Design of Parallel Flexure Systems via Freedom and Constraint Topologies (FACT)

by

Jonathan Brigham Hopkins

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

MASTER OF SCIENCE IN MECHANICAL ENGINEERING
AT THE
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

JUNE 2007

© 2007 Massachusetts Institute of Technology
All rights reserved.

The author hereby grants to MIT permission to reproduce
and to distribute publicly paper and electronic
copies of this thesis document in whole or in part
in any medium now known or hereafter created.

Signature of Author: ……………………………………………………………………………
Department of Mechanical Engineering
May 1, 2007

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

Accepted by: ……………………………………………………………………………………
Lallit Anand
Professor of Mechanical Engineering
Chairman, Department Committee on Graduate Students

Design of Parallel Flexure Systems via Freedom and Constraint Topologies (FACT)

by

Jonathan Brigham Hopkins

Submitted to the Department of Mechanical Engineering on
May 1, 2007 in Partial Fulfillment of the
Requirements for the Degree of Master of Science in
Mechanical Engineering

ABSTRACT

The aim of this thesis was to generate the knowledge required to represent the possible freedom topologies (motions of a mechanism) and the possible constraint topologies (flexural elements that guide the mechanism) in a form that designers can use to design parallel flexure systems. The framework that links these topologies enables designers to create three-dimensional, multi-axis flexure systems by using "Freedom and Constraint Topologies" (FACT). FACT embodies every possible design solution for parallel flexure systems. This information enables designers to consider every possible design and then select the design that is best suited for a specific application. 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) twenty six unique matching pairs of freedom and constraint spaces were identified; (2) it was proven that these spaces embody all possible solutions; (3) a design process was created to guide a designer from design requirements, to freedom spaces, to constraint spaces, to mechanism designs; (4) a sub-process was created to guide designers in the selection of redundant constraints that help satisfy stiffness and symmetry requirements without altering the mechanism's kinematics; (5) mathematical expressions were created to represent the freedom and constraint spaces in a form that enables computers to identify and manipulate them. In this thesis, three case studies are provided to demonstrate the FACT design process for mechanisms of varying complexity: (1) a compliant spherical ball joint, (2) a compliant probe for a five axis STM, and (3) a compliant rotary flexure are designed. The second case study demonstrates the sub-process for selecting redundant constraints.

Thesis Supervisor: Martin L. Culpepper
Title: Rockwell International 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 Professor, Martin Culpepper, who tirelessly motivated, guided and directed my efforts. This work would not have been possible without his faith in me. I am grateful to Professor Judy Vance, Denis Dorozhkin, and Haijun Su with whom I collaborated during the creation of this theory. To my dear friends in the PCS lab who offered their friendship and support, I am also in debt. And finally, I am grateful to NSF 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 an overview of the purpose, importance, and impact of this research. This chapter also provides an overview of the current research and background on related work that enables the reader to better understand the contributions of the present research in the context of past work.

1.1 Research Objectives

The purpose of this thesis was to generate the knowledge required to represent the possible freedom topologies (motions of a mechanism) and the possible constraint topologies (flexural elements that guide the mechanism) in a form that designers may use to design parallel flexure systems. The framework that links these topologies enables designers to create three-dimensional, multi-axis flexure systems by using “Freedom and Constraint Topologies” (FACT). FACT embodies every possible design solution for parallel flexure systems. This information enables designers to consider every possible design and then select the design that is best suited for a specific application. FACT was created to improve the design processes for small-scale flexure systems and precision machines.

The FACT design process utilizes the principles of constraint-based design [1], and the mathematics of screw theory and projective geometry, to enable novice and expert designers to create multi-axis parallel flexure systems. Prior to the creation of FACT, there was no guarantee that an expert designer, with years of experience, could generate all possible designs that would satisfy a given motion requirement. FACT embodies all possible design solutions and, therefore, any designer, novice or expert, can be confident that all design concepts have been considered. The FACT design process is based upon the information that is contained in FACT. The process guides designers through the decisions that must be made to create flexure system concepts. During the process, designers require information to make these decisions. This information is drawn from FACT. The combination of the FACT process and information provides the designer with everything that is needed to (a) design any multi-axis flexure system and (b) select redundant constraints that satisfy stiffness and symmetry requirements.

The preceding points are important because flexure systems have a large impact on everyday life [2]. Flexure systems (1) possess nanometer repeatability; (2) they are “friction free” for practical purposes and thus generate negligible internal heat or wear; (3) they require fewer components, and cost less to make, than conventional rigid mechanisms. The inherent precision of flexure systems makes them suitable for use in equipment and instruments that are used to create precision components for consumer products. Examples include disk drives, flat panel TVs, and fiber optic devices. These products require a multi-axis precision machine at some point in their development for inspection and/or fabrication purposes. The preceding examples are within areas of high economic or scientific impact. Any improvements in flexure system design could have the potential to increase the quality and decrease the cost of these, and other products. With respect to FACT, the ability to compare and select designs from a complete set of possible concepts should lead to better designs.

Flexure systems are also important to metrology for micro- and nano-fabrication, scanning-probe microscopy, lens fixtures, and other fields that require precision machinery. There is a growing need to create nanopositioners for emerging, three-dimensional nano-scale research and manufacturing applications [3-13]. For instance, compliant nanopositioners could also be used to control the parallelism between two plates separated by a distance of only a few nanometers. The ability to achieve accurate parallelism between two plates would enable scientists to perform experiments that involve the flow of fluids through tiny channels [14] as well as perform radiation tests through nano-scale gaps [15].

Designers can use FACT to create flexure systems that consist of an arbitrarily shaped rigid stage that is attached to some number of flexural elements that emulate ideal constraints. Ideal constraints restrict motion in one direction only. These flexural elements are grounded at one end and attached to the rigid stage at the other end as shown in **Figure 1.1**. FACT enables designers to determine any flexure system's optimal constraint topology for any given set of desired motions such as the rotational and translational degrees of freedom shown in the figure.

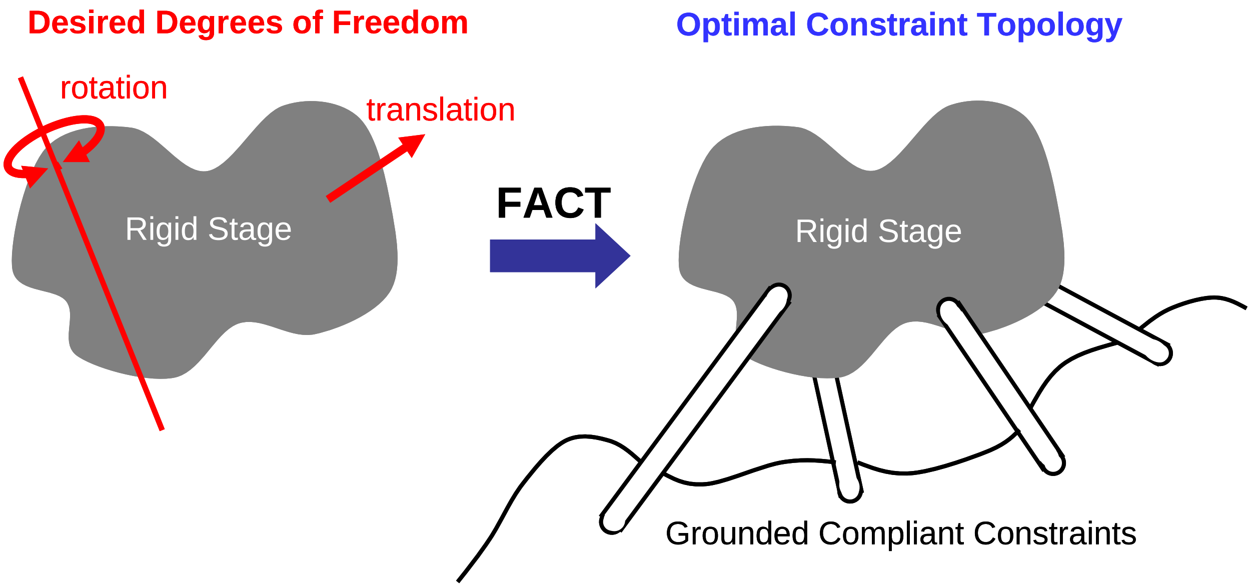

Desired Degrees of Freedom
rotation
translation
Rigid Stage
FACT
Optimal Constraint Topology
Rigid Stage
Grounded Compliant Constraints

Figure 1.1: FACT allows designers to determine a flexure system's optimal constraint topology such that it may move with desired degrees of freedom (shown in red).

Figure 1.2 shows three examples of compliant positioning stages that were designed using FACT.

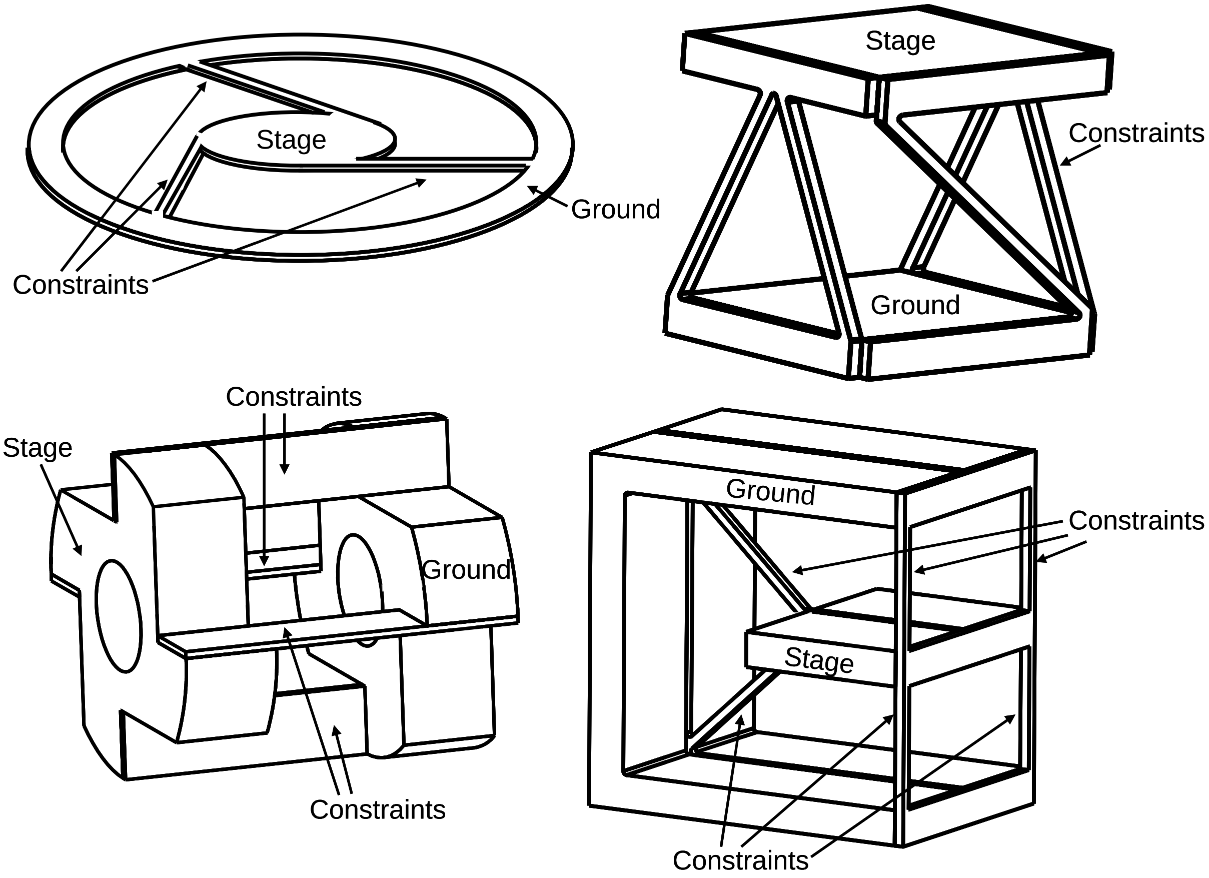


Figure 1.2: Flexure systems designed using FACT.

1.2 Elastic Mechanism Design Tools

FACT is the newest of four methods used to design elastic mechanisms. The other methods include: (1) the pseudo-rigid-body model [16], (2) topological synthesis [17], and (3) constraint-based design [18]. The final sub-section of this section compares and contrasts these methods with FACT.

1.2.1 The Pseudo-Rigid-Body Model

The psuedo-rigid-body model (PRBM) [16] is a method that is used to predict the large-motion kinematic and elastomechanic behavior of compliant mechanisms. The PRBM creates an equivalent rigid-linkage that emulates the behavior of the compliant mechanism under study.

Torsional springs are assigned to each joint within the linkage. Corresponding torsional stiffnesses are assigned to each spring using equations that rely on geometric parameters and predetermined constants. **Figure 1.3** shows a compliant four bar mechanism and its PRB analog. The main advantage of the PRBM is that well-known analysis methods for rigid mechanisms may be used to predict the behavior of compliant mechanisms.

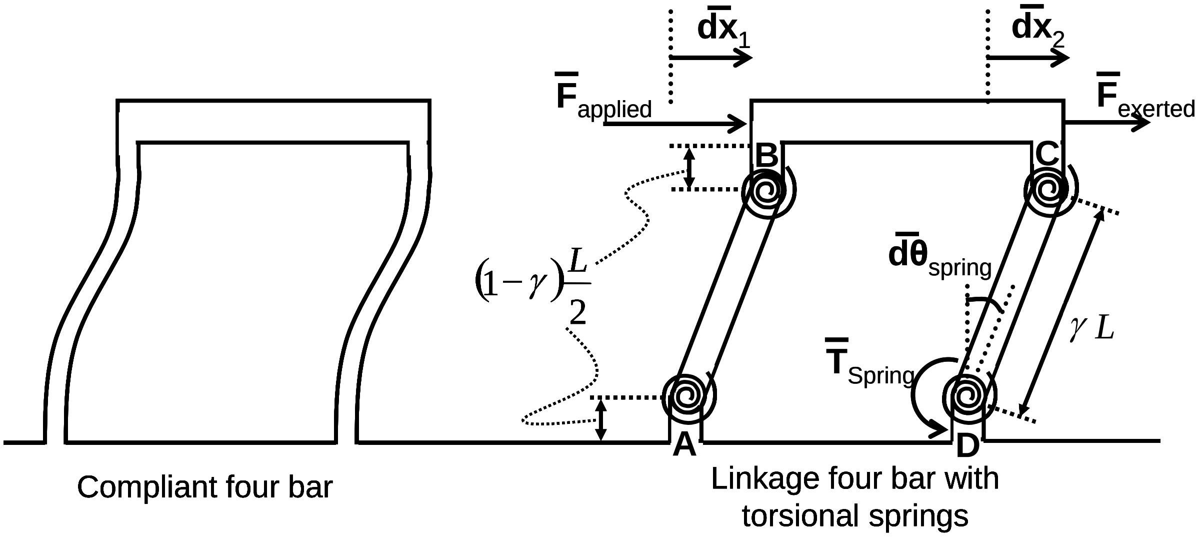


Figure 1.3: A compliant four bar that is modeled as a rigid linkage using the PRBM.[1]

The vector loop of the four bar shown in **Figure 1.3** is given as

$$\vec{R}_{AB} + \vec{R}_{BC} + \vec{R}_{CD} + \vec{R}_{DA} \quad = \quad 0\hat{i} \ + \ 0\hat{j} \ + \ 0\hat{k}\,. \tag{1.1}$$

The torsional spring constant, K_{spring}, of each spring shown in **Figure 1.3** is given as

$$K_{spring} = 2 \cdot \frac{\gamma}{L} \cdot K_{\theta} \cdot EI \,, \tag{1.2}$$

where E is the modulus, I is the moment of intertia, L is the length of the beam, γ is defined in **Figure 1.3**, and K_{θ} is a stiffness coefficient. **Equation (1.1)** and **Equation (1.2)** are used with

[1] Vectors are expressed as letters with bars above them in the figures of this thesis.

the principle of virtual work to link the applied loading of the compliant mechanism and its displacements as

$$\Sigma\int \vec{F}_{applied} \cdot d\vec{x}_1 = \Sigma\int \vec{T}_{spring} \cdot d\vec{\theta}_{spring} + \Sigma\int \vec{F}_{exerted} \cdot d\vec{x}_2 \ . \qquad (1.3)$$

The PRBM has been used to model and analyze a wide variety of commercial compliant mechanisms such as centrifugal clutches, bicycle derailleurs, thermal actuators, bistable mechanisms, and grippers [19,20].

1.2.2 Topological Synthesis

Topological synthesis [21-23] constructs the topology of compliant mechanisms by satisfying input and output displacement/force specifications using systems of linear beam elements. A rectangular design domain is divided into a number of nodes and connecting beam elements are modeled as an initial concept as shown in **Figure 1.4**. Through a sequence of iterations, a computer generates multiple design concepts by (a) eliminating beam elements from the design domain and then (b) testing each concept using FEA to identify an optimal design that will best meet the design requirements. Subsequent processes vary the thicknesses, lengths, and material properties of the beams within the selected concept to optimize the compliant mechanism's performance. The mechanism is designed to satisfy its elastic requirements without buckling under external loads. The optimization process considers criteria such as mechanical/geometrical advantage, volume or weight of the material used, work done by external forces, stress and strain levels, fatigue strength, ease of manufacturing, and ergonomics and aesthetics.

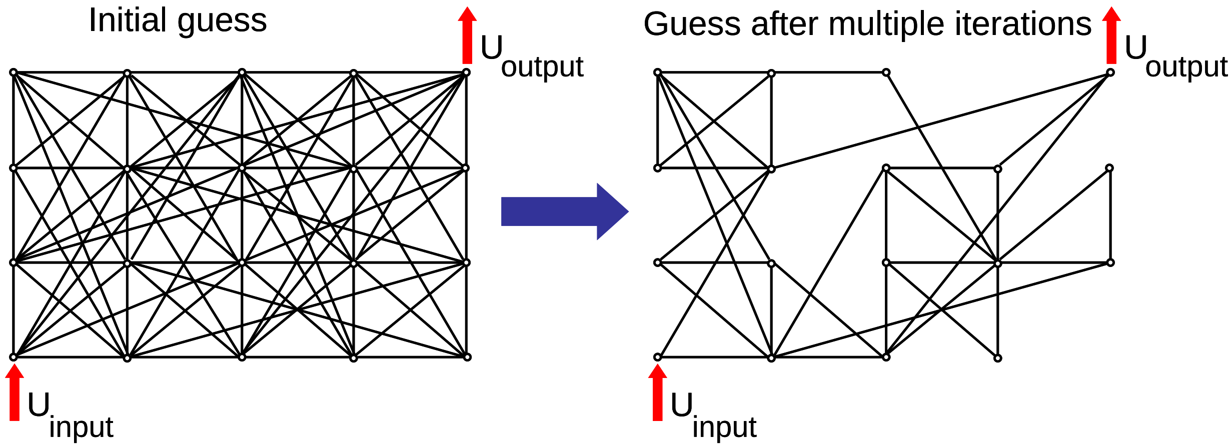


Figure 1.4: Topological synthesis eliminates beam elements from a design domain until a concept is generated that most optimally satisfies the design requirements (displacements shown in red).

Topological synthesis utilizes principles of energy to converge to an optimal design. Energy losses and inertial force effects are assumed to be negligible in the compliant mechanism such that the input energy, $F_{input} \bullet U_{input}$, is equal to the output energy, $F_{output} \bullet U_{output}$, plus the stored strain energy in the mechanism. The mutual potential, *MPE*, is maximized to attain the flexibility requirements and is given by

$$MPE = U_{input}^{T} K U_{output}, \qquad (1.4)$$

where *K* is the system's stiffness matrix.

Topological synthesis has been used to design compliant grippers, mechanical frequency doublers, as well as micro- and nano-devices.

1.2.3 Constraint-based Design

Constraint-based design [18] is based upon the axiom that the orientation and location of a mechanism's constraints determine the motion of the mechanism. Constraint-based design requires a human's abilities to recognize patterns, visualize motions, and synthesize compliant modules. The principles of constraint-based design are not easily programmable and no computer-aided tool exists for its application.

An experienced constraint-based designer is familiar with modules that move with easily visualized motions such as the mechanisms shown in **Figure 1.5**. The motions of the first module on the left side of this figure are easily visualized using instant centers. An instant center is a point that exists at the junction of lines of action from compliant constraints that are attached to the rigid body. The concept of an instant center is a key principle in constraint-based design. The motions of the second module shown on the right side of the figure are also easily visualized. This mechanism is well known to constraint-based designers as a compliant four bar, or parallel guiding mechanism.

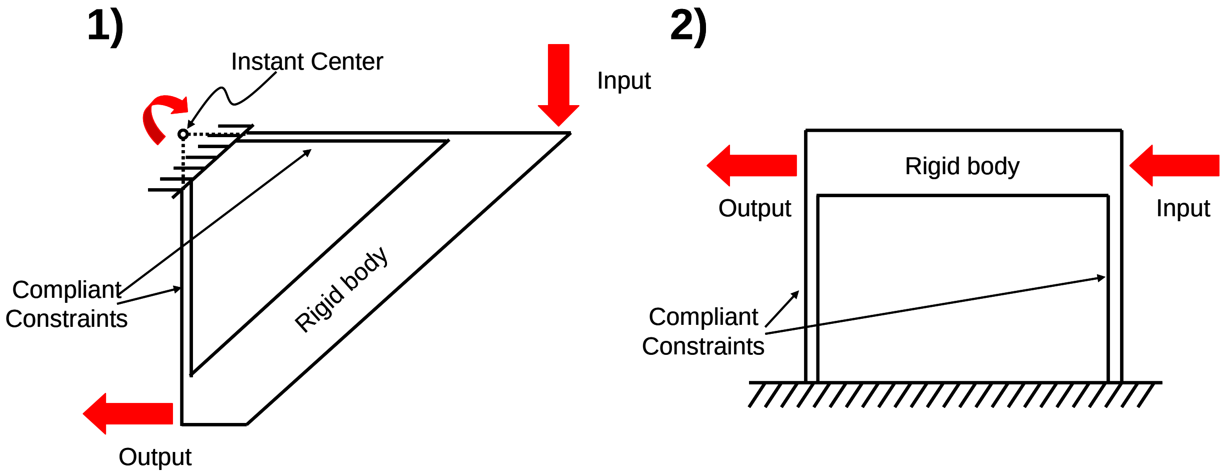


Figure 1.5: Two compliant modules with motions (red) that are easily visualized.

These modules may be combined to form more complicated compliant mechanisms that are capable of exhibiting complex motions such as those that could be obtained by the mechanism shown in **Figure 1.6** [24].

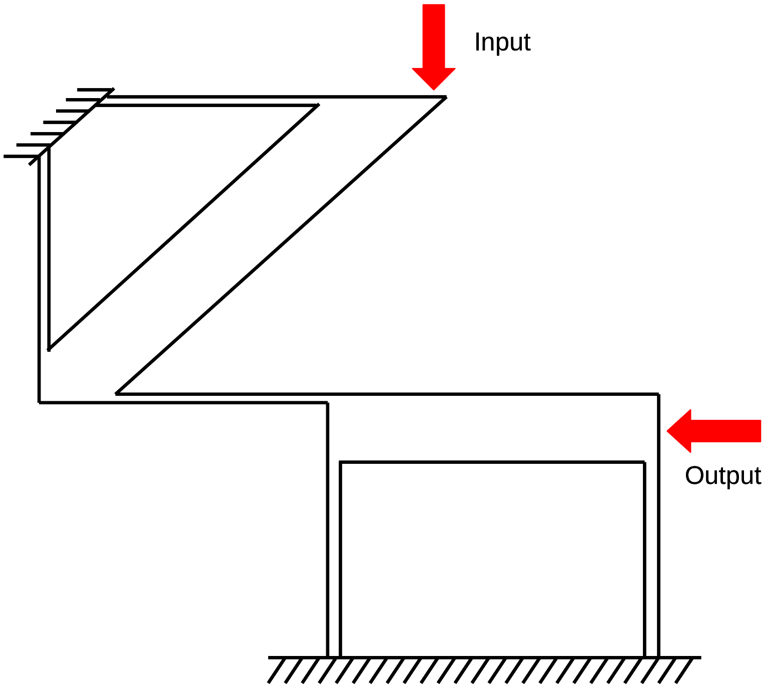


Figure 1.6: A compliant mechanism with motions (red) that are more difficult to visualize.

The relative lengths of the constraints and the placement of the instant centers, rigid bodies, and grounds are all determined by the designer to achieve the desired mechanism motions and transmission ratios.

Constraint-based design also includes rules involving modules that are in series or in parallel. The motions of a mechanism that consists of modules combined in series are determined by adding the motions of each individual module. The effective constraint of a mechanism that consists of modules combined in parallel is determined by adding the constraints of each individual module. Determining the kinematics and elastomechanics of these combined modules is a difficult task even for an experienced constraint-based designer.

1.2.4 Comparison of FACT with Conventional Design Tools

This section compares the FACT design method with the methods in the previous sub-sections.

Comparing the pseudo-rigid-body model with the FACT design method is difficult because their objectives are different. The pseudo-rigid-body model, although a powerful modeling tool, is not a synthesis tool. The FACT design method, on the other hand, is intended to generate flexure system concepts.

Topological synthesis is a concept synthesis tool. Topological synthesis requires little or no input from the designer during its design process. On occasion, therefore, it generates designs that may be difficult to manufacture and integrate into machines. Mechanisms created using topological synthesis are generally planar and are not capable of multi-axis motions. FACT, however, is capable of designing three-dimensional, multi-axis mechanisms that are capable of moving with complex motions such as screws that cause the mechanism's stage to translate along any axis through three-space while simultaneously rotating the stage with a desired pitch. FACT also enables designers to consider all possible solutions.

Constraint-based design is a knowledge-based design process that requires years of apprenticeship to master. The FACT design process, however, requires less design experience. A designer must only be capable of selecting lines from within spaces that are provided. Although the designer is given the liberty of making important design decisions, he/she is instructed during the process and is guaranteed a functioning design. Although the FACT design method is based on the principles of constraint-based design, FACT is much more quantitative in nature than constraint-based design and is, therefore, more systematic and general.

FACT enables the use of both the designer's intuition and the guidance of a systematic method to create three-dimensional elastic mechanisms that are capable of moving with complex motions. It utilizes the semi-qualitative principles of constraint-based design and the quantitative principles of screw theory to visually represent all possible constraint solutions for every possible parallel elastic mechanism using fully parameterized spaces. These spaces provide designers with an immediate visual understanding of the kinematics of complicated elastic mechanisms. FACT also guides the designer in selecting possible non-redundant constraints from within these constraint spaces to ensure correct kinematics. Furthermore, FACT guides the designer in intelligently selecting redundant constraints to control the stiffness, stability, symmetry and load capacity of mechanisms. The ability to control these parameters is a novel advance in the design of elastic mechanisms. Prior to FACT, no formal method existed that was capable of controlling useful system redundancy.

1.3 Thesis Overview

Chapter 2, **Chapter 3**, and **Chapter 4** of this thesis review principles of constraint-based design, screw theory, and projective geometry respectively. These principles are reviewed in the context of FACT. **Chapter 5** introduces the concepts of freedom and constraint space as spaces that fully describe the motions and constraints of any system. **Chapter 6** describes three geometric entities that appear as freedom and constraint spaces within FACT. **Chapter 7** and **Chapter 8** identify and describe every possible freedom and constraint space for every system that exists in three-dimensions. **Chapter 9** discusses the findings from the two previous chapters and notes a symmetry within the spaces. **Chapter 10** introduces the FACT design process and provides three case studies for demonstrating its utility. **Chapter 11** summaries the accomplishments of this research and concludes by listing ideas for potential future work.

CHAPTER 2:

“Constraint-based Design”

This chapter reviews key principles of constraint-based design. FACT is based upon the principles of constraint-based design. The last section of this chapter presents new insights into constraint-based design that inspired the creation of FACT.

2.1 Maxwell’s Contributions

James Clerk Maxwell’s observations in the field of “Exact Constraint” [25] were important to the development of constraint-based design. He formulated a basic mathematical relationship between constraints and degrees of freedom[2]. A constraint restricts motion in a particular direction. Every non-redundant constraint that is added to a body removes a single degree of freedom from that body. The equation that expresses this observation is written as

$$6 - N = R \tag{2.1}$$

where N is the number of non-redundant constraints and R is the number of independent degrees of freedom. Free standing objects in three-space have 6 degrees of freedom—three orthogonal translations and three orthogonal rotations. These degrees of freedom are shown in **Figure 2.1**.

[2] Although Maxwell is largely responsible for popularizing the concept of **Equation (2.1)**, it’s possible that he wasn’t its originator [26]

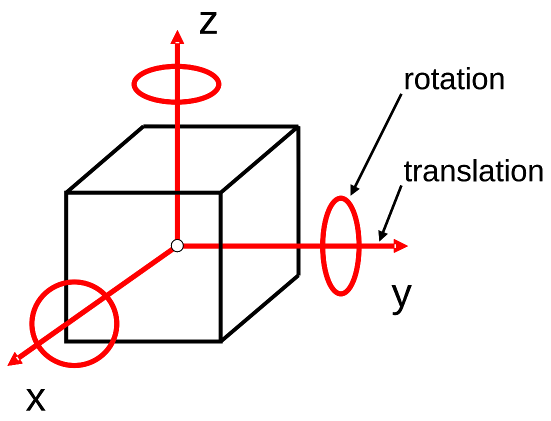


Figure 2.1: Every free standing object has 6 independent degrees of freedom—three orthogonal translations and three orthogonal rotations

A body is exactly constrained if it has 6 non-redundant constraints. Such a body will be unable to move since it has no degrees of freedom.

Figure 2.2 illustrates the traditional way of interpreting **Equation (2.1)**. In this figure a rigid block is constrained by two parallel compliant beams. According to **Equation (2.1)**, the block loses two of its 6 degrees of freedom. The four remaining degrees of freedom may be visualized and confirmed by logic. They include two translations along the y- and z-axes and two rotations along the x- and y-axes as shown in **Figure 2.2**.

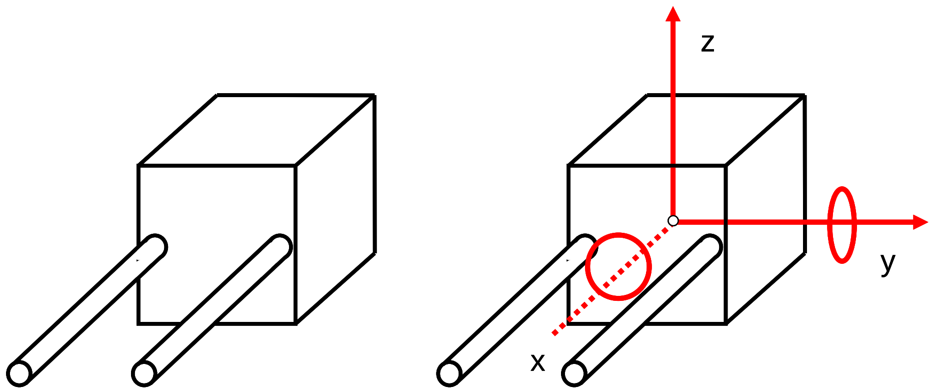


Figure 2.2: A block with two constraints and four degrees of freedom.

This traditional interpretation of **Equation (2.1)**, however, does not provide a comprehensive understanding of the system's kinematics. A more complete way of representing a system's kinematics will be shown later.

2.2 Blanding's Contributions

Douglass L. Blanding's research is also important in determining the relationship between constraints and degrees of freedom [27]. He modeled slender, compliant beams as ideal constraints. An ideal constraint is approximated as having infinite compliance perpendicular to the constraint's line of action and infinite stiffness along the constraint's line of action as shown in **Figure 2.3a**. This model, although simplistic, is adequately descriptive for finding the directions of greatest compliance for a rigid stage constrained by slender, compliant beams for small motions. ***In this thesis a constraint line is represented by a blue line*** that travels through the center of a physical constraint along its line of action.

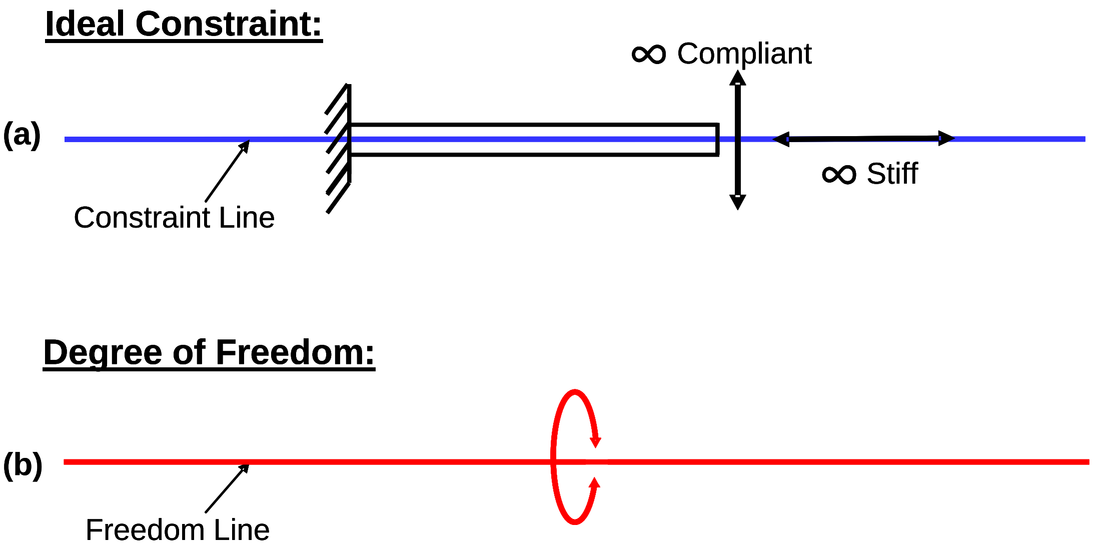


Figure 2.3: Modeling constraints and degrees of freedom as lines in three-space.

Blanding also observed that an object's degrees of freedom could be represented by rotations about lines called freedom lines shown in **Figure 2.3b**. ***In this thesis all freedom lines will be shown in red***. He noted that pure translational degrees of freedom could be modeled as freedom lines that are perpendicular to the direction of translation that is located infinitely far from the object that is translating. This concept is illustrated in **Figure 2.4**.

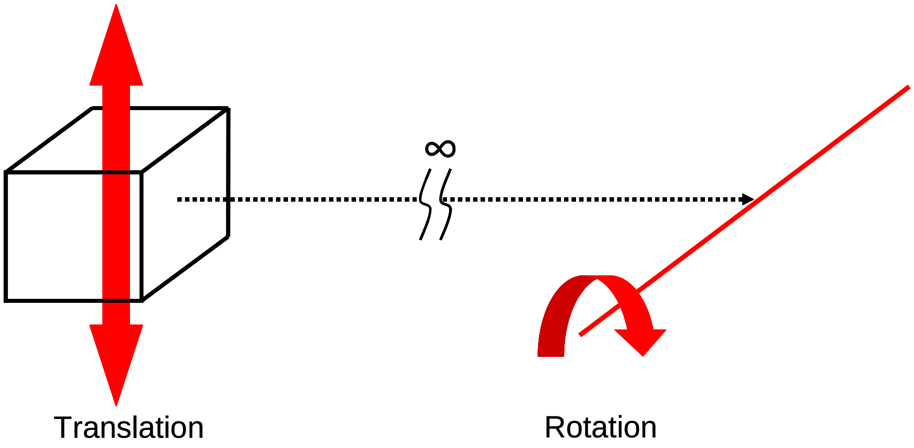


Figure 2.4: A pure rotational freedom line infinitely far from an object will emulate a pure translational degree of freedom

Blanding's Rule of Complementary Patterns defines the relationship between constraints and degrees of freedom. The spaces used in the FACT design process were largely determined using this rule. The rule of Complementary Patterns states the following:

Every freedom line intersects every constraint line.

Whether the points of intersection are the same or not does not matter as long as each freedom line intersects each constraint line.

Blanding also asserted that parallel lines intersect each other at a single point at infinity. **Figure 2.5** demonstrates this principle of projective geometry. Imagine first, two lines intersecting at a point in finite space. This point will gradually move toward infinity as the lines approach a parallel state.

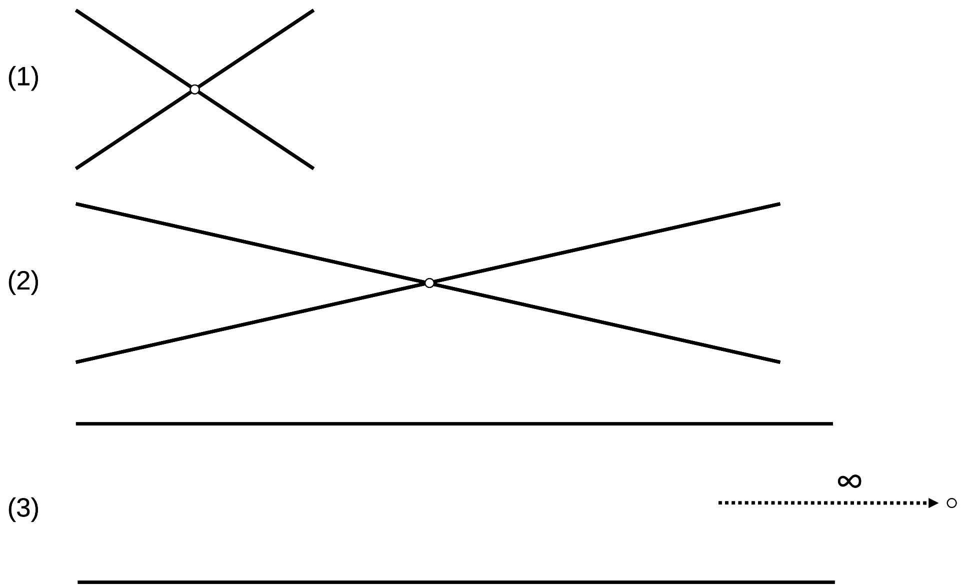


Figure 2.5: As intersecting lines become more parallel, the point of intersection approaches infinity.

Figure 2.6 shows an example of a block constrained by five non-redundant constraints. According to **Equation (2.1)**, this block should have one remaining degree of freedom. Blanding's Rule of Complementary Patterns finds this pure rotational freedom line to be the red line shown in **Figure 2.6**. This line is the only line that intersects every blue constraint line at least once. This red line intersects two of the blue lines at a single point on the edge of the block, and intersects the other three blue lines at a point infinitely far from the block since it is parallel to them.

As the relationship between constraints and degrees of freedom is independent of the stage's shape, size, and location, the block is not important to a basic understanding of the kinematics of the system and may be removed from the picture. Only constraint lines (blue) and freedom lines (red) remain as shown in **Figure 2.6**.

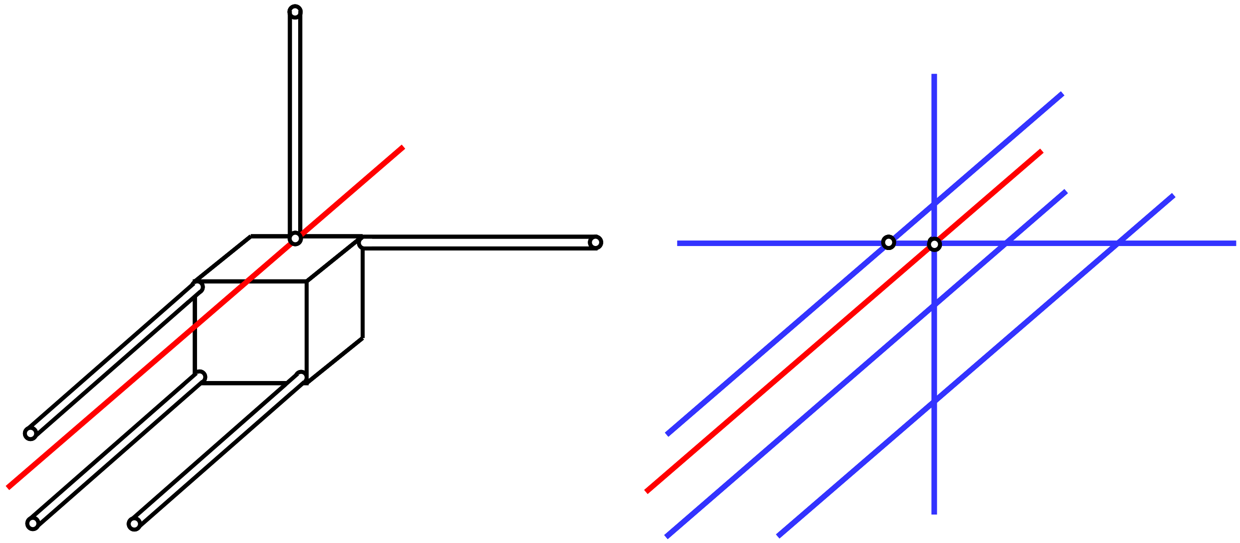


Figure 2.6: Constraint and freedom lines for a block constrained with five non-redundant constraints

Blanding also made two important assertions about freedom lines:

(1) If two intersecting freedom lines exist for a given constraint layout, an entire disk (i.e. pencil) of infinite freedom lines will also exist. This disk will lie in the same plane as the two intersecting freedom lines and will contain, as its center point, the intersection point of these two lines.

(2) If two parallel freedom lines exist for a given constraint layout, a plane containing infinite parallel freedom lines will also exist. The two parallel freedom lines will lie on the same plane as the plane containing the infinite parallel freedom lines and will be parallel to those lines.

Figure 2.7 depicts these assertions. These assertions will be mathematically verified later in this thesis along with Blanding's Rule of Complementary Patterns. These concepts help the designer find the complete set of motions for a given system.

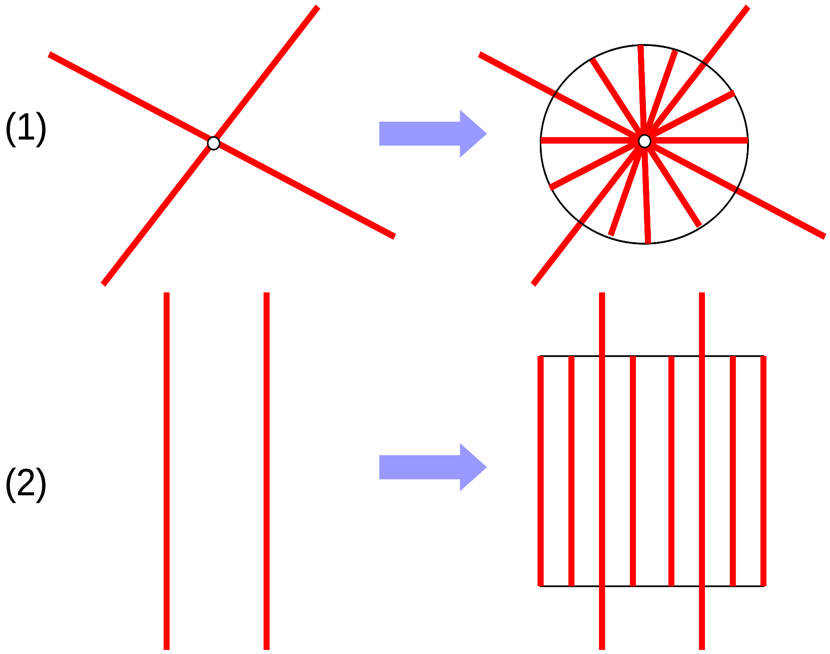


Figure 2.7: (1) If two intersecting freedom lines exist, a disk of infinite freedom lines will also exist. (2) If two parallel freedom lines exist, a plane of infinite parallel freedom lines will also exist.

2.3 New Insights in Constraint-based Design

In many systems an infinite number of freedom lines satisfy the Rule of Complementary Patterns. This section discusses such systems and introduces the notion that spaces exist that contain every freedom line for a given system. This notion enables one to visually represent the complete kinematics for any mechanism and is integral to the FACT design method.

Consider the rigid block in **Figure 2.8** that is constrained by three non-redundant constraints. The block is removed from the picture since only constraint lines are necessary to find freedom lines. Two of the constraint lines lie on the horizontal plane depicted in **Figure 2.8**. They intersect at a point that lies on the dashed intersection line between the horizontal and vertical plane. The third constraint line lies on the vertical plane.

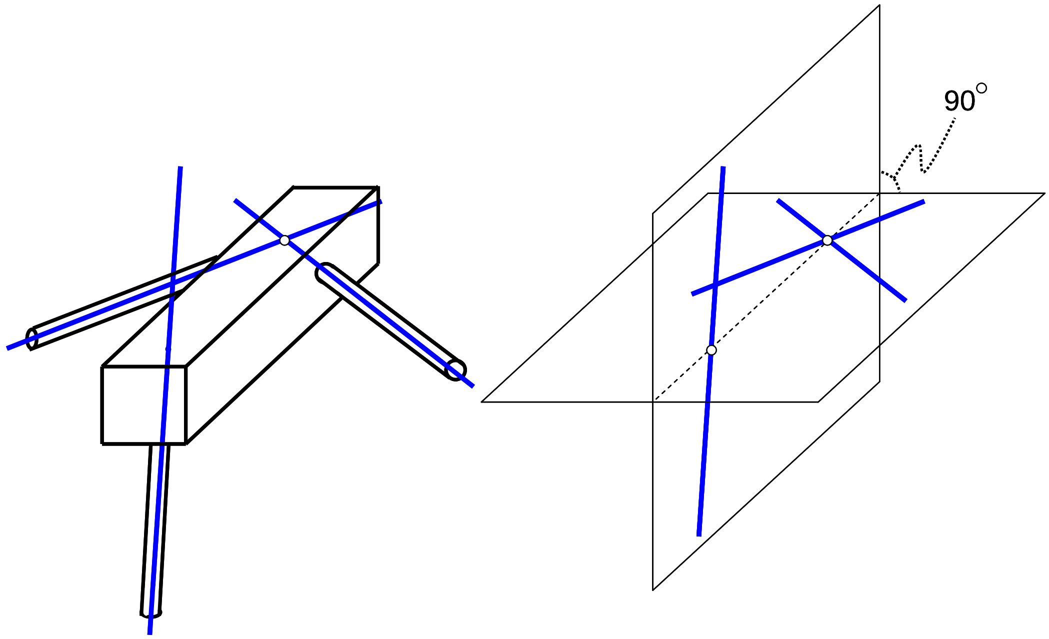


Figure 2.8: A flexure system with three non-redundant constraints shown with, and without, its rigid stage.

The Rule of Complementary Patterns will now be used to find all of the freedom lines. The first picture in **Figure 2.9** shows a disk containing an infinite number of red lines that lie on the horizontal plane. The center point of this disk is the intersection of the vertical constraint line and the horizontal plane. Every line inside this disk intersects the constraint line on the vertical plane, but they all also intersect the other two constraint lines that share the same plane. Since every red line inside this disk intersects all three blue constraint lines, Blanding's Rule of Complementary Patterns suggests that they are all freedom lines of the system. The second picture in **Figure 2.9** shows another disk of red lines that lie on the vertical plane with a center point coincident with the intersection point of the two constraint lines on the horizontal plane. All of these red lines also intersect all three constraint lines and are, therefore, also freedom lines of the system. No other lines outside of these two red disks will intersect all three of the constraint lines.

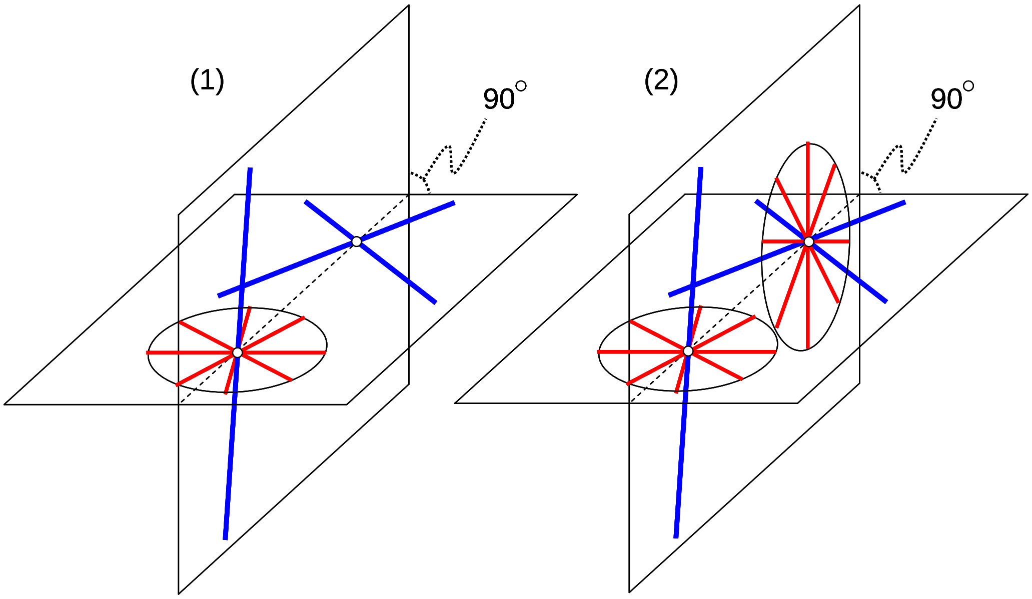


Figure 2.9: Every line that intersects all three constraint lines may be expressed as two disks of freedom lines. One disk lies on the horizontal plane and the other disk lies on the vertical plane.

The block is shown again in **Figure 2.10** with the two disks of pure rotational freedom lines. These disks are visual representations of the pure rotational kinematics of the constrained block. In other words, the block's permissible motions will be rotations about any of the red lines shown in **Figure 2.10**.

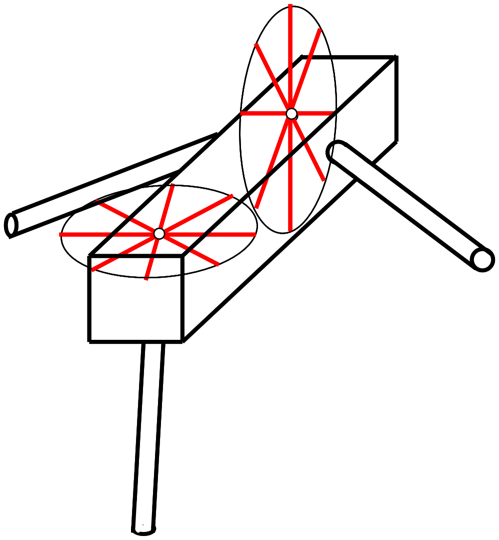


Figure 2.10: Kinematics of the block constrained by three non-redundant constraints that are expressed as pure rotational freedom lines (red).

It may seem contradictory to consider an infinite number of freedom lines when only three degrees of freedom are expected. Later it will be shown that the two disks of infinite freedom lines contain three independent freedom lines, a finding that is consistent with **Equation (2.1)**.

The concept of visually representing an infinite number of allowable motions of a flexure system using finite geometric shapes and spaces that contain an infinite number of freedom lines is a key concept to the FACT design method. This concept will be developed further in later chapters.

CHAPTER 3:

“Screw Theory”

This chapter reviews the basic principles of screw theory [28-30] to generate a mathematical relationship between degrees of freedom and constraints. This relationship lies at the heart of the FACT design method.

3.1 Twists as Degrees of Freedom

This section presents a model for mathematically describing the degrees of freedom of flexure systems. In the context of kinematics, Chasles Theorem [31] states that:

“Any motion of a rigid body in space may be described as a screw motion.”

All degrees of freedom will, therefore, be modeled as screws or twists in space. A twist is a 6×1 velocity vector that is represented as a single line in three-space. A twist may be described using three parameters: (1) a 3×1 location vector, $\vec{c}$, that points from the origin of an arbitrarily defined coordinate system to any point along the twist’s line, (2) a 3×1 orientation vector, $\vec{w}$, that points in the direction of the twist’s line and represents the twist’s rotational velocity, and (3) a scalar pitch value, *p*, where pitch is defined as the twist’s translation per rotation along its line. **Figure 3.1** depicts a standard twist using these parameters.

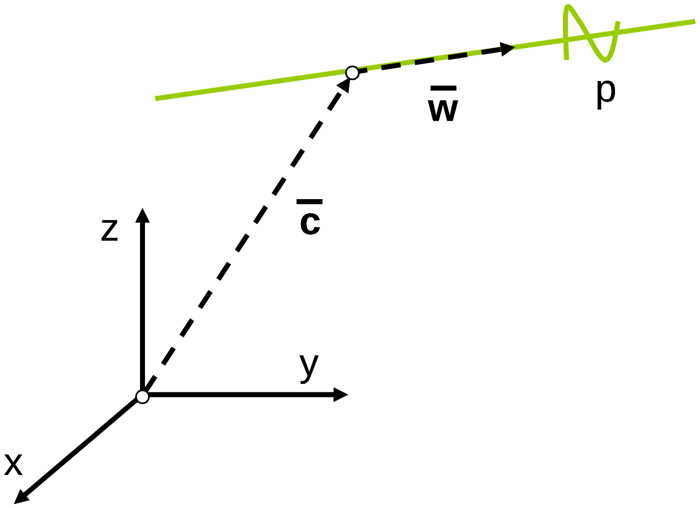


Figure 3.3: A twist (green line) with a location vector ($\vec{c}$), an orientation vector ($\vec{w}$), and a pitch of *p*

The 6×1 twist vector, $\vec{T}$, is given in **Equation (3.1)** in terms of the parameters defined previously. This twist vector is defined as

$$\vec{T} = \begin{bmatrix} \vec{w} \\ (\vec{c}\times\vec{w}) + p\vec{w} \end{bmatrix} = \begin{bmatrix} \vec{w} \\ \vec{v} \end{bmatrix} = \begin{bmatrix} w_x \\ w_y \\ w_z \\ v_x \\ v_y \\ v_z \end{bmatrix}, \qquad (3.1)$$

where $\vec{v}$ is a 3×1 vector that represents the twist's translational velocity and $\vec{w}$ is a 3×1 vector that represents the twist's rotational velocity. If the twist's pitch equals zero, the twist will be a rotational freedom line. If the pitch is infinite, the twist will be a pure translation along the twist's line of action. If the pitch is a non-zero finite value, it represents a motion that translates along the line of the twist while simultaneously rotating about the same line in a coupled fashion. For the remainder of this paper, twists with zero pitch values will be called pure rotations or freedom lines and they will be depicted as red lines. Twists with infinite pitch values will be called pure translations and they will be depicted as thick black lines. Twists with non-zero finite pitch values will be called screws and they will be depicted as green lines. This convention is shown in **Table 3.1**.

Table 3.1: Twist names and line colors for different categories of pitch

Pitch Value	Name of Twist	Color of Twist Line
$p = 0$	Pure Rotation or Freedom Line	🟥
$p = \infty$	Pure Translation	⬛
$p \neq 0 \neq \infty$	Screw	🟩

Suppose one wished to find the twist vector, $\vec{T}$, for a given screw with a pitch value of 2m/rad and a rotational velocity vector, $\vec{w}$, with a magnitude of $\sqrt{2}$ rad/s as shown in **Figure 3.2**. The screw's line of action (green) never intersects the x-y plane but intersects the z-axis at a distance of 1m above the origin. The projected line of the screw onto the x-y plane is 45 degrees from the x-axis as shown.

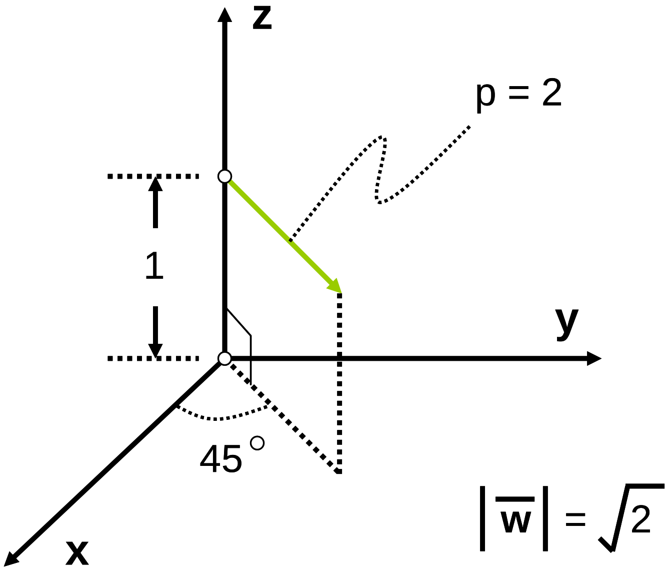


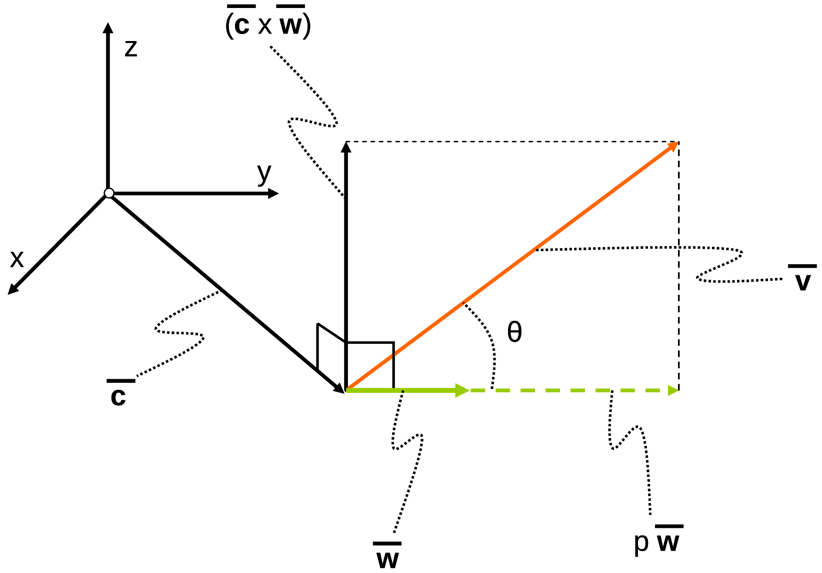
Figure 3.2: A twist with a pitch of 2m/rad and a rotational velocity vector, $\vec{w}$, with a magnitude of $\sqrt{2}$ rad/s

In order to find the twist vector, $\vec{T}$, a location vector, $\vec{c}$, must be identified. The most convenient location vector to choose along the twist line is $\vec{c} = \begin{bmatrix}0 & 0 & 1\end{bmatrix}^T$. A rotational velocity vector of $\vec{w} = \begin{bmatrix}1 & 1 & 0\end{bmatrix}^T$ will point in the correct direction and will have a magnitude equal to $\sqrt{2}$ rad/s. The pitch value, *p*, has been given a value of 2m/rad. At this point, the three important parameters that define a twist: $\vec{c}$, $\vec{w}$, and *p* have been found. If these parameters are plugged into **Equation (3.1)** the twist vector is found to be $\vec{T} = \begin{bmatrix}1 & 1 & 0 & 1 & 3 & 0\end{bmatrix}^T$.

3.1.1 Decomposing Twists

This section explains how twists may be decomposed. The ability to decompose a twist is important for visualizing where its line of action lies in three-space.

Suppose one is given a twist vector, $\vec{T}$, and wishes to decompose it to find its $\vec{c}$, $\vec{w}$, and *p* parameters. The rotational velocity vector, $\vec{w}$, consists of the first three components of the twist vector. The translational velocity vector, $\vec{v}$, consists of the last three components of the twist vector. Finding the pitch value using these two vectors requires a closer look at the definition of a twist. **Figure 3.3** is a picture of a twist line (green) showing both the vectors $\vec{w}$ and $\vec{v}$.



$(\overline{\mathbf{c}} \times \overline{\mathbf{w}})$
z
y
x
$\overline{\mathbf{v}}$
θ
$\overline{\mathbf{c}}$
$\overline{\mathbf{w}}$
p $\overline{\mathbf{w}}$

Figure 3.3: A graphical representation of Equation (3.1).

Note from **Figure 3.3** that

$$|\vec{v}|\cos\theta = p|\vec{w}|, \tag{3.2}$$

If the dot product of the two velocity vectors, $\vec{w}$ and $\vec{v}$, are taken, it can be shown that

$$\vec{w} \bullet \vec{v} = |\vec{w}||\vec{v}|\cos\theta, \tag{3.3}$$

where θ is the angle between the two velocity vectors as shown in **Figure 3.3**. If **Equation (3.2)** is plugged into **Equation (3.3)** and p is solved for, it can be shown that

$$p = \left(\frac{\vec{w} \bullet \vec{v}}{\vec{w} \bullet \vec{w}}\right). \tag{3.4}$$

Now that the rotational velocity vector, $\vec{w}$, and the pitch value, p, may be found given a twist vector, finding an acceptable location vector, $\vec{c}$, is the last step in successfully decomposing a twist. From **Equation (3.1)** it can be shown that

$$\vec{v} = (\vec{c} \times \vec{w}) + p\vec{w}. \tag{3.5}$$

If the cross product of the $\vec{c}$ and $\vec{w}$ vectors are taken, it can be shown that

$$\begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ c_x & c_y & c_z \\ w_x & w_y & w_z \end{vmatrix} = (c_y w_z - c_z w_y)\vec{i} - (c_x w_z - c_z w_x)\vec{j} + (c_x w_y - c_y w_x)\vec{k}\ . \tag{3.6}$$

Now if the resulting vector is added to the vector $p\vec{w}$ in accordance with **Equation (3.5)**, it can be shown that the equations for each component of the translational velocity vector, $\vec{v}$, is:

$$\begin{aligned} v_x &= c_y w_z - c_z w_y + p w_x \\ v_y &= c_z w_x - c_x w_z + p w_y \\ v_z &= c_x w_y - c_y w_x + p w_z . \end{aligned} \tag{3.7}$$

From these equations the location matrix may be defined as the matrix that relates the $\vec{w}$ and $\vec{v}$ vectors as

$$\begin{bmatrix} p & -c_z & c_y \\ c_z & p & -c_x \\ -c_y & c_x & p \end{bmatrix} \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} . \tag{3.8}$$

The location vector, $\vec{c}$, of a twist may be found using this matrix. The following example demonstrates how this is done.

Suppose one wishes to decompose the twist that was constructed in the previous example: $\vec{T} = [1 \quad 1 \quad 0 \quad 1 \quad 3 \quad 0]^T$. **Equation (3.1)** suggests that $\vec{w} = [1 \quad 1 \quad 0]^T$ and $\vec{v} = [1 \quad 3 \quad 0]^T$. The twist's pitch can be solved for by plugging these two vectors into **Equation (3.4)**. This pitch, *p*, is confirmed to equal 2m/rad. For this example, the location matrix equation is found using **Equation (3.8)** and is given as

$$\begin{bmatrix} 2 & -c_z & c_y \\ c_z & 2 & -c_x \\ -c_y & c_x & 2 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \\ 0 \end{bmatrix} . \tag{3.9}$$

From **Equation (3.9)**, it ban be determined that any location vector, $\vec{c}$, will satisfy the twist as long as its components are bound by the following conditions:

$$\begin{aligned} c_z &= 1 \\ c_x &= c_y \end{aligned} . \tag{3.10}$$

The orientation vector chosen from the previous example was $\vec{c} = [0 \quad 0 \quad 1]^T$, which satisfies **Equation (3.10)**.

3.2 Wrenches as Constraints

This section presents a model for mathematically describing the constraints of flexure systems. Whereas degrees of freedom are modeled as twists in screw theory, constraints are modeled as wrenches. A wrench is a 6×1 force vector that may be represented as a single line in three-space. It may be described using three parameters: (1) a 3×1 location vector, $\vec{r}$, that points from the origin of an arbitrarily defined coordinate system to any point along the wrench's line of action, (2) a 3×1 orientation vector, $\vec{f}$, that points along wrench's line of action and represents the wrench's translational or axial force, and (3) a scalar constant, *q*, that is analogous to a twist's pitch in that it couples the wrench's force with the torque. **Figure 3.4** depicts a standard wrench using these parameters.

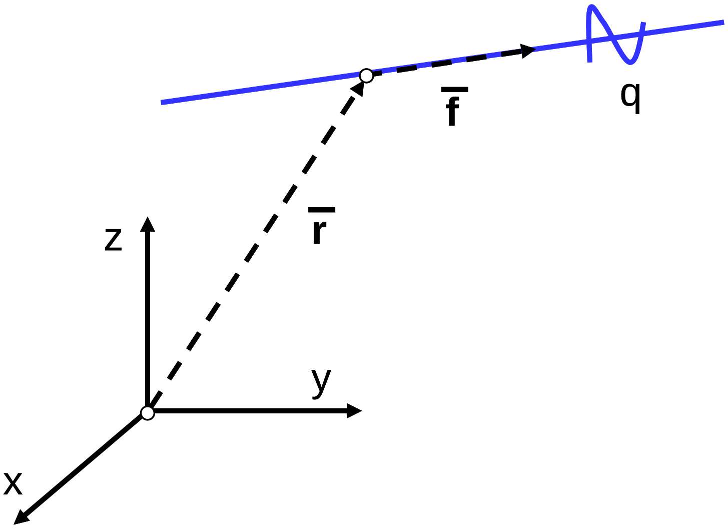


Figure 3.4: A wrench (blue line) with a location vector ($\vec{r}$), an orientation vector ($\vec{f}$), and a scalar torque value constant of *q*

The 6×1 wrench vector, $\vec{W}$, is given in **Equation (3.11)** in terms of the parameters defined above.

$$\vec{W} = \begin{bmatrix} \vec{f} \\ (\vec{r} \times \vec{f}) + q\vec{f} \end{bmatrix} = \begin{bmatrix} \vec{f} \\ \vec{\tau} \end{bmatrix} = \begin{bmatrix} f_x \\ f_y \\ f_z \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix}, \tag{3.11}$$

where $\vec{\tau}$ is a 3×1 vector that represents the wrench's rotational force or torque and $\vec{f}$ is a 3×1 vector that represents the wrench's translational or axial force. For constraints like the ones used to design flexure systems, *q* will always equal zero. This observation is true since flexure constraints are modeled as ideal constraints that are only capable of imposing axial forces on the objects they're constraining. Every other rotational or translational direction is infinitely compliant by definition of an ideal constraint. Since the ratio of axial stiffness to lateral stiffness is much greater than 1 for long, slender, compliant beams, one can reasonably model them as ideal constraints. Since the *q* values of ideal constraints always equal zero, wrenches will always be represented with a single color, blue.

Wrenches are analogous to twists. The orientation vectors, $\vec{f}$ and $\vec{w}$, are analogues; the location vectors, $\vec{r}$ and $\vec{c}$, are analogous; the vectors $\vec{\tau}$ and $\vec{v}$ are analogous; and the scalar values, *q* and *p*, are analogous. Any equation that was presented in the previous section on twists will, therefore, apply for wrenches as long as the appropriate parameters are replaced— $\vec{f}$ for $\vec{w}$, $\vec{r}$ for $\vec{c}$, $\vec{\tau}$ for $\vec{v}$, and *q* for *p*. Once one substitutes these values properly, the same equations and principles that applied for twists apply for creating and decomposing wrenches. The analogy breaks down only when one considers that $\vec{f}$ is a translational vector where $\vec{w}$ is a rotational vector and $\vec{\tau}$ is a rotational vector where $\vec{v}$ is a translational vector. The significance of these observations will be addressed shortly.

3.3 Twist and Wrench Relationship

This section presents the relationship between twists and wrenches. This relationship (a) will provide the mathematical relationship between degrees of freedom and constraints and (b) is needed to complete the FACT design method.

A twist is said to be complementary, or reciprocal, to a wrench if its dot product with the wrench equals zero [32] as shown in the following equation:

$$\vec{W} \bullet \vec{T} = \begin{bmatrix} \vec{f} & \left(\vec{r} \times \vec{f}\right) + q\vec{f} \end{bmatrix} \begin{bmatrix} \left(\vec{c} \times \vec{w}\right) + p\vec{w} \\ \vec{w} \end{bmatrix} = 0\,, \qquad (3.12)$$

where the $\vec{v}$ and $\vec{w}$ vectors have switched places within the twist vector. This switch is made so that (a) the translational vectors $\vec{f}$ and $\vec{v}$ and (b) the rotational vectors $\vec{\tau}$ and $\vec{w}$ will be multiplied with each other when the wrench is dotted with the twist. These products have units of power and may be added together. When the dot product of a wrench with a twist equals zero, the motion associated with this twist produces no power. In other words, this motion doesn’t enable the constraint to offer any resistance in directions perpendicular to its axis. In reality, a motion like this will be in a direction of least stiffness as opposed to the idealized condition of zero stiffness.

Appendix A shows how **Equation (3.12)** simplifies to the following equation:

$$p = d\tan\theta\,, \qquad (3.13)$$

where *p* is the pitch of the twist, *d* is the shortest distance between the twist (DOF) and wrench (constraint) lines, and *θ* is the skew angle between the twist and wrench lines as shown in **Figure 3.5**.

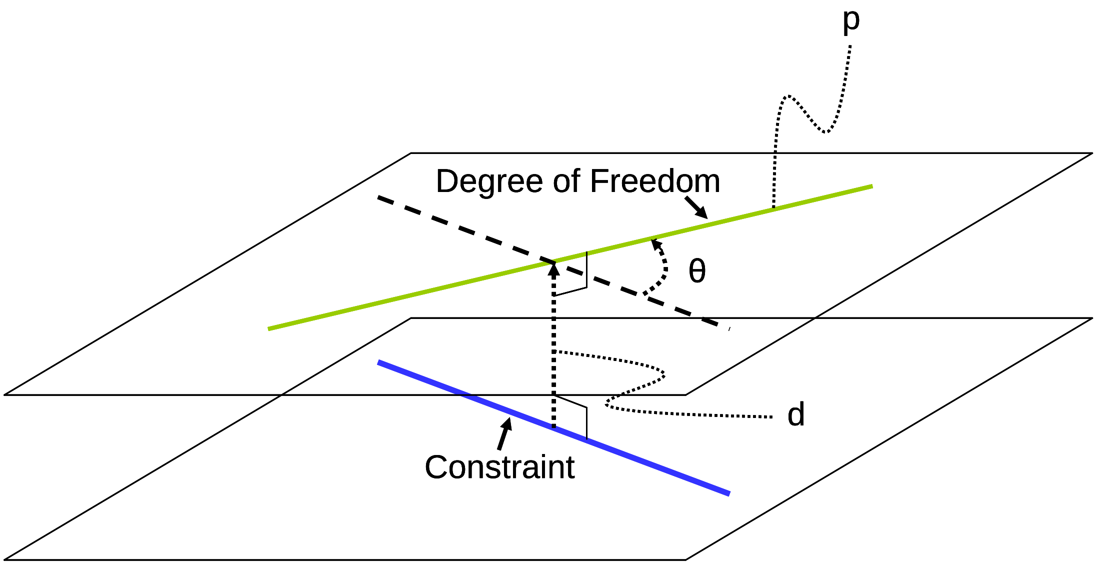


Figure 3.5: The parameters that are used to quantitatively define the relationship between a twist (green) and a wrench (blue) are *p*, *d*, and *θ*.

