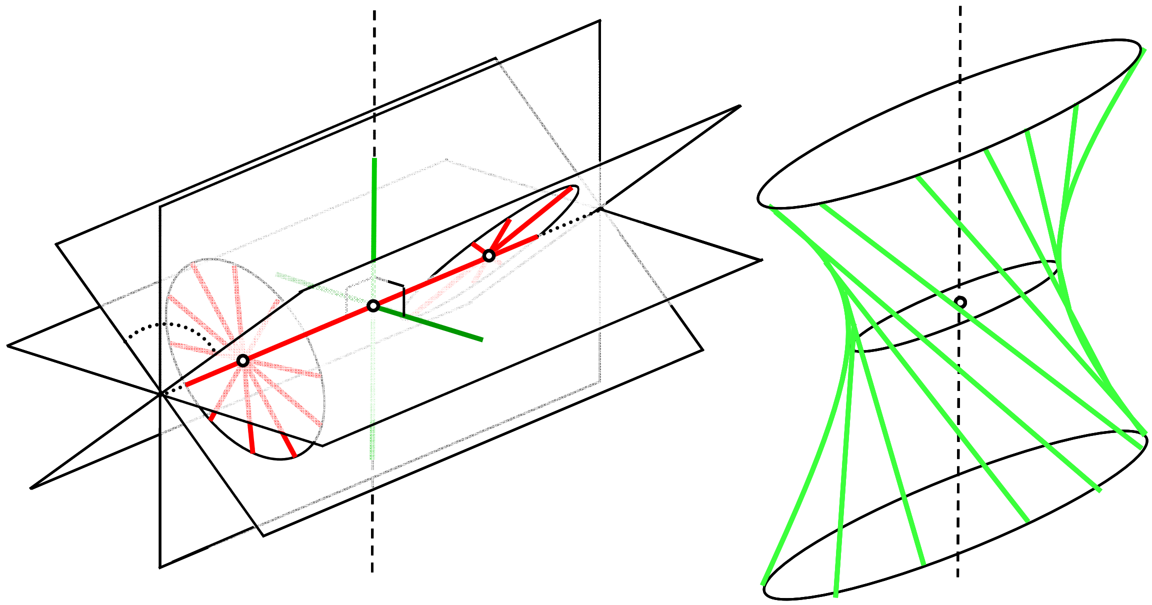

Figure 3.57: *Freedom space of 3 DOF Type 9*

The constraint space of this type is shown in Figure 3.58. It consists of two orthogonal wrenches and one orthogonal constraint line that intersects at a common point as shown on the left side of Figure 3.58. The two wrenches and the constraint line are collinear with the three orthogonal twists of the freedom space. The q-value of each of these wrenches equals the negative of the pitch value of its corresponding screw from the freedom space. Two disks of constraint lines exist in the constraint space. These disks lie on the same two planes on which the disks of rotation lines from the freedom space lie. Each disk of constraint lines shares the same center point as one of the disks of rotation lines. Wrenches also exist that lie on the surface of the same elliptical hyperboloid as the screw lines of the freedom space as shown on the right side of Figure 3.58. Many other sets of wrenches also exist in the constraint space. These wrenches lie on the surfaces of elliptical hyperboloids. These hyperboloids are oriented in such a way that they are not practical to visually depict.

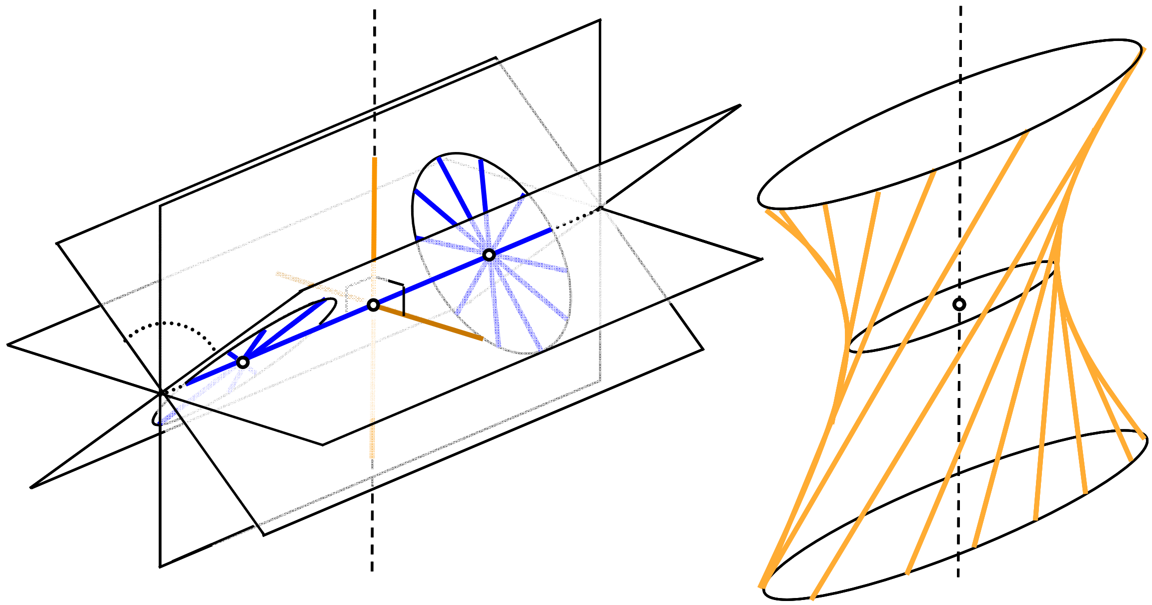

Figure 3.58: *Constraint space of 3 DOF Type 9*

3.4.10 3 DOF Type 10

The freedom space of this type is shown in Figure 3.59. It consists of three orthogonal twists that intersect at a common point as shown on the left side of Figure 3.59. The pitch values of all three of these twists must be different. Two of these twists are screws and are shown as green lines on the left side of Figure 3.59. The pitch values of these screws are either both negative or both positive. The vertical twist that is collinear with the dashed line shown on the left side of Figure 3.59 is a rotation line. The pitch value of the screw that lies on the intersection line of the planes shown on the left side of Figure 3.59 is between zero and the pitch value of the horizontal screw shown in dark green. Two disks of screws that have a pitch value equal to the pitch value of the screw that lies on the intersection line of the planes exist on two of these planes as shown on the left side of Figure 3.59. The angles between the planes of each disk and the vertical plane shown on the left side of Figure 3.59 are equal. The center points of each disk are equally spaced from the intersection of the orthogonal twists. An elliptical hyperboloid of screws also exists that is shown on the right side of Figure 3.55. The central elliptical cross-section of this hyperboloid is normal to the vertical dashed line. Many other sets of screws also exist in the freedom space. These screws lie on the surfaces of elliptical hyperboloids. These hyperboloids are oriented in such a way that they are not practical to visually depict.

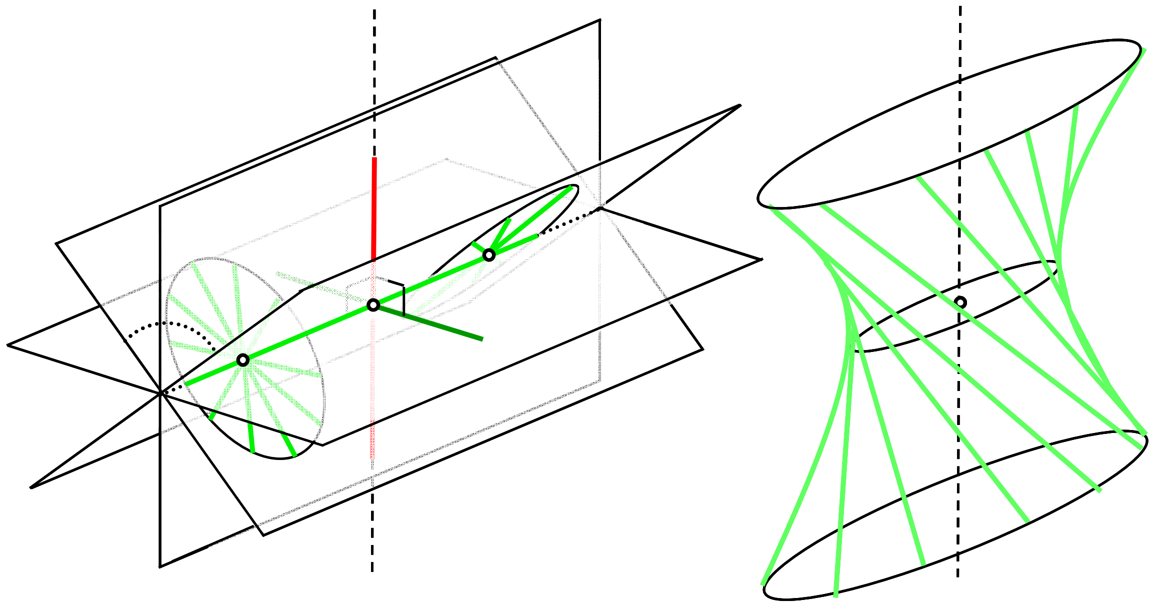

Figure 3.59: *Freedom space of 3 DOF Type 10*

The constraint space of this type is shown in Figure 3.60. It consists of two orthogonal wrenches and one orthogonal constraint line that intersects at a common point as shown on the left side of Figure 3.60. The two wrenches and the constraint line are collinear with the three orthogonal twists of the freedom space. The q-value of each of these wrenches equals the negative of the pitch value of its corresponding screw from the freedom space. Two disks of wrenches exist in the constraint space. These disks lie on the same two planes on which the disks of screws from the freedom space lie. Each disk of wrenches shares the same center point as one of the disks of screws. Wrenches also exist that lie on the surface of the same elliptical hyperboloid as the screw lines of the freedom space as shown on the right side of Figure 3.60. Many other sets of wrenches also exist in the constraint space. These wrenches lie on the surfaces of elliptical hyperboloids. These hyperboloids are oriented in such a way that they are not practical to visually depict.

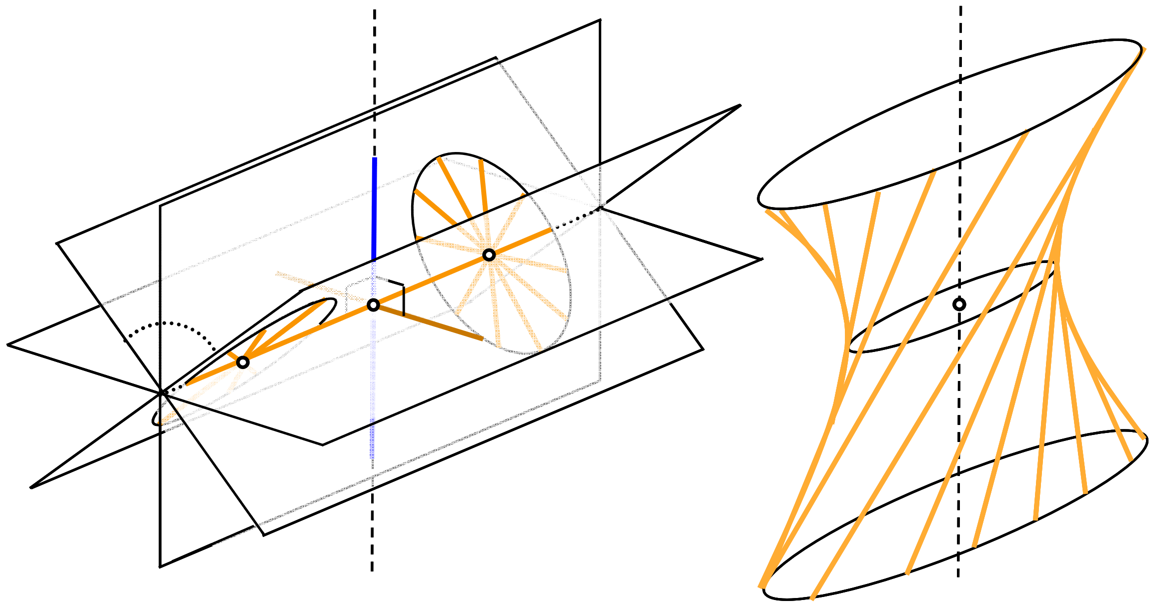

Figure 3.60: *Constraint space of 3 DOF Type 10*

3.4.11 3 DOF Type 11

The freedom space of this type is shown in Figure 3.61. It consists of three orthogonal screws that intersect at a common point as shown on the left side of Figure 3.61. The pitch values of all three of these screws are unique but have the same sign. The pitch value of the screw that lies on the intersection line of the planes shown on the left side of Figure 3.61 is between the pitch values of the other two orthogonal screws. Two disks of screws that have a pitch value equal to the pitch value of the screw that lies on the intersection line of the planes exist on two of these planes as shown on the left side of Figure 3.61. The angles between the planes of each disk and the vertical plane shown on the left side of Figure 3.61 are equal. The center points of each disk are equally spaced from the intersection of the orthogonal twists. An elliptical hyperboloid of screws also exists that is shown on the right side of Figure 3.61. The central elliptical cross-section of this hyperboloid is normal to the vertical dashed line. Many other sets of screws also exist in the freedom space. These screws lie on the surfaces of elliptical hyperboloids. These hyperboloids are oriented in such a way that they are not practical to visually depict.

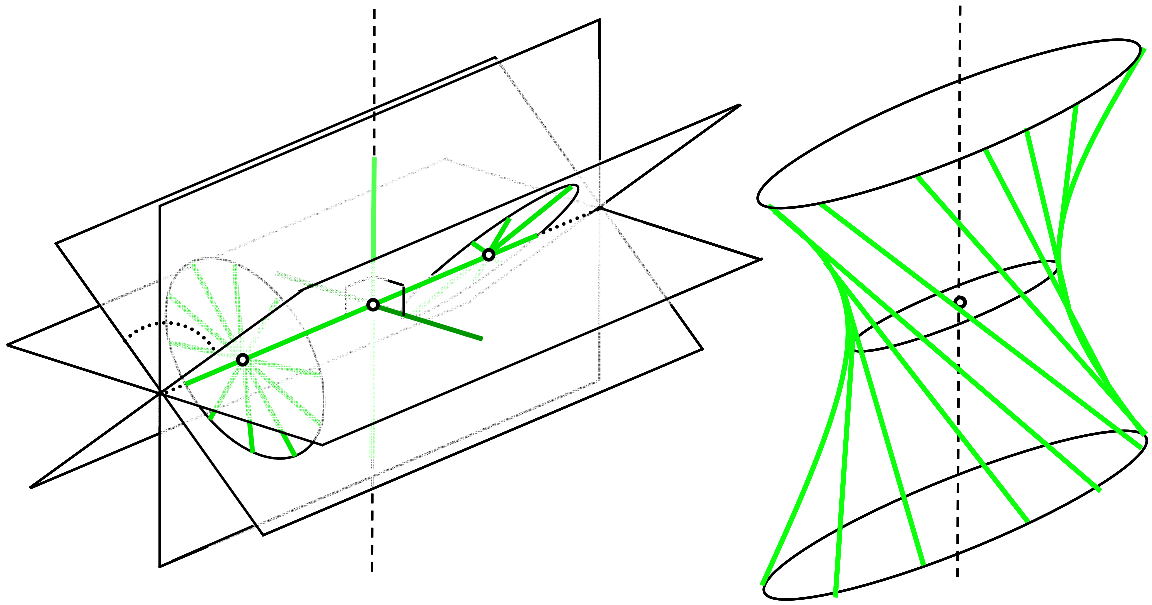

Figure 3.61: *Freedom space of 3 DOF Type 11*

The constraint space of this type is shown in Figure 3.62. It consists of three orthogonal wrenches that intersect at a common point as shown on the left side of Figure 3.62. These wrenches are collinear with the three orthogonal screws of the freedom space. The q-value of each of these wrenches equals the negative of the pitch value of its corresponding screw from the freedom space. Two disks of wrenches that have a q-value equal to the q-value of the wrench that lies on the intersection line of the intersecting planes shown on the left side of Figure 3.62 exist in the constraint space. These disks lie on the same two planes on which the disks of screws from the freedom space lie. Each disk of wrenches shares the same center point as one of the disks of screws. Wrenches also exist that lie on the surface of the same elliptical hyperboloid as the screws of the freedom space as shown on the right side of Figure 3.62. Many other sets of wrenches also exist in the constraint space. These wrenches lie on the surfaces of elliptical hyperboloids. These hyperboloids are oriented in such a way that they are not practical to visually depict.

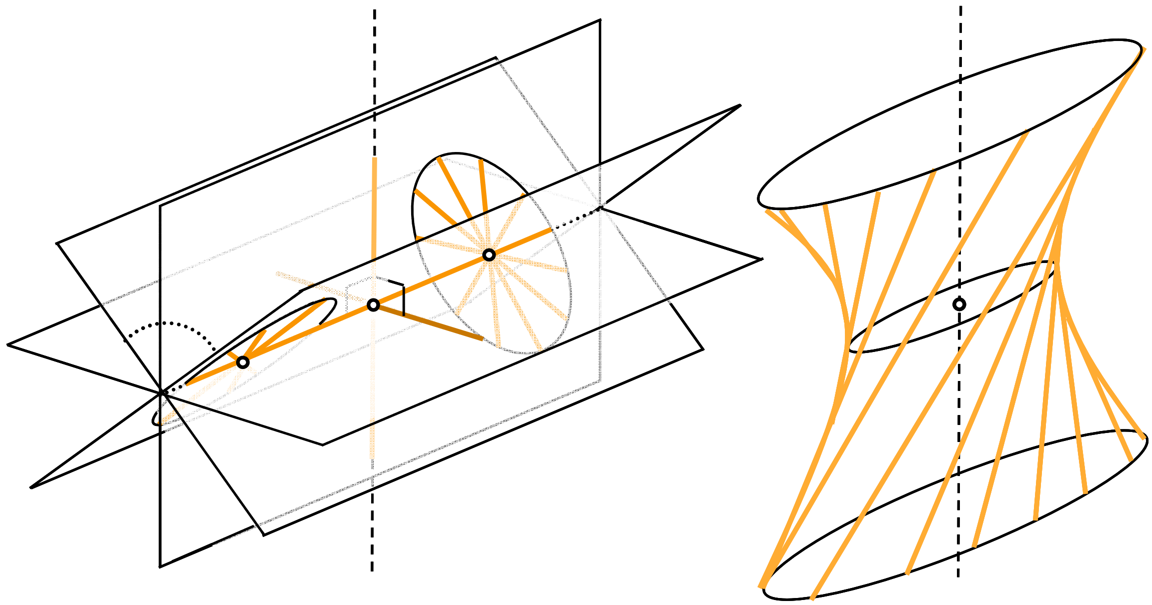

Figure 3.62: *Constraint space of 3 DOF Type 11*

3.4.12 3 DOF Type 12

The freedom space of this type is shown in Figure 3.63. The twists within this space may be visualized by rotating the cylindroid from the freedom space of 2 DOF Type 6 about the principle generator that is a screw as shown on the right side of Figure 3.63. The vertical dashed lines shown in the figure are collinear with this principle generator. Screws lie on the surface of a circular hyperboloid as shown on the left side of Figure 3.63. Rotation lines exist within the disk shown on the plane of the central circular cross-section of this hyperboloid. Each of the rotation lines in the disk represents one of the principle generators of the cylindroids that make up the space. The pitch values of all the screws in the freedom space are either all positive or all negative.

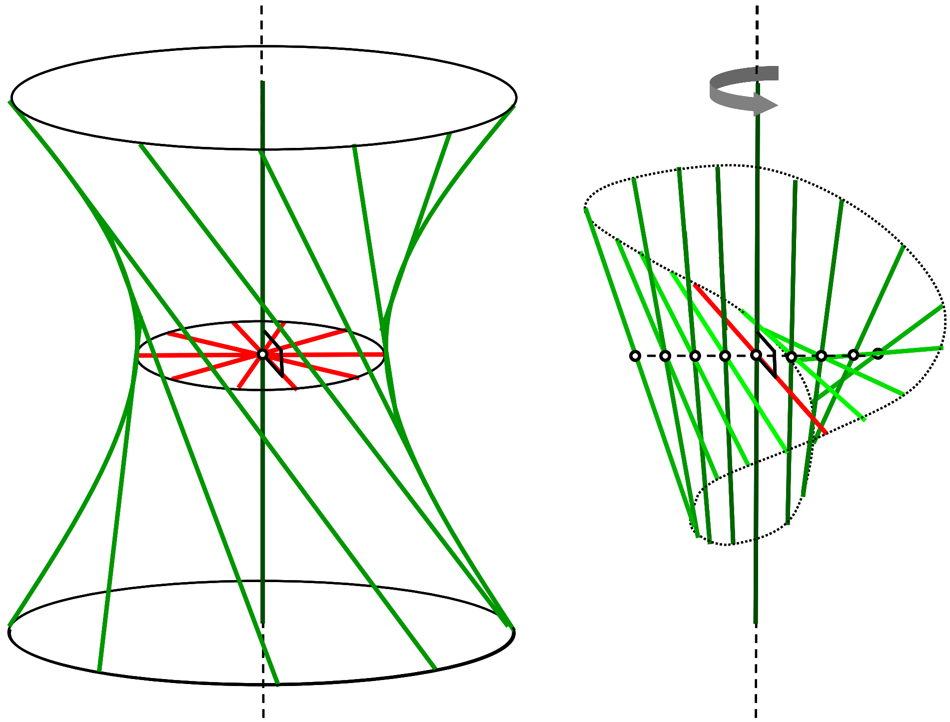

Figure 3.63: *Freedom space of 3 DOF Type 12*

The constraint space of this type is shown in Figure 3.64. The wrenches within this space may be visualized by rotating the cylindroid shown on the right side of the figure about one of its principle generators. The vertical dashed lines shown in Figure 3.63 and Figure 3.64 are collinear with this principle generator. The q-value of the vertical wrench is equal to the negative of the pitch value of the vertical screw from the freedom space. Wrenches lie on the surface of the same circular hyperboloid as the screws from the freedom space. Constraint lines lie within the disk shown on the plane of the central circular cross-section of this hyperboloid. Each constraint line in the disk represents one of the principle generators of the cylindroids that make up the space. The q-value of the vertical wrench is equal to the negative of the pitch value of the vertical screw from the freedom space.

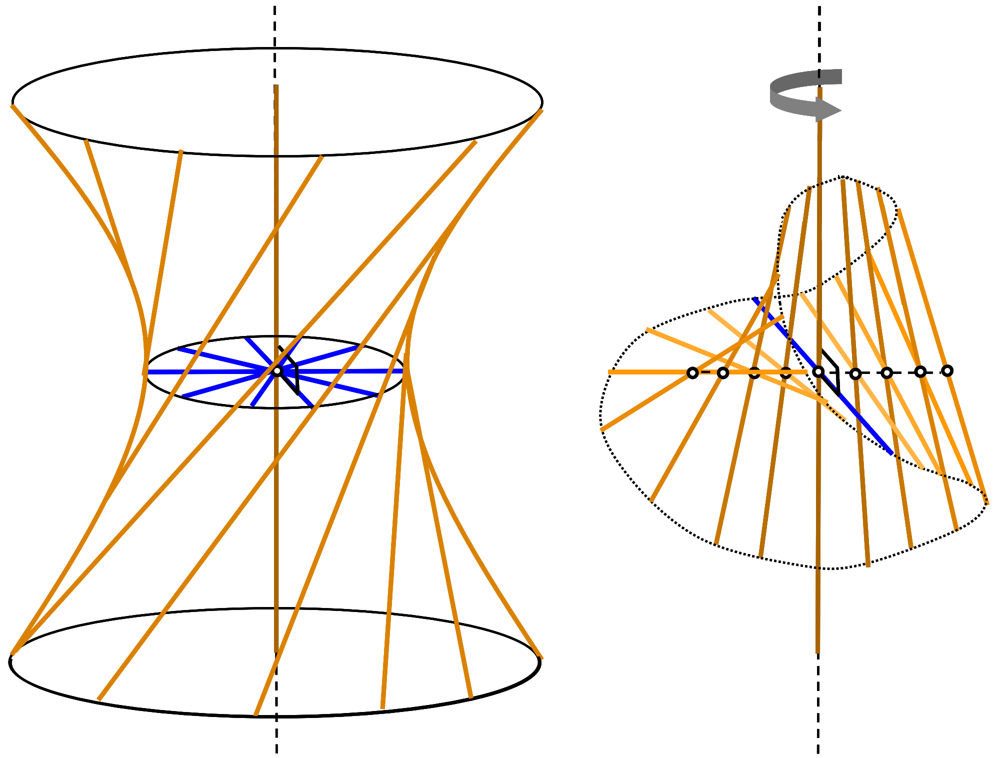

Figure 3.64: *Constraint space of 3 DOF Type 12*

3.4.13 3 DOF Type 13

The freedom space of this type is shown in Figure 3.65. The twists within this space may be visualized by rotating the cylindroid from the freedom space of 2 DOF Type 6 about the principle generator that is a rotation line as shown on the right side of Figure 3.65. The vertical dashed lines shown in the figure are collinear with this principle generator. Screws lie on the surface of a circular hyperboloid as shown on the left side of Figure 3.65. The screws in the disk shown on the plane of the central circular cross-section of this hyperboloid all have the same pitch value. Each of the screws in the disk represents one of the principle generators of the cylindroids that make up the space.

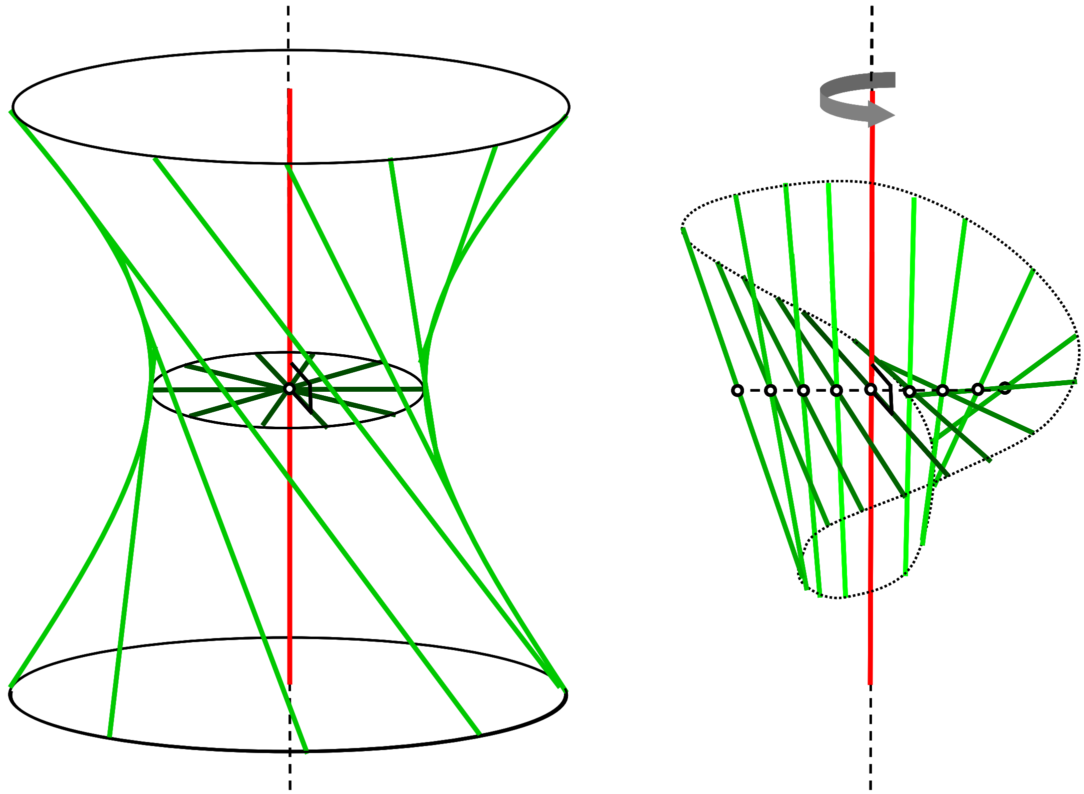

Figure 3.65: *Freedom space of 3 DOF Type 13*

The constraint space of this type is shown in Figure 3.66. The wrenches within this space may be visualized by rotating the cylindroid shown on the right side of the figure about the principle generator that is a constraint line. The vertical dashed lines shown in Figure 3.65 and Figure 3.66 are collinear with this principle generator. Wrenches lie on the surface of the same circular hyperboloid as the screws from the freedom space. The wrenches in the disk shown on the plane of the central circular cross-section of this hyperboloid all have a q-value that is equal to the negative of the pitch value of the screws in the disk of the freedom space. Each of the wrenches in the disk represents one of the principle generators of the cylindroids that make up the space.