Equation (3.13) is a general mathematical relationship between constraints and degrees of freedom. This equation mathematically proves Blanding's Rule of Complementary Patterns[3]. Notice that when *d* is zero, the twist intersects the wrench. According to Blanding, if a line intersects a constraint, it is a pure rotational freedom line. **Equation (3.13)** confirms that when *d* is zero, the pitch of the twist is zero and the twist will be a pure rotational freedom line as shown in **Figure 3.6**.

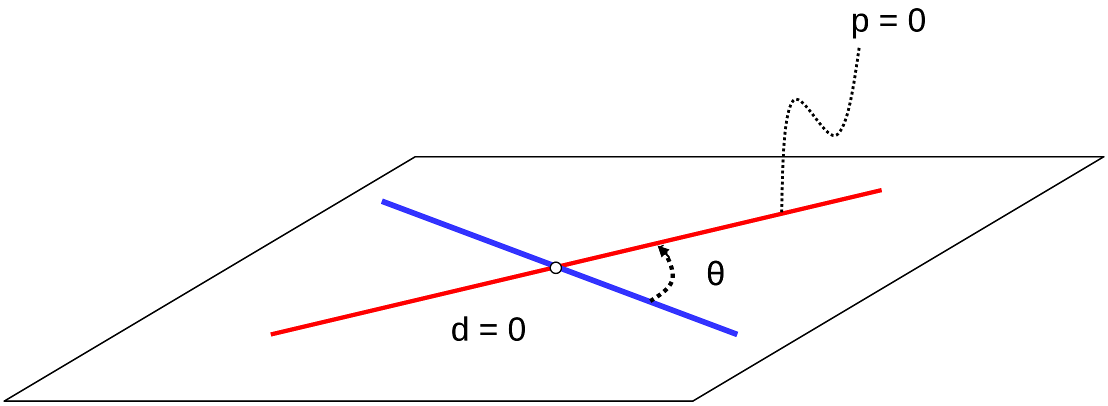


Figure 3.6: A twist will be a pure rotational freedom line if *d*=0

[3] My thanks to Haijun Su for bringing this observation to the author's attention

Blanding's Rule of Complementary Patterns states that if a line is parallel to a constraint, it is also a pure rotational freedom line since it intersects the constraint at infinity. When θ is zero such that the twist is parallel with the wrench, **Equation (3.13)** dictates that the pitch of the twist is zero. This confirms that the line is a pure rotational freedom line as shown in **Figure 3.7**.

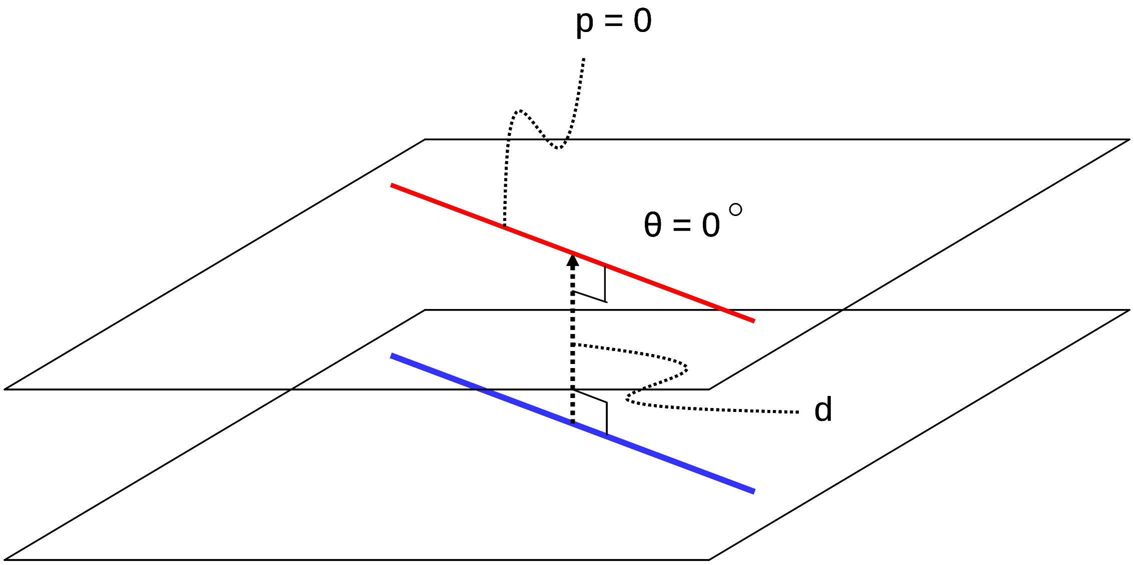


Figure 3.7: A twist will be a pure rotational freedom line if θ=0 degrees

Although Blanding's Rule of Complementary Patterns was presented in **Chapter 2** as the relationship between degrees of freedom and constraints, **Equation (3.13)** demonstrates that it is incomplete and only contains a part of the story. Blanding's Rule says nothing about a constraint's relationship with screws or pure translations. If, for instance, d is non-zero and the skew angle, θ, between the twist and wrench is 90 degrees, **Equation (3.13)** predicts that the pitch of the twist would approach infinity, which makes it a pure translation as **Figure 3.8** shows. If d is non-zero and the skew angle is between zero and 90 degrees or between zero and -90 degrees, the twist will be a screw with a finite pitch whose value is determined by **Equation (3.13)**. This was shown in **Figure 3.5**.

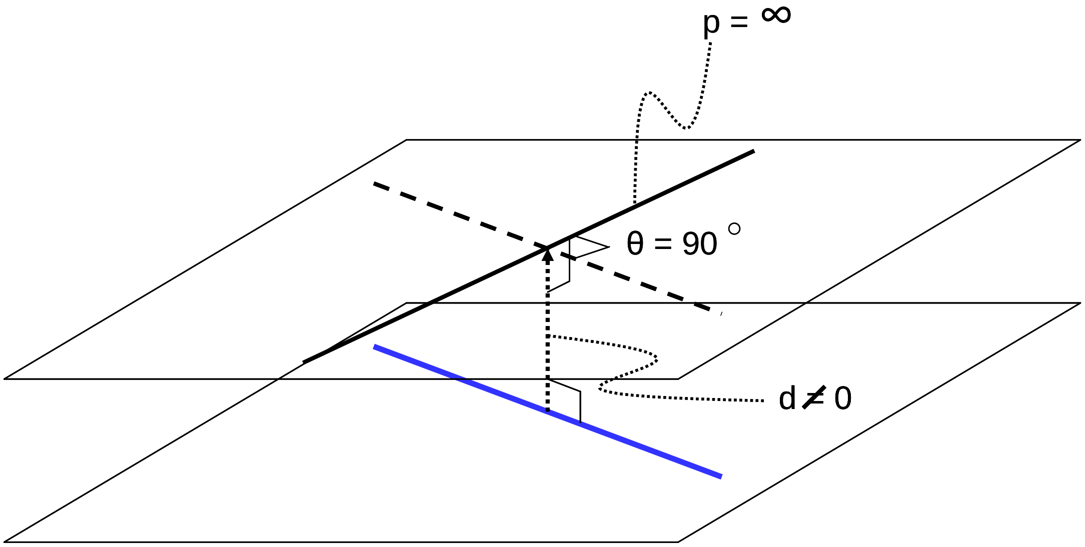


Figure 3.8: A twist will be a pure translation if *θ*=90 degrees and *d* is a non-zero distance

Suppose now that *d* is zero such that the twist intersects the wrench and that *θ* is 90 degrees such that the twist is perpendicular to the wrench. **Equation (3.13)** finds a pitch that equals zero multiplied by infinity under these conditions. Interpreting this result is difficult, so **Equation (A.4)** in **Appendix A** should be referred to for help. When a twist intersects a wrench at a right angle, this equation simplifies to the twist's pitch multiplied by zero equals zero, which will always be a true statement for any pitch value. For this special case, therefore, the twist could be a pure rotation, a pure translation, or any screw with any pitch value as **Figure 3.9** shows.

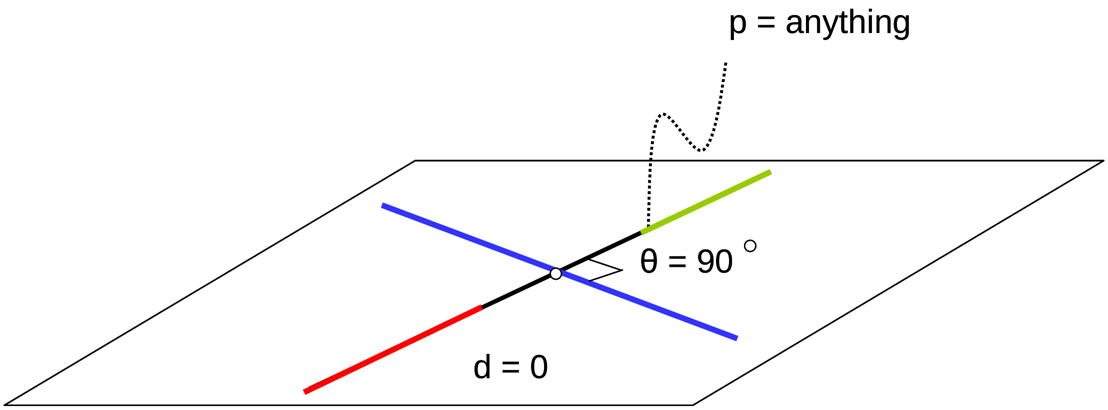


Figure 3.9: If *θ*=90 degrees and *d*=0, a twist may be a pure rotation with a zero pitch value (red), a pure translation with an infinite pitch value (black), or a screw with any finite non-zero pitch value (green)

3.4 Twists Complement Multiple Wrenches

In this section two approaches will be explored for finding a system's degrees of freedom given its constraint topology consisting of multiple constraints. One approach utilizes **Equation (3.13)** as a means of rapidly and visually finding plausible twists, and the other approach is completely mathematical and thus more thorough.

3.4.1 Visual Approach for Locating Twists

Equation (3.13) may be used to find twists that complement multiple constraints with an approach that requires visualizing twists in three-space. To do this, it is important to realize that every twist has one and only one pitch value. In other words, a twist does not exist if it has multiple pitch values. The second statement may seem more abstract than the first, but it is a better way of thinking about it for the visual approach of finding plausible twists. An example will help clarify this point.

Suppose one wished to find all the twists that complement a constraint topology that consists of two parallel constraints. One must try to visualize the locations of all possible twist lines for this system. Suppose one chose, first, to visualize a twist line that intersects the plane of the two parallel constraints and is orthogonal to a line on that plane that is also orthogonal to the two constraints as shown in **Figure 3.10**. If this is an allowable twist, it must satisfy **Equation (3.13)** for both constraints. Each constraint line will have the same skew angle, θ, with this twist line. The shortest distance lines, however, will be different, i.e., $d1$ is not equal to $d2$. **Equation (3.13)** would, therefore, assign two different pitch values to this potential twist which disqualifies it as an allowable motion. Since this twist cannot exist with two different pitch values, it fails the visual test and other locations for twists must be found that do pass the test.

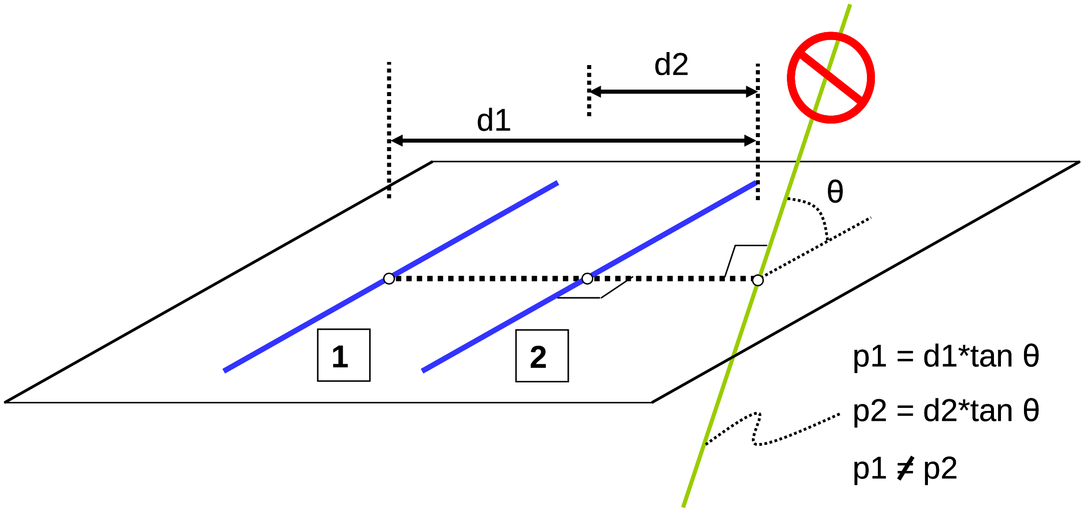


Figure 3.10: A twist (green) that fails the visual test because it possesses multiple pitch values

Suppose now one tries a twist line on a plane above the plane of the two parallel constraints shown in **Figure 3.11**. Both constraint lines share the same skew angle, θ, with the twist line. Both constraint lines also share the same shortest distance, d, with the twist line. Both constraints will, therefore, also predict the same pitch value, p, for the twist. Since this twist line has a single pitch value according to both constraints, it is a degree of freedom for the system. This should be the case for any twist on any plane parallel or coincident to the plane of the constraints. If the planes are coincident, the twists will always be pure rotations with zero pitch values according to Blanding's Rule of Complementary Patterns and according to **Equation (3.13)**. Also if the skew angle, θ, is zero for any twist on any of these planes, the twist will be a pure rotation.

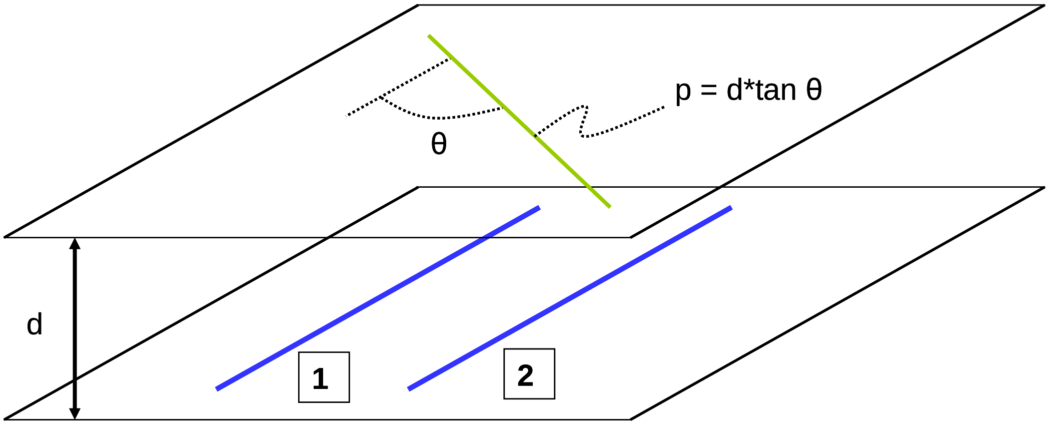


Figure 3.11: A twist (green) that passes the visual test because it has a single pitch value

In order to find every allowable twist, more twist locations must be considered. Suppose a twist line lies on a plane that is orthogonal to the two parallel constraints shown in **Figure 3.12**. Although the two shortest distances from the twist to each constraint, *d*1 and *d*2, will be different, the skew angle between the twist and each constraint will be 90 degrees. According to **Equation (3.13)**, the predicted value of the twist's pitch is infinite. Therefore, any twist on any plane that is orthogonal to both parallel constraints will be an allowable twist that is a pure translation.

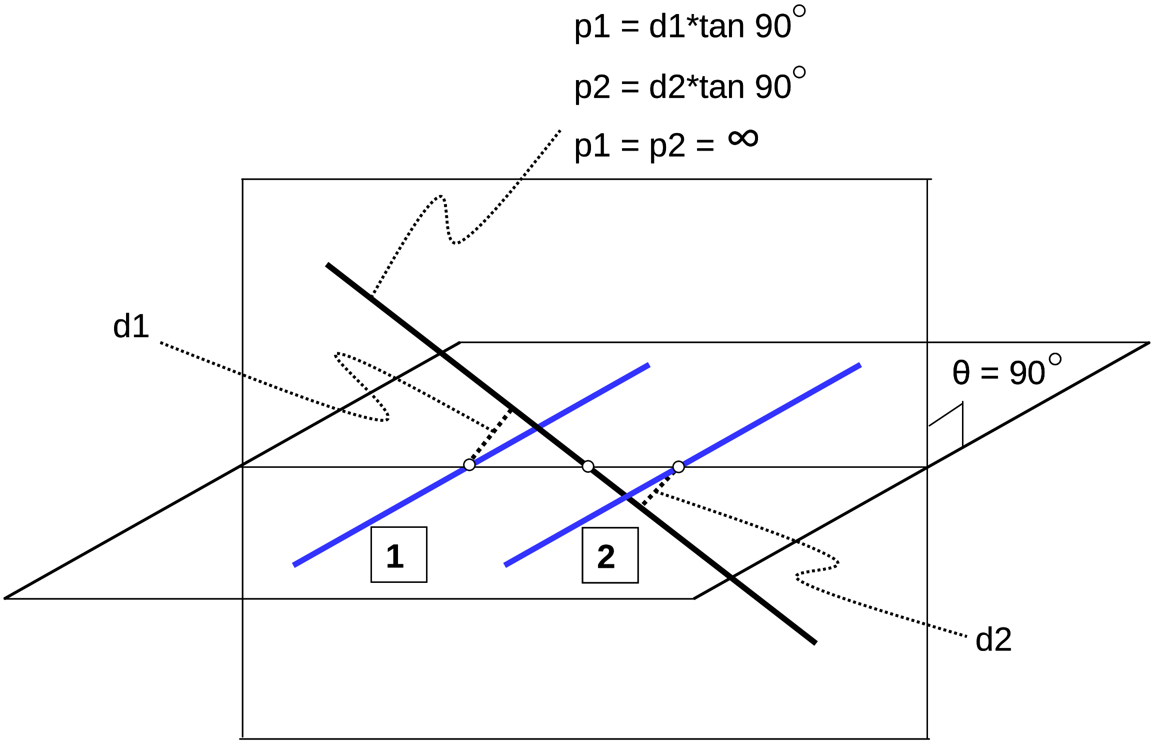


Figure 3.12: A twist (black) that passes the visual test because it has a single pitch value of infinity

One could continue to check for more allowable twists at an infinite number of locations, but by performing just a few more tests at several other locations, one quickly gains confidence that all the allowable twists have been found for this constraint system. This hypothesis that all the twists have been found, however, cannot conclusively be proven using the visual approach, but can be mathematically confirmed using the more thorough approach that will be presented in the next section.

3.4.2 Mathematical Approach for Locating Twists

In a way, the mathematical approach for locating twists has already been presented, but this approach was not presented as a method for finding twists for a multi-constraint system. Every twist that complements a system with a single constraint is essentially the null space of a matrix that contains this constraint's wrench vector. This statement is simply another way of re-describing **Equation (3.12)**. The allowable twists of a multi-constraint system where *q*=0 may be found by solving for the null space of a matrix that contains the system's wrenches as shown in the equation below:

$$\begin{bmatrix} \vec{f}_1 & \left(\vec{r}_1 \times \vec{f}_1\right) \\ \vec{f}_2 & \left(\vec{r}_2 \times \vec{f}_2\right) \\ \vdots & \vdots \\ \vec{f}_n & \left(\vec{r}_n \times \vec{f}_n\right) \end{bmatrix} \begin{bmatrix} \left(\vec{c} \times \vec{w}\right) + p\vec{w} \\ \vec{w} \end{bmatrix} = \vec{0}, \tag{3.14}$$

for *n* constraints. This approach will now be applied to the previous example of two parallel constraints.

One first converts the parallel constraint lines into a mathematical form by placing them in a coordinate system. Suppose the first constraint lies on the x-axis and the second parallel constraint intersects the y-axis an arbitrary distance, *d*, away from the origin as shown in **Figure 3.13**. The first constraint's location vector is given by $\vec{r}_1 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$, and its orientation vector is given by $\vec{f}_1 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T$. The second constraint's location vector is given by $\vec{r}_2 = \begin{bmatrix} 0 & d & 0 \end{bmatrix}^T$, and its orientation vector is given by $\vec{f}_2 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T$.

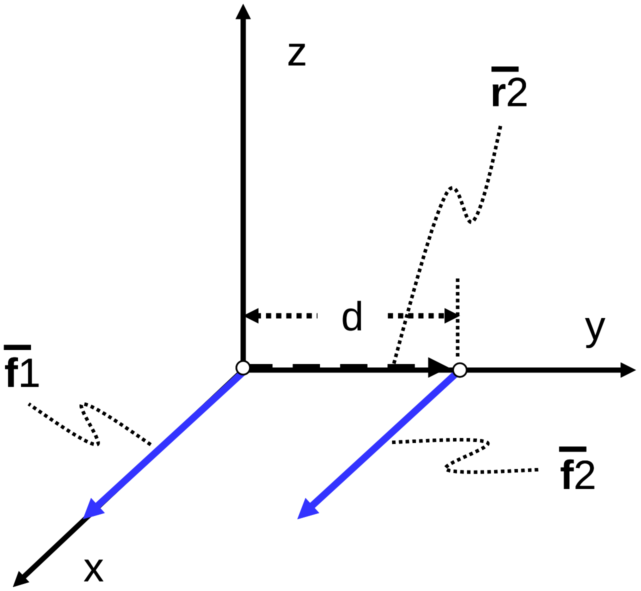


Figure 3.13: Two parallel constraint lines with the parameters necessary to describe them as wrenches

From these vectors, two wrenches may be constructed using **Equation (3.11)** and placed within a 2×6 wrench matrix where each row corresponds to each constraint. The null space of this matrix may be found using

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & -d \end{bmatrix} \begin{bmatrix} \vec{v} \\ \vec{w} \end{bmatrix} = \vec{0}\,. \tag{3.15}$$

The null space of this particular wrench matrix is a linear combination of four independent 6×1 vectors. To describe these vectors as conventional twists, their $\vec{v}$ and $\vec{w}$ vectors are switched so that they are expressed in the form shown in **Equation (3.1)**. The result is shown below:

$$A\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + B\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + C\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} + D\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} A \\ B \\ 0 \\ 0 \\ C \\ D \end{bmatrix} = \vec{T}\,, \tag{3.16}$$

where *A*, *B*, *C*, and *D* may be any real numbers. The 6×1 twist vector at the far right of **Equation (3.16)** is the complete mathematical representation of every possible twist for the system of two parallel constraints. This resultant twist's rotational and translational velocity vectors, $\vec{w}$ and $\vec{v}$, are the following:

$$\begin{aligned} \vec{w} &= \begin{bmatrix} A & B & 0 \end{bmatrix} \\ \vec{v} &= \begin{bmatrix} 0 & C & D \end{bmatrix}. \end{aligned} \tag{3.17}$$

Note that regardless of the *C* and *D* values in the $\vec{v}$ vector that the directional vector, $\vec{w}$, will always cause the twist's line to lie on a plane that is parallel or coincident with the x-y plane on which the two parallel constraints lie since its z-component is always zero. This observation is true as long as both *A* and *B* are not simultaneously zero. Note also, however, that if the *A* and *B* constants are simultaneously zero that the twist vector becomes a pure translation that points in the direction of $\vec{v}$ (If this is not clear now, **Chapter 4** will address this concept in greater depth). This pure translational twist will always lie on the y-z plane since the $\vec{v}$ vector's x-component is always zero. This plane will always be orthogonal to the two constraints. Therefore, as long as the resultant twist is not a zero vector (i.e. all four constants *A*, *B*, *C*, and *D* simultaneously equal zero) every allowable twist that complements the two parallel constraints will lie on a plane that is parallel or coincident with the plane of the constraints, or they will be pure translations and will lie on a plane that is orthogonal to both constraints. This conclusion is consistent with the conclusion made using the visual approach.

The reader should also have noted that although there are an infinite number of twists that describe the kinematics of this parallel constraint system, only four of those twists are independent. Remember also that the constraint topology consisted of only two constraints. This fact is consistent with Maxwell's observation described in **Equation (2.1)**. This observation will be discussed in greater detail in **Chapter 5** where it will be mathematically proven for every system.

CHAPTER 4:

“Projective Geometry”

This chapter applies basic concepts of projective geometry to find the relation between pure rotational twists and pure translational twists. This relation is useful to the FACT method because it creates multiple options for visually expressing the same degree of freedom.

4.1 Pure Translations

This section provides insights about pure translations. **Chapter 3** demonstrated that a twist with an infinite pitch is a pure translation. To conceptually understand why this is the case, recall that the pitch of a twist is the ratio of the translational motion along the twist’s line to the rotational motion about the twist’s line. If, for example, an object translates a large distance as it simultaneously rotates a small amount, that object’s motion may be described by a twist with a large pitch. It makes sense, therefore, that the motion of an object that purely translates without rotating at all is described by a twist with a pitch that approaches infinity.

A pure translational twist’s pure rotational velocity vector, $\vec{w}$, must also be a zero vector since a pure translational motion involves no rotational motion. A pure translational twist vector will, therefore, always be expressed as

$$\vec{T} = \begin{bmatrix} \vec{0} \\ \vec{v} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ v_x \\ v_y \\ v_z \end{bmatrix}, \qquad (4.1)$$

where $\vec{v}$ is a translational velocity vector whose components are never all zero. This fact may cause the reader to question how the $\vec{v}$ vector could have any non-zero components given its dependency on the $\vec{w}$ vector that has all zero components. The dependence of the $\vec{w}$ vector on the $\vec{v}$ vector was given in **Equation (3.5)** from **Chapter 3** and is given again here as

$$\vec{v} = (\vec{c} \times \vec{w}) + p\vec{w}. \tag{4.2}$$

Equation (4.2) helps answer the question. Note that although the cross product in the first half of the equation will become a zero vector as $\vec{w}$ approaches a zero vector, the second half of the equation will not be a zero vector as $\vec{w}$ approaches a zero vector because the pure translational twist's pitch, *p*, approaches infinity. If, therefore, pure translational twists' pitch values did not all approaching infinity, pure translational twists would all be zero vectors. This could not be the case since zero vectors contain no information about translations.

More differences between pure translational twists and other twists are worth mentioning. Other twists such as pure rotations or screws require the specification of a location vector, $\vec{c}$, and an orientation vector, $\vec{w}$, with a pitch value, *p*, to define its twist line in three-space. Since, however, no rotational motion occurs with a pure translation, the location vector, $\vec{c}$, is not necessary to specify. This conclusion is mathematically confirmed by noting that since the cross product in **Equation (4.2)** will always drop out of the equation when the $\vec{w}$ vector is a zero vector, the value of the location vector, $\vec{c}$, does not matter. In other words, the location of a pure translational twist in three-space is unimportant, as long as the direction of translation is correctly specified.

The orientation vector of pure translations is also different from the orientation vector of other twists. Recall that the orientation vector of pure rotations and screws is the rotational velocity vector, $\vec{w}$. Since this vector is a zero vector for pure translations, however, the velocity vector, $\vec{v}$, is the orientation vector of a pure translation. This vector is the vector that points along the twist's line and determines the direction of translation.

The following example will help clarify pure translations. Suppose one wished to decompose the twist $\vec{T} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 5 \end{bmatrix}^T$ to find its twist line in three-space for a box located at the origin as shown in **Figure 4.1**. One knows that this twist is a pure translation with a pitch of infinity because its vector is expressed in the form given by **Equation (4.1)**. The translational velocity vector, $\vec{v}$, is $\vec{v} = \begin{bmatrix} 0 & 0 & 5 \end{bmatrix}^T$. Since this vector is the orientation vector for pure translations, the twist's line is oriented parallel to the z-axis and will allow the box to translate in this direction. It is also important to note that it doesn't matter where the black twist line is located. Every translational twist line pointing along the z-axis will cause the box to move with the same translational motion as any other translational twist line pointing in the same direction located at any other position as shown in **Figure 4.1**.

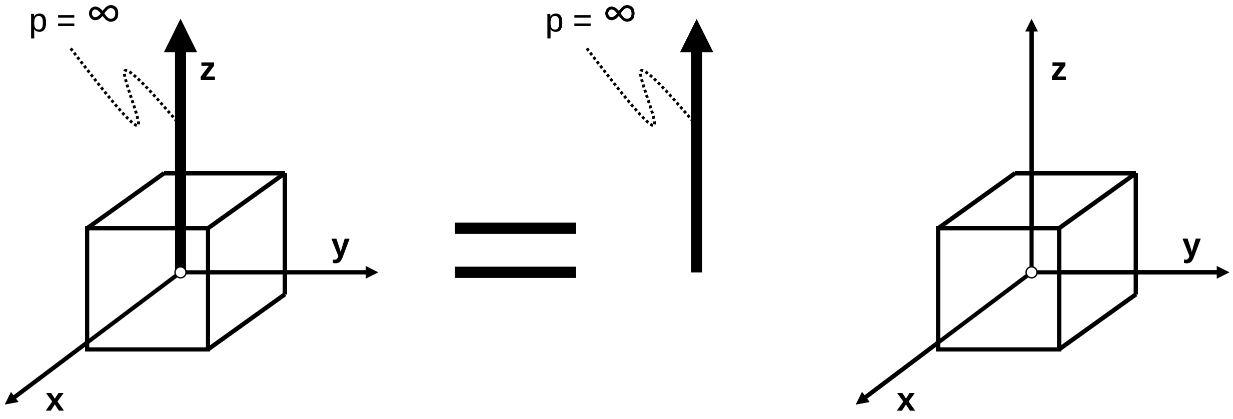


Figure 4.4: A box translating along the z-axis may be expressed using any pure translational twist line (thick black) that is parallel to the z-axis. The location of the pure translation doesn't matter, only its orientation.

4.2 Pure Rotational "Hoops"

This section explains how pure rotations may be used to express pure translations.

To begin, basic facts proven by projective geometry will be reviewed [33]. The first fact states that:

A line is a circle with an infinite radius.

This fact may be conceptually understood by considering the definition of curvature. The curvature at any point on any curved line is defined by the inverse of the radius of the circle whose curvature is identical to the curvature at the point of interest on the curved line as shown in **Figure 4.2**. This radius, *r*, is called the radius of curvature.

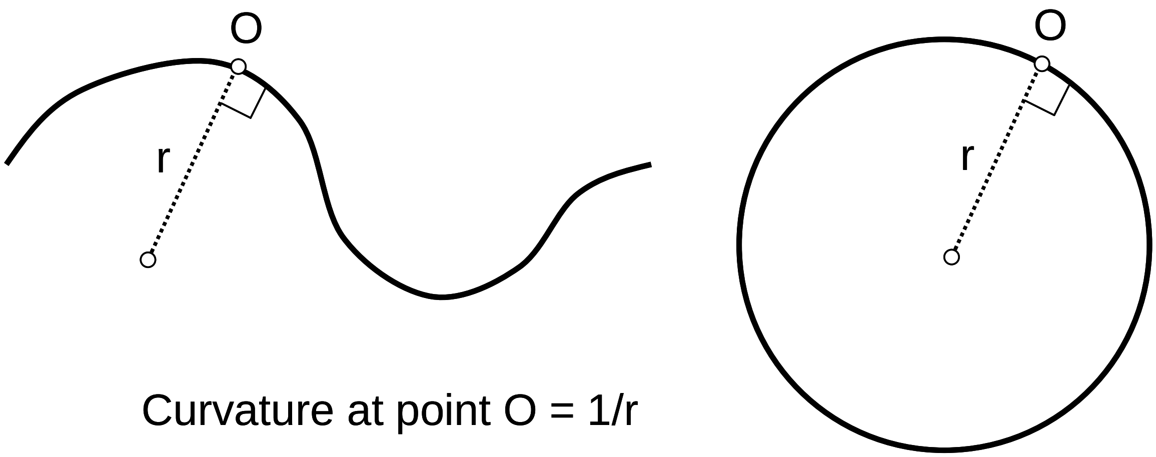


Figure 4.2: Definition of curvature for a point O on a curved line.

A line, by definition, is linear and has no curvature. The curvature of a circle will approach zero as its radius approaches infinity. A line is, therefore, essentially a circle with an infinite radius. Projective geometry texts provide a more extensive and thorough proof of this fact.

Another important axiom of projective geometry states that:

Any two planes are incident with at least one line [34].

This statement is true for any planes including planes that are parallel. To better understand this fact conceptually, imagine two planes that intersect at a line in finite space. As the planes become more and more parallel, this line of intersection moves farther and farther away until the planes become parallel at which point the line of intersection is at infinity as shown in **Figure 4.3**. This line must be a circle with an infinite radius because the two parallel planes shown as squares in the figure will intersect each other at the same line (dashed orange) along all four of the square's edges at infinity.

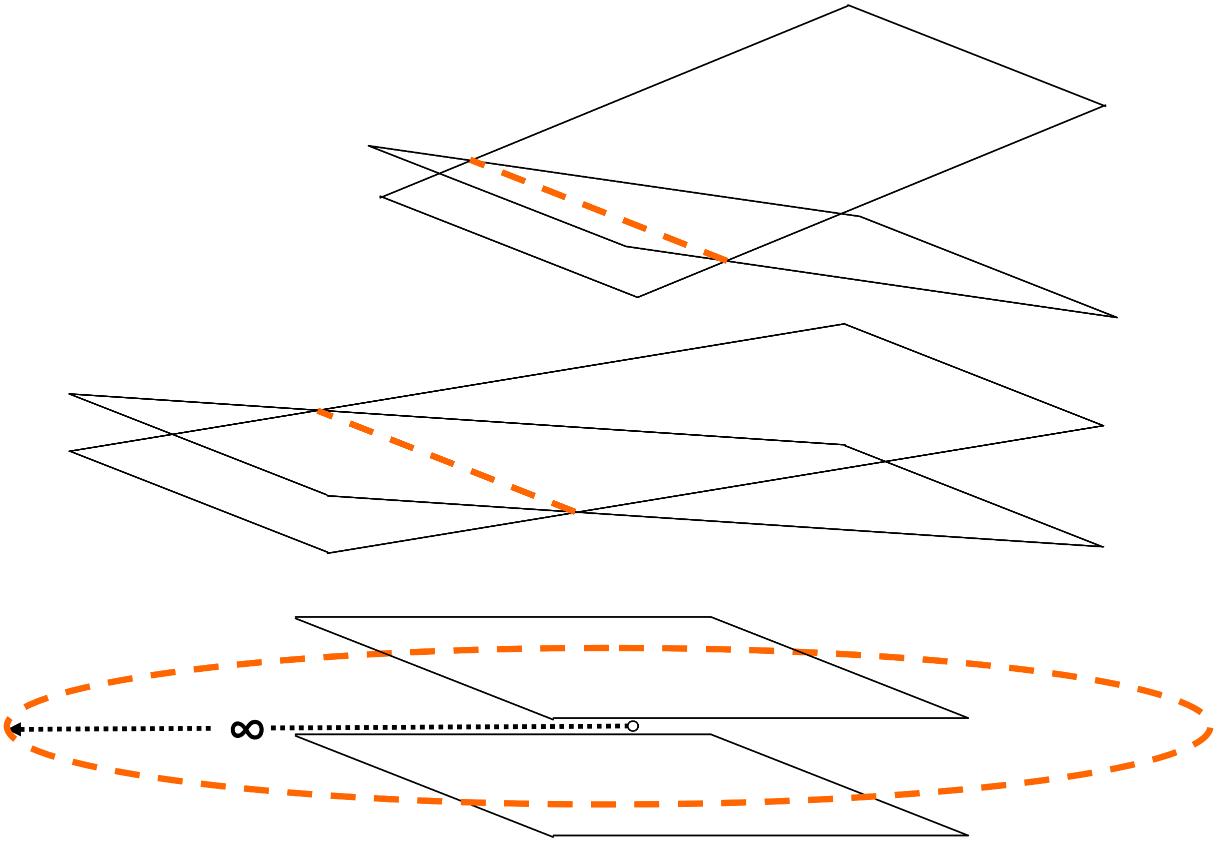


Figure 4.3: Parallel planes intersect at a line (dashed orange) at infinity

The concept depicted in **Figure 4.3** is similar to the concept depicted in **Figure 2.5** from **Chapter 2**. This observation leads to another important axiom of projective geometry:

Any two coplanar lines are incident with at least one point [34].

Note the duality of this axiom with the previous axiom. Recall from **Chapter 2** that this axiom applies to parallel lines as well. The reader may, however, erroneously deduce that if parallel lines intersect, they would intersect at two distinct points infinitely far apart. In actuality, however, these two points are the same point at infinity because both parallel lines are circles with an infinite radius.

These concepts will now be applied to finding the relationship between translational and rotational degrees of freedom. Consider a block in finite space that may vertically translate up and down along a pure translational twist line (black) as shown in **Figure 4.4**. Note that this translational motion may equivalently be expressed as a pure rotational twist line (red) that is orthogonally skew to the translational line and is infinitely far away from the block translating. Since this pure rotational freedom line is at infinity, it is best depicted as a circle with an infinite radius. In this thesis, such pure rotational circles that represent pure translational motions will be called pure rotational hoops.

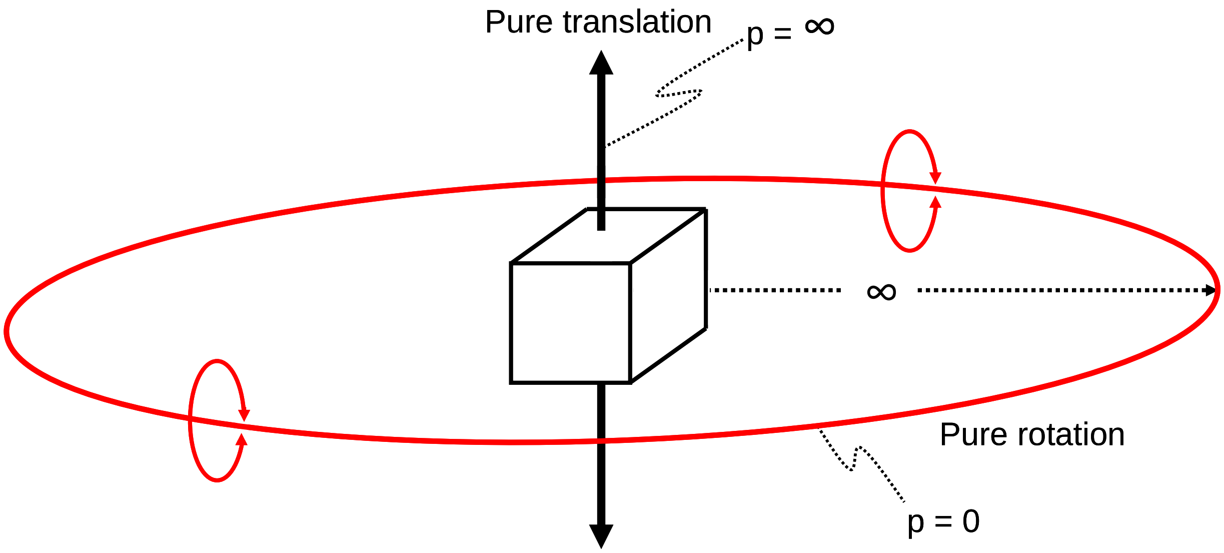


Figure 4.4: A pure translation expressed as a pure rotational hoop

Also note that since they exist at infinity, pure rotational hoops require no finite location vectors similar to pure translational twist lines. Similar to pure translations, therefore, the orientation of pure rotational hoops is all that matters. The normal vector of the plane of the pure rotational hoop always points in the direction of the translation it represents.

To prove this relationship, consider an example of a system consisting of an arbitrary number of constraint lines that lie in arbitrary locations on two parallel planes as shown in **Figure 4.5**. If Blanding's Rule of Complementary Patterns is applied to find the system's freedom lines, it can be determined that only one line exists that intersects every constraint line on the two parallel planes at least once. This freedom line is infinitely far away from the system's constraints and is shown as a pure rotational hoop whose normal vector is orthogonal to the two parallel planes in **Figure 4.5**. Since this pure rotational hoop exists within the system, one would expect the existence of a pure translation that points in a direction normal to the two parallel planes because pure rotational hoops and pure translations represent equivalent motions. **Equation (3.13)** from

Chapter 3 can be used to verify that such a pure translation exists. Although the shortest distance, *d*, between the twist line in **Figure 4.5** and each constraint line varies depending on where the constraints lie relative to the twist line, all twist lines that are orthogonal to both planes will always have a skew angle of 90 degrees with respect to every constraint line. According to **Equation (3.13)**, therefore, such twists will be pure translations with infinite pitch values as expected.

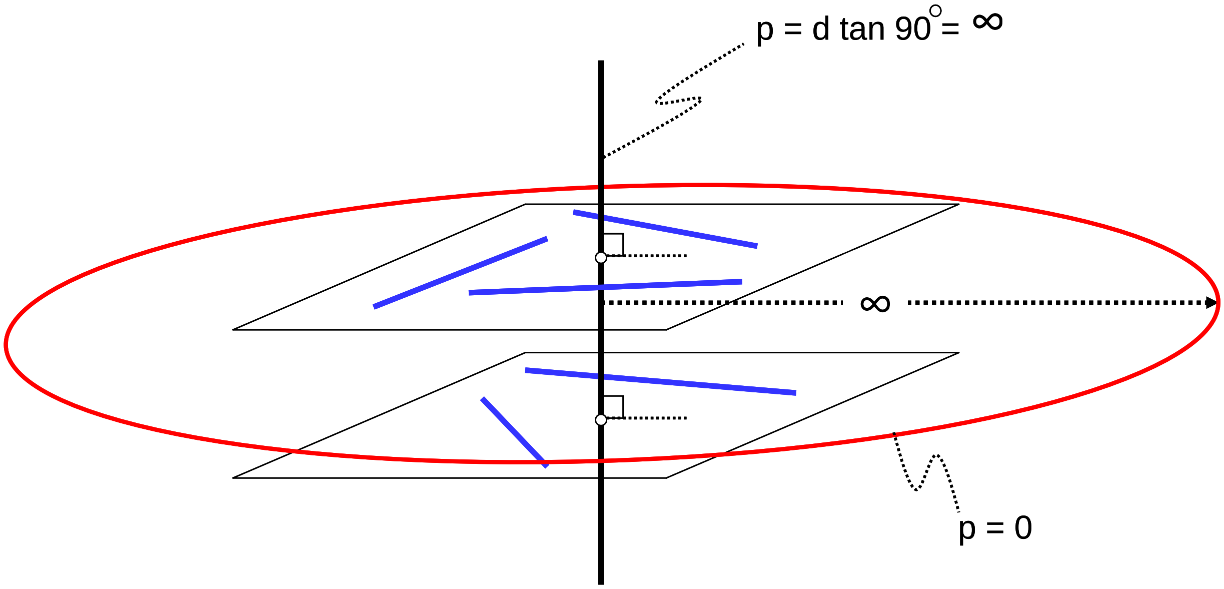


Figure 4.5: A system of constraints that lie on two parallel planes contains a pure rotational hoop and a pure translation in a direction normal to the planes. Both of these twists represent the same motion.

4.3 Multiple Pure Translations

This section presents observations of systems with multiple pure translations and provides tools that are needed to express infinite sets of pure translations as infinite sets of pure rotational hoops.

First note that no system exists that contains more than three independent pure translational twists. This observation is true since only three orthogonal independent directions exist in three dimensions. Furthermore, upon inspection of **Equation (4.1)**, note that only three non-zero components exist that could be varied to make three independent translational twists.

A system of three independent pure translations could be represented as a sphere containing an infinite number of pure translational lines (black) that all intersect at a single point. Only free standing objects possess three independent pure translations (unless the system may be constrained by constraints with non-zero *q* values). This observation becomes clear when one considers adding a single ideal constraint to a free standing object. The first degree of freedom the object loses will always be a pure translation along the axis of the constraint by definition of an ideal constraint.

A system of two independent pure translations could be represented as a disk of pure translational twist lines (black) as shown in **Figure 4.6**. These same motions could also be represented as pure rotational hoops. Imagine an infinite number of rotational hoops with normal vectors that point in directions that correspond to the directions of every pure translation within the black disk. These hoops would fill a space that resembles a beach-ball-like sphere that has a radius of infinity and contains an infinite number of pure rotational freedom lines on its surface. These freedom lines all intersect at a single point at infinity. This point is also intersected by the dashed black line that is orthogonal to the black disk of pure translations and passes through its center point as shown in **Figure 4.6**.

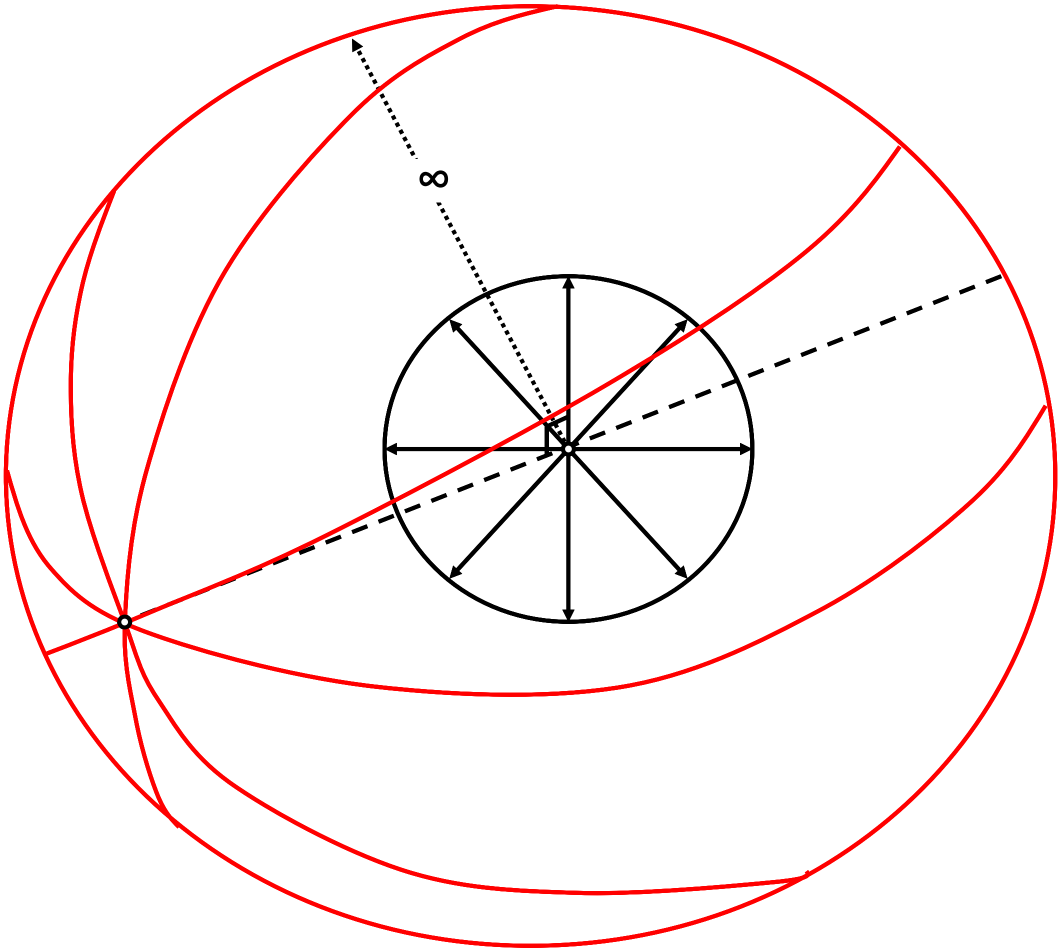


Figure 4.6: Two independent pure translational motions represented as a disk of pure translations (black) and as a sphere with an infinite radius consisting of an infinite number of pure rotational hoops (red)

If the pure rotational hoops (red) shown in **Figure 4.6** are still unclear, consider an infinite number of pure rotational lines that are parallel to the dashed black line and are all separated an equal distance from it. These pure rotational lines will more accurately represent the motions expressed by the pure translations in the disk the farther they are moved away from this dashed black line. Only when these pure rotational lines are infinitely far away from the dashed black line, will they exactly express the same motions as the pure translational disk. When these pure rotational lines are infinitely far away, they may be thought of as pure rotational hoops as was discussed in the previous section. Projective geometry demonstrates that all of these parallel lines will all intersect at the same point at infinity [34].

4.4 Finding Hoops

This section presents a quick method for identifying the existence and orientation of pure rotational hoops for any system with any arbitrary constraint topology.

The first step is to determine how many constraints exist in the system and then to identify their orientation vectors, $\vec{f}$. The second step is to take the cross product of all of these orientation vectors.

- If all of the resulting vectors are zero vectors, the system has two independent pure translations and may be expressed as the beach-ball-like sphere of hoops from **Figure 4.6** where the $\vec{f}$ vectors will all be parallel and will intersect the sphere's point at infinity.
- If at least one of the resulting vectors is a non-zero vector and if all of the non-zero vectors point in the same direction (if there is more than one non-zero vector), then there exists a single translation that may be expressed as a rotational hoop whose normal vector points in the direction of the non-zero vector or vectors resulting from the cross products.
- If any of the resulting non-zero vectors point in different directions with respect to each other, no rotational hoops will exist and not a single pure translation will exist in the system.

The following example uses this method. Suppose one wished to know if the system of three constraints shown in **Figure 4.7** has any pure translations or pure rotational hoops. First note the orientation vectors of the three constraints: $\vec{f}_1$, $\vec{f}_2$, and $\vec{f}_3$. The first two constraint lines are parallel. The third constraint line is skew relative to the first two parallel constraint lines with a skew angle of 90 degrees. Second, use the right-hand rule to determine the directions of all the vectors that result in taking the cross product of all of the orientation vectors. The cross product of the first two constraint orientation vectors is a zero vector. The cross product of the first constraint orientation vector with the third constraint orientation vector is a vector that points in a direction orthogonal to the plane that the first two constraints lie on. The cross product of the second constraint orientation vector with the third constraint orientation vector is also a vector that points in a direction orthogonal to the same plane. Now that the direction of all of the resulting vectors is known, check them with the conditions listed under the bullet points form the method discussed earlier. For this example, the second bullet point applies since the two resulting vectors that aren't zero vectors point in the same direction. This system will, therefore, have a pure translation that may be expressed as a rotational hoop whose normal vector, $\vec{v}$, points in the direction of the two non-zero resulting vectors, which is the direction orthogonal to the plane on which the first two constraints lie.

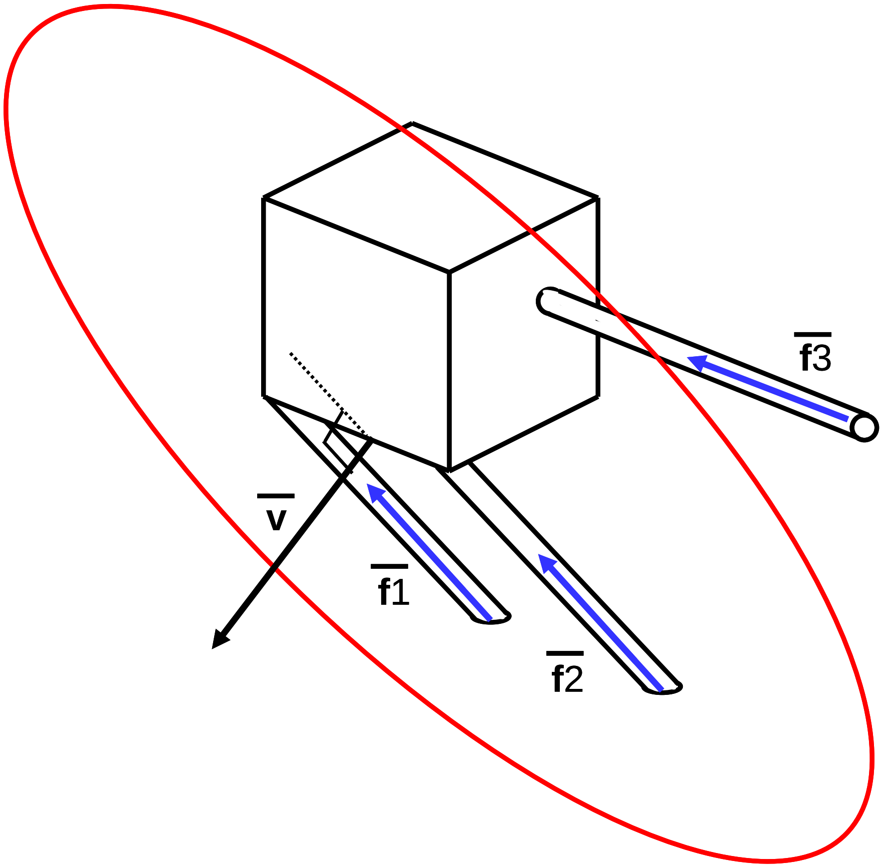


Figure 4.7: An example for identifying the rotational hoops/translations of a system quickly by taking the cross product of all the constraint line orientation vectors, $\vec{f}$.

CHAPTER 5:

"Freedom and Constraint Space"

This chapter introduces the concept of freedom and constraint spaces as visual representations of the complete kinematic and constraint topology of a flexure system

5.1 Freedom Space

This section introduces the concept of freedom spaces. To begin, the concept of a freedom set is defined. A freedom set is a space that contains an infinite number of twist lines that may be represented using a simple geometry such as a sphere, box or plane. The two red disks of freedom lines from the system studied in **Chapter 2** shown in **Figure 2.10**, were each examples of freedom sets. Although each line in those disks is a pure rotational freedom line, freedom sets may also be geometric entities that contain pure translations or screws.

An example should clarify the concept of identifying freedom sets. Consider the system shown in **Figure 5.1** of a block constrained by two constraints whose constraint lines (blue) intersect inside the block.

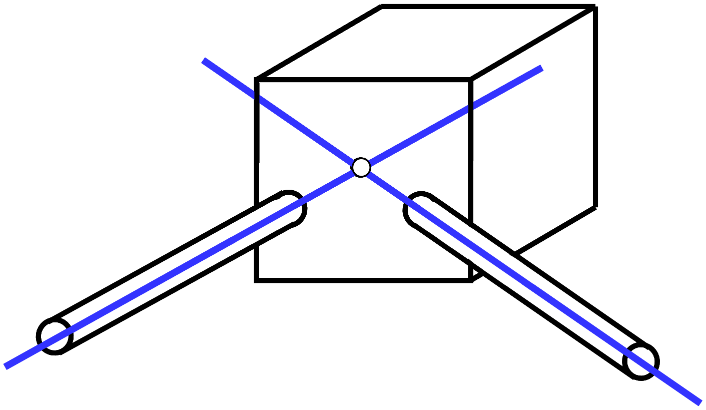


Figure 5.1: A block constrained by two compliant beams with constraint lines (blue) that intersect inside the block.

The allowable motions of this system include every pure rotational freedom line that intersects both constraint lines at least once according to Blanding's Rule of Complementary Patterns. As one searches for all of these lines that satisfy this condition, one notices that familiar geometric entities emerge that act as sets that contain infinite numbers of these lines. The sphere in **Figure 5.2**, for instance, represents a set of every line that intersects the intersection point of the two constraint lines (blue). Every line inside this sphere is a pure rotational freedom line (red) of the system. The plane outlined in red in **Figure 5.2** also represents a set of infinite freedom lines that all lie on the same plane of the two intersecting constraint lines. Every line on this plane will intersect both constraint lines at least once in finite space or at infinity if they are parallel to either of the constraint lines. The pure rotational hoop also intersects both constraint lines. The pure translation that this hoop represents points in the direction of the normal vector of the red plane.

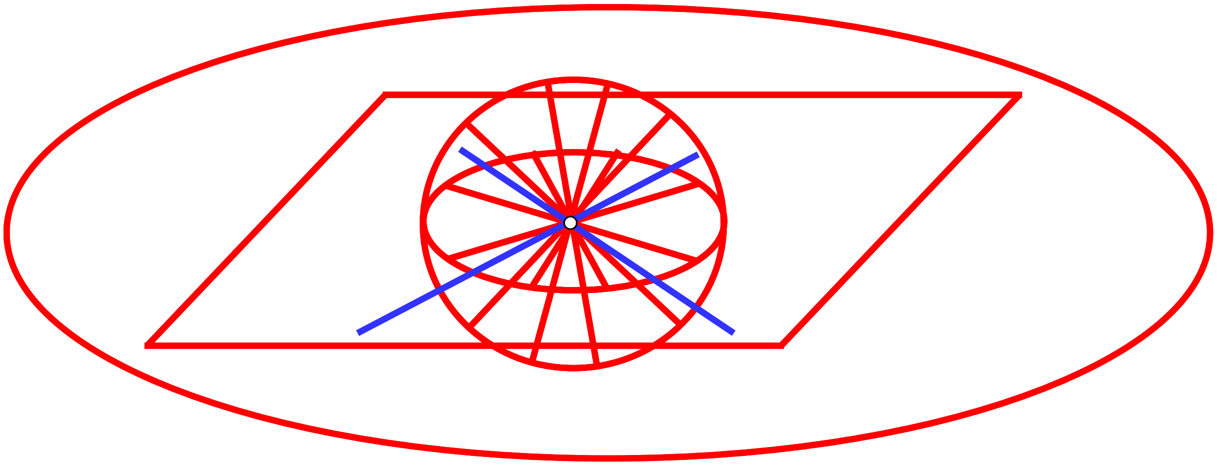


Figure 5.2: The pure rotational freedom lines (red) for the block system with two constraint lines (blue)

Not only have all of the pure rotational freedom lines been successfully located, but geometric entities have been determined that represent all of these lines collectively and thus the need to draw each line individually has been eliminated. In actuality, there are more twist lines for this system that have not yet been identified, but they are neither pure rotations nor pure translations. They are screws with non-zero finite pitch values that may be found using the visual or mathematical approach discussed in **Chapter 3**. If one was to find these screws, they could also be represented visually using geometric entities, but since this is more complicated this particular example will be revisited in **Chapter 7**. For now, it is sufficient to have found three freedom sets that contain every allowable pure rotational freedom line—a sphere, a plane, and a hoop. These freedom sets are each shown in **Figure 5.3**.

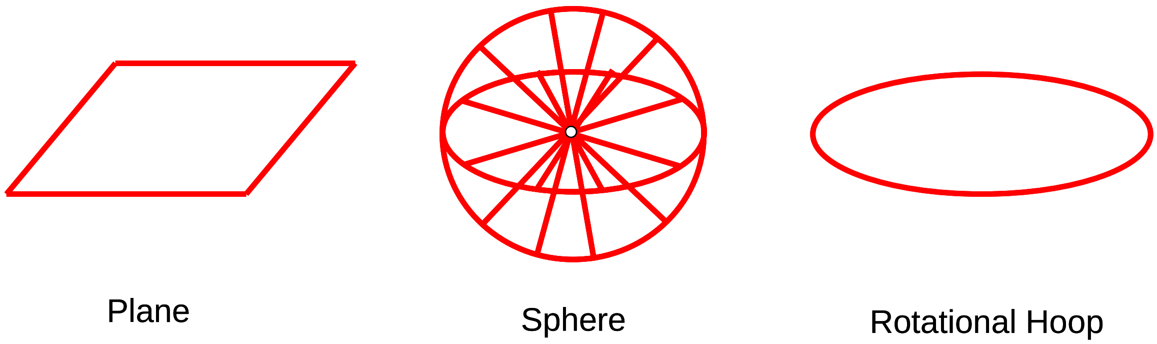


Figure 5.3: The three pure rotational freedom sets for the block system of two intersecting constraints

Now that the concept of a freedom set is understood, the concept of a freedom space is ready to be defined. The freedom space of a system is the combination of all of the system's freedom sets. This includes freedom sets containing pure translations or screws. Essentially, a system's freedom space is a visual representation of the complete kinematics of that system.

Since ideal constraints were used to develop this theory, the twists inside freedom spaces represent every infinitely compliant motion that the rigid block of the system could move with for small displacements. Consequently, everywhere that there isn't a twist line in three-space represents a motion of infinite stiffness that the rigid block could not move with. In reality, however, every conceivable line in three-space would have some finite stiffness associated with it. Therefore, this theory finds the freedom spaces that contain only practical twist lines of greatest compliance.

5.2 Constraint Space

This section introduces the concept of constraint spaces and presents a visual and mathematical approach for finding these spaces for any given system of twists.

5.2.1 Redundant and Non-redundant Constraints

Before constraint space may be properly introduced, it is important to first understand the concepts of redundant and non-redundant constraints. This section discusses these concepts.

To best explain the concept of constraint redundancy, the example of the block with two intersecting constraint lines from the previous section will be considered again. Suppose a third constraint is added to the block such that all three constraint lines lie in the same plane and intersect at the same point inside the block as shown in **Figure 5.4**.

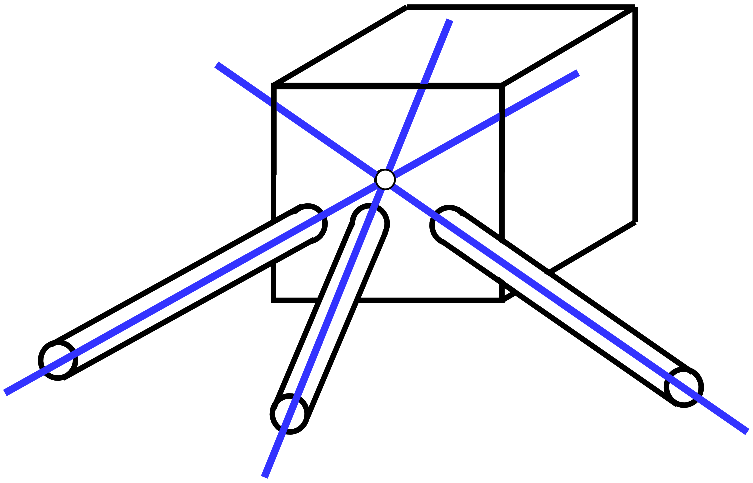


Figure 5.4: The block from the previous example with an extra constraint whose constraint line lies in the same plane as the other two constraint lines and intersects them at the same point.

If the twists that complement this new system of three constraints were found, the freedom space of this system would be identical to the freedom space of the system containing only two intersecting constraints. This third constraint, therefore, has no effect on the kinematics of the block. In fact, no matter how many ideal constraints are added to the block such that they all lie on the same plane and intersect at the same point, the freedom space of the system will not change. In practice, the flexure system's stiffness, load capacity, and stability change but the system's kinematics do not. To be convinced that the freedom space remains unchanged, consider a disk of infinite constraint lines (blue) and use Blanding's Rule of Complementary Patterns to find all freedom lines (red) that intersect all of these lines at least once as shown in **Figure 5.5.** Note that the freedom space is identical to the system with only two intersecting constraint lines.

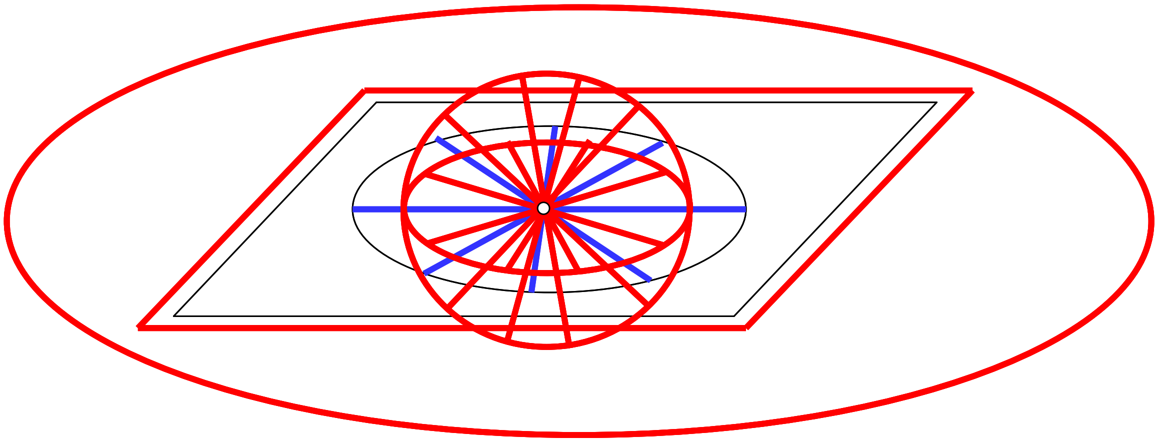


Figure 5.5: A disk containing an infinite number of constraint lines (blue) has the same freedom space (red) as the previous example of two intersecting constraint lines

Note also that if less than two constraints are selected from the disk, the freedom space of the system changes completely. The concluded can be drawn, therefore, that two non-redundant constraints exist within the system shown in **Figure 5.5**. A constraint is non-redundant if when it is added to or removed from a system, the kinematics or the freedom space of that system changes. Furthermore, a constraint is redundant if when it is added to or removed from a system, the freedom space of that system remains unchanged. Every constraint selected from the blue disk in **Figure 5.5**, therefore, is redundant as long as two of them have already been selected.

Again consider the block in **Figure 5.4**. Two of the three constraints are non-redundant while one of them is redundant. The reader may wonder which of the three constraints is redundant. The answer is that no constraint may be singled out as the redundant constraint. Any of the three constraints could be removed from the block and the kinematics of the system would remain the same.

This observation makes more sense in the context of the mathematical definition of redundant and non-redundant constraints. This definition will now be presented. Recall that constraints are modeled as wrenches. The number of wrench vectors that are independent in a particular system of constraints is the number of non-redundant constraints that that system has. Likewise, the number of wrench vectors that are dependent is the number of redundant constraints in that system.

The same example of the block with three intersecting constraints will be used to demonstrate this concept. First, express the three constraints as wrench vectors by using the location and orientation vectors, $\vec{r}$ and $\vec{f}$ respectively, defined in **Figure 5.6**.

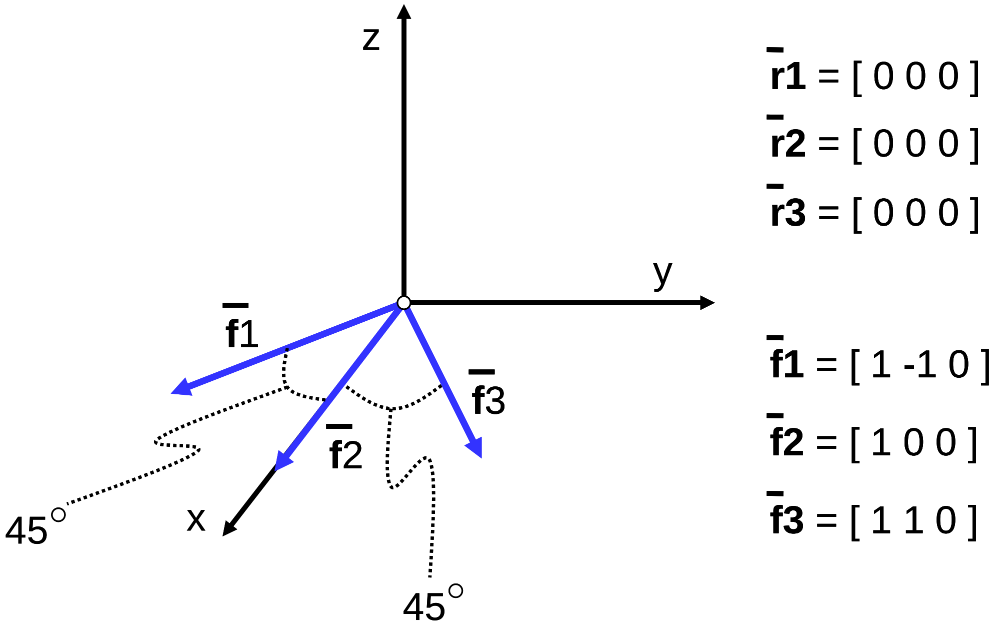


Figure 5.6: Three wrenches (blue) from the previous block example of three intersecting constraint lines shown in Figure 5.4.

Once these three wrench vectors have been constructed, create a wrench matrix by using each wrench vector as a row inside a 3×6 matrix. The number of independent and dependent wrenches may be determined by applying Gaussian Elimination to this wrench matrix as shown in **Figure 5.7**. The number of non-zero row vectors in the resulting matrix is the number of independent wrenches and the number of zero row vectors in the resulting matrix is the number of dependent wrenches. Consequently, this information reveals how many constraints are non-redundant and how many constraints are redundant.

$$\begin{array}{c} \overline{W}1 \longrightarrow \\ \overline{W}2 \longrightarrow \\ \overline{W}3 \longrightarrow \end{array} \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \xrightarrow{\text{Gaussian Elimination}} \begin{bmatrix} \textcircled{1} & 0 & 0 & 0 & 0 & 0 \\ 0 & \textcircled{1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Figure 5.7: Gaussian elimination of the wrench matrix reveals two non-redundant constraints and one redundant constraint for the system of three intersecting constraint lines. Pivots are circled in red.

As expected, two of the constraints are non-redundant and one of the constraints is redundant for the system of three intersecting constraint lines that lie in the same plane and intersect at a common point.

5.2.2 Constraint Sets and Space

A constraint set is a space that contains an infinite number of constraint lines that may be represented using a simple geometry such as a sphere, box or plane. The concept is identical to the concept of a freedom set except a constraint set is a space that contains constraint lines instead of twists. The blue disk of constraint lines in **Figure 5.5** is an example of a constraint set.

The concept of constraint space is also very similar to the concept of freedom space. The constraint space of a system is the combination of all of the system's constraint sets. Since all constraints will be modeled as ideal constraints in this thesis, every constraint space will contain blue constraint lines that are modeled as wrenches with *q* values equal to zero only. Essentially, the constraint space of a system is the system's complete constraint topology. Constraint space is a visual representation of all the possible locations from which a designer could select a constraint without changing the freedom space of the system. Since the blue disk of constraint lines from **Figure 5.5** is the only constraint set for the example studied in this section, it is also the complete constraint space of the system.

Unlike freedom space, however, the constraint sets within a system's constraint space must be labeled with the appropriate number of non-redundant constraints that exist within each set to inform the designer of the minimum number of constraints to select in order to ensure the desired system kinematics. The constraint set from the constraint space of **Figure 5.5**, for example, should be labeled with a two since it contains two non-redundant constraints.

For more complex systems, it is not always clear which constraints should be selected from within which constraint sets such that they are non-redundant and such that the system will move with the desired degrees of freedom. The concept of sub-constraint spaces will, therefore, be discussed in detail as a solution to this problem in **Chapter 8**, whereas the concept is only introduced in this chapter now for completeness. Every system's sub-constraint spaces lie within its complete constraint space. These sub-constraint spaces are also made of constraint sets that are labeled with the number of non-redundant constraints that exist within each set. These concepts will be clarified in later chapters.

5.2.3 Finding Constraint Space

This section presents two methods for determining a system's constraint space given the system's freedom space. One of these methods is a visual approach and the other is a mathematical approach.

5.2.3.1 Visual Approach for Locating Wrenches

To demonstrate the visual approach for finding a system's constraint space given its freedom space, the three-constraint, rectangular block system from **Chapter 2** will be studied again in this section. This system is shown in **Figure 5.8**.

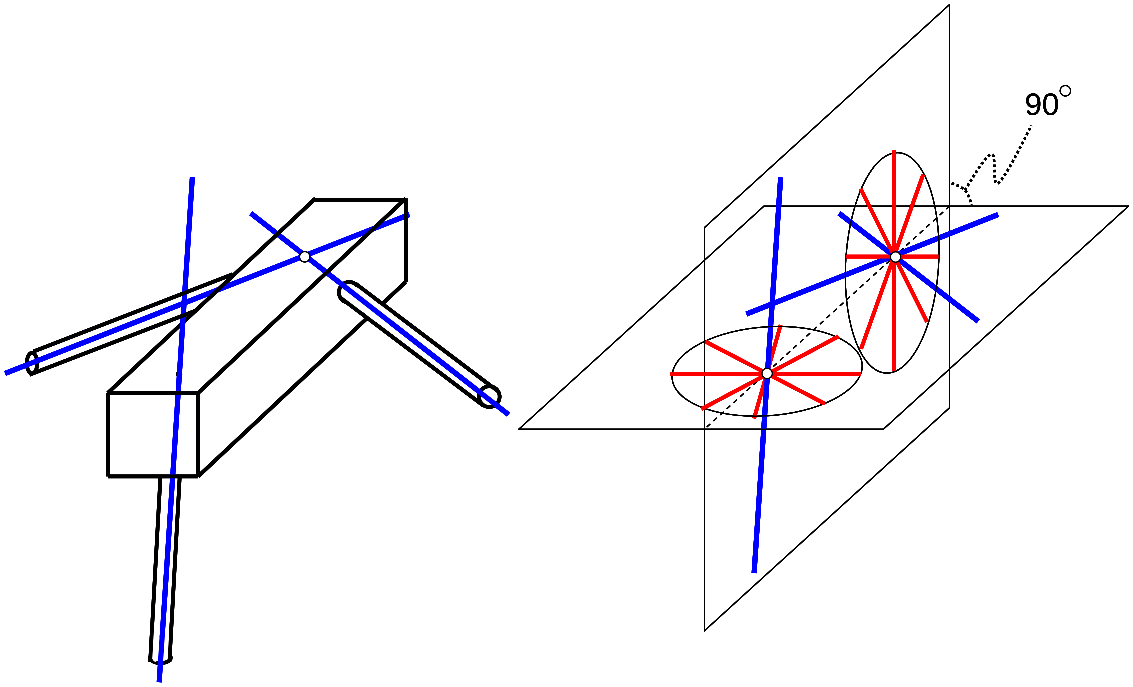


Figure 5.8: A block constrained with three constraints (blue) shown next to its pure rotational freedom space (red).

In **Chapter 2**, the conclusion was drawn that this system's freedom space consisted of two pure rotational disk freedom sets (red) as shown above. If the methods discussed in **Chapter 3** for finding this system's freedom space are applied, one would find that these disks alone do not make up the complete freedom space. Sets of screws also exist that have not been shown, which are not necessary for finding the complete constraint space of the system by using the visual approach of this section. In fact, the visual approach of this section will generally be able to find any system's complete constraint space even if all that is known are the pure rotations of the system.

The visual approach has, in a way, already been taught in the form of Blanding's Rule of Complementary Patterns. Until now, however, this rule has been used for finding every pure rotational freedom line for a system given a number of constraint lines. Now the same rule is used for finding every constraint line for a system given every pure rotational freedom line. Since every freedom line intersects every constraint line at least once, every constraint line must also intersect every freedom line at least once.

If one, therefore, wished to locate every constraint line (blue) within the system shown in **Figure 5.8**, one would need to find every line that intersects every freedom line (red) also shown in the figure. Note that every line that lies on the horizontal plane will intersect every freedom line in the red disk that also lies on that plane either in finite space or at infinity. Only the lines, however, on that horizontal plane that also intersect the center point of the other vertical disk of freedom lines will intersect every freedom line at least once. All the lines that satisfy these conditions may be represented as a disk of infinite constraint lines (blue) shown in the first picture in **Figure 5.9**. By this same reasoning, one can also find another disk of constraint lines (blue) that lies on the vertical plane and shares a center point with the disk of freedom lines on the horizontal plane shown in the second picture of **Figure 5.9**.

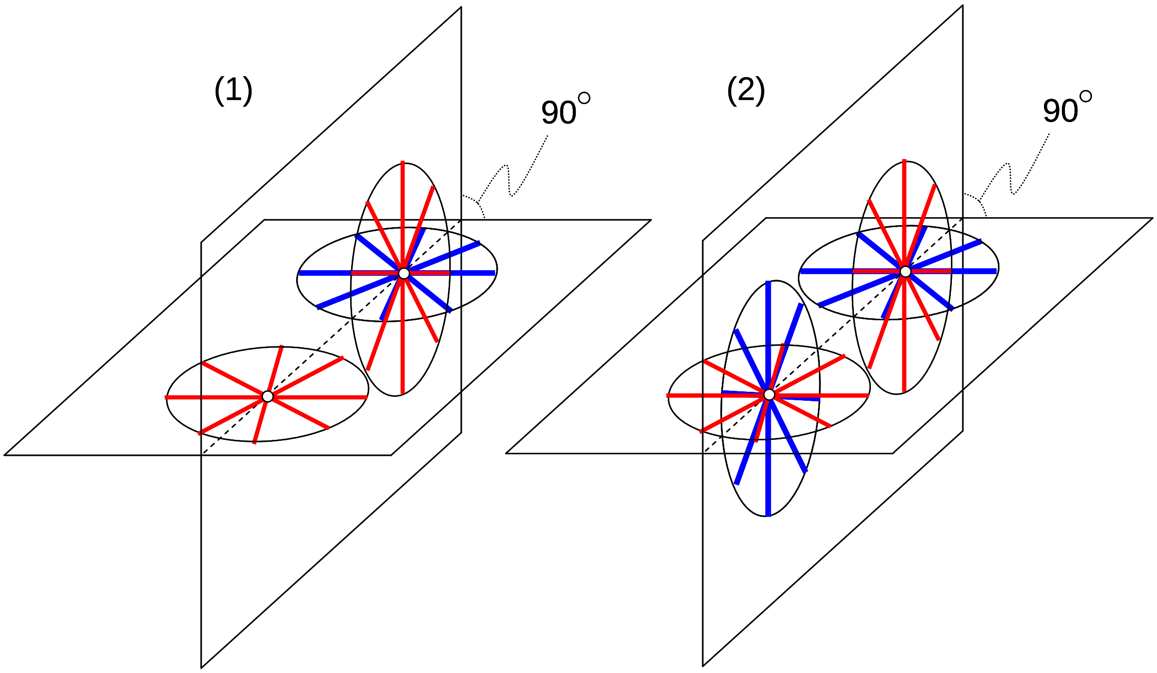


Figure 5.9: Finding all the constraint lines (blue) that intersect all the freedom lines (red) at least once

These two disks of constraint lines represent every line that satisfies Blanding's Rule of Complementary Patterns and consequently represent the system's complete constraint space. Each of these disks is a constraint set. Notice that the constraint lines of the original three constraints constraining the rectangular block shown in **Figure 5.8** are contained within the two disk constraint sets as shown with dark blue lines in the first picture of **Figure 5.10**. The second picture in **Figure 5.10** is a picture of the complete constraint space of the system with the two constraint sets labeled for instructing the designer in appropriately selecting non-redundant constraints from within each of the spaces. To properly select three non-redundant constraints for this system, therefore, **Figure 5.10** reads that any two constraints may be selected from within one of the disks, but only one constraint may be selected from within the other disk. That constraint cannot lie on the dashed intersection line of the vertical and horizontal planes.

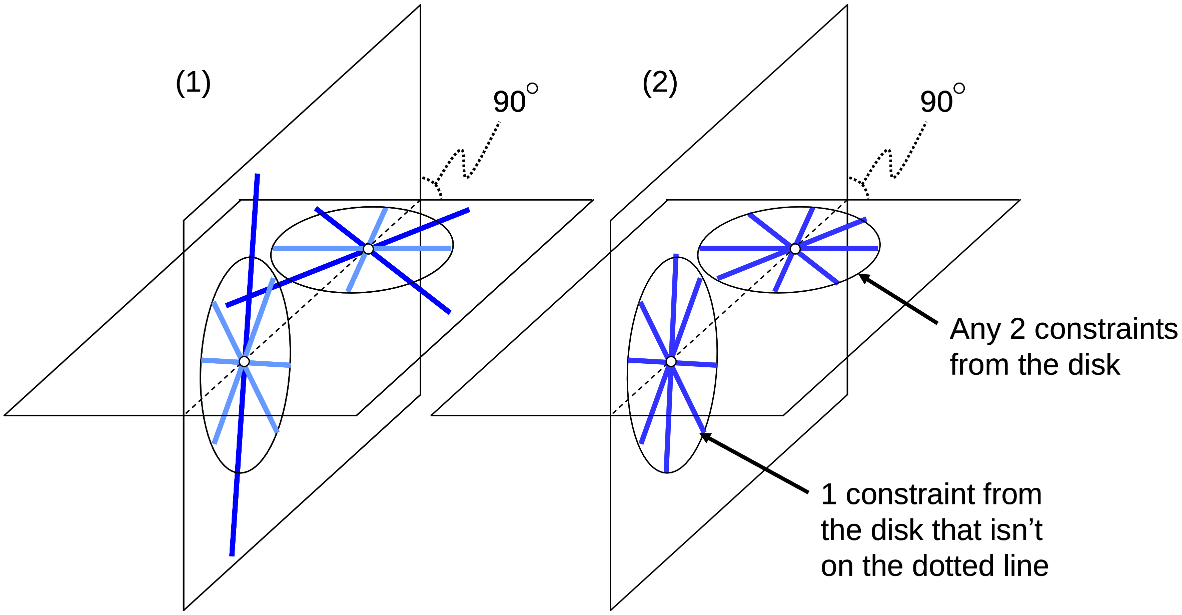

(1)
90°
(2)
90°
Any 2 constraints from the disk
1 constraint from the disk that isn't on the dotted line

Figure 5.10: Original three constraint lines (dark blue) lie within the system's complete constraint space

5.2.3.2 Mathematical Approach for Locating Wrenches

The same example used in **Section 5.2.3.1** to demonstrate the visual approach will be used again in this section to demonstrate the mathematical approach for finding a system's constraint space.

First determine the fewest number of twists needed to mathematically find the system's constraint space. One could prove that the three constraints in the example from **Figure 5.8** are non-redundant by first constructing a 3×6 wrench matrix and then by performing Gaussian Elimination on this matrix to check for row dependency similar to the approach described in the example from **Section 5.2.1**. Since these three constraints are non-redundant, **Equation (2.1)** from **Chapter 2** can be used to deduce that the freedom space of this system consists of three independent twists. Although the two pure rotational freedom set disks shown in **Figure 5.8** contain an infinite number of pure rotational twists, only three of these twists are necessary for describing the entire freedom space of the system.

Now select three specific independent twists from the freedom space of the system shown in **Figure 5.8**. Suppose the first two of these three pure rotational twists are selected from the disk on the horizontal plane and the third pure rotational twist is selected from the disk on the vertical as shown in **Figure 5.11**. Using the location vectors, $\vec{c}$, and the orientation vectors, $\vec{w}$, given in **Figure 5.11** and recalling that all the pitch values must equal zero because the twists are all pure rotations, construct three independent twist vectors.

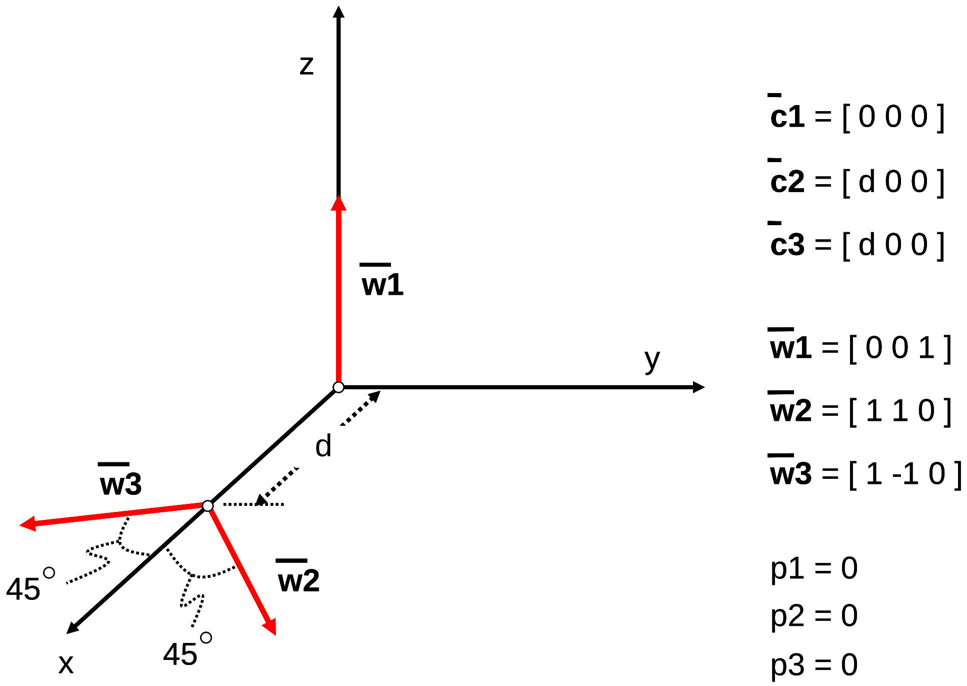


Figure 5.11: Three independent pure rotational twists (red) from the freedom space of the system shown in Figure 5.8.

These three twist vectors may be stacked inside a 3×6 matrix called a twist matrix where each row corresponds to one of the independent twists. The null space of this matrix represents the constraint space of the system and may be shown as

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & d \\ 1 & -1 & 0 & 0 & 0 & -d \end{bmatrix} \begin{bmatrix} \vec{\tau} \\ \vec{f} \end{bmatrix} = \vec{0}\,. \tag{5.1}$$

The null space of this particular twist matrix is a linear combination of three independent 6×1 vectors. To describe these vectors as conventional wrenches, their $\vec{\tau}$ and $\vec{f}$ vectors are switched so that they are expressed in the form shown in **Equation (3.11)**. The result is shown below:

$$A\begin{bmatrix}1\\0\\0\\0\\0\\0\end{bmatrix}+B\begin{bmatrix}0\\1\\0\\0\\0\\0\end{bmatrix}+C\begin{bmatrix}0\\0\\1\\0\\-d\\0\end{bmatrix}=\begin{bmatrix}A\\B\\C\\0\\-Cd\\0\end{bmatrix}=\vec{W}\,, \tag{5.2}$$

where *A*, *B*, and *C* may be any real numbers. The 6×1 wrench vector at the far right of **Equation (5.2)** is the complete mathematical representation of every possible wrench that complements the freedom space shown in **Figure 5.8**. This resultant wrench's axial force and torque vectors, $\vec{f}$ and $\vec{\tau}$, are the following:

$$\begin{aligned}\vec{f} &= \begin{bmatrix}A & B & C\end{bmatrix}\\ \vec{\tau} &= \begin{bmatrix}0 & -Cd & 0\end{bmatrix}.\end{aligned} \tag{5.3}$$

Recall, however, that for the purposes of this thesis, not every mathematically possible wrench must be found. Only the wrenches that model ideal compliant constraints with *q* values that equal zero should be found. One can, therefore, filter these unwanted answers out of the resultant wrench vector from **Equation (5.2)** by setting the *q* value equal to zero and by defining this *q* value in terms of $\vec{f}$ and $\vec{\tau}$ as

$$q = \left(\frac{\vec{f}\bullet\vec{\tau}}{\vec{f}\bullet\vec{f}}\right). \tag{5.4}$$

Equation (5.4) is analogous to the equation for the pitch of a twist given in **Equation (3.4)**. If **Equation (5.3)** is plugged into **Equation (5.4)** and *q* is set to zero, this equation simplifies to

$$-BCd = 0. \tag{5.5}$$

The only allowable wrenches are the ones, therefore, that make **Equation (5.5)** a true statement. This statement is true only when either *B* or *C* equals zero.

When *B* equals zero, the wrenches of interest are expressed as $\vec{W} = \begin{bmatrix} A & 0 & C & 0 & -Cd & 0 \end{bmatrix}^T$ and have orientation vectors expressed as $\vec{f} = \begin{bmatrix} A & 0 & C \end{bmatrix}^T$ for all real values of *A* and *C* such that the orientation vectors will always point in directions parallel to the x-z plane. To find the corresponding location vectors when *B* equals zero, apply the analogous location matrix equation from **Equation (3.8)** to these wrenches as

$$\begin{bmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{bmatrix} \begin{bmatrix} A \\ 0 \\ C \end{bmatrix} = \begin{bmatrix} 0 \\ -Cd \\ 0 \end{bmatrix}, \tag{5.6}$$

where the diagonal is zero because q=0. **Equation (5.6)** may be simplified into two equations written as

$$\begin{aligned} r_y &= 0 \\ Ar_z - Cr_x &= -Cd. \end{aligned} \tag{5.7}$$

If r_z is set equal to zero, a common location vector for all the wrenches is found when *B*=0. This location vector is $\vec{r} = \begin{bmatrix} d & 0 & 0 \end{bmatrix}^T$. In light of this wrench decomposition, every possible wrench for this system when *B*=0 can now be expressed as a disk of constraint lines that lies on the x-z plane and has a center point that lies on the x-axis a distance of *d* away from the origin as shown in **Figure 5.12**.

When *C* equals zero, the wrenches of interest are expressed as $\vec{W} = \begin{bmatrix} A & B & 0 & 0 & 0 & 0 \end{bmatrix}^T$ and have orientation vectors expressed as $\vec{f} = \begin{bmatrix} A & B & 0 \end{bmatrix}^T$ for all real values of *A* and *B* such that they will always point in directions parallel to the x-y plane. The location matrix equation could be applied to this decomposition to determine the location vectors, $\vec{r}$, when *C*=0, but since

$\vec{\tau} = [0 \quad 0 \quad 0]^T$, a common location vector for all the wrenches at this condition is known to be $\vec{r} = [0 \quad 0 \quad 0]^T$. In light of this wrench decomposition, every possible wrench for this system when *C*=0 can now be expressed as a disk of constraint lines that lies on the x-y plane and has a center point that lies at the origin as shown in **Figure 5.12**.

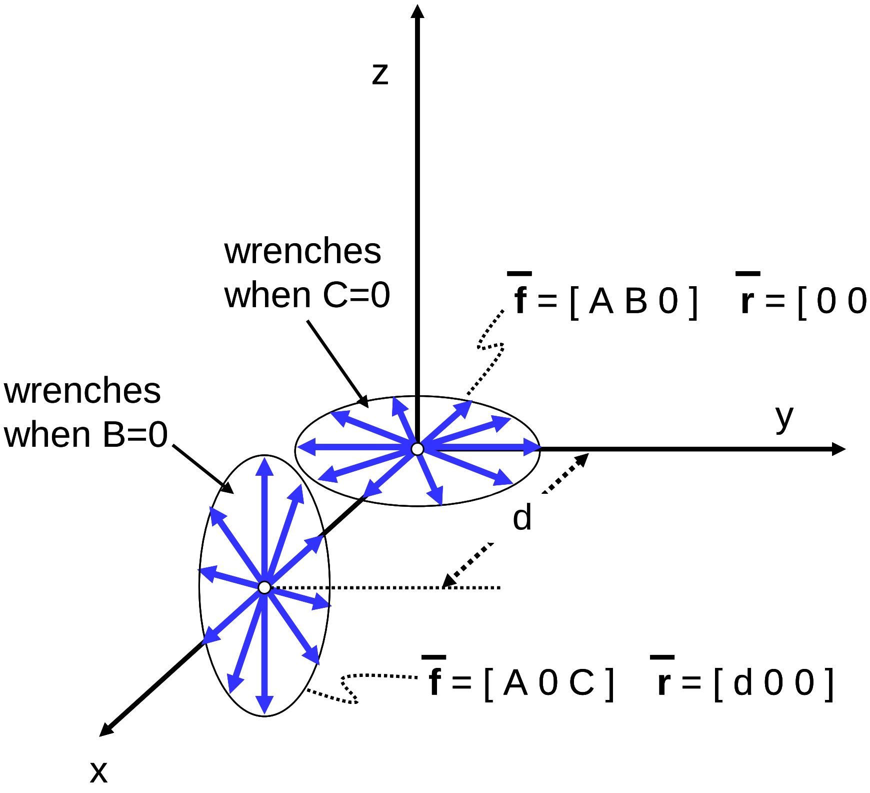

z
wrenches when C=0
$\bar{\mathbf{f}}$ = [A B 0]
$\bar{\mathbf{r}}$ = [0 0 0]
wrenches when B=0
y
d
$\bar{\mathbf{f}}$ = [A 0 C]
$\bar{\mathbf{r}}$ = [d 0 0]
x

Figure 5.12: Complete constraint space represented as wrenches (blue)

It has now been mathematically verified that these two disks of wrenches represent the only acceptable constraint lines within the system’s constraint space.

5.2.4 Problems with Finding Allowable Constraint Spaces

The fact that flexure system constraints are always modeled as having *q* values equal to zero creates difficulties in finding allowable constraint spaces when given desired degrees of freedom. This section discusses the consequences created by this fact.

Constraint spaces are simplified because only wrenches with *q* values equal to zero are of interest. Since freedom spaces are not limited to twists with zero pitch values, freedom spaces will generally include more twist lines that are more complicated to visually describe than constraint spaces. The condition that *q* only equals zero also allows one to visually express all constraint spaces using a single color, blue. Otherwise, two other colors would need to be implemented corresponding to *q* values that equal infinity and *q* values that equal a finite, non-zero real number. It is also significant to note that constraint spaces will never contain constraint hoops because it is not practical to attach a compliant constraint to a stage infinitely far away from the center of the stage. The stage would have to be infinitely large.

The condition that *q* only equals zero, however, often complicates matters. As was demonstrated previously, when a system's wrenches are determined, there is a need to filter out all the wrenches that don't satisfy this condition. This process may be tedious and in some instances mathematically impossible. The worst news is that this condition often limits the degrees of freedom with which a designer may wish to design a system. Instances exist in which some desired degrees of freedom may not be achieved without including other unwanted degrees of freedom.

The following is an example of this predicament. Suppose one wished to design a system that could only move with a disk of pure translations as shown in **Figure 5.13**. Every twist inside this disk may be expressed as the linear combination of two independent pure translational twists also shown in **Figure 5.13**.

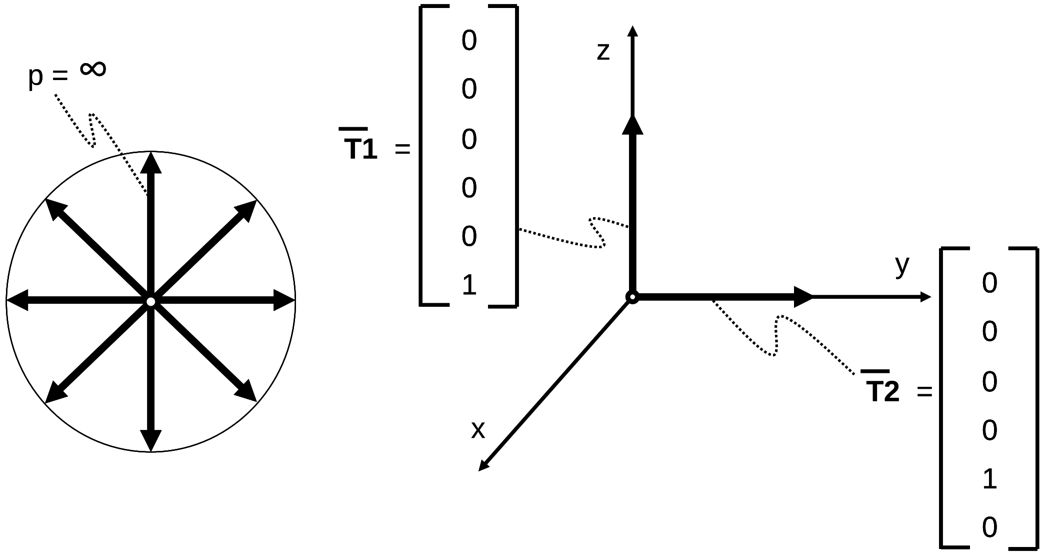

p = ∞
$\overline{T1} = \begin{bmatrix}0\\0\\0\\0\\0\\1\end{bmatrix}$
z
y
x
$\overline{T2} = \begin{bmatrix}0\\0\\0\\0\\1\\0\end{bmatrix}$

Figure 5.13: A disk of pure translations (thick black lines) may be mathematically described by two independent twists with $\vec{w} = 0$ and p=∞.

These two pure translational twists may be stacked into a 2×6 twist matrix where each row corresponds to one of the twists. The null space of this twist matrix is a linear combination of four independent vectors. To describe these vectors as conventional wrenches, their $\vec{\tau}$ and $\vec{f}$ vectors are switched so that they are expressed in the form shown in **Equation (3.11)**. The result is shown below:

$$A\begin{bmatrix}1\\0\\0\\0\\0\\0\end{bmatrix}+B\begin{bmatrix}0\\0\\0\\1\\0\\0\end{bmatrix}+C\begin{bmatrix}0\\0\\0\\0\\1\\0\end{bmatrix}+D\begin{bmatrix}0\\0\\0\\0\\0\\1\end{bmatrix}=\begin{bmatrix}A\\0\\0\\B\\C\\D\end{bmatrix}=\vec{W}, \qquad (5.8)$$

where *A*, *B*, *C* and *D* may be any real numbers. The 6×1 wrench vector at the far right of **Equation (5.8)** is the complete mathematical representation of every possible wrench that satisfies the freedom space shown in **Figure 5.13**. This resultant wrench's axial force and torque vectors, $\vec{f}$ and $\vec{\tau}$, are the following:

$$\vec{f} = \begin{bmatrix} A & 0 & 0 \end{bmatrix}$$
$$\vec{\tau} = \begin{bmatrix} B & C & D \end{bmatrix}. \tag{5.9}$$

The wrenches that have finite, non-zero q values must now be filtered out by plugging q=0 and **Equation (5.9)** into **Equation (5.4)**. The result is given by

$$AB = 0. \tag{5.10}$$

The only allowable wrenches are the ones that make **Equation (5.10)** a true statement. This equation will only be true if either A or B equals zero. If A, however, equals zero, **Equation (5.8)** suggests that the resultant wrench will be a pure torque wrench with a q value of ∞ and is, therefore, not acceptable. Therefore, B must equal zero if any wrenches are to satisfy the initial requirements. If, however, B must always equal zero, there are no longer four independent wrenches that complement the two desired degrees of freedom with q values equal to zero since only three constants A, C, and D remain in **Equation (5.8)**.

If one wants to design a flexure system that moves with a disk of pure translations, one must use only three non-redundant constraints whose linear combination results in a vector expressed as $\vec{W} = \begin{bmatrix} A & 0 & 0 & 0 & C & D \end{bmatrix}^T$. Using only three non-redundant constraints will, however, require that the system consist of three independent twists instead of the desired two independent pure translations according to Maxwell's observation in **Equation (2.1)**. This means that the designer will have to make due with an extra, undesired, rotational degree of freedom.

5.2.5 Proof of Maxwell’s Equation

Maxwell’s observation described in **Equation (2.1)** will now be proven. First note that wrench matrices will always have 6 columns corresponding to the 6 components inside a single wrench vector. The number of rows a wrench matrix has depends on the number of wrenches or constraints the system has. The number of independent wrenches or non-redundant constraints in the system may be determined by performing Gaussian elimination on the wrench matrix. The number of independent wrenches is the number of rows in the matrix that are not eliminated that contain pivots. In other words, the number of non-redundant constraints is the rank of the matrix. From linear algebra, it is common knowledge that the number of independent vectors that results from finding the null space of a matrix is the number of columns that matrix has subtracted from that matrix’s rank. Therefore, applying this statement to wrench matrices, the number of independent twists that complement any system of constraints is 6 subtracted from the number of non-redundant constraints inside that constraint topology. This is what **Equation (2.1)** says.

5.3 Unique and Finite Spaces

This section introduces the idea that freedom and constraint spaces are uniquely linked and finite in number. This idea enables FACT to function.

The first significant fact to note is that the freedom space of any system is unique to its constraint space and that the constraint space of any system, therefore, is also unique to its freedom space. This fact should seem obvious after mathematically generating constraint spaces from freedom spaces and after generating freedom spaces from constraint spaces. **Figure 5.14** and **Figure 5.15** demonstrate the uniqueness of the freedom and constraint spaces for the systems studied in this chapter. This principle that complementary spaces are uniquely linked with each other is called the Principle of Complementary Topologies.

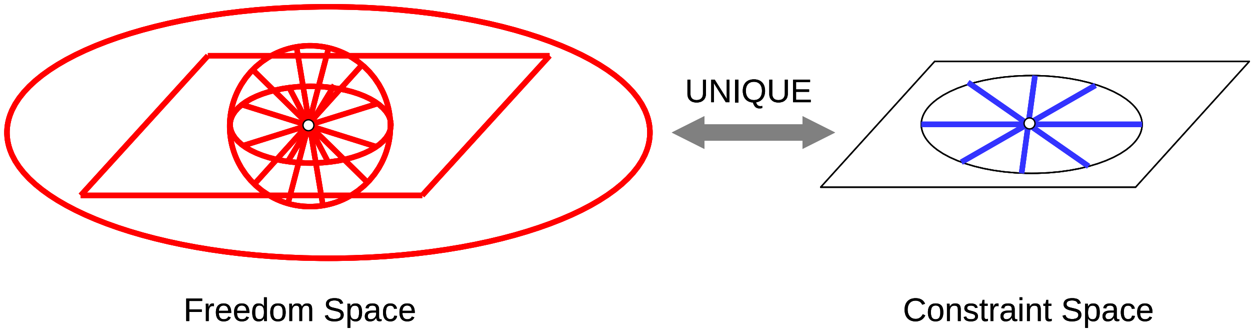


Figure 5.14: The freedom space of the system studied in Section 5.2.1 is unique to its constraint space

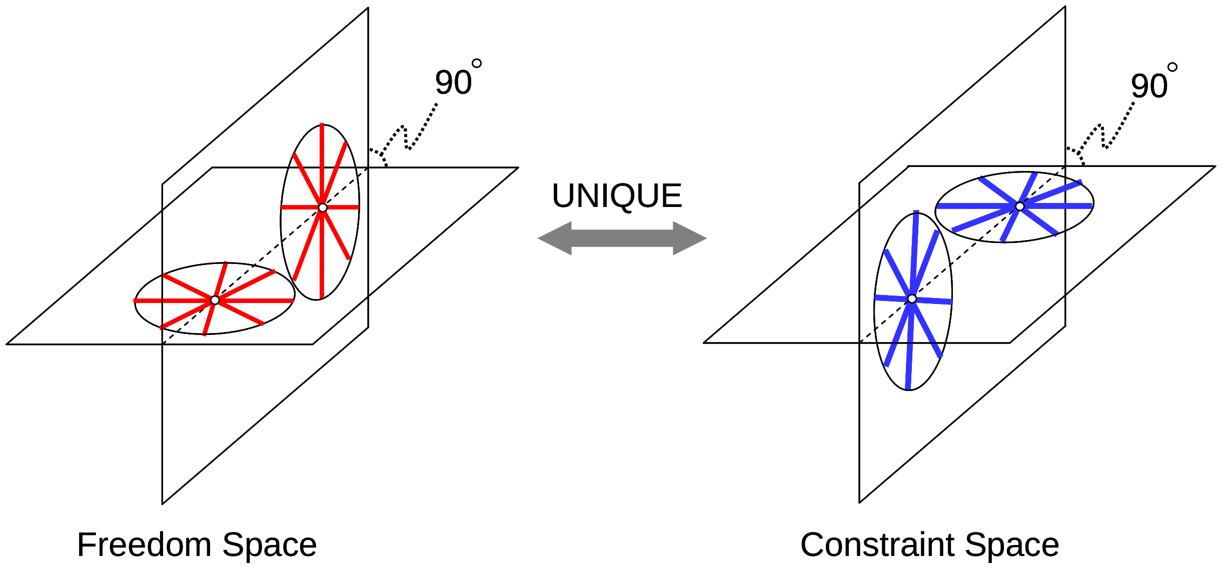


Figure 5.15: The freedom space of the system studied in Section 5.2.3 is unique to its constraint space

The next important fact worth noting is not obvious at all. Since an infinite number of possible constraint lines with an infinite number of possible orientations and locations in three-space exist, it would be tempting to hypothesize that an infinite number of possible freedom and constraint spaces exist as well. This is, however, not the case. There are actually a finite number of freedom and constraint spaces. More specifically, there are 26 freedom spaces each with a unique constraint space. Finding these 26 spaces is important because once all the spaces have been found, they may be used as powerful tools for designing or analyzing all possible flexure systems. One could say that once all 26 freedom and constraint spaces have been found, every possible flexure system has already been designed.

The idea that only a finite number of freedom and constraint spaces exists is suggested by the fact that no more than 6 constraints may be non-redundant. Any system with 6 non-redundant constraints will have an empty freedom space and will be immovable. Any other constraint added to such a system must be redundant and, therefore, cannot generate any new freedom spaces. This fact suggests that all the possible freedom spaces exist within systems that contain one through 6 non-redundant constraints which suggests a limit to the number of freedom spaces that exist.

The task then is to determine how many different ways non-redundant constraints may be organized within each of the 6 cases to find every freedom and constraint space. Each case is identified by the number of non-redundant constraints within the system. The second case, for example, contains systems with two non-redundant constraints. Only three different freedom and constraint spaces exist within the second case because there are only three ways of organizing two non-redundant constraints: parallel, intersecting, and skew. Once every different way of organizing the non-redundant constraints inside each of the 6 cases has been found, every freedom and constraint space may likewise be found. This will become clear in **Chapter 7** and **Chapter 8** where the 26 pairs of spaces will be identified and described.

CHAPTER 6:

"Ruled Surfaces"

This chapter introduces and mathematically describes three ruled surfaces that appear often in the 26 freedom and constraint spaces. A ruled surface is a surface that may be swept out by moving a line in space. In other words, it is a surface made of an infinite number of lines. Any given point that lies on such a surface is intersected by at least one line that also lies entirely on that surface. These surfaces appear often as sets in many freedom and constraint spaces since freedom and constraint sets are, by definition, spaces that contain twist and constraint lines [35].

6.1 Hyperbolic Paraboloid

This section describes hyperbolic paraboloids [36]. A hyperbolic paraboloid is a "saddle shaped" infinite three dimensional surface with hyperbolic and parabolic cross-sections. A typical hyperbolic paraboloid is shown in **Figure 6.1**. Note also that every line drawn on its surface is a straight line that lies entirely on the surface at all points along the line out to infinity. For this reason it is a ruled surface.

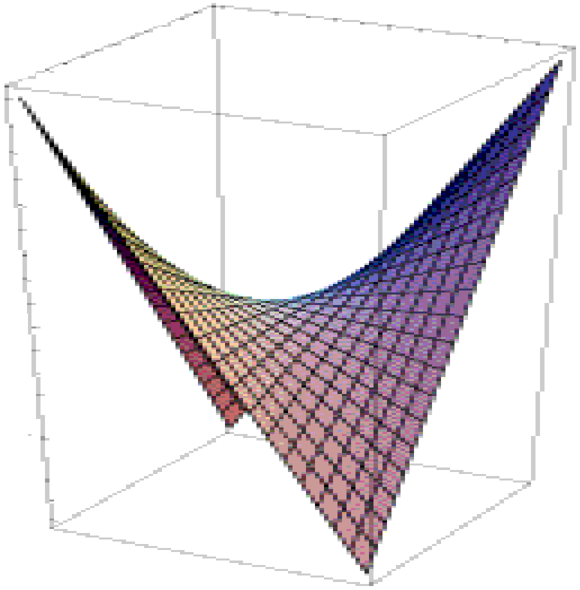


Figure 6.5: Hyperbolic Paraboloid [36]

Every hyperbolic paraboloid may be expressed as

$$z = -\frac{x^2}{a^2} + \frac{y^2}{b^2}\,. \tag{6.1}$$

where *a* and *b* may be any real numbers. This equation is true only when a coordinate system is properly assigned to the center of the hyperbolic paraboloid as shown in **Figure 6.2**.

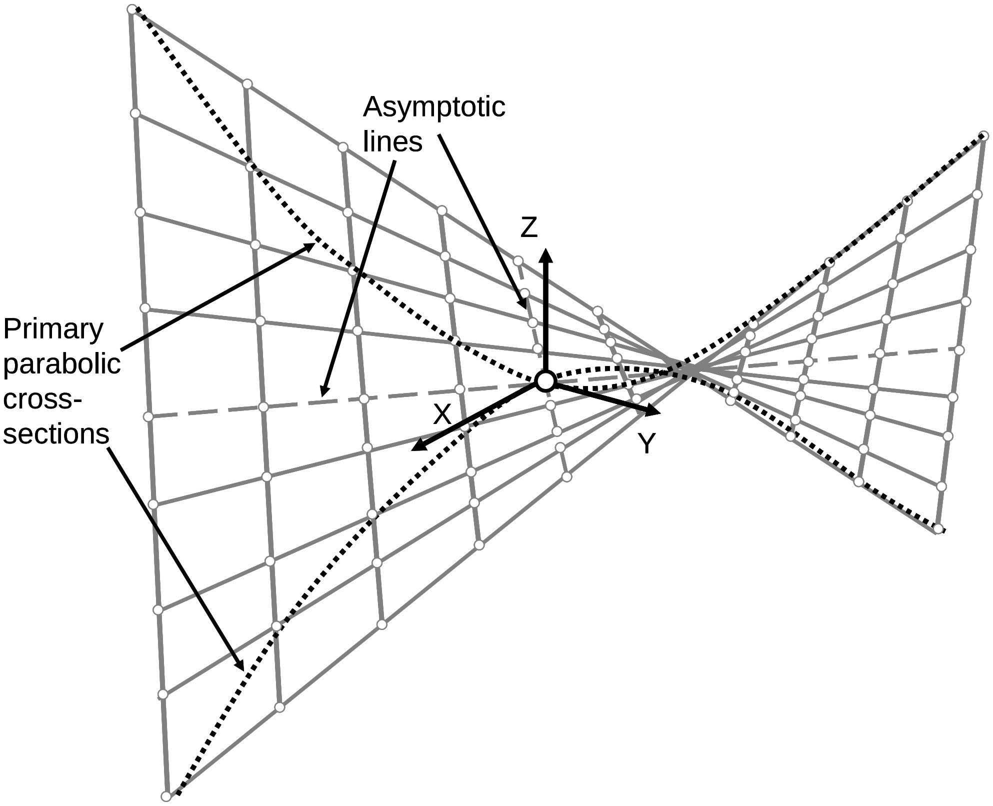


Figure 6.2: Hyperbolic paraboloid shown with coordinate system.

When *x*=0, **Equation (6.1)** becomes a parabola centered at the origin that lies on the z-y plane. Every point on that parabola has a positive z value. When *y*=0, **Equation (6.1)** becomes another parabola centered at the origin that lies on the z-x plane. Every point on that parabola has a negative z value. These primary parabolas are orthogonal and touch each other only at the origin. They are shown with dotted black lines in **Figure 6.2**. If either *x* or *y* are set to some constant value, and the other variable is allowed to vary, similar parabolic cross-sections are created at infinite other locations on planes parallel to either the z-y or z-x planes. Every parabola that lies on a plane parallel to the z-y plane will be a positive parabola that rises upward, and every parabola that lies on a plane parallel to the z-x plane will be a negative parabola that sinks downward.

Hyperbolic cross-sections are created when a hyperbolic paraboloid is sliced along planes that are parallel to the x-y plane. Each plane cut at different heights along the z-axis will create two opposing hyperbolas of equal size and shape. These hyperbolas are shown in orange and purple on the hyperbolic paraboloid in **Figure 6.3**.

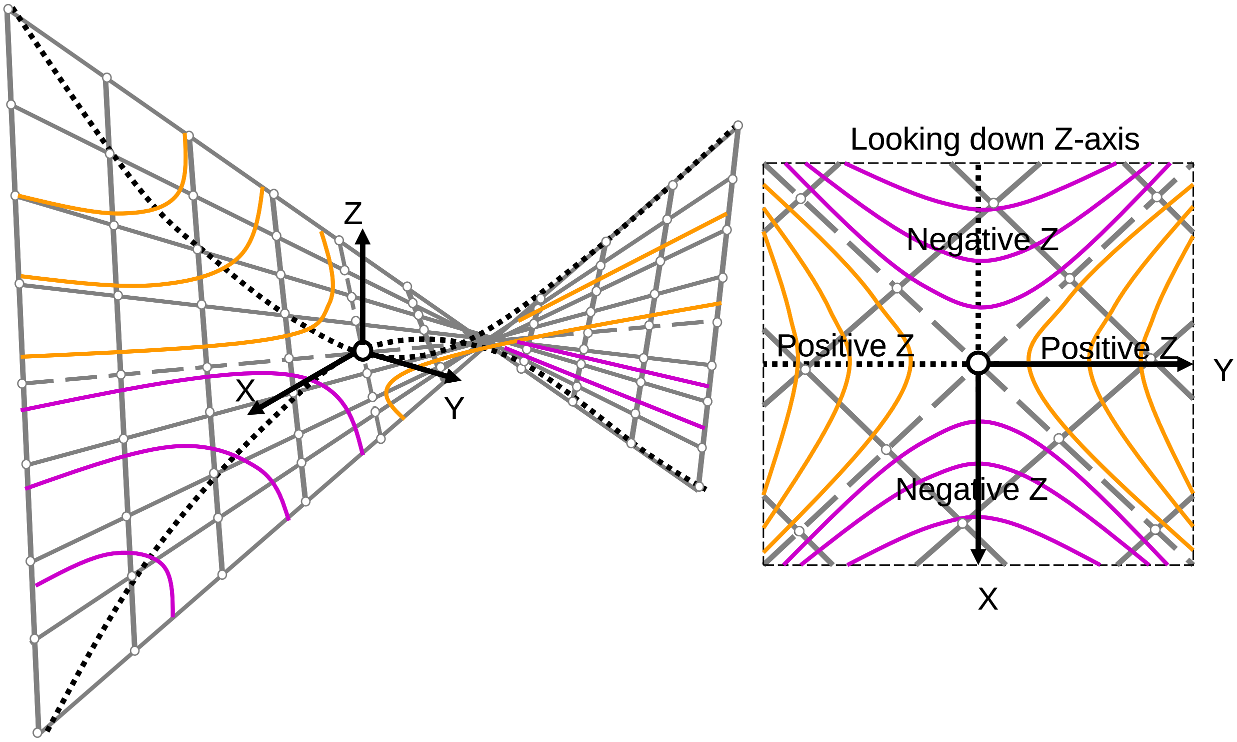


Figure 6.3: Purple hyperbolic cross-sections have negative z-values and orange hyperbolic cross-sections have positive z-values.

If the hyperbolas lie on planes parallel to and above the x-y plane at positive z values, the hyperbolas are colored orange. If the hyperbolas lie on planes parallel to and below the x-y plane at negative z values, the hyperbolas are colored purple. All of these hyperbolic cross-sections approach the same two asymptotic lines. These lines lie on the x-y plane and intersect at the origin. Their equation, therefore, is found by setting z=0 in **Equation (6.1)** and by solving for y in terms of x. This simplifies to the two equations

$$y = \pm \frac{b}{a} x \tag{6.2}$$

shown in **Figure 6.2** and **Figure 6.3** as dashed grey lines.

Note also that if *a*=*b* from **Equation (6.1)** such that both primary parabolas rise and sink at the same rate, these asymptotic lines will be orthogonal to each other and will be offset from the x- and y-axes by 45 degrees. If they are not equal to each other, the angle between the asymptotic lines varies as shown in **Figure 6.4**. This observation will be revisited in **Chapter 7**.

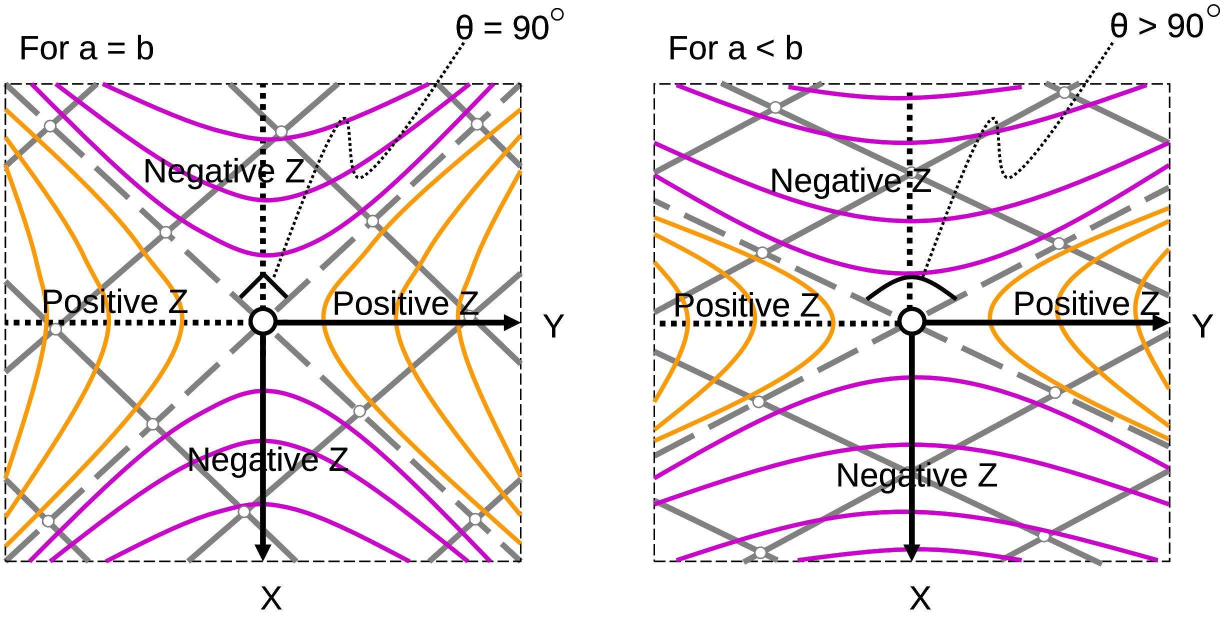


Figure 6.4: If *a*=*b*, the asymptotic lines (dashed grey) are orthogonal. Otherwise they are not.

Hyperbolic paraboloids are not only all ruled surfaces, they are also all doubly ruled surfaces. A doubly ruled surface is a surface on which two families or sets of lines lie. Any point on the surface of a doubly ruled surface will, therefore, be intersected by two and only two straight lines that both lie entirely on the surface. This may be seen in both hyperbolic paraboloids shown in **Figure 6.1** and **Figure 6.2**.

6.2 Hyperboloid

This section describes both circular and elliptical hyperboloids [37]. A circular hyperboloid is an infinite three-dimensional surface with hyperbolic and circular cross-sections. An elliptical hyperboloid is an infinite three-dimensional surface with hyperbolic and elliptical cross-sections. A typical hyperboloid is shown in **Figure 6.5**.

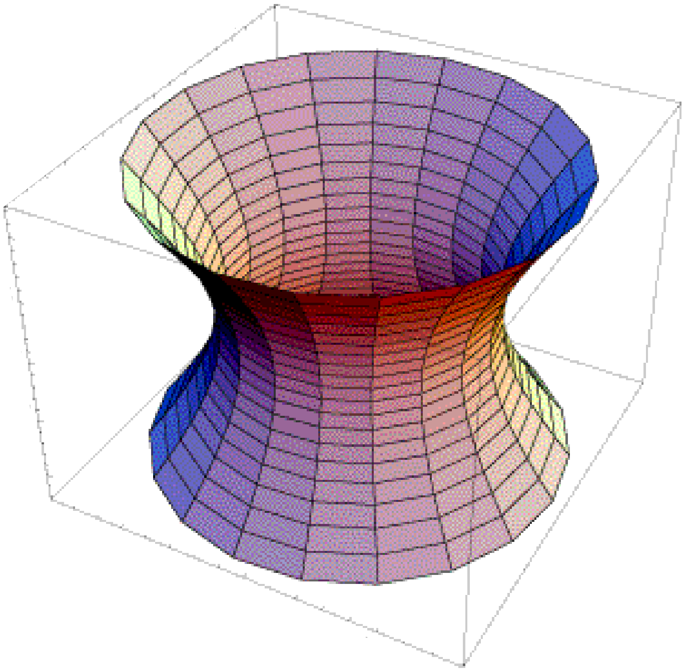


Figure 6.5: Hyperboloid [37]

Every circular hyperboloid may be expressed as

$$\frac{x^2+y^2}{L^2}-\frac{z^2}{c^2}=1. \tag{6.3}$$

where *L* is the radius of the circular cross-section on the x-y plane and *c* is a real number that determines the rate that the hyperboloid fans out as it travels away from the circle along the z-axis. This equation is true only when a coordinate system is properly assigned to the center of the circular hyperboloid as shown in **Figure 6.6**.

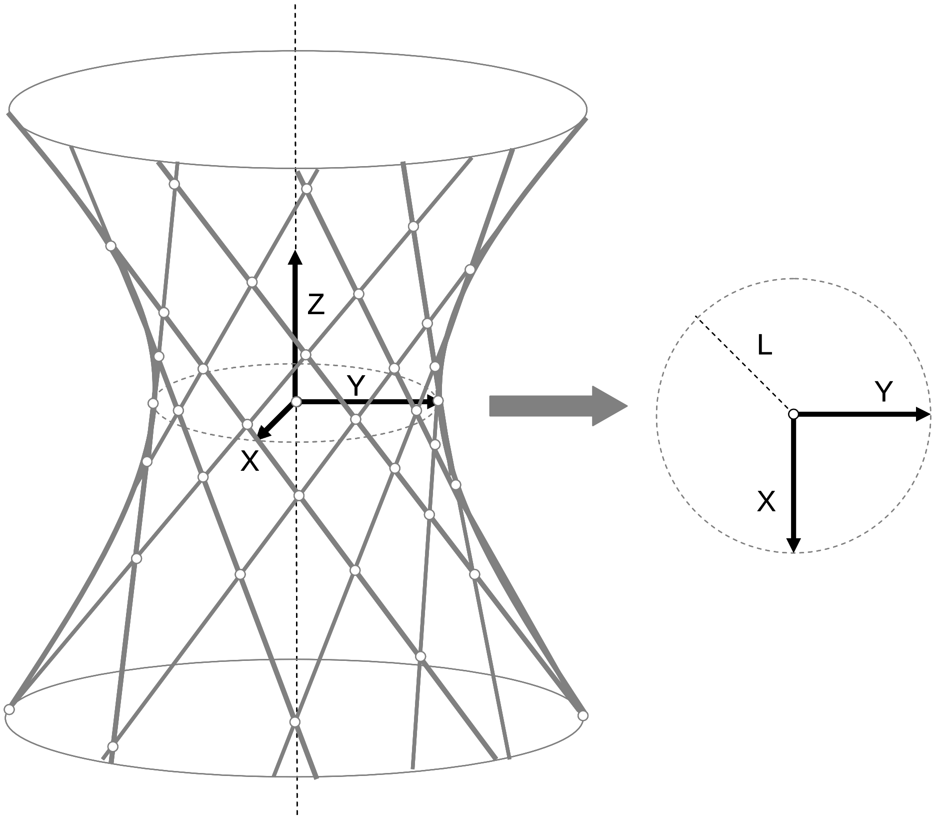


Figure 6.6: Circular hyperboloid with a coordinate system. Radius of the circular cross-section on the x-y plane is *L*.

Two opposing hyperbolic cross-sections are created by slicing a circular hyperboloid along a plane that intersects the z-axis. Since the hyperboloid is circular, every plane rotated about the z-axis will produce identical hyperbolic cross-sections. Circular cross-sections are created by slicing a circular hyperboloid along planes that are parallel or coincident with the x-y plane. The radii of the circles increase the farther the circles are away from the x-y plane. The *c* parameter from **Equation 6.3** determines the rate at which the radii of these circles increase as they are moved along the z-axis.

Note from **Figure 6.6** that circular hyperboloids are also doubly ruled surfaces. Every point on the hyperboloid's surface has exactly two lines that intersect at that point and lie entirely on the hyperboloid's surface.

Elliptical hyperboloids are very similar to circular hyperboloids and may be expressed as

$$\frac{x^2}{a^2}+\frac{y^2}{b^2}-\frac{z^2}{c^2}=1. \tag{6.4}$$

where *a* and *b* are the lengths of the major and minor axes of the elliptical cross-section on the x-y plane. The major and minor axes of this elliptical cross-section are oriented along the x- and y-axes. Similar to circular hyperboloids, *c* determines the rate that the elliptical hyperboloid fans out as it moves away from the x-y plane along the z-axis. An elliptical hyperboloid is shown in **Figure 6.7**.

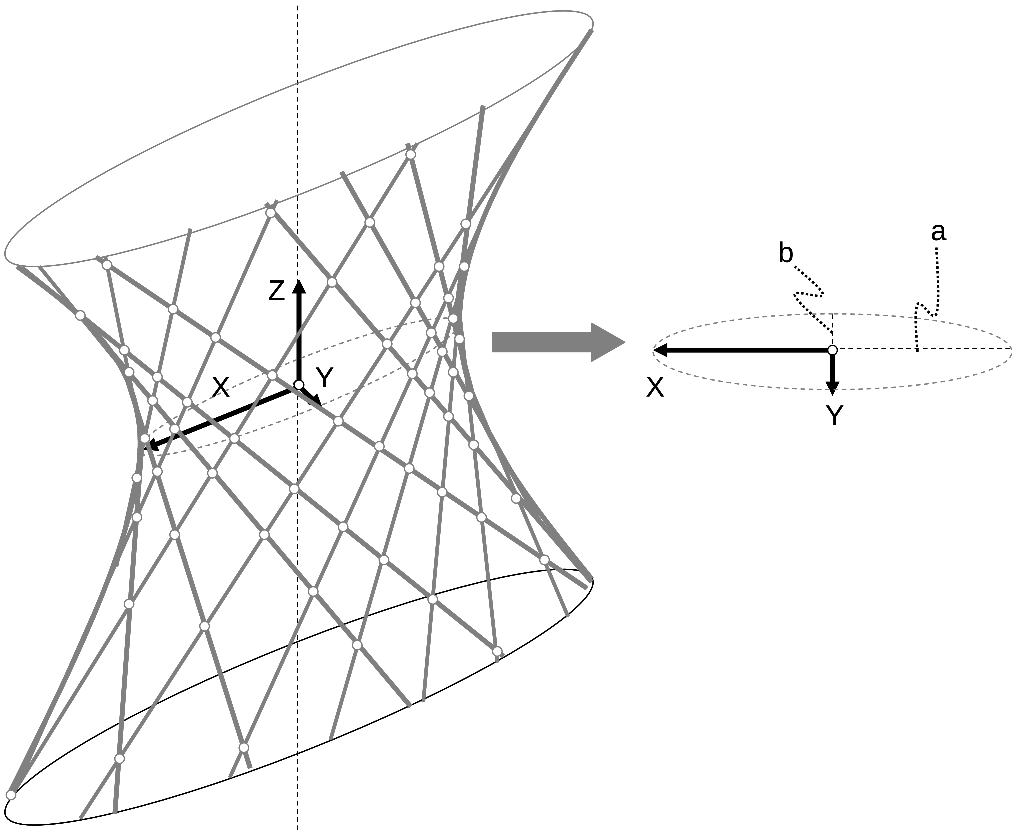


Figure 6.7: Elliptical hyperboloid with a coordinate system. The elliptical cross-section on the x-y plane has a major axis of *a* and a minor axis of *b*.

Two opposing hyperbolic cross-sections are created by slicing an elliptical hyperboloid along a plane that intersects the z-axis. Since the hyperboloid is elliptical, every plane rotated about the z-axis will produce different hyperbolic cross-sections. Elliptical cross-sections are created by slicing an elliptical hyperboloid along planes that are parallel or coincident with the x-y plane. The major and minor axes of the ellipses increase the farther away they are from the x-y plane.

Note also that an elliptical hyperboloid becomes a circular hyperboloid when $a=b$ in **Equation (6.4)**. The major and minor axes of the elliptical cross-section on the x-y plane will become the radius, L, of the circular cross-section on the x-y plane of the hyperboloid. Finally, note that an elliptical hyperboloid is also a doubly ruled surface.

6.3 Cylindroid

This section describes a cylindroid [38]. A cylindroid, also known as plücker's conoid, is an infinite three-dimensional ruled surface that is shown in **Figure 6.8**.

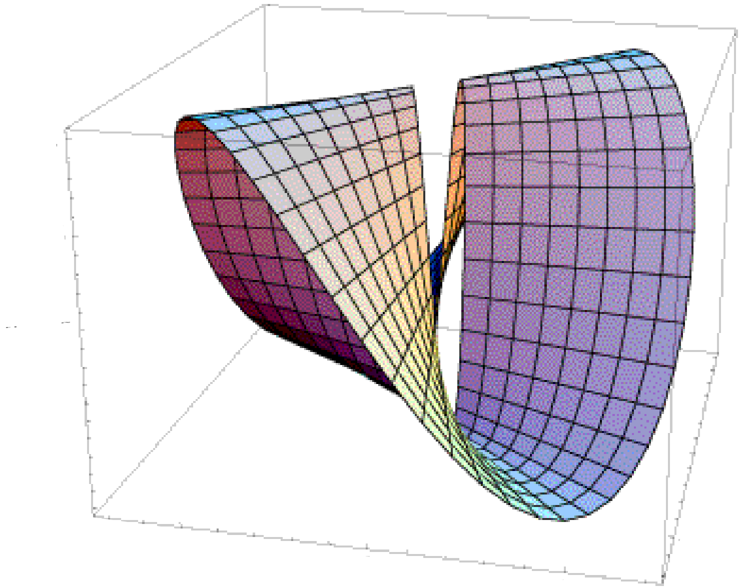


Figure 6.8: Cylindroid [38]

Every cylindroid may be expressed using polar coordinates as

$$
\begin{aligned}
x &= r\cos\theta \\
y &= r\sin\theta \\
z &= -h\cos\theta\sin\theta.
\end{aligned}
\tag{6.5}
$$

where *h* is the height of the cylindroid along the z-axis. These equations are true only when a coordinate system is properly assigned to the center of the cylindroid as shown in **Figure 6.9**.

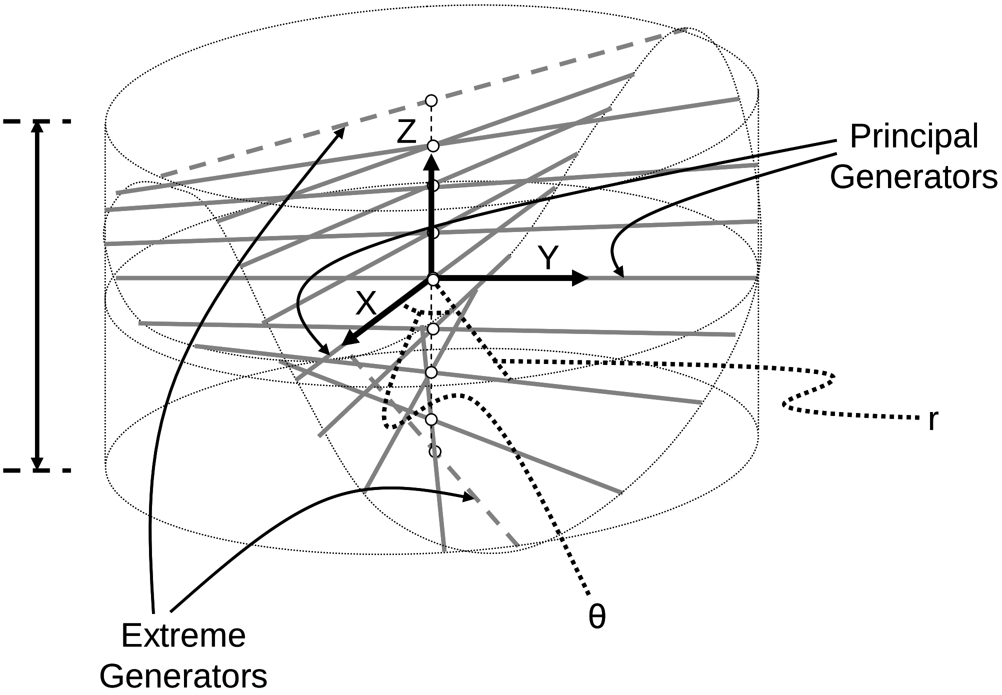


Figure 6.9: Cylindroid with a coordinate system labeled with significant parameters

Every cylindroid has two orthogonal principal generators that intersect each other at the origin. One of them always lies along the x-axis and the other always lies along the y-axis. Every cylindroid has two extreme generators shown with dashed grey lines in **Figure 6.9**. These extreme generators are always orthogonally skew with each other. The principal generators always lie on a plane that is parallel to and half way between the two planes that the extreme generators lie on. The skew angles of the principal generators relative to the extreme generators are always 45 degrees. Every point along the z-axis between the two extreme generators is the intersection point of two lines within the cylindroid that both lie on a plane that is parallel to the x-y plane. If a cylinder of radius, *r*, is placed along the z-axis as shown in **Figure 6.9**, the points of intersection on the cylinder's surface from every line within the cylindroid would look similar to a Pringle-Chip-like, sinusoidal contour around the surface of the cylinder (shown as a thin dotted line in **Figure 6.9**). The only parameter that really changes a cylindroid is the distance between the two extreme generators along the z-axis, *h*. The radius of the cylinder doesn't matter since the lines extend to infinity.

A cylindroid is not a doubly ruled surface. Only a single line will intersect any given point on its surface (unless the point lies along the z-axis between the two extreme generators). This line will also lie entirely on the surface of the cylindroid.

CHAPTER 7:

“Cases 1, 2, and 3”

This chapter describes and validates every constraint space with its unique freedom space within the first three cases. The reader may recall from the final section of **Chapter 5** that 6 total cases exist. The case of a system corresponds to the number of non-redundant constraints in that system. The number of types a case has is the number of freedom and constraint space pairs within that case, or the number of “different” ways the non-redundant constraints may be arrange within the system to produce fundamentally different freedom spaces.

7.1 Case 1:

This section describes the first case of 6. The first case consists of all systems that contain only one non-redundant constraint. To find the number of types within this case, one must determine how many different ways a single non-redundant constraint may be arranged such that different freedom spaces are created. This may seem obvious for the case of a single non-redundant constraint since any single constraint line in three-space will produce the exact same freedom space as any other single constraint line oriented anywhere else in three-space. The orientations of the freedom spaces may be different for differently oriented single constraint lines, but the fundamental sets they produce will all be the same shapes and spaces with respect to each other and are, therefore, the same freedom spaces. It follows then that this case has only one type or pair of freedom and constraint spaces.

7.1.1 Case 1, Type 1:

The constraint space of this type is very simple. It consists of a single constraint line as shown in **Figure 7.1**.

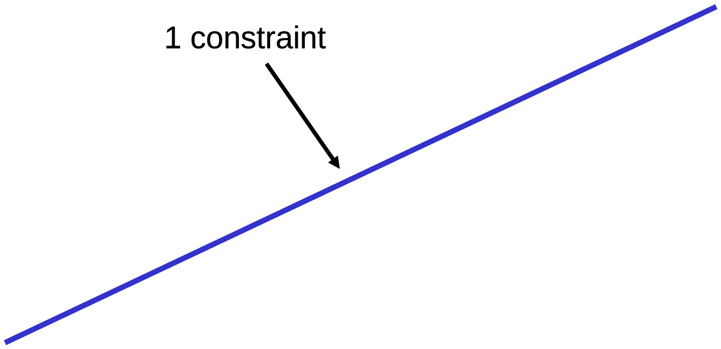


Figure 7.6: Constraint space of Case 1, Type 1

The only way a redundant constraint could be added to this system without changing its freedom space would be to add it somewhere along the same line as shown in **Figure 7.2**.

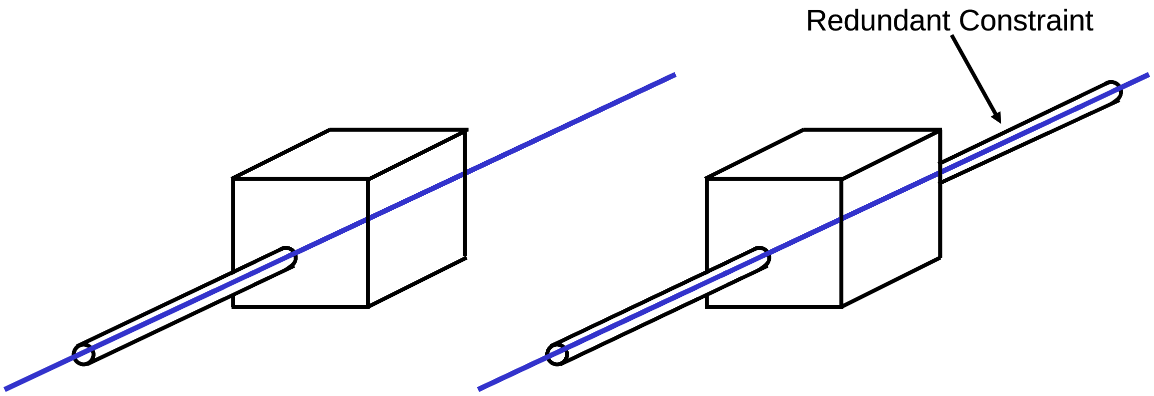


Figure 7.2: Once a single non-redundant constraint has been selected from the constraint space consisting of a single constraint line, all other constraints selected from that space will be redundant.

The freedom space of this type is, however, complex. Since this type belongs to the first case, one would expect its freedom space to be a linear combination of 5 independent twists from **Equation (2.1)**. First every pure rotational freedom line will be found that satisfies Blanding's Rule of Complementary Patterns for the single constraint line. These rotational freedom lines form freedom sets that are shown in the following three figures.

Figure 7.3 depicts an infinite number of spherical freedom sets whose center points all lie on the constraint line. In fact, every point along the constraint line is the center point of a single spherical freedom set that contains every line that intersects that point in three-space.

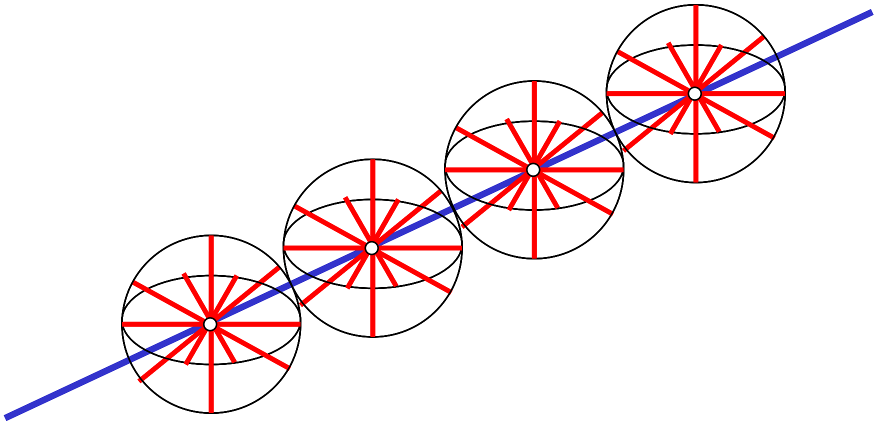


Figure 7.3: Spherical pure rotational freedom sets (red) that complement the single constraint line (blue)

Figure 7.4 shows an infinitely large box that represents a freedom set that contains every line that is parallel to the constraint line.