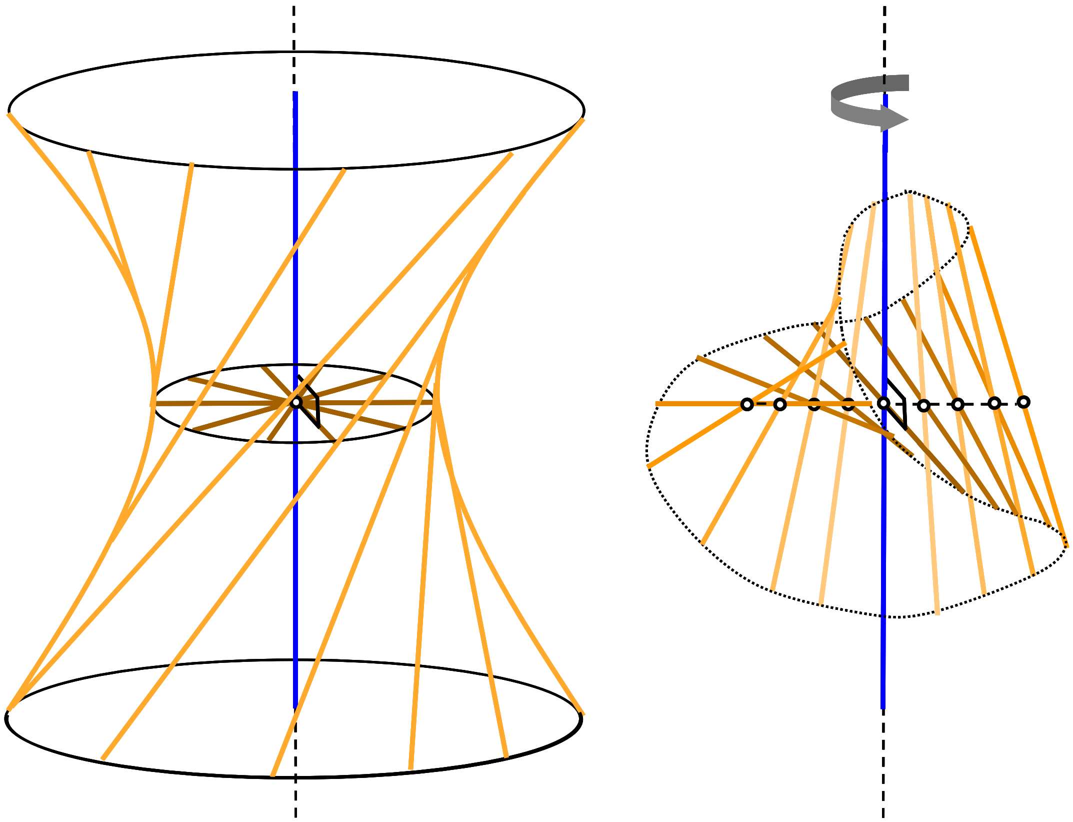

Figure 3.66: Constraint space of 3 DOF Type 13

3.4.14 3 DOF Type 14

The freedom space of this type is shown in Figure 3.67. The screws within this space may be visualized by rotating the cylindroid from the freedom space of 2 DOF Type 7 about one of its principle generators as shown on the right side of Figure 3.67. The vertical dashed lines shown in the figure are collinear with this principle generator. Screws lie on the surface of a circular hyperboloid as shown on the left side of Figure 3.67. The screws in the disk shown on the plane of the central circular cross-section of this hyperboloid all have the same pitch value. Each of the screws in the disk represents one of the principle generators of the cylindroids that make up the space. The pitch value of these screws has the same sign from the pitch value of the screw that is collinear with the vertical dashed line. No rotation lines exist within this freedom space.

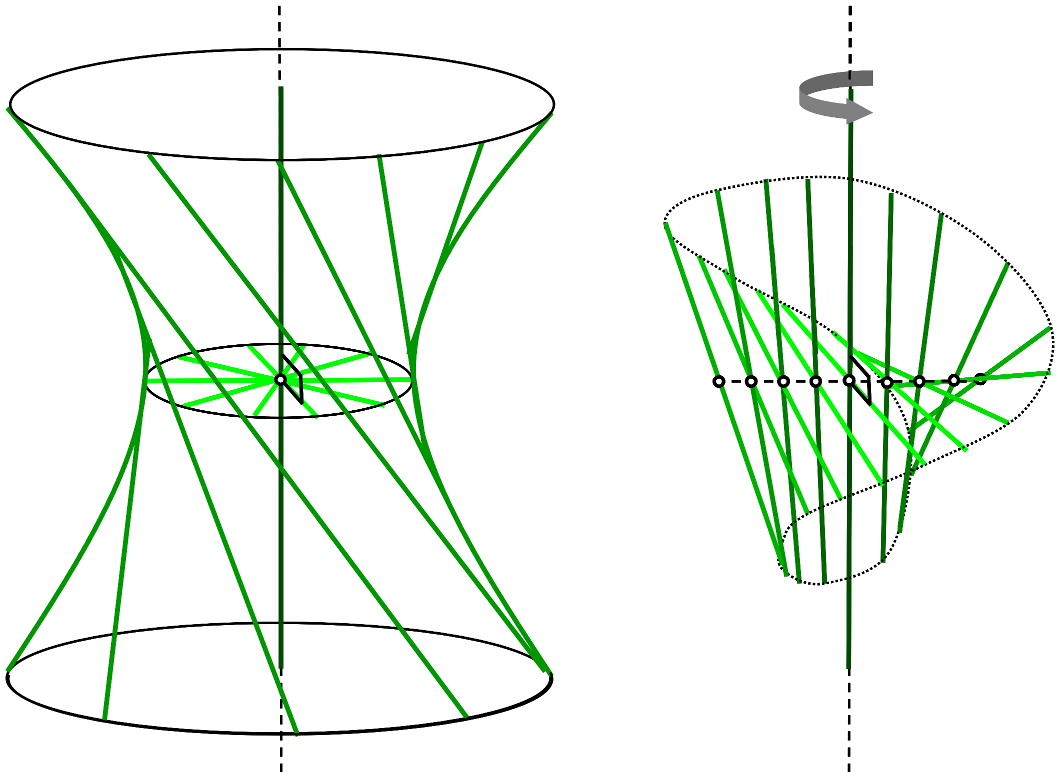

Figure 3.67: *Freedom space of 3 DOF Type 14*

The constraint space of this type is shown in Figure 3.68. The wrenches within this space may be visualized by rotating the cylindroid shown on the right side of the figure about the principle generator. The vertical dashed lines shown in Figure 3.67 and Figure 3.68 are collinear with this principle generator. The q-value of the vertical wrench is equal to the negative of the pitch value of the vertical screw from the freedom space. Wrenches lie on the surface of the same circular hyperboloid as the screws from the freedom space. The wrenches in the disk shown on the plane of the central circular cross-section of this hyperboloid all have a q-value that is equal to the negative of the pitch value of the screws in the disk of the freedom space. Each of the wrenches in the disk represents one of the principle generators of the cylindroids that make up the space.

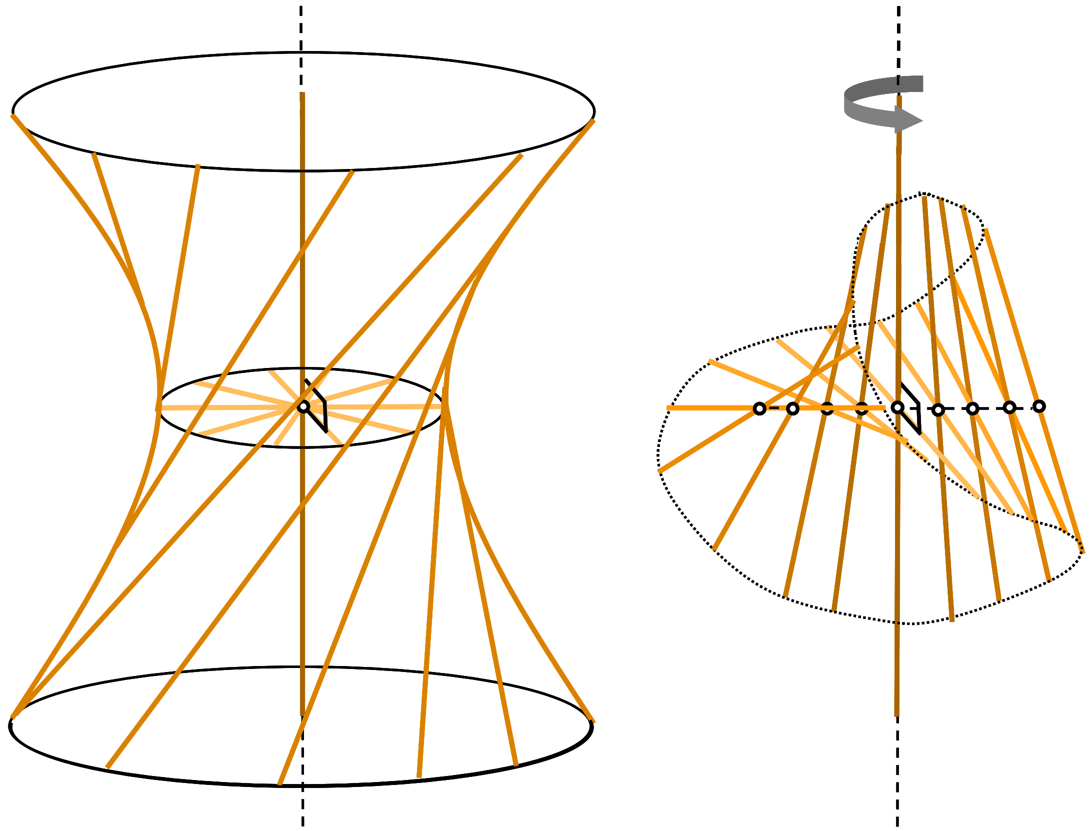

Figure 3.68: *Constraint space of 3 DOF Type 14*

3.4.15 3 DOF Type 15

The freedom space of this type is shown in Figure 3.69. It consists of an infinite number of consecutive cylindroids from the freedom space of 2 DOF Type 6. The axes of these cylindroids are parallel. The central plane in the middle of Figure 3.69 consists of one set of parallel rotation lines and a set of parallel screw lines. Both sets of lines are orthogonal to each other. Each line within each set represents a principle generator of one of the consecutive cylindroids that make up the freedom space of Figure 3.69. The sets of parallel screws on the top and bottom planes of Figure 3.69 represent the extreme generators of the consecutive cylindroids. Two different sets of parallel screws lie on the other parallel planes in between these top and bottom planes. The freedom space also consists of a pure translation that is normal to these planes.

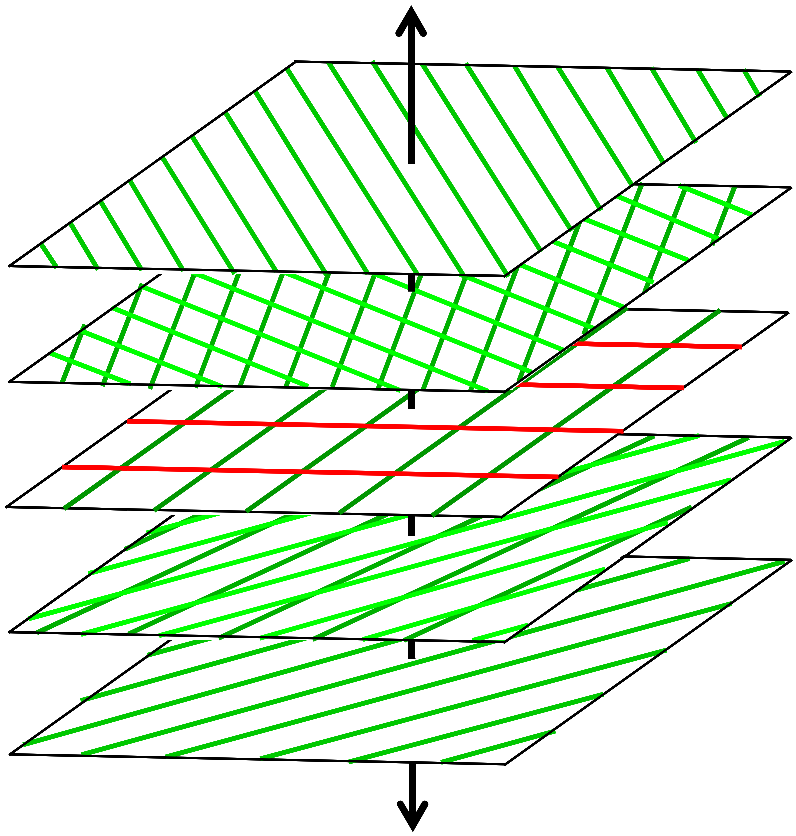

Figure 3.69: *Freedom space of 3 DOF Type 15*

The constraint space of this type is shown in Figure 3.70. If the translation, rotation lines, and screw lines of the freedom space from Figure 3.69 where replaced with a pure torque line, constraint lines, and wrench lines with q-values equal to the negative pitch values of their corresponding screw lines respectively, and all the planes above the middle plane were replaced with the planes below the middle plane and all the planes below the middle plane were replaced with the planes above the middle plane, the resulting space would represent the constraint space shown in Figure 3.70.

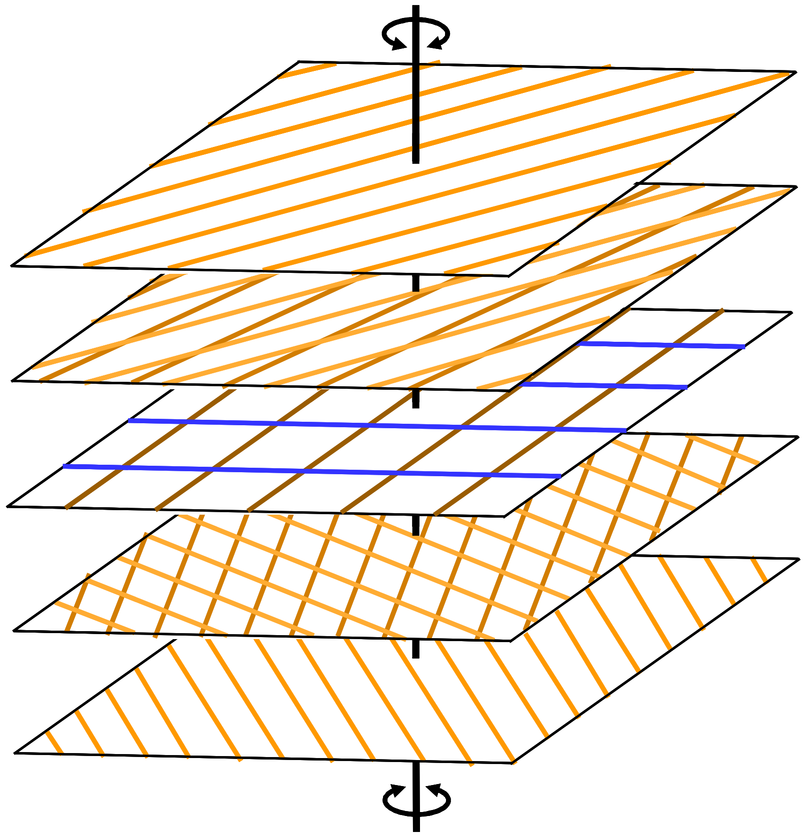

Figure 3.70: *Constraint space of 3 DOF Type 15*

3.4.16 3 DOF Type 16

The freedom space of this type is shown in Figure 3.71. It consists of an infinite number of consecutive cylindroids from the freedom space of 2 DOF Type 7.

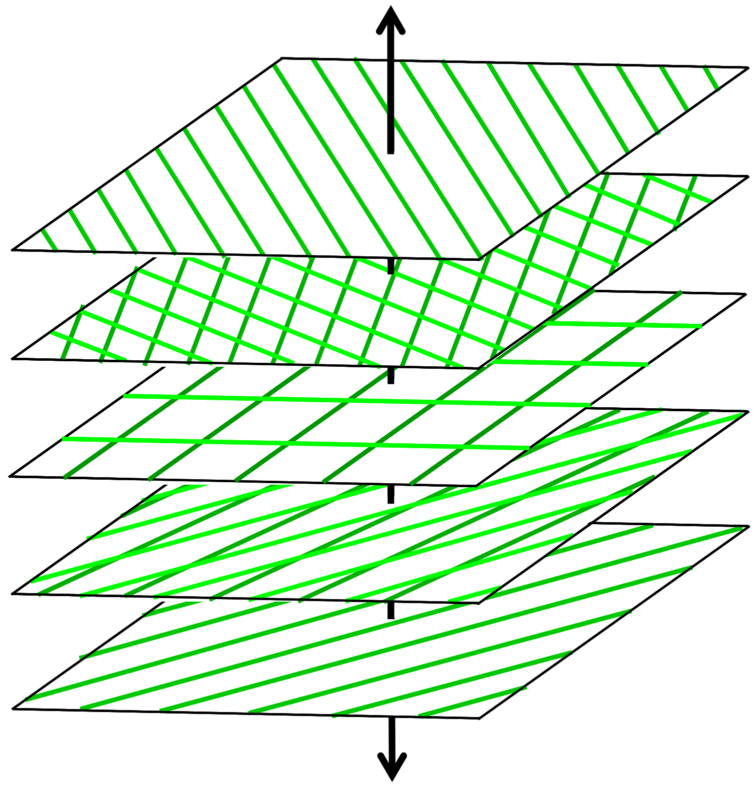

Figure 3.71: *Freedom space of 3 DOF Type 16*

The axes of these cylindroids are parallel. One of these cylindroids is shown in Figure 3.72. The central plane in the middle of Figure 3.71 consists of two sets of parallel screws. Both sets of screws have different pitch values and are orthogonal to each other. The pitch values of these sets are either all positive or all negative. Each screw within each set represents a principle generator of one of the consecutive cylindroids that make up the freedom space of Figure 3.71. The sets of parallel screws on the top and bottom planes of Figure 3.71 represent the extreme generators of the consecutive cylindroids. Two different sets of parallel screws lie on the other parallel planes in between these top and bottom planes. The freedom space also consists of a pure translation that is normal to these planes.

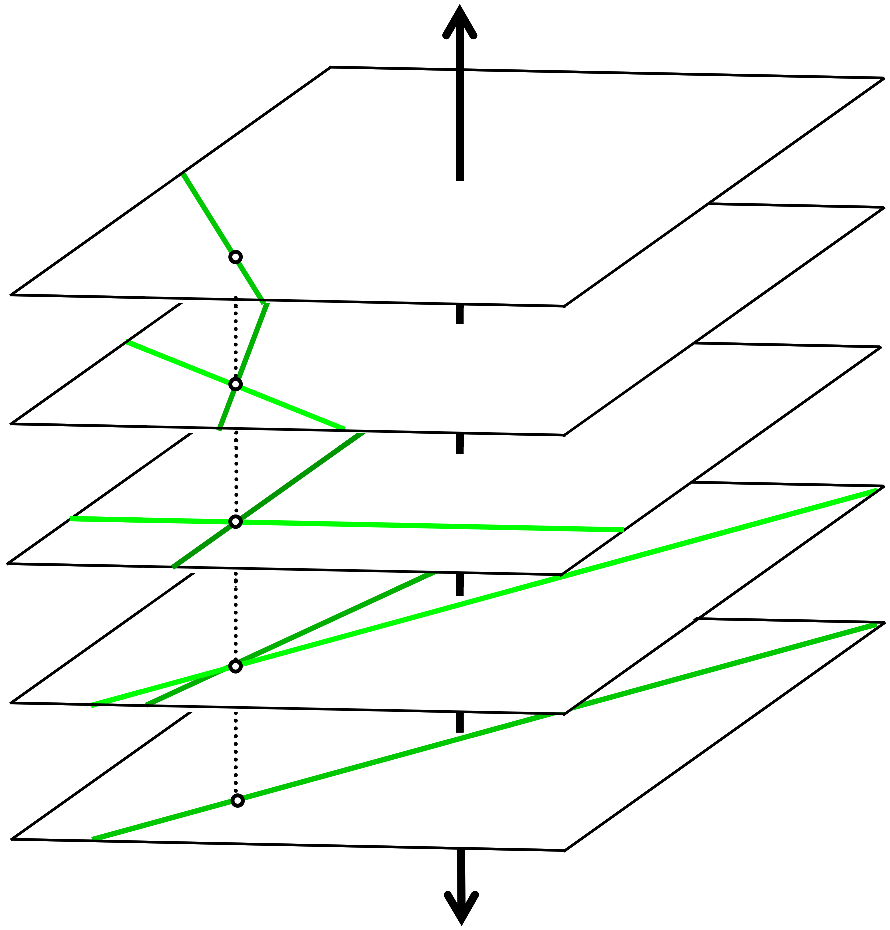

Figure 3.72: *A cylindroid from the freedom space of 3 DOF Type 16*

The constraint space of this type is shown in Figure 3.73. If the translation and screw lines of the freedom space from Figure 3.71 where replaced with a pure torque line and wrench lines with q-values equal to the negative pitch values of their corresponding screw lines respectively, and all the planes above the middle plane were replaced with the planes below the middle plane and all the planes below the middle plane were replaced with the planes above the middle plane, the resulting space would represent the constraint space shown in Figure 3.73.

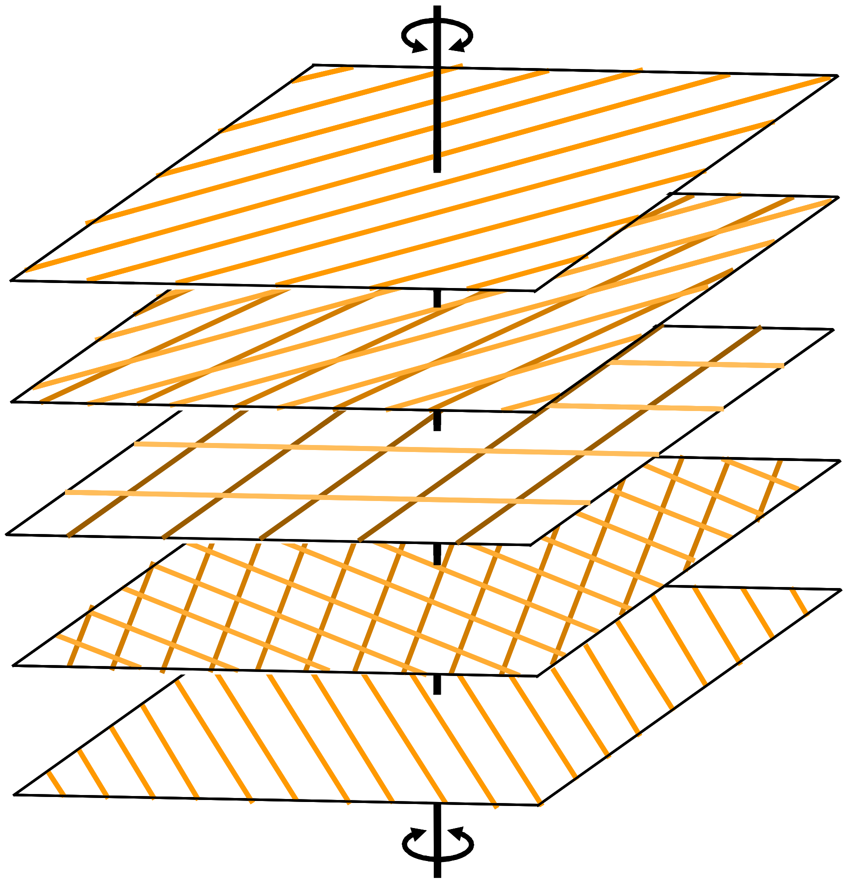

Figure 3.73: *Constraint space of 3 DOF Type 16*

3.4.17 3 DOF Type 17

The freedom space of this type is shown in Figure 3.74. It consists of all the screws with a common pitch value that lie on a plane and a single translation that points in a direction perpendicular to this plane.

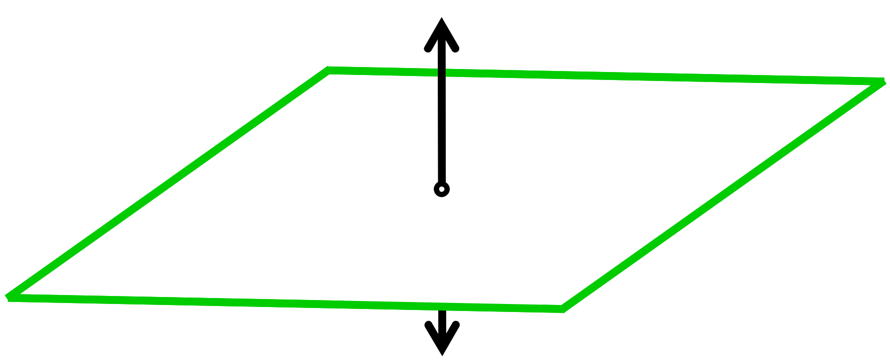

Figure 3.74: *Freedom space of 3 DOF Type 17*

The constraint space of this type is shown in Figure 3.75. It consists of all the wrenches that lie on the same plane as the screws of the freedom space and a single pure torque line that points in a direction perpendicular to this plane. The wrenches of the constraint space have a q-value equal to the negative pitch value of the screws of the freedom space.

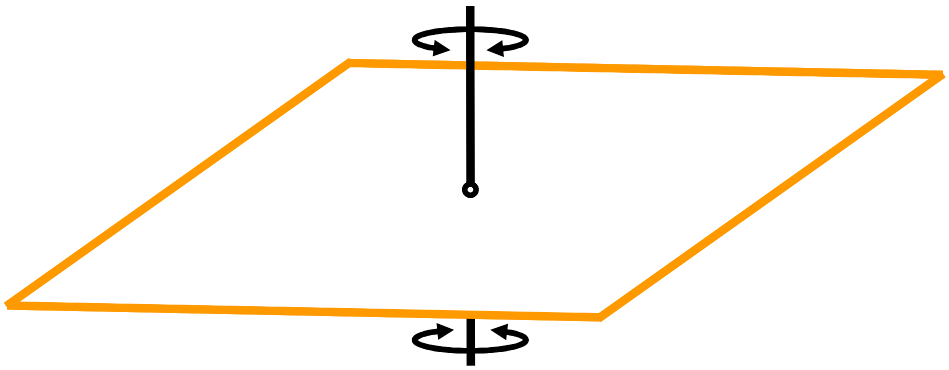

Figure 3.75: *Constraint space of 3 DOF Type 17*

3.4.18 3 DOF Type 18

The freedom space of this type is shown in Figure 3.76. It consists of all parallel screws with a common pitch value that point in a specific direction and all translations that point in directions perpendicular to these screws.

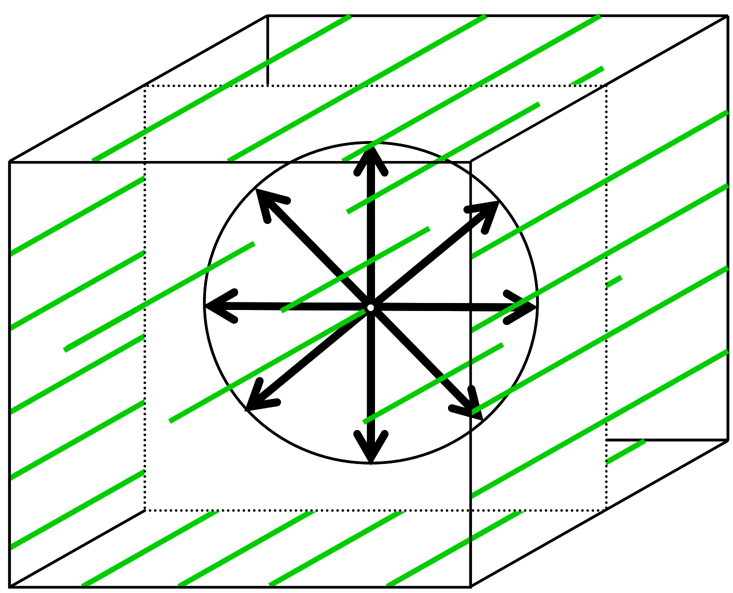

Figure 3.76: *Freedom space of 3 DOF Type 18*

The constraint space of this type is shown in Figure 3.77. It consists of all parallel wrenches that point in the same direction as the screws of the freedom space and all pure torque lines that point in directions perpendicular to these wrenches. The wrenches of the constraint space have a q-value equal to the negative pitch value of the screws of the freedom space.

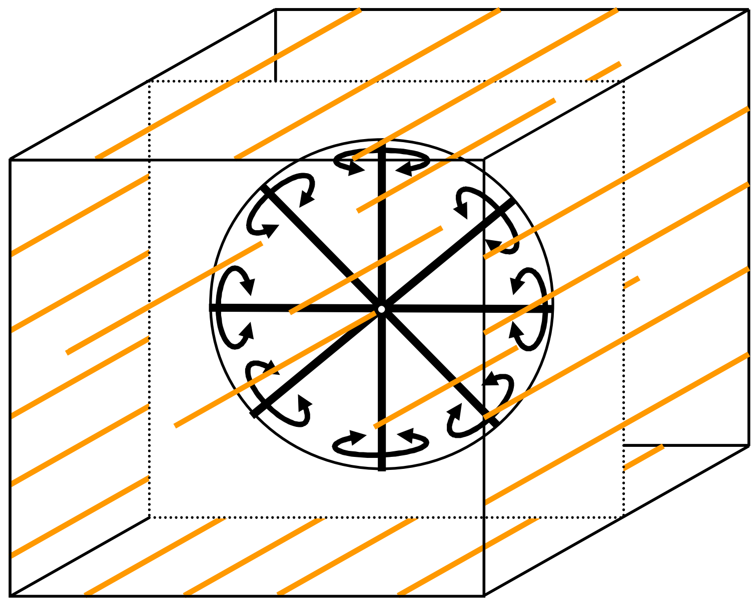

Figure 3.77: *Constraint space of 3 DOF Type 18*

3.4.19 3 DOF Type 19

The freedom space of this type is shown in Figure 3.78. It consists of all screws with a common pitch value that intersect a single point.

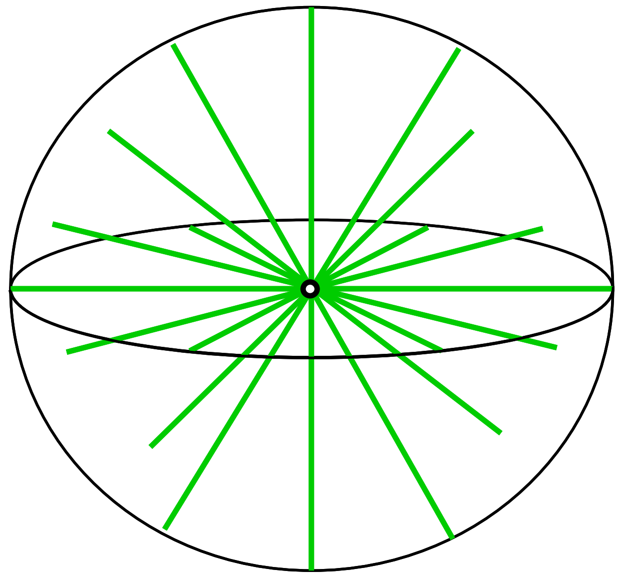

Figure 3.78: *Freedom space of 3 DOF Type 19*

The constraint space of this type is shown in Figure 3.79. It consists of all wrenches that intersect the same point as the screws of the freedom space. The wrenches of the constraint space have a q-value equal to the negative pitch value of the screws of the freedom space.