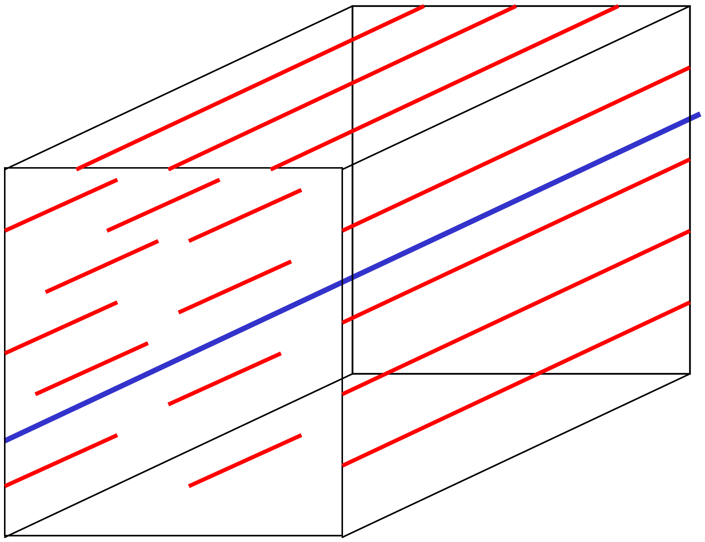


Figure 7.4: Box freedom set of parallel lines (red) that complements the single constraint line (blue)

The pure rotational hoops shown in **Figure 7.5** all intersect the constraint line at a point at infinity and represent all pure translations orthogonal to the constraint line.

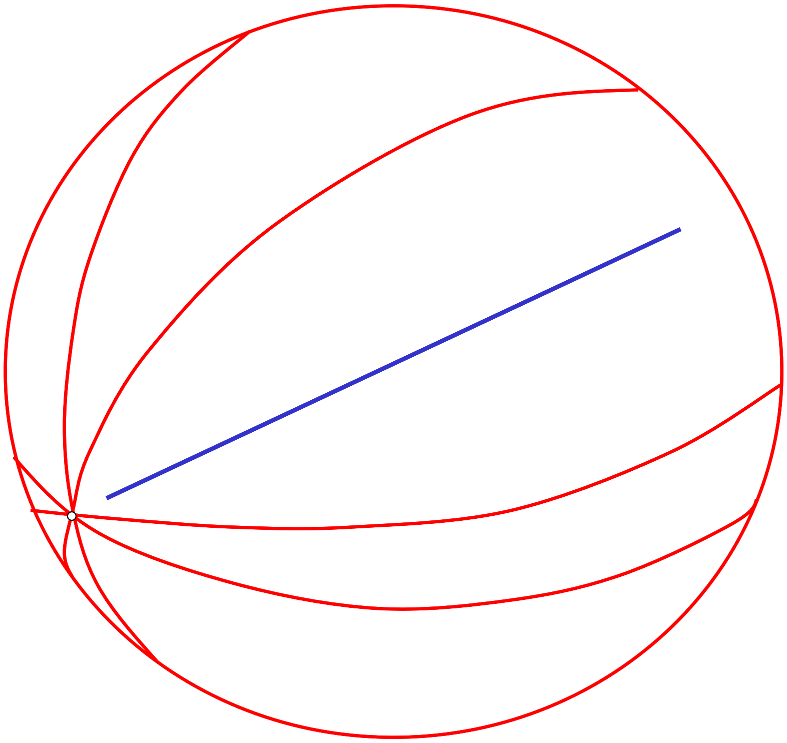


Figure 7.5: Pure rotational hoops (red) that complement the single constraint line (blue)

These pure rotational freedom sets do not, however, represent every possible twist for a single non-redundant constraint system. Screws with non-zero finite pitch values also exist. These screws may be represented as green lines that are tangent to the surface of a cylinder with a radius of *d* as shown in **Figure 7.6**. The screws' pitch values may be determined using **Equation (3.13)** where θ is the skew angle between the screw line and the constraint line.

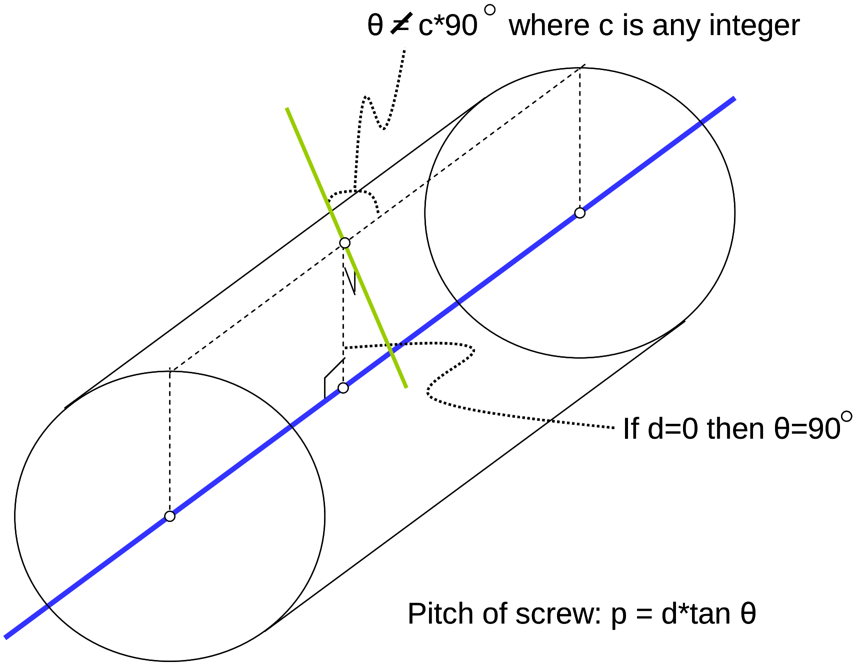


Figure 7.6: Screw lines with finite non-zero pitch values (green) that complement the single constraint line (blue)

These screws combine with the pure rotations and translations shown above to form the complete freedom space of this type. This freedom space is shown in **Figure 7.7**. The thick dashed black line corresponds to the line along which the constraint line lies.

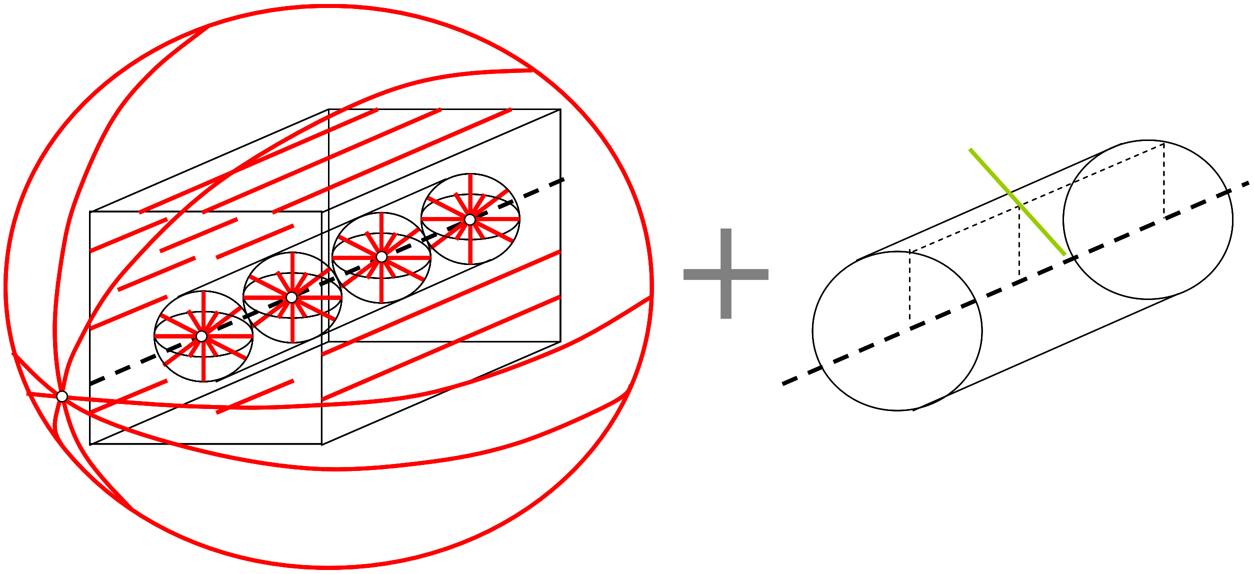


Figure 7.7: Freedom space of Case 1, Type 1

It is also interesting to note that the entire freedom space could actually mathematically be visualized using **Figure 7.6** alone without the restrictions placed on the parameters *d* and *θ* shown in the figure. The red lines inside each sphere correspond to every possible twist line where *d*=0. The red parallel lines in the box correspond to every possible twist line where *θ*=0 and *d* is any finite value. The red hoops correspond to every twist line where θ=0 and *d* is infinite or they correspond to every twist line where *θ*=90 degrees and *d* is finite (pure translations).

7.2 Case 2:

This section describes the second case of 6. The second case consists of all systems that contain two non-redundant constraints. To determine the number of types within this case, the question must be asked, "How many different ways may two non-redundant constraints be arranged such that different freedom spaces are created?" One reasons that two lines may be arranged in only three fundamentally different ways: intersecting, parallel and skew. It follows then that this case has three types or three pairs of freedom and constraint spaces.

7.2.1 Case 2, Type 1:

This section reviews the case of two non-redundant constraints that intersect at a point in finite space. This case and type have actually already been considered in the example system studied in **Section 5.1** and **Section 5.2.1** of **Chapter 5**. It was determined that the constraint space of such a system is a disk of constraint lines as shown again here in **Figure 7.8**. Once any two constraints have been selected from the disk, any other constraint selected from the same disk will be redundant.

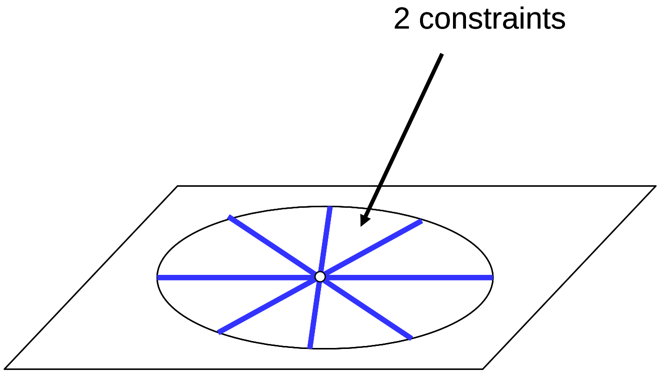


Figure 7.8: Constraint space of Case 2, Type 1

In **Chapter 5** Blanding's Rule of Complementary Patterns was used to find the pure rotational freedom sets within the freedom space of this case and type shown again here in **Figure 7.9**. For a thorough description of these sets refer to **Chapter 5**.

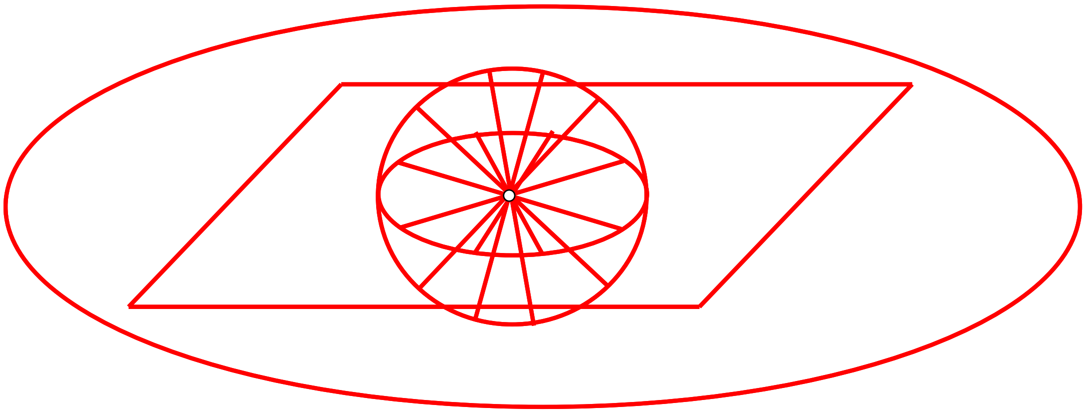


Figure 7.9: Pure rotational freedom sets of Case 2, Type 1.

These sets do not, however, represent every allowable twist within the freedom space of this system. Screws with finite non-zero pitch values also exist. To find these screws the visual approach discussed in **Chapter 3** in **Section 3.4.1** will be used.

Consider a twist line oriented in any direction at any location on any plane parallel to the plane of the disk of constraints as shown in **Figure 7.10**. This line will always be parallel to one of the constraint lines in the disk (shown as a blue dashed line in the figure). Since this twist line is parallel to one of the constraint lines, one would expect the twist line to be a pure rotation with zero pitch. But since it is not parallel to and does not intersect the other constraint lines in the disk, the twist line can't have a zero pitch and still satisfy **Equation (3.13)**. This line cannot, therefore, be an allowable twist line.

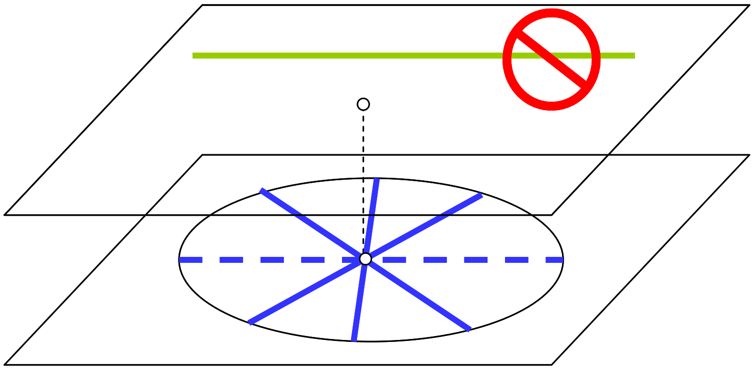


Figure 7.10: A twist line (green) on a plane parallel to the disk of constraint lines (blue) is not an allowable twist.

Now consider a twist line on a plane that intersects the plane of the disk of constraint lines at an angle, θ, that is not 90 degrees. The line on this plane also does not pass through the center of the disk as shown in **Figure 7.11**. As long as the twist line on this plane is not parallel to the disk of constraint lines, it will intersect one of the constraint lines (shown as a dashed blue line in the figure). If θ is not 90 degrees between the two planes, the twist line will never be perpendicular to the constraint line it intersects. One, therefore, expects this line to be a pure rotational freedom line based on Blanding's Rule of Complementary Patterns. But when one considers the other constraint lines in the disk that this twist line's pitch value must simultaneously complement using **Equation (3.13)**, one finds that such a line is not an allowable twist line.

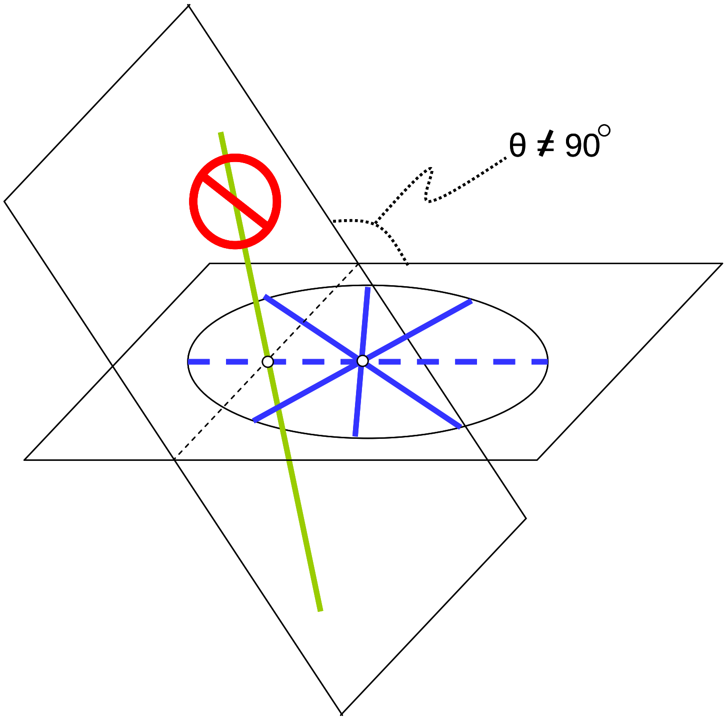


Figure 7.11: A twist line (green) on a plane that intersects the plane of the disk of constraint lines (blue) at an angle that is not 90 degrees is not an allowable twist.

The only other line that could be considered would be a line on a plane that intersects the plane of the disk of constraint lines at an angle of 90 degrees as shown in **Figure 7.12**. Since the angle between these planes is 90 degrees, the line on the plane will be perpendicular to the constraint line it intersects. Recall that a twist line that intersects a constraint line at a 90 degree angle could have any pitch value. The question is, therefore, could such a line have a single pitch value and simultaneously satisfy **Equation (3.13)** for every constraint line in the disk? Surprisingly the answer is yes.

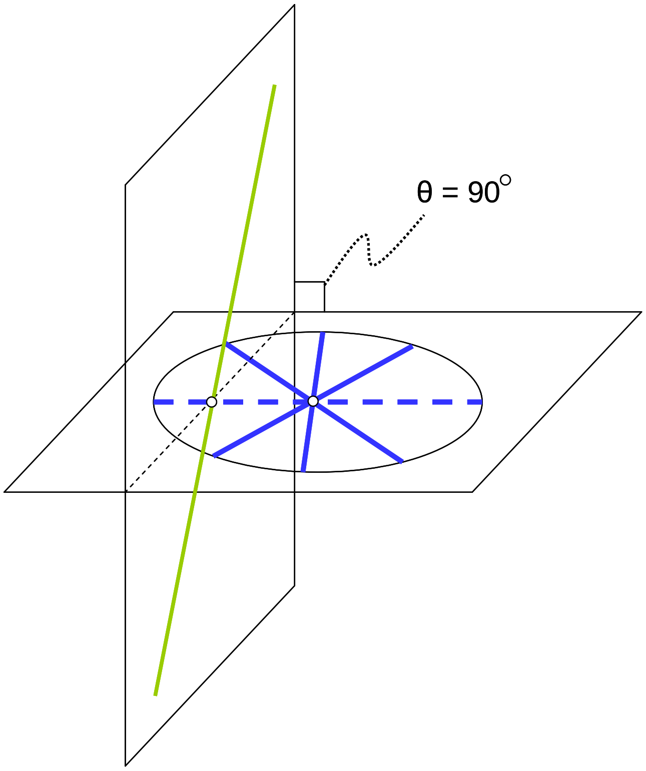


Figure 7.12: A twist line on a plane that intersects the plane of the disk of constraint lines at an angle of 90 degrees is an allowable twist.

In order to prove that this type of line is an allowable twist line, a coordinate system must be established and parameters that describe the disk and the twist line must be defined as shown in **Figure 7.13**.

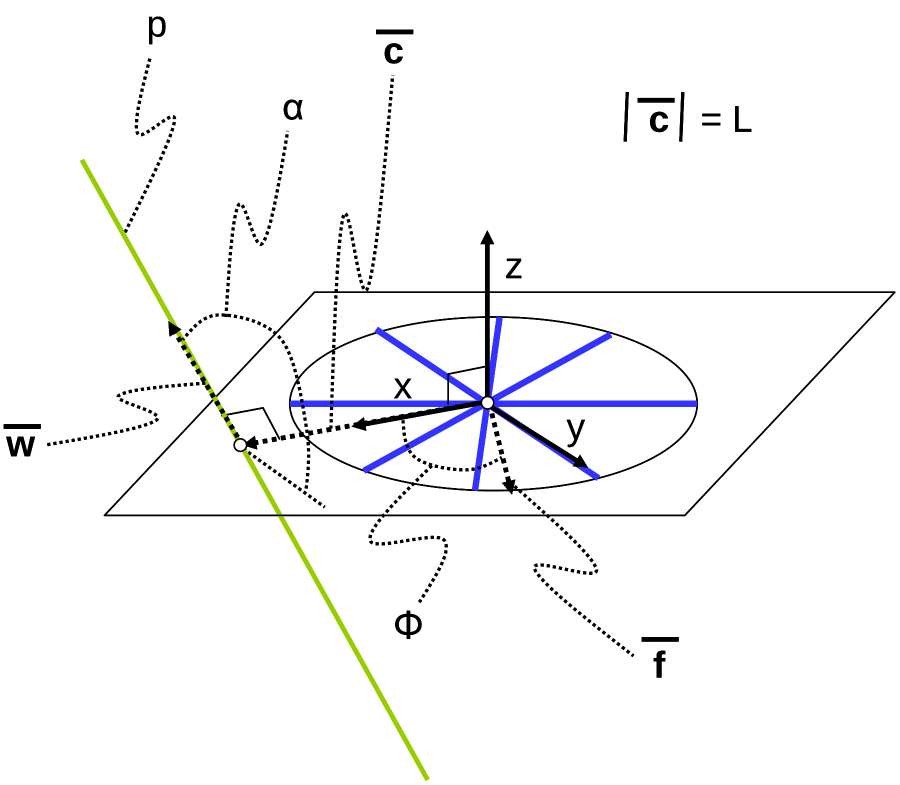


Figure 7.13: Parameters and coordinate system established for the twist line and disk from Figure 7.12.

Every location vector for the constraint lines in the disk may be expressed as $\vec{r} = [0 \quad 0 \quad 0]^T$. Every orientation vector for the constraint lines in the disk may be expressed as $\vec{f} = [\cos\phi \quad \sin\phi \quad 0]^T$ where *Φ* may be any real value. Every value of *Φ* will correspond to one of these infinite constraint lines. The location vector for the green twist line is $\vec{c} = [L \quad 0 \quad 0]^T$ where *L* is the shortest distance from that line to the center of the disk. The orientation vector for the green twist line is $\vec{w} = [0 \quad \cos\alpha \quad \sin\alpha]^T$ where *α* is the angle from the plane of the disk to the twist line as shown in **Figure 7.13**. One can find the pitch, *p*, of this twist line by plugging these vectors into the pitch equation given in **Appendix B** as **Equation (B.4)**. Once these substitutions have been performed, the parameter *Φ* drops entirely out of the equation since the disk is symmetric about the z-axis and since every constraint line complements the twist line with a single pitch value. The pitch of a twist line that complements a disk of constraint lines, therefore, elegantly and surprisingly simplifies to

$$p = L \tan \alpha \,. \qquad (7.1)$$

Equation (7.1) has some obvious parallels to **Equation (3.13)**. Recall that **Equation (3.13)** is the classic pitch equation that relates a twist to a single constraint line. **Equation (7.1)** is a new pitch equation that relates a twist to a disk of infinite constraint lines. The shortest distances *L* and *d* are analogous as well as the angles *α* and *θ*.

Every possible twist with a pitch value given by **Equation (7.1)** has, therefore, been found for Case 2, Type 1 and is represented in **Figure 7.13**. The twists that correspond to *L* values of zero will be the pure rotations with zero pitch values represented by the red sphere shown in **Figure 7.9**. The twists that correspond to *α* angles of zero or 180 degrees will be pure rotations with zero pitch values represented by the red plane also shown in **Figure 7.9**. The twists that correspond to *α* angles of 90 degrees will be pure translations with infinite pitch values that point in directions normal to the plane of the disk of constraints and are represented by the pure rotational hoop shown in **Figure 7.9**. The complete freedom space, therefore, of Case 2, Type 1 including all the screws with finite, non-zero pitch values is shown in **Figure 7.14**.

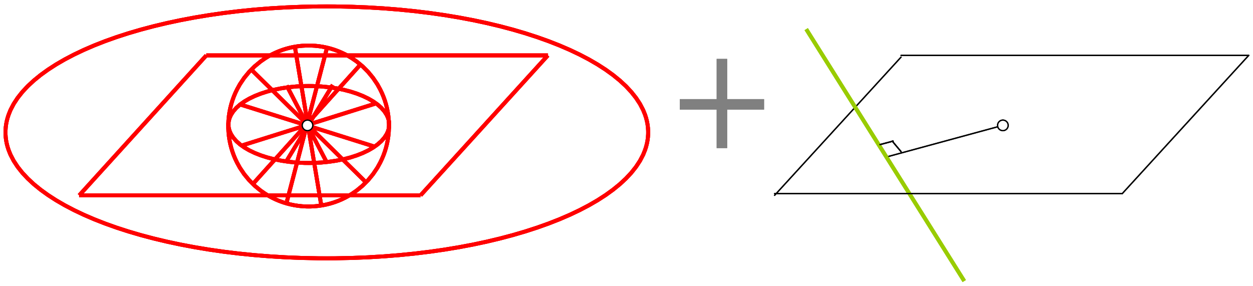


Figure 7.14: Freedom space of Case 2, Type 1.

Figure 7.15 shows how the freedom and constraint spaces of Case 2, Type 1 fit together. The screws with finite, non-zero pitch values are not shown to avoid cluttering the figure.

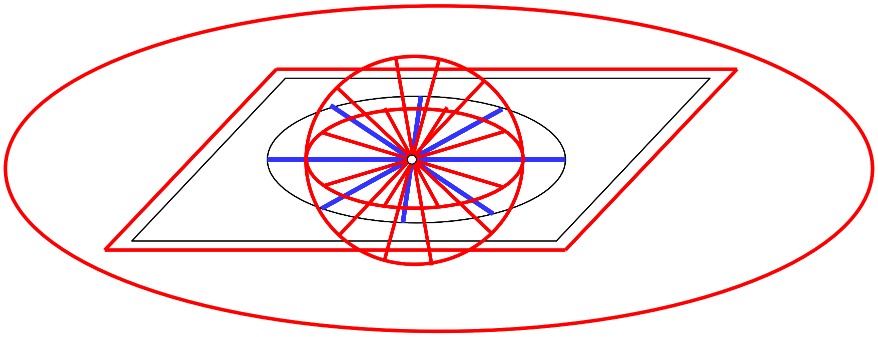


Figure 7.15: Freedom space (red) and constraint space (blue) of Case 2, Type 1 together (without the screws shown).

7.2.2 Case 2, Type 2:

This section reviews the case of two non-redundant constraints that are parallel. This case and type have actually already been considered in the example system studied in **Section 3.4** of **Chapter 3**. Using the visual and mathematical approach, the conclusion was drawn that the system's complete freedom space consists of all twist lines that lie on planes parallel to or coincident with the plane of the two parallel constraints as well as twist lines that are pure translations that lie on planes that are orthogonal to both constraints. With this fact in mind, Blanding's Rule of Complementary Patterns can be used to find the freedom sets that contain pure rotations only.

The first pure rotational freedom set is the plane coincident with the plane of the two parallel constraints shown in **Figure 7.16**. Any line on that plane will be a pure rotational freedom line.

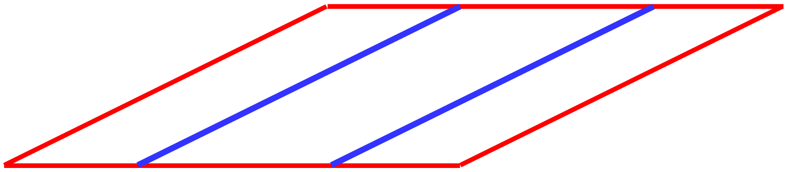


Figure 7.16: Planar freedom set (red) containing all pure rotational freedom lines that lie on the plane of the two parallel constraint lines (blue)

The second pure rotational freedom set is an infinitely large box that contains all the freedom lines that are parallel to the parallel constraint lines as shown in **Figure 7.17**. Every one of these lines will lie on a plane that is parallel to or coincident with the plane of parallel constraints.

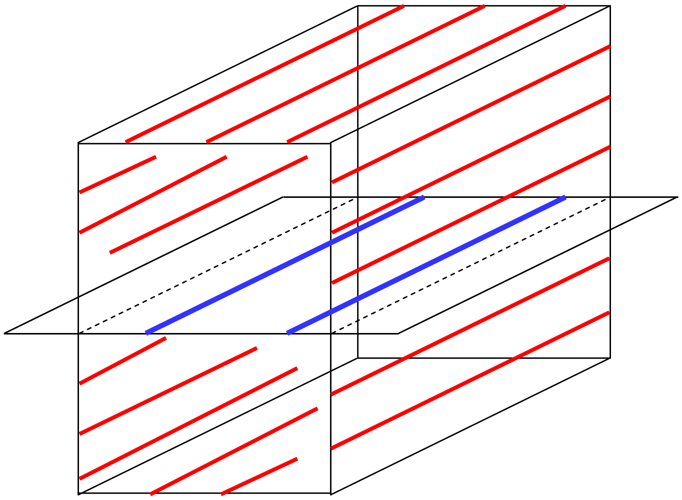


Figure 7.17: Box freedom set containing all the pure rotational freedom lines (red) that are parallel to the two non-redundant constraint lines (blue)

The pure translations that lie on planes that are orthogonal to the two parallel constraint lines can be expressed as pure rotational hoops that intersect both parallel constraint lines at a point at infinity as shown in **Figure 7.18**.

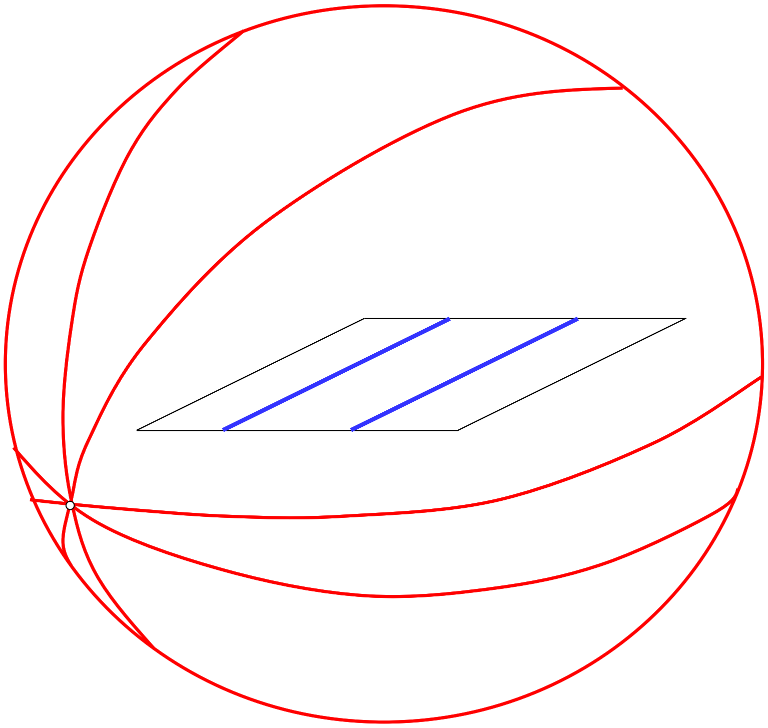


Figure 7.18: Pure rotational hoops (red) that intersect the parallel constraint lines (blue) at a point at infinity.

The screws of the system with finite, non-zero pitch values will exist on planes that are parallel to the plane of the parallel constraints and will be represented as green lines with pitch values determine by **Equation (3.13)**. They are shown in **Figure 7.19**.

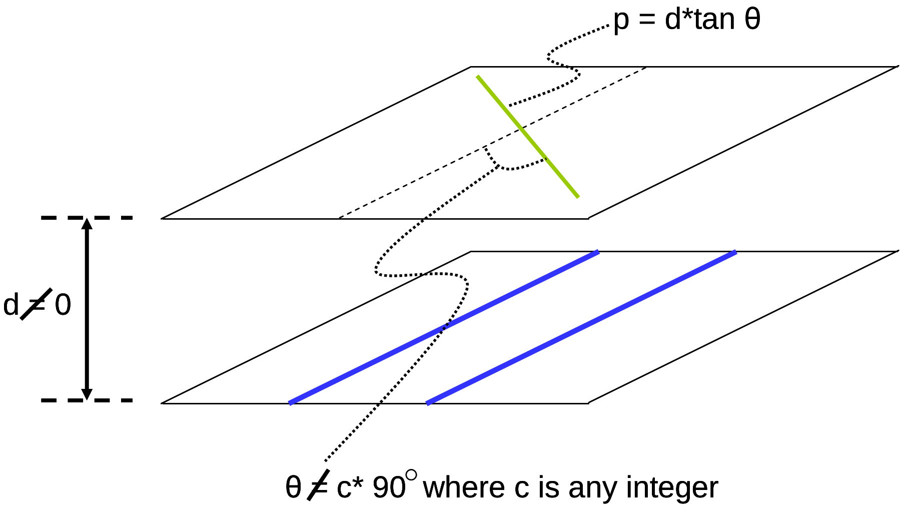


Figure 7.19: Screws (green) with finite, non-zero pitch values that complement parallel constraint lines (blue)

Note that if d=0, the twist line shown in **Figure 7.19** will represent every pure rotational freedom line with zero pitch on the plane of the parallel constraint lines shown in **Figure 7.16**. If θ=0 or 180 degrees for all values of d, the twist line will represent every pure rotational freedom line with zero pitch in the box freedom set shown in **Figure 7.17**. If θ=90 or 270 degrees for any d, the twist line will represent a pure translation with an infinite pitch shown as pure rotational hoops in **Figure 7.18**.

The complete freedom space for Case 2, Type 2 is shown in **Figure 7.20**. This figure depicts every possible pure rotation, pure translation and screw with a finite, nonzero pitch value for the system of two parallel constraints.

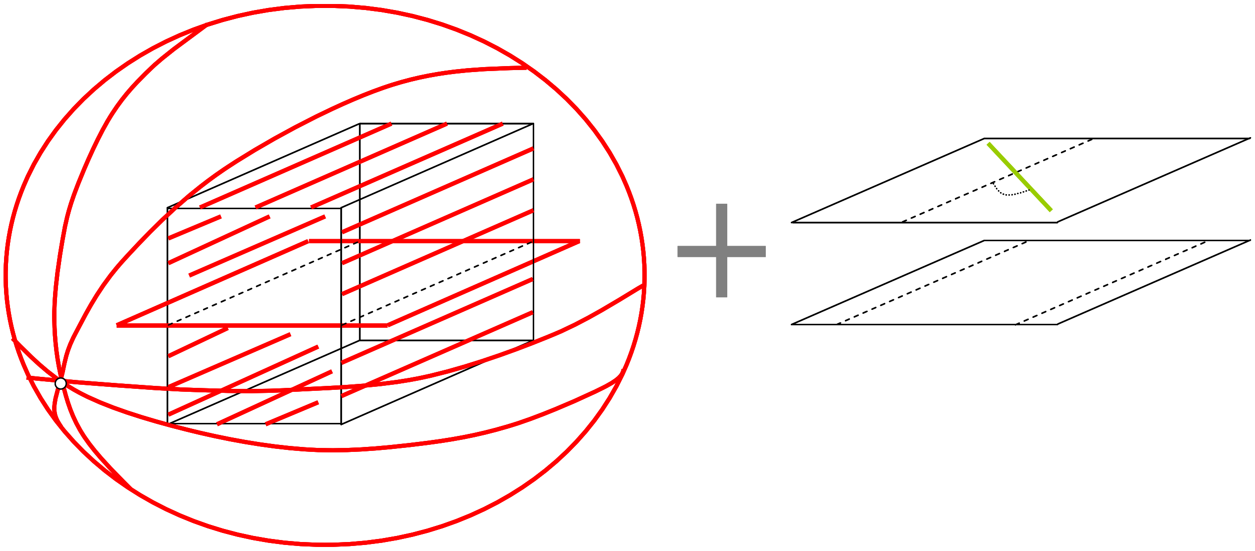


Figure 7.20: Freedom space of Case 2, Type 2

The constraint space of this system is not hard to find. The linear combination of any two parallel constraint lines produces a plane containing infinite parallel constraint lines with q values equal to zero as shown in **Figure 7.21**. Note also that once any two constraints from this space have been selected, any other parallel constraint selected from this plane will have no effect on the system's freedom space and will, therefore, be redundant. Any constraint selected from any other space will change the freedom space completely. **Figure 7.21** is, therefore, the system's complete constraint space.

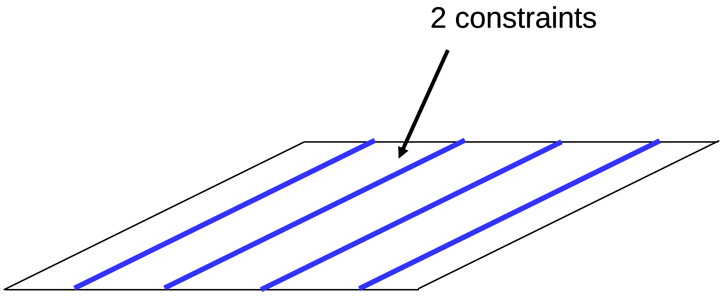


Figure 7.21: Constraint space of Case 2, Type 2

Figure 7.22 shows how the freedom and constraint spaces of Case 2, Type 2 fit together. The screws with finite, non-zero pitch values are not shown to avoid cluttering the figure.

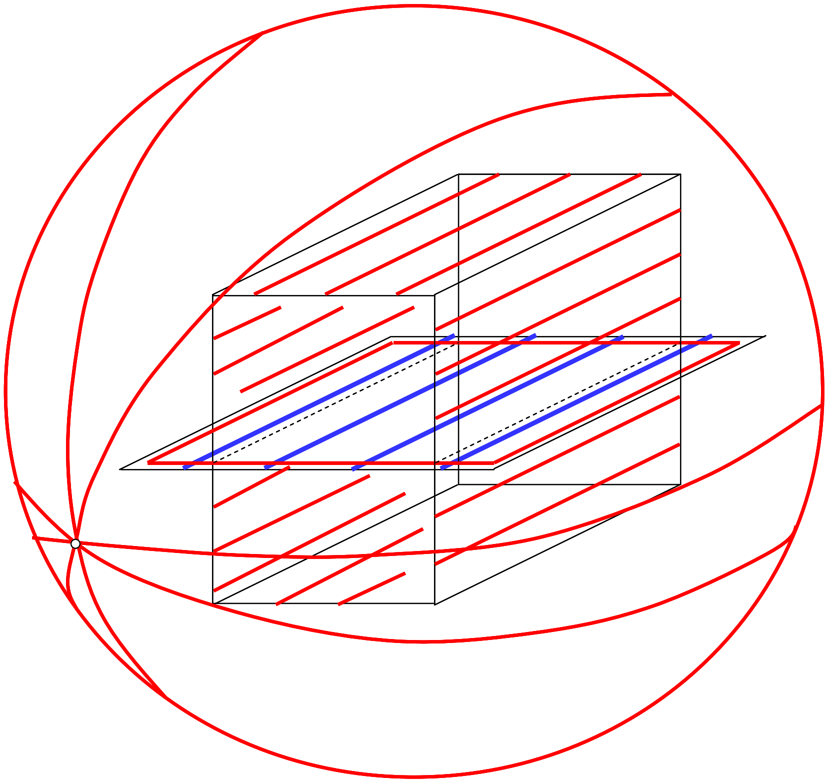


Figure 7.22: Freedom space (red) and constraint space (blue) of Case 2, Type 2 together (without the screws shown).

7.2.3 Case 2, Type 3:

This section examines the case of two non-redundant constraint lines that are skew. In order to find the freedom space of this system, Blanding's Rule of Complementary Patterns will be applied to locate all the pure rotational freedom sets.

The first two pure rotational freedom sets that will be considered are parallel planes that contain an infinite number of parallel pure rotational freedom lines as shown in **Figure 7.23**. Each plane contains one of the skew constraint lines. The freedom lines on each plane will be parallel to the skew constraint line that does not lie on the same plane. Each freedom line in both of these sets intersects one of the constraint lines at a point in finite space and intersects the other constraint line at a point at infinity.

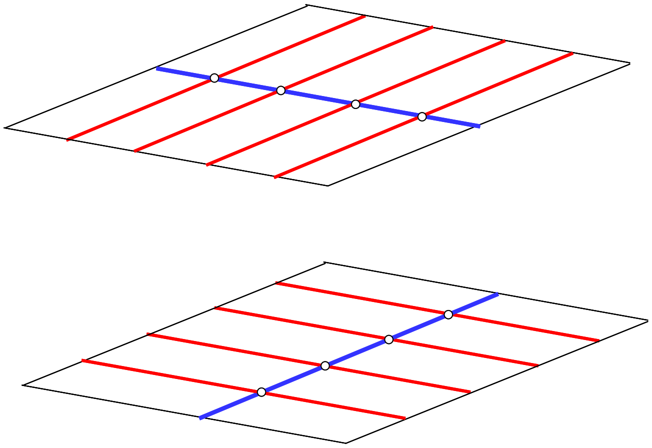


Figure 7.23: Two pure rotational planar freedom sets that contain parallel freedom lines (red) for a system with two skew constraint lines (blue).

Another freedom set within the system is a disk of pure rotational freedom lines with a center point that is intersected by one of the skew constraint lines and shares a common plane with the other skew constraint line as shown in the first picture in **Figure 7.24**. Every line within this disk will intersect both constraint lines once. There are other such pure rotational, disk-like freedom sets each of which corresponds to a single angle, α, between the plane of the disk and the plane show in **Figure 7.24**. The second picture in **Figure 7.24** shows three such disks of freedom lines for three different values of α. There are an infinite number of such disks for every value of α between zero and 180 degrees along the lower constraint line.

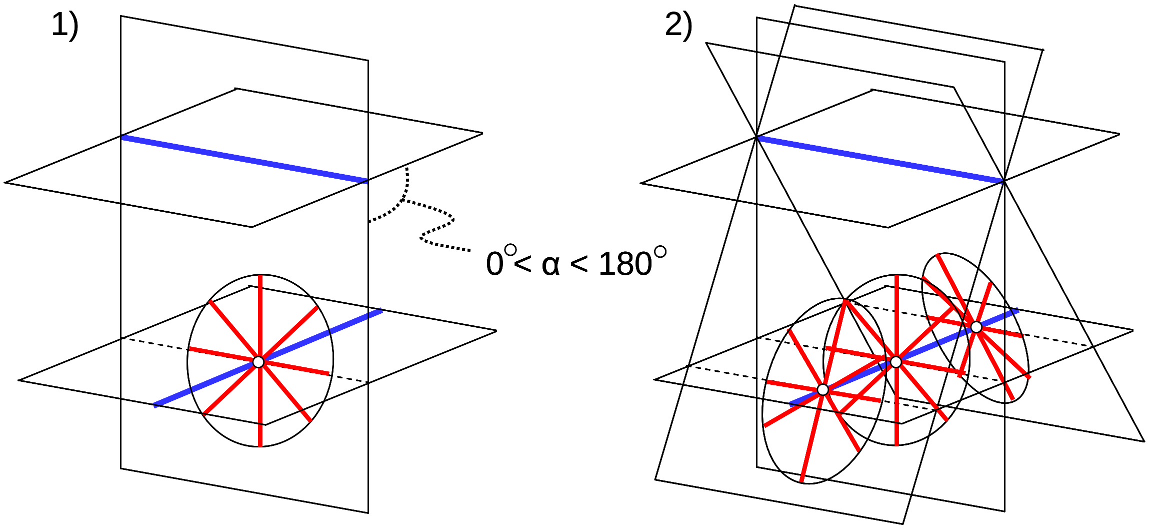


Figure 7.24: Pure rotational freedom set disks (red) exist for every value of *α* between zero and 180 degrees along the lower skew constraint line (blue)

An infinite number of such disks exist along both skew constraint lines as shown in **Figure 7.25**. The disks of freedom lines along the upper skew constraint line are unique to a single value of the angle, *β*, between the plane of these disks and the other lower plane shown in **Figure 7.25**. **Figure 7.25** technically only shows 6 freedom set disks—three disks along the lower constraint line for three different values of *α* and three other disks along the upper constraint line for three different values of *β*. Although infinite disks exist within the freedom space of the system, only 6 disks were shown in the figure to prevent visual clutter.

Note also that every line in any freedom disk on one of the skew constraint lines will share a single freedom line with every freedom disk on the other skew constraint line.

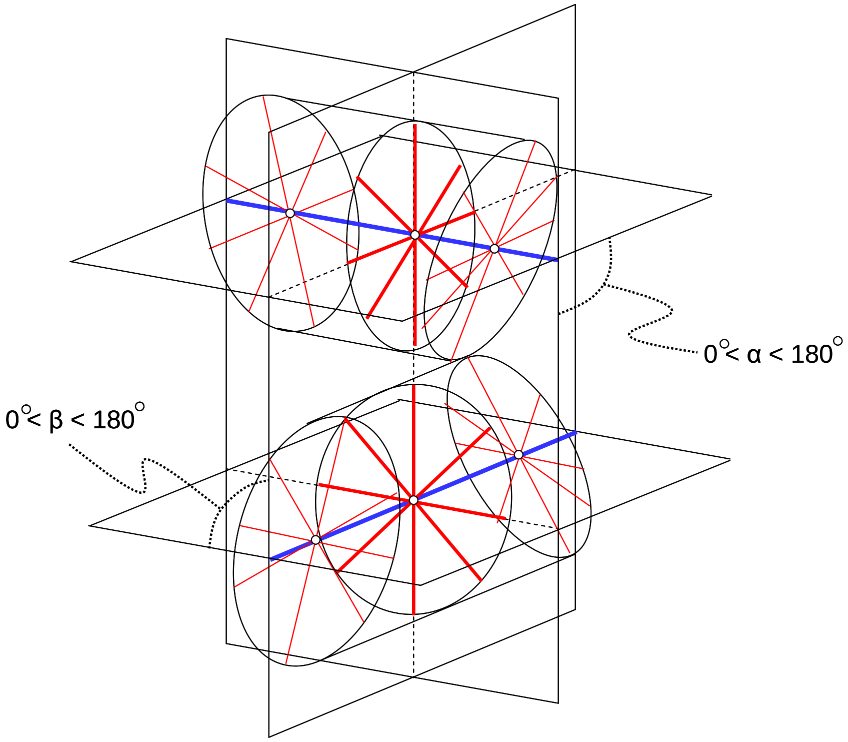


Figure 7.25: Pure rotational freedom set disks (red) exist for every value of *α* between zero and 180 degrees along the lower skew constraint line (blue). Pure rotational freedom set disks (red) also exist for every value of *β* between zero and 180 degrees along the upper skew constraint line (blue).

Note that **Figure 7.23** through **Figure 7.25** are drawn for skew constraint lines that have a skew angle of 90 degrees. The tubes containing infinite freedom set disks along each constraint line exist independent of the skew angle between the constraint lines. The only restriction on these disks is that the center of the disks must be intersected by one of the constraint lines and the other constraint line must lie on the plane of those disks.

A pure rotational hoop also exists with a normal vector that is parallel to the normal vectors of the two skew constraint line planes as shown in **Figure 7.26**. This hoop is found using the approach described in **Section 4.4** of **Chapter 4**.

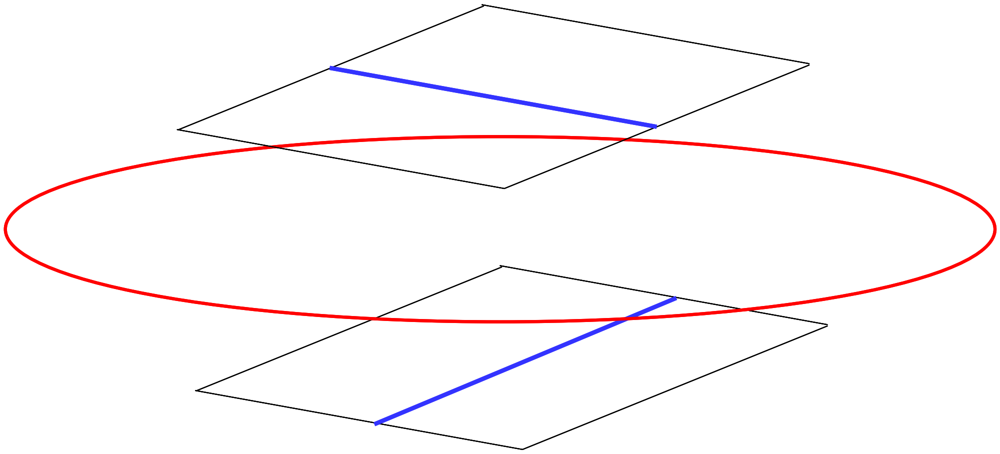


Figure 7.26: Pure rotational hoop (red) that represents a pure translation in the direction that is normal to the two parallel planes that the two skew constraints (blue) lie on.

An infinite number of screws with finite, non-zero pitch values also exist within the system. The location of these twists, however, could not be visually described in any particular freedom space that the author could find. Where these twists are can be understood since they may be found using the mathematical approach described in **Section 3.4.2** of **Chapter 3**. The pure rotational freedom sets alone should give the designer a good enough idea of the kinematics of the system without showing the screws. The kinematics of this system is so complicated that a designer would be hard pressed to find an application for its motions. The freedom space of Case 2, Type 3 without its screws is shown in **Figure 7.27**.

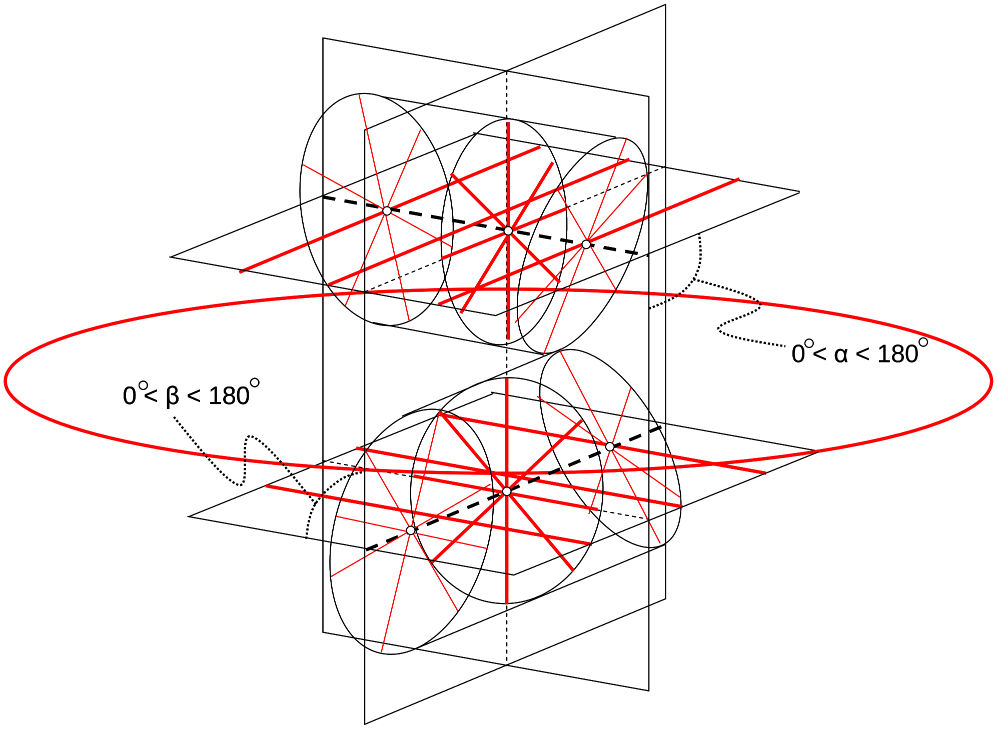


Figure 7.27: Freedom space of Case 2, Type 3 without screws

The complete constraint space of this system is essentially only two skew constraint lines shown in **Figure 7.28**. The linear combination of any two skew wrenches will not yield any other wrench solutions with *q*=0. Only constraints added along these two skew lines will be redundant.

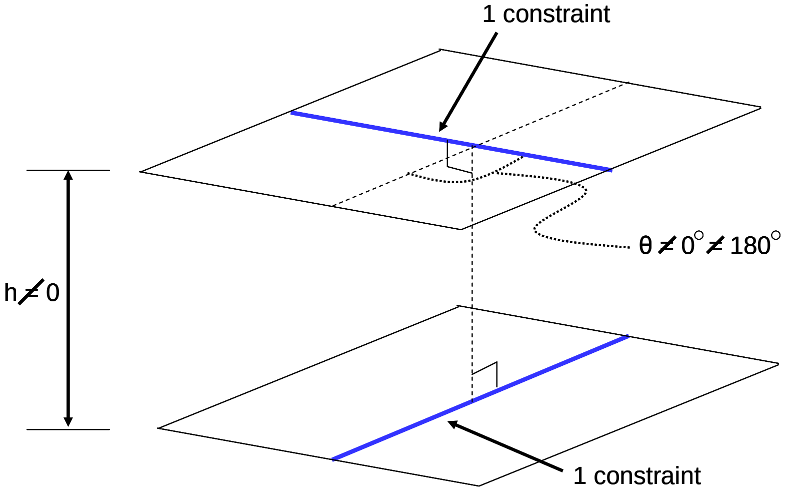


Figure 7.28: Constraint space of Case 2, Type 3

If two parallel planes are found that contain two skew lines as shown in **Figure 7.28**, the distance between these two planes, *h*, will be the shortest distance segment between the two skew lines. This shortest distance line (dashed black line) will be perpendicular to the two planes and will intersect the two skew lines as shown in **Figure 7.28**. If the skew lines are viewed from "above" looking down the axis of the shortest distance line, the angle between the two skew lines will be the skew angle, θ.

Note also if h=0 this case and type will become Case 2, Type 1 and the lines will intersect. If θ=0 or 180 degrees this case and type will become Case 2, Type 2 and the lines will be parallel. Only for the restrictions on the parameters shown in **Figure 7.28** will the lines remain skew.

Figure 7.29 shows how the freedom and constraint spaces of Case 2, Type 3 fit together. The screws with finite, non-zero pitch values are not shown to avoid cluttering the figure.

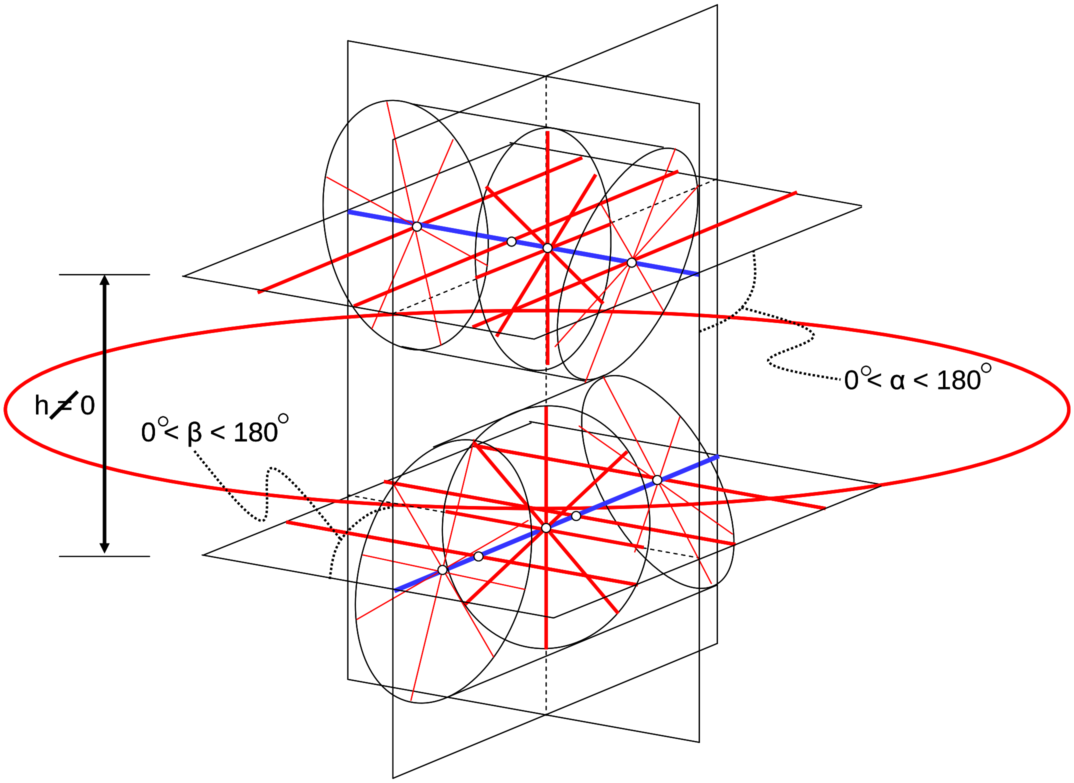


Figure 7.29: Freedom space (red) and constraint space (blue) of Case 2, Type 3 together (without the screws shown).

7.3 Case 3:

This section describes the third case of 6. The third case consists of all systems that contain three non-redundant constraints. To determine the number of types within this case, the question must be asked, “How many different ways may three non-redundant constraints be arranged such that different freedom spaces are created?” The answer is that only nine different types or nine pairs of freedom and constraint spaces exist that may be created by combining three non-redundant constraints. This may be proven by drawing every freedom space that results from adding a third constraint line to a pair of intersecting constraint lines. Then draw every freedom space that results from adding a third constraint line to a pair of parallel constraint lines. Finally draw every freedom space that results from adding a third constraint line to a pair of skew constraint lines. When this third line is added to a pair of intersecting, parallel and skew lines with every combination possible, only nine fundamentally different freedom spaces are created. This finding will now be demonstrated.

7.3.1 Third Line Added to Two Intersecting Lines

This section explores every possible way a third constraint line could be added to a system of two intersecting constraint lines. The fundamentally different freedom and constraint space pairs that are produced from this study are numbered as types within the third case and are described in detail.

To begin, consider a system where the third line that is added to the two intersecting constraint lines lies on the same plane as these two lines. Only two different freedom spaces are created by arranging three lines in this way. The first constraint line arrangement is shown in **Figure 7.30**. In this figure the third line intersects the two intersecting lines at the same point. Since this system is familiar from **Chapter 5**, it is known that the third constraint is redundant and that the system belongs to Case 2, Type 1.

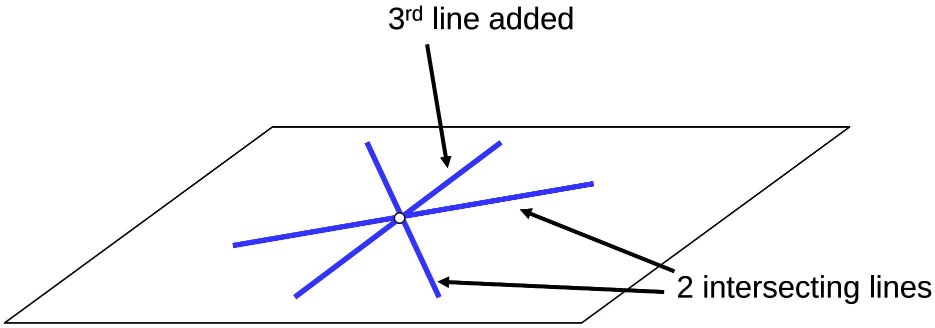


Figure 7.30: Third constraint line intersects the other two constraint lines at the same point. All three lines lie on the same plane.

If one considers the third line to be located and oriented anywhere else on the plane of the two intersecting constraint lines, a new system is created. This is shown in **Figure 7.31**. It does not matter if the third constraint line is parallel to one of the two intersecting constraint lines or not. As long as the third constraint line does not pass through the intersection point of the other two intersecting constraint lines, a new freedom space is born that will be described later.

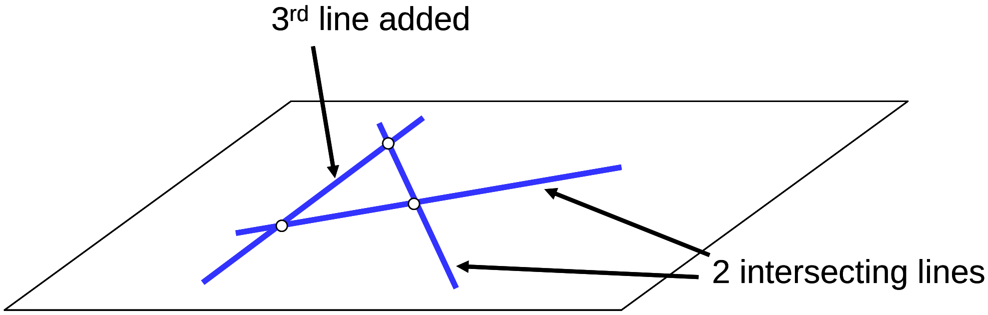


Figure 7.31: Third constraint line lies on the same plane as the other two intersecting constraint lines but does not pass through their point of intersection.

No other way exists for adding a third constraint line to the plane of two intersecting constraint lines to create a freedom space that is different from the two freedom spaces mentioned above.

The case of a third constraint line added to a plane that is parallel to the plane of the two intersecting constraint lines will now be considered. This case is shown in **Figure 7.32**. The distance between these planes is arbitrary and the third line could be oriented in any direction and could be located anywhere on its plane. Such a system will always produce the same fundamental freedom space.

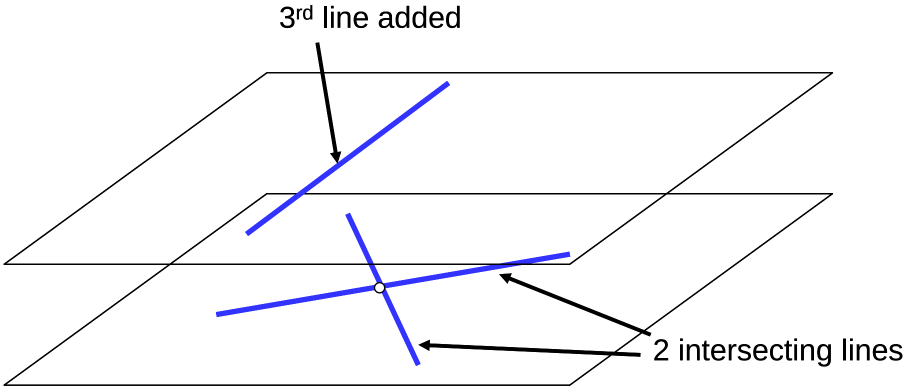


Figure 7.32: Third constraint line lies on a plane that is parallel to the plane of the other two intersecting constraint lines.

Only two different systems exist when the third line that is added does not lie on a plane parallel to or coincident with the plane of the two intersecting constraint lines. The first of these systems occurs when this third line intersects the plane of the two intersecting constraint lines at any point that is not their intersection point as shown in **Figure 7.33**. It doesn't matter if this third line intersects one of the two intersecting constraints at a different location than their intersection point and it also doesn't matter what the intersection angles, *Φ* and *θ*, equal. The freedom space produced will always be the same fundamental shape.

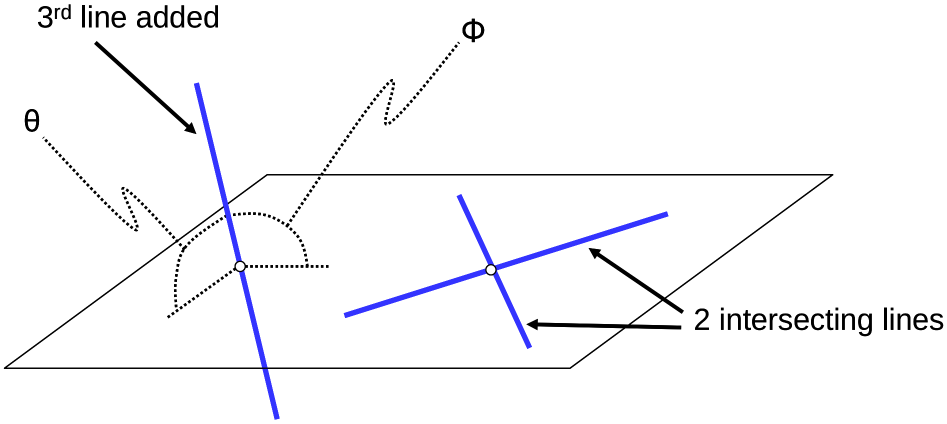


Figure 7.33: Third constraint line intersects the plane of the two intersecting constraint lines at a point that is not the point of intersection of these two lines.

The second system that exists when the third constraint line intersects the plane of the two intersecting constraint lines occurs when this third line does intersect these two intersecting lines at their point of intersection as shown in **Figure 7.34**. The intersection angles *Φ1* and *Φ2* shown in the figure do not matter to the shape of the freedom space of the system.

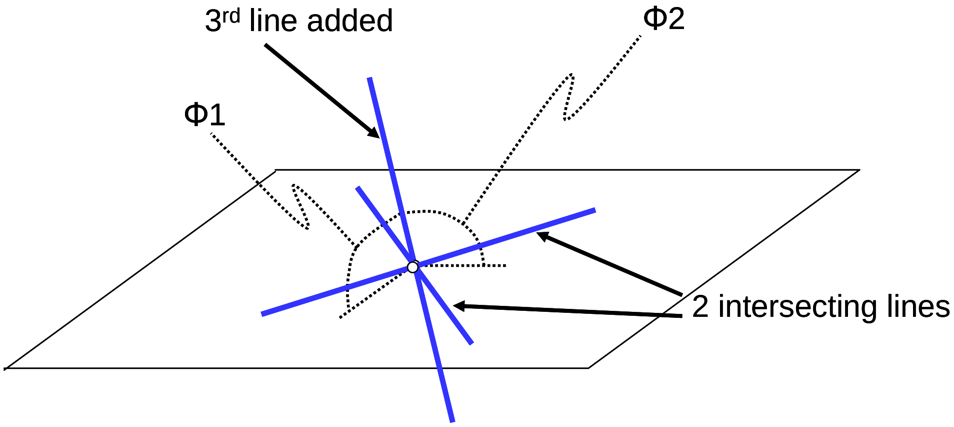


Figure 7.34: Third constraint line intersects the plane of the two intersecting constraint lines at their intersection point.

Since no other ways exist for adding a third constraint line to a pair of intersecting constraint lines, the four types or pairs of different freedom and constraint spaces that are produced from the constraint arrangements discussed above will now be described.

7.3.1.1 Case 3, Type 1:

This section describes the freedom and constraint space pair created using the three non-redundant constraints shown in **Figure 7.31**. The constraint space of this type is shown in **Figure 7.35**. It is a single planar constraint set that represents any constraint line (blue) that lies on that plane. Once three non-redundant constraint lines have been selected that don't all intersect at the same point (including at infinity), any other constraint selected from this plane will be redundant.

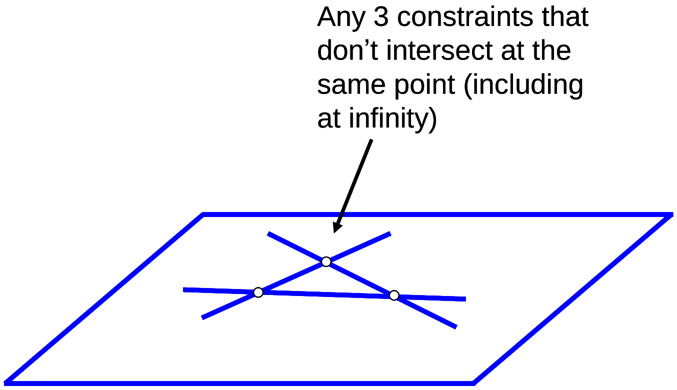


Figure 7.35: Constraint space of Case 3, Type 1

The freedom space of this system is found using the Rule of Complementary Patterns. It consists of a planar pure rotational freedom set (red) that contains every freedom line on its plane and a single pure rotational hoop (red) with a normal vector that points in the same direction as the normal vector of the planar freedom set as shown in **Figure 7.36**.

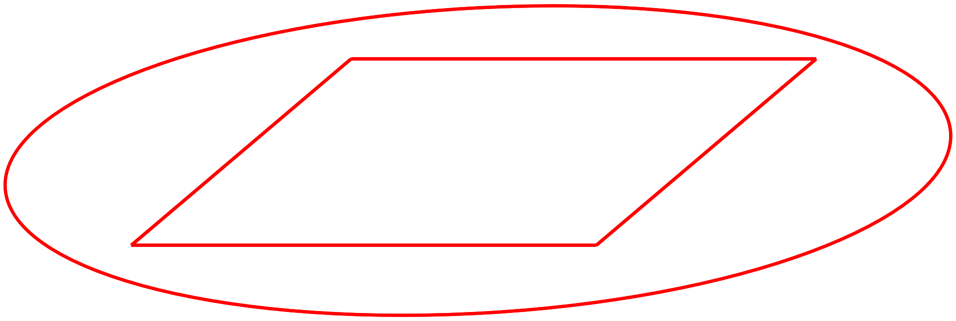


Figure 7.36: Freedom space of Case 3, Type 1

No screws with finite, non-zero pitch values exist in this system. **Figure 7.36** is, therefore, the complete freedom space and visual representation of the kinematics for this particular system of three non-redundant constraints. One can mathematically prove that no screws exist within this system by using the mathematical approach described in **Section 3.4.2** of **Chapter 3**. Once the complete freedom space has been expressed as a resultant twist that is a linear combination of three independent twists, one finds that applying **Equation (3.4)** to this resultant twist will always produce twists with zero pitch values. Thus, the freedom space will always contain pure rotations.

Figure 7.37 shows how the freedom and constraint spaces of Case 3, Type 1 fit together.

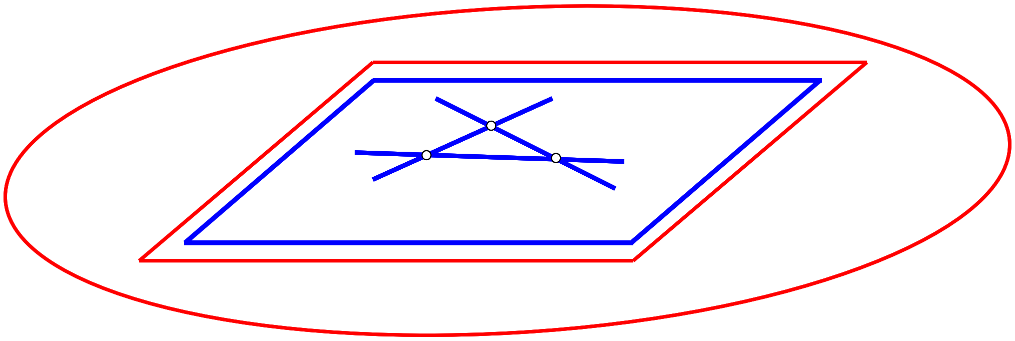


Figure 7.37: Freedom space (red) and constraint space (blue) of Case 3, Type 1 together.