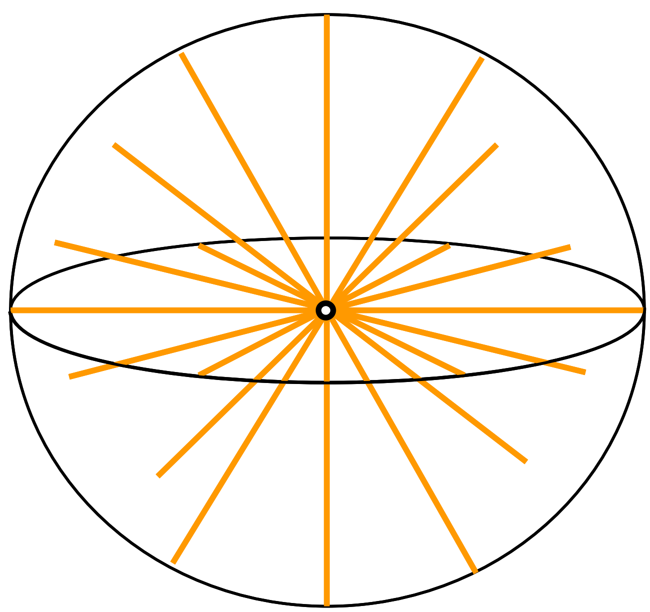

Figure 3.79: *Constraint space of 3 DOF Type 19*

3.4.20 3 DOF Type 20

The freedom space of this type is shown in Figure 3.80. It consists of translations that point in all directions.

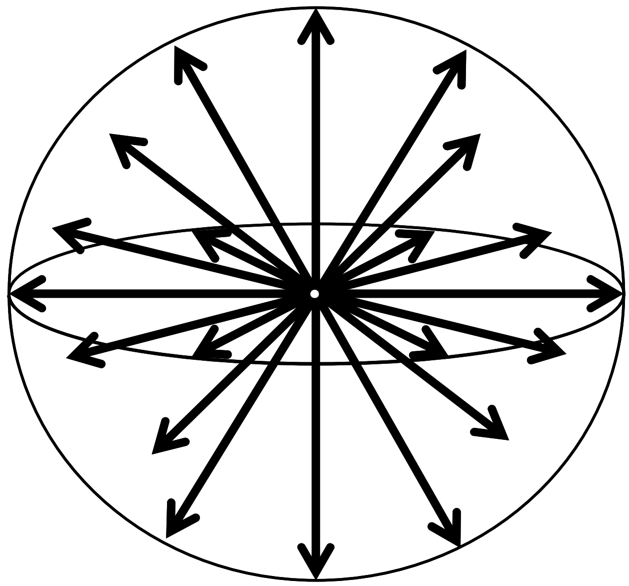

Figure 3.80: *Freedom space of 3 DOF Type 20*

The constraint space of this type is shown in Figure 3.81. It consists of pure torque lines that point in all directions.

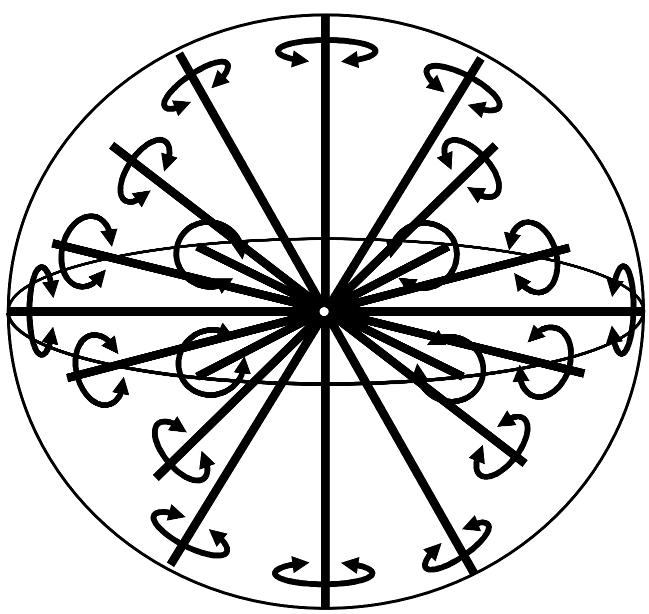

Figure 3.81: *Constraint space of 3 DOF Type 20*

3.4.21 3 DOF Type 21

The freedom space of this type is shown in Figure 3.82. It consists of all parallel rotations and screws of every pitch value that lie on a common plane. The freedom space also consists of all translations that are perpendicular to a line that lies on this plane and is perpendicular to the parallel rotations and screws.

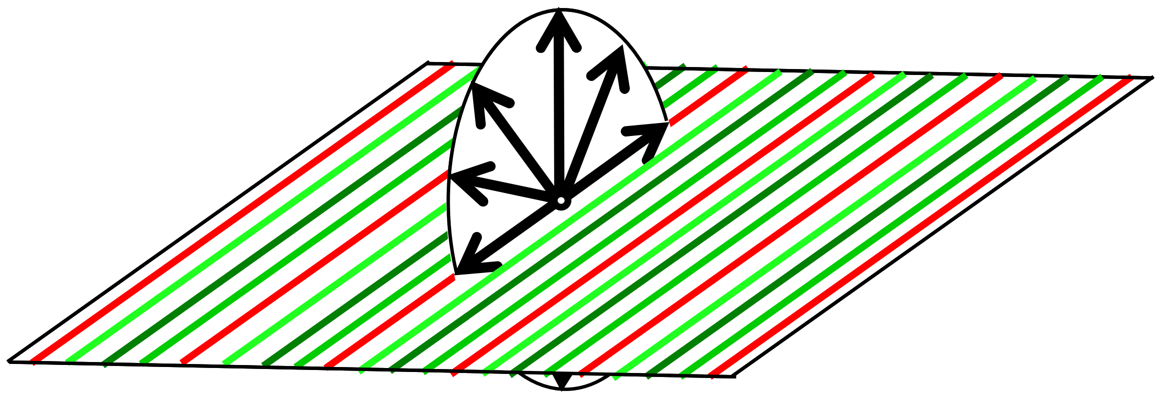

Figure 3.82: *Freedom space of 3 DOF Type 21*

The constraint space of this type is shown in Figure 3.83. It consists of all parallel constraint lines and wrenches of every q-value that lie on the same plane as the freedom space and are perpendicular to its rotations and screws. The constraint space also consists of all pure torque lines that are perpendicular to these rotations and screws.

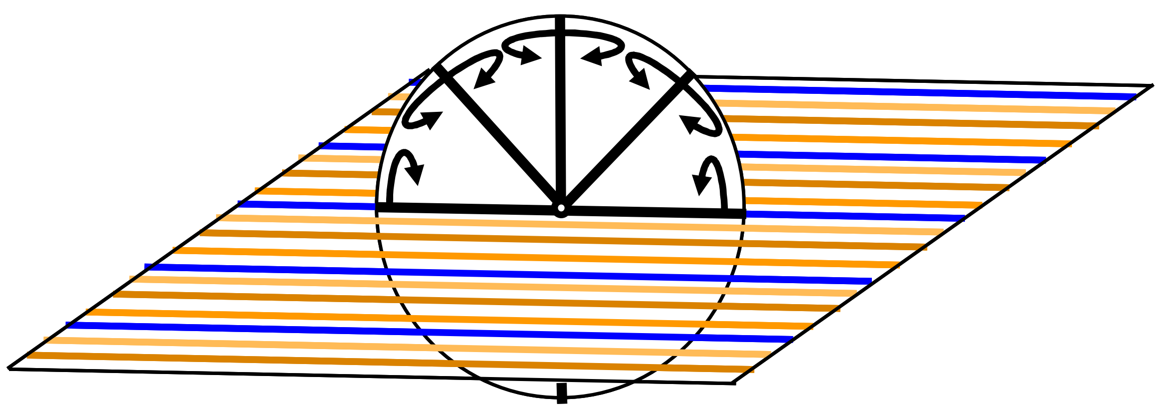

Figure 3.83: *Constraint space of 3 DOF Type 21*

3.4.22 3 DOF Type 22

The freedom space of this type is shown in Figure 3.84. It consists of a central plane of parallel rotations. The freedom space also consists of other planes that are parallel to the central plane. These planes consist of screws with common pitch values that are parallel to the rotations on the central plane. The freedom space also consists of a disk of translations as shown in Figure 3.84. One translation in the disk is normal to the parallel planes and one translation points in a direction that is an angle β from the direction of the rotation lines. The pitch p of the parallel screws on a plane a distance d from the central plane of rotations is given by Equation 3.2. The screws that lie on planes below the central plane have pitch values with an opposite sign as the pitch values of the screws that lie on the planes above the central plane.

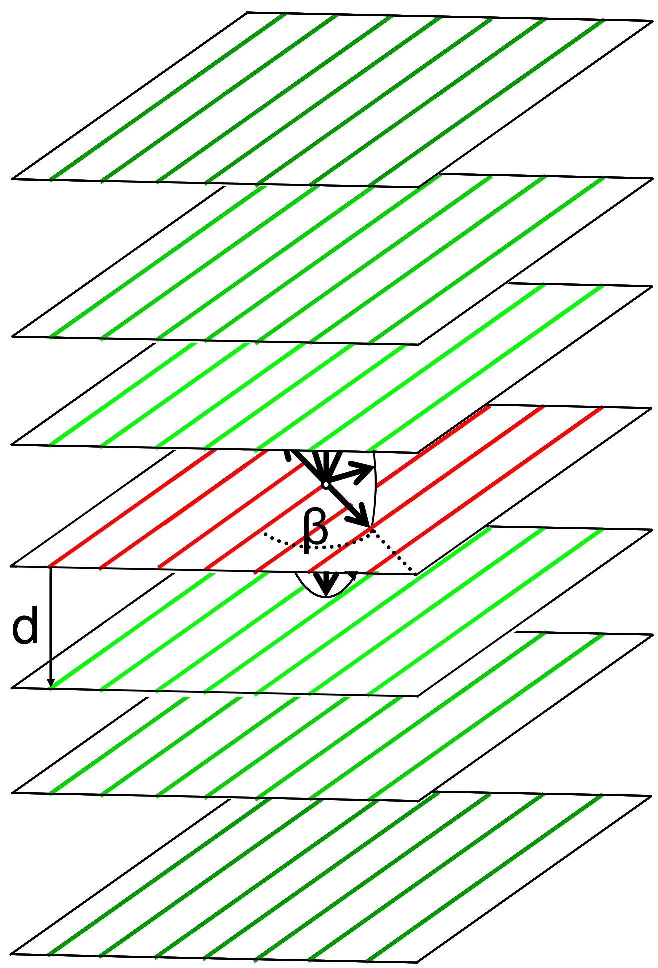

Figure 3.84: *Freedom space of 3 DOF Type 22*

The constraint space of this type is shown in Figure 3.85. It consists of a central plane of parallel constraint lines. This central plane is coplanar with the central plane of the freedom space. The parallel constraint lines are perpendicular to the translations of the freedom space. The constraint space also consists of other planes that are parallel to the central plane. These planes consist of wrenches with common q-values that are parallel to the constraint lines on the central plane. The q-value of any such set of parallel wrenches is equal to the negative pitch value of the parallel screws that lie on the same plane from the freedom space. The constraint space also consists of a disk of pure torque lines as shown in Figure 3.85. These pure torque lines are perpendicular to the rotation and screw lines from the freedom space. The wrenches that lie on planes below the central plane have q-values with an opposite sign as the q-values of the wrenches that lie on the planes above the central plane.

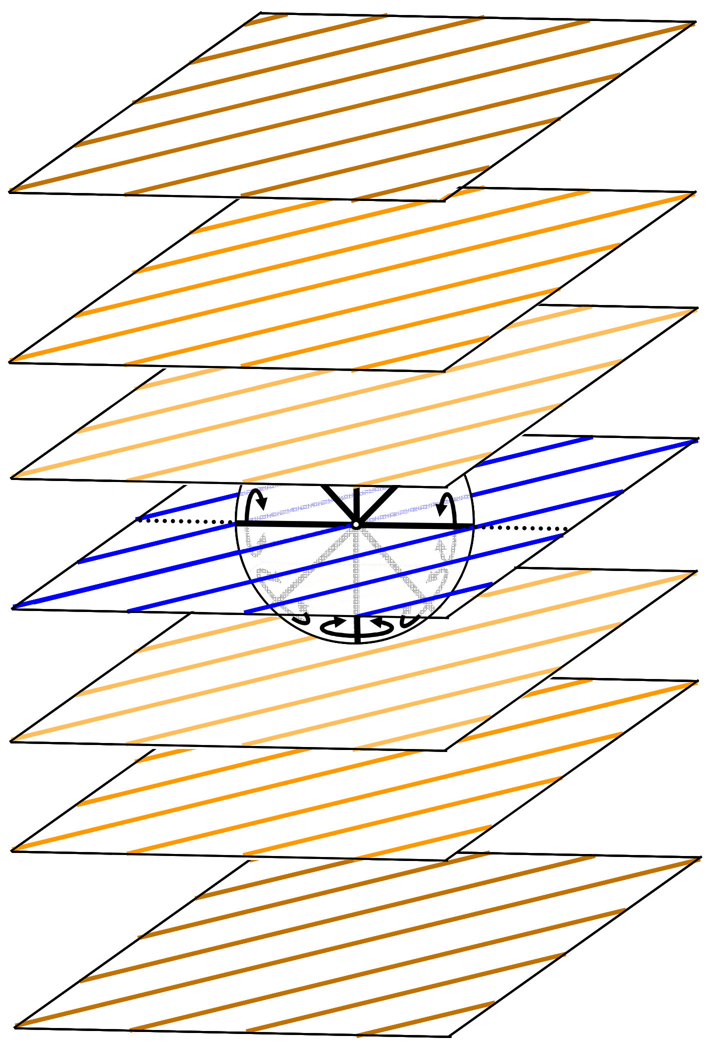

Figure 3.85: *Constraint space of 3 DOF Type 22*

3.5 Types within the 4, 5, and 6 DOF Column

If constraint lines are changed to rotation lines, wrenches are changed to screws, and pure torque lines are changed to translations, the freedom spaces of every type within the 4 DOF column from Figure 2.1 are identical to the constraint spaces within the 2 DOF column from the same figure. Likewise, if rotation lines are changed to constraint lines, screws are changed to wrenches, and translations are changed to pure torque lines, the constraint spaces of every type within the 4 DOF column from Figure 2.1 are identical to the freedom spaces within the 2 DOF column from the same figure. The geometry of these spaces will, therefore, not be described in this section because they have already been described in a pervious section.

Similarly, the freedom spaces of every type within the 5 DOF column from Figure 2.1 are identical to the constraint spaces within the 1 DOF column and the constraint spaces of every type within the 5 DOF column from Figure 2.1 are identical to the freedom spaces within the 1 DOF column. The freedom space within the 6 DOF column from Figure 2.1 is also identical to the constraint space within the 0 DOF column and the constraint space within the 6 DOF column from Figure 2.1 is identical to the freedom space within the 0 DOF column. The geometry of these spaces will also not be described in this section because they have already been described in pervious sections.

3.6 Practical FACT Chart

The geometries of the shapes from the practical FACT chart from Figure 2.2 are identical to the geometries of the shapes from the mathematical FACT chart from Figure 2.1 that were described in detail in this chapter. The freedom spaces of both charts remain unchanged. The only difference between these charts is that the constraint spaces that lie outside the parallel pyramid are eliminated for the chart of Figure 2.2 and the wrenches and pure torque lines within every constraint space that lies inside the parallel pyramid are also eliminated leaving only the constraint lines.

CHAPTER 4:
Optimal Actuator Placement

In this chapter a visual approach is introduced that enables the placement of actuators within multi-axis parallel flexure systems such that position and orientation errors are minimized. A stiffness matrix, which links twists and wrenches, is used to generate geometric shapes that guide designers in selecting optimal actuator locations and orientations. The geometric shapes, called actuation spaces, enable designers to *(i)* visualize the regions wherein actuators should be placed so as to minimize errors, *(ii)* guide designers in selecting these actuators to maximize the decoupling of actuator inputs, and *(iii)* determine actuator forces and displacements for actuating specific degrees of freedom. These new principles, the means to practice them, and a comparison of theory verses measured behavior, are demonstrated within a case study.

4.1 Technical Scope

The contents of this chapter apply to *(i)* parallel flexure systems wherein a single rigid stage is connected directly to ground via its flexure elements, *(ii)* small motion flexure systems, *(iii)* systems that use linear actuators (i.e. actuators capable only of pushing/pulling along their axes) and *(iv)* quasi-static actuation scenarios where the dynamic behavior associated with stage inertia may be ignored.

4.2 Fundamental Principles

This section reviews the principles that *(i)* mathematically link stage motions and actuation forces, *(ii)* enable the generation of actuation spaces and *(iii)* demonstrate how to use shapes in practical flexure design. Hand-based calculation would require excessive time; therefore the modeling and process have been built into a MATLAB script. This script constructs any parallel flexure system's twist-wrench stiffness matrix (TWSM), and generates that system's actuation space. A copy of the script is found in Appendix B.

4.2.1 Linking Motions to Actuation Forces

The goal of this section is to link twists (motions) to the wrenches (actuation forces) that cause these motions. This type of linking is fundamentally a type of stiffness; therefore the twist-wrench stiffness matrix (TWSM) is introduced. This matrix is different from the conventional stiffness matrix in that it links twist vectors to wrench vectors rather than displacement vectors to force vectors. The TWSM, $[K_{TW}]$, provides the link between the desired motion, $\mathbf{T_i}$, and the unique actuation force (or effective actuation force), $\mathbf{W_i}$, that actuates the motion via:

$$\mathbf{W_i} = \left[K_{TW}\right] \cdot \mathbf{T_i}\,, \tag{4.1}$$

A system's TWSM contains information pertaining to the orientation, location, geometry and material properties of each flexible constraint. The details of the general TWSM's derivation and structure are provided in Section 4.5 and the TWSM's use is demonstrated in Section 4.6.

Note that the motion $\mathbf{T_i}$ from Equation 4.1 does not need to be a system DOF. In this chapter, however, it is desirable to identify the wrenches that actuate the system DOFs because these are the wrenches that determine the actuator locations and orientations that cause the stage to move with its intended motions (i.e. the motions that produce minimal stage position and orientation errors). Equation 4.1, therefore, is paramount to this chapter because the TWSM of a flexure system is the means through which the actuators are optimally placed. The actuation force shown in Figure 4.1, for example, was determined by multiplying the twist vector of the desired rotation line by the flexure system's TWSM according to Equation 4.1. The resulting wrench vector was then decomposed to determine the location and orientation of its force. In Section 4.6, the TWSM of this flexure system is calculated.

Others have developed specialized, compliance matrices that link wrenches to twists [69-71]. The TWSM introduced in this chapter, however, is suited for the generation of geometric shapes that may be used to guide designers in placing actuators for all parallel flexure systems. The generation and use of these shapes are a major contribution of this thesis.

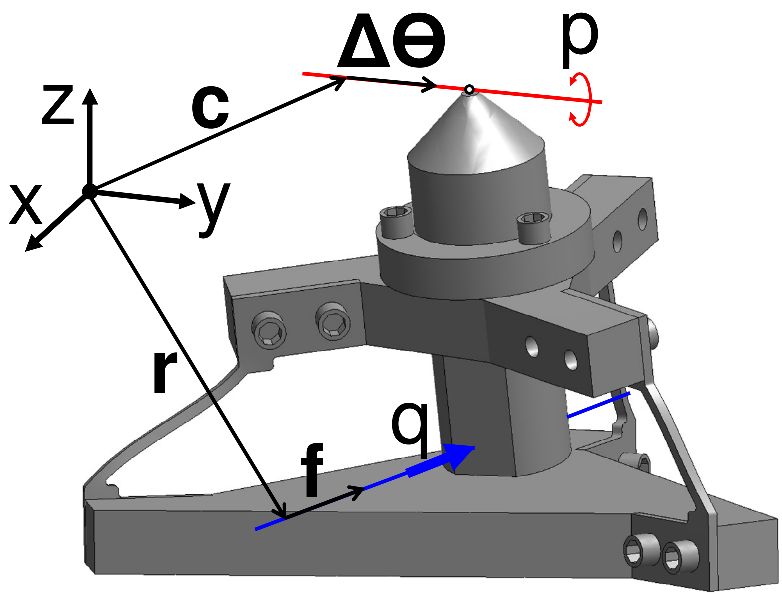

Figure 4.1: *Parameters defined for a desired twist (red line with a circular arrow) that is optimally actuated by a wrench (blue arrow).*

4.2.2 Geometric Shapes as Design Tools

Consider the 3-DOF parallel flexure system from Figure 4.1. It is capable of nominally guiding three independent rotation DOFs that intersect at the probe's tip as shown in Figure 4.2A. If the three independent rotations are simultaneously actuated and the relative ratio of their rotations is controlled, the resulting rotation line may be any of the lines within the sphere in Figure 4.2B. This sphere is the system's freedom space. Freedom space is the geometric shape that visually represents a system's kinematics, e.g. all the twists the flexure permits. The freedom space of a system is modeled using a twist matrix, $[T]$, defined as

$$[T]=[\mathbf{T}_1 \quad \mathbf{T}_2 \quad \cdots \quad \mathbf{T}_n], \tag{4.2}$$

where n is the number of independent twists or DOFs the system possesses.

Suppose now it is desirable to identify the unique actuator locations and orientations for the DOFs shown in Figure 4.2A. From Equation 4.1 it is evident that

$$[W]=[\mathbf{W}_1 \quad \mathbf{W}_2 \quad \cdots \quad \mathbf{W}_n]=[K_{TW}]\cdot[T], \tag{4.3}$$

where $[W]$ is a wrench matrix that contains n independent wrench vectors that each represent the unique actuator locations and orientations desired. For this example, the three actuation wrenches (n=3) are shown in Figure 4.2A. Their corresponding twists may be identified by like subscript. $\mathbf{W_1}$ and $\mathbf{W_2}$ are pure force wrenches whereas $\mathbf{W_3}$ is a pure moment. If one combines these wrenches, the result is pure force wrenches that lie on the plane shown in Figure 4.2B. This plane is the system's actuation space. An actuation space is a shape that represents the actuator locations and orientations that must be used to minimize parasitic errors when the number of actuators is to equal the number of DOFs to be actuated. A system's actuation space is a graphical representation of the wrench matrix from Equation 4.3. For practical purposes, the allowable wrenches have been limited to be pure forces, therefore $\mathbf{W_3}$ must be emulated using a linear force actuator. In this example $\mathbf{W_3}$ could be replaced by a linear force actuator that lies on the plane with a line of action far removed from the axis of $\mathbf{T_3}$.

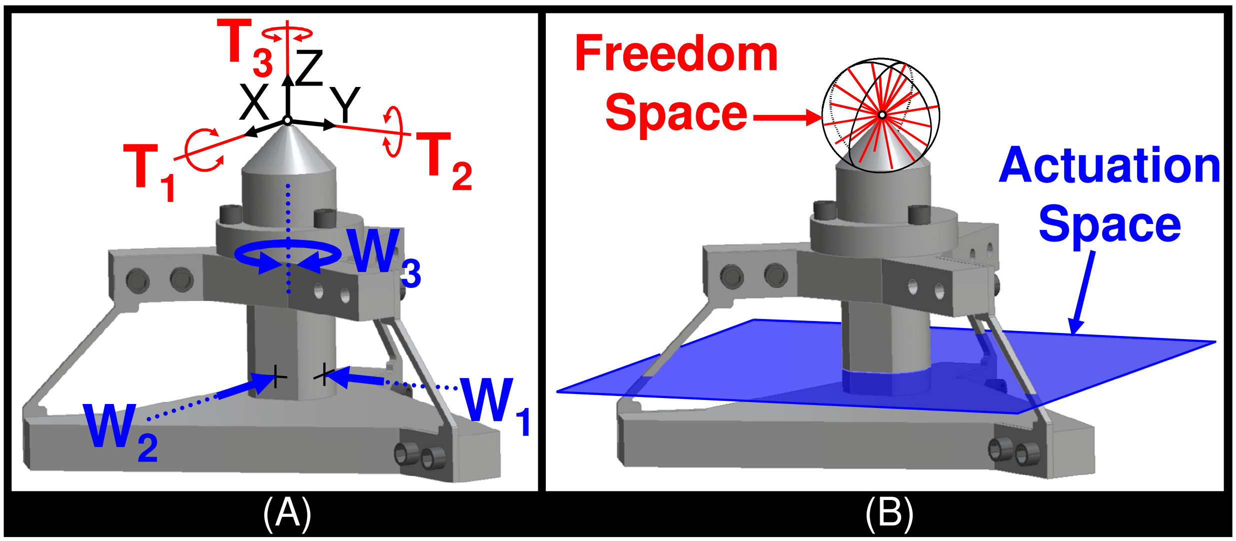

Figure 4.2: *Three independent DOFs represented as twists,* $\mathbf{T_i}$, *and their corresponding actuator placements represented as wrenches,* $\mathbf{W_i}$ *(A). Freedom and actuation spaces of the flexure system (B).*

It is possible to use multiple actuators that don't belong to the system's actuation space if the actuators work in coordination to emulate the behavior of an actuator that does belong to that actuation space. As such, not all actuators must lie within the system's actuation space unless the designer wishes to use the minimum number of actuators (i.e. equal to the number of DOFs). Consider, for instance, two linear actuators that lie in a plane parallel to the actuation space plane. If their outputs were coupled such that they always applied forces in opposite directions with equal magnitudes, they would successfully actuate the rotation, $\mathbf{T_3}$, with minimal error. In essence coupled actuators not in the actuation space do the job of a single actuator in that space with much more complexity and cost.

4.3 Actuation Space

In this section examples of actuation spaces are provided for various multi-DOF flexure systems. A discussion is provided that explains how actuation space may be used as a design tool for modifying a system such that it may be actuated in a practical way.

4.3.1 Examples of Actuation Space

Consider the flexure system in Figure 4.3A. This system may be used to achieve parallelism between two plates. The flexure constraints' lines of action all lie on the same plane, but do not intersect at the same point. This arrangement permits the circular plate to rotate about the x and y axes and translate in the z direction. The linear combination of the system's DOFs may be represented by a planar freedom space of rotation lines, i.e. zero-pitch twists, and an arrow along the z axis which describes a translation, i.e. infinite-pitch twist. Equation 4.3 may be used to show that the actuation space for this system is a box of infinite extent that contains every line that is perpendicular to the plane of the flexure as shown in Figure 4.3B.

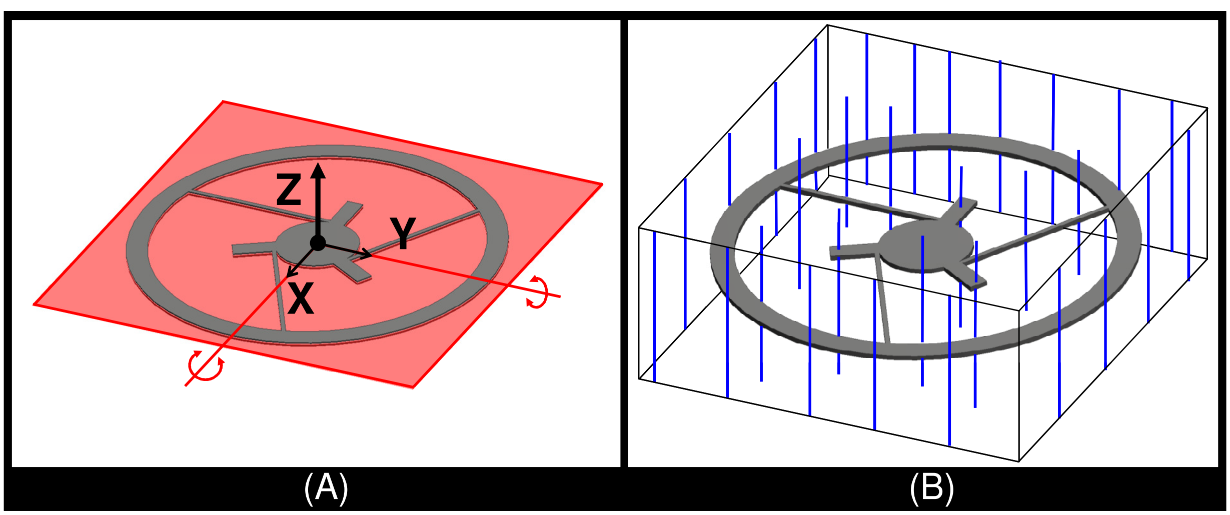

Figure 4.3: *A flexure system with a planar freedom space (A) and a box-shaped actuation space of parallel lines (B).*

Consider the two-DOF flexure system shown in Figure 4.4. Its constraints permit the stage to translate in the direction of the arrow and rotate about the line labeled $\mathbf{T_1}$ and $\mathbf{T_2}$ respectively as shown in Figure 4.4A. The linear combination of these two twists results in a freedom space that contains every parallel rotation line in the plane shown. Equation 4.3 may be used to determine that this system's actuation space is a disk or pencil of pure force lines that lie on the plane shown in Figure 4.4B. Note that the wrench labeled $\mathbf{W_1}$ corresponds with the twist labeled $\mathbf{T_1}$, and the wrench labeled $\mathbf{W_2}$ corresponds with the twist labeled $\mathbf{T_2}$. Each of the other pure force wrenches in the disk corresponds to a unique rotational twist on the plane of parallel lines in the freedom space. The distance from the disk to the stage may be determined using the theory presented in this thesis. The numerical example in Section 4.6 provides the details for finding this distance.