7.3.1.2 Case 3, Type 2:

This section describes the freedom and constraint space pair created using the three non-redundant constraints shown in **Figure 7.32**. Blanding's Rule of Complementary Patterns may be applied to these constraint lines to determine the pure rotational freedom lines that form the system's freedom space. Only the pure rotational freedom sets within the freedom space are shown in **Figure 7.38**. This freedom space consists of three pure rotational freedom sets. One set is planar and contains every freedom line on that plane that is parallel to the third constraint line that lies on the plane parallel to the plane of the two intersecting constraint lines shown in **Figure 7.32**. These parallel freedom lines lie on the plane of these two intersecting constraint lines. The other freedom set is a disk of pure rotational freedom lines whose center point is coincident with the point of intersection of the two intersecting constraint lines. The third constraint line in **Figure 7.32** lies on the plane of this disk of freedom lines. The angle, θ, between these two planar freedom sets shown in **Figure 7.38** depends on which third constraint line is chosen on the plane parallel to the plane of the other two intersecting constraint lines. A pure rotational hoop whose normal vector points in the direction of the normal vector of the planar freedom set of parallel freedom lines also exists and is shown in **Figure 7.38**.

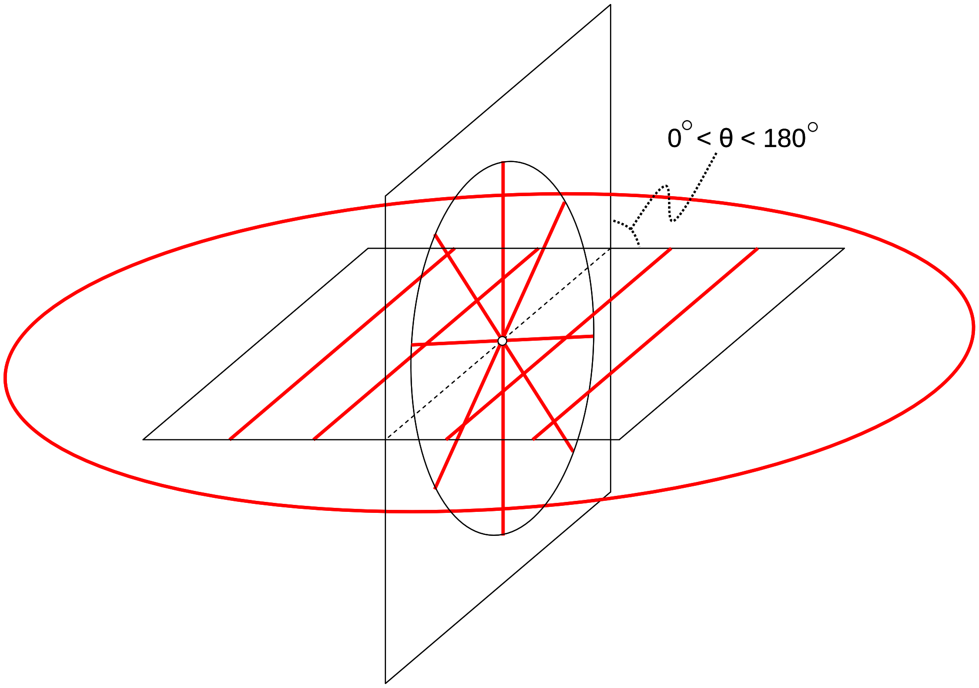


Figure 7.38: Pure rotational freedom sets within the freedom space of Case 3, Type 2.

Screws also exist within the freedom space of this system. This system's constraint space will, however, be considered before returning to the discussion of these screws.

If one finds every line that intersects every pure rotational freedom line in the freedom sets shown in **Figure 7.38**, one finds the system's complete constraint space. This constraint space is shown in **Figure 7.39**. It contains two constraint sets. One of these sets is planar and contains all constraint lines that lie on a plane and are parallel to the parallel freedom lines in the planar freedom set. This constraint set shares the same plane as the disk of freedom lines shown in **Figure 7.38**. The constraint space also contains a disk of constraint lines with a center point that is coincident with the center point of the disk of freedom lines in the freedom space. This disk constraint set also shares the same plane as the planar freedom set that contains parallel freedom lines.

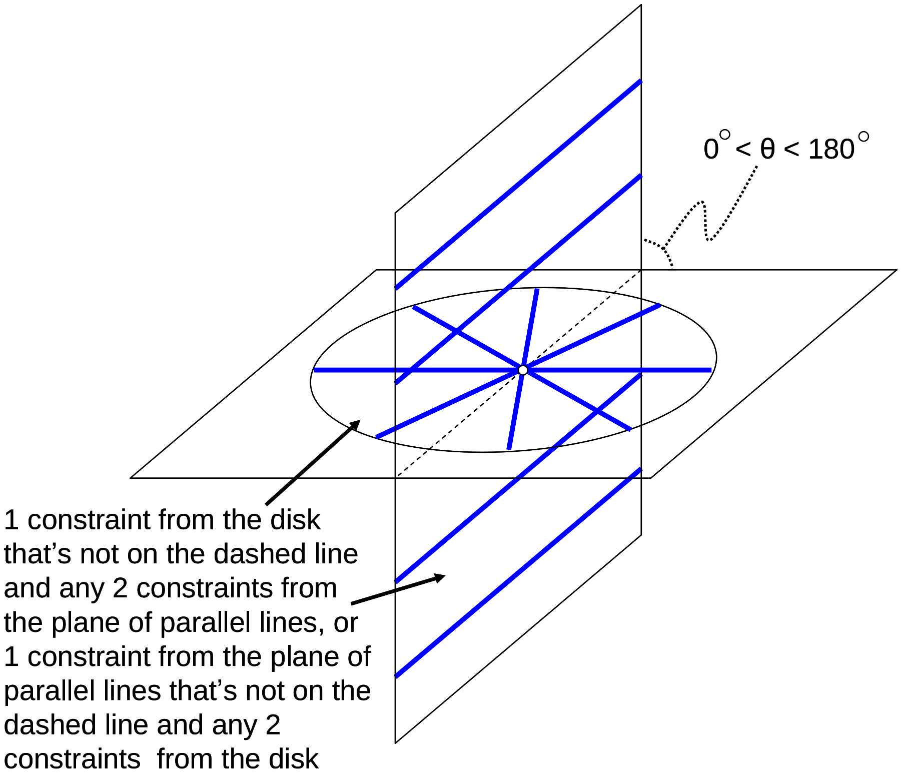

0° < θ < 180°
1 constraint from the disk that's not on the dashed line and any 2 constraints from the plane of parallel lines, or 1 constraint from the plane of parallel lines that's not on the dashed line and any 2 constraints from the disk

Figure 7.39: Constraint space of Case 3, Type 2.

Note also that each constraint set is labeled with proper instructions for guiding the designer in selecting the three non-redundant constraints for the system. The designer could not just select any three constraints from the constraint space and expect them to be non-redundant. He/she would have to select either two from the disk and one from the plane of parallel lines that's not on the line of intersection of the two planar constraint sets, or he/she would have to select two from the plane of parallel lines and one from the disk that's not on the line of intersection of the two planar constraint sets for the three constraints to be non-redundant. Once three non-redundant constraints have been selected using these instructions, any other constraint selected from the space will be redundant. Also note that if the designer decides to select two constraints from the disk and one constraint from the plane of parallel lines, he/she will recreate **Figure 7.32** from which this type was developed in the first place.

Returning again to the system's freedom space, its screws will now be found using the constraint space as a guide. If the angle, θ, between the two planar constraint sets is 90 degrees, the screws are described in **Figure 7.40**. The allowable screws are twist lines that lie on planes that are parallel to the vertical plane of parallel constraint lines and intersect the constraint line that is perpendicular to this plane and lies within the disk of constraint lines. This line is shown as a dashed blue line in the figure. The shortest distance between the allowable twist line and the vertical plane of parallel constraint lines is *L*. The angle between the twist line and the horizontal plane of the disk constraint set is *α*. Restrictions on these parameters are shown in the figure to ensure that the twist lines will be screws with pitch values that are finite and non-zero.

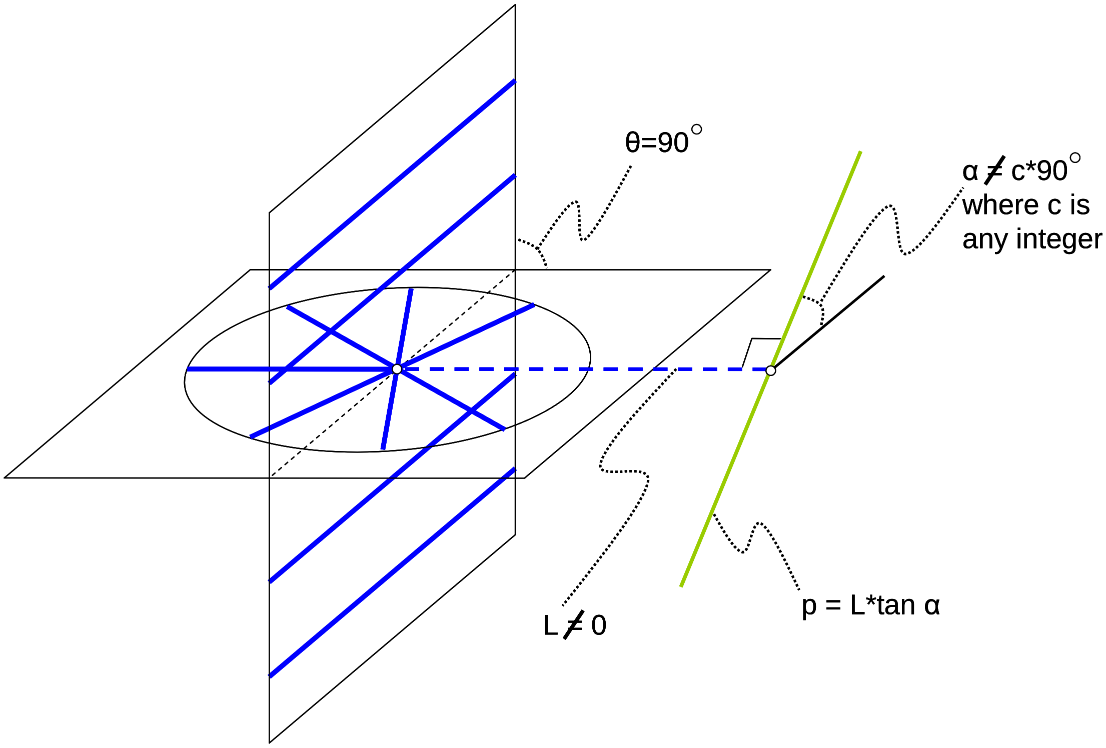


Figure 7.40: Screws (green) with finite, non-zero pitch values for Case 3, Type 2 when θ=90 degrees.

These screws were found by remembering the screws that complement a disk of constraint lines from **Figure 7.13** and remembering the screws that complement a plane of parallel constraint lines from **Figure 7.19**. The screws shown in **Figure 7.40** complement both the constraint lines in the disk and the parallel constraint lines on the plane with a pitch value given by **Equation (7.1)**.

Note that when *L*=0, the twist line will represent the red disk of pure rotational freedom lines shown in **Figure 7.38**. When *α*=0 or 180 degrees for any value of *L*, the twist line will represent the red plane of parallel pure rotational freedom lines shown in **Figure 7.38**. And when *α*=90 degrees, the twist line will represent a pure translation pointing in the direction of the normal vector of the pure rotational hoop also shown in the same figure.

Unfortunately, when the angle between the constraint sets, *θ*, is not 90 degrees, the screws are not easily found and visually described. One can, however, have a good idea of where they are. The allowable screws will always intersect and be orthogonal to one of the lines in the disk of constraint lines and they will always lie on a plane that is parallel to the plane of parallel constraint lines. They may always be found using the mathematical approach described in **Section 3.4.2** of **Chapter 3**.

Now that every pure rotation, pure translation and screw has been located for this system, the complete freedom space of this type will now be given. This freedom space is shown in **Figure 7.41**.

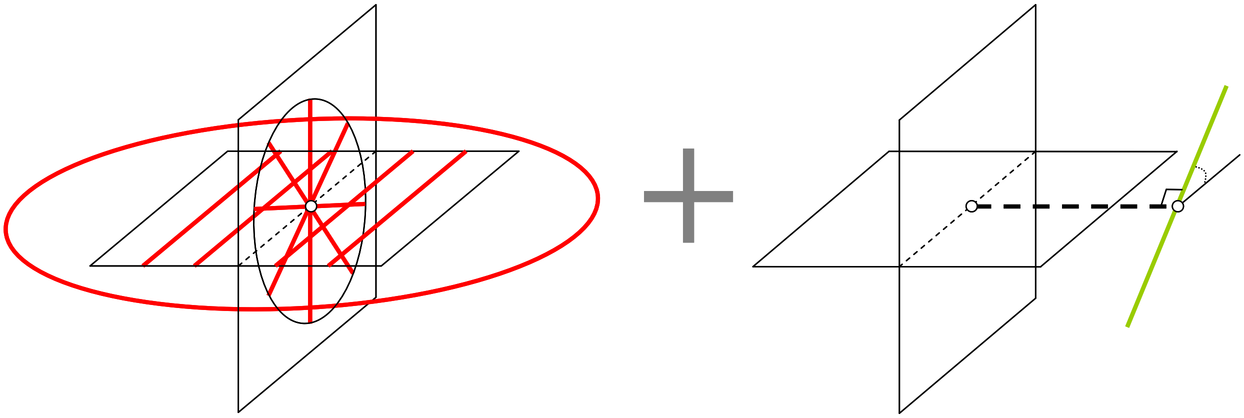


Figure 7.41: Freedom space of Case 3, Type 2

Figure 7.42 shows how the freedom and constraint spaces of Case 3, Type 2 fit together. The screws with finite, non-zero pitch values are not shown to avoid cluttering the figure.

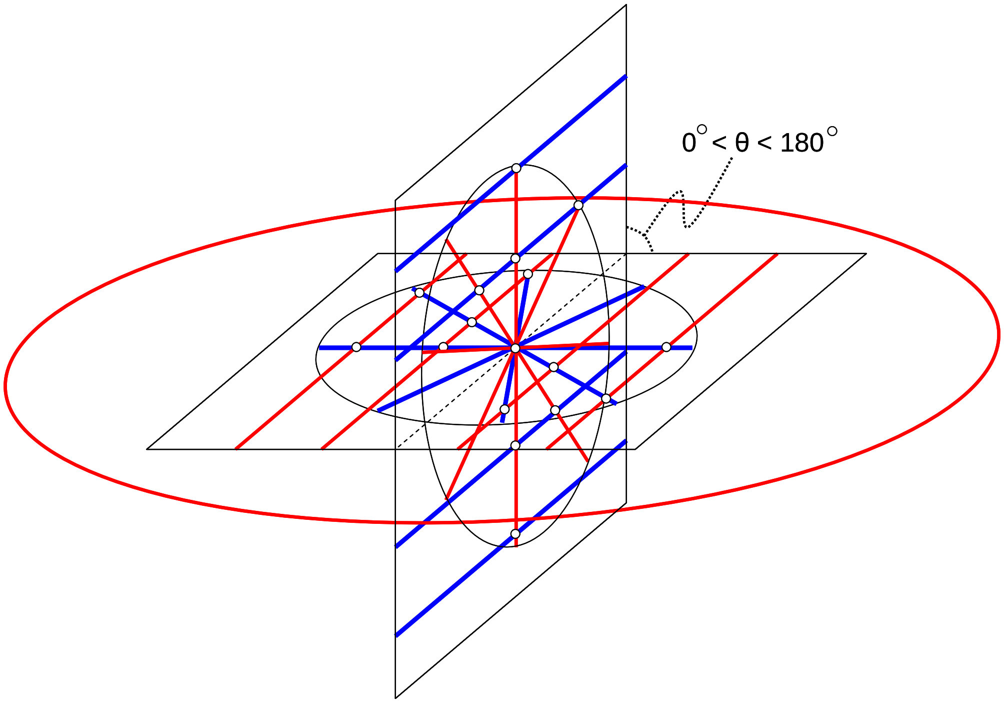


Figure 7.42: Freedom space (red) and constraint space (blue) of Case 3, Type 2 together (without the screws shown).

Note also that if *θ*=0 or 180 degrees, this case and type becomes Case 3, Type 1.

7.3.1.3 Case 3, Type 3:

This section describes the freedom and constraint space pair created using the three non-redundant constraints shown in **Figure 7.33**. The pure rotational freedom sets of this system were determined in **Section 2.3** of **Chapter 2** and are shown again here in **Figure 7.43**. These two pure rotational freedom sets consist of disks that contain pure rotational freedom lines. Their center points are separated by a distance of *d* and the angle between the planes of the two disks is *θ*.

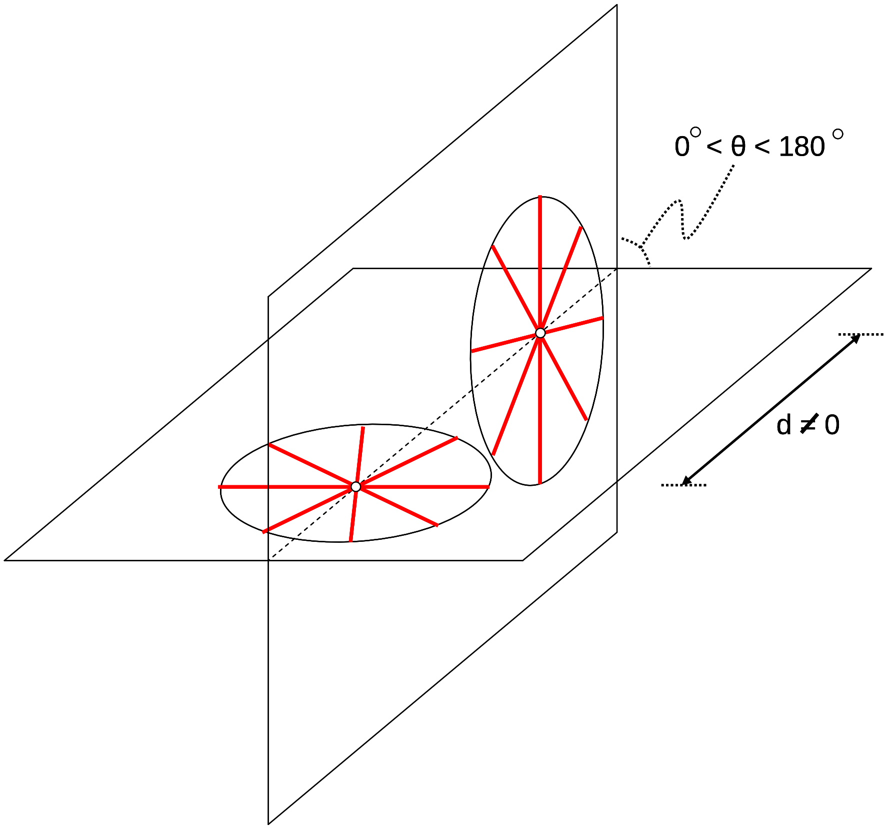


Figure 7.43: Freedom space of Case 3, Type 3 without the screws.

The freedom space of this system also contains screws with finite, non-zero pitch values. These screw lines are, however, not easily visually expressed. They may be mathematically found using the method described in **Section 3.4.2** of **Chapter 3**, but they will not be shown here for the sake of not cluttering the figure.

The complete constraint space of this system may be found by locating every line that intersects every freedom line inside the two pure rotational disks at least once. This procedure was done in **Section 5.2.3** of **Chapter 5**. The constraint space of this system is shown again here in **Figure 7.44**. This space also contains two disk sets. Both center points of these disk constraint sets are coincident with the center points of the two disk freedom sets. The freedom and constraint disks that share center points, however, do not share common planes. The freedom and constraint disks that don't share common center points, however, do share common planes.

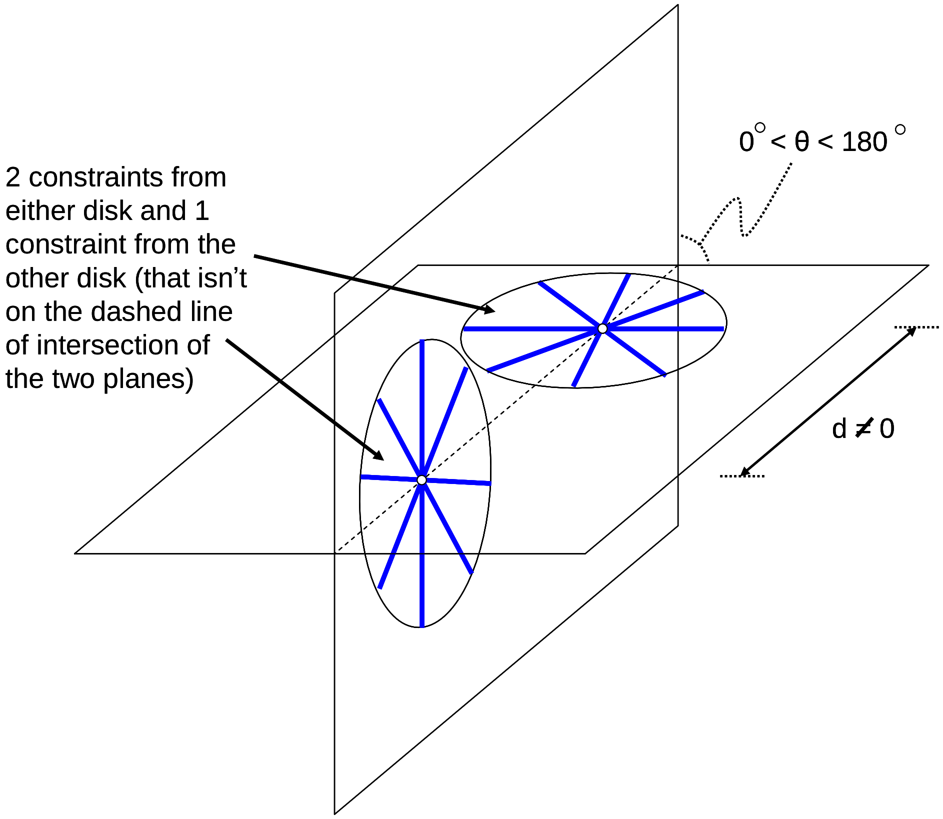

2 constraints from either disk and 1 constraint from the other disk (that isn't on the dashed line of intersection of the two planes)
0° < θ < 180 °
d ≠ 0

Figure 7.44: Constraint space of Case 3, Type 3.

Again note the instructions to the designer for selecting three non-redundant constraints. In order to select three appropriate non-redundant constraints, any two constraints may be selected from one of the disks in the constraint space and then only one constraint may be selected from the other disk that does not lie on the intersection line of the two planes.

Figure 7.45 shows how the freedom and constraint spaces of Case 3, Type 3 fit together. The screws with finite, non-zero pitch values are not shown to avoid cluttering the figure.

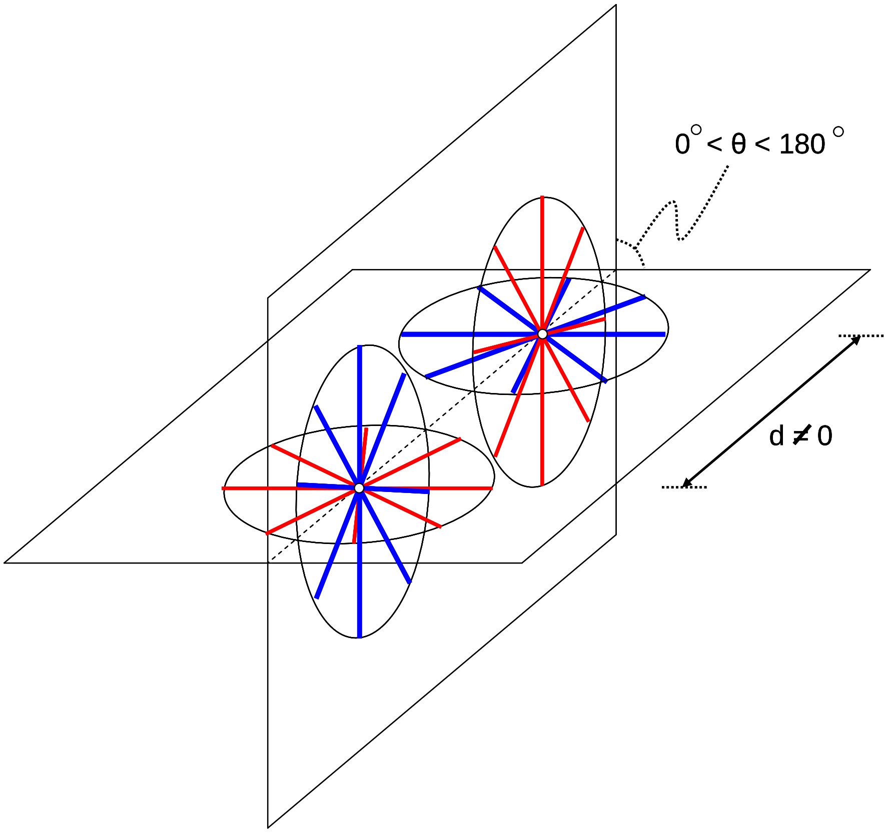


Figure 7.45: Freedom space (red) and constraint space (blue) of Case 3, Type 3 together (without the screws shown).

Note that if *θ*=0 or 180 degrees that this case and type becomes Case 3, Type 1. Also, if *d*=0, this case and type becomes Case 3, Type 4, which will now be considered.

7.3.1.4 Case 3, Type 4:

This section describes the freedom and constraint space pair created using the three non-redundant constraints shown in **Figure 7.34**. The freedom space of this system is found by using the Rule of Complementary Patterns. The only allowable twist lines for this system are the pure rotational freedom lines that intersect all three non-redundant constraint lines at their point of intersection. These lines create the spherical freedom set shown in **Figure 7.46**. The pure rotational freedom set shown in this figure is the complete freedom space of the system since no pure translations or screws exist. One can mathematically prove that there are no screws in the system by using the mathematical approach described in **Section 3.4.2** of **Chapter 3**. Once the complete freedom space has been expressed as a linear combination of three independent twists, **Equation (3.4)** could be applied to the resultant twist of the system to show that all twists in the freedom space will have zero pitch values.

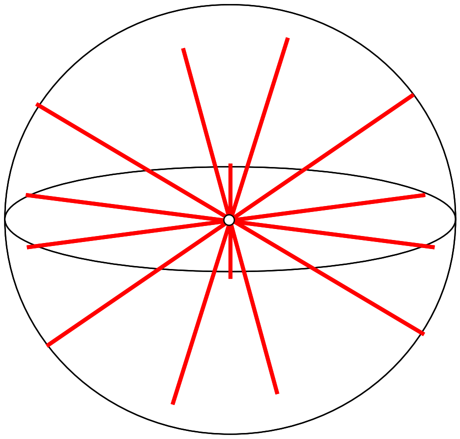


Figure 7.46: Freedom space of Case 3, Type 4.

The complete constraint space of this system is a similar spherical set that contains every constraint line that passes through the same center point as shown in **Figure 7.47**. The designer is instructed to select three constraints from the sphere that don't all lie on the same plane in order to appropriately select three non-redundant constraints. If the designer were to select three constraints from the sphere that do lie on the same plane, a disk of constraints would be selected and the system would become Case 2, Type 1.

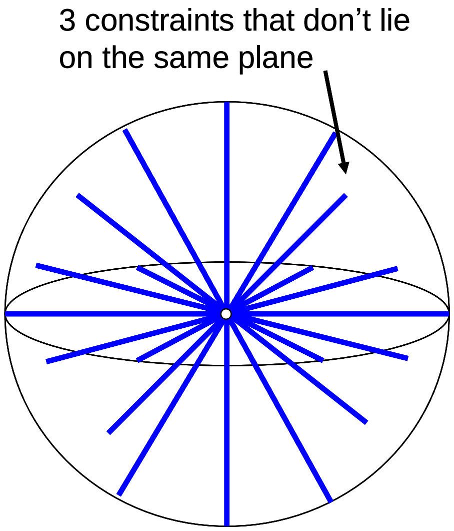


Figure 7.47: Constraint space of Case 3, Type 4.

Figure 7.48 shows how the freedom and constraint spaces of Case 3, Type 4 fit together.

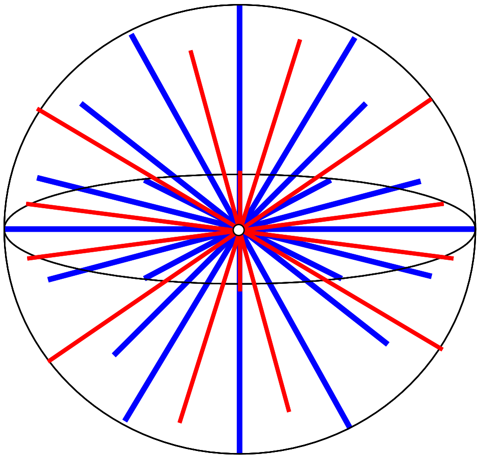


Figure 7.48: Freedom space (red) and constraint space (blue) of Case 3, Type 4 together.

7.3.2 Third Line Added to Two Parallel Lines

This section explores every possible way a third constraint line could be added to a system of two parallel constraint lines. The fundamentally different freedom and constraint space pairs that are produced from this study are numbered as types within the third case and are described in detail.

To begin, consider a system where the third line that is added to the two parallel constraint lines lies on the same plane as these two lines. Only two such line arrangements exist that produce different freedom spaces. The first is shown in **Figure 7.49**. In this figure the third line is parallel to the two parallel lines. This third constraint is redundant. It, therefore, belongs to Case 2, Type 2.

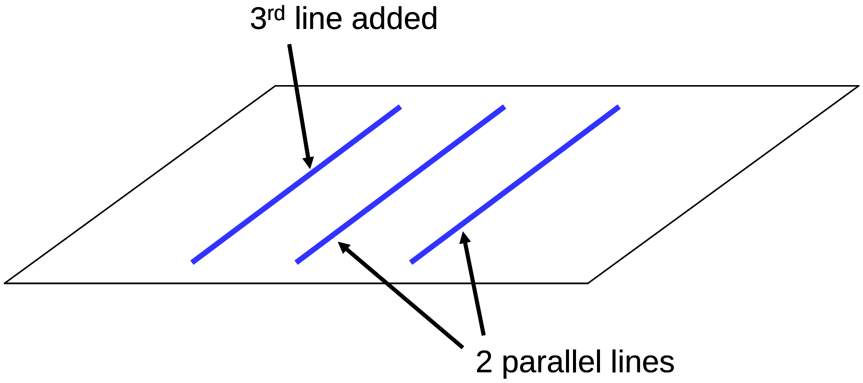


Figure 7.49: Third constraint line is parallel to the two parallel constraint lines and they all lie on the same plane.

The second way a third constraint line could be added to the plane of the two parallel constraint lines is to make sure this third line is not parallel to the other two as shown in **Figure 7.50**. Note, however, that this system belongs to the constraint space of Case 3, Type 1.

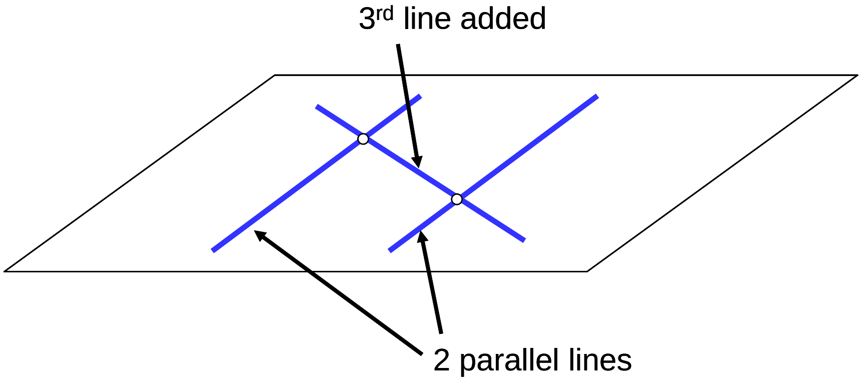


Figure 7.50: Third constraint line is not parallel to the two parallel constraint lines but they all lie on the same plane.

Consider now a system where the third constraint line that is added lies on a plane that is parallel to the plane of the two parallel constraint lines. Only two such line arrangements exist that produce different freedom spaces. The first is shown in **Figure 7.51**. In this figure the third line is parallel to the two parallel lines. This constraint layout will be shown to produce a fundamentally new freedom and constraint space pair that will be described later as Case 3, Type 5.

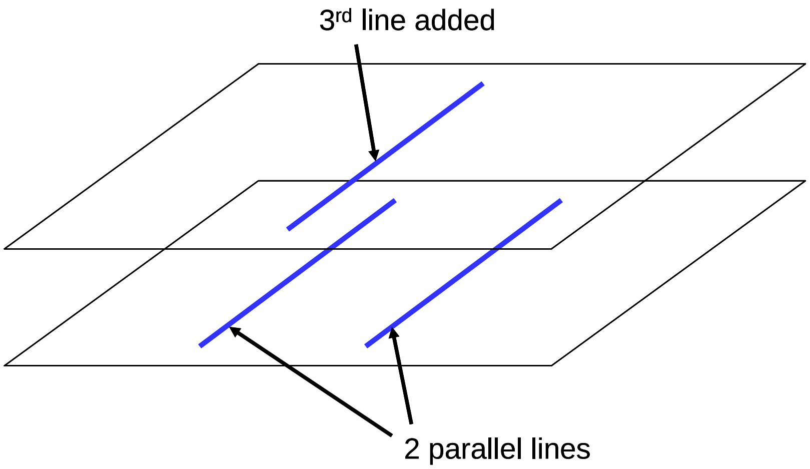


Figure 7.51: Third constraint line is parallel to the two parallel constraint lines and it lies on a plane that is parallel to the plane of the two parallel constraint lines

The second way a third constraint line could be added to a plane that is parallel to the plane of the two parallel constraint lines is to make sure this third line is not parallel to the other two lines as shown in **Figure 7.52**. This constraint layout will be shown to also produce a fundamentally new freedom and constraint space pair that will be described later as Case 3, Type 6.

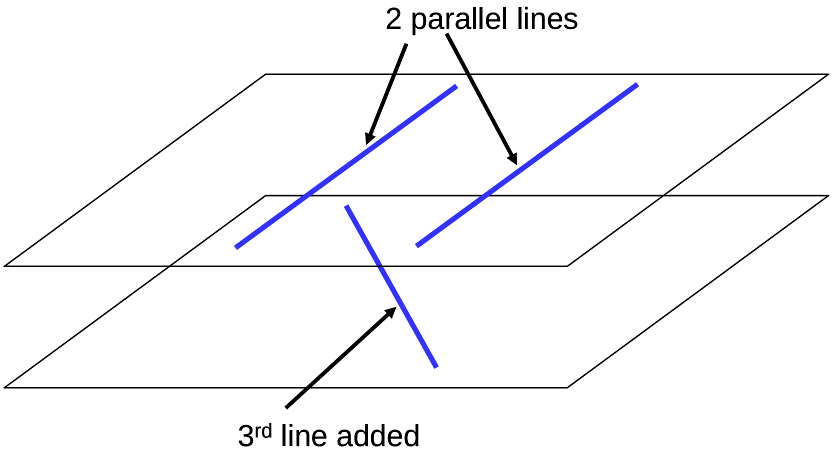


Figure 7.52: Third constraint line is not parallel to the two parallel constraint lines but it lies on a plane that is parallel to the plane of the two parallel constraint lines

Note also that it does not matter whether the plane of this third constraint line is "above" or "below" the plane of two parallel constraint lines. One could view any of these lines from any perspective and they would result in the same system with the same kinematics as long as the constraint lines maintain the same relationship with respect to each other. For instance, it may appear that the third line in **Figure 7.52** is on a plane below the plane of two parallel lines, but if one imagines the figure flipped upside down, the third constraint line would be on a plane above the two parallel constraint lines. Nothing about the system itself has changed, just the perspective that it is viewed from.

Consider now a system where the third constraint line that is added intersects the plane of the two parallel constraint lines at a single point as shown in **Figure 7.53**. If this line intersects one of the two parallel lines, it will belong to the system of Case 3, Type 2 already considered in the previous section on intersecting lines. If the third line doesn't intersect one of the two parallel lines, the system will still belong to Case 3, Type 2. The two parallel constraint lines are simply two lines from the plane of parallel constraint lines shown in **Figure 7.39** and the third line is one of the constraint lines from the disk that is not on the intersection line of the two planar constraint sets.

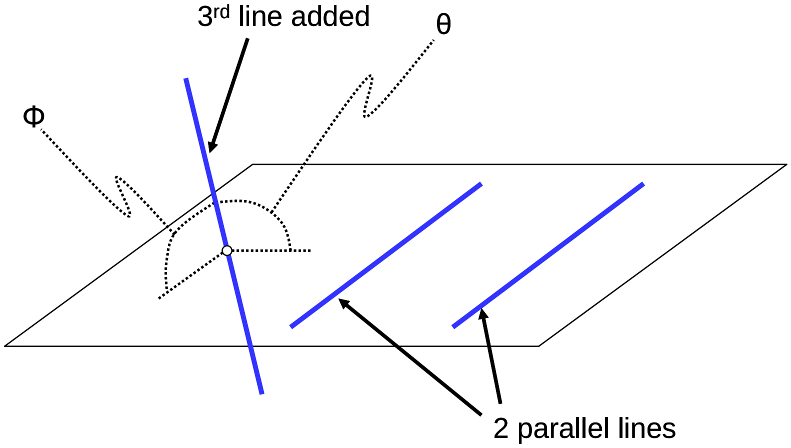


Figure 7.53: Third constraint line intersects the plane of the two parallel constraint lines at a single point

Since no other ways exist for adding a third constraint line to a pair of parallel constraint lines, the two new types or pairs of freedom and constraint spaces that are produced in the third case from the different combinations of three non-redundant constraints where at least two of them are parallel are ready to be described.

7.3.2.1 Case 3, Type 5:

This section describes the freedom and constraint space pair created using the three non-redundant constraints shown in **Figure 7.51**. The constraint space is shown in **Figure 7.54** and consists of a single constraint set represented by an infinitely large box that contains all parallel constraint lines in three-space that point in a particular direction.

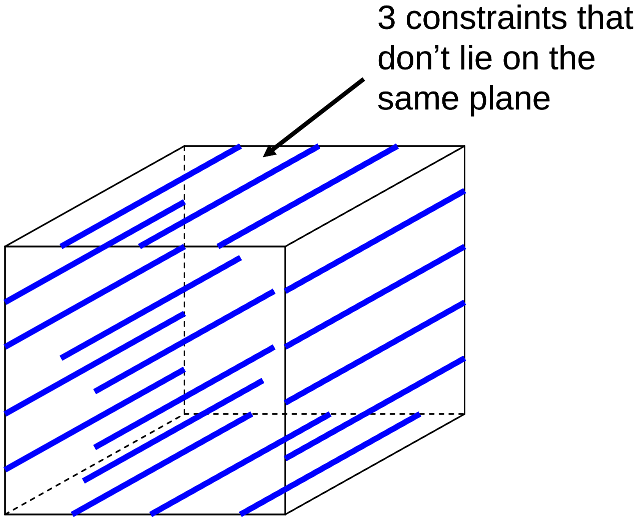


Figure 7.54: Constraint space of Case 3, Type 5

Note also the instruction to the designer to select three constraints that don't all lie on the same plane. If the designer were to choose three constraints from the same plane, only two of these would be non-redundant and this case and type would become Case 2, Type 2. Once three constraints have, however, been selected that don't all lie on the same plane, any other constraint selected from the constraint space will be redundant.

The freedom space of this system is found by locating all the pure rotational freedom lines that intersect every constraint line at least once. This freedom space is shown in **Figure 7.55**. It consists of a box-like freedom set that contains every parallel line in three-space that is parallel to the constraint lines in the constraint space. The freedom space also consists of pure rotational hoops that all intersect these parallel lines at a single point at infinity. These hoops represent a disk of pure translations that is perpendicular to the parallel lines in the freedom and constraint spaces.

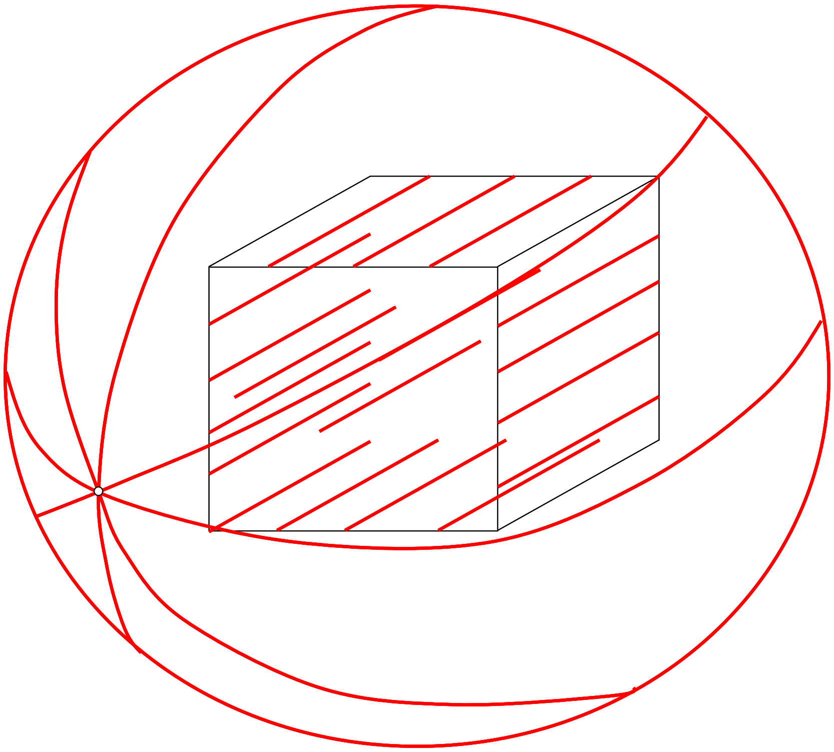


Figure 7.55: Freedom space of Case 3, Type 5

The pure rotational freedom sets shown in this figure represent the complete freedom space for the system since this system contains no screws. One can mathematically prove that there are no screws in the system by using the mathematical approach described in **Section 3.4.2** of **Chapter 3**.

Figure 7.56 shows how the freedom and constraint spaces of Case 3, Type 5 fit together.

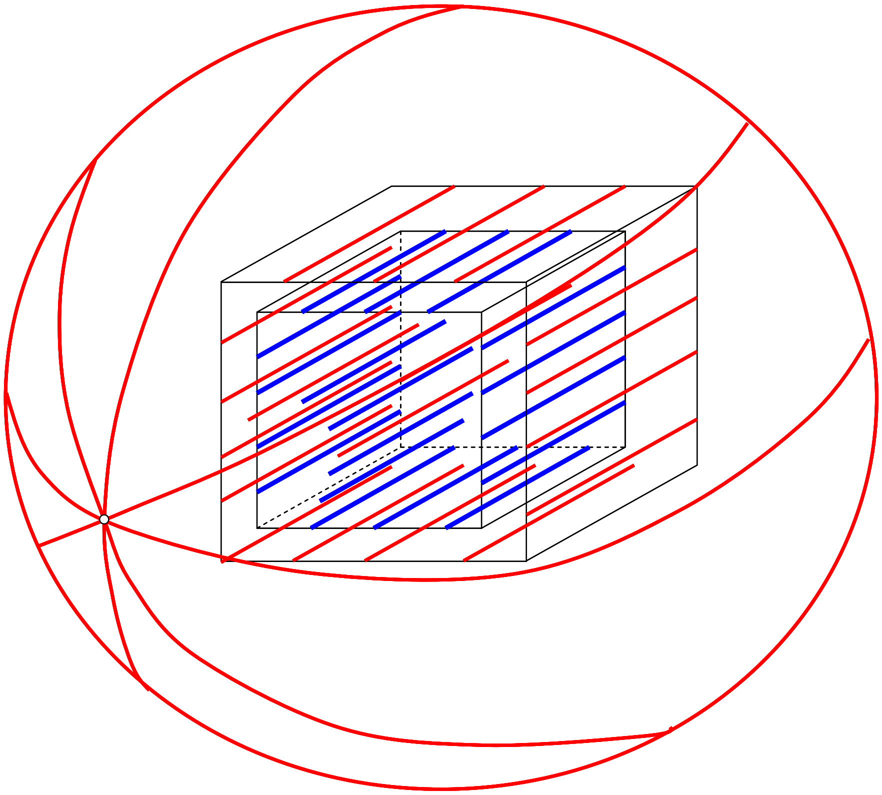


Figure 7.56: Freedom space (red) and constraint space (blue) of Case 3, Type 5 together.

7.3.2.2 Case 3, Type 6:

This section describes the freedom and constraint space pair created using the three non-redundant constraints shown in **Figure 7.52**. The pure rotational freedom sets created by locating every freedom line that intersects the three non-redundant constraint lines at least once are shown in **Figure 7.57**. Two of these sets are planar sets. The top planar set contains every freedom line that is parallel to the third constraint line on the bottom plane shown in **Figure 7.52**. The two parallel constraint lines also shown in that figure share the same plane as this top planar set. The bottom planar set contains every freedom line that is parallel to the two parallel constraint lines from **Figure 7.52**. The third constraint line shares the same plane with this bottom planar set. A pure rotational hoop also exists with a normal vector that points in the same direction as the normal vectors of the two planar sets. This hoop represents a pure translation in that direction. The distance between the planar sets is *d* and the skew angle between the parallel lines on the top set and the parallel lines on the bottom set is θ.

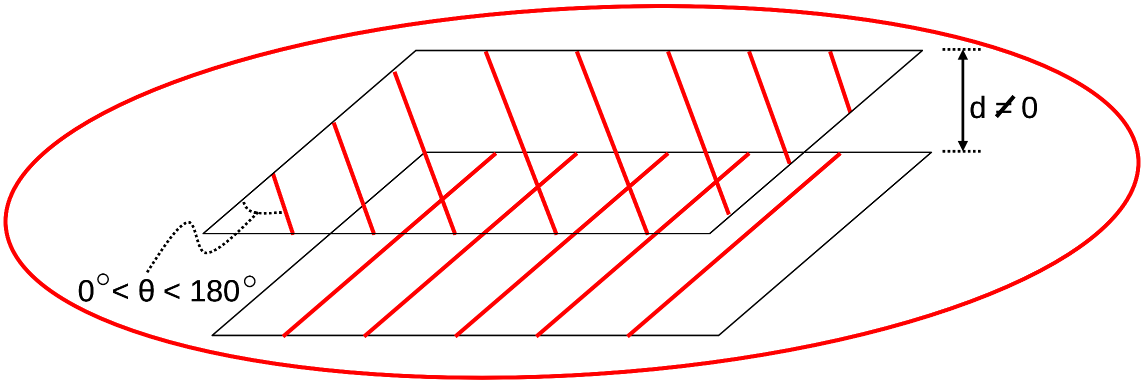


Figure 7.57: Pure rotational freedom sets of Case 3, Type 6

The screws of this system are shown in **Figure 7.58**. Every allowable screw will lie on a plane that is parallel to or coincident with the two planar freedom sets of parallel freedom lines. If the skew angle, *θ*, between the pure rotational freedom lines is 90 degrees, no screw will lay on either of the planes of the planar freedom sets and every screw will be sandwiched on planes between these sets. If the skew angle of the pure rotational freedom lines is not 90 degrees, as is the case in **Figure 7.58**, parallel groups of screws will lie on the same planes as these pure rotational freedom lines. Planar sets of parallel screws will also lie above and below the planar pure rotational freedom sets. The skew angle between these screws will be 90 degrees. Every existing screw will lie on a plane sandwiched between these top and bottom planar screw sets.

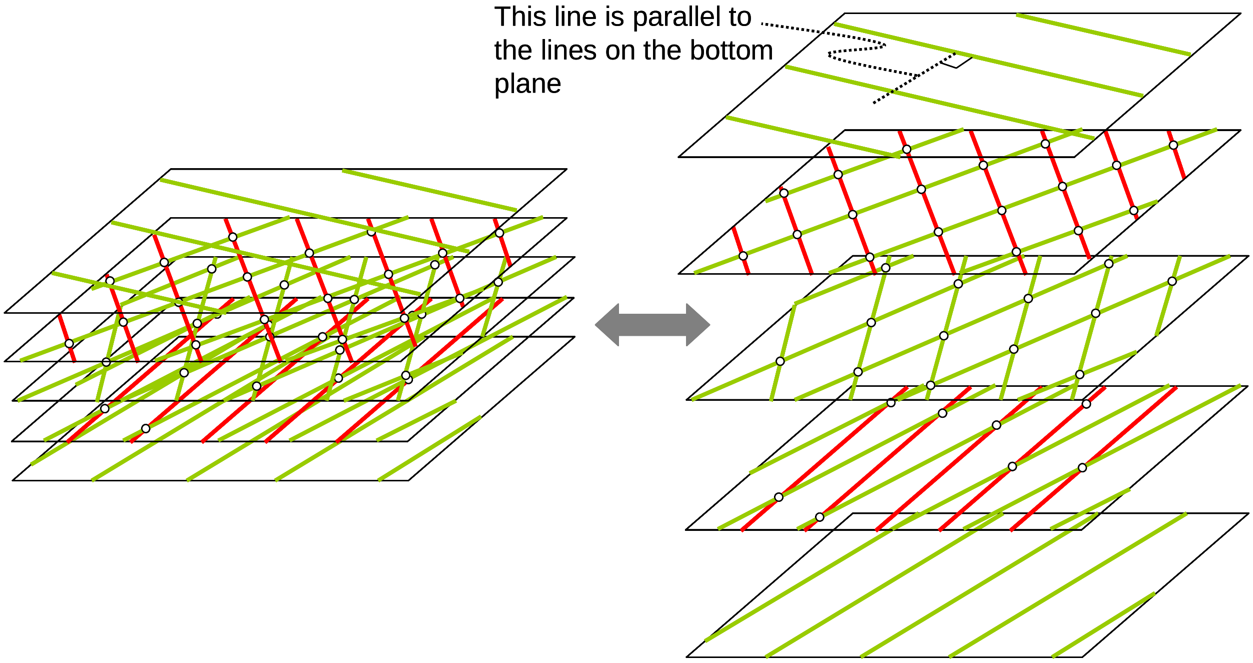


Figure 7.58: Screws (green) from Case 3, Type 6 depicted with the planar freedom sets (red). The right side of the figure shows the planes separated to help the reader better see the lines on each plane.

The top and bottom planar sets will only contain one group of parallel lines. The parallel lines on these planes will have a skew angle of 90 degrees. An infinite number of planes will be sandwiched between these top and bottom planes. These planes will all contain two groups of parallel lines. All twists within a single group of parallel lines will have the same pitch values but the two groups of parallel twists on each plane will always have different pitch values.

Another way of visualizing this freedom space is to view it as a group of infinite and identical cylindroids of the same height arranged side by side with principal generators that all lay on the same plane but all intersect at different locations on that plane. Each of these cylindroids will contain exactly two pure rotational freedom lines that are skew with respect to each other. The rest of the lines in the cylindroid will be screws with different non-zero but finite pitch values. The extreme generators of the cylindroids create the top and bottom planar sets and their skew pure rotational freedom lines create the two pure rotational planar freedom sets of parallel lines. The mathematical relationship between the two groups of parallel lines on each plane with respect to their position between the top and bottom planes may be determined using the equation of a cylindroid given in **Chapter 6** as **Equation (6.5)**. The reason why this freedom space is an infinite number of cylindroids each containing two skew pure rotational freedom lines is given in **Chapter 8**.

The complete freedom space of Case 3, Type 6 with all of its pure rotations, translations and screws is shown in **Figure 7.59**.

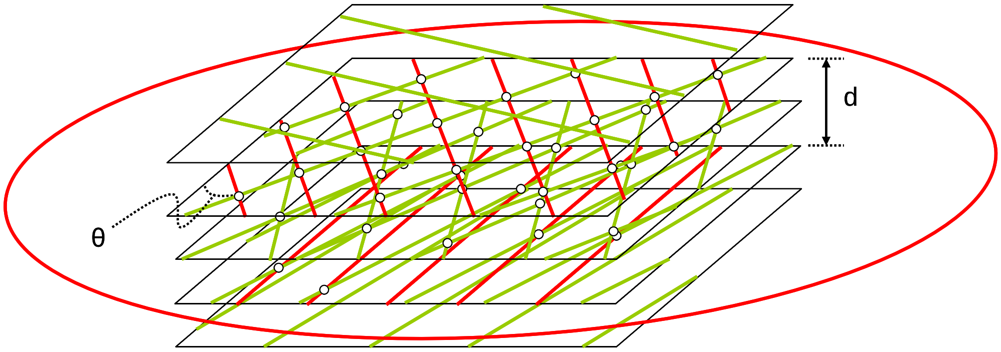


Figure 7.59: Freedom space of Case 3, Type 6

The complete constraint space of this case and type may be found by locating every line that intersects every pure rotational freedom line shown in **Figure 7.57**. This constraint space is shown in **Figure 7.60**. It consists of two planar constraint sets that contain a single group of parallel constraint lines on each plane. These planar constraint sets share the same planes as the two planar pure rotational freedom sets. The constraint lines on the top plane are parallel with the freedom lines on the bottom plane. The constraint lines on the bottom plane are parallel with the freedom lines on the top plane.

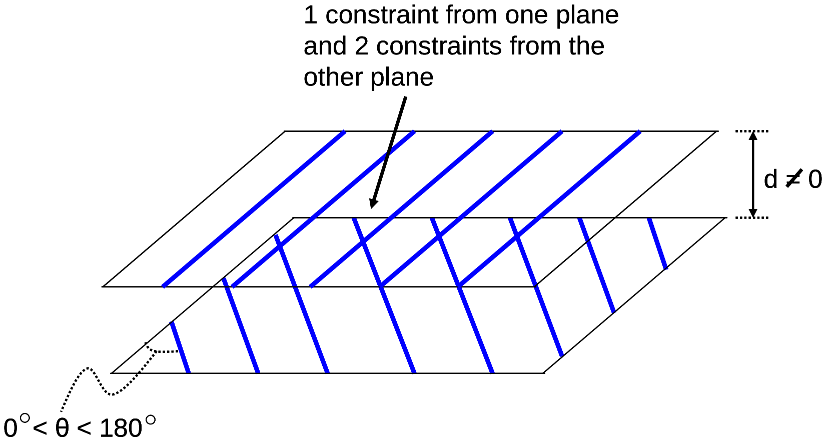


Figure 7.60: Constraint space of Case 3, Type 6

The designer is provided with instructions for appropriately selecting three non-redundant constraints from the constraint space. Once one constraint has been selected from one of the planar constraint sets and two constraints have been selected from the other planar constraint set, any other constraint selected from this space will be redundant. Note also that if the designer selected two constraints from the top plane and one constraint from the bottom plane, the constraint arrangement is the same as that shown in **Figure 7.52**.

Note also that if *d*=0, this case and type becomes Case 3, Type 1. If *θ*=0 or 180 degrees, this case and type becomes Case 3, Type 5. If *d*=0 and *θ*=0 or 180 degrees simultaneously, this case and type becomes Case 2, Type 2.

Figure 7.61 shows how the freedom and constraint spaces of Case 3, Type 6 fit together. The screws with finite, non-zero pitch values are not shown to avoid cluttering the figure.

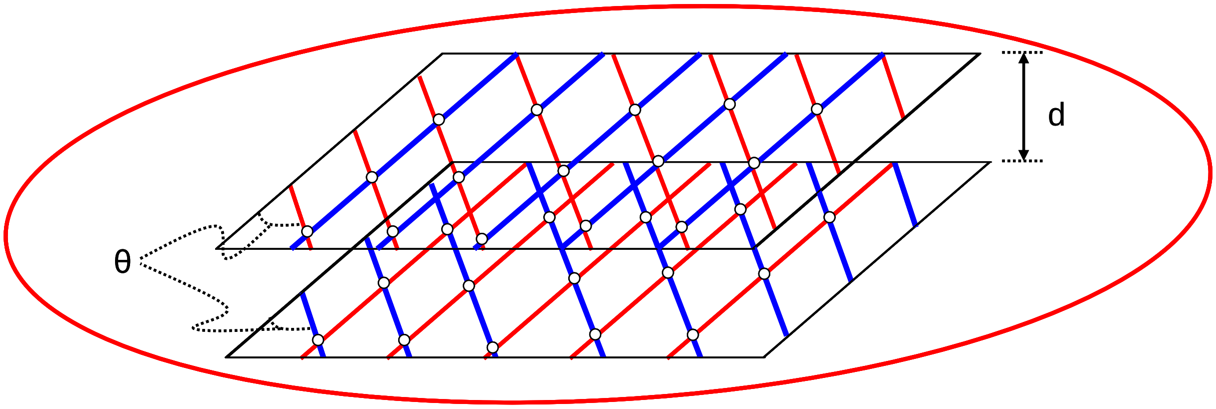


Figure 7.61: Freedom space (red) and constraint space (blue) of Case 3, Type 6 together (without the screws shown).

7.3.3 Third Line Added to Two Skew Lines

This section explores every possible way a third constraint line could be added to a system of two skew constraint lines. The fundamentally different freedom and constraint space pairs that are produced from this study are numbered as types within the third case and are described in detail in this section.

To begin, consider systems where the third line that is added to the two skew constraint lines is not parallel to and does not intersect either of these lines such that new freedom and constraint space pairs are created. Only two such line arrangements exist. The first consists of a third constraint line added to a plane that is parallel to the two parallel planes of the two skew constraint lines where the third constraint line is also skew to both of these skew lines as shown in **Figure 7.62**. The second line arrangement consists of a third constraint line added such that it intersects the two parallel planes of the two skew constraint lines without intersecting either of the lines themselves as shown in **Figure 7.63**. These are the only two fundamentally different ways three skew lines may be combined. This statement, although initially non-intuitive, will become obvious by the completion of this chapter.

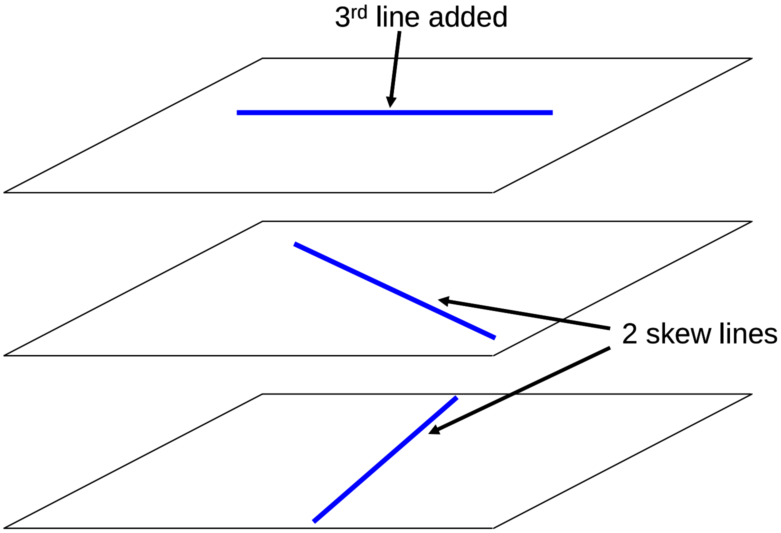


Figure 7.62: Third constraint line added to a plane that is parallel to the two parallel planes of the two skew constraint lines where the third constraint line is not parallel to either of them.

Note if a third constraint line had been added on one of the two parallel planes of the two skew constraint lines, this third line would intersect or be parallel to the skew constraint line with which it shares a plane. The type of such a system would, therefore, have already been considered in a previous section. Or if the third constraint line had been added on a plane parallel to the two parallel planes of the two skew constraint lines, but the third line was parallel to one of these skew lines, its type would also have already been considered in a previous section. It doesn’t matter if the plane that the third line is added on lies above, between, or below the two parallel planes of the two skew constraint lines. Any three skew lines similar to those shown in **Figure 7.62** will belong to a single type within Case 3.

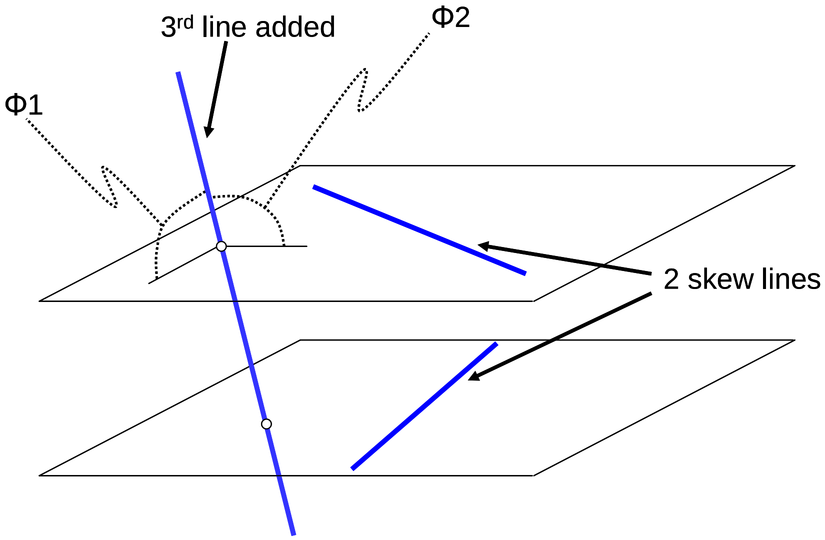


Figure 7.63: Third constraint line added such that it intersects the two planes of the two skew constraint lines but does not intersect either of the skew lines themselves.

Note if the third constraint line that intersects the two parallel planes of the two skew constraint lines intersects one or both of these lines, its type would have already been considered in the section on intersecting lines.

Now that the two different ways three skew constraint lines may be arranged have been identified, each of these ways will be studied to determine how many different types may be identified and described within each.

7.3.3.1 Three Skew Lines on Three Parallel Planes

This section will examine the system of three skew lines on three parallel planes shown in **Figure 7.62**. After studying this system, Case 3, Type 7 will be identified and mathematically described.

To begin the study, a particular arrangement of these three skew lines will be considered where the third skew line is intersected by the shortest distance line of the other two skew lines as shown in **Figure 7.64**. Recall that the shortest distance line of two skew lines is the line perpendicular to the two parallel planes of the two skew lines that also intersects both of these skew lines. It is called the shortest distance line because the line segment created between the two intersection points of both skew lines and this line will be the shortest line segment possible that intersects both skew lines. For such a system, if one were to look down the shortest distance line from above, it would appear like the three skew constraint lines intersect at the same point like a disk as shown in **Figure 7.64**.

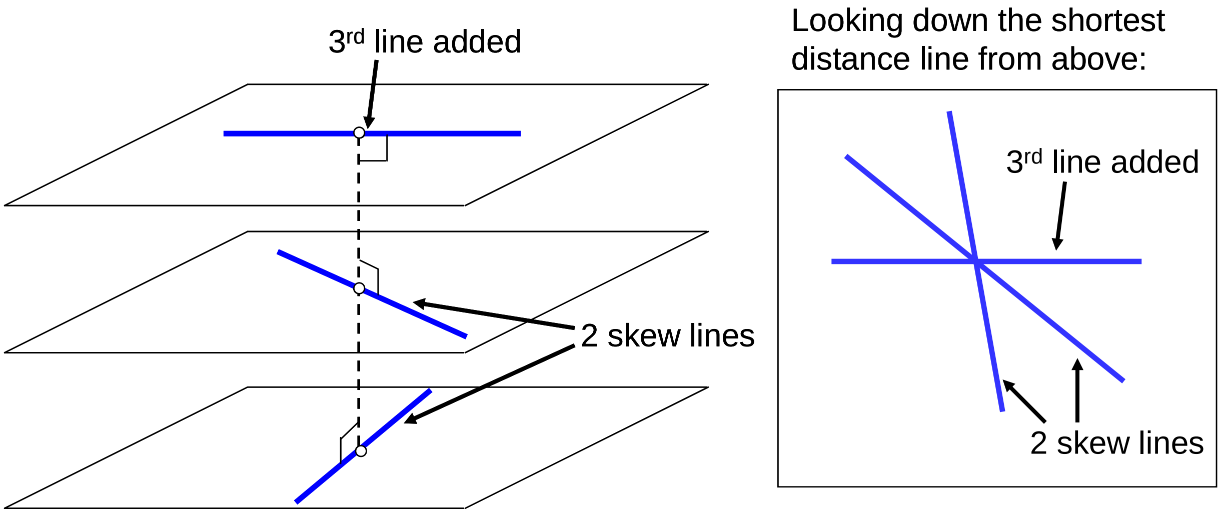


Figure 7.64: Third constraint line added to a plane parallel to the plane of the two skew constraint lines that is intersected by the shortest distance line (dashed black) of these two skew constraint lines.

Applying Blanding's Rule of Complementary Patterns to finding the pure rotational freedom lines that intersect all three constraint lines of this system is not easily done by inspection. If the constraint line on the middle plane is treated as a series of infinite points in the midst of two other skew constraint lines on the top and bottom parallel planes, **Appendix C** can be applied to finding each freedom line that corresponds to each point along the middle constraint line that also intersects both of the two skew constraint lines. **Figure 7.65** shows three such freedom lines that correspond to three points along the dotted middle skew line. Using **Appendix C** to determine the location and orientation vectors of these new freedom lines, one also learns that each of these freedom lines is skew with respect to each other.

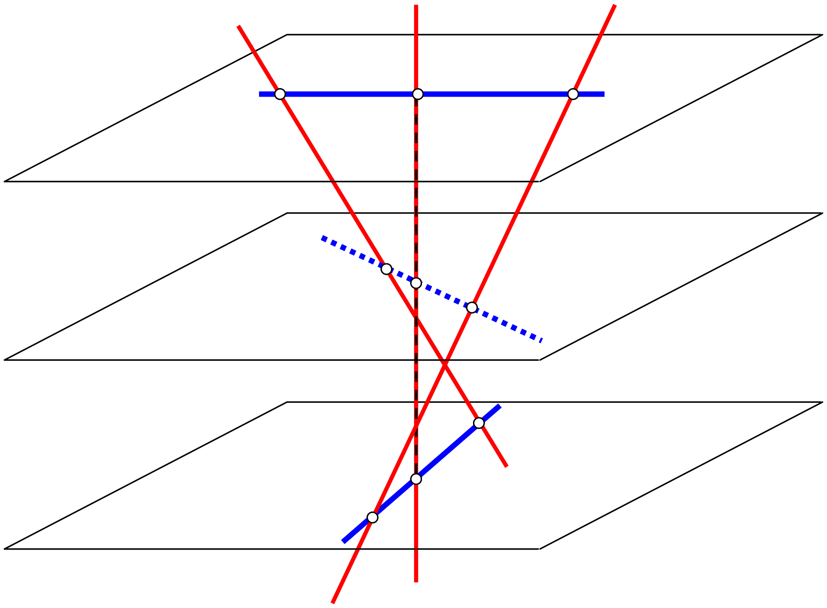


Figure 7.65: Three freedom lines (red) found by choosing three random points along the middle constraint line (dotted blue) and applying Appendix C to each of these points and to the other two skew constraint lines (blue).

After drawing a couple more of these freedom lines, a pattern becomes apparent. A freedom set is formed that consists of a ruled surface of freedom lines that rotate as they translate along an axis that is perpendicular to these lines. This freedom set is shown in **Figure 7.66**. The author calls this freedom set a ribbon set because it looks like a long ribbon that stretches out to infinity with a single 180 degree twist at its center. This twist occurs because the lines on the ribbon's surface almost rotate a full 180 degrees from one end of the ribbon to the other. The twist at its center is very much exaggerated in **Figure 7.66**. In actuality, this twist happens quite rapidly at the ribbon's center and most of the rest of the ribbon consists of lines that are almost parallel and are asymptotically becoming more parallel as the ribbon extends to infinity at both of its ends.

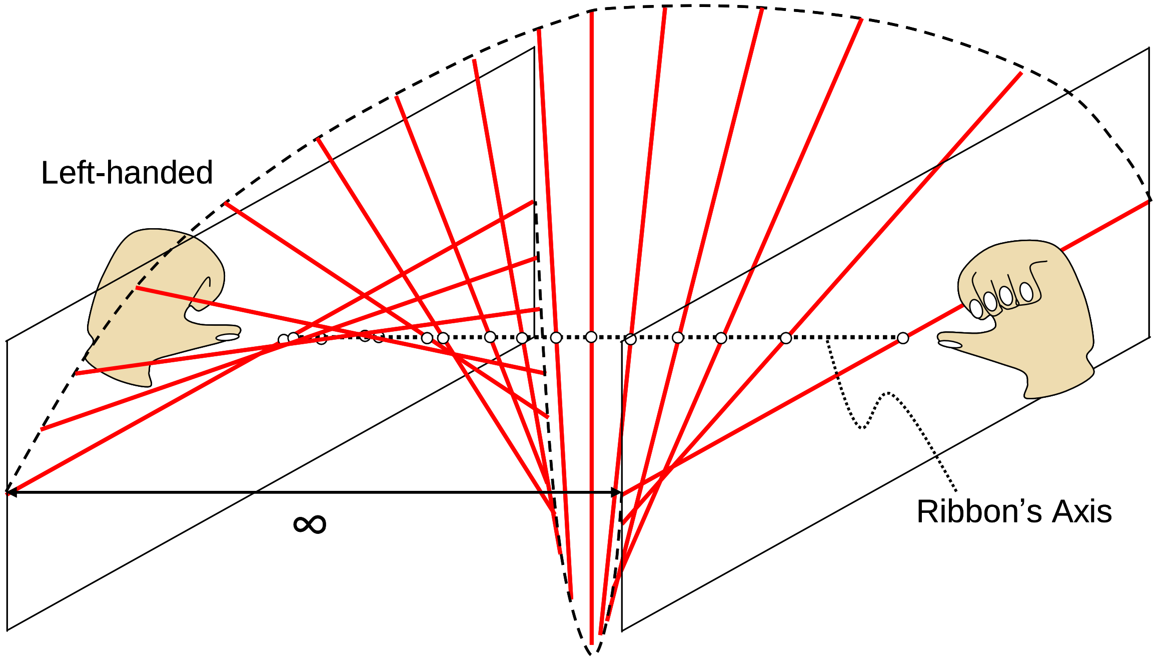


Figure 7.66: Left-handed, orthogonal, ribbon freedom set of pure rotational freedom lines (red) that rotate as they translate along an axis (dotted black line). The rate that the lines rotate as they translate has not accurately been drawn to emphasize the ribbon surface’s twist at its center.