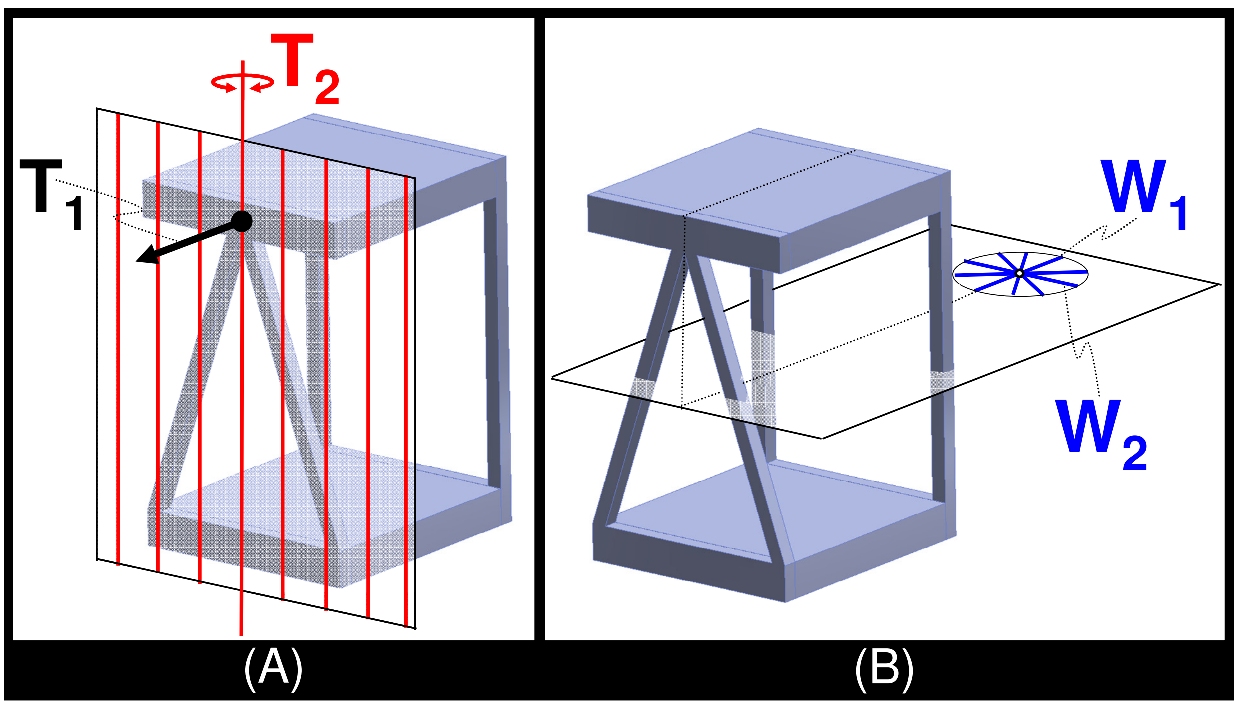

Figure 4.4: *A flexure system with a planar freedom space of parallel rotation lines (A) and a disk-shaped actuation space (B).*

4.3.2 Actuation Space as a Design Tool

This section explains how actuation spaces may be used to compare different flexure concepts and ascertain which concept is easiest to actuate. Figure 4.5 provides the twenty-six types of actuation spaces that exist for every flexure system. These spaces are divided between 6 cases, where the case number represents the number of independent linear actuators and the type represents a particular arrangement of those actuators. The most practical and useful actuation spaces in precision engineering are marked with "●". Others may be useful in certain applications; however they are primarily included here for completeness. Note that the actuation spaces are identical to the constraint spaces within the parallel pyramid shown in Figure 2.2. A description of their geometry is, therefore, found in Chapter 3. The derivation of these spaces is provided in Hopkins [21].

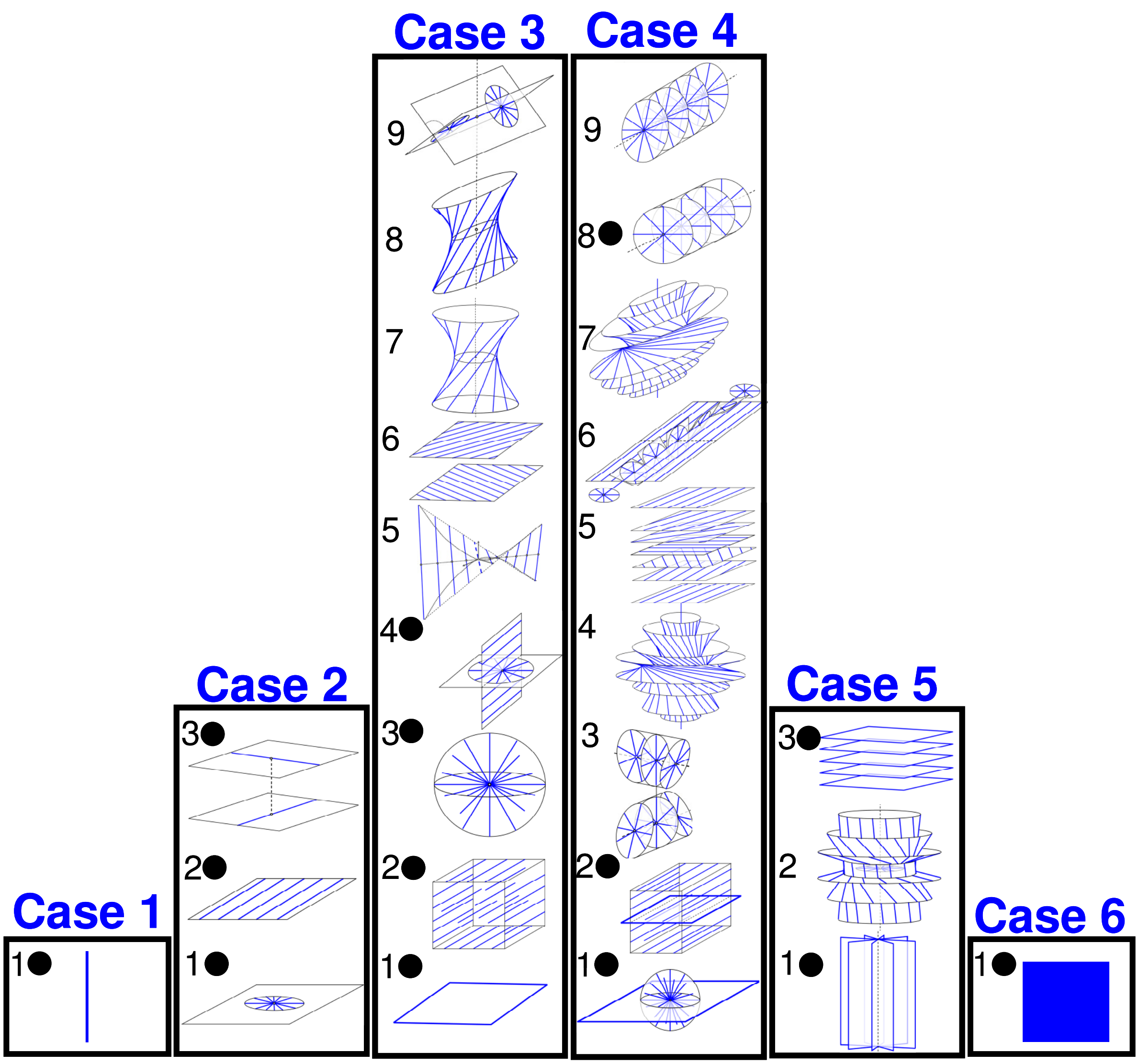

Figure 4.5: *The twenty-six actuation spaces that pertain to systems actuated by linear actuators*

All flexure systems that may be fully actuated with linear actuators will possess one of the twenty-six actuation spaces from Figure 4.5. If Equation 4.3 generates an actuation space that is not found in Figure 4.5, the designer knows that this system cannot be actuated with linear actuators alone. The actuation spaces for the examples of this chapter are found in Figure 4.5. For Figure 4.2B see Case 3 Type 1, for Figure 4.3B see Case 3 Type 2 and for Figure 4.4B see Case 2 Type 1.

A powerful facet of this approach is that it provides a means to rapidly understand how difficult it may be to actuate a particular flexure concept. The chart in Figure 4.5 contains a complete set of actuation spaces as prescribed by mathematics; however some of these shapes are not practical. For example, Case 4 Type 7 in Figure 4.5 would require actuators that are arranged with lines of actuation upon several nested elliptical hyperboloids. It is therefore easy to know that minimizing parasitic errors in this design would be difficult.

A second powerful aspect of this approach is that it enables a designer to understand where actuators need to be placed relative to the flexure and other elements of a machine. Consider, for example, the design in Figure 4.4B. Its actuation space is a disk or pencil of lines, but none of the lines pass through the stage. A better design may thus require a larger stage or a mechanical extension to connect the actuators to the existing stage.

4.4 Actuator Placement and Outputs

Given an actuation space, a designer must:

(i) Select the minimum number of actuators – Typically equal to the number of DOFs.

(ii) Place and orient actuators – It is necessary to align their lines of action to lie along lines within the actuation space and intersect the stage, or an extension of the stage. This intersection should be selected such that actuators may be attached in a practical way. For example, orthogonal intersections of actuation lines and flexure stages are easy to realize while obtuse/acute angles may be more problematic.

(iii) Ensure independent actuators – If the actuators' lines of action do not yield wrenches that are independent, it will not be possible to control all of the flexure's DOFs. Gaussian elimination is a common approach for ensuring wrench independence. A comprehensive list of qualitative "rules of thumb" exist for guiding designers in selecting independent actuators from

any of the twenty-six actuation spaces from Figure 4.5. These rules are embodied by shapes called sub-constraint spaces and are found in Hopkins [21].

Instructions *(i)* – *(iii)* will be applied in the following example. Reconsider the stage in Figure 4.2B. This system possesses three DOFs, therefore three independent linear actuators must be selected and placed/oriented coincident with lines that lie on the plane in Figure 4.2B. The lines of action are arranged, as shown in Figure 4.6A, to provide a measure of symmetry. These lines of action are independent as they are not parallel and do not intersect at a common point. The stage has been extended downward to intersect the actuation space and features have been added to the extension to facilitate orthogonal attachment of the actuators.

4.4.1 Force-base Actuation

In this scenario, it is desirable to link actuator forces to stage displacements required to cause motion, **T**, as shown in Figure 4.6B. We first define an actuation matrix, $[W_A]$, that consists of n independent wrenches, $\mathbf{W_{An}}$, which each represents an actuator placement from within the system's actuation space:

$$[W_A] = [\mathbf{W_{A1}} \quad \mathbf{W_{A2}} \quad \cdots \quad \mathbf{W_{An}}]. \tag{4.4}$$

The $\mathbf{f_{An}}$ vectors – defined in Equation 1.3 – within the wrenches in Equation 4.4 must be unit vectors. For this example n=3. The magnitude of each force, F_{An}, required to actuate **T** is:

$$[F_{A1} \quad F_{A2} \quad \cdots \quad F_{An}]^T = \left([W_A]^T \cdot \quad [W_A]\right)^{-1} \cdot [W_A]^T \cdot [K_{TW}] \cdot \mathbf{T}, \tag{4.5}$$

where $[K_{TW}]$ is the system's twist-wrench stiffness matrix. For this example, the relative ratios of force magnitudes, F_{A1}:F_{A2}:F_{A3}, should be -2:1:1 as shown in Figure 4.6B. This result may be generated automatically by the MATLAB tool in Appendix B.

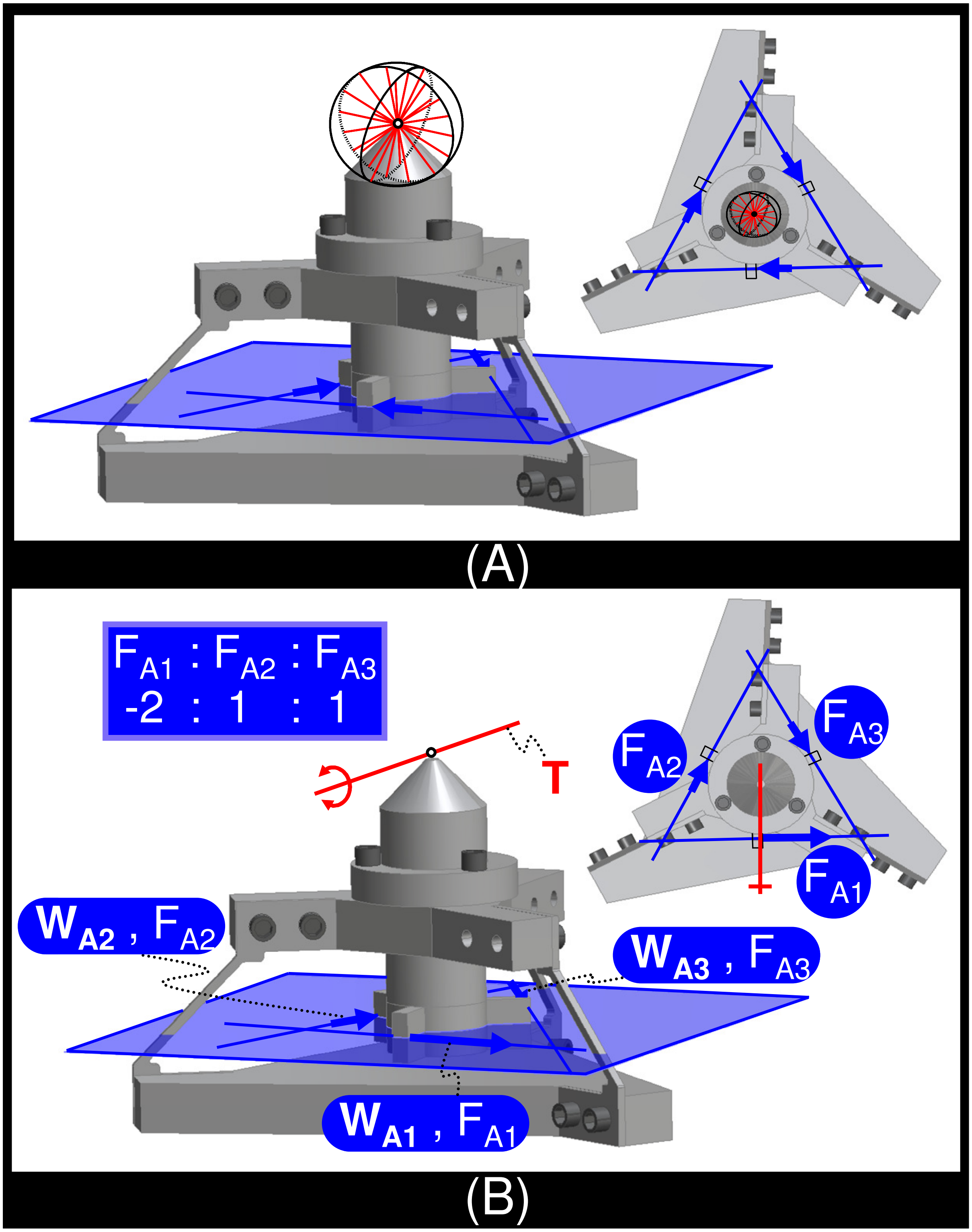

Figure 4.6: *A triangular actuation scheme for accessing every rotation in the sphere with minimal error (A), and the corresponding actuator force ratio for accessing the particular rotation,* ***T*** *(B).*

4.4.2 Displacement Actuation

In this scenario, it is desirable to link actuator displacements to the stage displacements that are required to cause motion, **T**. The motion, $\mathbf{T_{An}}$, is first identified. This motion is caused by each individual actuator wrench, $\mathbf{W_{An}}$, with force magnitudes of F_{An} given by

$$\mathbf{T_{An}} = [K_{TW}]^{-1} \cdot \mathbf{W_{An}} \cdot F_{An} \tag{4.6}$$

This twist's orientation vector, $\mathbf{\Delta\Theta_{An}}$, location vector, $\mathbf{c_{An}}$, and pitch, p_{An}, may be obtained via decomposition and used to define the displacement magnitude, δ_{An}, of each actuator wrench, $\mathbf{W_{An}}$, as:

$$\delta_{An} = \left(((\mathbf{c_{An}} - \mathbf{r_{An}}) \times \mathbf{\Delta\theta_{An}}) + p_{An} \cdot \mathbf{\Delta\theta_{An}}\right) \bullet \mathbf{f_{An}}, \tag{4.7}$$

where $\mathbf{r_{An}}$ is a location vector that points from the origin of the stage's coordinate system to the location at which actuator, $\mathbf{W_{An}}$, attaches to the stage, and $\mathbf{f_{An}}$ is a unit vector that points along this actuator's line of action. For this example, the relative ratios of displacement magnitudes, δ_{A1}:δ_{A2}:δ_{A3}, should also be -2:1:1 since the stiffness associated with pushing/pulling at each actuation site is the same due to symmetry. The MATLAB tool in Appendix B may be used to generate this result.

4.5 Twist-Wrench Stiffness Matrix

4.5.1 Defining the Twist-Wrench Stiffness Matrix

As shown in Equation 4.1, a TWSM relates an actuation force, **W**, to the motion, **T**, that it causes. This matrix is defined as:

$$[K_{TW}]_{6x6} = \sum_{a=1}^{C} \left[N_R^{(a)}\right] \cdot \left[S^{(a)}\right] \cdot \left[N^{(a)}\right]^{-1}, \tag{4.8}$$

where C is the number of constraints the parallel flexure system possesses and

$$\left[N_R^{(a)}\right]_{6x6} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{n_1} & \mathbf{n_2} & \mathbf{n_3} \\ \mathbf{n_1} & \mathbf{n_2} & \mathbf{n_3} & \mathbf{L}\times\mathbf{n_1} & \mathbf{L}\times\mathbf{n_2} & \mathbf{L}\times\mathbf{n_3} \end{bmatrix}, \tag{4.9}$$

$$\left[N^{(a)}\right]_{6x6} = \begin{bmatrix} \mathbf{n_1} & \mathbf{n_2} & \mathbf{n_3} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{L}\times\mathbf{n_1} & \mathbf{L}\times\mathbf{n_2} & \mathbf{L}\times\mathbf{n_3} & \mathbf{n_1} & \mathbf{n_2} & \mathbf{n_3} \end{bmatrix}. \tag{4.10}$$

$$\left[S^{(a)}\right]_{6x6} = \begin{bmatrix} \frac{l}{EI_1} & 0 & 0 & 0 & -\frac{l^2}{2EI_1} & 0 \\ 0 & \frac{l}{EI_2} & 0 & \frac{l^2}{2EI_2} & 0 & 0 \\ 0 & 0 & \frac{l}{GJ} & 0 & 0 & 0 \\ 0 & \frac{l^2}{2EI_2} & 0 & \frac{l^3}{3EI_2} & 0 & 0 \\ -\frac{l^2}{2EI_1} & 0 & 0 & 0 & \frac{l^3}{3EI_1} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{l}{EA} \end{bmatrix}^{-1}, \tag{4.11}$$

Equations 4.9 and 4.10 describe constraint *(a)*'s location and orientation, and Equation 4.11 describes constraint *(a)*'s stiffness characteristics. The variables within Equation 4.9 through Equation 4.11 are defined in the following text and with reference to Figure 4.7. Here a blade flexure is used as an example, however these equations apply in general to any flexible constraint.

The vector, **0**, is a 3x1 vector of zeros. The orthogonal 3x1 unit vectors, $\mathbf{n_1}$, $\mathbf{n_2}$, and $\mathbf{n_3}$, are oriented along constraint *(a)*'s planes of symmetry as shown in Figure 4.7A. The 3x1 location vector, **L**, points from the origin of the stage's coordinate system to the central point where constraint *(a)* attaches to the stage. The inverse of the matrix from Equation 4.11 [22, 72] relates the reaction moments and forces, Γ_j and f_j, that the constraint applies to the stage due to its rotations and translations, $\Delta\Theta_j$ and $\Delta\delta_j$, as shown in Figure 4.7B according to

$$\left[\Delta\theta_1 \quad \Delta\theta_2 \quad \Delta\theta_3 \quad \Delta\delta_1 \quad \Delta\delta_2 \quad \Delta\delta_3\right]^T = \left[S^{(a)}\right]_{6x6}^{-1} \cdot \left[\Gamma_1 \quad \Gamma_2 \quad \Gamma_3 \quad f_1 \quad f_2 \quad f_3\right]^T. \tag{4.12}$$

The constraint has a modulus of elasticity, E, a shear modulus, G, a bending moment of inertia about the $\mathbf{n_1}$ axis, I_1, and about the $\mathbf{n_2}$ axis, I_2, a polar moment of inertia, J, a cross-sectional area, A, and a length of l.

Furthermore, it is interesting to note that the six eigenvectors of the TWSM of Equation 4.8 correspond to the only twist and actuation wrench pairs that are collinear. Their eigen values are the corresponding scalar stiffness values of each pair.

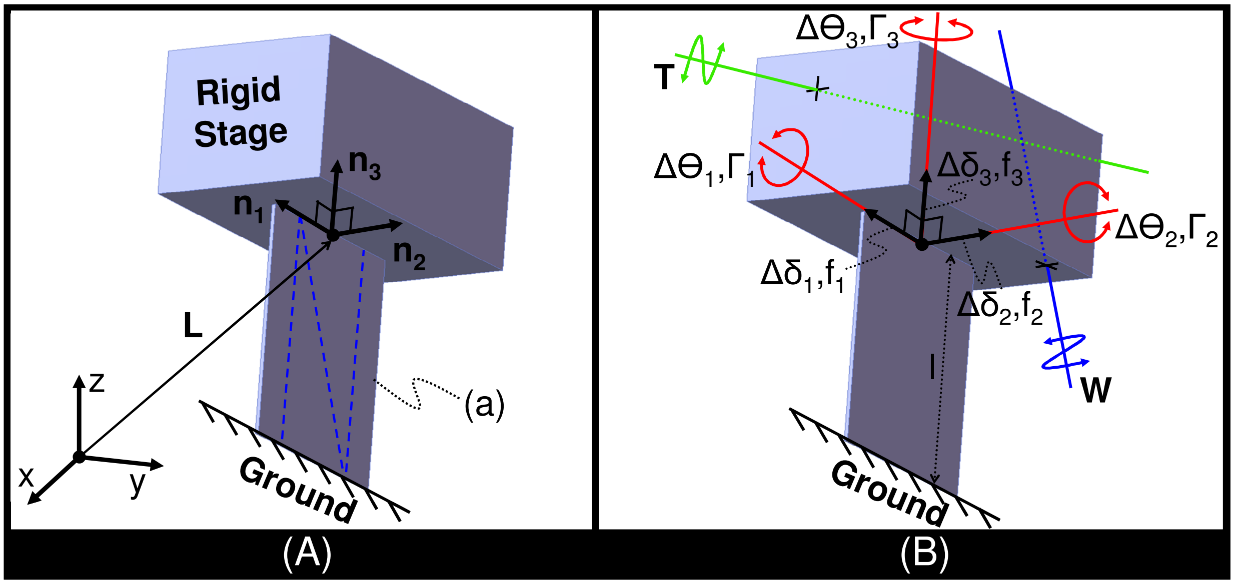

Figure 4.7: *Parameters that define the TWSM, (A) and (B).*

4.5.2 Derivation of the Twist-Wrench Stiffness Matrix

When one imposes a displacement upon a stage, the displacement may be represented by a twist, **T**. This twist may be written as a linear combination of six independent twists that intersect at the point **L**, where the flexible constraint *(a)* attaches to the stage. These six twists align with the constraint's axes of principle stiffness as shown in Figure 4.76B. Three of these twists may be expressed as orthogonal translations shown as black arrows in Figure 4.7B, while the other three may be expressed as orthogonal rotations shown as lines with circular arrows about their axes. The first three columns of the matrix in Equation 4.10 represent the three orthogonal unit rotational twist vectors and the last three columns of this matrix represent the three orthogonal unit translational twist vectors. The linear combination of these six twists may therefore be written as

$$\mathbf{T} = \left[N^{(a)}\right] \cdot \left[\Delta\theta_1 \quad \Delta\theta_2 \quad \Delta\theta_3 \quad \Delta\delta_1 \quad \Delta\delta_2 \quad \Delta\delta_3\right]^T . \tag{4.13}$$

Upon rearranging Equation 4.13 and multiplying the result by Equation 4.11, one obtains a vector that contains the moments, Γ_j, and forces, f_j, that are imposed upon the stage by constraint *(a)*:

$$\left[\Gamma_1 \quad \Gamma_2 \quad \Gamma_3 \quad f_1 \quad f_2 \quad f_3\right]^T = \left[S^{(a)}\right] \cdot \left[N^{(a)}\right]^{-1} \cdot \mathbf{T} \tag{4.14}$$

The reaction wrench imposed on the stage by constraint *(a)*, $\mathbf{W}^{(\mathbf{a})}$, as a result of $\mathbf{T}$ may be found by multiplying Equation 4.14 and Equation 4.9.

$$\mathbf{W}^{(\mathbf{a})} = \left[N_R^{(a)}\right] \cdot \left[S^{(a)}\right] \cdot \left[N^{(a)}\right]^{-1} \cdot \mathbf{T}. \tag{4.15}$$

Equation 4.15 provides the linear combination of the six orthogonal wrenches that constraint *(a)* imposes on the stage shown in Figure 4.7B. Note that the first three columns of the matrix in Equation 4.9 represent three orthogonal unit pure moment wrenches, and the last three columns of this matrix represent three orthogonal unit pure force wrenches. The wrench, $\mathbf{W}$, necessary to actuate the stage may be found by summing each constraint's reaction wrench

$$\mathbf{W} = \sum_{a=1}^{C} \mathbf{W}^{(\mathbf{a})}. \tag{4.16}$$

Recalling Equation 4.1, the TWSM of the system in Equation 4.8 may be found by substituting Equation 4.15 into Equation 4.16.

4.6 Numerical Example

The flexure system from Figure 4.2 was built to form the basis of assessing the predictions of the theory and principles of this paper. This flexure system is shown with its dimensions in Figure 4.8.