It was briefly mentioned that the freedom lines within this ribbon space are perpendicular to the axis that they translate on. For this reason, such a ribbon will be called an orthogonal ribbon. Every ribbon freedom set that is created from three skew non-redundant constraints like those shown in **Figure 7.64** will be orthogonal ribbons. If the third constraint line in that figure did not intersect the shortest distance line of the other two skew constraint lines, the resultant ribbon freedom set would not be orthogonal. Non-orthogonal ribbons will be considered later on in this section.

The ribbon freedom set shown in **Figure 7.66** is also a left-handed ribbon. It is called a left-handed ribbon because the fingers of a left hand determine the direction the lines on the ribbon rotate as they translate if the left hand’s thumb is pointing along the ribbon’s axis in the direction of translation. Note also that it does not matter which direction along the ribbon’s axis the thumb is pointed as long as it is the left hand’s thumb that is doing the pointing.

Before completing the study of this system's freedom space, its constraint space will be considered briefly. One can find this space by apply the Rule of Complementary Patterns to the three skew pure rotational freedom lines shown in **Figure 7.65**. If the middle skew freedom line in this figure is treated as an infinite series of points and the other two skew freedom lines also shown in the figure are considered, the principles from **Appendix C** can be applied for finding every constraint line that intersects all three skew freedom lines simultaneously. In this way, one can mathematically determine the system's complete constraint space. The constraint lines in the constraint space will not only intersect the three pure rotational freedom lines shown in **Figure 7.65**, they will also intersect every freedom line in the entire ribbon freedom set thus satisfying the Rule of Complementary Patterns.

The constraint set that is formed by finding these constraint lines is shown in **Figure 7.67**. This constraint set is also an orthogonal ribbon space that extends infinitely far in both directions and contains a 180 degree twist at its center. The only difference between this ribbon constraint set and its complementary ribbon freedom set, other than its location in space, is that it is a right-handed ribbon instead of a left-handed ribbon. That is the fingers of a right hand will determine the direction the constraint lines rotate as they translate along the ribbon's axis if the right hand's thumb is pointing along this axis.

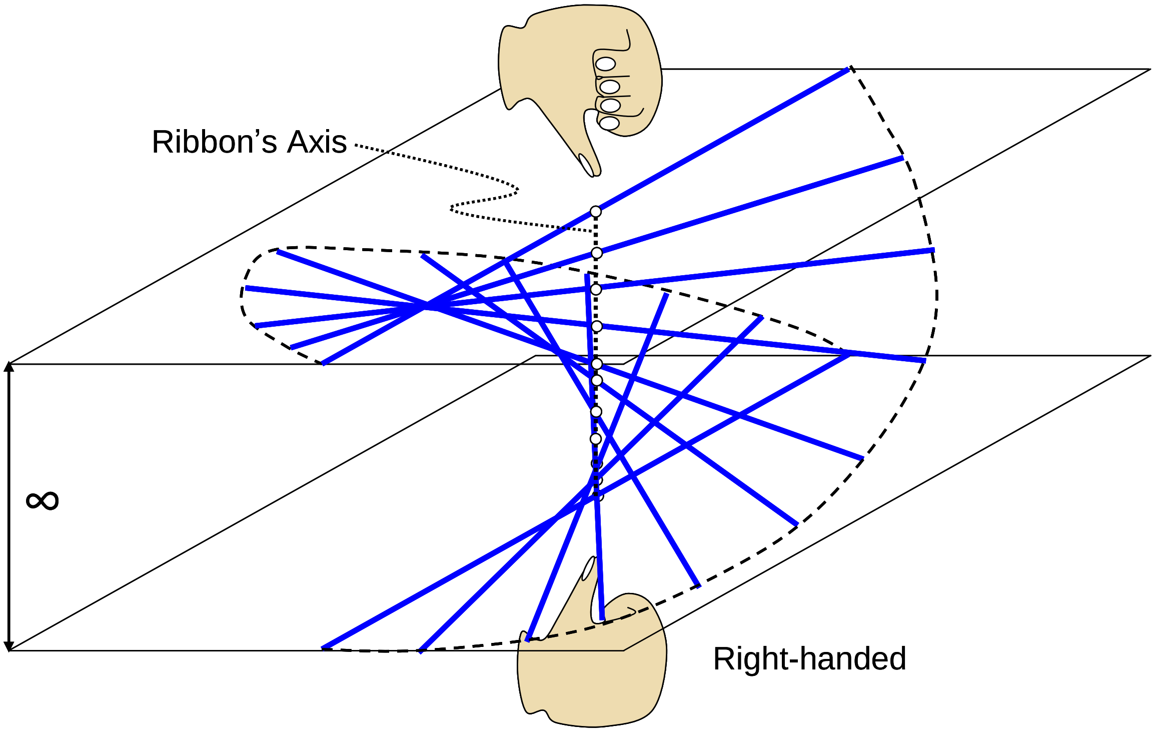


Figure 7.67: Right-handed orthogonal ribbon constraint set of constraint lines (blue) that rotate as they translate along an axis (dotted black line). The rate that the lines rotate as they translate has not accurately been drawn to emphasize the ribbon surface's twist at its center.

This space includes the three skew constraint lines from **Figure 7.64** on its surface. Any three constraint lines chosen from this space will produce the same ribbon freedom set as the ribbon freedom set created by the original three skew constraint lines from **Figure 7.64**. Any other constraint selected from this space once the three non-redundant constraints have been selected will be redundant and will have no effect on the freedom space of the system. One can correctly deduce, therefore, that every line in any ribbon space is a linear combination of three other independent twists or wrenches in that space where every twist or wrench in that space is a pure rotation (*p*=0) or an ideal constraint (*q*=0).

In order to better visualize how these ribbon freedom and constraint sets fit together, a program was written using MATLAB that draws some of the actual line segments within each of these spaces and generates them using three initial skew constraint lines that satisfy the conditions of the skew lines shown in **Figure 7.64**. The code is given in **Appendix D** and an example picture is shown here in **Figure 7.68**.

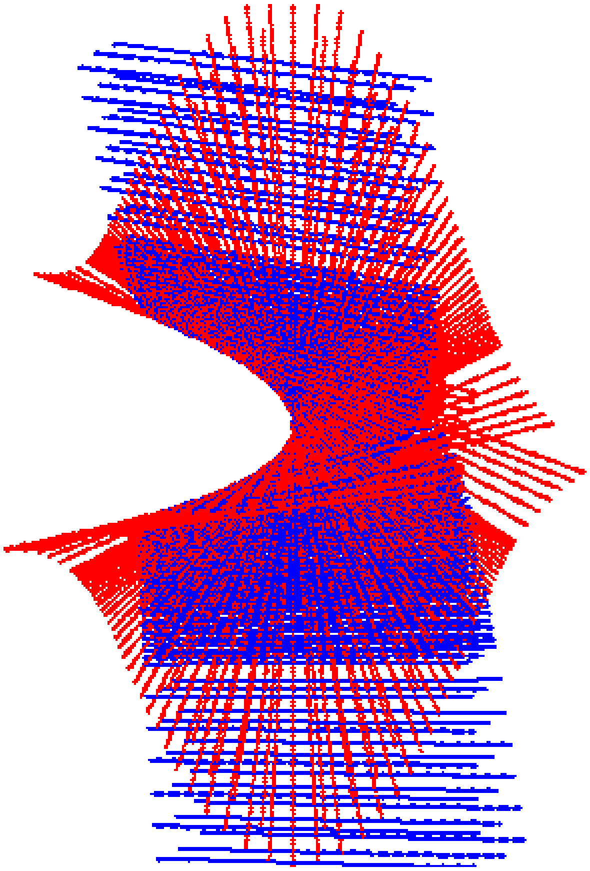


Figure 7.68: Example of a left-handed, orthogonal, ribbon freedom set (red) with its complementary right-handed, orthogonal, ribbon constraint set (blue)

The relationship between ribbon freedom sets and their complementary ribbon constraint sets will now be considered. From the previous example, it should not come as a surprise that any left-handed ribbon freedom set will have a complementary right-handed ribbon constraint set. It should, therefore, also not come as a surprise that any right-handed ribbon freedom set will have a complementary left-handed ribbon constraint set.

The rate that the lines rotate as they translate within the ribbon freedom and constraint sets will also be of the same magnitude at corresponding locations along each ribbons' axis when these spaces are complementary. To demonstrate this fact, one first must define a ribbon's pitch. The pitch of a ribbon is defined as the rate the lines within the ribbon translate as they rotate. In other words, it is the change in the position of the lines within the ribbon along the ribbon's axis divided by the change of the skew angle between these lines. This is shown in **Figure 7.69** with an orthogonal, left-handed, ribbon constraint set.

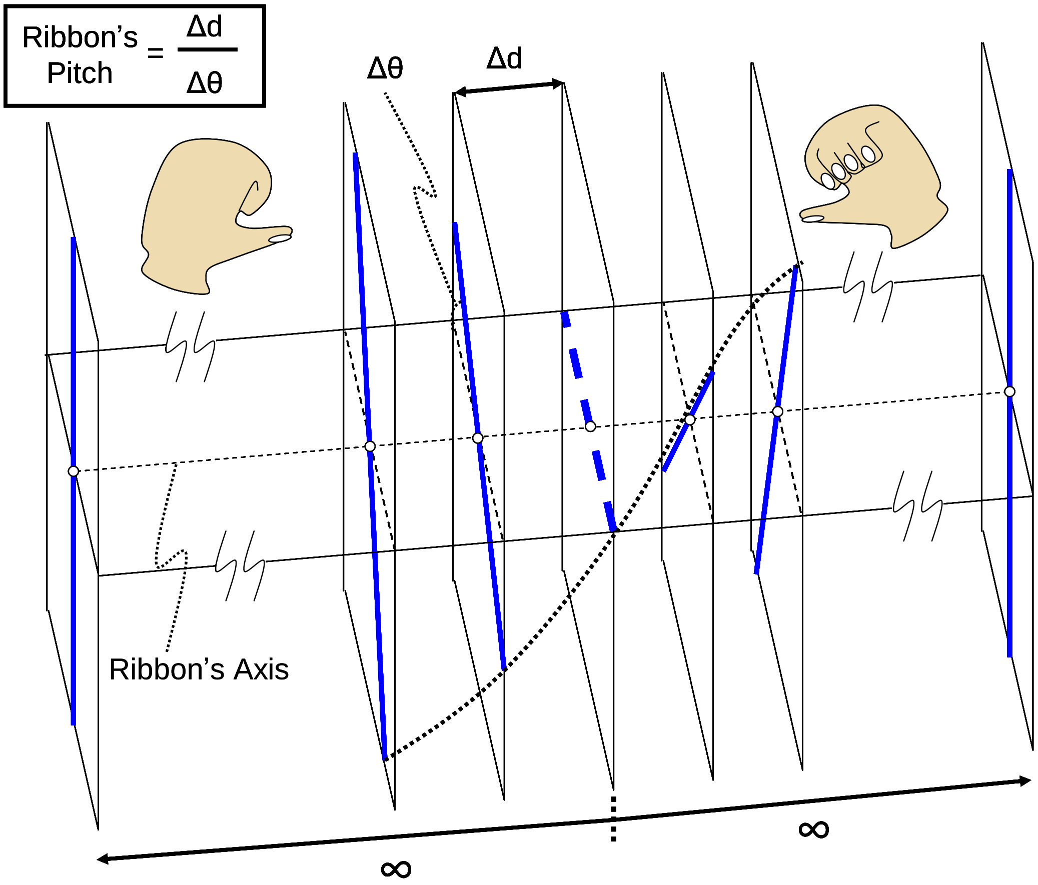


Figure 7.69: Definition of a ribbon's pitch for an orthogonal left-handed ribbon constraint set

Note from **Figure 7.69** that every plane that contains a constraint line is orthogonal to the axis of the ribbon (which is why the ribbon is called orthogonal). The center point of the ribbon lies on the intersection of the ribbon's axis and the dashed blue constraint line shown in the figure. This central constraint line is asymptotically orthogonal to the constraint lines that are infinitely far away from the center point at both ends of the ribbon.

This ribbon constraint set's complementary orthogonal right-handed ribbon freedom set is shown in **Figure 7.70**.

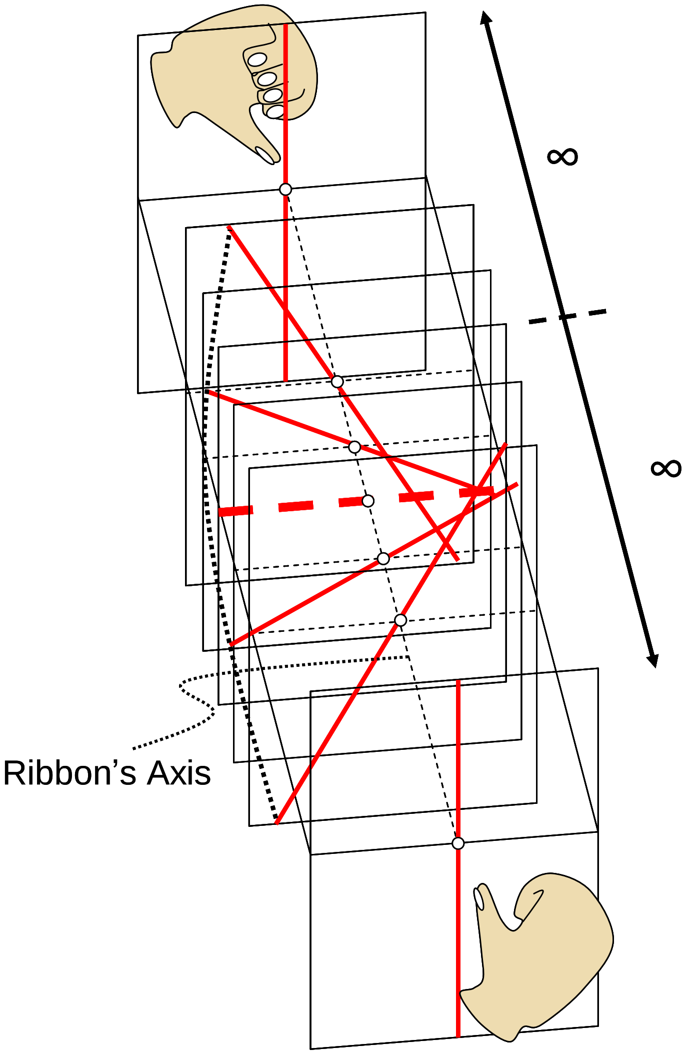


Figure 7.70: Complementary orthogonal right-handed ribbon freedom set

Note again that each plane that contains a pure rotational freedom line is orthogonal to the axis of this ribbon. The central freedom line (dashed red) is also asymptotically orthogonal to the two freedom lines infinitely far away on both ends of the ribbon.

Figure 7.71 shows how these two complementary ribbons fit together.

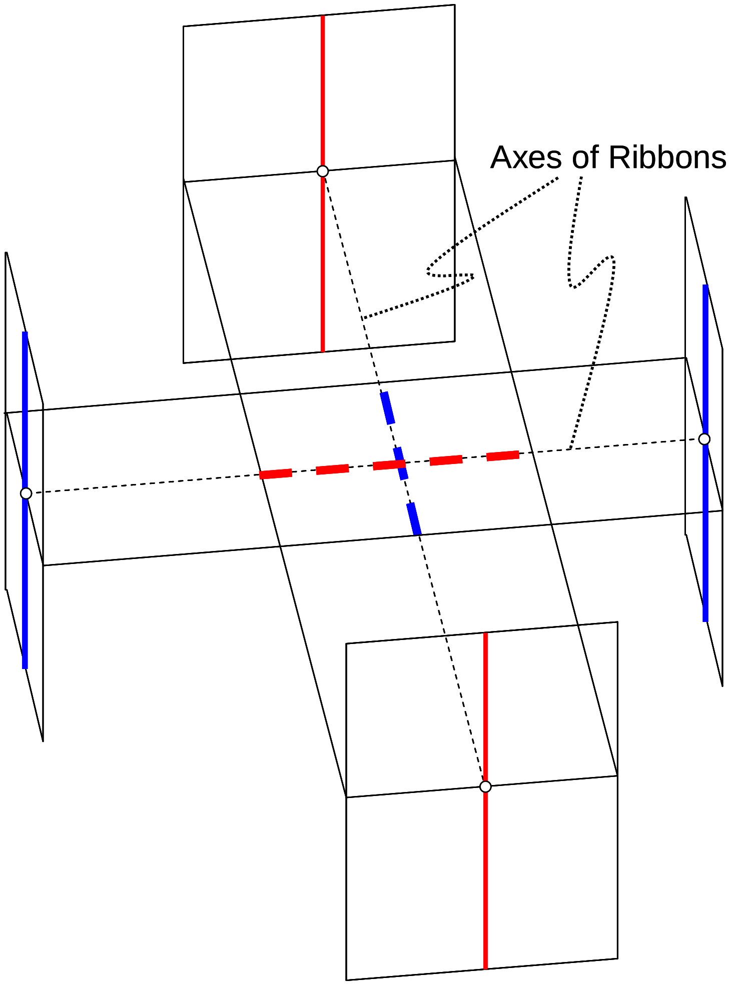


Figure 7.71: How complementary orthogonal ribbons fit together

From **Figure 7.71** one learns that the axes of orthogonal complementary ribbons are themselves orthogonal and intersect at the central point of both ribbons. The central constraint line (dashed blue) from the constraint ribbon set is coincident with the axis of the freedom ribbon set. And the central freedom line (dashed red) from the freedom ribbon set is coincident with the axis of the constraint ribbon set. Both freedom lines at the end of the freedom ribbon set and both constraint lines at the end of the constraint ribbon set are all asymptotically parallel and orthogonal to the two ribbons' axes.

Appendix E proves that both ribbons' pitch values are equivalent in magnitude but have opposite signs for corresponding locations along the two complementary ribbons' axes. It also shows that the double derivative of each ribbon's pitch with respect to position along the ribbon's axis results in a constant that is equal in magnitude but opposite in sign for complementary ribbons. This constant is a significant number since it contains all the information necessary for describing and characterizing an orthogonal ribbon. If a ribbon is right-handed, this constant will be a positive value. If a ribbon is left-handed, this constant will be a negative value.

An orthogonal constraint ribbon space may also be described by a single characteristic screw that is orthogonal to and intersects the ribbon's central constraint line as well as its axis as shown in **Figure 7.72**. Every constraint line on the ribbon's surface satisfies this characteristic screw according to **Equation (3.13)**. Note that every right-handed ribbon will be characterized by a screw with a negative pitch value and that every left-handed ribbon will be characterized by a screw with a positive pitch value.

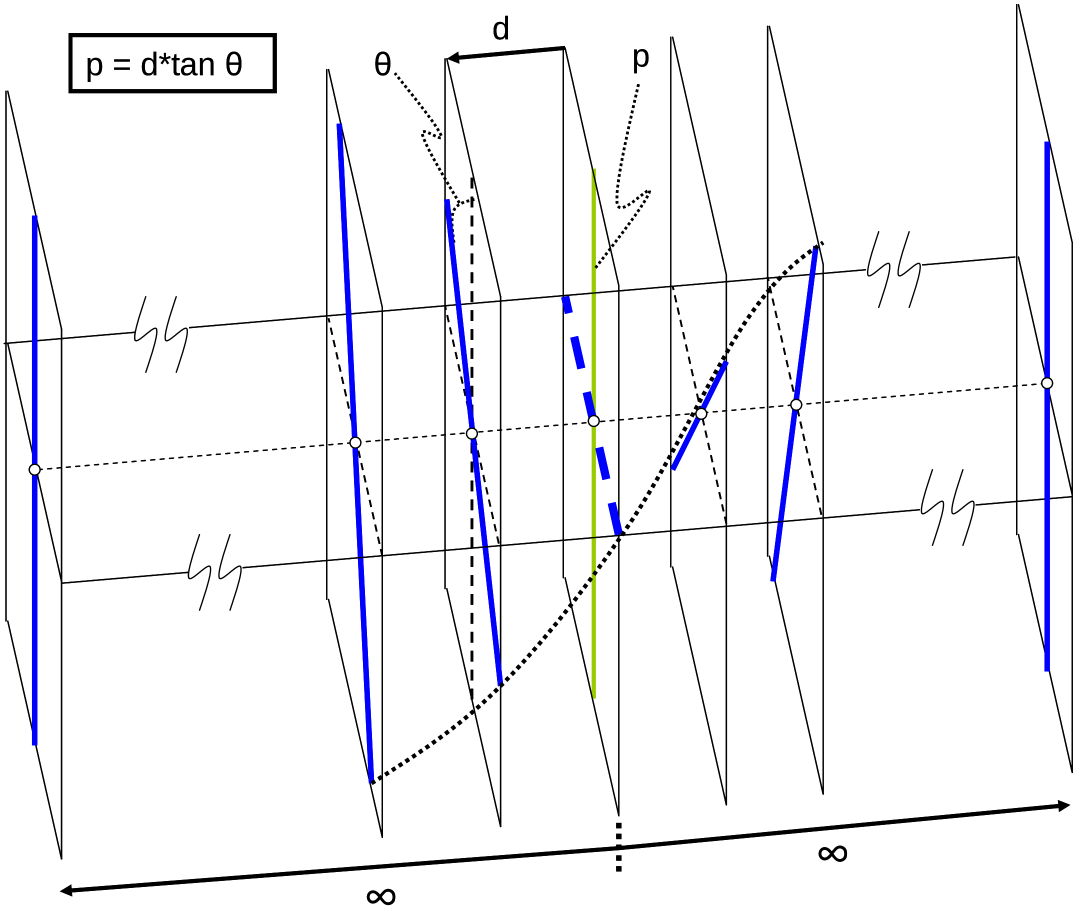


Figure 7.72: Characteristic screw (green) that defines an orthogonal constraint ribbon space made of constraint lines (blue)

Since the pitch, *p*, of this characteristic screw defines the rate that the constraint lines rotate as they translate along the ribbon's axis in the same way that the double derivative constant, *K*, of the ribbon's pitch characterizes the ribbon, one would expect these two characteristic values to be related. This relationship is given as

$$p = -\frac{2}{K}. \tag{7.2}$$

Equation (7.2) is proven in **Appendix F**. Both of these values, therefore, fully define any pair of complementary orthogonal ribbon spaces.

An important observation states that:

The surfaces that the lines within complementary freedom and constraint ribbon sets lie on are hyperbolic paraboloids.

This should not come as a surprise in light of the fact that hyperbolic paraboloids are doubly ruled surfaces as discussed in **Chapter 6**. From Blanding's Rule of Complementary Patterns one should expect doubly ruled surfaces to be the shapes that contain the lines within complementary freedom and constraint sets where one of the two rulings is a group of constraint lines and the other ruling is a group of pure rotational freedom lines. Every point on such surfaces will be intersected by a single freedom line and a single constraint line that both lay entirely on the surface. Every freedom line will intersect every constraint line and every constraint line will intersect every freedom line on the same surface.

An example of a pair of complementary ribbon sets are shown on the surface of a hyperbolic paraboloid in **Figure 7.73**. The characteristic screw of these ribbons is a line along the z-axis for the coordinate system shown in the figure. The thick dashed red line is the axis of the blue constraint ribbon and is the central line of the red freedom ribbon. The thick dashed blue line is the axis of the red freedom ribbon and is the central line of the blue constraint ribbon. Both thick dashed blue and red lines intersect at the origin. The two dotted black primary parabolas lie on orthogonal planes (x-z and y-z planes) and also intersect at the origin.

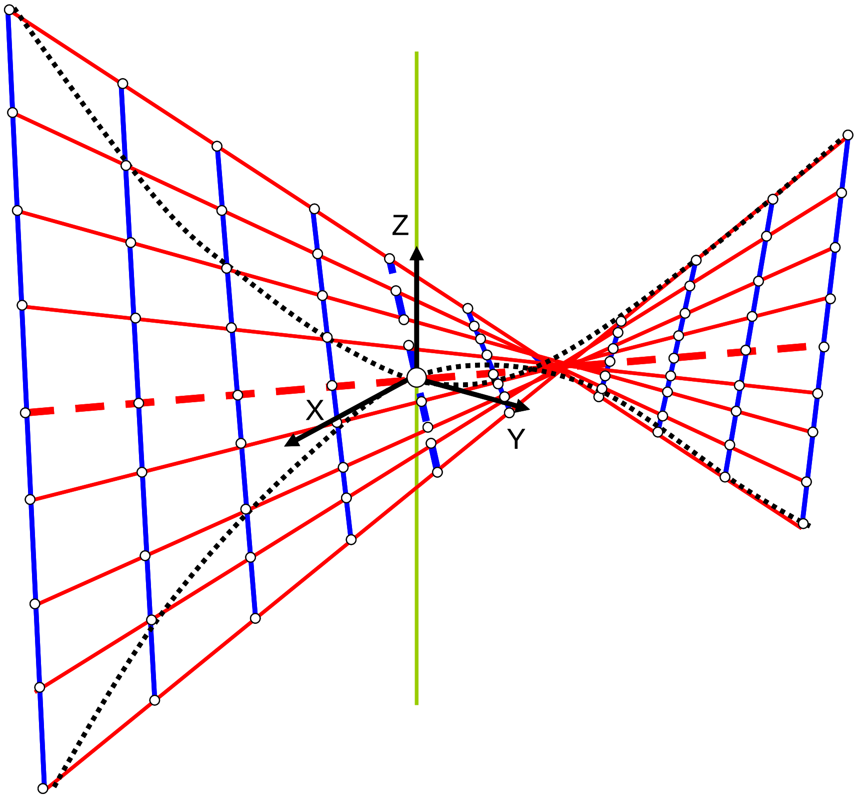


Figure 7.73: A pair of complementary freedom and constraint ribbon sets (red and blue respectively) that lie on the surface of a hyperbolic paraboloid with their characteristic screw (green).

The surface of this hyperbolic paraboloid is mathematically described using **Equation (6.1)**. **Appendix G** proves that hyperbolic paraboloids composed of orthogonal ribbon sets have equal *a* and *b* values and are described in terms of their characteristic screw's pitch, *p*, as

$$z = \frac{-x^2 + y^2}{2p}. \tag{7.3}$$

Before concluding the study of orthogonal freedom and constraint ribbon sets, the existence of a pure rotational hoop that exists within the freedom space of a system with three skew constraint lines that lie on parallel planes and fulfill the geometric requirements specified in **Figure 7.64** should be mentioned. This pure rotational hoop's normal vector will always point along the shortest distance line in a direction normal to the parallel planes of the skew constraint lines. Systems like those shown in **Figure 7.64** will always consist of complementary orthogonal ribbon freedom and constraint sets and will, therefore, always have a pure translation that points along the axis of the orthogonal constraint ribbon.

The study of three skew constraint lines that lie on three parallel planes is now ready to be extended to the case where the third constraint line does not intersect the shortest distance line of the other two skew constraint lines. This new system is shown in **Figure 7.74**. Ribbon spaces created within this type of system are non-orthogonal ribbons.

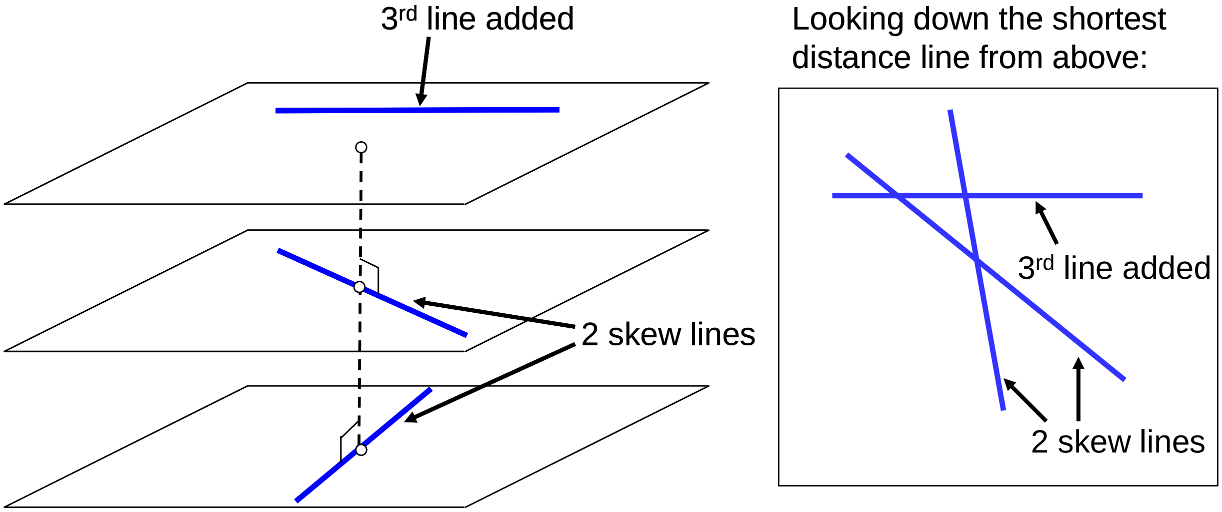


Figure 7.74: Third constraint line added to a plane parallel to the plane of the two skew constraint lines that does not intersected the shortest distance line (dashed black) of these two skew constraint lines.

Note from the right side of **Figure 7.74** that such systems' three skew constraint lines will not look like a disk of lines when viewed from above like was the case for systems of three skew constraint lines that create orthogonal ribbons shown in **Figure 7.64**.

If one follows the same procedures that were applied to the system from **Figure 7.64** for finding the pure rotational freedom lines and their complementary constraint lines for the current system of study shown in **Figure 7.74**, one finds that two complementary non-orthogonal freedom and constraint ribbon sets are created. A characteristic screw is orthogonal to and intersects both ribbons' axes at their center points and the lines within each of these complementary non-orthogonal ribbon sets all lay on the surface of a hyperbolic paraboloid.

Significant differences between the non-orthogonal ribbons created using the system from **Figure 7.74** and the orthogonal ribbons created using the system from **Figure 7.64** should be noted. The axes of the complementary non-orthogonal ribbons are not orthogonal. The group of parallel planes that each constraint line lies on as well as the group of parallel planes that each freedom line lies on are not orthogonal to their non-orthogonal ribbon's respective axis. The values for a and b from the hyperbolic paraboloid's equation given in **Chapter 6** as **Equation (6.1)** are not equal for non-orthogonal ribbons. The normal vector of the pure rotational hoop does not point along the axis of the non-orthogonal constraint ribbon set and the pitch of the characteristic screw cannot fully describe both complementary non-orthogonal ribbon spaces. The following paragraphs will demonstrate these observations.

First one must locate the constraint ribbon's axis for the system of three skew constraints given in **Figure 7.74**. This axis will be along the shortest possible line segment that intersects all three skew constraints. An example of a non-orthogonal ribbon's axis is shown in **Figure 7.75**.

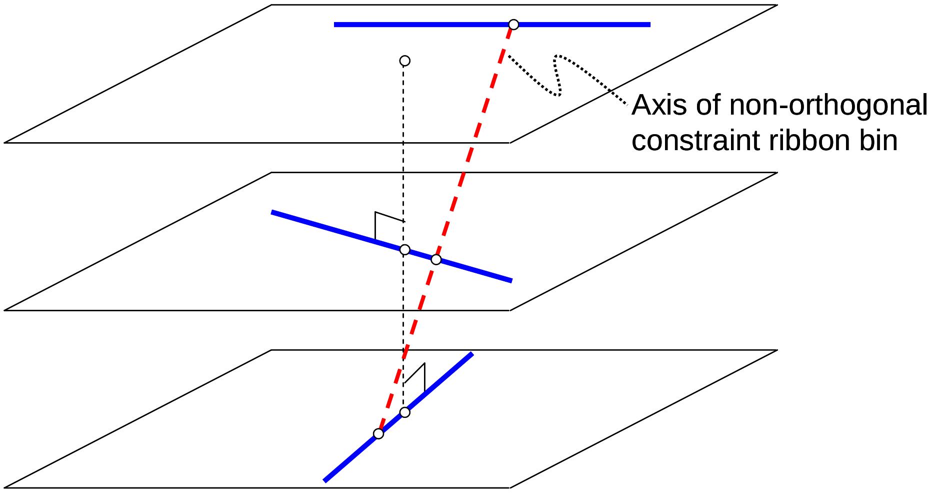


Figure 7.75: Axis (dashed red) of the non-orthogonal constraint ribbon set is the shortest line segment that intersects all three non-redundant skew constraint lines (blue)

Note that the three parallel planes from **Figure 7.75** will never be orthogonal to the non-orthogonal constraint ribbon set's axis. This axis is the central freedom line within the non-orthogonal freedom ribbon set. This central freedom line will never be orthogonal to the central constraint line within the non-orthogonal constraint ribbon set since the central constraint line will have to lie on a plane that is parallel to the three planes shown in **Figure 7.75**. Since the central constraint line is the axis of the non-orthogonal freedom ribbon set, the axes of these complementary non-orthogonal ribbon sets may never be orthogonal.

Suppose one were to view a hyperbolic paraboloid that consisted of orthogonal freedom and constraint ribbon sets from "above" such that one were looking down its characteristic screw and compared this view to a similar view of a hyperbolic paraboloid that consisted of non-orthogonal freedom and constraint ribbon sets. If both hyperbolic paraboloids' characteristic screws lied along the z-axis, one would see the freedom and constraint lines (red and blue respectively) as shown in **Figure 7.76**.

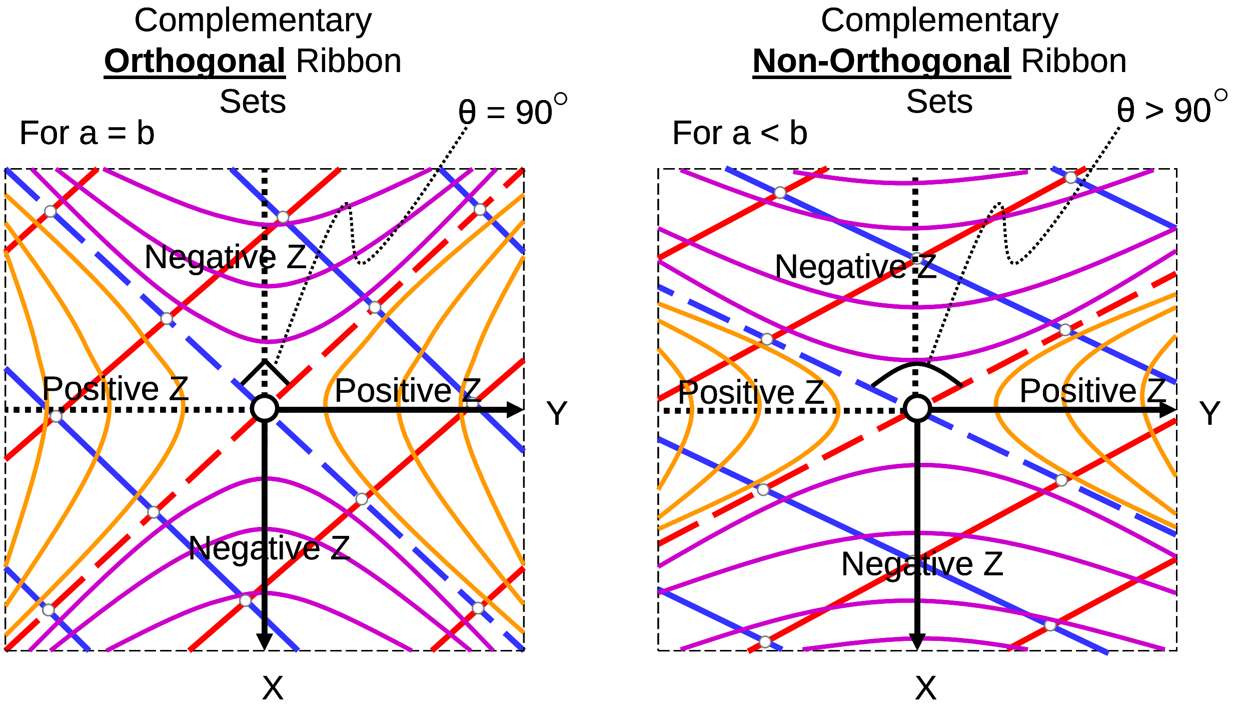


Figure 7.76: Looking down the characteristic screw (along z-axis) of a hyperbolic paraboloid similar to the one shown in Figure 7.73. The hyperbolic paraboloid on the left is composed of complementary orthogonal freedom (red) and constraint (blue) ribbon sets while the hyperbolic paraboloid on the right is composed of complementary non-orthogonal freedom (red) and constraint (blue) ribbon sets.

Section 6.1 of **Chapter 6** explains the meaning of the orange and purple hyperbolas shown in **Figure 7.76**. The dashed asymptotic lines represent the axes or central lines of the ribbons. Every freedom line (red) will appear to be orthogonal to every constraint line (blue) from the view of the hyperbolic paraboloid containing complementary orthogonal ribbon sets shown on the left side of **Figure 7.76**. This will not be the case for the freedom and constraint lines in the non-orthogonal ribbon sets shown on the right side of the figure. The angle, θ, shown on the right side of **Figure 7.76** will equal 180 degrees subtracted from the angle between the axis line shown in **Figure 7.75** and its projected line onto the parallel planes of the skew constraint lines.

It is also interesting to note from **Figure 7.76** that the hyperbolas in each of the four quadrants created by the asymptotic lines will all look identical for the case of complementary orthogonal ribbon sets. This makes sense since the two primary parabolas (dotted black) will rise and fall at the same rate since a=b. For the case of complementary non-orthogonal ribbon sets, the hyperbolas that lie in quadrants directly across from each other will look identical while the hyperbolas that lie in neighboring quadrants will always look different. This is due to the fact that the primary parabolas will rise and fall at different rates since a does not equal b for hyperbolic paraboloids that contain complementary non-orthogonal ribbon sets.

It is shown in **Appendix H** that the equation of a hyperbolic paraboloid that contains complementary non-orthogonal ribbon sets is given as **Equation (6.1)** where a does not equal b and where the characteristic screw's pitch, p, is expressed in terms of a and b as

$$p = \frac{ab}{2} . \qquad (7.4)$$

Note that for the case of non-orthogonal ribbon sets the characteristic screw's pitch is not enough to fully describe the hyperbolic paraboloid. Either a or b also must be provided or solved for from the original three skew constraint lines to completely describe the surface of the hyperbolic paraboloid.

It is also shown in **Appendix H** that the normal vector, $\vec{n}$, of the pure rotational hoop in the freedom space of the system of non-orthogonal ribbon sets will always point in the direction

$$\vec{n} = \begin{bmatrix} -b & a & 0 \end{bmatrix}. \qquad (7.5)$$

Equation (6.1), **Equation (7.4)** and **Equation (7.5)** are the general solutions for any hyperbolic paraboloid containing either orthogonal or non-orthogonal ribbon sets. If a=b in **Equation (7.4)**

and if the resulting characteristic screw's pitch is substituted into **Equation (6.1)**, **Equation (7.3)** is again proven for hyperbolic paraboloids that contain orthogonal ribbon sets. If *a*=*b*, **Equation (7.5)** suggests that the pure translation in the freedom space for a system of orthogonal ribbon sets will point along the axis of the orthogonal constraint ribbon set which was previously confirmed to be true.

7.3.3.1.1 Case 3, Type 7:

This section describes the freedom and constraint space of Case 3, Type 7. The complete constraint space of this type is shown in **Figure 7.77**. It consists of constraint lines that lie on the surface of a hyperbolic paraboloid and is, therefore, described by **Equation (6.1)**. The constraint space includes orthogonal and non-orthogonal ribbons as well as right- and left-handed ribbons.

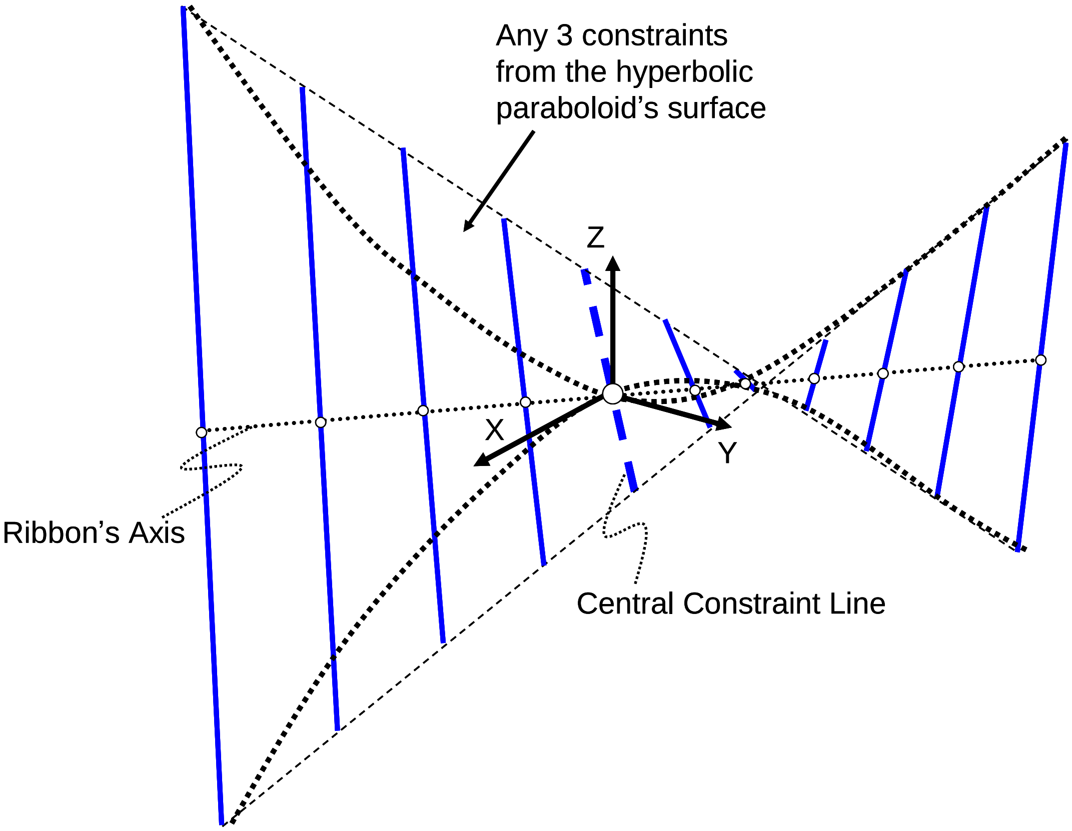


Figure 7.77: Constraint space of Case 3, Type 7

Note the instructions to the designer for selecting three non-redundant constraints from the ribbon. Any additional constraints selected will be redundant.

Figure 7.78 shows every pure rotational freedom line within the freedom space of Case 3, Type 7. These lines produce a single freedom ribbon set as well as a single pure rotational hoop. The freedom ribbon set will be the complementary ribbon to the constraint space ribbon and they will both lie on the same hyperbolic paraboloid. This freedom ribbon set will be an orthogonal ribbon if the constraint space is an orthogonal ribbon and it will be a non-orthogonal ribbon if the constraint space is a non-orthogonal ribbon. The freedom ribbon set will be a left-handed ribbon if the constraint space is a right-handed ribbon and it will be a right-handed ribbon if the constraint space is a left-handed ribbon. The normal vector of the pure rotational hoop points in the direction given in **Equation (7.5)**.

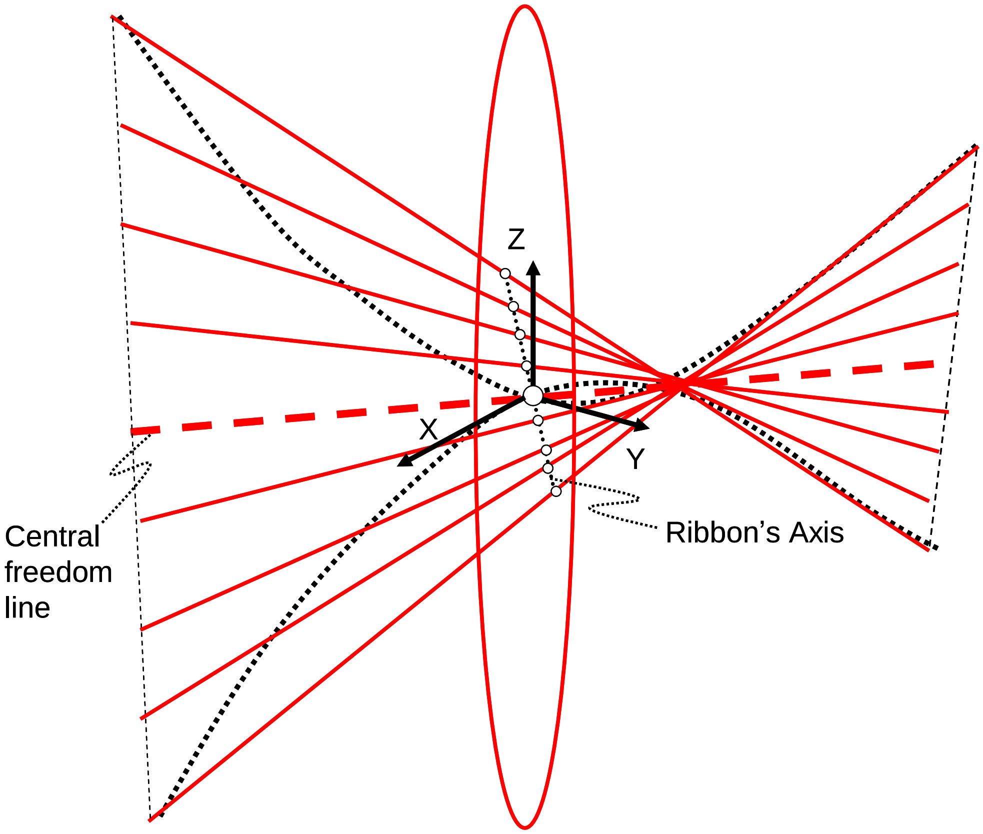


Figure 7.78: Pure rotational freedom sets within the freedom space of Case 3, Type 7

The twists shown in **Figure 7.78** do not fully describe the freedom space of the system. Screws with finite, non-zero pitch values also exist that have not yet been mentioned. The characteristic screw that passes through the z-axis is only one of infinite screws that exist within the freedom space of this case and type. The system's screws only lie on the parallel planes of the skew pure rotational freedom lines regardless of whether the complementary ribbon sets are orthogonal or not. If the ribbon sets are orthogonal, however, the screws will exist not only on the parallel planes of the skew pure rotational freedom lines, but they will only exist within disks on these planes with center points that lie on the axis of the freedom ribbon set. One of the twists within each of these disks along the freedom ribbon set's axis will be a pure rotational freedom line that belongs to the freedom ribbon set, one will be a pure translational twist that points in the direction of the normal vector of the pure rotational hoop, and the rest will be screws with varying pitch values that depend on the location of the twist within the disk. This is shown and proven in **Appendix I**.

A depiction of the complete freedom space of Case 3, Type 7 is shown in **Figure 7.79** for the case of orthogonal complementary ribbon sets. Only one disk of twists is shown to prevent cluttering the figure on the right. The disk contains a single freedom line (red), a single pure translation line (black), and an infinite number of screw lines (green). The dotted black line represents the freedom ribbon set's axis, which is orthogonal to the plane of the disk and intersects it at its center. If the complementary ribbon sets were not orthogonal ribbon sets, the plane of twists depicted in **Figure 7.79** would not be confined to a disk of twists nor would the plane be orthogonal to the freedom ribbon set's axis. Screws with finite, non-zero pitch values would cover the plane. The location of these screws and their pitch values may be determined using the mathematical approach discussed in **Chapter 3** of **Section 3.4.2**.

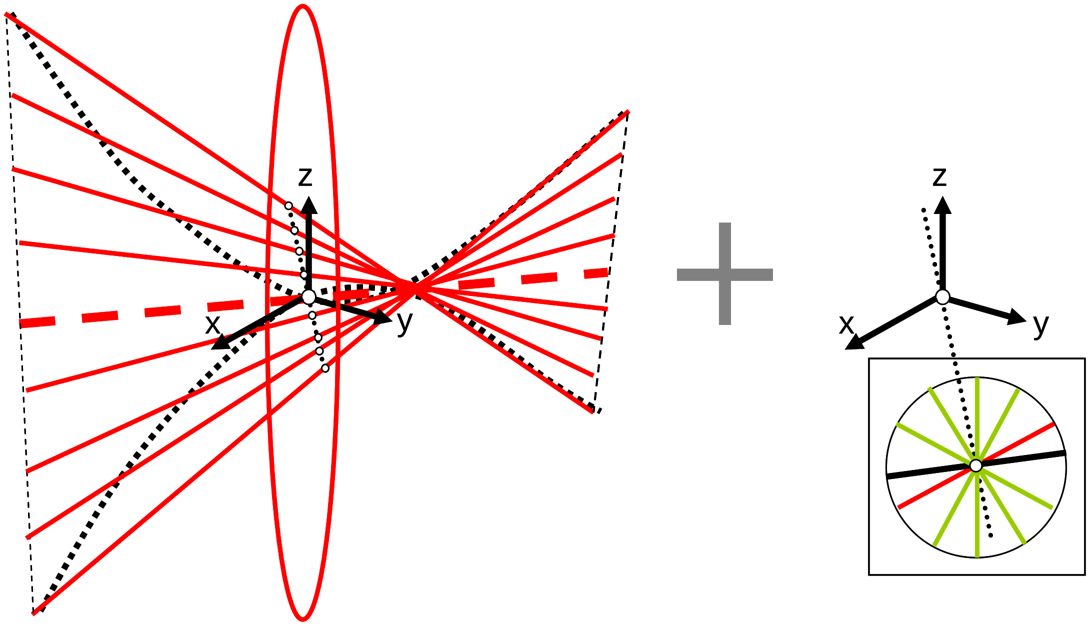


Figure 7.79: Freedom space of Case 3, Type 7 for the case of an orthogonal freedom ribbon set

Figure 7.80 shows how the freedom and constraint spaces of Case 3, Type 7 fit together. The screws with finite, non-zero pitch values are not shown to avoid cluttering the figure.

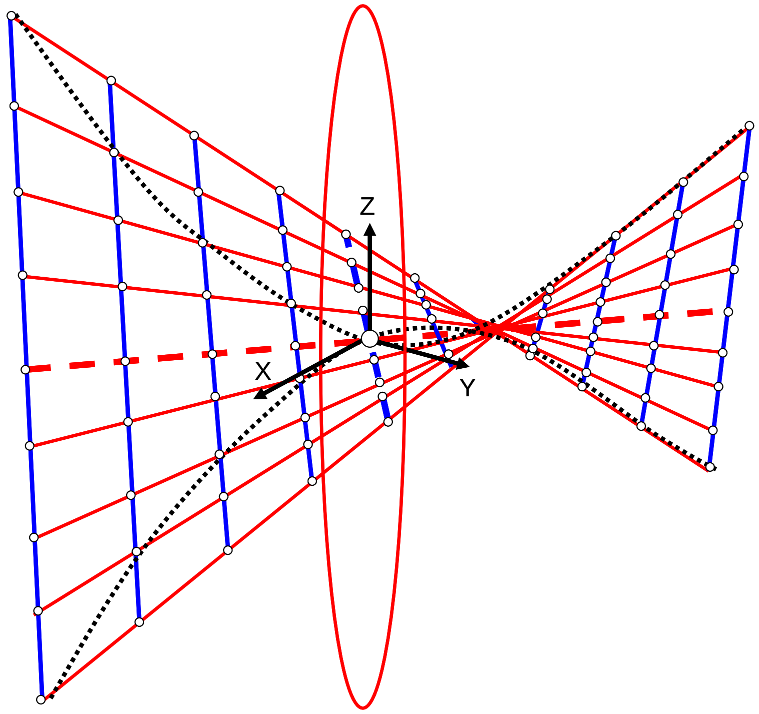


Figure 7.80: Freedom space (red) and constraint space (blue) of Case 3, Type 7 together (without the screws shown).

7.3.3.2 Skew Line Intersects the Parallel Planes of Two Other Skew Lines

This section will examine the system of a third constraint line that intersects the parallel planes of two other skew constraint lines. This system is shown in **Figure 7.63** and represents the last possible way three non-redundant constraint lines could be combined to produce new freedom spaces. After studying this system, Case 3, Type8 and Case 3, Type 9 will be identified and mathematically described.

To begin the study of this system, one must consider an important observation:

“Three skew lines always define a one-sheeted hyperboloid, except in the case where they are all parallel to a single plane but not to each other. In this case, they determine a hyperbolic paraboloid. [39]”

The truth of the second half of this quotation was validated in the previous section, but it is the first half of the statement that is of interest for this section. The quote essentially states that any three constraint lines that are arrange like the system shown in **Figure 7.63** will lie on the surface of a hyperboloid constraint space.

If one wishes to find all the pure rotational freedom lines for such a system, one must apply the Rule of Complementary Patterns to locate every line that intersects every constraint line within the hyperboloid constraint set. The solution was already given in **Chapter 6**. Recall that every hyperboloid is a doubly ruled surface with two rulings or groups of lines that lie entirely on its surface. One group of lines are the constraint lines and the other group of lines are the freedom lines as shown in **Figure 7.81**. Every line within either group of lines will intersect every line within the other group of lines in finite space or at infinity. The only pure rotational freedom set that exists for such a system is, therefore, the hyperboloid freedom set that is identical in shape and size to its complementary hyperboloid constraint set. The freedom lines on the hyperboloid’s surface are, however, a different ruling or group of lines than the constraint lines also on its surface.

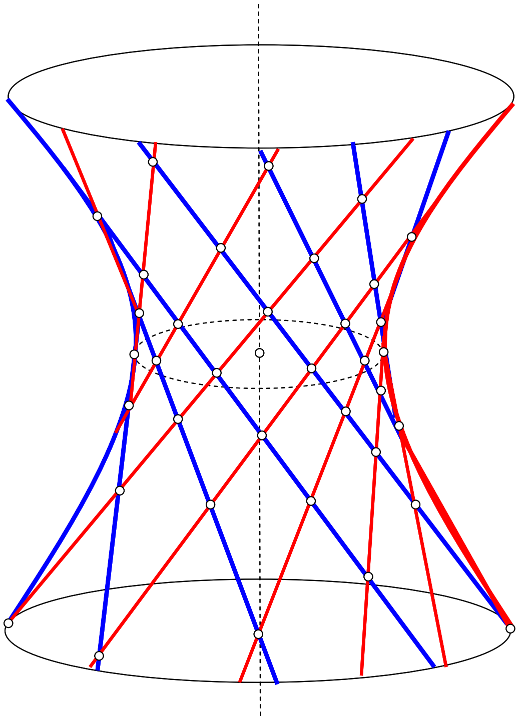


Figure 7.81: Hyperboloid with a ruling of constraint lines (blue) and a ruling of freedom lines (red)

At this point, the significance of doubly ruled surfaces in identifying freedom and constraint spaces should be apparent. One would, in fact, expect every doubly ruled surface to be some type of complementary freedom and constraint space since doubly ruled surfaces always contain two rulings of lines that all intersect each other at least once and thus satisfy Blanding's Rule of Complementary Patterns. Consequently, only two doubly ruled surfaces exist, hyperbolic paraboloids and hyperboloids [40]. It is interesting to note that both of these surfaces are the surfaces that contain the complementary freedom and constraint sets for any system that contains three or more skew constraint lines.

Recall from **Chapter 6** that only two types of hyperboloids exist, circular and elliptical. Every system composed of three skew constraint lines like the ones shown in **Figure 7.63** will, therefore, lie on the surface of one of these types of hyperboloids. A closer look at each of these hyperboloids will now be taken to learn what kind of skew constraint lines will lie on the surface of which hyperboloid type.

Recall from **Chapter 6** that every circular hyperboloid has an axis line running through its center as shown in **Figure 7.82**. The ruled surface of such a hyperboloid will result when a single constraint line that is perpendicular to a line that is also perpendicular to this axis line is rotated about the axis line as shown in the figure. Every constraint line that lies on the hyperboloid's surface will, therefore, lie an equal distance, *L*, away from this axis line. The angle between the constraint line and a line that is tangent to the hyperboloid's central circular cross-section is defined as *α* and is shown in **Figure 7.82**. Every constraint line on the surface of the circular hyperboloid will have equal *α* angles. If an axis line may be found in the midst of three skew constraint lines such that the shortest distance segments between this axis and the three skew constraint lines will be of equal length, *L*, and all lie on the same plane, and if the three skew constraint lines create equal *α* angles with this plane, then the three skew constraint lines will lie on the surface a circular hyperboloid constraint space. In every other instance of skew constraint lines arranged like those shown in **Figure 7.63**, the constraint space will be an elliptical hyperboloid.

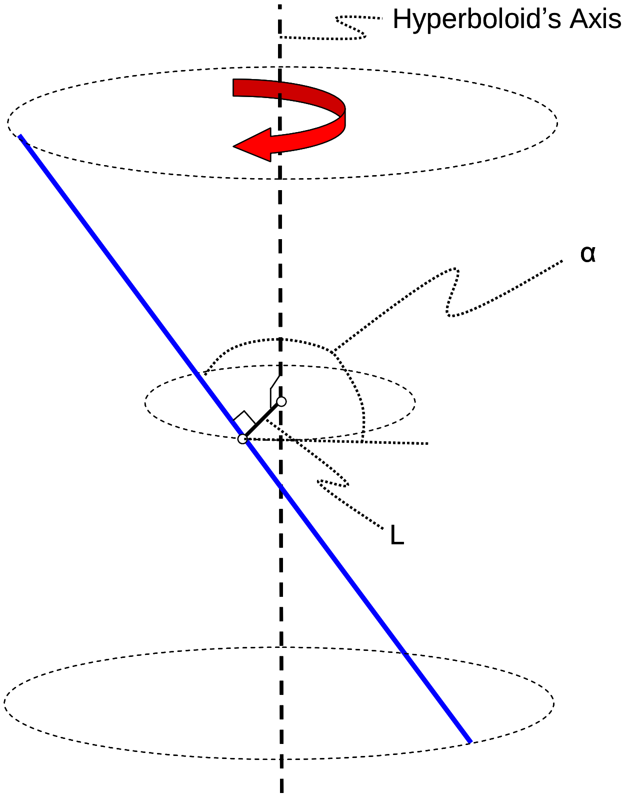


Figure 7.82: Parameter's necessary for defining a circular hyperboloid with respect to a single constraint line (blue) on its surface.

Hyperboloids that have α angles between zero and 90 degrees are called left-handed hyperboloids. Hyperboloids that have α angles between 90 and 180 degrees are called right-handed hyperboloids. **Chapter 8** will explain the reasoning behind this convention. A pair of left- and right-handed circular hyperboloids is shown in **Figure 7.83** (If α equaled zero, 90, or 180 degrees, the lines would no longer lie on the surface of a hyperboloid).

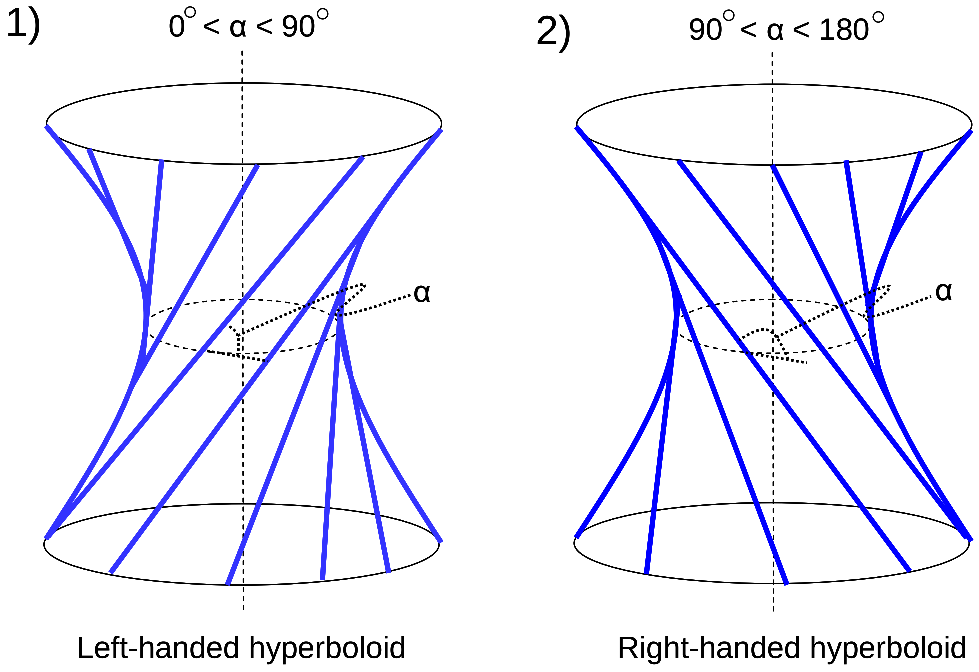


Figure 7.83: Condition for distinguishing between left-handed hyperboloids and right-handed hyperboloids

Note also that if a constraint set is a left-handed hyperboloid, its complementary freedom set will be a right-handed hyperboloid. If a constraint set is a right-handed hyperboloid, its complementary freedom set will be a left-handed hyperboloid. This is true for both circular and elliptical hyperboloid freedom and constraint sets.

The equation for a circular hyperboloid in terms of the parameters given in **Figure 7.82** is

$$\frac{x^2+y^2}{L^2}-\frac{z^2}{(L\tan\alpha)^2}=1, \tag{7.6}$$

where the hyperboloid's axis lies along the z-axis. This equation is proven in **Appendix J**.

Parameters that fully characterize an elliptical hyperboloid are shown in **Figure 7.84**. The major and minor axes of the hyperboloid's central elliptical cross-section are orthogonal to two constraint lines that are both skew to the hyperboloid's axis. The length of the line segment along the major axis is *a* while the length of the line segment along the minor axis is *b*. The angles, α_1 and α_2, are the angles between the constraint lines and the lines that are tangent to the hyperboloid's central elliptical cross-section as shown in the figure. Only when α_1 equals α_2 and *a* equals *b* will the hyperboloid be circular. In any other instance, it will be elliptical.

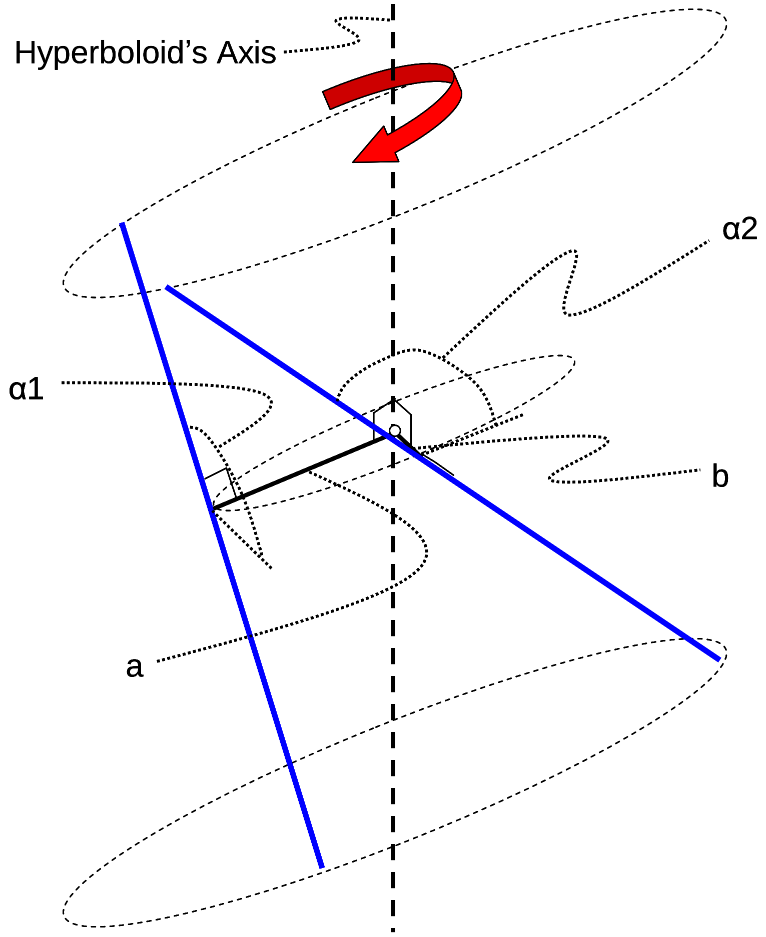


Figure 7.84: Parameter's necessary for defining an elliptical hyperboloid with respect to two constraint lines (blue) that lie on its surface.

Two possible equations for defining an elliptical hyperboloid in terms of the parameters shown in **Figure 7.84** are

$$\frac{x^2}{a^2}+\frac{y^2}{b^2}-\frac{z^2}{(b\tan\alpha_1)^2}=1, \tag{7.7}$$

and

$$\frac{x^2}{a^2}+\frac{y^2}{b^2}-\frac{z^2}{(a\tan\alpha_2)^2}=1, \tag{7.8}$$

where the major axis of the central elliptical cross-section lies along the x-axis, the minor axis of the central elliptical cross-section lies along the y-axis, and the hyperboloid's axis lies along the z-axis. These equations are proven in **Appendix K**. Note that only three of the four parameters shown in **Figure 7.84** are necessary for fully defining an elliptical hyperboloid.

7.3.3.2.1 Case 3, Type 8:

This section describes the freedom and constraint space of Case 3, Type 8. The complete constraint space of this type is shown in **Figure 7.85**. It consists of a single circular hyperboloid constraint set that could either be right-handed or left-handed. It is mathematically described using **Equation (7.6)** for any real values of *L* and *α*.

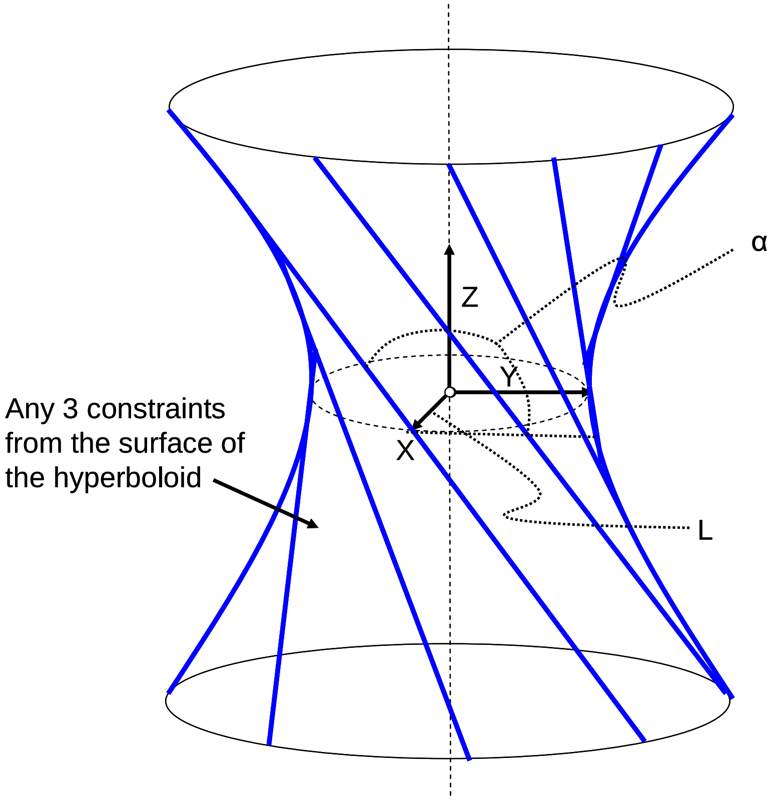


Figure 7.85: Constraint space of Case 3, Type 8

Note the instruction to the designer for selecting non-redundant constraints from the hyperboloid. Any constraint selected after the first three will be redundant. Also note that if L=0, this case and type will become Case 3, Type 4. If α=0 or 180 degrees, this case and type will become Case 3, Type 1. If α=90 degrees, this case and type will become Case 3, Type 5. If L=0 and α=0 or 180 degrees, this case and type will become Case 2, Type 1. If L=0 and α=90 degrees, this case and type will become Case 1, Type 1.

The only pure rotational freedom set within the freedom space of Case 3, Type 8 is shown in **Figure 7.86**. It is a circular hyperboloid that is mathematically described by **Equation (7.6)** with an L parameter equal to the L parameter of the circular hyperboloid constraint set. The circular hyperboloid freedom set's α angle will, however, cause tan(α) to be equal in magnitude but opposite in sign to the tan(α) of the α angle of the circular hyperboloid constraint set. In other words, the circular hyperboloid freedom set will be right-handed if the circular hyperboloid constraint set is left-handed and the circular hyperboloid freedom set will be left-handed if the circular hyperboloid constraint set is right-handed.

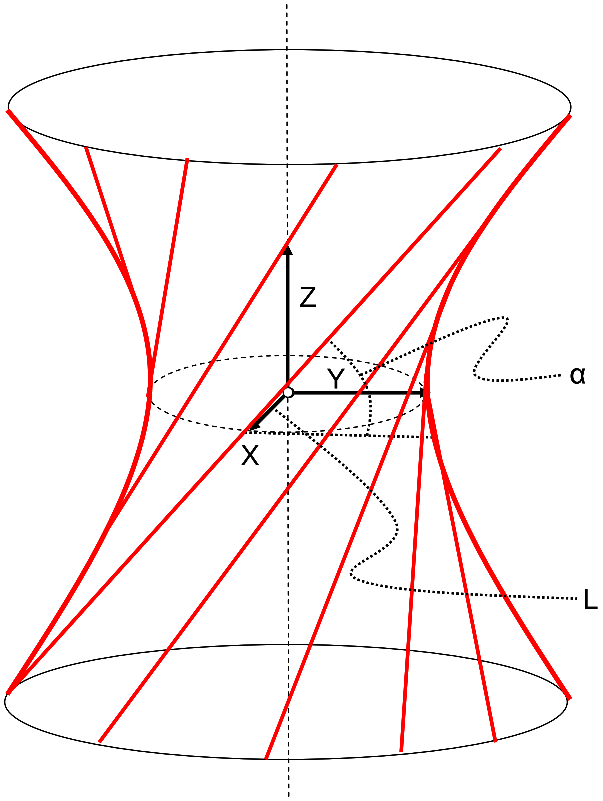


Figure 7.86: Freedom space of Case 3, Type 8 without screws

No pure translations exist within the freedom space of Case 3, Type 8, but an infinite number of screws do exist. Visually representing these screws is, however, extremely difficult and will not be done in this thesis. The system's screws may be located using the mathematical method given in **Chapter 3** described in **Section 3.4.2**. One screw, however, is worth mentioning and it lies along the axis of the circular hyperboloid pure rotational freedom set.