4.6.1 Calculating the Flexure System's TWSM

In order to calculate the TWSM of the flexure system shown in Figure 4.8, the stiffness matrix associated with each individual constraint, *(a)*, must be determined. If the intersection point of the system's constraint lines coincide with the origin of the coordinate system shown in Figure 4.8A, the location vector, $\mathbf{L}$, of constraint *(1)* in meters is

$$\mathbf{L} = \left[0.102 \cdot \cos(60^\circ) \quad (0.102 \cdot \sin(60^\circ) \quad -0.102\right]^T. \tag{4.17}$$

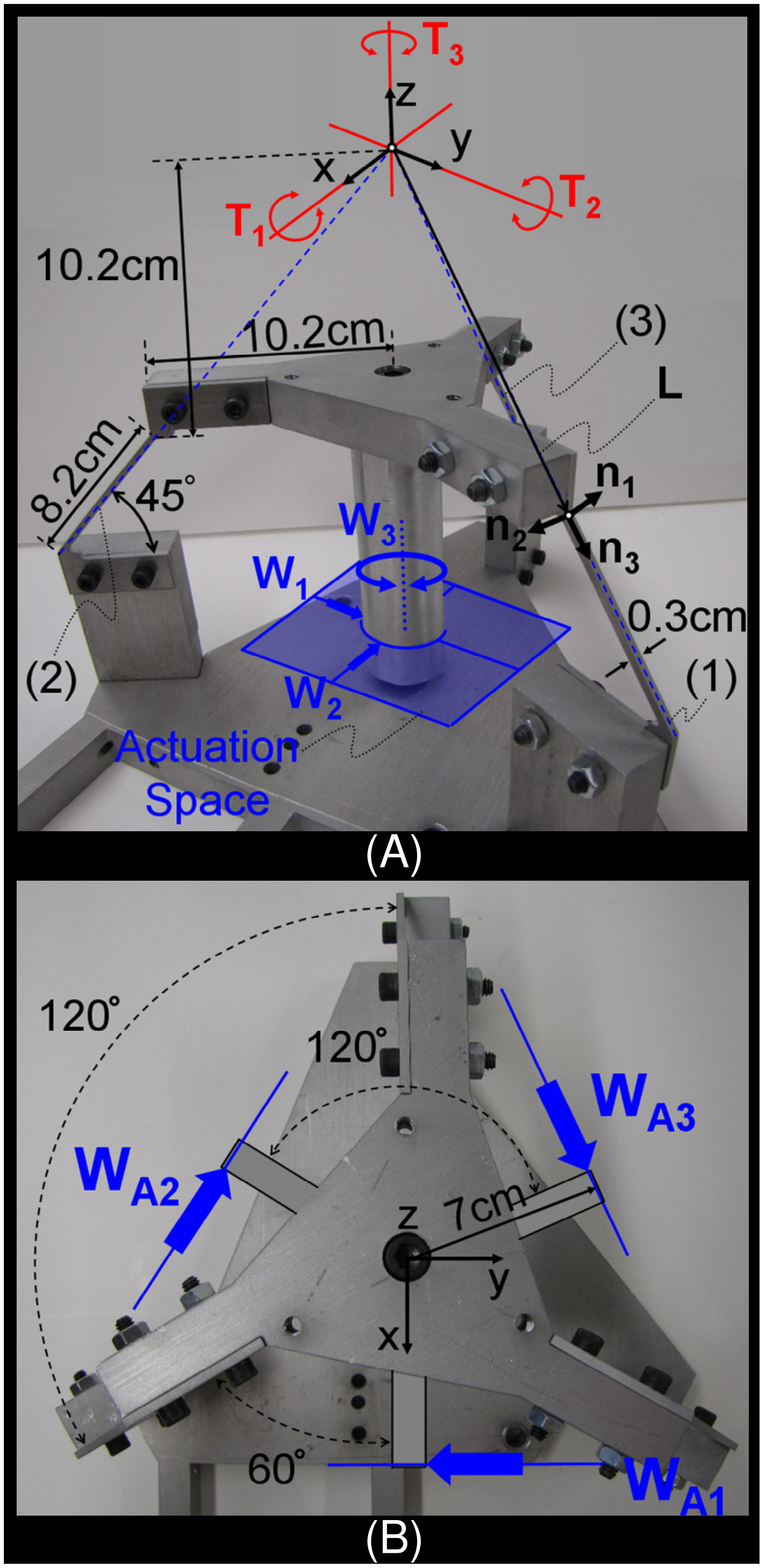

Figure 4.8: *Numerical example flexure system with parameters (A), and actuator setup with parameters (B).*

The orthogonal unit vectors, $\mathbf{n_1}$, $\mathbf{n_2}$, and $\mathbf{n_3}$, that point along constraint *(1)*'s principle stiffness axes are

$$\mathbf{n}_1 = \begin{bmatrix} \frac{\cos(60^\circ)}{\sqrt{2}} & \frac{\sin(60^\circ)}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}^T, \tag{4.18}$$

$$\mathbf{n_2} = \begin{bmatrix} \sin(60^\circ) & -\cos(60^\circ) & 0 \end{bmatrix}^T, \text{ and} \tag{4.19}$$

$$\mathbf{n_3} = \begin{bmatrix} \frac{\cos(60^\circ)}{\sqrt{2}} & \frac{\sin(60^\circ)}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}^T. \tag{4.20}$$

The matrices from Equation 4.9 and Equation 4.10 may be constructed for constraint *(1)* using Equation 4.17 through Equation 4.20. Using the geometry defined in Figure 4.8 and recognizing that constraint *(1)* has a square cross-section and is made of aluminum with an elastic modulus of 68 GPa and a shear modulus of 25GPa, the matrix from Equation 4.11 may also be constructed for constraint *(1)*. If the matrices from Equation 4.9 through Equation 4.11 are constructed in like manner for constraint *(2)* and constraint *(3)*, then the entire stiffness matrix, $[K_{TW}]$, or TWSM of the flexure system may be calculated using Equation 4.8 where $C=3$. The TWSM of the flexure system from Figure 4.8 is

$$[K_{TW}] = \begin{bmatrix} 0 & -3925.69 & 0 & 5620037.80 & 0 & 0 \\ 3925.69 & 0 & 0 & 0 & 5620037.80 & 0 \\ 0 & 0 & 0 & 0 & 0 & 11210106.50 \\ 787.03 & 0 & 0 & 0 & 3925.69 & 0 \\ 0 & 787.03 & 0 & -3925.69 & 0 & 0 \\ 0 & 0 & 528.80 & 0 & 0 & 0 \end{bmatrix}. \tag{4.21}$$

4.6.2 Calculating the System's Actuation Space

In order to determine the actuation space for the flexure system from Figure 4.8, a twist matrix that describes the system's freedom space must be constructed using Equation 4.2. Three independent twists, $\mathbf{T_1}$, $\mathbf{T_2}$, and $\mathbf{T_3}$, selected from the freedom space are shown in Figure 4.8A. These twists' location vectors, $\mathbf{c}$, may be zero vectors since they all pass through the origin. The twists' pitch values, p, are zero since they are all pure rotations. Using Equation 1.1 each twist's 6x1 vector may be individually constructed and placed within the twist matrix, $[T]$, according to Equation 4.2 as

$$[T] = \begin{bmatrix} \pi/180 & 0 & 0 & 0 & 0 & 0 \\ 0 & \pi/180 & 0 & 0 & 0 & 0 \\ 0 & 0 & \pi/180 & 0 & 0 & 0 \end{bmatrix}^T, \tag{4.22}$$

where the rotational magnitude of each orientation vector, $\mathbf{\Delta\Theta}$, is set to one degree expressed in radians. The wrench matrix, $[W]$, that describes the system's actuation space may be calculated by substituting the matrix from Equation 4.22 and the system's TWSM from Equation 4.21 into Equation 4.3. The wrenches, $\mathbf{W_1}$, $\mathbf{W_2}$, and $\mathbf{W_3}$, from this matrix are shown in Figure 4.8A. Equations 1.3, 1.10, and 1.11 may then be applied to decompose these wrenches. The resulting location vectors, $\mathbf{r_1}$ and $\mathbf{r_2}$, (expressed in meters) of $\mathbf{W_1}$ and $\mathbf{W_2}$ are

$$\mathbf{r_1} = \mathbf{r_2} = [0 \quad 0 \quad -0.20]^T. \tag{4.23}$$

Both $\mathbf{W_1}$ and $\mathbf{W_2}$ have q-values equal to zero. These wrenches are thus pure force wrenches with orientation vectors, $\mathbf{f_1}$ and $\mathbf{f_2}$, (expressed in N) defined as

$$\mathbf{f_1} = [0 \quad 68.52 \quad 0]^T, \text{ and} \tag{4.24}$$

$$\mathbf{f_2} = [-68.52 \quad 0 \quad 0]^T. \tag{4.25}$$

The q-value of $\mathbf{W_3}$ is infinite and thus $\mathbf{W_3}$ is a pure moment wrench with no meaningful location vector. Its orientation vector, $\boldsymbol{\tau_3}$, (expressed in Nm) is

$$\boldsymbol{\tau_3} = [0 \quad 0 \quad 9.23]^T. \tag{4.26}$$

The linear combination of $\mathbf{W_1}$, $\mathbf{W_2}$, and $\mathbf{W_3}$ results in pure force wrenches that lie on a plane a certain distance from the system's origin as shown in Figure 4.8A. It is evident from Equation 4.23 that this distance is 20 cm. Referring to Equation 4.24 and Equation 4.25, the stiffness value associated with $\mathbf{T_1}$ and $\mathbf{W_1}$ and with $\mathbf{T_2}$ and $\mathbf{W_2}$ is 68.52 N/degree. That is to say that if a force of 68.52 N is applied to the stage of the flexure system along the line of $\mathbf{W_1}$ or $\mathbf{W_2}$, the stage will rotate a degree about the line of $\mathbf{T_1}$ or $\mathbf{T_2}$ respectively. Referring to Equation 4.26, the stiffness value associated with $\mathbf{T_3}$ and $\mathbf{W_3}$ is 9.23 Nm/degree meaning that a pure moment of 9.23 Nm applied to the stage in a direction normal to the actuation plane will result in a degree of rotation about the line of $\mathbf{T_3}$.

4.6.3 Calculating Force and Displacement Ratios

In this section, a numerical example of the theory from Sections 4.4.1 and 4.4.2 is presented. Suppose three linear actuators, $\mathbf{W_{A1}}$, $\mathbf{W_{A2}}$, and $\mathbf{W_{A3}}$, are placed in the system's actuation space with the same configuration as the actuators shown in Figure 4.6. Suppose that these actuators are orthogonally attached to the stage extensions at a distance of 7cm from the center of the stage as shown in Figure 4.8B.

In order to determine the force magnitudes of the actuators, F_{A1}, F_{A2}, and F_{A3}, that actuate a single degree of rotation about the line of $\mathbf{T_1}$ shown in Figure 4.8A, an actuation matrix, $[W_A]$, must first be constructed. The location vector, $\mathbf{r_{A1}}$, (expressed in meters) of the first actuator, $\mathbf{W_{A1}}$, is

$$\mathbf{r_{A1}} = \begin{bmatrix} 0.07 & 0 & -0.20 \end{bmatrix}^T . \tag{4.27}$$

This vector points from the origin of the coordinate system to the point where the actuator attaches to the stage. The orientation vector, $\mathbf{f_{A1}}$, (expressed in N) of the first actuator, $\mathbf{W_{A1}}$, is set to be a unit vector and is defined as

$$\mathbf{f_{A1}} = \begin{bmatrix} 0 & -1 & 0 \end{bmatrix}^T . \tag{4.28}$$

If the location and unit orientation vectors of $\mathbf{W_{A2}}$ and $\mathbf{W_{A3}}$ are also determined, and these vectors are placed in a matrix according to Equation 4.4, the resulting actuation matrix is

$$[W_A] = \begin{bmatrix} 0 & -1 & 0 & -0.20 & 0 & -0.07 \\ -0.866 & 0.5 & 0 & 0.1 & 0.1732 & -0.07 \\ 0.866 & 0.5 & 0 & 0.1 & -0.1732 & -0.07 \end{bmatrix}^T . \tag{4.29}$$

Note that the q-values of all the actuators are zero since they are all linear actuators. Substituting Equation 4.29, Equation 4.21, and $\mathbf{T_1}$ from the first row of the matrix in Equation 4.22 into Equation 4.5, the desired force magnitudes, F_{A1}, F_{A2}, and F_{A3}, (expressed in N) are obtained for rotating the stage about the line of $\mathbf{T_1}$ a single degree:

$$\begin{bmatrix} F_{A1} & F_{A2} & F_{A3} \end{bmatrix}^T = \begin{bmatrix} -45.68 & 22.84 & 22.84 \end{bmatrix}^T . \tag{4.30}$$

Note that $\mathbf{T_1}$ will always be actuated for any magnitude if the actuator force ratio, F_{A1}:F_{A2}:F_{A3}, is set to -2:1:1.

In order to determine the displacement magnitudes of the actuators, δ_{A1}, δ_{A2}, and δ_{A3}, that actuate a single degree of rotation about the line of $\mathbf{T_1}$ shown in Figure 4.3A, the force magnitudes from Equation 4.30 are used to individually determine each actuator's displacement. The twist, $\mathbf{T_{A1}}$, that would result if only the first actuator's force, F_{A1}, were imposed on the stage may be determined by substituting Equation 4.21, $\mathbf{W_{A1}}$ from the first row of Equation 4.29, and the value of F_{A1} from Equation 4.30 into Equation 4.6. Equations 1.1, 1.10, and 1.11 may then be applied to decompose this twist. Its pitch, p_{A1}, is zero since it is a pure rotation, its location vector, $\mathbf{c_{A1}}$, is a zero vector since it passes through the origin, and its orientation vector, $\mathbf{\Delta\Theta_{A1}}$, (expressed in radians) is

$$\mathbf{\Delta\theta_{A1}} = \begin{bmatrix} 0.0116 & 0 & 0.0060 \end{bmatrix}^T . \tag{4.31}$$

Substituting these values and those expressed in Equation 4.27 and Equation 4.28 into Equation 4.7 results in a displacement magnitude, δ_{A1}, of -0.00248m. Repeating this process for the other two actuators yields actuator displacement magnitudes δ_{A2}, and δ_{A3} (expressed in meters) of

$$\begin{bmatrix} \delta_{A1} & \delta_{A2} & \delta_{A3} \end{bmatrix}^T = \begin{bmatrix} -0.00275 & 0.00137 & 0.00137 \end{bmatrix}^T . \tag{4.32}$$

Note that $\mathbf{T_1}$ will always be actuated for any magnitude if the actuator displacement ratio, δ_{A1}:δ_{A2}:δ_{A3},, is set to -2:1:1. Furthermore, the actuator force and displacement ratios are the same for this particular flexure system. The reason for this is that the stiffness values associated with all three actuator attachment locations are equivalent due to flexure symmetry.

4.7 Experimental Validation

In this section, the theory and predictions of the previous section are experimentally validated using the flexure system from Figure 4.8. This flexure system was actuated by a micrometer head, shown in Figure 4.9A, which could be adjusted to vary the location of force/displacement it applied to a post that was attached to the bottom of the stage. A coordinate measuring machine (CMM) was used to measure the stage's response as the location of actuation was varied. The setup is illustrated in Figure 4.9B.

The stage was actuated with forces that yielded rotations of 0.2 degrees. The measurements were used to ascertain the parasitic errors, *e*, at a location that coincides with the intersection of the three constraining flexures' lines of action where the probe's tip would be located as shown in Figure 4.9A.

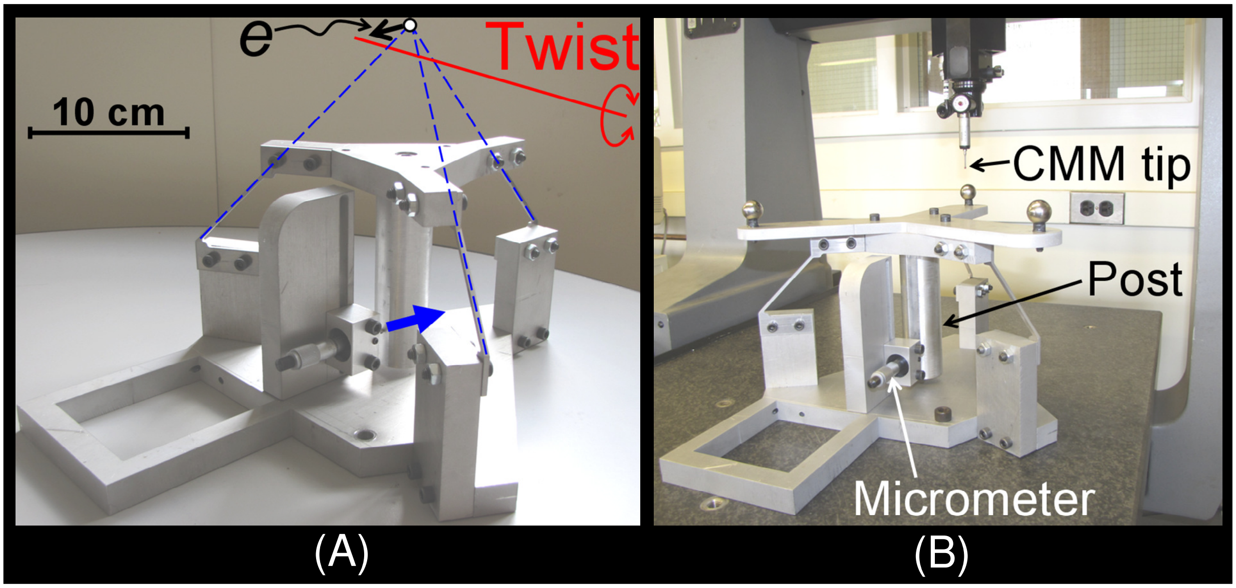

Figure 4.9: *Experimental flexure setup for measuring error, e (A), and the CMM used to measure the twists imposed by the micrometer (B).*

The measured and predicted results are shown in Figure 4.10. The theory in this chapter predicted that minimum parasitic errors would occur when the actuation force was applied on a plane that was 20 cm below the tip of the probe. According to the measured data, the minimum error occurred when the actuation force was applied around 17.5 cm below the tip of the probe. The error bars specify regions wherein the error exists with 95% confidence. Note that the magnitude of the measured data is much larger than the magnitude of the theoretical prediction. The reason for this discrepancy is that the measured tip displacements capture both the large deformation kinematics as well as the parasitic errors whereas the theoretical prediction captures only the parasitic error trend.

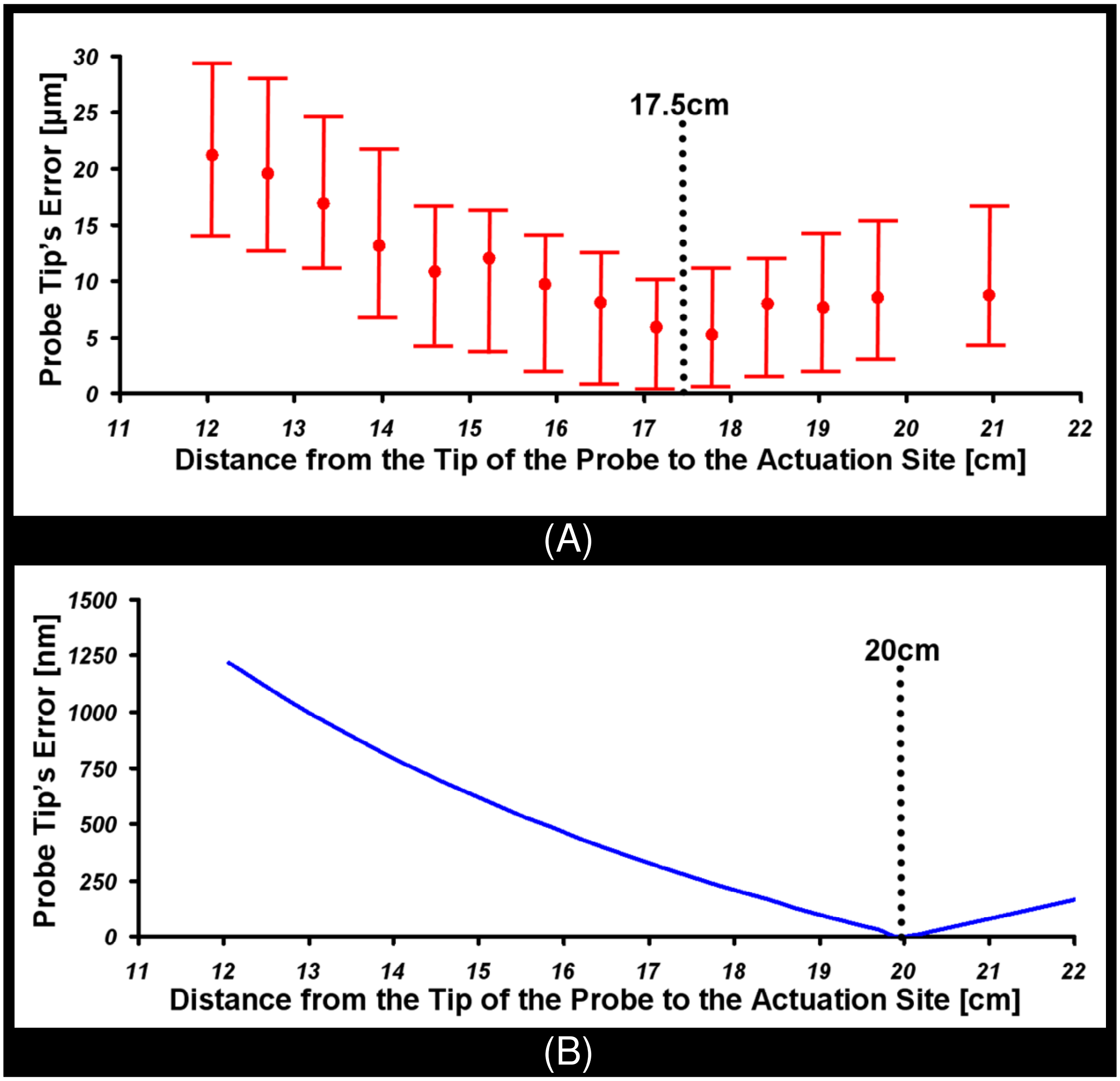

Figure 4.10: *Experimental (A) vs. theoretical (B) errors of the probe's tip*

The slope of the error trend is a measure of the sensitivity of the flexure system to actuator placement. It is desirable, therefore, to design flexure systems that have error plots with shallow slopes near the location of minimal error to accommodate for inaccurate placement of the actuator.

CHAPTER 5:
Conclusions

This chapter summarizes the contributions of this thesis and discusses future work to advance the FACT design process.

5.1 Thesis Contributions

This section reviews the major accomplishments of this thesis:

(1) The theory, principles, and best practices needed to generate serial flexure system concepts that possess a desired set of DOFs have been created. This knowledge is embodied in the form of freedom and constraint spaces that visually represent the rigorous mathematics of screw theory, projective geometry, and constraint-based design.

(2) Intermediate freedom spaces have been created to help designers appropriately stack conjugated flexure elements. Guidelines have been provided with these intermediate spaces that enable designers to avoid or utilize underconstraint in flexure design.

(3) The complete chart of freedom and constraint spaces has been created and described. This chart has been adapted for the purpose of synthesizing flexure systems. With this chart, designers may generate all flexure concepts (both parallel and serial) that achieve any desired set of motions. These concepts may then be compared before the final design is selected.

(4) The twist-wrench stiffness matrix has been created that enables designers to model and predict a flexure system's elastomechanic behavior. While the FACT design process enables designers to synthesize flexure systems that possess desired kinematic characteristics, the twist-wrench stiffness matrix enables designer to design flexure systems that possess desired stiffness characteristics.

(5) Actuation space has been created that enables designers to recognize all the regions where actuators could be placed for minimizing motion errors. Guidelines have been created that help designers select the optimal number, location, and orientation of actuators.

(6) The complete chart of actuation spaces for linear actuators has been created. This chart helps designers quickly identify whether a flexure concept is possible to actuate with linear actuators. If it is possible, the chart helps designers recognize whether the design may possess a practical actuation scheme.

(7) The theory has been created that enables designers to rapidly calculate the force and displacement actuator outputs for actuating a specific DOF.

5.2 Future Work

This section describes future research efforts to further the FACT design process: Currently research is underway that will enable designers to:

(1) synthesize large deformation flexure systems. These flexure systems would possess stages that follow prescribed motion paths.

(2) optimally place actuators for serial flexure systems as well as for parallel flexure systems. Each rigid stage in the serial chain would have its own actuation space.

(3) optimally place actuators for circumstances where dynamic actuation is applicable, i.e. actuator inputs are time dependent.

(4) synthesize flexure systems that possess desired dynamic characteristics (mode shapes, natural frequencies, etc).

(5) synthesize parallel flexure systems that possess DOFs that could not normally be achieved without the use of serial flexure systems. The constraints of these parallel flexure systems would

be far removed from the stage of interest. As spaces from the FACT chart are displaced to infinity, they transform into other spaces that also belong to the FACT chart. The knowledge of how these spaces transform when displaced to infinity should be studied so that designers may achieve this objective.

(6) synthesize compliant transmission elements that possess multiple decoupled inputs. These inputs would link to decoupled outputs with desired transmission ratios.

(7) perform sensitivity analysis of constraint fabrication, orientation and location errors using the mathematics of actuation and constraint spaces.

(8) utilize buckling for low stiffness, large motion flexure bearings

Appendix A:
Derivation of Spaces within Chart

This appendix provides the derivation of the spaces within the chart of Figure 2.1. More specifically, the logic is presented that was used to consider every way twists could be combined to generate all possible freedom spaces. The details that describe how a freedom space and its complementary constraint space are generated given a certain configuration of twists are provided in Hopkins [21].

The 0 DOF column consists of a single type because there is only one way no twists may be combined. The freedom space of this type is, therefore, empty space. The 1 DOF column consists of three types because, according to screw theory, there are only three kinds of motions—rotations, screws, and translations. Subsequently, these three motions are the freedom spaces within the 1 DOF column. The 2 DOF column consists of ten types because only ten freedom spaces are generated by combining two twists of every type with all possible configurations. The proof of this fact is found in Hopkins [21]. Every freedom space from the 2 DOF column may be generated by using the twist configurations shown in Figure A.1.

Consider, for instance, Figure A.1A:

(1) If the two pitch values, p_1 and p_2, are both zero, the freedom space of 2 DOF Type 2 is generated.

(2) If the two pitch values, p_1 and p_2, are the same, finite, non-zero value, the freedom space of 2 DOF Type 5 is generated.

(3) If one of the two pitch values, p_1 and p_2, is infinity and the other pitch value is a finite value, the freedom space of 2 DOF Type 8 is generated.

(4) If the two pitch values, p_1 and p_2, are different and finite values, the freedom space of 2 DOF Type 9 is generated.

Now consider Figure A.1B:

(5) If the two pitch values, p_1 and p_2, are both zero, the freedom space of 2 DOF Type 1 is generated.

(6) If the two pitch values, p_1 and p_2, are different, non-zero, finite values with opposite signs, the freedom space of 2 DOF Type 3 is generated.

(7) If the two pitch values, p_1 and p_2, are the same, finite, non-zero value, the freedom space of 2 DOF Type 4 is generated.

(8) If one of the two pitch values, p_1 and p_2, is zero and the other pitch value is finite and non-zero, the freedom space of 2 DOF Type 6 is generated.

(9) If the two pitch values, p_1 and p_2, are different, non-zero, finite values with the same sign, the freedom space of 2 DOF Type 7 is generated.

(10) If the two pitch values, p_1 and p_2, are both equal to infinity, the freedom space of 2 DOF Type 10 is generated.

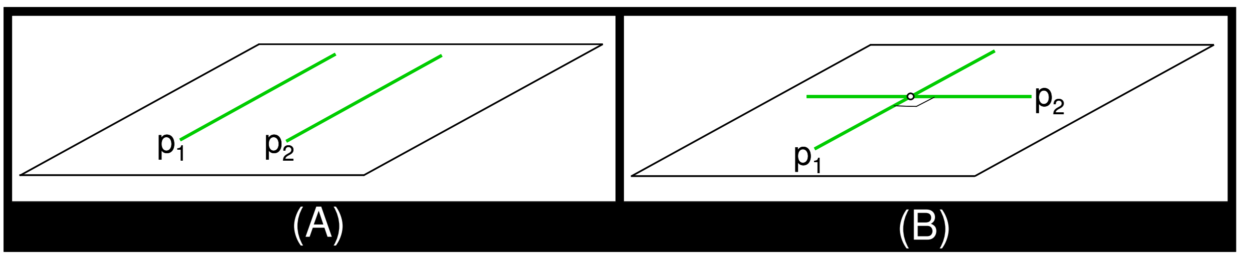

Figure A.1: *Twist configurations that generate every type within the 2 DOF column*

Every freedom space within the 3 DOF column may be generated by adding a third twist to the twist configurations from Figure A.1 for each of the ten conditions previously listed. This third twist should be arranged in every configuration and should possess every pitch value in order to generate every freedom space. These configurations for adding a third twist to the configuration from Figure A.1A and Figure A.1B are shown in Figure A.3 respectively.

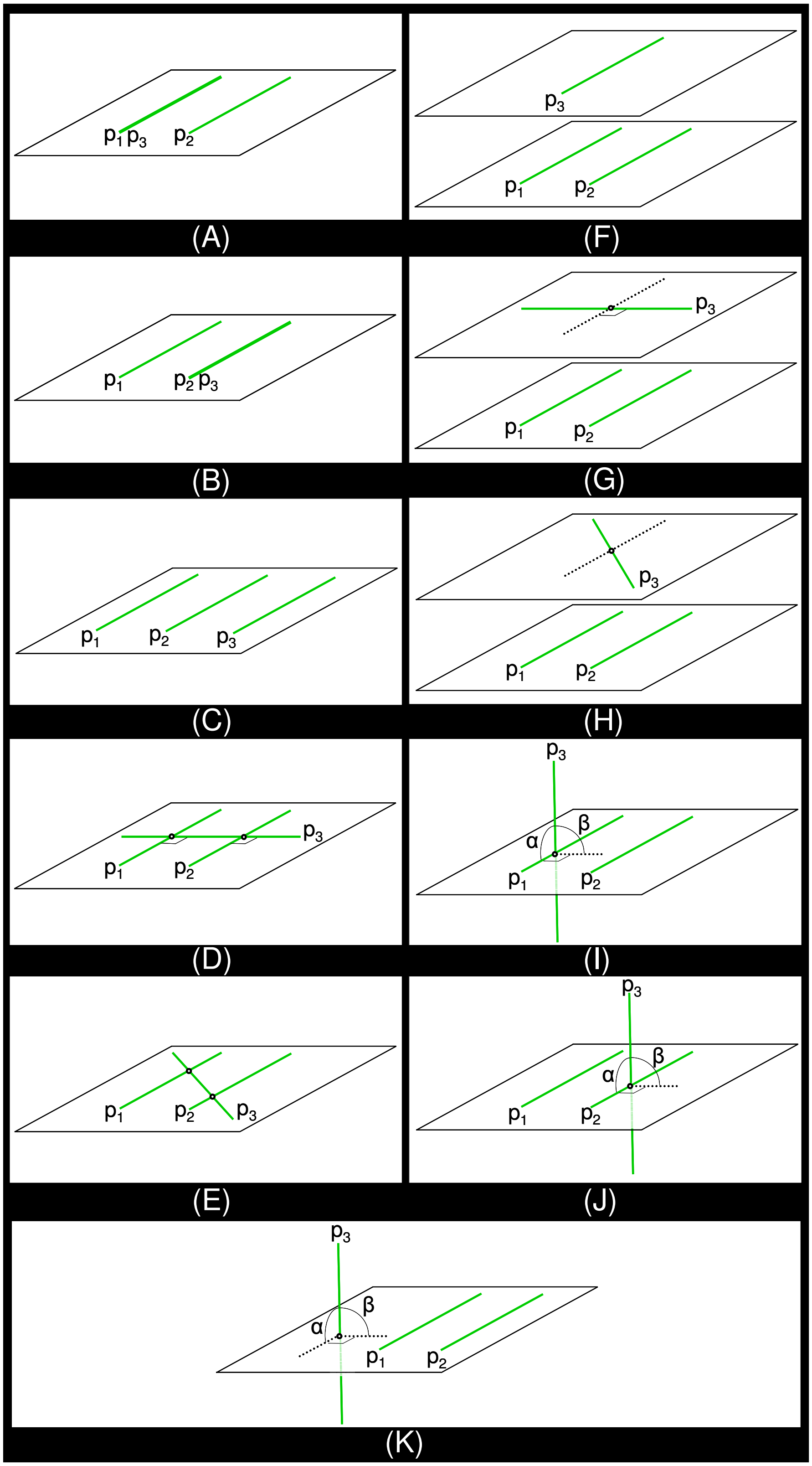

Figure A.2: *All twist configurations that build off of the configuration from Figure A.1A to generate types within the 3 DOF column.*

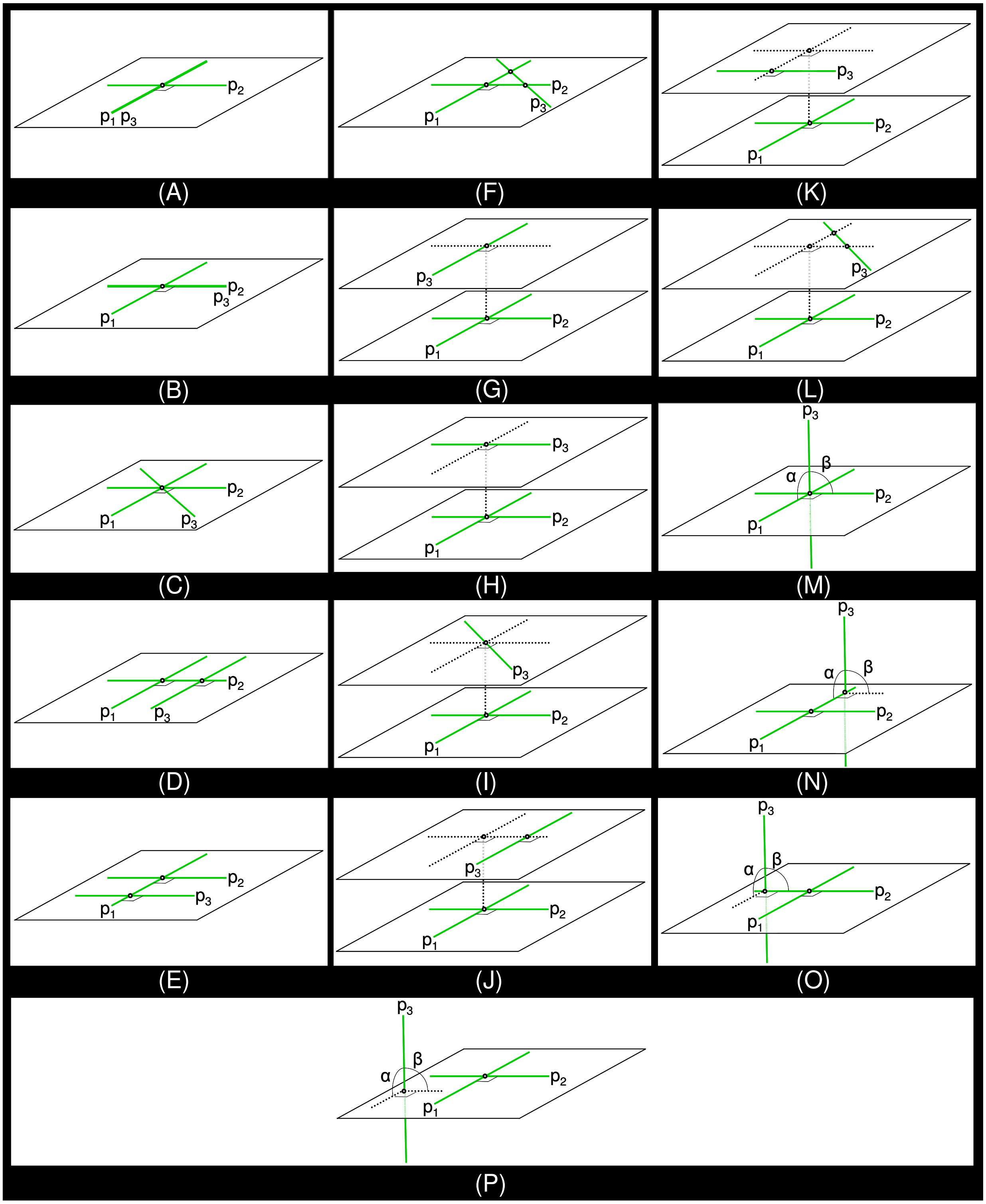

Figure A.3: *All twist configurations that build off of the configuration from Figure A.1B to generate types within the 3 DOF column.*

For condition (1) where p_1 and p_2 are both equal to zero,

if p_3 is equal to infinity from

- Figure A.2A, the 3 DOF Type 21 freedom space is generated.
- Figure A.2B, the 3 DOF Type 21 freedom space is generated.
- Figure A.2C, the 3 DOF Type 21 freedom space is generated.
- Figure A.2D, the 3 DOF Type 2 freedom space is generated.
- Figure A.2E, the 3 DOF Type 22 freedom space is generated.
- Figure A.2F, the 3 DOF Type 21 freedom space is generated.
- Figure A.2G, the 3 DOF Type 2 freedom space is generated.
- Figure A.2H, the 3 DOF Type 22 freedom space is generated.
- Figure A.2I and α and β equal 90 degrees, the 2 DOF Type 2 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 21 freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 2 freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 22 freedom space is generated.
- Figure A.2J and α and β equal 90 degrees, the 2 DOF Type 2 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 21 freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 2 freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 22 freedom space is generated.
- Figure A.2K and α and β equal 90 degrees, the 2 DOF Type 2 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 21 freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 2 freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 22 freedom space is generated.

if p_3 is equal to zero from

- Figure A.2A, the 2 DOF Type 2 freedom space is generated.
- Figure A.2B, the 2 DOF Type 2 freedom space is generated.
- Figure A.2C, the 2 DOF Type 2 freedom space is generated.
- Figure A.2D, the 3 DOF Type 1 freedom space is generated.

- Figure A.2E, the 3 DOF Type 1 freedom space is generated.
- Figure A.2F, the 3 DOF Type 2 freedom space is generated.
- Figure A.2G, the 3 DOF Type 6 freedom space is generated.
- Figure A.2H, the 3 DOF Type 6 freedom space is generated.
- Figure A.2I and α and β equal 90 degrees, the 3 DOF Type 4A freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 4A freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 4B freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 4B freedom space is generated.
- Figure A.2J and α and β equal 90 degrees, the 3 DOF Type 4A freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 4A freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 4B freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 4B freedom space is generated.
- Figure A.2K and α and β equal 90 degrees, the 3 DOF Type 4A freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 4A freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 4B freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 4B freedom space is generated.

if p_3 is a finite non-zero value from

- Figure A.2A, the 3 DOF Type 21 freedom space is generated.
- Figure A.2B, the 3 DOF Type 21 freedom space is generated.
- Figure A.2C, the 3 DOF Type 21 freedom space is generated.
- Figure A.2D, the 3 DOF Type 15 freedom space is generated.
- Figure A.2E, the 3 DOF Type 6 freedom space is generated.
- Figure A.2F, the 3 DOF Type 22 freedom space is generated.
- Figure A.2G, the 3 DOF Type 6 freedom space is generated.
- Figure A.2H, the 3 DOF Type 6 or 15 freedom spaces may be generated.
- Figure A.2I and α and β equal 90 degrees, the 3 DOF Type 4A freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type

4A freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 4B freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 4B freedom space is generated.

- Figure A.2J and α and β equal 90 degrees, the 3 DOF Type 4A freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 4A freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 4B freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 4B freedom space is generated.
- Figure A.2K and α and β equal 90 degrees, the 3 DOF Type 4A freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 4A freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 4B freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 4B freedom space is generated.

For condition (2) where p_1 and p_2 are the same, finite, non-zero value,

if p_3 is equal to infinity from

- Figure A.2A, the 3 DOF Type 21 freedom space is generated.
- Figure A.2B, the 3 DOF Type 21 freedom space is generated.
- Figure A.2C, the 3 DOF Type 21 freedom space is generated.
- Figure A.2D, the 3 DOF Type 18 freedom space is generated.
- Figure A.2E, the 3 DOF Type 22 freedom space is generated.
- Figure A.2F, the 3 DOF Type 21 freedom space is generated.
- Figure A.2G, the 3 DOF Type 18 freedom space is generated.
- Figure A.2H, the 3 DOF Type 22 freedom space is generated.
- Figure A.2I and α and β equal 90 degrees, the 2 DOF Type 5 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 21 freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 18 freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 22 freedom space is generated.

- Figure A.2J and α and β equal 90 degrees, the 2 DOF Type 5 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 21 freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 18 freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 22 freedom space is generated.
- Figure A.2K and α and β equal 90 degrees, the 2 DOF Type 5 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 21 freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 18 freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 22 freedom space is generated.

if p_3 is equal to zero from

- Figure A.2A, the 3 DOF Type 21 freedom space is generated.
- Figure A.2B, the 3 DOF Type 21 freedom space is generated.
- Figure A.2C, the 3 DOF Type 21 freedom space is generated.
- Figure A.2D, the 3 DOF Type 15 freedom space is generated.
- Figure A.2E, the 3 DOF Type 6 freedom space is generated.
- Figure A.2F, the 3 DOF Type 22 freedom space is generated.
- Figure A.2G, the 3 DOF Type 6 freedom space is generated.
- Figure A.2H, the 3 DOF Type 6 or 15 freedom spaces may be generated.
- Figure A.2I and α and β equal 90 degrees, the 3 DOF Type 5A freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 5A freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 5B freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 5B freedom space is generated.
- Figure A.2J and α and β equal 90 degrees, the 3 DOF Type 5A freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 5A freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 5B freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 5B freedom space is generated.
- Figure A.2K and α and β equal 90 degrees, the 3 DOF Type 45 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type

5A freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 5B freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 5B freedom space is generated.

if p_3 is equal to p_1 and p_2 from

- Figure A.2A, the 2 DOF Type 5 freedom space is generated.
- Figure A.2B, the 2 DOF Type 5 freedom space is generated.
- Figure A.2C, the 2 DOF Type 5 freedom space is generated.
- Figure A.2D, the 3 DOF Type 17 freedom space is generated.
- Figure A.2E, the 3 DOF Type 17 freedom space is generated.
- Figure A.2F, the 3 DOF Type 18 freedom space is generated.
- Figure A.2G, the 3 DOF Type 6, 15, or 16 freedom spaces may be generated.
- Figure A.2H, the 3 DOF Type 6, 15, or 16 freedom spaces may be generated.
- Figure A.2I and α and β equal 90 degrees, the 3 DOF Type 5A freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 5A freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 5B freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 5B freedom space is generated.
- Figure A.2J and α and β equal 90 degrees, the 3 DOF Type 5A freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 5A freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 5B freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 5B freedom space is generated.
- Figure A.2K and α and β equal 90 degrees, the 3 DOF Type 5A freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 5A freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 5B freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 5B freedom space is generated.

if p_3 is a finite non-zero value that does not equal p_1 and p_2 from

- Figure A.2A, the 3 DOF Type 21 freedom space is generated.
- Figure A.2B, the 3 DOF Type 21 freedom space is generated.
- Figure A.2C, the 3 DOF Type 21 freedom space is generated.

- Figure A.2D, the 3 DOF Type 6 or 16 freedom spaces may be generated.
- Figure A.2E, the 3 DOF Type 6, 15, or 16 freedom spaces may be generated.
- Figure A.2F, the 3 DOF Type 22 freedom space is generated.
- Figure A.2G, the 3 DOF Type 6, 15, or 16 freedom spaces may be generated.
- Figure A.2H, the 3 DOF Type 6, 15, or 16 freedom spaces may be generated.
- Figure A.2I and α and β equal 90 degrees, the 3 DOF Type 5A freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 5A freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 5B freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 5B freedom space is generated.
- Figure A.2J and α and β equal 90 degrees, the 3 DOF Type 5A freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 5A freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 5B freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 5B freedom space is generated.
- Figure A.2K and α and β equal 90 degrees, the 3 DOF Type 5A freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 5A freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 5B freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 5B freedom space is generated.

For condition (3) where p_1 is a finite value and p_2 is infinity,

if p_3 is equal to infinity from

- Figure A.2A, the 2 DOF Type 8 freedom space is generated.
- Figure A.2B, the 2 DOF Type 8 freedom space is generated.
- Figure A.2C, the 2 DOF Type 8 freedom space is generated.
- Figure A.2D, the 3 DOF Type 21 freedom space is generated.
- Figure A.2E, the 3 DOF Type 21 freedom space is generated.
- Figure A.2F, the 2 DOF Type 8 freedom space is generated.
- Figure A.2G, the 3 DOF Type 21 freedom space is generated.

- Figure A.2H, the 3 DOF Type 21 freedom space is generated.
- Figure A.2I and α and β equal 90 degrees, the 3 DOF Type 21 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 21 freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 21 freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 21 freedom space is generated.
- Figure A.2J and α and β equal 90 degrees, the 3 DOF Type 21 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 21 freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 21 freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 21 freedom space is generated.
- Figure A.2K and α and β equal 90 degrees, the 3 DOF Type 21 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 21 freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 21 freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 21 freedom space is generated.

if p_3 is equal to zero from

- Figure A.2A, the 2 DOF Type 8 freedom space is generated.
- Figure A.2B, the 3 DOF Type 21 freedom space is generated.
- Figure A.2C, the 3 DOF Type 21 freedom space is generated.
- Figure A.2D, the 3 DOF Type 4A freedom space is generated.
- Figure A.2E, the 3 DOF Type 4A freedom space is generated.
- Figure A.2F, the 3 DOF Type 21 freedom space is generated.
- Figure A.2G, the 3 DOF Type 4A freedom space is generated.
- Figure A.2H, the 3 DOF Type 5A freedom space is generated.
- Figure A.2I and α and β equal 90 degrees, the 3 DOF Type 4A freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 4A freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 4A freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 4A freedom space is generated.

- Figure A.2J and α and β equal 90 degrees, the 3 DOF Type 4A freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 5A freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 4A freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 5A freedom space is generated.
- Figure A.2K and α and β equal 90 degrees, the 3 DOF Type 4A freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 5A freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 4A freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 5A freedom space is generated.

if p_3 is a finite, non-zero value from

- Figure A.2A, the 2 DOF Type 8 freedom space is generated.
- Figure A.2B, the 3 DOF Type 21 freedom space is generated.
- Figure A.2C, the 3 DOF Type 21 freedom space is generated.
- Figure A.2D, the 3 DOF Type 5A freedom space is generated.
- Figure A.2E, the 3 DOF Type 5A freedom space is generated.
- Figure A.2F, the 3 DOF Type 21 freedom space is generated.
- Figure A.2G, the 3 DOF Type 5A freedom space is generated.
- Figure A.2H, the 3 DOF Type 4A or 5A freedom spaces may be generated.
- Figure A.2I and α and β equal 90 degrees, the 3 DOF Type 5A freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 5A freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 5A freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 5A freedom space is generated.
- Figure A.2J and α and β equal 90 degrees, the 3 DOF Type 5A freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 4A or 5A freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 5A freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 4A or 5A freedom spaces may be generated.
- Figure A.2K and α and β equal 90 degrees, the 3 DOF Type 5A freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type

4A or 5A freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 5A freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 4A or 5A freedom spaces may be generated.

For condition (4) where p_1 and p_2 are different and finite values,

if p_3 is equal to infinity from

- Figure A.2A, the 3 DOF Type 21 freedom space is generated.
- Figure A.2B, the 3 DOF Type 21 freedom space is generated.
- Figure A.2C, the 3 DOF Type 21 freedom space is generated.
- Figure A.2D, the 3 DOF Type 22 freedom space is generated.
- Figure A.2E, the 3 DOF Type 22 freedom space is generated.
- Figure A.2F, the 3 DOF Type 21 freedom space is generated.
- Figure A.2G, the 3 DOF Type 22 freedom space is generated.
- Figure A.2H, the 3 DOF Type 22 freedom space is generated.
- Figure A.2I and α and β equal 90 degrees, the 3 DOF Type 21 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 2 DOF Type 9 or 3 DOF Type 21 freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 22 freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 22 freedom space is generated.
- Figure A.2J and α and β equal 90 degrees, the 3 DOF Type 21 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 2 DOF Type 9 or 3 DOF Type 21 freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 22 freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 22 freedom space is generated.
- Figure A.2K and α and β equal 90 degrees, the 3 DOF Type 21 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 2 DOF Type 9 or 3 DOF Type 21 freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 22 freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 22 freedom space is generated.

if p_3 is equal to zero from

- Figure A.2A, the 2 DOF Type 9 or 3 DOF Type 21 freedom spaces may be generated.
- Figure A.2B, the 2 DOF Type 9 or 3 DOF Type 21 freedom spaces may be generated.
- Figure A.2C, the 2 DOF Type 9 or 3 DOF Type 21 freedom spaces may be generated.
- Figure A.2D, the 3 DOF Type 4B freedom space is generated.
- Figure A.2E, the 3 DOF Type 4B freedom space is generated.
- Figure A.2F, the 3 DOF Type 22 freedom space is generated.
- Figure A.2G, the 3 DOF Type 4B freedom space is generated.
- Figure A.2H, the 3 DOF Type 5B freedom space is generated.
- Figure A.2I and α and β equal 90 degrees, the 3 DOF Type 4A or 5A freedom spaces may be generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 4A or 5A freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 4B or 5B freedom spaces may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 4B or 5B freedom spaces may be generated.
- Figure A.2J and α and β equal 90 degrees, the 3 DOF Type 4A or 5A freedom spaces may be generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 4A or 5A freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 4B or 5B freedom spaces may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 4B or 5B freedom spaces may be generated.
- Figure A.2K and α and β equal 90 degrees, the 3 DOF Type 4A or 5A freedom spaces may be generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 4A or 5A freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 4B or 5B freedom spaces may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 4B or 5B freedom spaces may be generated.

if p_3 is a finite, non-zero value from

- Figure A.2A, the 2 DOF Type 9 or 3 DOF Type 21 freedom spaces may be generated.
- Figure A.2B, the 2 DOF Type 9 or 3 DOF Type 21 freedom spaces may be generated.
- Figure A.2C, the 2 DOF Type 9 or 3 DOF Type 21 freedom spaces may be generated.
- Figure A.2D, the 3 DOF Type 5B freedom space is generated.

- Figure A.2E, the 3 DOF Type 5B freedom space is generated.
- Figure A.2F, the 3 DOF Type 22 freedom space is generated.
- Figure A.2G, the 3 DOF Type 5B freedom space is generated.
- Figure A.2H, the 3 DOF Type 4B or 5B freedom spaces may be generated.
- Figure A.2I and α and β equal 90 degrees, the 3 DOF Type 4A or 5A freedom spaces may be generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 4A or 5A freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 4B or 5B freedom spaces may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 4B or 5B freedom spaces may be generated.
- Figure A.2J and α and β equal 90 degrees, the 3 DOF Type 4A or 5A freedom spaces may be generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 4A or 5A freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 4B or 5B freedom spaces may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 4B or 5B freedom spaces may be generated.
- Figure A.2K and α and β equal 90 degrees, the 3 DOF Type 4A or 5A freedom spaces may be generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 4A or 5A freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 4B or 5B freedom spaces may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 4B or 5B freedom spaces may be generated.

For condition (5) where p_1 and p_2 are equal to zero,

if p_3 is equal to infinity from

- Figure A.3A, the 3 DOF Type 4A freedom space is generated.
- Figure A.3B, the 3 DOF Type 4A freedom space is generated.
- Figure A.3C, the 3 DOF Type 4A freedom space is generated.
- Figure A.3D, the 3 DOF Type 4A freedom space is generated.
- Figure A.3E, the 3 DOF Type 4A freedom space is generated.

- Figure A.3F, the 3 DOF Type 4A freedom space is generated.
- Figure A.3G, the 3 DOF Type 4A freedom space is generated.
- Figure A.3H, the 3 DOF Type 4A freedom space is generated.
- Figure A.3I, the 3 DOF Type 4A freedom space is generated.
- Figure A.3J, the 3 DOF Type 4A freedom space is generated.
- Figure A.3K, the 3 DOF Type 4A freedom space is generated.
- Figure A.3L, the 3 DOF Type 4A freedom space is generated.
- Figure A.3M and α and β equal 90 degrees, the 3 DOF Type 1 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 4B freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 4B freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 4B freedom space is generated.
- Figure A.3N and α and β equal 90 degrees, the 3 DOF Type 1 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 4B freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 4B freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 4B freedom space is generated.
- Figure A.3O and α and β equal 90 degrees, the 3 DOF Type 1 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 4B freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 4B freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 4B freedom space is generated.
- Figure A.3P and α and β equal 90 degrees, the 3 DOF Type 1 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 4B freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 4B freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 4B freedom space is generated.

if p_3 is equal to zero from

- Figure A.3A, the 2 DOF Type 1 freedom space is generated.
- Figure A.3B, the 2 DOF Type 1 freedom space is generated.
- Figure A.3C, the 2 DOF Type 1 freedom space is generated.

- Figure A.3D, the 3 DOF Type 1 freedom space is generated.
- Figure A.3E, the 3 DOF Type 1 freedom space is generated.
- Figure A.3F, the 3 DOF Type 1 freedom space is generated.
- Figure A.3G, the 3 DOF Type 4A freedom space is generated.
- Figure A.3H, the 3 DOF Type 4A freedom space is generated.
- Figure A.3I, the 3 DOF Type 4A freedom space is generated.
- Figure A.3J, the 3 DOF Type 4B freedom space is generated.
- Figure A.3K, the 3 DOF Type 4B freedom space is generated.
- Figure A.3L, the 3 DOF Type 4B freedom space is generated.
- Figure A.3M and α and β equal 90 degrees, the 3 DOF Type 3 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 3 freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 3 freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 3 freedom space is generated.
- Figure A.3N and α and β equal 90 degrees, the 3 DOF Type 9 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 9 freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 9 freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 9 freedom space is generated.
- Figure A.3O and α and β equal 90 degrees, the 3 DOF Type 9 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 9 freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 9 freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 9 freedom space is generated.
- Figure A.3P and α and β equal 90 degrees, the 3 DOF Type 9 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 9 freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 9 freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 9 freedom space is generated.

if p_3 is a finite, non-zero value from

- Figure A.3A, the 3 DOF Type 4A freedom space is generated.

- Figure A.3B, the 3 DOF Type 4A freedom space is generated.
- Figure A.3C, the 3 DOF Type 4A freedom space is generated.
- Figure A.3D, the 3 DOF Type 4B freedom space is generated.
- Figure A.3E, the 3 DOF Type 4B freedom space is generated.
- Figure A.3F, the 3 DOF Type 4B freedom space is generated.
- Figure A.3G, the 3 DOF Type 4A freedom space is generated.
- Figure A.3H, the 3 DOF Type 4A freedom space is generated.
- Figure A.3I, the 3 DOF Type 4A freedom space is generated.
- Figure A.3J, the 3 DOF Type 4B freedom space is generated.
- Figure A.3K, the 3 DOF Type 4B freedom space is generated.
- Figure A.3L, the 3 DOF Type 4B freedom space is generated.
- Figure A.3M and α and β equal 90 degrees, the 3 DOF Type 12 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 9 freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 9 freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 9 freedom space is generated.
- Figure A.3N and α and β equal 90 degrees, the 3 DOF Type 9 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 9 freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 9 or 12 freedom spaces may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 9 freedom space is generated.
- Figure A.3O and α and β equal 90 degrees, the 3 DOF Type 9 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 9 or 12 freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 9 freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 9 freedom space is generated.
- Figure A.3P and α and β equal 90 degrees, the 3 DOF Type 9 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 9 freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 9 freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 9 or 12 freedom spaces may be generated.

For condition (6) where p_1 and p_2 are different, non-zero, finite values with opposite signs, if p_3 is equal to infinity from

- Figure A.3A, the 3 DOF Type 5A freedom space is generated.
- Figure A.3B, the 3 DOF Type 5A freedom space is generated.
- Figure A.3C, the 3 DOF Type 4A or 5A freedom spaces may be generated.
- Figure A.3D, the 3 DOF Type 5A freedom space is generated.
- Figure A.3E, the 3 DOF Type 5A freedom space is generated.
- Figure A.3F, the 3 DOF Type 4A or 5A freedom spaces may be generated.
- Figure A.3G, the 3 DOF Type 5A freedom space is generated.
- Figure A.3H, the 3 DOF Type 5A freedom space is generated.
- Figure A.3I, the 3 DOF Type 4A or 5A freedom spaces may be generated.
- Figure A.3J, the 3 DOF Type 5A freedom space is generated.
- Figure A.3K, the 3 DOF Type 5A freedom space is generated.
- Figure A.3L, the 3 DOF Type 4A or 5A freedom spaces may be generated.
- Figure A.3M and α and β equal 90 degrees, the 3 DOF Type 6 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 5B freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 5B freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 4B or 5B freedom spaces may be generated.
- Figure A.3N and α and β equal 90 degrees, the 3 DOF Type 6 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 5B freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 5B freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 4B or 5B freedom spaces may be generated.
- Figure A.3O and α and β equal 90 degrees, the 3 DOF Type 6 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 5B freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 5B freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 4B or 5B freedom spaces may be generated.

- Figure A.3P and α and β equal 90 degrees, the 3 DOF Type 6 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 5B freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 5B freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 4B or 5B freedom spaces may be generated.

if p_3 is equal to zero from

- Figure A.3A, the 3 DOF Type 5A freedom space is generated.
- Figure A.3B, the 3 DOF Type 5A freedom space is generated.
- Figure A.3C, the 3 DOF Type 4A or 5A freedom spaces may be generated.
- Figure A.3D, the 3 DOF Type 5B freedom space is generated.
- Figure A.3E, the 3 DOF Type 5B freedom space is generated.
- Figure A.3F, the 3 DOF Type 5B freedom space is generated.
- Figure A.3G, the 3 DOF Type 4A or 5A freedom spaces may be generated.
- Figure A.3H, the 3 DOF Type 4A or 5A freedom spaces may be generated.
- Figure A.3I, the 2 DOF Type 3 or 3 DOF Type 5A freedom spaces may be generated.
- Figure A.3J, the 3 DOF Type 4B or 5B freedom spaces may be generated.
- Figure A.3K, the 3 DOF Type 4B or 5B freedom spaces may be generated.
- Figure A.3L, the 3 DOF Type 4B, 5B, or 6 freedom spaces may be generated.
- Figure A.3M and α and β equal 90 degrees, the 3 DOF Type 9 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 7 or 8 freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 7 or 8 freedom spaces may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated.
- Figure A.3N and α and β equal 90 degrees, the 3 DOF Type 7 or 8 freedom spaces may be generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated.

- Figure A.3O and α and β equal 90 degrees, the 3 DOF Type 7 or 8 freedom spaces may be generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated.
- Figure A.3P and α and β equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated.

if p_3 is equal to p_1 from

- Figure A.3A, the 2 DOF Type 3 freedom space is generated.
- Figure A.3B, the 3 DOF Type 5A freedom space is generated.
- Figure A.3C, the 3 DOF Type 5A freedom space is generated.
- Figure A.3D, the 3 DOF Type 6 freedom space is generated.
- Figure A.3E, the 3 DOF Type 5B freedom space is generated.
- Figure A.3F, the 3 DOF Type 5B freedom space is generated.
- Figure A.3G, the 3 DOF Type 5A freedom space is generated.
- Figure A.3H, the 3 DOF Type 4A or 5A freedom spaces may be generated.
- Figure A.3I, the 3 DOF Type 4A or 5A freedom spaces may be generated.
- Figure A.3J, the 3 DOF Type 5B freedom space is generated.
- Figure A.3K, the 3 DOF Type 5B freedom space is generated.
- Figure A.3L, the 3 DOF Type 5B freedom space is generated.
- Figure A.3M and α and β equal 90 degrees, the 3 DOF Type 7 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 7 freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 8 freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 8 freedom space is generated.

- Figure A.3N and α and β equal 90 degrees, the 3 DOF Type 8 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 7 or 8 freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 7 or 8 freedom spaces may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 7 or 8 freedom spaces may be generated.
- Figure A.3O and α and β equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom space may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated.
- Figure A.3P and α and β equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom space may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated.

if p_3 is equal to p_2 from

- Figure A.3A, the 3 DOF Type 5A freedom space is generated.
- Figure A.3B, the 2 DOF Type 3 freedom space is generated.
- Figure A.3C, the 3 DOF Type 5A freedom space is generated.
- Figure A.3D, the 3 DOF Type 5B freedom space is generated.
- Figure A.3E, the 3 DOF Type 6 freedom space is generated.
- Figure A.3F, the 3 DOF Type 5B freedom space is generated.
- Figure A.3G, the 3 DOF Type 4A or 5A freedom spaces may be generated.
- Figure A.3H, the 3 DOF Type 5A freedom space is generated.
- Figure A.3I, the 3 DOF Type 4A or 5A freedom spaces may be generated.
- Figure A.3J, the 3 DOF Type 5B freedom space is generated.
- Figure A.3K, the 3 DOF Type 5B freedom space is generated.
- Figure A.3L, the 3 DOF Type 5B freedom space is generated.

- Figure A.3M and α and β equal 90 degrees, the 3 DOF Type 7 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 7 freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 8 freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 8 freedom space is generated.
- Figure A.3N and α and β equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom space may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated.
- Figure A.3O and α and β equal 90 degrees, the 3 DOF Type 8 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 7 or 8 freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 7 or 8 freedom spaces may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 7 or 8 freedom spaces may be generated.
- Figure A.3P and α and β equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom space may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated.

if p_3 is a finite, non-zero value that is not equal to p_1 or p_2 from

- Figure A.3A, the 3 DOF Type 5A freedom space is generated.
- Figure A.3B, the 3 DOF Type 5A freedom space is generated.
- Figure A.3C, the 3 DOF Type 4A or 5A freedom spaces may be generated.
- Figure A.3D, the 3 DOF Type 5B freedom space is generated.
- Figure A.3E, the 3 DOF Type 5B freedom space is generated.
- Figure A.3F, the 3 DOF Type 4B or 5B freedom spaces may be generated.
- Figure A.3G, the 3 DOF Type 4A or 5A freedom spaces may be generated.
- Figure A.3H, the 3 DOF Type 4A or 5A freedom spaces may be generated.