Figure 7.87 shows how the freedom and constraint spaces of Case 3, Type 8 fit together. The screws with finite, non-zero pitch values are not shown to avoid cluttering the figure.

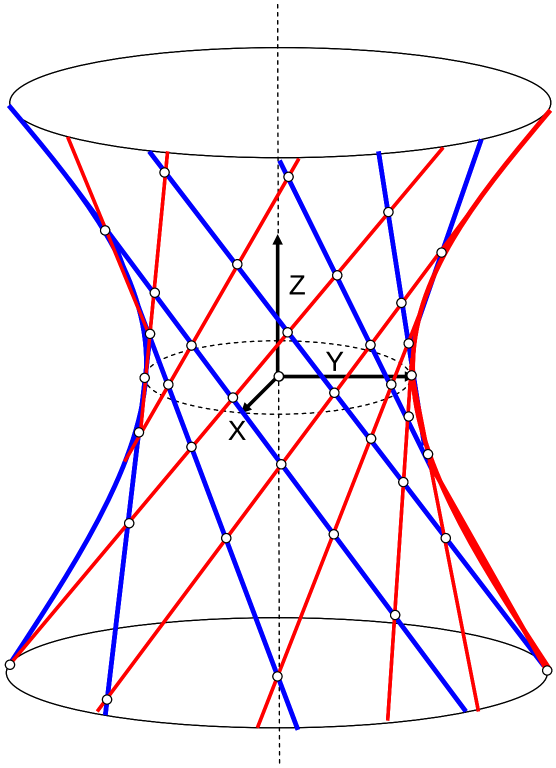


Figure 7.87: Freedom space (red) and constraint space (blue) of Case 3, Type 8 together (without the screws shown).

7.3.3.2.2 Case 3, Type 9:

This section describes the freedom and constraint space of Case 3, Type 9. The complete constraint space of this type is shown in **Figure 7.88**. It consists of a single elliptical hyperboloid constraint set that could either be right-handed or left-handed. It is mathematically described using either **Equation (7.7)** or **Equation (7.8)** for any real values of a, b, α_1, and α_2.

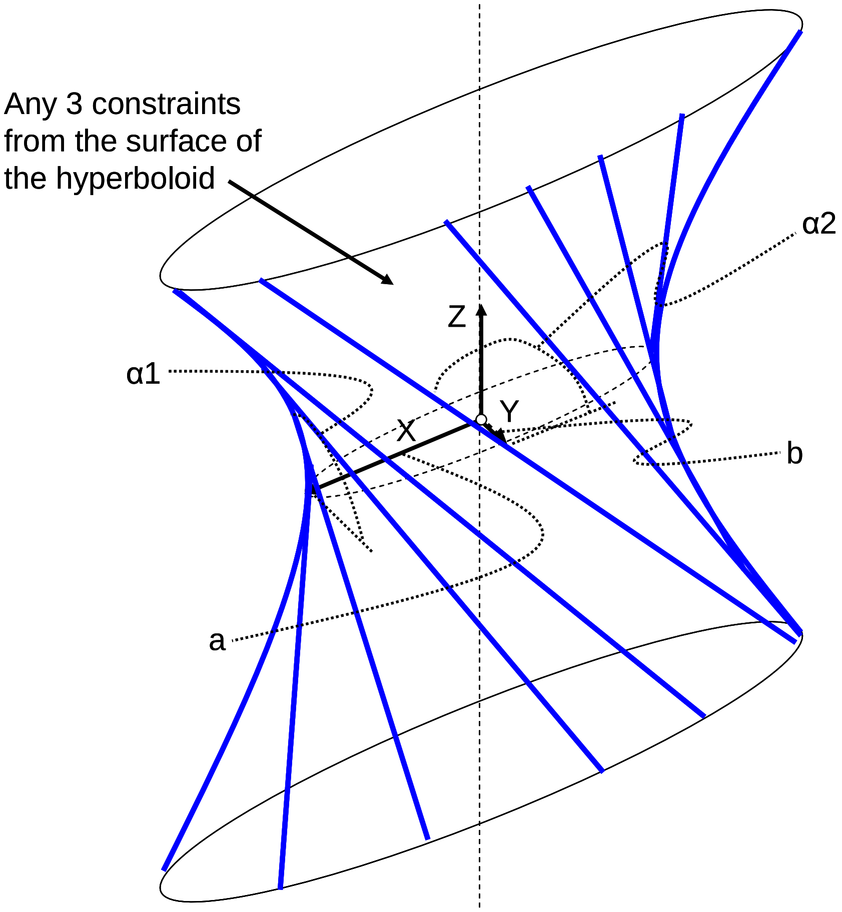


Figure 7.88: Constraint space of Case 3, Type 9

Note the instruction to the designer for selecting non-redundant constraints from the hyperboloid. Any constraint selected after the first three will be redundant. Also note that if *a*=0 and *b*=0, this case and type will become Case 3, Type 4. If α_1=0 or 180 and α_2=0 or 180 degrees, this case and type will become Case 3, Type 1. If α_1=90 degrees and α_2=90 degrees, this case and type will become Case 3, Type 5. If *a*=0, *b*=0, α_1=0 or 180 degrees and α_2=0 or 180 degrees, this case and type will become Case 2, Type 1. If *a*=0, *b*=0, α_1=90 degrees and α_2=90 degrees, this case and type will become Case 1, Type 1. It has also already been mentioned that if *a*=*b* and $\alpha_1 = \alpha_2$, this case and type will become Case 3, Type 8.

The only pure rotational freedom set within the freedom space of Case 3, Type 9 is shown in **Figure 7.89**. It is also an elliptical hyperboloid that is mathematically described using either

Equation (7.7) or **Equation (7.8)** with *a* and *b* parameters that are equivalent to the *a* and *b* parameters of the elliptical hyperboloid constraint set. The elliptical hyperboloid freedom set's α_1 angle will cause tan(α_1) to be equal in magnitude but opposite in sign to the tan(α_1) of the α_1 angle of the elliptical hyperboloid constraint set, and the elliptical hyperboloid freedom set's α_2 angle will cause tan(α_2) to be equal in magnitude but opposite in sign to the tan(α_2) of the α_2 angle of the elliptical hyperboloid constraint set. In other words, the elliptical hyperboloid freedom set will be right-handed if the elliptical hyperboloid constraint set is left-handed and the elliptical hyperboloid freedom set will be left-handed if the elliptical hyperboloid constraint set is right-handed.

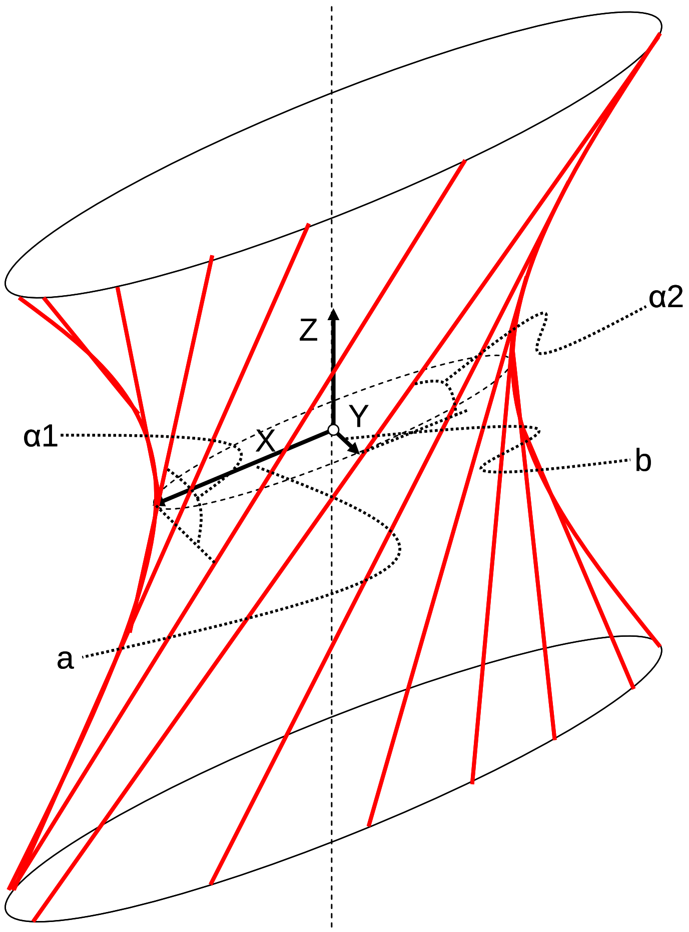


Figure 7.89: Freedom space of Case 3, Type 9 without screws

No pure translations exist within the freedom space of Case 3, Type 9, but an infinite number of screws do exist. Visually representing these screws is, however, extremely difficult and will not be done in this thesis. The system's screws may be located using the mathematical method given in **Chapter 3** described in **Section 3.4.2**.

Figure 7.90 shows how the freedom and constraint spaces of Case 3, Type 9 fit together. The screws with finite, non-zero pitch values are not shown to avoid cluttering the figure.

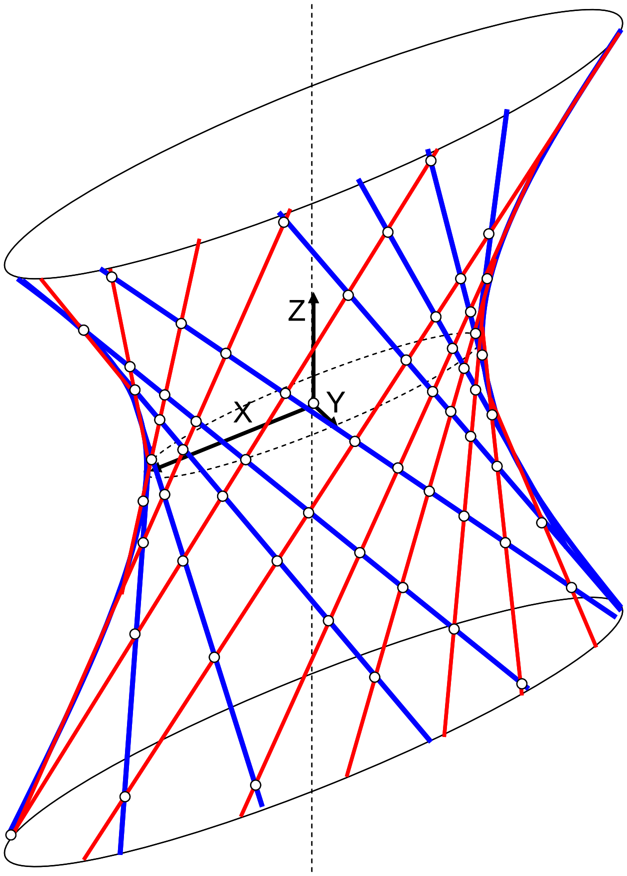


Figure 7.90: Freedom space (red) and constraint space (blue) of Case 3, Type 9 together (without the screws shown).

CHAPTER 8:

“Cases 4, 5, and 6”

This chapter describes and validates every constraint space with its unique freedom space within the last three cases—cases 4, 5, and 6. The reader may recall from the final section of **Chapter 5** that there are 6 total cases where the case of a system corresponds to the number of non-redundant constraints in that system. In **Chapter 7** it was shown that the number of types a case has is the number of freedom and constraint space pairs within that case, or the number of different ways the non-redundant constraints may be arranged within the system to produce fundamentally different freedom spaces. Although these principles still apply for the cases presented in this chapter, a different approach will be used for finding each type’s freedom and constraint space pairs.

The reader may wonder why the last three cases are presented in a different chapter than the first three cases. The reason for this is largely because the first three cases have types that consist of “small” constraint spaces with “large” freedom spaces and the last three cases have types that consist of “large” constraint spaces with “small” freedom spaces. (A “small” space is a space with fewer sets that contain fewer infinite lines than “large” spaces have. An infinite planar set of parallel lines, for example, contains fewer lines than an infinite box set containing all parallel lines in three-space.) This observation is not surprising since the last three cases will always have types with constraint spaces that consist of more independent wrenches than the number of independent twists that create their unique freedom spaces. One can easily prove this fact using **Equation (2.1)**. Although, one would expect the types within Case 3 to have the same number of independent wrenches as independent twists, one must remember that the constraint spaces within Case 3 will be smaller than the freedom spaces within Case 3 since the constraint spaces consist only of wrenches with *q* values equal to zero while their freedom spaces consists of twists with any pitch values. The types within Case 1 and Case 2 will have smaller constraint spaces than freedom spaces in light of **Equation (2.1)**. The reader can also visually confirm these facts by noting how much less complicated the constraint spaces from **Chapter 7** are from their freedom spaces. In this chapter the opposite will be found to be true.

Since the freedom spaces within the types of Cases 4 through 6 are much less complicated than their constraint spaces, it makes sense that the pairs of freedom and constraint spaces should be found by starting first with every possible combination of their independent twist lines instead of starting with every possible combination of their non-redundant constraint lines like was done in **Chapter 7** for Cases 1 through 3. This fundamentally different approach to finding the freedom and constraint space pairs within the last three cases is another reason why they belong in a separate chapter from the first three cases.

Before beginning the study of these last three cases, the obvious should first be pointed out. Only one type exists within Case 6. This type consists of an empty freedom space that contains no twist lines of any kind. This statement is known to be true since an object constrained by 6 non-redundant constraints is fixed and does not move. One can also deduce that Case 5 contains only three types. Each type consists of a single twist line that is either a pure rotational freedom line, a non-zero finite pitch screw line, or a pure translational line. No other fundamentally different freedom spaces exist that consist of a single independent twist vector. The number of types within Case 4 is, however, not obvious at all. Finding and describing the freedom and constraint space pairs within Case 4 will be a significant portion of this chapter.

8.1 Finding Case 4 Freedom Spaces

This section proves that there are only 10 possible ways to combine two independent twists to create freedom spaces for systems with four non-redundant constraints. It will be shown that only 9 of these freedom spaces are possible spaces for flexure systems with constraints capable only of providing axial forces (q=0). This section will also describe the geometry of these freedom spaces.

To begin the proof, first recall that only three different types of twists exist to combine: (1) pure rotational freedom lines (p=0), (2) screw lines with finite non-zero pitch values, and (3) pure translational lines (p=∞). Furthermore, only four different ways exist for combining two lines with respect to each other: (1) they may be coincident, (2) they may be parallel, (3) they may intersect at a single point in finite space, or (4) they may be skew.

First, the freedom spaces that result from linearly combining any two twists that are either coincident or parallel with respect to each other will be determined.

8.1.1 Coincident and Parallel Pairs of Twists

This section mathematically proves and describes all the freedom spaces that result from linearly combining two general twists that are either coincident or parallel with respect to each other. **Figure 8.1** defines the parameters of two such parallel twist lines separated by a distance of d. If d=0, the twists are coincident. Depending on the values assigned to the different pitches, the two twists could either be pure rotations, screws, or pure translations.

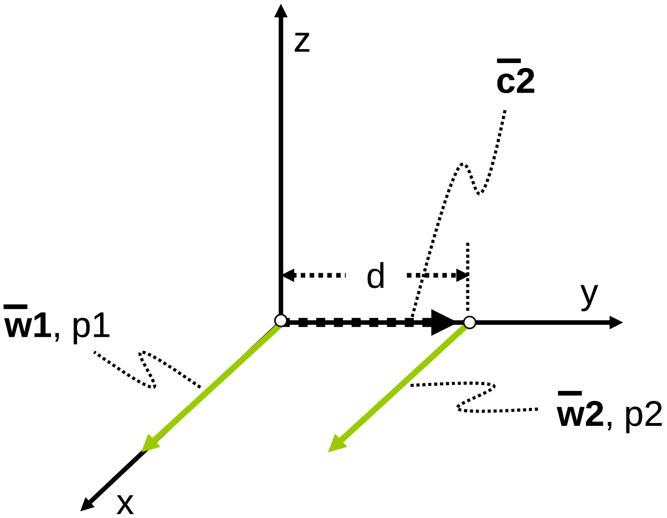


Figure 8.7: Parameters defined for two parallel twists

Using **Figure 8.1** and the principles discussed in **Chapter 3**, the two twists are defined as

$$
\begin{aligned}
\vec{T}_1 &= \begin{bmatrix} 1 & 0 & 0 & p_1 & 0 & 0 \end{bmatrix} \\
\vec{T}_2 &= \begin{bmatrix} 1 & 0 & 0 & p_2 & 0 & -d \end{bmatrix}.
\end{aligned}
\tag{8.1}
$$

The general freedom space that results from the linear combination of these two independent twists is, therefore, mathematically represented as

$$\vec{T} = A\vec{T}_1 + B\vec{T}_2 = \left[(A+B) \quad 0 \quad 0 \quad (Ap_1 + Bp_2) \quad 0 \quad -dB \right], \tag{8.2}$$

where *A* and *B* are any real numbers. **Equation (8.2)** suggests that every twist within the general freedom space has a rotational velocity vector, $\vec{w}$, and a translational velocity vector, $\vec{v}$, of

$$\begin{aligned} \vec{w} &= \left[(A+B) \quad 0 \quad 0 \right] \\ \vec{v} &= \left[(Ap_1 + Bp_2) \quad 0 \quad -dB \right]. \end{aligned} \tag{8.3}$$

If **Equation (8.3)** is plugged into **Equation (3.4)**, one finds that every twist within the general freedom space will have a pitch value, *p*, of

$$p = \frac{Ap_1 + Bp_2}{A+B}. \tag{8.4}$$

If one applies **Equation (8.3)** and **Equation (8.4)** to the location matrix equation given in **Equation (3.8)** to find the location vector, $\vec{c}$, of the twists within the general freedom space, one finds that

$$\vec{c} = \begin{bmatrix} c_x & \dfrac{dB}{(A+B)} & 0 \end{bmatrix}, \tag{8.5}$$

where c_x may be any real value. **Equation (8.5)** only applies to twists that have location vectors that are not pure translations. In other words *A*+*B* cannot equal zero for **Equation (8.5)** to exist.

These equations will now be applied for determining the freedom spaces that exist when the two twist lines are coincident.

8.1.1.1 Coincident Pairs of Twists

This section mathematically proves and describes all the freedom spaces that result from linearly combining two general twists that are coincident. If *d*=0 in all of the previous equations, they will all be applicable equations for describing the twists that result from a linear combination of two coincident twist lines. To see that this is true refer to **Figure 8.1**.

Equation (8.5) suggests that every resulting twist within the freedom space of two coincident twist lines (where *d*=0) will pass through the origin. Also **Equation (8.3)** suggests that every resulting twist line will have an orientation vector, $\vec{w}$, that also points along the x-axis in the same direction as the original coincident twist lines. In short, every resulting twist line will also be coincident with the original two coincident twist lines.

If both coincident twists are pure rotations ($p_1 = p_2 = 0$), **Equation (8.4)** suggests that every resulting twist will also be a pure rotation with *p*=0. The freedom space resulting from these two twists is, therefore, a single pure rotational freedom line that belongs to Case 5 since these two twists are really not independent.

If both coincident twists are screws with equal pitch values ($p_1 = p_2 \neq 0$) that are non-zero and finite, **Equation (8.4)** suggests that every resulting twist will also be a screw with a pitch value equal to the original two screw pitch values ($p_1 = p_2 = p$). The freedom space resulting from these two twists is, therefore, a single screw with a non-zero finite pitch value that belongs to Case 5 since these two twists are really not independent.

If both coincident twists are pure translations with infinite pitch values such that

$$\begin{aligned}\vec{T}_1 &= \begin{bmatrix}0 & 0 & 0 & 1 & 0 & 0\end{bmatrix} \\ \vec{T}_2 &= \begin{bmatrix}0 & 0 & 0 & 1 & 0 & 0\end{bmatrix},\end{aligned} \tag{8.6}$$

the linear combination of these two pure translations must result in twists that are also pure translations that point in the same direction along the x-axis. This is clear by inspection of the resultant twist, $\vec{T}$, given as

$$\vec{T} = A\vec{T}_1 + B\vec{T}_2 = \begin{bmatrix} 0 & 0 & 0 & (A+B) & 0 & 0 \end{bmatrix}. \tag{8.7}$$

The freedom space resulting from these two twists is, therefore, a single pure translation that belongs to Case 5 since these two twists are really not independent.

If one of the coincident twists is a pure rotation ($p_1 = 0$) and the other is a screw with a non-zero finite pitch value ($p_2 \neq 0$), **Equation (8.4)** suggests that the resulting twists within the freedom space will be coincident twists with pitch values, *p*, of

$$p = \frac{Bp_2}{A+B}. \tag{8.8}$$

Or if both coincident twists are screws with pitch values that don't equal each other ($p_1 \neq p_2 \neq 0$) but are non-zero and finite, the resulting twists within the freedom space will be coincident twists with pitch values, *p*, given in **Equation (8.4)**.

If one of the coincident twists is a pure rotation ($p_1 = 0$), and the other is a pure translation ($p_2 = \infty$) such that

$$\begin{aligned} \vec{T}_1 &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\ \vec{T}_2 &= \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \end{aligned} \tag{8.9}$$

the linear combination of these coincident twists is given as

$$\vec{T} = A\vec{T}_1 + B\vec{T}_2 = \begin{bmatrix} A & 0 & 0 & B & 0 & 0 \end{bmatrix}. \tag{8.10}$$

Every twist within the freedom space of such a system will, therefore, be coincident with the original twists and will have pitch values, *p*, given by

$$p = \frac{B}{A}. \tag{8.11}$$

If one of the coincident twists is a screw with a finite non-zero pitch value and the other twist is a pure translation with an infinite pitch, a similar analysis will show that the resultant twist will either be a coincident screw or a coincident pure rotation.

Since every possible combination has now been considered, the conclusion can be drawn that any two coincident twists with different pitch values will result in a freedom space that consists of an infinite number of twists that are all coincident with the original two twists with pitch values that correspond to every real number ranging from negative infinity to positive infinity. This freedom space is shown in **Figure 8.2**. One of the lines is red corresponding to the single pure rotational freedom line that exists within the space. There is also a pure rotational hoop with a normal vector that points in the direction of the pure rotational freedom line. This hoop represents the pure translation that also exists within this freedom space. There is also a green line that is coincident with the pure rotational freedom line that represents an infinite number of screws each with a unique non-zero finite pitch value. This freedom space belongs to Case 4 because it consists of two independent twists.

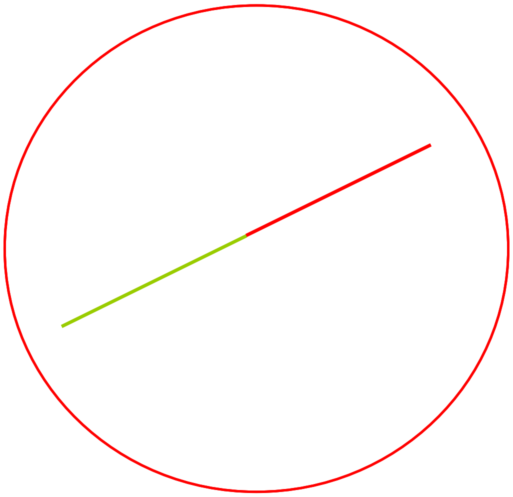


Figure 8.2: Freedom space within Case 4 resulting from the linear combination of two coincident twist lines with different pitch values

8.1.1.2 Parallel Pairs of Twists

This section mathematically proves and describes all the freedom spaces that result from linearly combining two general twists that are parallel. **Equation (8.1)** through **Equation (8.5)** are all applicable equations for any pair of parallel twists ($d \neq 0$ from **Figure 8.1)**.

If both parallel twists are pure rotations ($p_1 = p_2 = 0$), **Equation (8.4)** suggests that all the resultant twists will also be pure rotations (*p*=0) as long as *A*+*B* does not equal zero. It is also known form the location vector given in **Equation (8.5)** that these resultant pure rotations all lay on the x-y plane. As long as *A*+*B* does not equal zero, these resultant freedom lines are all parallel to the original two parallel twists. This fact is known because the orientation vector given in **Equation (8.3)** suggests that they will always point in the direction of the x-axis. If *A*+*B*=0, however, the resultant twist is a pure translation that points along the z-axis perpendicular to the plane of the pure rotational freedom lines. This fact is known because the rotational velocity vector in **Equation (8.3)** will be a zero vector and the translational velocity vector's x- and y-components will equal zero while its z-component will equal –*dB* when *A*+*B*=0 and $p_1 = p_2 = 0$.

The complete freedom space resulting from two parallel pure rotational freedom lines, therefore, is shown in **Figure (8.3)**. It consists of a plane that contains an infinite number of parallel pure rotational freedom lines and a pure rotational hoop with a normal vector that is parallel to the normal vector of the plane of parallel lines.

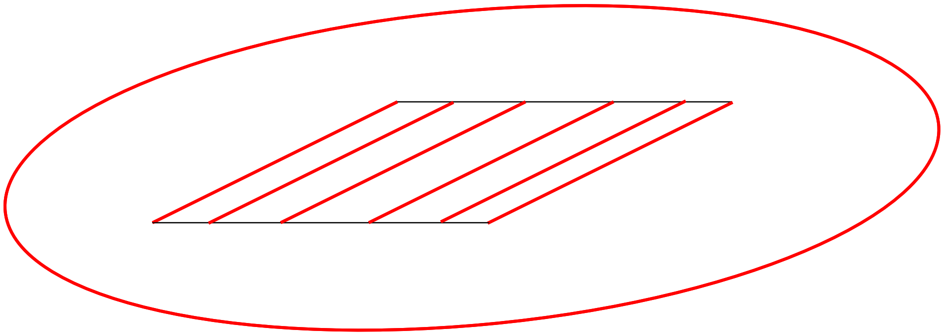


Figure 8.3: Freedom space within Case 4 resulting from the linear combination of parallel pure rotational freedom lines

If both parallel twists are screws with equal finite non-zero pitch values ($p_1 = p_2 \neq 0$), **Equation (8.4)** suggests that all the resultant twists will also be screws with equal pitch values ($p = p_1 = p_2$) as long as *A*+*B* does not equal zero. It is also known form the location vector given in **Equation (8.5)** that these resultant screws of equal pitch all lay on the x-y plane. As long as *A*+*B* does not equal zero, these resultant screws are all parallel to the original two parallel twists. This fact is known because the orientation vector given in **Equation (8.3)** suggests that they will always point in the direction of the x-axis. If *A*+*B*=0, however, the resultant twist is a pure translation that points along the z-axis perpendicular to the plane of the screw lines. This fact is known because the rotational velocity vector in **Equation (8.3)** will be a zero vector and the translational velocity vector's x- and y-components will equal zero while its z-component will equal –*dB* when *A*+*B*=0 and $p = p_1 = p_2$.

The complete freedom space resulting from two parallel screw lines of equal pitch, therefore, is shown in **Figure (8.4)**. It consists of a plane that contains an infinite number of parallel screw lines of equal pitch and a pure rotational hoop with a normal vector that is parallel to the normal vector of the plane of parallel lines.

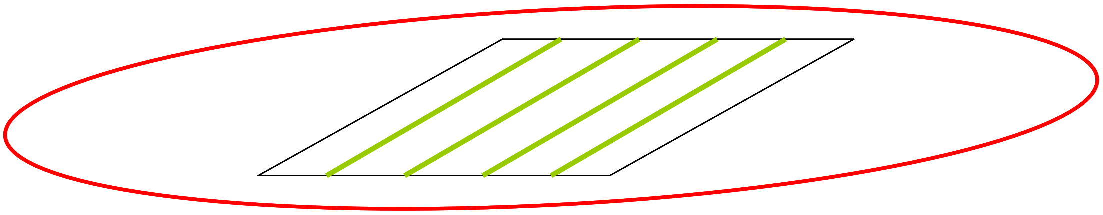


Figure 8.4: Freedom space within Case 4 resulting from the linear combination of parallel screw lines of equal pitch

If both parallel twist lines are pure translations, their linear combination results in a freedom space that consists of a single pure translation. This resultant pure translation points in the same direction as the original two twist lines. This fact is known from the study of pure translations in **Chapter 4**. Recall that pure translations are only directional and cannot be described using location vectors. There is, therefore, no difference between coincident pure translational lines and parallel pure translational lines. It doesn't make sense to distinguish between them. Both conditions result in twists that may be described using **Equation (8.6)**. The freedom space created from two "parallel" pure translations, therefore, belongs to a system within Case 5 since it consists of a single independent twist as mentioned earlier.

For the same reason, there is no difference between the freedom space created by combining a pure rotational freedom line with a "coincident" pure translational line and the freedom space created by combining a pure rotational freedom line with a "parallel" pure translational line. Both scenarios create the freedom space shown in **Figure 8.2**. Again the same principle applies when considering the freedom space created by combining a screw line with a "parallel" pure translational line. This combination results in the same freedom space as the freedom space created by combining a screw line with a "coincident" pure translational line. This freedom space is also the one shown in **Figure 8.2**.

If one of the parallel twist lines is a pure rotational freedom line ($p_1 = 0$) and the other is a screw line with a finite non-zero pitch value, **Equation (8.4)** suggests that the resultant twist's pitch is given by **Equation (8.8)** as long as *A*+*B* does not equal zero. These lines all lie on the x-y plane and are all parallel to the original parallel twist lines. This fact is known from the location vector given in **Equation (8.5)** for parallel twist lines. Note also from **Equation (8.5)** and **Equation (8.8)** that the farther away the resultant screws are from the pure rotational freedom line, the larger their pitch increases. This increase is linear. If *A*+*B*=0, **Equation (8.3)** suggests that the resultant twist is a pure translation that points in the direction

$$\vec{v} = \begin{bmatrix} Bp_2 & 0 & -dB \end{bmatrix}. \tag{8.12}$$

The projection of this vector onto the plane containing the pure rotational freedom line and the infinite screw lines is always parallel to these lines.

If the two parallel twists are screws with different finite non-zero pitch values ($p_1 \neq p_2 \neq 0$), a similar freedom space is created. The pitch of these resultant twists is given in **Equation (8.4)**. These resultant twists all lie on the x-y plane and are all parallel to the original parallel twists. One of the resultant twists on that plane will be a pure rotational freedom line. If A+B=0, it can be determined that the resultant twist is a pure translation that points in the direction of the $\vec{v}$ vector from **Equation (8.3)**.

The freedom space resulting from the linear combination of two parallel twists with different finite pitch values is, therefore, shown in **Figure 8.5**. It consists of a plane containing a single pure rotational freedom line and an infinite number of screws with finite non-zero pitch values. The farther these screws are located from the pure rotational freedom line, the larger their pitch values become. Every twist line on this plane is parallel. A pure rotational hoop also exists with a normal vector that points in a direction that is neither parallel nor orthogonal to the twist lines on the plane. The projection of this normal vector onto this plane is, however, parallel to the twist lines on the plane. As the hoop's normal vector gets closer to being parallel to the pure rotation on the plane of twists, the pitch values of the screws on the plane will increase.

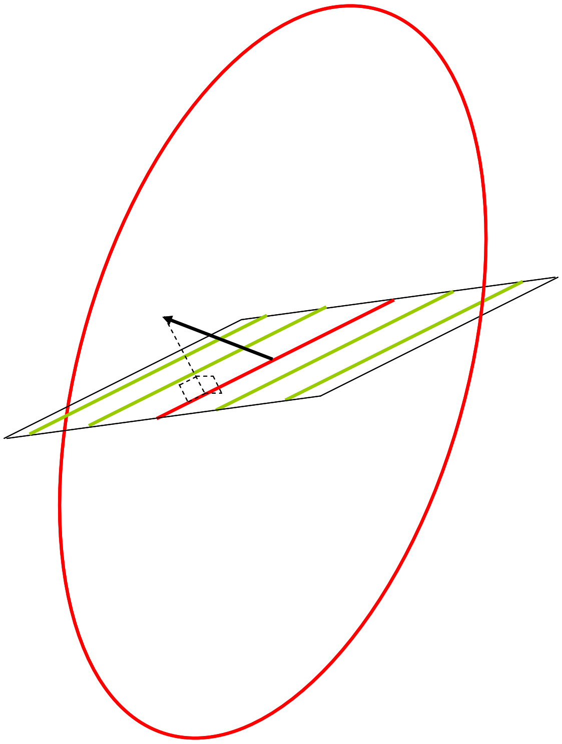


Figure 8.5: Freedom space within Case 4 resulting from the linear combination of two parallel twist lines of different finite pitch values

If the pure translation is perpendicular to the plane of twists, the freedom space becomes the freedom space shown in either **Figure 8.3** or in **Figure 8.4**. If the pure translation is parallel to the twists on the plane, the freedom space becomes the freedom space shown in **Figure 8.2**.

Every possible way any two twists may be combined with coincident or parallel orientations has now been considered. So far four freedom spaces within Case 4 have been found and mathematically described that resulted from these linear combinations. Twists that intersect or are skew with respect to each other are now ready to be considered.

8.1.2 Intersecting and Skew Pairs of Twists

This section mathematically proves and describes all the freedom spaces that result from linearly combining two general twists that either intersect or are skew. To begin the study, the case of two intersecting, orthogonal twists will be considered. A general example of such twists is shown in **Figure 8.6**.

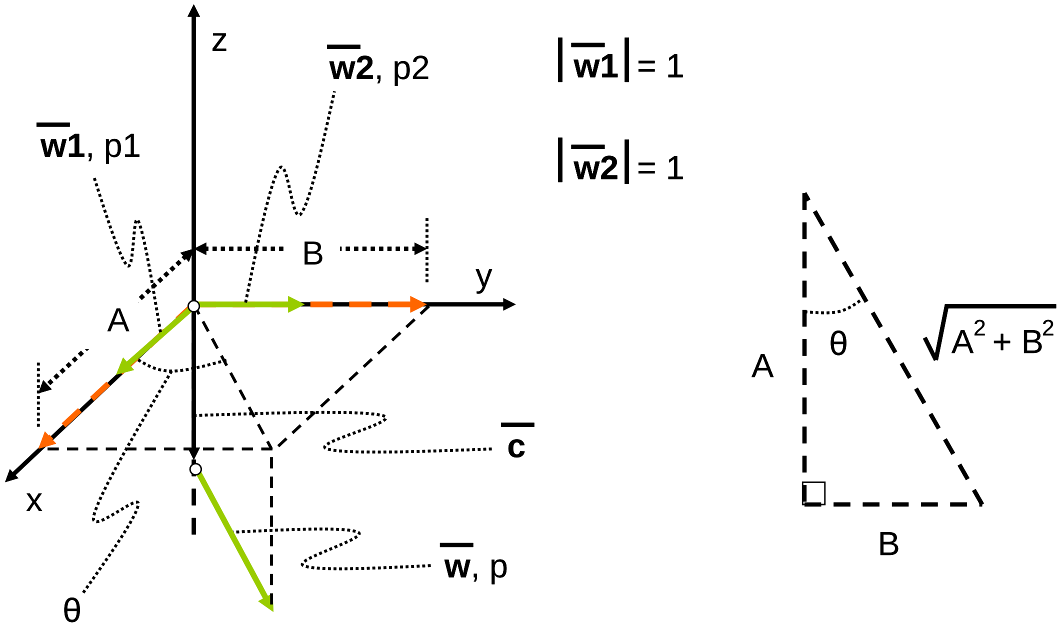

z
$\overline{\mathbf{w2}}$, p2
$|\overline{\mathbf{w1}}| = 1$
$\overline{\mathbf{w1}}$, p1
$|\overline{\mathbf{w2}}| = 1$
B
y
A
x
$\overline{\mathbf{c}}$
$\overline{\mathbf{w}}$, p
θ
A
θ
$\sqrt{A^2 + B^2}$
B

Figure 8.6: Parameters defined for two orthogonally intersecting twists that lie along the x and y axes. Another twist is also shown that represents a general linear combination of the other two twists.

Using **Figure 8.6** and the principles discussed in **Chapter 3**, the two orthogonal twists can be defined as

$$\vec{T}_1 = [1 \quad 0 \quad 0 \quad p_1 \quad 0 \quad 0]$$
$$\vec{T}_2 = [0 \quad 1 \quad 0 \quad 0 \quad p_2 \quad 0]. \tag{8.13}$$

The general freedom space that results from the linear combination of these two independent twists is, therefore, mathematically represented as

$$\vec{T} = A\vec{T}_1 + B\vec{T}_2 = [A \quad B \quad 0 \quad Ap_1 \quad Bp_2 \quad 0], \tag{8.14}$$

where A and B are any real numbers. **Equation (8.14)** suggests that every twist within the general freedom space will have a rotational velocity vector, $\vec{w}$, and a translational velocity vector, $\vec{v}$, of

$$\begin{aligned} \vec{w} &= \begin{bmatrix} A & B & 0 \end{bmatrix} \\ \vec{v} &= \begin{bmatrix} Ap_1 & Bp_2 & 0 \end{bmatrix}. \end{aligned} \tag{8.15}$$

If **Equation (8.15)** is plugged into **Equation (3.4)**, one finds that every twist within the general freedom space will have a pitch value, *p*, of

$$p = \frac{A^2 p_1 + B^2 p_2}{A^2 + B^2}. \tag{8.16}$$

If one applies **Equation (8.15)** and **Equation (8.16)** to the location matrix equation given in **Equation (3.8)** to find a possible location vector, $\vec{c}$, for the twists within the general freedom space, one finds that

$$\vec{c} = \begin{bmatrix} c_x & \frac{B}{A} c_x & \frac{(p_2 - p_1)AB}{(A^2 + B^2)} \end{bmatrix}, \tag{8.17}$$

where c_x may be any real value. **Equation (8.17)** applies only to twists that have location vectors that are not pure translations. In other words, *A* and *B* cannot simultaneously equal zero for **Equation (8.17)** to exist.

Before proceeding to identify and describe all the freedom spaces that are created by combining two orthogonal, intersecting twists, it must first be emphasized that the linear combination of any two such twists will result in an infinite number of twists that all lie on the surface of a cylindroid or a disk, which is essentially a collapsed cylindroid where the *h* parameter shown in **Figure 6.9** from **Chapter 6** equals zero. The two orthogonal, intersecting twists will be the principal generators of the resulting cylindroid. This fact is proven in **Appendix L**.

Now suppose the two orthogonal intersecting twists are both pure rotations such that $p_1 = p_2 = 0$. **Equation (8.16)** suggests that all the other twists in the resulting freedom space must also have pitch values equal to zero. It is known that this freedom space consisting of pure rotational freedom lines is a disk because of **Equation (L.6)** in **Appendix L**. This freedom space is shown in **Figure 8.7**.

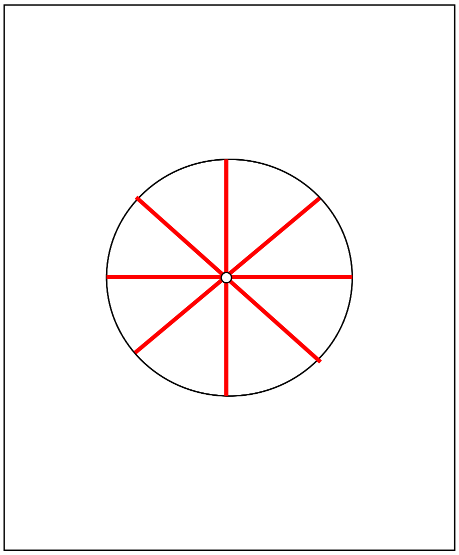


Figure 8.7: Freedom space within Case 4 resulting from the linear combination of two intersecting pure rotational freedom lines.

Suppose the two orthogonal intersecting twists are both screws with equivalent finite, non-zero pitch values such that $p_1 = p_2 \neq 0$. **Equation (8.16)** suggests that all the other twists in the resulting freedom space must also have equivalent pitch values. This freedom space consisting of screws is a disk because of **Equation (L.6)** in **Appendix L**. This freedom space is shown in **Figure 8.8**.

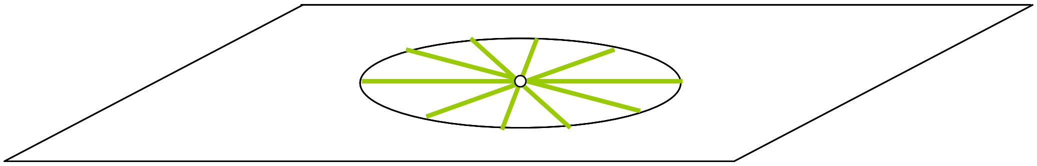


Figure 8.8: Freedom space within Case 4 resulting from the linear combination of two intersecting screws with equivalent pitch values. Every screw within the disk has the same pitch value.

Suppose the two orthogonal intersecting twists are both pure translations with infinite pitch values such that

$$\vec{T}_1 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$
$$\vec{T}_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \tag{8.18}$$

The linear combination of these two pure translations must result in twists that are also pure translations and point in directions parallel to the x-y plane. This fact is evident by inspection of the resultant twist, $\vec{T}$, given as

$$\vec{T} = A\vec{T}_1 + B\vec{T}_2 = \begin{bmatrix} 0 & 0 & 0 & A & B & 0 \end{bmatrix}. \tag{8.19}$$

The freedom space resulting from these two twists is, therefore, a disk containing all pure translations and is shown in **Figure 8.9**.

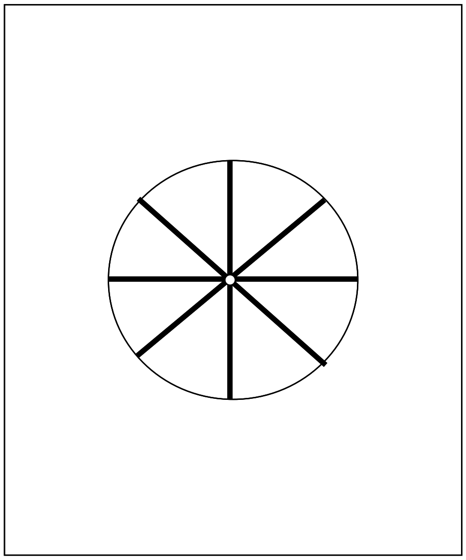


Figure 8.9: Freedom space within Case 4 resulting from the linear combination of two pure rotations that point in different directions.

Recall, however, from **Section 5.2.4** in **Chapter 5** that this particular freedom space cannot have a feasible complementary constraint space that consists of four non-redundant constraints that are only capable of imparting axial forces (where *q*=0). In the realm of flexure systems, therefore, this freedom space will not be counted among the Case 4 types.

If one of the orthogonal intersecting twists is a pure rotation ($p_1 = 0$) and the other is a screw with a non-zero finite pitch value ($p_2 \neq 0$), it may be determined from **Equation (L.6)** in **Appendix L** that the resulting freedom space will be a cylindroid of twists with a height, *h*, equal to the magnitude of the pitch value, p_2. Since $p_1 = 0$, it may also be determined from **Equation (8.16)** that every twist within this freedom space will have a pitch value, *p*, equal to

$$p = \frac{B^2 p_2}{A^2 + B^2} . \tag{8.20}$$

Since $p_2 \neq 0$, **Equation (8.20)** suggests that the only way any twist within this cylindroid freedom space may be a pure rotational freedom line (where *p*=0) is if *B*=0. But when this condition is true, **Equation (8.14)** suggests that $\vec{T}_2$ has no effect on the linear combination and that the only pure rotational freedom line that exists within the freedom space is the principal generator, $\vec{T}_1$, that was declared to be a pure rotation from the beginning.

A new freedom space that consists of an infinite number of twists on the surface of a cylindroid has, therefore, been identified where every twist within the cylindroid is a screw with a finite non-zero pitch value except for a single pure rotational freedom line that will always be one of the cylindroid's principal generators. This freedom space is shown in **Figure 8.10**.

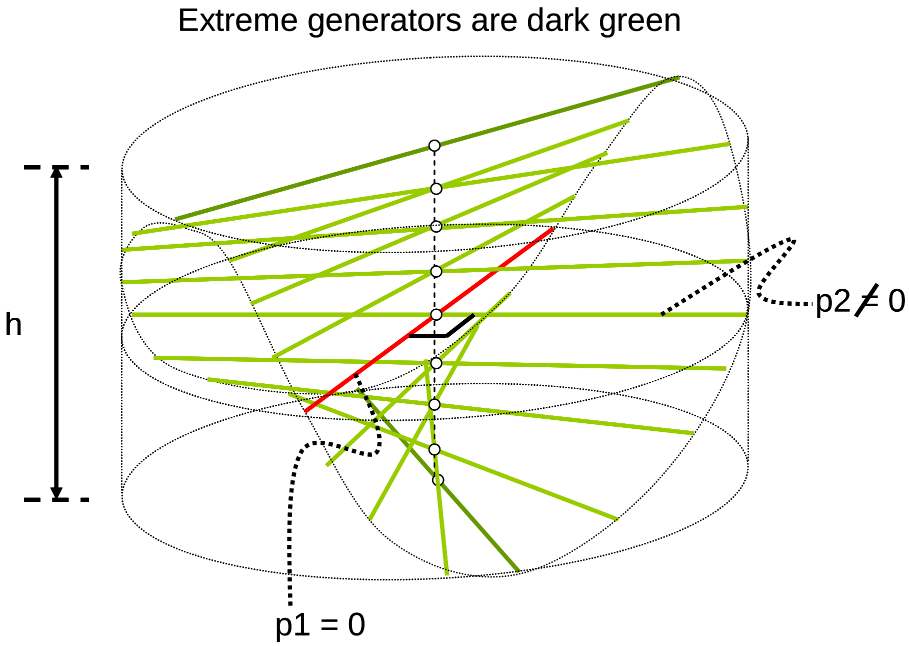


Figure 8.10: Freedom space within Case 4 that consists of a single pure rotational freedom line (red) and an infinite number of screws (green) on the surface of a cylindroid where the pure rotational freedom line is one of the cylindroid's principal generators.

If one of the orthogonal intersecting twists is a pure rotation ($p_1 = 0$) and the other is a pure translation ($p_2 = \infty$), the same freedom space that was shown in **Figure 8.3** will result from their linear combination. This should be clear by noting that two such orthogonal twists already exist within the freedom space shown in **Figure 8.3**.

If both orthogonal intersecting twists are screws with different finite non-zero pitch values such that $p_1 \neq p_2 \neq 0$, **Equation (L.6)** in **Appendix L** suggests that the resulting freedom space will be a cylindroid of twists with a height, *h*, equal to the magnitude of the difference between the two pitch values, $p_1 - p_2$. Every twist within the freedom space has a pitch value, *p*, given by **Equation (8.16)**. If this value is set equal to zero in an attempt to determine how many pure rotational freedom lines exist within the freedom space, it is found that

$$A = \pm B\sqrt{-\frac{p_2}{p_1}}. \tag{8.21}$$

Equation (8.21) suggests that as long as the two principal generator twists within the cylindroid freedom space have pitch values of opposite signs, two pairs of real *A* and *B* constants exist such that **Equation (8.14)** results in two pure rotational twists. In other words, if the principal generators' pitch values have different signs, the resulting freedom space will be a cylindroid containing two pure rotations. The rest of the twists in the freedom space will be screws with finite non-zero pitch values. The two pure rotational freedom lines must be skew. If they intersect, they would belong to the freedom space shown in **Figure 8.7**.

A new freedom space that consists of an infinite number of twists on the surface of a cylindroid has, therefore, been identified where every twist within the cylindroid is a screw with a finite non-zero pitch value except for two skew pure rotational freedom lines. This freedom space is shown in **Figure 8.11**.

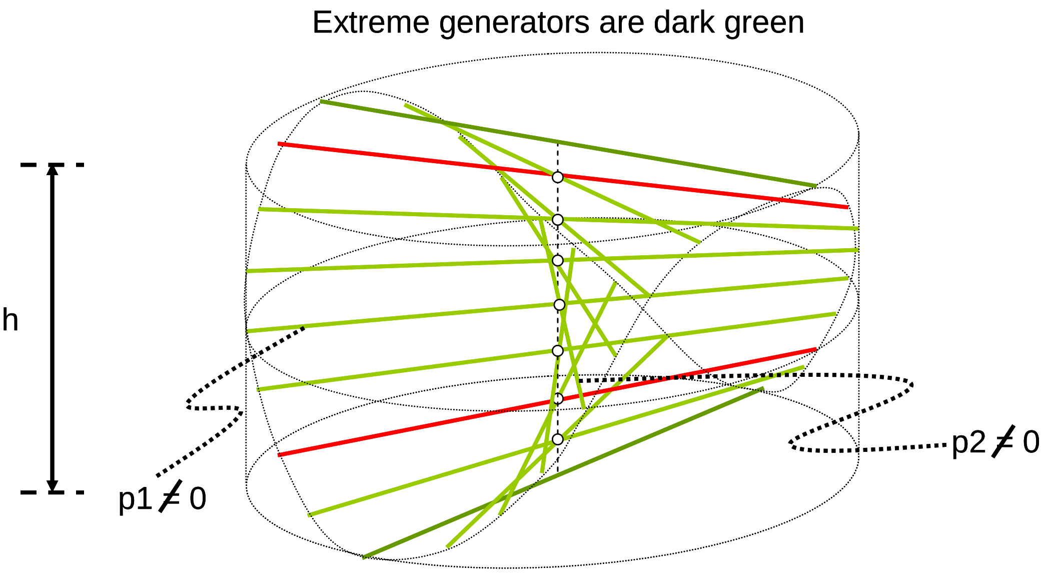


Figure 8.11: Freedom space within case 4 that consists of two skew pure rotational freedom lines (red) and an infinite number of screws (green) that all lie on the surface of a cylindroid.

Note from the plus and minus sign in **Equation (8.21)** that no cylindroid freedom space will ever contain more than two pure rotations. There will always be either two skew pure rotations within the cylindroid, one single pure rotation as the principal generator of the cylindroid, or no pure rotations in the cylindroid. The latter case will now be considered.

Equation (8.21) suggests that if the two principal generator twists within the cylindroid freedom space have pitch values of similar signs, no real pairs of *A* and *B* constants exist. In other words, if the principal generators' pitch values have similar signs, the resulting freedom space will be a cylindroid containing no pure rotations. It will consist entirely of screws with finite non-zero pitch values. This new freedom space is shown in **Figure 8.12**.

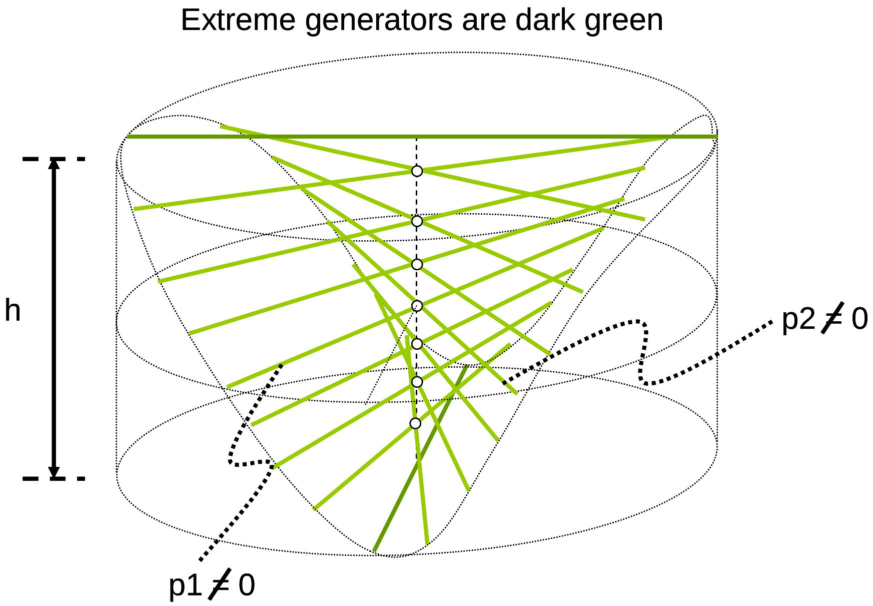


Figure 8.12: Freedom space within Case 4 that consists entirely of screws (green) that all lie on the surface of a cylindroid.

If one of the orthogonal intersecting twists is a screw with a finite non-zero pitch value ($p_1 \neq 0$) and the other is a pure translation ($p_2 = \infty$), the same freedom space that was shown in **Figure 8.4** will result from their linear combination. This should be clear after noting that two such orthogonal twists already exist within the freedom space shown in **Figure 8.4**.

Every possible freedom space that results from linearly combining two orthogonal, intersecting twists has now been found and described. The linear combination of twists that don't just intersect at 90 degrees but intersect at any angle will now be considered. The reader may be surprised to recognize that the freedom spaces that result from these linear combinations have already been found. Suppose, for instance, that two intersecting twists are both pure rotations. Regardless of their angle of intersection, the freedom space created from their linear combination is the space shown in **Figure 8.7**. If these two twists are both screws with equivalent pitch values, the freedom space is the space shown in **Figure 8.8**. If the two twists are both pure translations, the freedom space is the space shown in **Figure 8.9**. If one of the twists is a pure rotation and the other twist that intersects it at an angle that is not 90 degrees is a screw, the freedom space is the space shown in **Figure 8.11**. If one of the twists is a pure rotation and the other twist that intersects it at an angle that is not 90 degrees is a pure translation, the freedom space is the space shown in **Figure 8.5**. If one of the twists is a screw and the other twist that intersects it at an angle that is not 90 degrees is a pure translation, the freedom space is also the space shown in **Figure 8.5**. If both of the twists that intersect at an angle that is not 90 degrees are screws of different finite non-zero pitch values, the freedom space is either the space shown in **Figure 8.10**, **Figure 8.11**, or **Figure 8.12** depending on the angle of intersection and the pitch values of the twists. Any two twists that intersect at any arbitrary angle will, therefore, not generate any new freedom spaces that have not yet been considered previously.

The freedom spaces that are generated by linearly combining two skew twists will now be considered. Again the reader may be surprised to recognize that no new freedom spaces are generated from the linear combination of these twists either. This statement is proven true by simply noting that the linear combination of every possible pair of skew twist lines always results in a cylindroid freedom space regardless of the twists' pitch values, their skew angle or their shortest distance line segment's length. Since only three types of cylindroid freedom spaces exist, the linear combination of any two skew twists always results in either the freedom space shown in **Figure 8.10**, **Figure 8.11**, or **Figure 8.12**.

It makes no sense to consider the linear combination of two skew pure translations or to consider any pure translation being skew to any other twist for that matter. For two lines to be skew with respect to each other, they have to not only have a direction, but also a location. Pure translations only have direction; they have no location. **Chapter 4** discusses this fact in greater detail. In short, the linear combination of two "skew" pure translations will belong to the freedom space shown in **Figure 8.9** and the linear combination of any pure translation that is "skew" with either a pure rotation or a screw will belong to the freedom space shown in **Figure 8.5**.

Every possible way any two twists with any pitch values may be linearly combined has now been considered and, consequently, 10 different freedom spaces that result from these linear combinations have been discovered (9 of which are feasible for flexure system design where their constraint space contains four non-redundant constraints with *q*=0).

The author was also able to confirm the existence of each of these freedom spaces by applying the approach used to find the freedom spaces for the first three cases discussed in **Chapter 7**. This approach is performed by considering every possible way four non-redundant constraints may be arranged in three-space and then by applying Blanding's Rule of Complementary Patterns to locate every pure rotational freedom line that exists for each system. The proof of this approach is extremely lengthy for the fourth case and the freedom spaces that contain no pure rotations are easily overlooked using this approach. For this reason, the author opted to use the more mathematical and thorough approach applied in this section.

A program was also coded using MATLAB to help the author visualize and understand the freedom spaces found in this section and to prove that they are indeed the only existing freedom spaces within Case 4. This program is explained and provided in **Appendix M**.

Before continuing on to the next section, the freedom space shown in **Figure 7.59** from **Chapter 7** will briefly be revisited. It should now be understood why the linear combination of any one of the pure rotational freedom lines on the top plane with any one of the skew pure rotational freedom lines on the bottom plane results in one of the cylindroids in the freedom space consisting of an infinite number of cylindroids that lay side by side.

8.2 Sub-constraint Space

This section introduces the concept of sub-constraint space. An example is given to show the reader how the author went about finding every sub-constraint space within every constraint space.

Sub-constraint spaces, like constraint spaces, are spaces that contain an infinite number of constraint lines. They instruct the designer how to pick the number of non-redundant constraints from within the system's constraint space. Sub-constraint spaces always lie within the constraint space of the system and are generally made up of multiple constraint sets.

The following example will help clarify this concept. Consider the constraint space shown in **Figure 8.13**. This constraint space consists of two constraint sets, a box containing every parallel constraint line in three-space and a plane containing every constraint line that exists on that plane. The constraint space contains four non-redundant constraints and consequently belongs to Case 4. This space will be studied and derived later, but for this example it is sufficient to simply know that it exists.

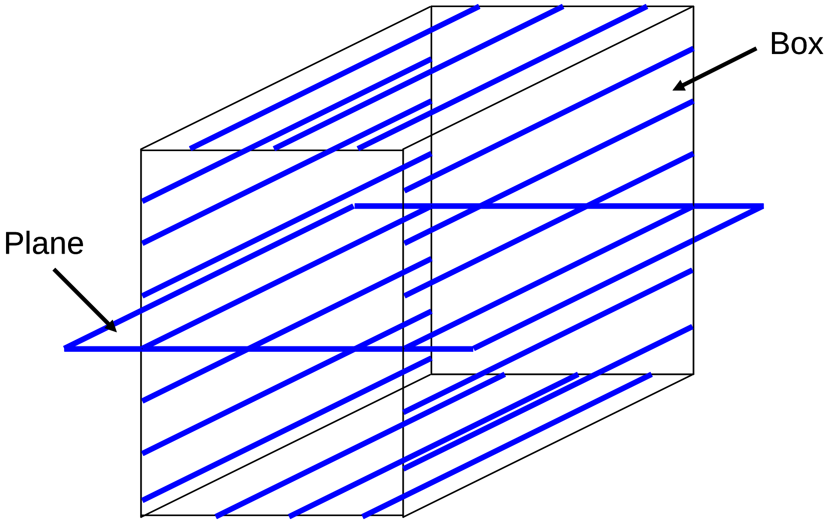


Figure 8.13: Constraint space that contains 4 non-redundant constraints

This space contains four non-redundant constraints, but it is not known which four constraints to select from within the space such that they will all be non-redundant. The four constraints should not all be selected from the plane, for instance, since the constraint space of Case 3, Type 1 suggests that at least one of these four constraints will be redundant. Note, therefore, that some form of instruction is required to inform the designer of all the possible ways non-redundant constraints may be selected from the constraint space. The spaces of constraints that inform the designer of each way non-redundant constraints may be selected from a system are the system's sub-constraint spaces.

To find every sub-constraint space for the system shown in **Figure 8.13**, one must determine how many ways four constraints could be selected from the two sets within the constraint space and then check which of these ways will result in all four constraints being non-redundant. Consider the combinations shown in **Table 8.1**.

Table 8.1: Five different ways four constraints could be selected from the two constraint sets within the constraint space shown in Figure 8.13. The sub-constraint spaces come from the combinations in red.

	Sets within the constraint space	
	Box	**Plane**
Number of constraints to select from within each set	4	0
	0	4
	3	1
	1	3
	2	2

The combinations suggested in the first two rows will never allow the designer to appropriately select four non-redundant constraints. If four constraints are chosen from the box and no constraints are chosen from the plane, the constraint space of Case 3, Type 5 suggests that at least one of these constraints must be redundant. It was already shown earlier that if no constraints are chosen from the box and all four constraints are chosen from the plane, that at least one of these constraints will be redundant.

But suppose one chooses three constraints from the box and one constraint from the plane. If the constraint chosen from the plane is parallel to the constraint lines in the box, it will always be redundant. The designer must, therefore, be instructed to select a constraint from the plane that is not parallel to the constraint lines in the box. Recall also from Case 2, Type 2 that if the three parallel constraints selected from the box all lie on the same plane, one of them will always be redundant. The designer, therefore, must also be instructed to select three constraints from the box that don't all lie on the same plane. As long as these two conditions are satisfied, the designer will always appropriately select four non-redundant constraints where three are selected from the box and one is selected from the plane. This first sub-constraint space is shown in **Figure 8.14**. Note also that it doesn't matter if the designer selects constraints from the box that also lie on the plane. As long as the instructions shown in **Figure 8.14** are properly observed, the four constraints chosen will always be non-redundant.

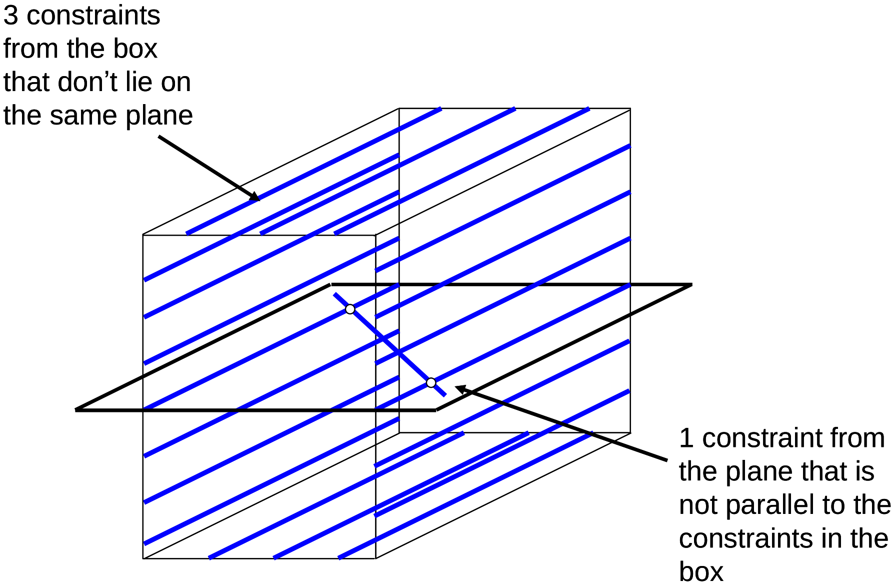

3 constraints from the box that don't lie on the same plane
1 constraint from the plane that is not parallel to the constraints in the box

Figure 8.14: First sub-constraint space for the constraint space shown in Figure 8.13 corresponding to the third row of Table 8.1

Now suppose one chooses three constraints from the plane and one constraint from the box. If the constraint chosen from the box lies on the plane such that all four constraints lie on the plane, Case 3, Type 1 suggests that the constraint will always be redundant. The designer, therefore, must be instructed to select a constraint from the box that does not lie on the plane. Recall also from Case 3, Type 1 that if the three parallel constraints selected from the plane all intersect at the same point in finite space as a disk or at infinity as parallel lines, one of the three constraints will be redundant. The designer, therefore, must also be instructed to select three constraints from the plane that don't all intersect at the same point. As long as these two conditions are satisfied, the designer will always appropriately select four non-redundant constraints where three are selected from the plane and one is selected from the box. This second sub-constraint space is shown in Figure 8.15.

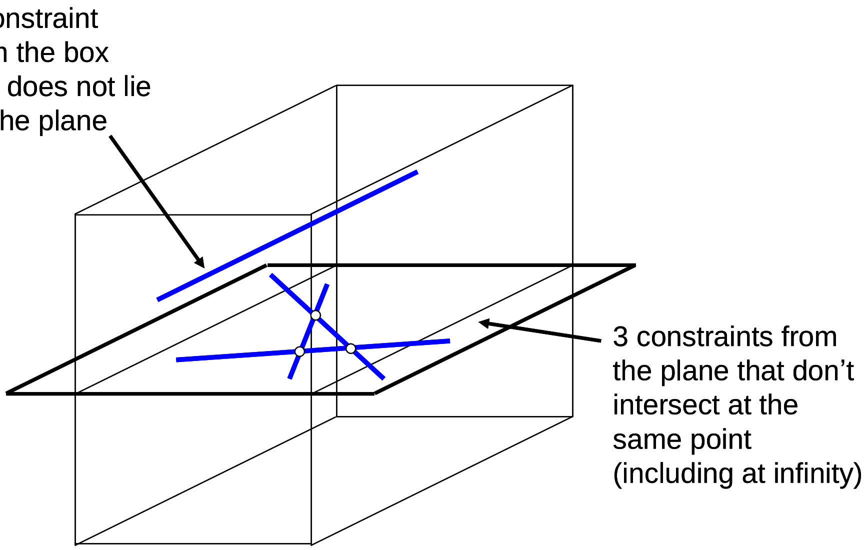

1 constraint from the box that does not lie on the plane
3 constraints from the plane that don't intersect at the same point (including at infinity)

Figure 8.15: Second sub-constraint space for the constraint space shown in Figure 8.13 corresponding to the fourth row of Table 8.1

Now suppose one chooses two constraints from the plane and two constraints from the box corresponding to the combination shown on the last row of **Table 8.1**. Three different ways exist for choosing these lines with this combination.

The first way that two constraints may be chosen from the plane and from the box of the constraint space is shown in the third sub-constraint space given in **Figure 8.16**. This sub-constraint space consists of two constraint sets, a plane of parallel constraint lines and a disk of constraint lines. The plane of parallel lines lies within the box of lines within the system's constraint space. The disk of lines lies within the plane of lines within the system's constraint space. The center of the disk is separated from the intersection line of the two planes by a non-zero distance, *d*. If *d*=0, this sub-constraint space will become the constraint space of Case 3, Type 2. The angle between the two intersecting planes, α, may be any value greater than zero degrees and any value less than 180 degrees. Note that as this angle varies between these values, this plane may express every parallel line found in the box of parallel lines in the constraint space of the system. The designer is instructed to select any two constraints from the plane of parallel lines and any two constraints from the disk of lines. Even if the designer picks the intersection line as one of the constraints from the plane of parallel lines, the four constraints selected will always be non-redundant as long as these instructions are observed.

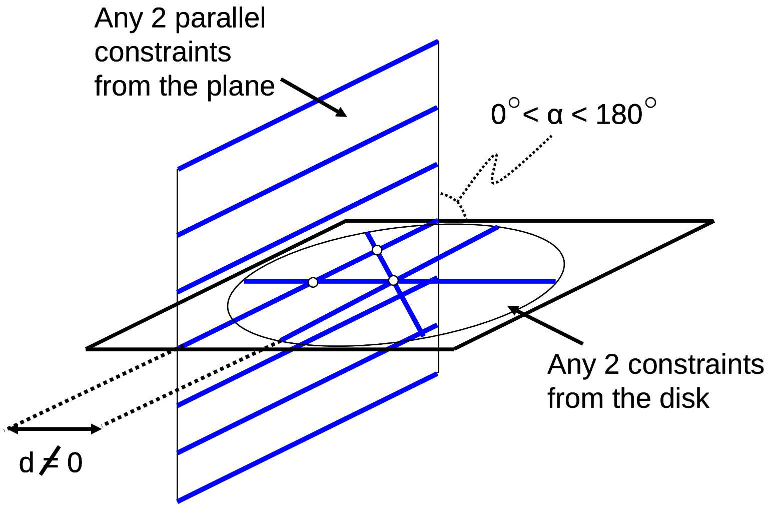

Any 2 parallel constraints from the plane
0°< α < 180°
Any 2 constraints from the disk
d ≠ 0

Figure 8.16: Third sub-constraint space for the constraint space shown in Figure 8.13 corresponding to the fifth row of Table 8.1

The second way that two constraints may be chosen from the plane and from the box of the constraint space is shown in the fourth sub-constraint space given in **Figure 8.17**. This sub-constraint space consists of two constraint sets: a plane of parallel constraint lines, and a disk of constraint lines. The plane of parallel lines lies within the box of lines within the system's constraint space and is parallel to the plane of the disk of lines. The disk of lines lies within the plane of lines within the system's constraint space. The plane of parallel lines is separated from the disk of parallel lines by a non-zero distance, *h*. If *h*=0, this sub-constraint space will become the constraint space of Case 3, Type 1. Note that as this distance varies between negative and positive infinity, this plane may express every parallel line found in the box of parallel lines in the constraint space of the system. The designer is instructed to select any two constraints from the plane of parallel lines and any two constraints from the disk of lines. If the designer observes these instructions, the four constraints selected will always be non-redundant.

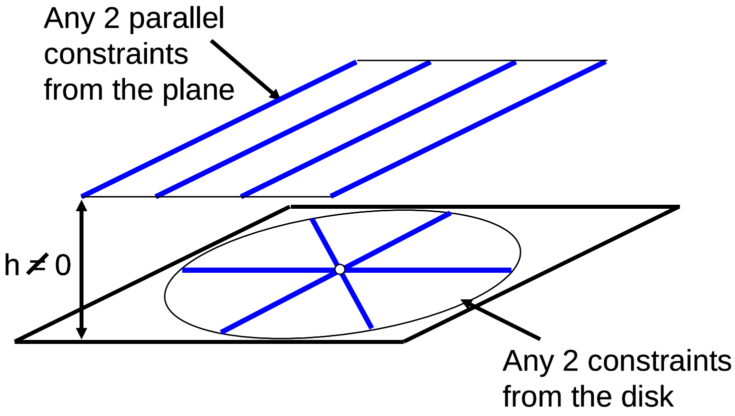

Any 2 parallel constraints from the plane
h ≠ 0
Any 2 constraints from the disk

Figure 8.17: Fourth sub-constraint space for the constraint space shown in Figure 8.13 corresponding to the fifth row of Table 8.1

The third and final way that two constraints may be chosen from the plane and from the box of the constraint space is shown in the fifth sub-constraint space given in **Figure 8.18**. This sub-constraint space consists of two constraint sets, a plane of parallel constraint lines and another plane of parallel constraint lines. The plane of parallel lines with lines that are parallel to the intersection line of the two planes lies within the box of lines within the system's constraint space. The plane of parallel lines with parallel lines that are oriented an angle β from the intersection line of the two planes is coincident with the plane of lines within the system's constraint space. This angle, β, must be greater than zero degrees but less than 180 degrees for this sub-constraint space to exist within the constraint space of this system. If β equaled zero or 180 degrees, this space would belong to the constraint space of Case 3, Type 5. The angle between the two intersecting planes, α, may also be any value greater than zero degrees and any value less than 180 degrees. Note that as this angle varies between these values, this plane expresses every parallel line found in the box of parallel lines in the constraint space of the system. If α equals zero or 180 degrees, this space would belong to the constraint space in Case 3, Type 1. If the two planes of parallel lines were parallel and separated by a distance of *h*, the space would belong to Case 3, Type 6. The designer is instructed to select any two constraints from both planes of parallel lines as shown in **Figure 8.18**. Even if the designer selects the line of intersection between the planes as one of the constraints from the vertically oriented plane in the figure, the four constraints selected will always be non-redundant if these instructions are observed.