- Figure A.3I, the 3 DOF Type 4A or 5A freedom spaces may be generated.
- Figure A.3J, the 3 DOF Type 4B or 5B freedom spaces may be generated.
- Figure A.3K, the 3 DOF Type 4B or 5B freedom spaces may be generated.
- Figure A.3L, the 3 DOF Type 4B, 5B, or 6 freedom spaces may be generated.
- Figure A.3M and α and β equal 90 degrees, the 3 DOF Type 8 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 7, 8 or 9 freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom space is generated.
- Figure A.3N and α and β equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom space may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated.
- Figure A.3O and α and β equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom space may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated.
- Figure A.3P and α and β equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom space may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated.

For condition (7) where p_1 and p_2 are the same, finite, non-zero value,

if p_3 is equal to infinity from

- Figure A.3A, the 3 DOF Type 5A freedom space is generated.

- Figure A.3B, the 3 DOF Type 5A freedom space is generated.
- Figure A.3C, the 3 DOF Type 5A freedom space is generated.
- Figure A.3D, the 3 DOF Type 5A freedom space is generated.
- Figure A.3E, the 3 DOF Type 5A freedom space is generated.
- Figure A.3F, the 3 DOF Type 5A freedom space is generated.
- Figure A.3G, the 3 DOF Type 5A freedom space is generated.
- Figure A.3H, the 3 DOF Type 5A freedom space is generated.
- Figure A.3I, the 3 DOF Type 5A freedom space is generated.
- Figure A.3J, the 3 DOF Type 5A freedom space is generated.
- Figure A.3K, the 3 DOF Type 5A freedom space is generated.
- Figure A.3L, the 3 DOF Type 5A freedom space is generated.
- Figure A.3M and α and β equal 90 degrees, the 3 DOF Type 17 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 5B freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 5B freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 5B freedom space is generated.
- Figure A.3N and α and β equal 90 degrees, the 3 DOF Type 17 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 5B freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 5B freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 5B freedom space is generated.
- Figure A.3O and α and β equal 90 degrees, the 3 DOF Type 17 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 5B freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 5B freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 5B freedom space is generated.
- Figure A.3P and α and β equal 90 degrees, the 3 DOF Type 17 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 5B freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 5B freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 5B freedom space is generated.

if p_3 is equal to zero from

- Figure A.3A, the 3 DOF Type 5A freedom space is generated.
- Figure A.3B, the 3 DOF Type 5A freedom space is generated.
- Figure A.3C, the 3 DOF Type 5A freedom space is generated.
- Figure A.3D, the 3 DOF Type 5B freedom space is generated.
- Figure A.3E, the 3 DOF Type 5B freedom space is generated.
- Figure A.3F, the 3 DOF Type 5B freedom space is generated.
- Figure A.3G, the 3 DOF Type 5A freedom space is generated.
- Figure A.3H, the 3 DOF Type 5A freedom space is generated.
- Figure A.3I, the 3 DOF Type 5A freedom space is generated.
- Figure A.3J, the 3 DOF Type 5B freedom space is generated.
- Figure A.3K, the 3 DOF Type 5B freedom space is generated.
- Figure A.3L, the 3 DOF Type 5B freedom space is generated.
- Figure A.3M and α and β equal 90 degrees, the 3 DOF Type 13 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 8 freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 8 freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 8 freedom space is generated.
- Figure A.3N and α and β equal 90 degrees, the 3 DOF Type 8 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 8 freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 7, 8, or 10 freedom spaces may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 8 freedom space is generated.
- Figure A.3O and α and β equal 90 degrees, the 3 DOF Type 8 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 7, 8, or 10 freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 8 freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 8 freedom space is generated.
- Figure A.3P and α and β equal 90 degrees, the 3 DOF Type 8 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 8

freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 8 freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 7, 8, or 10 freedom spaces may be generated.

if p_3 is equal to p_1 and p_2 from

- Figure A.3A, the 2 DOF Type 4 freedom space is generated.
- Figure A.3B, the 2 DOF Type 4 freedom space is generated.
- Figure A.3C, the 2 DOF Type 4 freedom space is generated.
- Figure A.3D, the 3 DOF Type 17 freedom space is generated.
- Figure A.3E, the 3 DOF Type 17 freedom space is generated.
- Figure A.3F, the 3 DOF Type 17 freedom space is generated.
- Figure A.3G, the 3 DOF Type 5A freedom space is generated.
- Figure A.3H, the 3 DOF Type 5A freedom space is generated.
- Figure A.3I, the 3 DOF Type 5A freedom space is generated.
- Figure A.3J, the 3 DOF Type 5B freedom space is generated.
- Figure A.3K, the 3 DOF Type 5B freedom space is generated.
- Figure A.3L, the 3 DOF Type 5B freedom space is generated.
- Figure A.3M and α and β equal 90 degrees, the 3 DOF Type 19 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 19 freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 19 freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 19 freedom space is generated.
- Figure A.3N and α and β equal 90 degrees, the 3 DOF Type 8, 10, or 11 freedom spaces may be generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 8, 10, or 11 freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 8, 10, or 11 freedom spaces may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 8, 10, or 11 freedom spaces may be generated.
- Figure A.3O and α and β equal 90 degrees, the 3 DOF Type 8, 10, or 11 freedom spaces may be generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 8, 10, or 11 freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 8, 10, or 11 freedom spaces may be

generated. If α and β do not equal 90 degrees, the 3 DOF Type 8, 10, or 11 freedom spaces may be generated.

- Figure A.3P and α and β equal 90 degrees, the 3 DOF Type 8, 10, or 11 freedom spaces may be generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 8, 10, or 11 freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 8, 10, or 11 freedom spaces may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 8, 10, or 11 freedom spaces may be generated.

if p_3 is a finite, non-zero value not equal to p_1 and p_2 from

- Figure A.3A, the 3 DOF Type 5A freedom space is generated.
- Figure A.3B, the 3 DOF Type 5A freedom space is generated.
- Figure A.3C, the 3 DOF Type 5A freedom space is generated.
- Figure A.3D, the 3 DOF Type 5B freedom space is generated.
- Figure A.3E, the 3 DOF Type 5B freedom space is generated.
- Figure A.3F, the 3 DOF Type 5B freedom space is generated.
- Figure A.3G, the 3 DOF Type 5A freedom space is generated.
- Figure A.3H, the 3 DOF Type 5A freedom space is generated.
- Figure A.3I, the 3 DOF Type 5A freedom space is generated.
- Figure A.3J, the 3 DOF Type 5B freedom space is generated.
- Figure A.3K, the 3 DOF Type 5B freedom space is generated.
- Figure A.3L, the 3 DOF Type 5B freedom space is generated.
- Figure A.3M and α and β equal 90 degrees, the 3 DOF Type 7 or 14 freedom spaces may be generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 8, 10 , or 11 freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 8, 10, or 11 freedom spaces may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 8, 10, or 11 freedom spaces may be generated.
- Figure A.3N and α and β equal 90 degrees, the 3 DOF Type , 8, 10, or 11 freedom spaces may be generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 8, 10 or 11 freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 7, 8, 10, 11, 13 or 14 freedom spaces

may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 8, 10 or 11 freedom spaces may be generated.

- Figure A.3O and α and β equal 90 degrees, the 3 DOF Type 8, 10, or 11 freedom spaces may be generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 7, 8, 10, 11, 13, or 14 freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 8, 10, or 11 freedom spaces may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 8, 10, or 11 freedom spaces may be generated.
- Figure A.3P and α and β equal 90 degrees, the 3 DOF Type 8, 10, or 11 freedom spaces may be generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 8, 10, or 11 freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 8, 10, or 11 freedoms space may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 7, 8, 10, 11, 13, or 14 freedom spaces may be generated.

For condition (8) where p_1 is equal to zero and p_2 is a finite and non-zero value,

if p_3 is equal to infinity from

- Figure A.3A, the 3 DOF Type 5A freedom space is generated.
- Figure A.3B, the 3 DOF Type 4A freedom space is generated.
- Figure A.3C, the 3 DOF Type 5A freedom space is generated.
- Figure A.3D, the 3 DOF Type 5A freedom space is generated.
- Figure A.3E, the 3 DOF Type 4A freedom space is generated.
- Figure A.3F, the 3 DOF Type 5A freedom space is generated.
- Figure A.3G, the 3 DOF Type 5A freedom space is generated.
- Figure A.3H, the 3 DOF Type 4A freedom space is generated.
- Figure A.3I, the 3 DOF Type 5A freedom space is generated.
- Figure A.3J, the 3 DOF Type 5A freedom space is generated.
- Figure A.3K, the 3 DOF Type 4A freedom space is generated.
- Figure A.3L, the 3 DOF Type 5A freedom space is generated.

- Figure A.3M and α and β equal 90 degrees, the 3 DOF Type 15 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 5B freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 4B freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 5B freedom space is generated.
- Figure A.3N and α and β equal 90 degrees, the 3 DOF Type 15 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 5B freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 4B freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 5B freedom space is generated.
- Figure A.3O and α and β equal 90 degrees, the 3 DOF Type 15 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 5B freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 4B freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 5B freedom space is generated.
- Figure A.3P and α and β equal 90 degrees, the 3 DOF Type 15 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 5B freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 4B freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 5B freedom space is generated.

if p_3 is equal to zero from

- Figure A.3A, the 2 DOF Type 6 freedom space is generated.
- Figure A.3B, the 3 DOF Type 4A freedom space is generated.
- Figure A.3C, the 3 DOF Type 4A freedom space is generated.
- Figure A.3D, the 3 DOF Type 15 freedom space is generated.
- Figure A.3E, the 3 DOF Type 4B freedom space is generated.
- Figure A.3F, the 3 DOF Type 4B freedom space is generated.
- Figure A.3G, the 3 DOF Type 4A freedom space is generated.
- Figure A.3H, the 3 DOF Type 4A or 5A freedom spaces may be generated.
- Figure A.3I, the 3 DOF Type 4A or 5A freedom spaces may be generated.
- Figure A.3J, the 3 DOF Type 4B freedom space is generated.

- Figure A.3K, the 3 DOF Type 4B or 5B freedom spaces may be generated.
- Figure A.3L, the 3 DOF Type 4B or 5B freedom spaces may be generated.
- Figure A.3M and α and β equal 90 degrees, the 3 DOF Type 12 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 12 freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 9 freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 9 freedom space is generated.
- Figure A.3N and α and β equal 90 degrees, the 3 DOF Type 9 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 9 freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 9 or 12 freedom spaces may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 9 or 12 freedom spaces may be generated.
- Figure A.3O and α and β equal 90 degrees, the 3 DOF Type 7 or 8 freedom spaces may be generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated.
- Figure A.3P and α and β equal 90 degrees, the 3 DOF Type 7 or 8 freedom spaces may be generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 7 or 8 freedom spaces may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated.

if p_3 is equal to p_2 from

- Figure A.3A, the 3 DOF Type 5A freedom space is generated.
- Figure A.3B, the 2 DOF Type 6 freedom space is generated.
- Figure A.3C, the 3 DOF Type 5A freedom space is generated.
- Figure A.3D, the 3 DOF Type 5B freedom space is generated.
- Figure A.3E, the 3 DOF Type 15 freedom space is generated.
- Figure A.3F, the 3 DOF Type 5B freedom space is generated.

- Figure A.3G, the 3 DOF Type 5A freedom space is generated.
- Figure A.3H, the 3 DOF Type 5A freedom space is generated.
- Figure A.3I, the 3 DOF Type 4A or 5A freedom spaces may be generated.
- Figure A.3J, the 3 DOF Type 4B or 5B freedom spaces may be generated.
- Figure A.3K, the 3 DOF Type 4B or 5B freedom spaces may be generated.
- Figure A.3L, the 3 DOF Type 4B or 5B freedom spaces may be generated.
- Figure A.3M and α and β equal 90 degrees, the 3 DOF Type 13 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 8 freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 13 freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 8 freedom space is generated.
- Figure A.3N and α and β equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated.
- Figure A.3O and α and β equal 90 degrees, the 3 DOF Type 8 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 7, 8, or 10 freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 8 freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 7, 8, or 10 freedom spaces may be generated.
- Figure A.3P and α and β equal 90 degrees, the 3 DOF Type 7, 8, 9 or 10 freedom spaces may be generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 7, 8, 9 or 10 freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 7, 8, 9 or 10 freedom spaces may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 7, 8, 9, or 10 freedom spaces may be generated.

if p_3 is a finite, non-zero value that is not equal to p_2 from

- Figure A.3A, the 3 DOF Type 5A freedom space is generated.
- Figure A.3B, the 3 DOF Type 4A freedom space is generated.

- Figure A.3C, the 3 DOF Type 4A or 5A freedom spaces may be generated.
- Figure A.3D, the 3 DOF Type 5B freedom space is generated.
- Figure A.3E, the 3 DOF Type 4B freedom space is generated.
- Figure A.3F, the 3 DOF Type 4B or 5B freedom spaces may be generated.
- Figure A.3G, the 3 DOF Type 4A or 5A freedom spaces may be generated.
- Figure A.3H, the 3 DOF Type 4A or 5A freedom spaces may be generated.
- Figure A.3I, the 2 DOF Type 6 or 3 DOF Type 4A or 5A freedom spaces may be generated.
- Figure A.3J, the 3 DOF Type 4B or 5B freedom spaces may be generated.
- Figure A.3K, the 3 DOF Type 4B or 5B freedom spaces may be generated.
- Figure A.3L, the 3 DOF Type 4B, 5B, or 15 freedom spaces may be generated.
- Figure A.3M and α and β equal 90 degrees, the 3 DOF Type 9 or 10 freedom spaces may be generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 7, 8 , or 9 freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 7, 8, 9, or 10 freedom spaces may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated.
- Figure A.3N and α and β equal 90 degrees, the 3 DOF Type 7, 8, 9, or 10 freedom spaces may be generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 7, 8, 9, or 10 freedom spaces may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated.
- Figure A.3O and α and β equal 90 degrees, the 3 DOF Type 7 or 8 freedom spaces may be generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 7, 8, 9, or 10 freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated.
- Figure A.3P and α and β equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF

Type 7, 8, 9, or 10 freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom space may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 7, 8, 9, 10, 12, or 13 freedom spaces may be generated.

For condition (9) where p_1 and p_2 are different, non-zero, finite values with the same sign, if p_3 is equal to infinity from

- Figure A.3A, the 3 DOF Type 5A freedom space is generated.
- Figure A.3B, the 3 DOF Type 5A freedom space is generated.
- Figure A.3C, the 3 DOF Type 4A or 5A freedom spaces may be generated.
- Figure A.3D, the 3 DOF Type 5A freedom space is generated.
- Figure A.3E, the 3 DOF Type 5A freedom space is generated.
- Figure A.3F, the 3 DOF Type 4A or 5A freedom spaces may be generated.
- Figure A.3G, the 3 DOF Type 5A freedom space is generated.
- Figure A.3H, the 3 DOF Type 5A freedom space is generated.
- Figure A.3I, the 3 DOF Type 4A or 5A freedom spaces may be generated.
- Figure A.3J, the 3 DOF Type 5A freedom space is generated.
- Figure A.3K, the 3 DOF Type 5A freedom space is generated.
- Figure A.3L, the 3 DOF Type 4A or 5A freedom spaces may be generated.
- Figure A.3M and α and β equal 90 degrees, the 3 DOF Type 16 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 5B freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 5B freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 4B or 5B freedom spaces may be generated.
- Figure A.3N and α and β equal 90 degrees, the 3 DOF Type 16 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 5B freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 5B freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 4B or 5B freedom spaces may be generated.

- Figure A.3O and α and β equal 90 degrees, the 3 DOF Type 16 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 5B freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 5B freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 4B or 5B freedom spaces may be generated.
- Figure A.3P and α and β equal 90 degrees, the 3 DOF Type 16 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 5B freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 5B freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 4B or 5B freedom spaces may be generated.

if p_3 is equal to zero from

- Figure A.3A, the 3 DOF Type 5A freedom space is generated.
- Figure A.3B, the 3 DOF Type 5A freedom space is generated.
- Figure A.3C, the 3 DOF Type 4A or 5A freedom spaces may be generated.
- Figure A.3D, the 3 DOF Type 5B freedom space is generated.
- Figure A.3E, the 3 DOF Type 5B freedom space is generated.
- Figure A.3F, the 3 DOF Type 4B or 5B freedom spaces may be generated.
- Figure A.3G, the 3 DOF Type 5A freedom space is generated.
- Figure A.3H, the 3 DOF Type 5A freedom space is generated.
- Figure A.3I, the 3 DOF Type 5A freedom space is generated.
- Figure A.3J, the 3 DOF Type 4B or 5B freedom spaces may be generated.
- Figure A.3K, the 3 DOF Type 4B or 5B freedom spaces may be generated.
- Figure A.3L, the 3 DOF Type 4B or 5B freedom spaces may be generated.
- Figure A.3M and α and β equal 90 degrees, the 3 DOF Type 10 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated.
- Figure A.3N and α and β equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF

Type 7, 8, or 9 freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 7, 8, 9, 10, or 12 freedom spaces may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated.

- Figure A.3O and α and β equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 7, 8, 9, 10, or 12 freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated.
- Figure A.3P and α and β equal 90 degrees, the 3 DOF Type 7 or 8 freedom spaces may be generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated.

if p_3 is equal to p_1 from

- Figure A.3A, the 2 DOF Type 7 freedom space is generated.
- Figure A.3B, the 3 DOF Type 5A freedom space is generated.
- Figure A.3C, the 3 DOF Type 5A freedom space is generated.
- Figure A.3D, the 3 DOF Type 16 freedom space is generated.
- Figure A.3E, the 3 DOF Type 5B freedom space is generated.
- Figure A.3F, the 3 DOF Type 5B freedom space is generated.
- Figure A.3G, the 3 DOF Type 5A freedom space is generated.
- Figure A.3H, the 3 DOF Type 5A freedom space is generated.
- Figure A.3I, the 3 DOF Type 5A freedom space is generated.
- Figure A.3J, the 3 DOF Type 5B freedom space is generated.
- Figure A.3K, the 3 DOF Type 4B or 5B freedom spaces may be generated.
- Figure A.3L, the 3 DOF Type 4B or 5B freedom spaces may be generated.
- Figure A.3M and α and β equal 90 degrees, the 3 DOF Type 14 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type

14 freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 8, 10, or 11 freedom spaces may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 8, 10, or 11 freedom spaces may be generated.

- Figure A.3N and α and β equal 90 degrees, the 3 DOF Type 8, 10, or 11 freedom spaces may be generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 8, 10, or 11 freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 7, 8, 10, 11, 13, or 14 freedom spaces may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 7, 8, 10, 11, 13, or 14 freedom spaces may be generated.
- Figure A.3O and α and β equal 90 degrees, the 3 DOF Type 7, 8, 10, 11, 13, or 14 freedom spaces may be generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 7, 8, 10, 11, 13, or 14 freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 7, 8, 9, 10, 11, 12, 13, or 14 freedom space may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 7, 8, 9, 10, 11, 12, 13, or 14 freedom spaces may be generated.
- Figure A.3P and α and β equal 90 degrees, the 3 DOF Type 7, 8, 9, 10, 11, 12, 13, or 14 freedom spaces may be generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 7, 8, 9, 10, 11, 12, 13, or 14 freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 7, 8, 9, 10, 11, 12, 13, or 14 freedom spaces may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 7, 8, 9, 10, 11, 12, 13, or 14 freedom spaces may be generated.

if p_3 is equal to p_2 from

- Figure A.3A, the 3 DOF Type 5A freedom space is generated.
- Figure A.3B, the 2 DOF Type 7 freedom space is generated.
- Figure A.3C, the 3 DOF Type 5A freedom space is generated.
- Figure A.3D, the 3 DOF Type 5B freedom space is generated.
- Figure A.3E, the 3 DOF Type 16 freedom space is generated.
- Figure A.3F, the 3 DOF Type 5B freedom space is generated.
- Figure A.3G, the 3 DOF Type 5A freedom space is generated.

- Figure A.3H, the 3 DOF Type 5A freedom space is generated.
- Figure A.3I, the 3 DOF Type 5A freedom space is generated.
- Figure A.3J, the 3 DOF Type 4B or 5B freedom spaces may be generated.
- Figure A.3K, the 3 DOF Type 5B freedom space is generated.
- Figure A.3L, the 3 DOF Type 4B or 5B freedom spaces may be generated.
- Figure A.3M and α and β equal 90 degrees, the 3 DOF Type 14 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 8, 10, or 11 freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 14 freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 8, 10, or 11 freedom spaces may be generated.
- Figure A.3N and α and β equal 90 degrees, the 3 DOF Type 7, 8, 10, 11, 13, or 14 freedom spaces may be generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 7, 8, 9, 10, 11, 12, 13, or 14 freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 7, 8, 10, 11, 13, or 14 freedom spaces may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 7, 8, 9, 10, 11, 12, 13, or 14 freedom spaces may be generated.
- Figure A.3O and α and β equal 90 degrees, the 3 DOF Type 8, 10, or 11 freedom spaces may be generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 7, 8, 10, 11, 13, or 14 freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 8, 10, or 11 freedom spaces may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 7, 8, 10, 11, 13, or 14 freedom spaces may be generated.
- Figure A.3P and α and β equal 90 degrees, the 3 DOF Type 7, 8, 9, 10, 11, 12, 13, or 14 freedom spaces may be generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 7, 8, 9, 10, 11, 12, 13, or 14 freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 7, 8, 9, 10, 11, 12, 13, or 14 freedom spaces may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 7, 8, 9, 10, 11, 12, 13, or 14 freedom spaces may be generated.

if p_3 is a finite, non-zero value that is not equal to p_1 or p_2 but has a different sign from

- Figure A.3A, the 3 DOF Type 5A freedom space is generated.
- Figure A.3B, the 3 DOF Type 5A freedom space is generated.
- Figure A.3C, the 3 DOF Type 4A or 5A freedom spaces may be generated.
- Figure A.3D, the 3 DOF Type 5B freedom space is generated.
- Figure A.3E, the 3 DOF Type 5B freedom space is generated.
- Figure A.3F, the 3 DOF Type 4B or 5B freedom spaces may be generated.
- Figure A.3G, the 3 DOF Type 4A or 5A freedom spaces may be generated.
- Figure A.3H, the 3 DOF Type 4A or 5A freedom spaces may be generated.
- Figure A.3I, the 3 DOF Type 4A or 5A freedom spaces may be generated.
- Figure A.3J, the 3 DOF Type 4B or 5B freedom spaces may be generated.
- Figure A.3K, the 3 DOF Type 4B or 5B freedom spaces may be generated.
- Figure A.3L, the 3 DOF Type 4B or 5B freedom spaces may be generated.
- Figure A.3M and α and β equal 90 degrees, the 3 DOF Type 8 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 7, 8 or 9 freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom space is generated.
- Figure A.3N and α and β equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated.
- Figure A.3O and α and β equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated.

- Figure A.3P and α and β equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 7, 8, or 9 freedom spaces may be generated.

if p_3 is a finite, non-zero value that is not equal to p_1 or p_2 but has the same sign from

- Figure A.3A, the 3 DOF Type 5A freedom space is generated.
- Figure A.3B, the 3 DOF Type 5A freedom space is generated.
- Figure A.3C, the 3 DOF Type 4A or 5A freedom spaces may be generated.
- Figure A.3D, the 3 DOF Type 4B or 5B freedom spaces may be generated.
- Figure A.3E, the 3 DOF Type 4B or 5B freedom spaces may be generated.
- Figure A.3F, the 3 DOF Type 4B or 5B freedom spaces may be generated.
- Figure A.3G, the 3 DOF Type 4A or 5A freedom spaces may be generated.
- Figure A.3H, the 3 DOF Type 4A or 5A freedom spaces may be generated.
- Figure A.3I, the 2 DOF Type 7 or 3 DOF Type 4A or 5A freedom spaces may be generated.
- Figure A.3J, the 3 DOF Type 4B or 5B freedom spaces may be generated.
- Figure A.3K, the 3 DOF Type 4B or 5B freedom spaces may be generated.
- Figure A.3L, the 3 DOF Type 4B, 5B, or 16 freedom spaces may be generated.
- Figure A.3M and α and β equal 90 degrees, the 3 DOF Type 11 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 7, 8, 9, 10, 11, 12, 13, or 14 freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 7, 8, 9, 10, 11, 12, 13, or 14 freedom spaces may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 7, 8, 9, 10, 11, 12, 13, or 14 freedom spaces may be generated.
- Figure A.3N and α and β equal 90 degrees, the 3 DOF Type 7, 8, 9, 10, 11, 12, 13, or 14 freedom spaces may be generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 7, 8, 9, 10, 11, 12, 13, or 14 freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 7, 8, 9, 10, 11, 12, 13, or 14 freedom spaces may be generated. If α and β do not equal

90 degrees, the 3 DOF Type 7, 8, 9, 10, 11, 12, 13, or 14 freedom spaces may be generated.

- Figure A.3O and α and β equal 90 degrees, the 3 DOF Type 7, 8, 9, 10, 11, 12, 13, or 14 freedom spaces may be generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 7, 8, 9, 10, 11, 12, 13, or 14 freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 7, 8, 9, 10, 11, 12, 13, or 14 freedom spaces may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 7, 8, 9, 10, 11, 12, 13, or 14 freedom spaces may be generated.
- Figure A.3P and α and β equal 90 degrees, the 3 DOF Type 7, 8, 9, 10, 11, 12, 13, or 14 freedom spaces may be generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 7, 8, 9, 10, 11, 12, 13, or 14 freedom spaces may be generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 7, 8, 9, 10, 11, 12, 13, or 14 freedom spaces may be generated. If α and β do not equal 90 degrees, the 3 DOF Type 7, 8, 9, 10, 11, 12, 13, or 14 freedom spaces may be generated.

For condition (10) where p_1 and p_2 are equal to infinity,

if p_3 is equal to infinity from

- Figure A.3A, the 2 DOF Type 10 freedom space is generated.
- Figure A.3B, the 2 DOF Type 10 freedom space is generated.
- Figure A.3C, the 2 DOF Type 10 freedom space is generated.
- Figure A.3D, the 2 DOF Type 10 freedom space is generated.
- Figure A.3E, the 2 DOF Type 10 freedom space is generated.
- Figure A.3F, the 2 DOF Type 10 freedom space is generated.
- Figure A.3G, the 2 DOF Type 10 freedom space is generated.
- Figure A.3H, the 2 DOF Type 10 freedom space is generated.
- Figure A.3I, the 2 DOF Type 10 freedom space is generated.
- Figure A.3J, the 2 DOF Type 10 freedom space is generated.
- Figure A.3K, the 2 DOF Type 10 freedom space is generated.

- Figure A.3L, the 2 DOF Type 10 freedom space is generated.
- Figure A.3M and α and β equal 90 degrees, the 3 DOF Type 20 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 20 freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 20 freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 20 freedom space is generated.
- Figure A.3N and α and β equal 90 degrees, the 3 DOF Type 20 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 20 freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 20 freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 20 freedom space is generated.
- Figure A.3O and α and β equal 90 degrees, the 3 DOF Type 20 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 20 freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 20 freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 20 freedom space is generated.
- Figure A.3P and α and β equal 90 degrees, the 3 DOF Type 20 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 20 freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 20 freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 20 freedom space is generated.

if p_3 is equal to zero from

- Figure A.3A, the 3 DOF Type 21 freedom space is generated.
- Figure A.3B, the 3 DOF Type 21 freedom space is generated.
- Figure A.3C, the 3 DOF Type 21 freedom space is generated.
- Figure A.3D, the 3 DOF Type 21 freedom space is generated.
- Figure A.3E, the 3 DOF Type 21 freedom space is generated.
- Figure A.3F, the 3 DOF Type 21 freedom space is generated.
- Figure A.3G, the 3 DOF Type 21 freedom space is generated.
- Figure A.3H, the 3 DOF Type 21 freedom space is generated.
- Figure A.3I, the 3 DOF Type 21 freedom space is generated.

- Figure A.3J, the 3 DOF Type 21 freedom space is generated.
- Figure A.3K, the 3 DOF Type 21 freedom space is generated.
- Figure A.3L, the 3 DOF Type 21 freedom space is generated.
- Figure A.3M and α and β equal 90 degrees, the 3 DOF Type 2 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 22 freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 22 freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 22 freedom space is generated.
- Figure A.3N and α and β equal 90 degrees, the 3 DOF Type 2 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 22 freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 22 freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 22 freedom space is generated.
- Figure A.3O and α and β equal 90 degrees, the 3 DOF Type 2 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 22 freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 22 freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 22 freedom space is generated.
- Figure A.3P and α and β equal 90 degrees, the 3 DOF Type 2 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 22 freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 22 freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 22 freedom space is generated.

if p_3 is a finite, non-zero value from

- Figure A.3A, the 3 DOF Type 21 freedom space is generated.
- Figure A.3B, the 3 DOF Type 21 freedom space is generated.
- Figure A.3C, the 3 DOF Type 21 freedom space is generated.
- Figure A.3D, the 3 DOF Type 21 freedom space is generated.
- Figure A.3E, the 3 DOF Type 21 freedom space is generated.
- Figure A.3F, the 3 DOF Type 21 freedom space is generated.
- Figure A.3G, the 3 DOF Type 21 freedom space is generated.

- Figure A.3H, the 3 DOF Type 21 freedom space is generated.
- Figure A.3I, the 3 DOF Type 21 freedom space is generated.
- Figure A.3J, the 3 DOF Type 21 freedom space is generated.
- Figure A.3K, the 3 DOF Type 21 freedom space is generated.
- Figure A.3L, the 3 DOF Type 21 freedom space is generated.
- Figure A.3M and α and β equal 90 degrees, the 3 DOF Type 18 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 22 freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 22 freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 22 freedom space is generated.
- Figure A.3N and α and β equal 90 degrees, the 3 DOF Type 18 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 22 freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 22 freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 22 freedom space is generated.
- Figure A.3O and α and β equal 90 degrees, the 3 DOF Type 18 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 22 freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 22 freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 22 freedom space is generated.
- Figure A.3P and α and β equal 90 degrees, the 3 DOF Type 18 freedom space is generated. If α does not equal 90 degrees but β equals 90 degrees, the 3 DOF Type 22 freedom space is generated. If α equals 90 degrees but β does not equal 90 degrees, the 3 DOF Type 22 freedom space is generated. If α and β do not equal 90 degrees, the 3 DOF Type 22 freedom space is generated.

Thus the 3 DOF column consists of 22 types. The 4 DOF column consists of ten types just like the 2 DOF column because the chart is symmetric about the 3 DOF column. Likewise, the 5 DOF column consists of three types and the 6 DOF column consists of one type similar to the 1 DOF column and 0 DOF column respectively.

Appendix B:
Actuation Space MATLAB Code

This appendix contains the MATLAB code for three functions. The first two functions construct and decompose twists and/or wrenches. The last function uses these two functions to generate the actuation space of a parallel flexure system. The user is asked to input information about the system's constraint topology, geometry and material properties. The computer then provides the user with the independent twists that make up the system's freedom space and asks the user to input the desired magnitudes of these twists. The computer then generates the independent wrenches that linearly combine to form the system's actuation space. The user is then asked to place actuators within the actuation space and to specify which DOFs he/she would like to actuate. The force and displacement actuator outputs are then provided to the user for actuating each of these DOFs. Finally, the third function returns the system's twist-wrench stiffness matrix.

```
function [T] = ConstructPlucker(c,w,p)
%Returns a twist vector given location vector, rotation direction vector,
%and pitch
T = [ w (cross(c,w)+p*w)];

function [c w p] = DecomposePlucker(T)
%Decomposes twist vectors into their pitch, location, and orientation
%vectors
%Input: Twist (6x1 matrix)
%Output: c=location vector w=orientation vector p=pitch
if(dot(T(1:3),T(1:3))<=1e-15)
    %pure translation
    c = 'There is no location associated with this plucker vector';
    w = [T(4) T(5) T(6)];  %Orientation vector defined
    p = 'Infinite';
else
    w = [T(1) T(2) T(3)];
    v = [T(4) T(5) T(6)];
    p = dot(w,v)/dot(w,w);
    c = [0 0 0];  %Initialize location vector
    if(w(1)==0 && w(2)==0)
        c(1) = -v(2)/w(3);
        c(2) = v(1)/w(3);
        c(3)= 0;
    elseif(w(1)==0 && w(3)==0)
```

```
        c(1) = v(3)/w(2);
        c(2) = 0;
        c(3) = -v(1)/w(2);
    elseif(w(2)==0 && w(3)==0)
        c(1) = 0;
        c(2) = -v(3)/w(1);
        c(3) = v(2)/w(1);
    elseif(w(1)==0)
        c(1) = (v(2)-p*w(2))/(-w(3));
        c(2) = v(1)/w(3);
        c(3)= 0;
    elseif(w(2)==0)
        c(1) = -v(2)/w(3);
        c(2) = (v(1)-p*w(1))/w(3);
        c(3)= 0;
    elseif(w(3)==0)
        c(1) = 0;
        c(2) = -v(3)/w(1);
        c(3) = (v(2)-p*w(2))/w(1);
    else
        c(1) = 0;
        c(2) = (v(3)-p*w(3))/(-w(1));
        c(3) = (v(2)-p*w(2))/w(1);
    end
    if(abs(c(1))>=1e5 && w(1)~=0)
        t=-c(1)/w(1);
        c=c+(w*t);
    elseif(abs(c(2))>=1e5 && w(2)~=0)
        t=-c(2)/w(2);
        c=c+(w*t);
    elseif(abs(c(3))>=1e5 && w(3)~=0)
        t=-c(3)/w(3);
        c=c+(w*t);
    end
end

function [K]=ActuationSpace()
%This code requires a user to input information about a parallel flexure
system's
%constraints.  The system's freedom space is calculated.  The stystem's
actuation
%space is calculated.  The user is asked to input actuators from within the
%actuation space as well as DOFs that they wish to actuate.  The system's
%actuator force and displacement outputs are calculated for actuating the
chosen
%DOFs. The function returns the systm's TWSM or K matrix
float(1e100, 1e100);
format long;
%The user is asked for the type and number of the constriants
cylinder = input('How many slender cylindrical constraints make up the
topology of the parallel flexure system?:  ');
square = input('How many slender square constraints make up the topology of
the parallel flexure system?:  ');
blade = input('How many flexure blades make up the topology of the parallel
flexure system?:  ');
C=(cylinder+square+(blade*3));
Topology=zeros(C,12);
```

```
%If the constraints are slender cylinders, the user is asked to input the
constraint characteristics
if(cylinder~=0)
    for i=1:cylinder
        r=input(strcat('Input the point where cylindrical constraint
#',int2str(i),' attaches to the rigid body:  '));
        f=input(strcat('Input the direction along cylindrical constraint
#',int2str(i),'s axis:  '));
        n=f/sqrt(dot(f,f));
        l=input(strcat('Input the length of cylindrical constraint
#',int2str(i),':  '));
        d=input(strcat('Input the diameter of cylindrical constraint
#',int2str(i),':  '));
        Topology(i,1:3)=r;
        Topology(i,4:6)=n;
        Topology(i,7)=l;
        Topology(i,8)=d;
    end
end
%If the constraints have a small square cross-section, the user is asked to
input the constraint characteristics
if(square~=0)
    for i=(1+cylinder):(cylinder+square)
        r=input(strcat('Input the point where square constraint #',int2str(i-
cylinder),' attaches to the rigid body:  '));
        f=input(strcat('Input the direction along square constraint
#',int2str(i-cylinder),'s axis:  '));
        n1=f/sqrt(dot(f,f));
        per=input(strcat('Input the direction perpendicular to the plane of
one of the side faces of square constraint #',int2str(i-cylinder),':  '));
        n2=per/sqrt(dot(per,per));
        l=input(strcat('Input the length of square constraint #',int2str(i-
cylinder),':  '));
        w=input(strcat('Input the width/thickness of square constraint
#',int2str(i-cylinder),':  '));
        Topology(i,1:3)=r;
        Topology(i,4:6)=n1;
        Topology(i,7)=l;
        Topology(i,8)=w;
        Topology(i,10:12)=n2;
    end
end
%If the constraints are blade flexures, the user is asked to input the
constraint characteristics
if(blade~=0)
    for i=(1+cylinder+square):(cylinder+square+blade)
        r=input(strcat('Input the central point where felxure blade
#',int2str(i-cylinder-square),' attaches to the rigid body:  '));
        f=input(strcat('Input the direction along the length of felxure blade
#',int2str(i-cylinder-square),':  '));
        n1=f/sqrt(dot(f,f));
        per=input(strcat('Input the direction perpendicular to the plane of
felxure blade #',int2str(i-cylinder-square),':  '));
        n2=per/sqrt(dot(per,per));
        l=input(strcat('Input the length of flexure blade #',int2str(i-
cylinder-square),':  '));
```

```
        w=input(strcat('Input the width of flexure blade #',int2str(i-
cylinder-square),':  '));
        t=input(strcat('Input the thickness of flexure blade #',int2str(i-
cylinder-square),':  '));
        Topology(i,1:3)=r;
        Topology(i+blade,1:3)=cross(n1,n2)+r;
        Topology(i+blade+blade,1:3)=r;
        Topology(i,4:6)=n1;
        Topology(i+blade,4:6)=n1;
        dir=cross(n1,n2)+n1;
        Topology(i+blade+blade,4:6)=dir/sqrt(dot(dir,dir));
        Topology(i,7)=l;
        Topology(i,8)=w;
        Topology(i,9)=t;
        Topology(i,10:12)=n2;
    end
end
E=input('Input the Youngs Modulus of the constraints material:  ');
G=input('Input the Shear Modulus of the constraints material:  ');
%The system's constraint space is calcuated as a wrench matrix
ConstraintSpace=zeros(C,6);
for i=1:C;

ConstraintSpace(i,1:6)=ConstructPlucker(Topology(i,1:3),Topology(i,4:6),0);
end
%The number of independent constraints is determined
[L U] = lu(ConstraintSpace);
for i=1:10;
    [L U] = lu(transpose(U));
end
[Row Col] = size(U);
indep=0;
for k=1:Row;
    if(dot(U(k,1:Col),U(k,1:Col)) ~= 0)
        indep=indep+1;
    end
end
%The system's freedom space is calculated and provided in the form of the
%system's independent decomposed twists
FreedomSpace=zeros(6,6-indep);
if(indep>=6)
    disp('The parallel flexure system is fully constrained and cannot move.')
else
    Complement=null(ConstraintSpace);
    FreedomSpace(1:3,1:(6-indep))=Complement(4:6,1:(6-indep));
    FreedomSpace(4:6,1:(6-indep))=Complement(1:3,1:(6-indep));

    disp(strcat('The freedom space of the parallel flexure system consists of
:',int2str(6-indep),' independent twists.'));
    for i=1:(6-indep);
        [c w p]=DecomposePlucker(FreedomSpace(1:6,i));
        FreedomSpace(1:6,i)=FreedomSpace(1:6,i)/sqrt(dot(w,w));
        [c w p]=DecomposePlucker(FreedomSpace(1:6,i));
        disp(strcat('A possible location vector for Twist #',int2str(i),' is:
'));
        disp(c);
        disp(strcat('Twist #',int2str(i),'s orientation vector is: '));
```

```
        disp(w);
        disp(strcat('Twist #',int2str(i),'s pitch is: '));
        disp(p);
    end
    for i=1:(6-indep);
        M=input(strcat('Input the desired magnitude for Twist
#',int2str(i),':  '));
        FreedomSpace(1:6,i)=M*FreedomSpace(1:6,i);
    end

    %The K matrix or TWSM of the system is constructed.
    %Note that the following is correct only if n3 points into the stage.
    %If n3 points into the constraint, ss(1,5) and ss(5,1) become positive
    %values and ss(2,4) and ss(4,2) become negative values
    K=zeros(6,6);
    if(cylinder~=0)
        for i=1:cylinder;
            N=zeros(6,6);
            n3=Topology(i,4:6);
            orth=null(n3);
            n2=transpose(orth(1:3,1));
            n2=n2/sqrt(dot(n2,n2));
            n1=cross(n2,n3);
            N(1:3,1)=transpose(n1);
            N(1:3,2)=transpose(n2);
            N(1:3,3)=transpose(n3);
            N(4:6,4)=transpose(n1);
            N(4:6,5)=transpose(n2);
            N(4:6,6)=transpose(n3);
            N(4:6,1)=transpose(cross(Topology(i,1:3),n1));
            N(4:6,2)=transpose(cross(Topology(i,1:3),n2));
            N(4:6,3)=transpose(cross(Topology(i,1:3),n3));
            NR=zeros(6,6);
            NR(1:6,1:3)=N(1:6,4:6);
            NR(1:6,4:6)=N(1:6,1:3);
            ss=zeros(6,6);
            L=Topology(i,7);
            D=Topology(i,8);
            I=(pi*(D^4))/64;
            J=(pi*(D^4))/32;
            A=(pi*(D^2))/4;
            ss(1,1)=L/(E*I);
            ss(1,5)=-(L^2)/(2*E*I);
            ss(2,2)=L/(E*I);
            ss(2,4)=(L^2)/(2*E*I);
            ss(3,3)=L/(G*J);
            ss(4,2)=(L^2)/(2*E*I);
            ss(4,4)=(L^3)/(3*E*I);
            ss(5,1)=-(L^2)/(2*E*I);
            ss(5,5)=(L^3)/(3*E*I);
            ss(6,6)=L/(E*A);
            S=inv(ss);
            K1=NR*S*inv(N);
            K=K+K1;
        end
    end
    if(square~=0)
```

```
    for i=(1+cylinder):(cylinder+square);
        N=zeros(6,6);
        n3=Topology(i,4:6);
        n2=Topology(i,10:12);
        n1=cross(n2,n3);
        N(1:3,1)=transpose(n1);
        N(1:3,2)=transpose(n2);
        N(1:3,3)=transpose(n3);
        N(4:6,4)=transpose(n1);
        N(4:6,5)=transpose(n2);
        N(4:6,6)=transpose(n3);
        N(4:6,1)=transpose(cross(Topology(i,1:3),n1));
        N(4:6,2)=transpose(cross(Topology(i,1:3),n2));
        N(4:6,3)=transpose(cross(Topology(i,1:3),n3));
        NR=zeros(6,6);
        NR(1:6,1:3)=N(1:6,4:6);
        NR(1:6,4:6)=N(1:6,1:3);
        ss=zeros(6,6);
        L=Topology(i,7);
        W=Topology(i,8);
        I=(W^4)/12;
        J=(W^4)/6;
        A=W^2;
        ss(1,1)=L/(E*I);
        ss(1,5)=-(L^2)/(2*E*I);
        ss(2,2)=L/(E*I);
        ss(2,4)=(L^2)/(2*E*I);
        ss(3,3)=L/(G*J);
        ss(4,2)=(L^2)/(2*E*I);
        ss(4,4)=(L^3)/(3*E*I);
        ss(5,1)=-(L^2)/(2*E*I);
        ss(5,5)=(L^3)/(3*E*I);
        ss(6,6)=L/(E*A);
        S=inv(ss);
        K1=NR*S*inv(N);
        K=K+K1;
    end
end
if(blade~=0)
    for i=(1+cylinder+square):(cylinder+square+blade);
        N=zeros(6,6);
        n3=Topology(i,4:6);
        n2=Topology(i,10:12);
        n1=cross(n2,n3);
        N(1:3,1)=transpose(n1);
        N(1:3,2)=transpose(n2);
        N(1:3,3)=transpose(n3);
        N(4:6,4)=transpose(n1);
        N(4:6,5)=transpose(n2);
        N(4:6,6)=transpose(n3);
        N(4:6,1)=transpose(cross(Topology(i,1:3),n1));
        N(4:6,2)=transpose(cross(Topology(i,1:3),n2));
        N(4:6,3)=transpose(cross(Topology(i,1:3),n3));
        NR=zeros(6,6);
        NR(1:6,1:3)=N(1:6,4:6);
        NR(1:6,4:6)=N(1:6,1:3);
        ss=zeros(6,6);
```

```
            L=Topology(i,7);
            W=Topology(i,8);
            T=Topology(i,9);
            I1=W*(T^3)/12;
            I2=T*(W^3)/12;
            J=I1+I2;
            A=W*T;
            ss(1,1)=L/(E*I1);
            ss(1,5)=-(L^2)/(2*E*I1);
            ss(2,2)=L/(E*I2);
            ss(2,4)=(L^2)/(2*E*I2);
            ss(3,3)=L/(G*J);
            ss(4,2)=(L^2)/(2*E*I2);
            ss(4,4)=(L^3)/(3*E*I2);
            ss(5,1)=-(L^2)/(2*E*I1);
            ss(5,5)=(L^3)/(3*E*I1);
            ss(6,6)=L/(E*A);
            S=inv(ss);
            K1=NR*S*inv(N);
            K=K+K1;
        end
    end
    %The actuation space of the system is calculated and provided in the
    %form of independent decomposed actuation wrenches
    ActuationSpace=K*FreedomSpace;
    disp(strcat('The resulting actuation space of the parallel flexure system
consists of :',int2str(6-indep),' independent wrenches.'));
    for i=1:(6-indep);
        [r f q]=DecomposePlucker(ActuationSpace(1:6,i));
        disp(strcat('A possible location vector for Actuation Wrench
#',int2str(i),' is: '));
        disp(r);
        disp(strcat('Actuation Wrench #',int2str(i),'s orientation vector is:
'));
        disp(f);
        disp(strcat('Actuation Wrench #',int2str(i),'s q value is: '));
        disp(q);
    end
    %The user is asked to select actuators from within the actuation space
    Actuators=zeros(6,(6-indep));
    Rlocation=zeros(3,(6-indep));
    Forientation=zeros(3,(6-indep));
    disp(strcat('Select  ' ,int2str(6-indep),' independent actuators from
within the actuation space.'));
    for i=1:(6-indep);
        Rlocation(:,i)=input(strcat('Input the point where actuator
#',int2str(i),' attaches to the rigid body:  '));
        f=input(strcat('Input the direction of actuator #',int2str(i),'s
axis:  '));
        Forientation(:,i)=f/sqrt(dot(f,f));
        q=input(strcat('Input the q-value of actuator #',int2str(i),':  '));

Actuators(1:6,i)=ConstructPlucker(transpose(Rlocation(:,i)),transpose(Forient
ation(:,i)),q);
    end
    %The user inputs the twists to be actuated by the chosen actuators.
    %The actuator magnitude and displacement outputs are calculated for
```

```
    %actuating the chosen twists.
    Num=input('Input the number of twists that you would like to actuate from 
within the freedom space:  ');
    for j=1:Num;
        disp(strcat('Select twist #',int2str(j),' from within the freedom 
space that you wish to actuate.'));
        c=input('Input a location vector for this twist (If the twist is a 
translation, input n in quotes):  ');
        w=input('Input an orientation vector for this twist:  ');
        p=input('Input the pitch for this twist (If the twist is a 
translation, type any integer):  ');
        M=input('Input the magnitude for this twist:  ');
        n=w/sqrt(dot(w,w));
        w=M*n;
        if (c=='n')
            T = [0 0 0 w];
        else
            T = ConstructPlucker(c,w,p);
        end
        A=Actuators;
        B=transpose(Actuators);
        Mag=zeros(6-indep,1);
        Mag=inv(B*A)*B*K*transpose(T);
        for i=1:(6-indep);
            disp(strcat('The force magnitude of actuator #',int2str(i),' is: 
'));
            disp(Mag(i));
        end
        da=zeros(6-indep,1);
        for m=1:(6-indep);
           Ta=(inv(K))*Actuators(1:6,m)*Mag(m);
           [ca,wa,pa]=DecomposePlucker(Ta);
           if (pa=='Infinite');
               da(m)=dot(wa,Forientation(:,m));
           else
               da(m)=dot((cross((ca-
transpose(Rlocation(:,m))),wa)+(pa*wa)),Forientation(:,m));
           end
           disp(strcat('The displacement magnitude of actuator 
#',int2str(m),' is:  '));
           disp(da(m));
        end
    end
end
```

REFERENCES:

[1] Evans C. Precision engineering: an evolutionary view. Bedfordshire, UK: Cranfield Press, 1989

[2] Smith ST. Flexures: Elements of elastic mechanisms. Newark, NJ: Gordon and Breach Science Publishers; 2000

[3] Hooke R. De potentia restitutiv. 1678

[4] Timoshenko SP. History of strength of materials. New York, NY: Dovers Publications, 1983

[5] Maxwell JC. General considerations concerning scientific apparatus. The Scientific Papers of James Clerk Maxwell. New York: Dover Press, 1890;2;505–21

[6] Clay RS, Roy J. Micro. Soc., 1937;57(1);1–7

[7] Jones RV. Instruments and experiences. New York, NY: Wiley; 1988

[8] Soemers H. Design principles for precision mechanisms. University of Twente, Netherlands 2010.

[9] Nomura T, Suzuki R. Six–axis controlled nano–meter–order positioning stage for micro–fabrication. Nanotechnology 1992;3(1);21–8

[10] Bamberger H, Shoham M. A new configuration of a six DOF parallel robot for MEMS fabrication. Proceedings of the 2004 IEEE International Conference on Robotics and Automation. New Orleans, LA. April 2004, 4545–50

[11] Chen SC, Culpepper ML. Design of a six–axis micro–scale nanopositioner—µHexFlex. Prec Eng 2006;30;314–24

[12] Taniguchi M, Ikeda M, Inagaki A, Funatsu R. Ultra–precision wafer positioning by six–axis micro–motion mechanism. International Journal of Japanese Society of Prec Eng 1992;26;35–40

[13] Dagalakis N, Amatucci E. Kinematic modeling of a 6 DOF tri–stage micro–positioner. Proceedings of the 16th Annual ASPE Meeting, November, 2001

[14] McInroy JE, Hamann JC. Design and control of flexure jointed hexapods. IEEE Transactions in Robotics and Automation 2000;16;372–81

[15] Zago L, Genequand P, Moerschell J. Extremely compact secondary mirror unit for the SOFIA telescope capable of six–degree–of–freedom alignment plus chopping. Proceedings of the SPIE International Symposium on Astronomical Telescopes and Instrumentation. March 1998

[16] Culpepper ML, Anderson G. Design of a low–cost nano–manipulator which utilizes a monolithic, spatial compliant mechanism. Prec Eng 2004;28;469–82

[17] Gao P, Swei S. A six DOF micro–manipulator based on piezoelectric translators. Nanotechnology 1999;10;447–52

[18] Du E, Cui H, Zhu Z. Review of nanomanipulators for nanomanufacturing. International Journal of Nanomanufacturing 2006;1(1);83–104

[19] Hopkins, J.B. and Culpepper, M.L., "Synthesis of Multi-degree of Freedom, Parallel Flexure System Concepts via Freedom and Constraint Topology (FACT)—Part I: Principles," Journal of Precision Engineering, 34 (2010): pp. 259-270

[20] Hopkins, J.B. and Culpepper, M.L., "Synthesis of multi-degree of Freedom, Parallel Flexure System Concepts via Freedom and Constraint Topology (FACT)—Part II: Practice," Journal of Precision Engineering, 34 (2010): pp. 271-278

[21] Hopkins JB. Design of parallel flexure systems via freedom and constraint topologies (FACT), Masters Thesis. Massachusetts Institute of Technology; 2007.

[22] Hale LC. Principles and techniques for designing precision machines. PhD Thesis. Massachusetts Institute of Technology; 1999

[23] Blanding DL. Exact constraint: machine design using kinematic principles. New York, NY: ASME Press; 1999

[24] Kim CJ, Moon Y, Kota S. A building block approach to the conceptual synthesis of compliant mechanisms utilizing compliance stiffness ellipsoids. Journal of Mechanical Design, 2008, 130.

[25] Howell LL., Compliant Mechanisms. John Wiley & Sons, Inc., New York, 2001

[26] Howell LL, Midha A. Parametric deflection approximations for end–loaded, large–deflection beams incompliant mechanisms. J Mech Design. 1995;117(1);156–65

[27] Howell LL. Compliant Mechanisms, New York, NY: Wiley; 2001

[28] Kennedy JA, Howell LL, Greenwood W. Compliant high–precision e–quintet ratcheting (CHEQR) mechanism for safety and arming devices. Prec Eng 2007;31(1);13–21

[29] Cannon BR, Lillian TD, Magleby SP, Howell LL, Linford MR. A compliant end–effector for microscribing. Prec Eng 2005;29(1);86–94

[30] Ananthasuresh GK, Kota S, Kikuchi N. Strategies for systematic synthesis of compliant MEMS. Proceedings of the 1994 ASME Winter Annual Meeting, November 1994

[31] Frecker MI, Ananthasuresh GK, Nishiwaki N, Kikuchi N, Kota S. Topological synthesis of compliant mechanisms using multi–criteria optimization. J Mech Design 1997;119;238–45

[32] Sigmund O. On the design of compliant mechanisms using topology optimization. Mech Struct Mach 1997;25(4);495–526

[33] Schultz G, Heimbold G. Dislocated actuator/sensor positioning and feedback design for flexible structures, Journal of Guidance Control Dynamics 5 361-7, 1983.

[34] Lim KB. Method for optimal actuator and sensor placement for large flexible structures, Journal of Guidance Control Dynamics 5 361-7, 1992.

[35] Hac A, Liu L. Sensor and actuator location in motion control of flexible structures, Journal of Sound Vibration 167 239-61, 1993.

[36] Han J, Lee I. Optimal placement of piezoelectric sensors and actuators for vibration control of a composite plate using genetic algorithms, Smart Mater. Struct. 8 257-267, 1999.

[37] Liu PX, Rao VS, Derriso M. Active control of smart structures with optimal actuator and sensor locations, Proc. of SPIE Conference on Smart Structures and Materials 2002: Modeling, Signal Processing, and Control, Vol. 4693, Paper No. 1, 2002.

[38] Zhang J, Zheng J, Gao R. Application of improved genetic algorithms for sensor and actuator placement of active flexible structures, Proc. of IMECE2006-14202, 2006 ASME International Mechanical Engineering Congress and Exposition. Pp 1475-1479.

[39] Naghshineh K, Heck LP, Olkin JA, Kamman JW. Evaluation of an actuator placement method for active noise control applications, Journal of Vibration and Acoustics vol 120, issue 4, pp 875-879, 1998

[40] Kumar, Ramesh K, Narayanan S. The optimal location of piezoelectric actuators and sensors for vibration control of plates, Smart MatMaterials and Structures, Vol 16 pp. 2680-2691(12), 2007.

[41] Slocum AH. Precision machine design. Society of Manufacturing Engineers Dearborn, Michigan, Prentice-Hall, Inc. 1992

[42] Huang Y, Cheng C. Robust tracking Control of a novel piezodriven monolithic flexure-hinge stage. Proceedings of the 2004 IEEE International Conference on Control Applications, Taipei, Taiwan, September, 2004.

[43] Gao P, Swei S, Yuan Z. A new piezodriven precision micropositioning stage utilizing flexure hinges. Nanotechnology 1999;10;394-398

[44] Golda, D.S. Design of high-speed, meso-scale nanopositioners driven by electromagnetic actuators. PhD Thesis. Massachusetts Institute of Technology, 2008.

[45] Ball RS. A treatise on the theory of screws. Cambridge, UK: The University Press; 1900

[46] Phillips J. Freedom in machinery: volume 1, introducing screw theory. New York, NY: Cambridge University Press; 1984

[47] Phillips J. Freedom in machinery: volume 2, screw theory exemplified. New York, NY: Cambridge University Press; 1990

[48] Bothema R, Roth B. Theoretical kinematics. Dover, 1990

[49] Hunt KH. Kinematic geometry of mechanisms. Oxford, UK: Clarendon Press; 1978

[50] Merlet JP. Singular configurations of parallel manipulators and grassmann geometry. The International Journal of Robotics Research, 8(5):45-56, 1989

[51] Hao F, McCarthy JM. Conditions for line-based singularities in spatial platform manipulators. Journal of Robotic Systems 1998;15(1);43–55

[52] McCarthy JM. Geometric design of linkages. Cambridge, MA: Springer Press; 2000

[53] Dimentberg FM. Screw Calculus and its Applications to Mechanics, Foreign Technology Division, WP-APB, Ohio, 1968.

[54] Waldron KJ. The constraint analysis of mechanisms, Journal of Mechanisms. 1966(1): 101-114.

[55] Dai JS, Huang Z, Lipkin H. Mobility of overconstrained parallel mechanisms. ASME Journal of Mechanical Design, 2006, 128(1): 220-229.

[56] Dai JS, Li D, Zhang Q, Jin GG. Mobility analysis of a complex structured ball based on mechanism decomposition and equivalent screw system analysis. Mechanism and Machine Theory, 2004, 39(4): 445-458

[57] Fang YF, Tsai LW. Structure synthesis of a class of 4-dof and 5-dof parallel manipulators with identical limb structures, The International Journal of Robotics Research, 2002, 21(9): 799-810

[58] Huang Z, Fang YF, and Tao WS. Studying on the kinematic characteristics of 3-dof in-parallel actuated platform mechanisms. Mechanism and Machine Theory, 1996, 31(8): 1009-1018

[59] Huang Z, Li QC. Type synthesis of symmetrical lower-mobility parallel mechanisms using the constraint-synthesis method, The International Journal of Robotics Research, 2003, 22(1): 59-79

[60] Kong XW, Gosselin C. Type Synthesis of Parallel Mechanisms, Springer-Verlag, 2007

[61] Ding X, Selig, JM. On the compliance of coiled springs, International Journal of Mechanical Science, 2004, 46(5), 703-727

[62] Yu JJ, Bi SS, and Zong GH, et al. A method to evaluate and calculate the mobility of a general compliant parallel manipulator, ASME International DETC2004, Volume 2: 28th Biennial Mechanisms and Robotics Conference, 2004, 743-748

[63] Gibson CG, Hunt KH. Geometry of screw systems-I, classification of screw systems. Mechanisms and Machine Theory, 1990, 25(1): 1-10

[64] Gibson CG, Hunt KH. Geometry of screw systems-II, classification of screw systems. Mechanisms and Machine Theory, 1990, 25(1): 11-27

[65] Rico JM, Duffy J. Classification of screw systems—I: one- and two-systems. Mechanism and Machine Theory, 1992, 27(4): 459-470

[66] Rico JM, Duffy J. Classification of screw systems—II: three-systems. Mechanism and Machine Theory, 1992, 27(4): 471-490

[67] Rico JM, Duffy J. Orthogonal spaces and screw systems, Mech. Mach. Theory, 1992, 27(4): 451-458.

[68] Dai JS, Rees JJ. Interrelationship between screw systems and corresponding reciprocal systems and applications. Mech. and Mach. Theory, 2001, 36(5): 633-651

[69] Pham H, Yeh H, Chen I. Micromanipulation system design based on selective actuation mechanisms. The International Journal of Robotics Research; Vol. 25, No. 2, February 2006, pp. 171-186.

[70] Patil CB. Robust design of selectively compliant flexure-based precision mechanisms. PhD Thesis. The University of Texas at Austin, May 2008.

[71] Kim CJ. Functional characterization of compliant building blocks utilizing eigentwists and eigenwrenches. Proceedings of the ASME 2008 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference. IDETC/CIE August 2008, Brooklyn, New York.

[72] Van Eijk J. On the Design of Plate-Spring Mechanisms. Ph.D. Thesis. Delft University of Technology. April 1985.