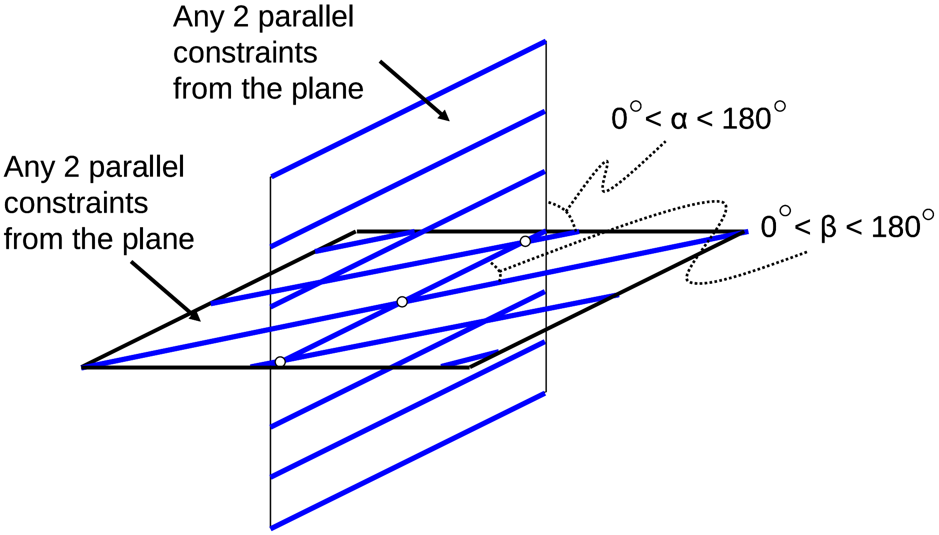

Any 2 parallel constraints from the plane
Any 2 parallel constraints from the plane
0°< α < 180°
0°< β < 180°

Figure 8.18: Fifth sub-constraint space for the constraint space shown in Figure 8.13 corresponding to the fifth row of Table 8.1.

Every sub-constraint space has now been found that instructs the designer how to select four constraints from the constraint space of **Figure 8.13** such that they will be non-redundant. No combination of four non-redundant constraints from within this system exists that doesn't belong to one of these five sub-constraint spaces.

The author used this type of logic to determine all the sub-constraint spaces for every constraint space within Case 4. The sub-constraint spaces will be provided and described later with each constraint space presented in this chapter. Note, however, that selecting sub-constraint spaces is somewhat subjective. Multiple approaches exist for visually representing all the ways non-redundant constraints may be selected from a constraint space. All that is important is that the spaces presented include every combination of non-redundant constraints. The author developed what he believes to be the fewest number of sub-constraint spaces with the fewest and clearest instructions for each constraint space.

Finally, the reader may wonder why the concept of sub-constraint space is surfacing now with Case 4 constraint spaces and why sub-constraint spaces were never an issue with the first three cases discussed in **Chapter 7**. The answer is that all the constraint spaces within the first three cases are so simple that they only have a single sub-constraint space and this sub-constraint space is identical to its constraint space. Recall that for these first three cases, instructions were provided that aided the designer in selecting non-redundant constraints, but there was always only one way to present these instructions for selecting every possible combination of non-redundant constraints and it was always best visually represented using the constraint space itself. Case 4 constraint spaces are, however, much more complicated and may have multiple spaces from which different combinations of non-redundant constraints may be selected as was recently demonstrated.

It will also be shown later that the sub-constraint spaces of Cases 5 are the constraint spaces and sub-constraint spaces of Case 4. The sub-constraint spaces of Case 6 are the constraint spaces of Case 5, the constraint spaces of Case 4, and the sub-constraint spaces of Case 4. This concept will become clear in **Chapter 10**.

8.3 Case 4:

This section describes the fourth case of 6. The fourth case consists of all systems that contain four non-redundant constraints. It has already been proven that 9 freedom spaces exist within this case. These freedom spaces were described in detail in **Section 8.1**. In this section, these freedom spaces will be reviewed briefly and their unique constraint spaces will be determined for all 9 types. Sub-constraint spaces will also be described for every constraint space in this section.

8.3.1 Case 4, Type 1:

This section describes the unique constraint space with its sub-constraint spaces for a system with the freedom space described in **Section 8.1.2** and shown in **Figure 8.7**. For completeness it is shown again here in **Figure 8.19**. Note that this freedom space consists entirely of pure rotational freedom lines. There are no screws or pure translations.

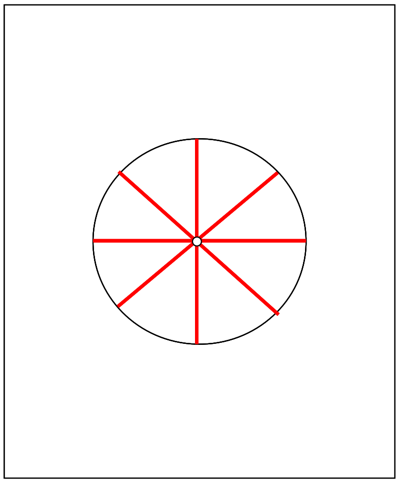


Figure 8.19: Freedom space of Case 4, Type 1

In order to determine this freedom space's complementary constraint space, Blanding's Rule of Complementary Patterns may be applied to determine every constraint line that intersects every freedom line shown in the disk of the freedom space. The resulting constraint space is shown in **Figure 8.20**. It consists of two constraint sets, a plane (outlined in blue) that contains every constraint line on the plane and a sphere that contains every constraint line that intersects a point that lies on the plane of the other constraint set. Another plane is also shown in the figure that intersects the planar constraint set at an angle of θ. The purpose of this plane will become clear when the sub-constraint spaces of the system are described.

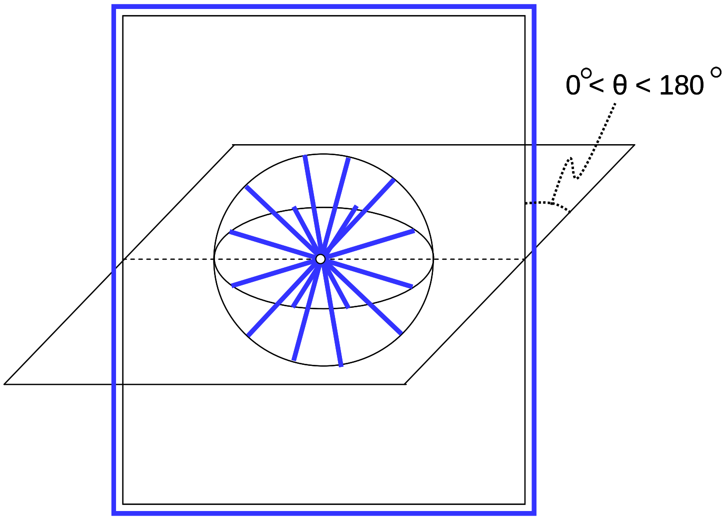


Figure 8.20: Constraint space of Case 4, Type 1

Figure 8.21 shows how the freedom and constraint spaces of this system fit together. The disk of the freedom space lies on the plane of the planar constraint set and its center point is coincident with the center point of the spherical constraint set.

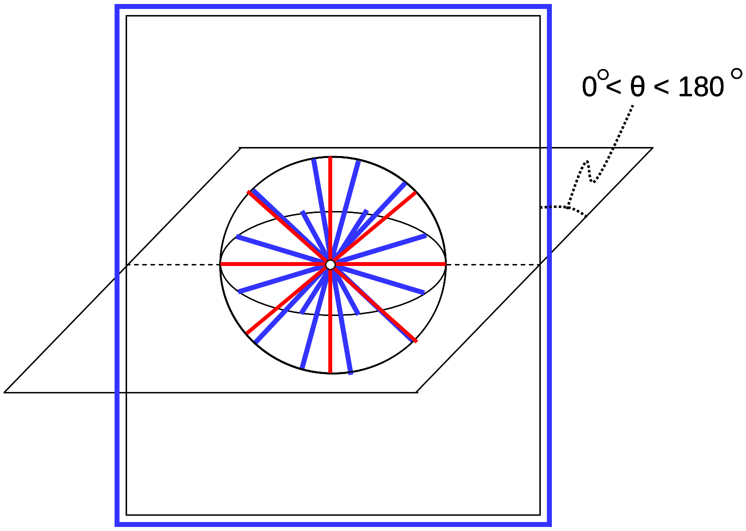


Figure 8.21: Freedom space (red) and constraint space (blue) of Case 4, Type 1 together.

Four sub-constraint spaces exist within this constraint space. The first is shown in **Figure 8.22**. It consists of two constraint sets, a plane of all constraint lines that lie on the plane and a sphere of constraint lines that intersect at a point on the plane. Instructions for choosing the non-redundant constraints are included in the figure.

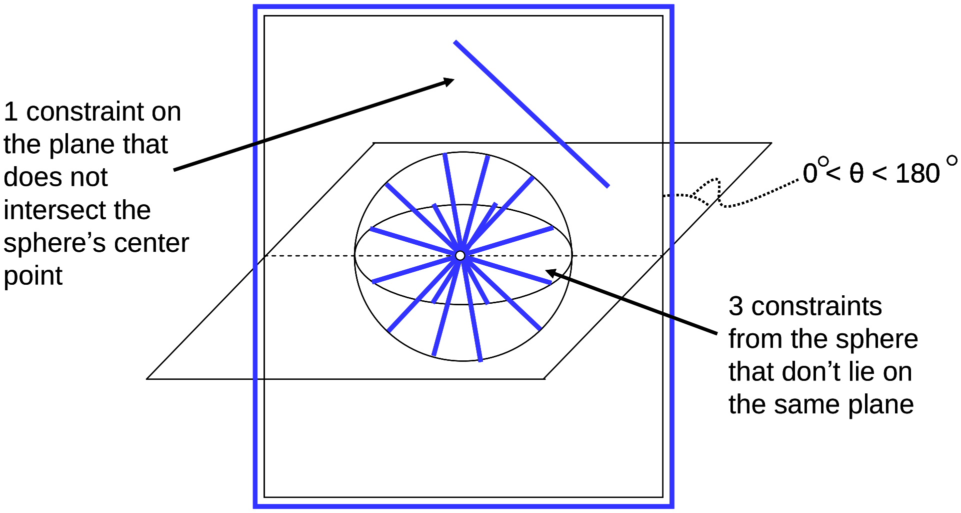

1 constraint on the plane that does not intersect the sphere's center point
0°< θ < 180 °
3 constraints from the sphere that don't lie on the same plane

Figure 8.22: First sub-constraint space of Case 4, Type 1.

The second sub-constraint space is shown in **Figure 8.23**. It consists of two constraint sets, a disk of constraint lines on the vertical plane and a disk of constraint lines on the horizontal plane shown in the figure. The disk on the vertical plane lies on the planar constraint set from the system's constraint space and its center point must lie a non-zero distance, *d*, from the intersection line of the two planes. As long as this requirement is met, this disk may be located anywhere on this plane. If *d*=0, the space would belong to either Case 3, Type 3 or Case 3, Type 4 depending on where the disk's center point lies with respect to the center point of the other disk on the intersection line. The disk that lies on the horizontal plane lies within the spherical constraint set from the system's constraint space. As the angle, θ, between these two planes is varied between zero and 180 degrees, this disk represents every constraint line within the spherical constraint set from the system's constraint space. Instructions for choosing the non-redundant constraints are included in the figure.

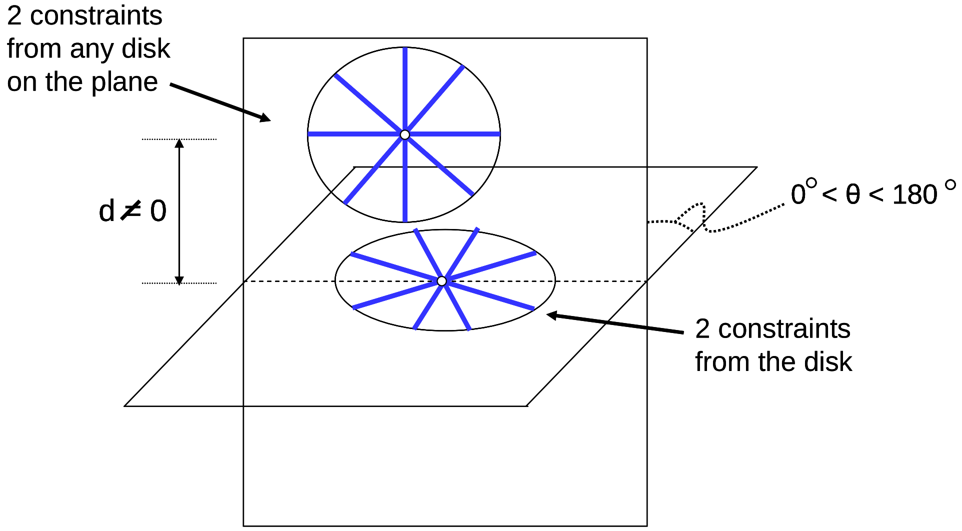

2 constraints from any disk on the plane
$d \neq 0$
$0^\circ < \theta < 180^\circ$
2 constraints from the disk

Figure 8.23: Second sub-constraint space of Case 4, Type 1.

The third sub-constraint space is shown in **Figure 8.24**. It consists of two constraint sets, a plane of parallel lines on the vertical plane and a disk of constraint lines on the horizontal plane shown in the figure. The parallel lines on the vertical plane lie within the planar constraint set of the system's constraint space and are oriented an angle α from the intersection line of the two planes. This angle is allowed to vary between zero and 180 degrees. If α=0 or 180 degrees, the space becomes the constraint space of Case 3, Type 2. The disk that lies on the horizontal plane lies within the spherical constraint set from the system's constraint space. As the angle θ between these two planes is varied between zero and 180 degrees, this disk represents every constraint line within the spherical constraint set from the system's constraint space. Instructions for choosing the non-redundant constraints are included in the figure.

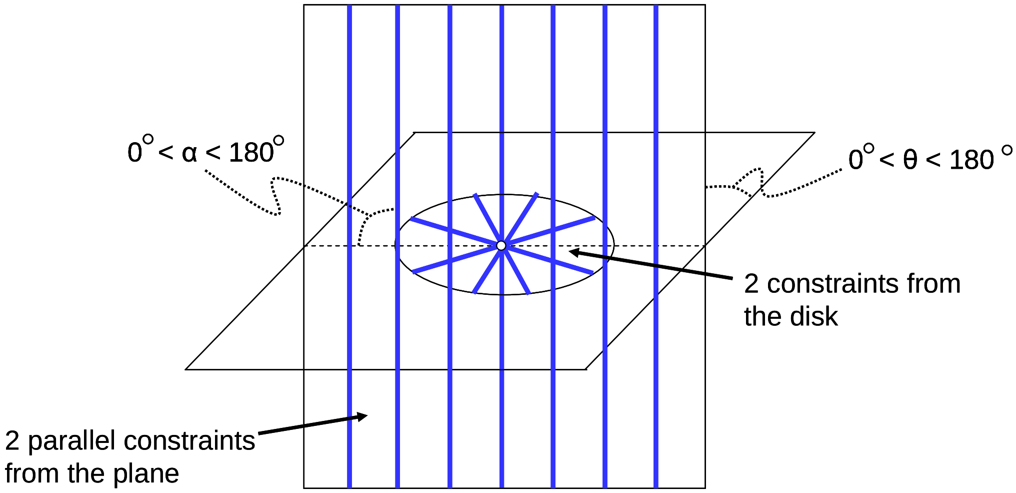

0° < α < 180°
0° < θ < 180 °
2 constraints from the disk
2 parallel constraints from the plane

Figure 8.24: Third sub-constraint space of Case 4, Type 1.

The fourth and final sub-constraint space of Case 4, Type 1 is shown in **Figure 8.25**. It consists of two constraint sets, a plane of all constraint lines that lie on the plane and a sphere of constraint lines that intersect at a point on the plane. Instructions for choosing the non-redundant constraints are included in the figure.

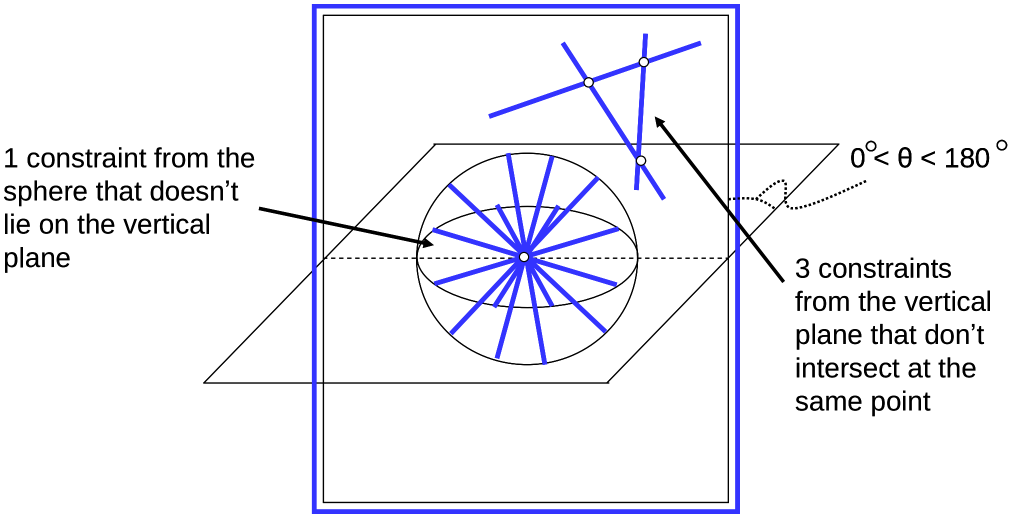

1 constraint from the sphere that doesn't lie on the vertical plane
0°< θ < 180 °
3 constraints from the vertical plane that don't intersect at the same point

Figure 8.25: Fourth sub-constraint space of Case 4, Type 1.

8.3.2 Case 4, Type 2:

This section describes the unique constraint space with its sub-constraint spaces for a system with the freedom space described in **Section 8.1.1** and shown in **Figure 8.3**. For completeness, it is shown again here in **Figure 8.26**. Note that this freedom space consists entirely of pure rotational freedom lines with a single pure rotational hoop. There are no screws in the system.

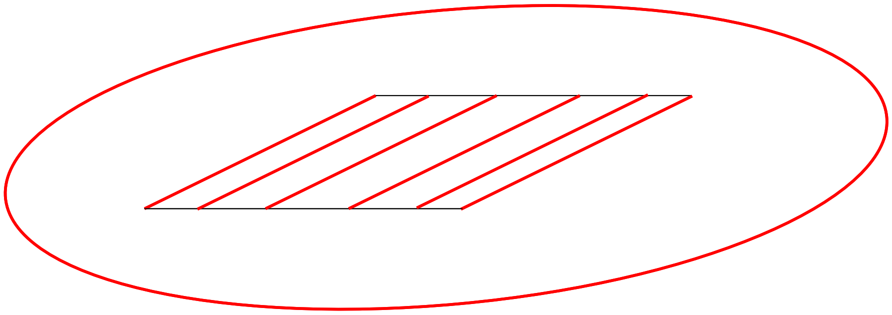


Figure 8.26: Freedom space of Case 4, Type 2

In order to determine this freedom space's complementary constraint space, Blanding's Rule of Complementary Patterns may be applied to determine every constraint line that intersects every freedom line shown in the freedom space. The resulting constraint space is the space studied in **Section 8.2** and is shown again here in **Figure 8.27**. It consists of two constraint sets, a plane (outlined in blue) that contains every constraint line on the plane and a box that contains every parallel constraint line in three-space.

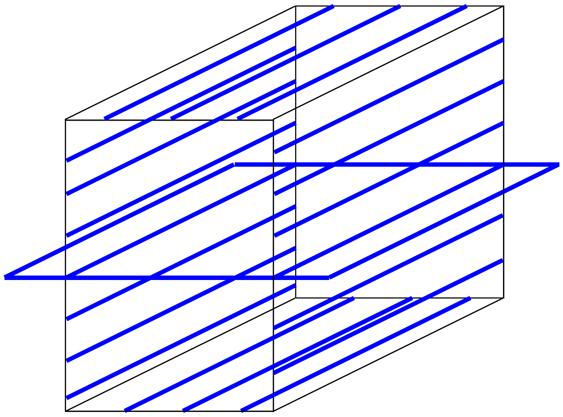


Figure 8.27: Constraint space of Case 4, Type 2

Figure 8.28 shows how the freedom and constraint spaces of this system fit together. The plane of parallel lines in the freedom space is coincident with the plane of the planar constraint set.

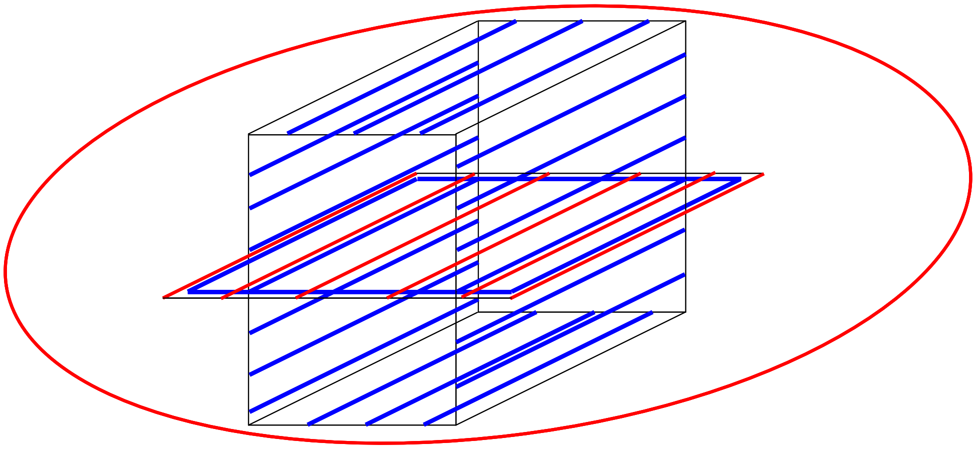


Figure 8.28: Freedom space (red) and constraint space (blue) of Case 4, Type 2 together.

From **Section 8.2** five sub-constraint spaces within the constraint space of this system were found and described. For completeness, they will again be shown here in **Figure 8.29** through **Figure 8.33**. For a detailed description of their geometry review **Section 8.2**.

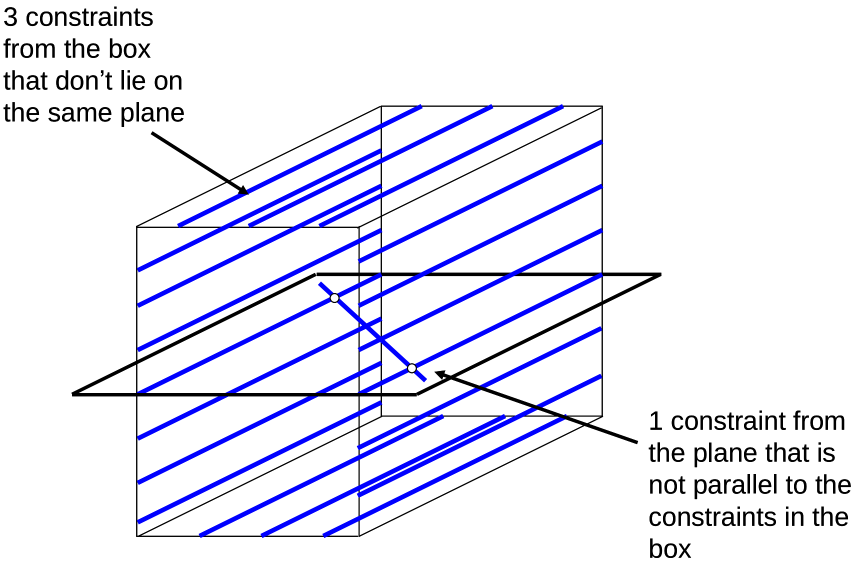


Figure 8.29: First sub-constraint space of Case 4, Type 2

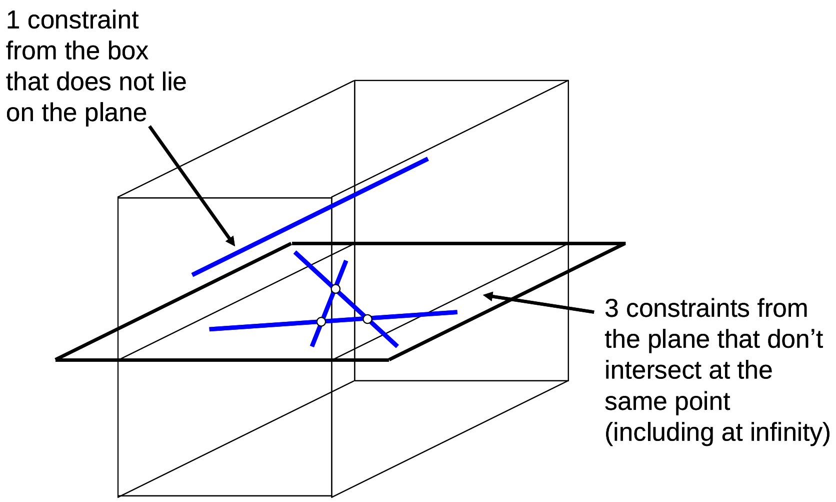


Figure 8.30: Second sub-constraint space of Case 4, Type 2

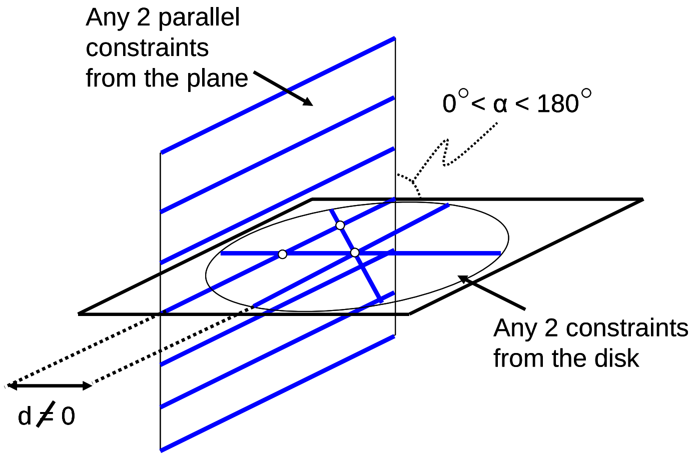


Figure 8.31: Third sub-constraint space of Case 4, Type 2

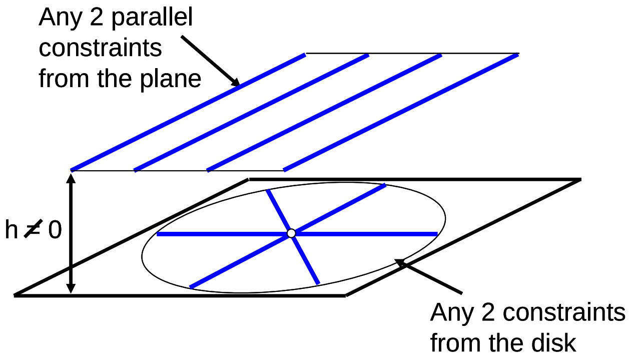


Figure 8.32: Fourth sub-constraint space of Case 4, Type 2

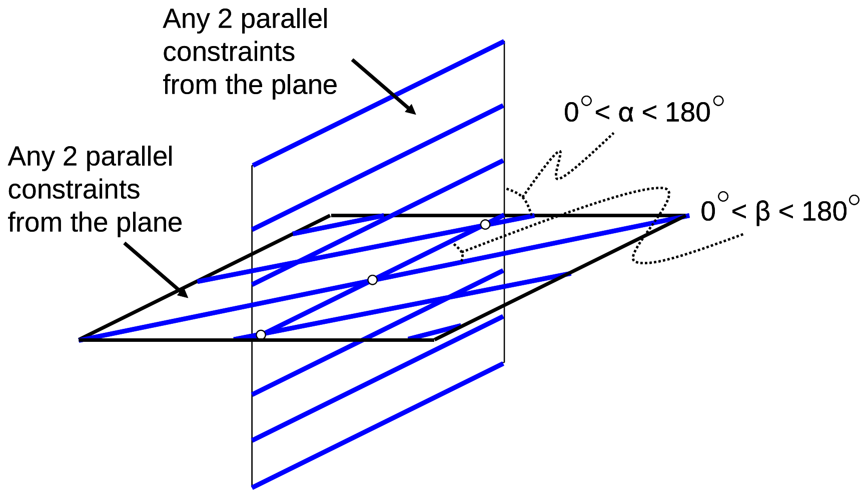

Any 2 parallel constraints from the plane
Any 2 parallel constraints from the plane
0° < α < 180°
0° < β < 180°

Figure 8.33: Fifth sub-constraint space of Case 4, Type 2

8.3.3 Case 4, Type 3:

This section describes the unique constraint space with its sub-constraint spaces for a system with the freedom space described in **Section 8.1.2** and shown in **Figure 8.11**. For completeness, it will be shown again here in **Figure 8.34**. In this figure the two skew pure rotational freedom lines are arbitrarily shown as extreme generators of the cylindroid and therefore, have a 90 degree skew angle with respect to each other. As discussed earlier, however, this condition doesn't necessarily have to be true to belong to this particular type. All that matters is that the pure rotational freedom lines are skew with respect to each other.

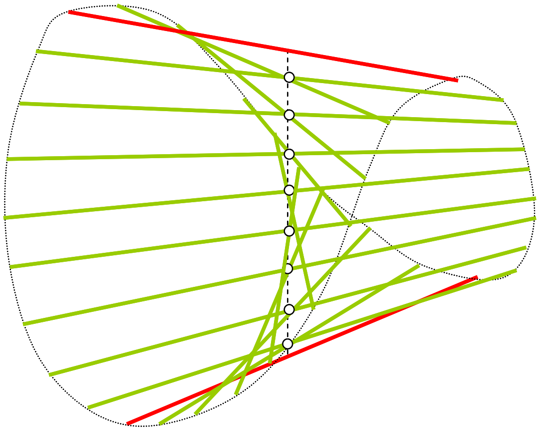


Figure 8.34: Freedom space of Case 4, Type 3

In order to determine this freedom space's complementary constraint space, Blanding's Rule of Complementary Patterns may be applied to determine every constraint line that intersects the two pure rotational freedom lines shown in the cylindroid freedom space. One can correctly deduce that if all the constraint lines are found that complement these two pure rotational freedom lines, the other screws in the cylindroid will also be allowable motions since they are all linear combinations of the two pure rotations. The constraint space that complements the entire freedom space is, therefore, shown in **Figure 8.35**. This space is almost identical to the freedom space of pure rotational freedom lines from Case 2, Type 3 shown and described in **Figure 7.27** of **Chapter 7**. The only difference between this constraint space and the freedom space from Case 2, Type 3 is that this constraint space has no constraint hoop representing a constraint capable of pure torque (*q*=∞).

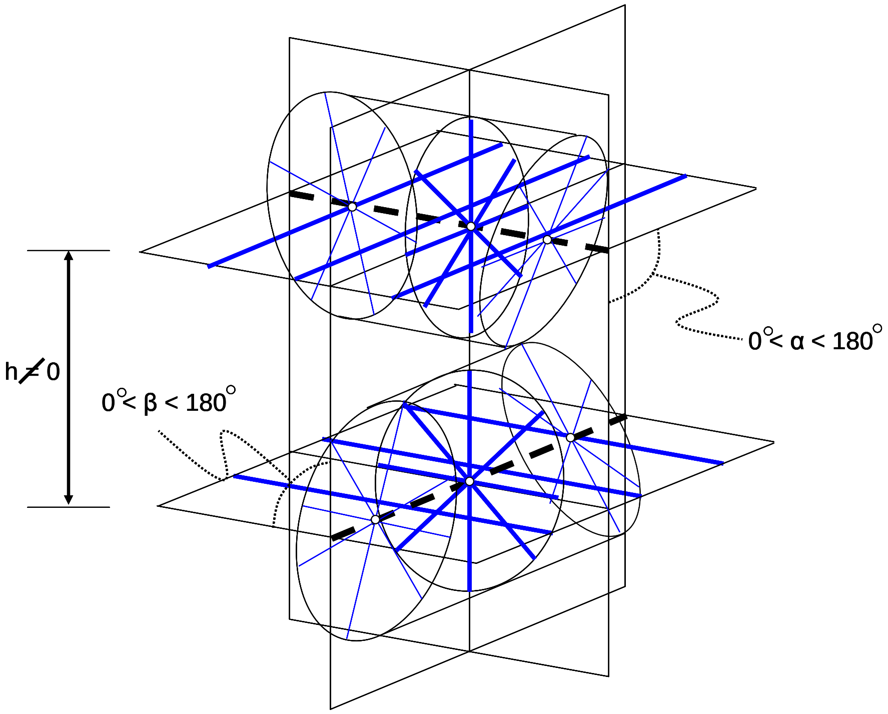


Figure 8.35: Constraint space of Case 4, Type 3

Figure 8.36 shows how the freedom and constraint spaces of this system fit together. The two skew pure rotations are the lines (thick dashed black) that run through the center of the two constraint disk tubes.

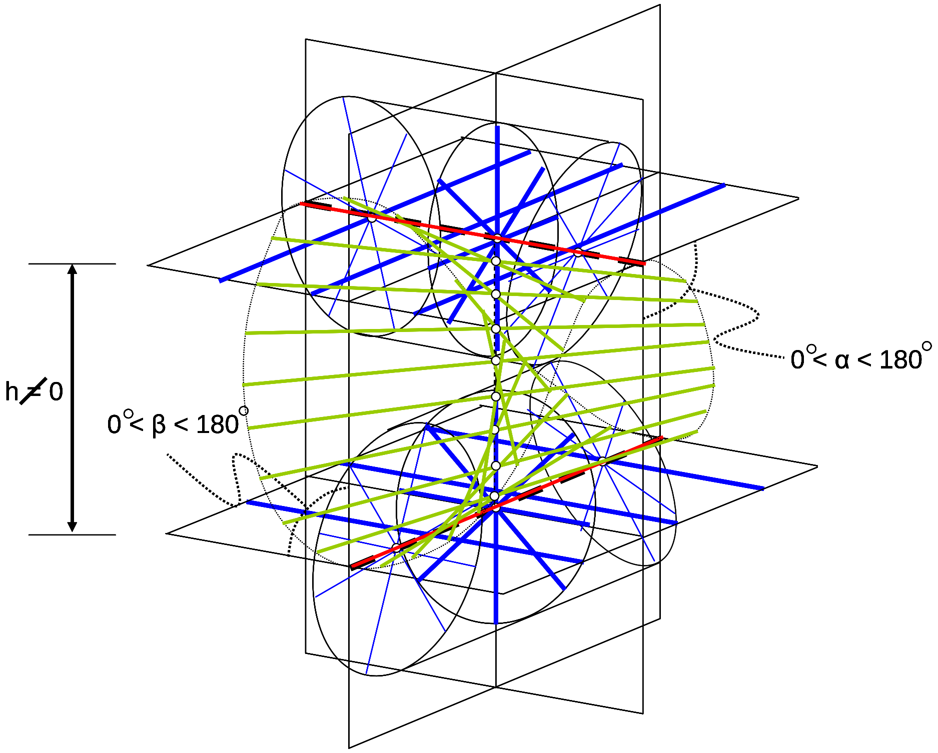


Figure 8.36: Freedom space (red and green) and constraint space (blue) of Case 4, Type 3 together.

Seven sub-constraint spaces exist within this constraint space. The first is shown in **Figure 8.37**. It consists of two constraint sets, a plane of parallel constraint lines and a disk of constraint lines. The angle, α, between the plane of this disk and the plane of parallel constraint lines must be greater than zero degrees but less than 180 degrees. Instructions for choosing the non-redundant constraints are included in the figure.

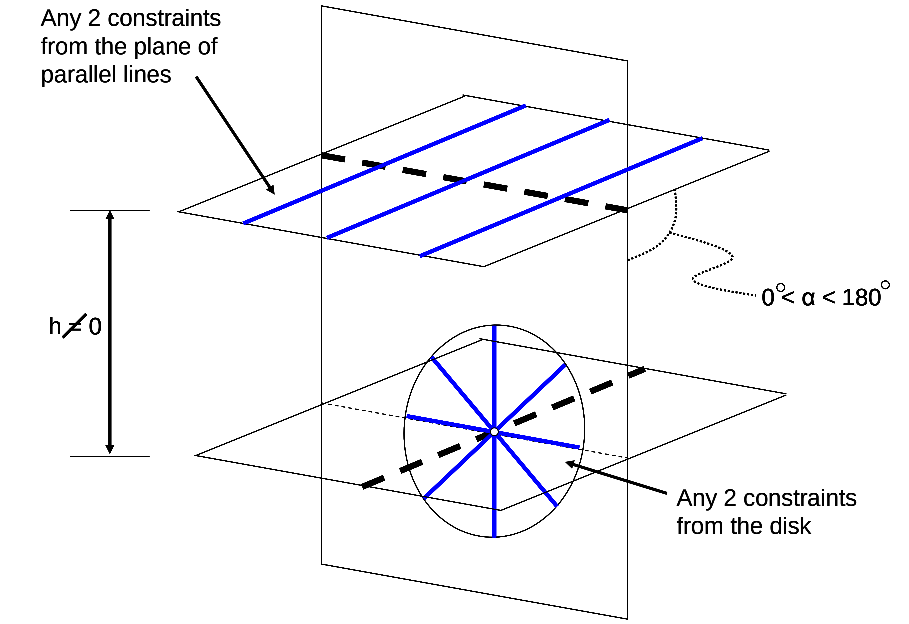

Any 2 constraints from the plane of parallel lines
h ≠ 0
0° < α < 180°
Any 2 constraints from the disk

Figure 8.37: First sub-constraint space of Case 4, Type 3

The second sub-constraint space is shown in **Figure 8.38**. It consists of two constraint set disks as shown in the figure. Instructions for choosing the non-redundant constraints are also included.

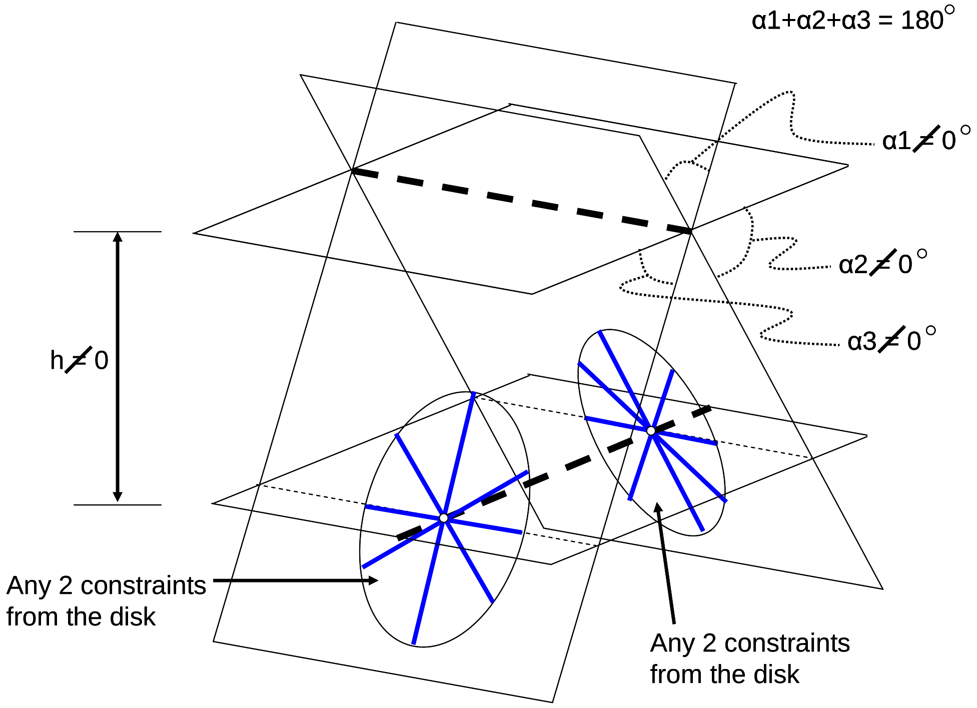
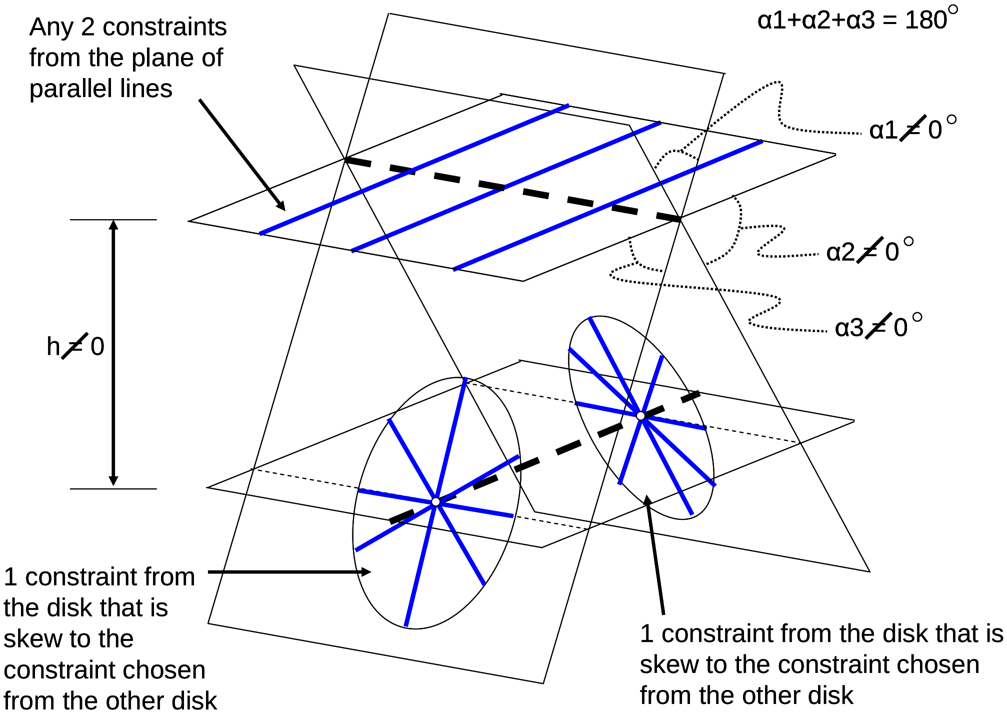

α1+α2+α3 = 180°
α1 ≠ 0°
α2 ≠ 0°
α3 ≠ 0°
h ≠ 0
Any 2 constraints from the disk
Any 2 constraints from the disk

Figure 8.38: Second sub-constraint space of Case 4, Type 3

The third sub-constraint space is shown in **Figure 8.39**. It consists of three constraint sets, two disks and a plane of parallel constraint lines. Instructions for choosing the non-redundant constraints are shown in the figure.



Any 2 constraints from the plane of parallel lines

1 constraint from the disk that is skew to the constraint chosen from the other disk

1 constraint from the disk that is skew to the constraint chosen from the other disk

Figure 8.39: Third sub-constraint space of Case 4, Type 3

The fourth sub-constraint space is shown in **Figure 8.40**. It consists of three constraint set disks, two of which are intersected at their center points by the same dashed black line shown in the figure. The disk intersected at its center point by the other dashed black line shares two lines with these two disks. These shared lines shown in orange are the intersection lines of the planes shown in the figure. The instructions for choosing non-redundant constraints are also shown in this figure.

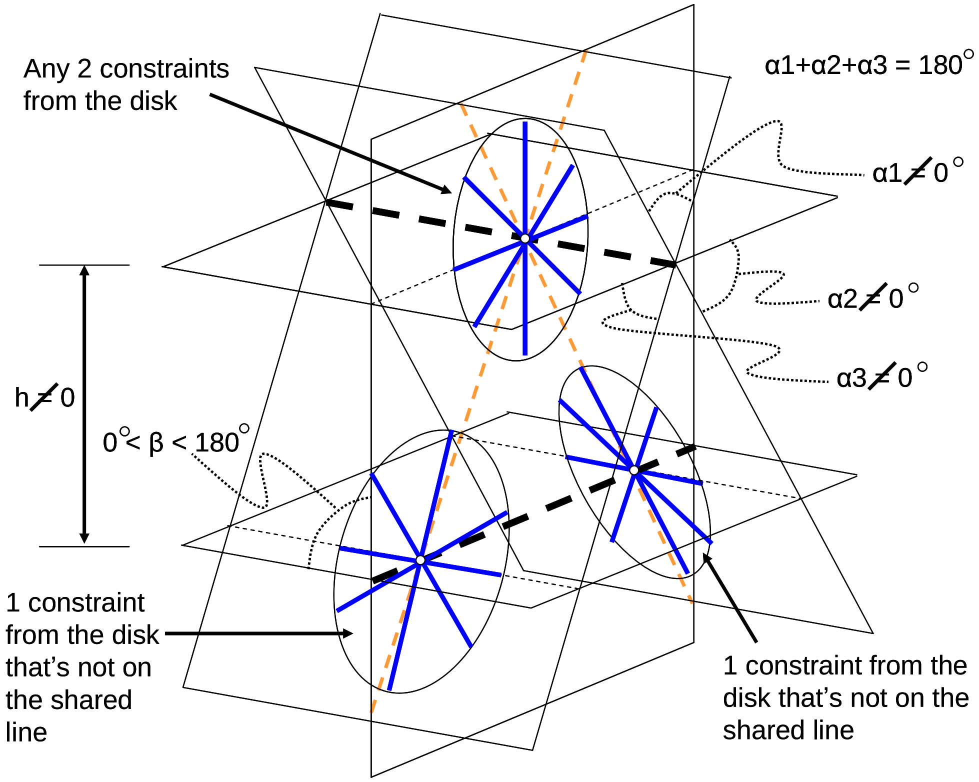

Any 2 constraints from the disk
α1+α2+α3 = 180°
α1 ≠ 0°
α2 ≠ 0°
α3 ≠ 0°
h ≠ 0
0°< β < 180°
1 constraint from the disk that's not on the shared line
1 constraint from the disk that's not on the shared line

Figure 8.40: Fourth sub-constraint space of Case 4, Type 3. Shared lines are dashed orange.

The fifth sub-constraint space is shown in **Figure 8.41**. It contains the same constraint sets as the sets within the fourth sub-constraint space, but it is considered a different sub-constraint space because it consists of a different set of instructions for guiding the designer in selecting a different set of non-redundant constraints.

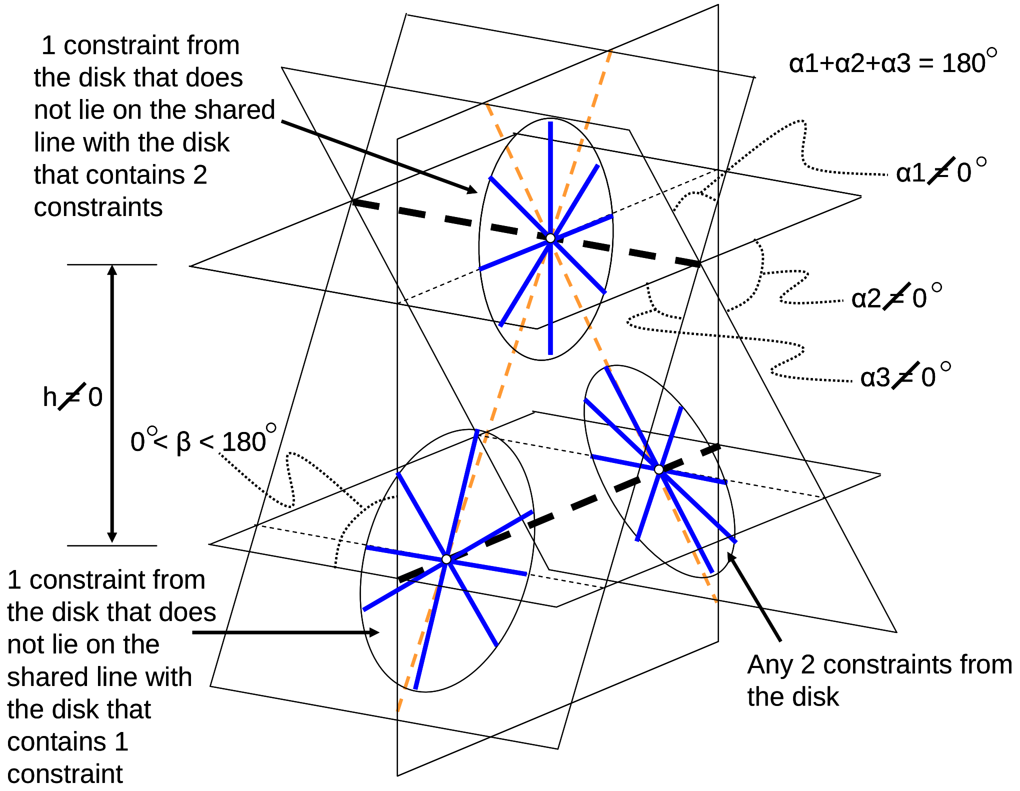

1 constraint from the disk that does not lie on the shared line with the disk that contains 2 constraints
α1+α2+α3 = 180°
α1 ≠ 0°
α2 ≠ 0°
α3 ≠ 0°
h ≠ 0
0° < β < 180°
1 constraint from the disk that does not lie on the shared line with the disk that contains 1 constraint
Any 2 constraints from the disk

Figure 8.41: Fifth sub-constraint space of Case 4, Type 3. Shared lines are dashed orange.

The sixth sub-constraint space is shown in **Figure 8.42**. It consists of three constraint set disks all of which are intersected at their center points by the same dashed black line as shown in the figure. The instructions for choosing non-redundant constraints are also shown in this figure.

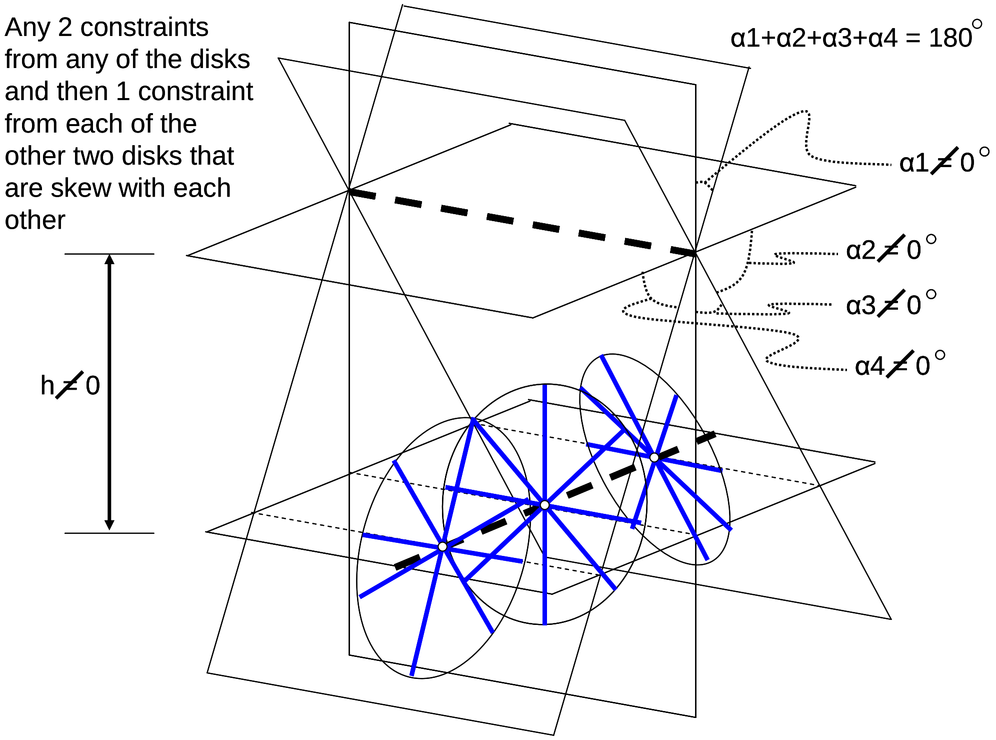

Any 2 constraints from any of the disks and then 1 constraint from each of the other two disks that are skew with each other

Figure 8.42: Sixth sub-constraint space of Case 4, Type 3

The seventh sub-constraint space is shown in **Figure 8.43**. It consists of four constraint set disks, two of which are intersected at their center points by one of the dashed black lines. The other two disks are intersected at their center points by the other dashed black line as shown in the figure. Each disk shares a line with the two disks on the opposite dashed black line. These shared lines shown in orange are the intersection lines of the planes shown in the figure. Instructions for choosing the non-redundant constraints are also shown in this figure.

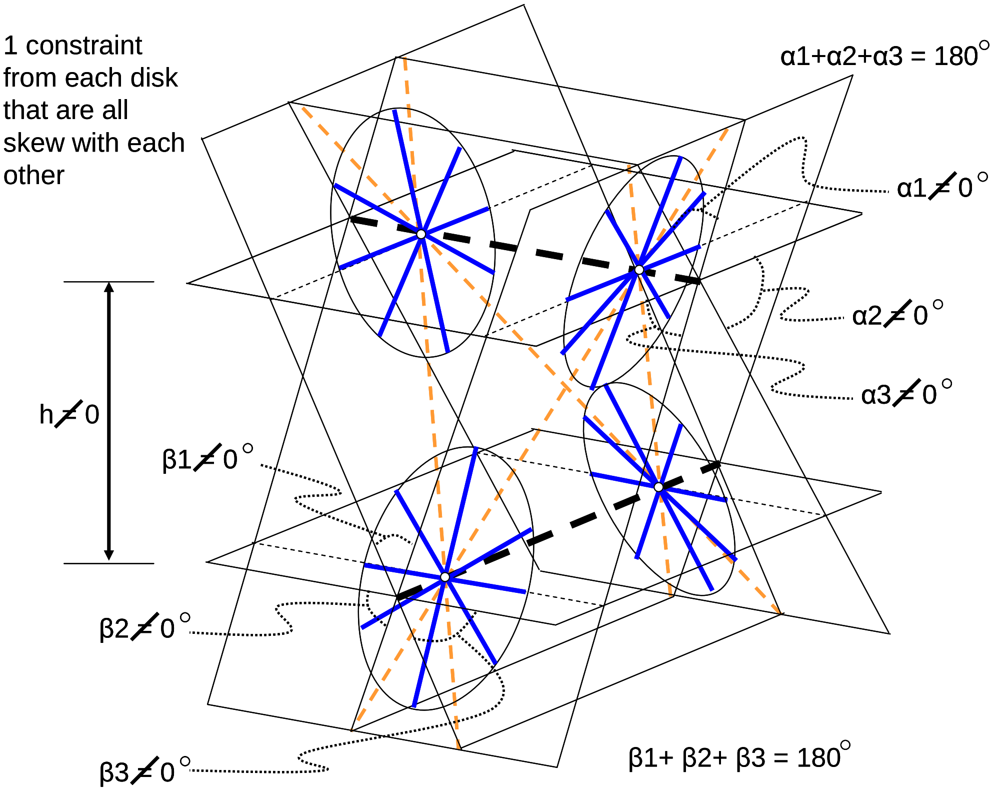


Figure 8.43: Seventh sub-constraint space of Case 4, Type 3. Shared lines are dashed orange.

8.3.4 Case 4, Type 4:

This section describes the unique constraint space with its sub-constraint spaces for a system with the freedom space described in **Section 8.1.1** and shown in **Figure 8.2**. For completeness, it is shown again here in **Figure 8.44**. In this freedom space, the pure rotational freedom line is coincident with the screw lines. All of these lines are oriented in the same direction as the pure translation.

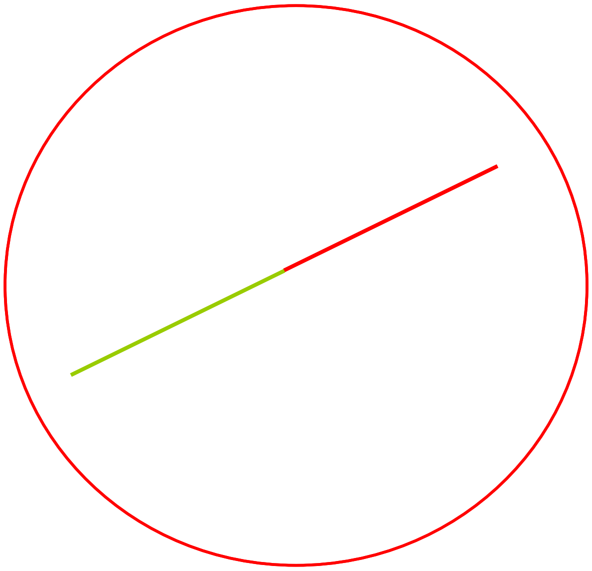


Figure 8.44: Freedom space of Case 4, Type 4

In order to determine this freedom space's complementary constraint space, Blanding's Rule of Complementary Patterns may be applied to determine every constraint line that intersects the pure rotational freedom line and the pure rotational hoop at least once. If these two pure rotational twists are complemented by constraint lines, one can also deduce that the same constraint lines will also complement the screw lines since the screws are linear combinations of these two pure rotational twists. The resulting constraint space is shown in **Figure 8.45**. The constraint space consists of an infinite number of disks of constraint lines. The plane of each disk is parallel to each other and the line that intersects all the disks through their center points is perpendicular to the planes of these disks.

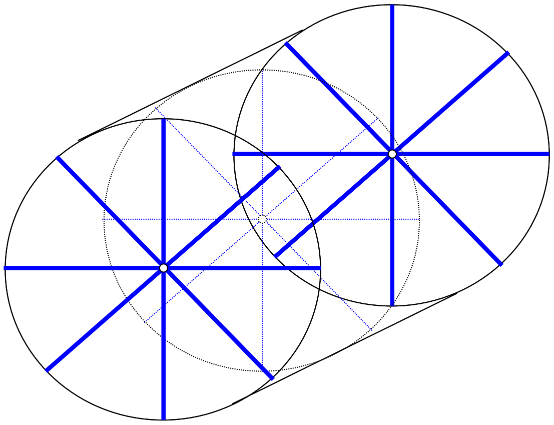


Figure 8.45: Constraint space of Case 4, Type 4

Figure 8.46 shows how the freedom and constraint spaces of this system fit together. The line that intersects all the disks at their center points is coincident with the line of twists in the freedom space.

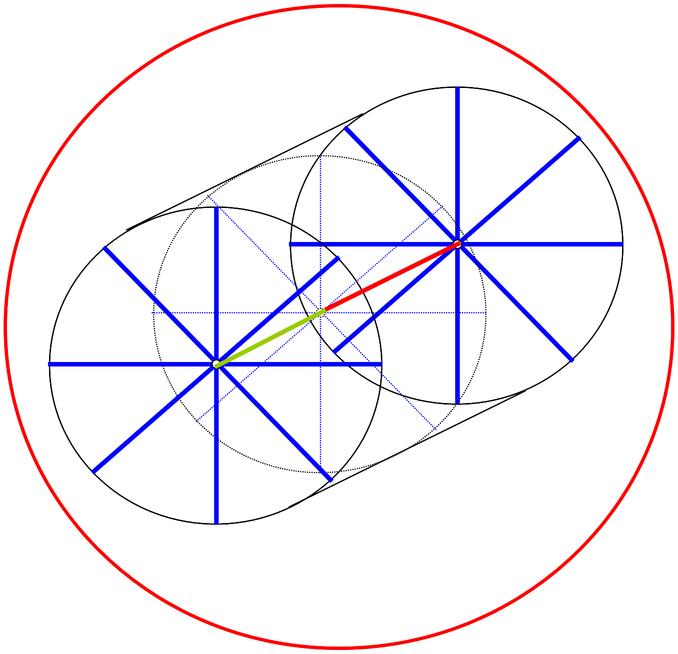


Figure 8.46: Freedom space (red and green) and constraint space (blue) of Case 4, Type 4 together.

Five sub-constraint spaces exist within this constraint space. The first is shown in **Figure 8.47**. It consists of two planar constraint sets of parallel lines. Every parallel constraint line on either of these planes is orthogonal to the intersection line of the two planes. The angle, θ, between the two planes of parallel constraint lines must be greater than zero degrees and less than 180 degrees. If this angle equals one of these values, the space becomes Case 2, Type 2. Instructions for choosing the non-redundant constraints from this sub-constraint space are included in the figure.

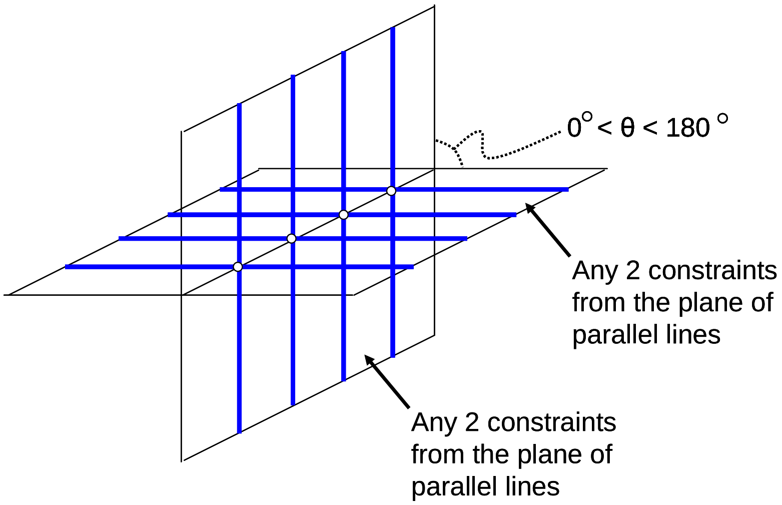

0° < θ < 180 °
Any 2 constraints from the plane of parallel lines
Any 2 constraints from the plane of parallel lines

Figure 8.47: First sub-constraint space of Case 4, Type 4

The second sub-constraint space is shown in **Figure 8.48**. It consists of two constraint set disks. The planes of these disks are parallel and separated by a non-zero distance, *d*. Instructions for choosing the non-redundant constraints are also included in the figure.

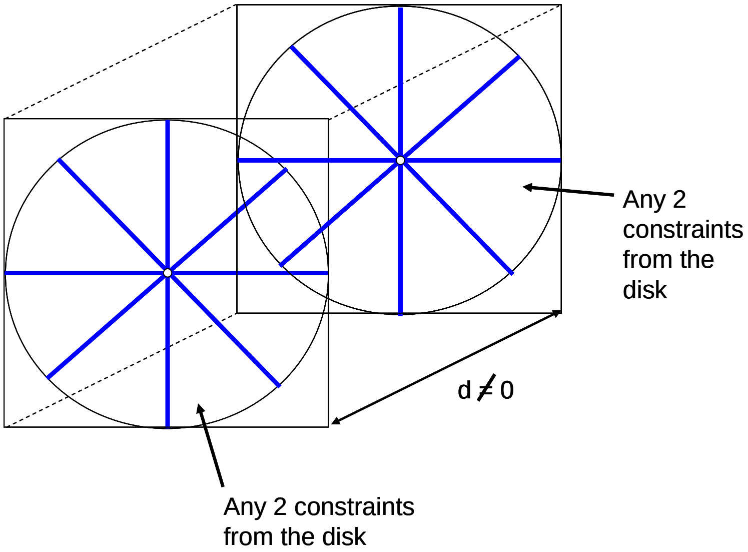

Any 2 constraints from the disk
d ≠ 0
Any 2 constraints from the disk

Figure 8.48: Second sub-constraint space of Case 4, Type 4

The third sub-constraint space is shown in **Figure 8.49**. It consists of three constraint sets, a disk and two planes of parallel constraint lines. The parallel lines on each plane are perpendicular to the planes' intersection line. The plane of the disk is also perpendicular to this intersection line.

Instructions for choosing the four non-redundant constraints from this sub-constraint space are shown in the figure.

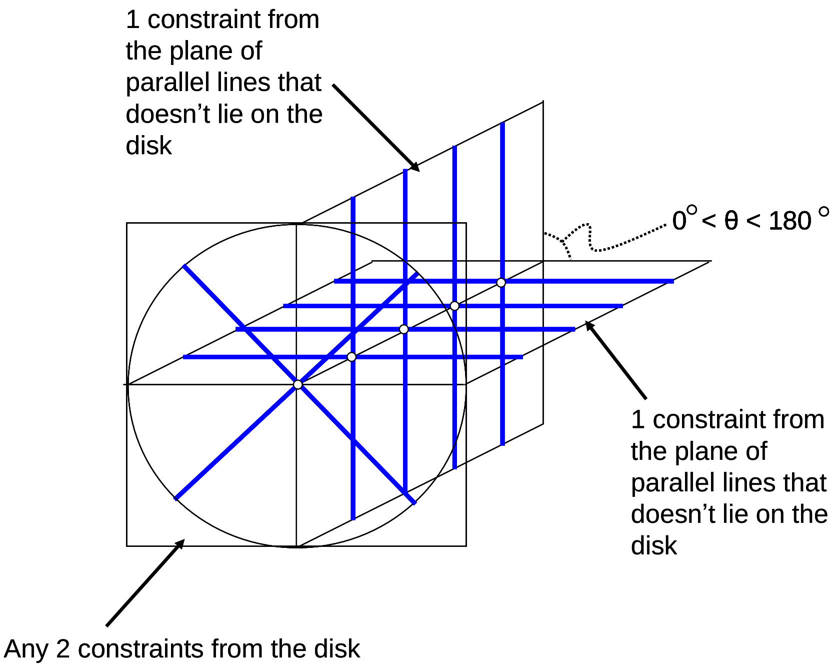
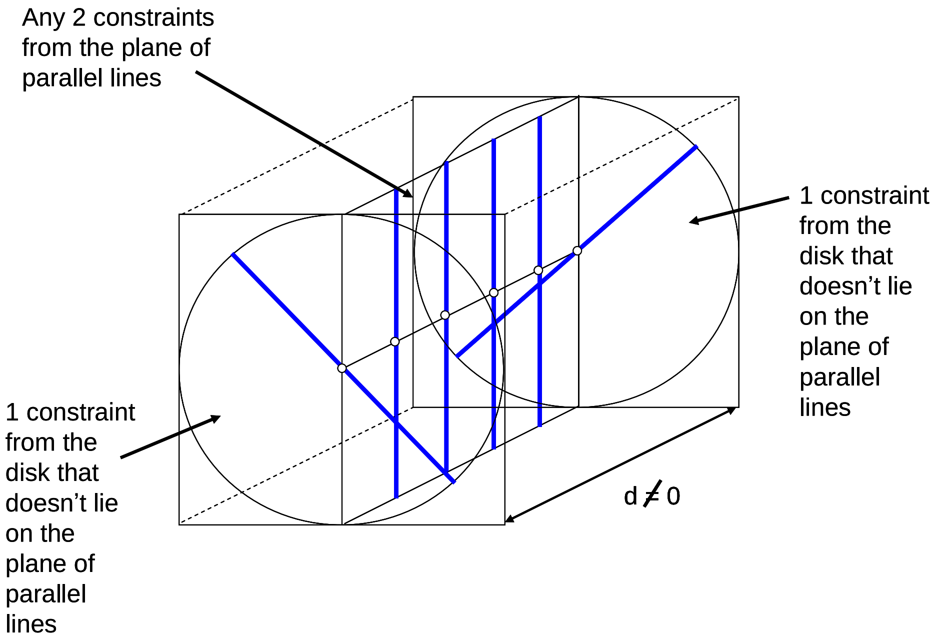

1 constraint from the plane of parallel lines that doesn't lie on the disk
$0^\circ < \theta < 180^\circ$
1 constraint from the plane of parallel lines that doesn't lie on the disk
Any 2 constraints from the disk

Figure 8.49: Third sub-constraint space of Case 4, Type 4

The fourth sub-constraint space is shown in **Figure 8.50**. It consists of three constraint sets, two disks and a plane containing parallel lines. The planes of the disks are parallel and are separated by a non-zero distance, *d*, as shown in the figure. Instructions for choosing the non-redundant constraints are also shown.

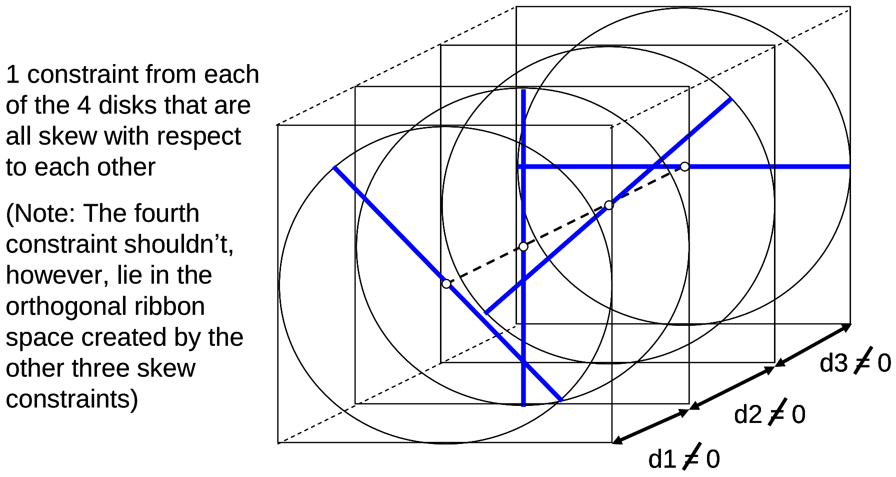

Any 2 constraints from the plane of parallel lines

1 constraint from the disk that doesn't lie on the plane of parallel lines

1 constraint from the disk that doesn't lie on the plane of parallel lines

Figure 8.50: Fourth sub-constraint space of Case 4, Type 4

The fifth sub-constraint space is shown in **Figure 8.51**. It consists of four constraint set disks. The planes of these four disks are parallel and the dashed black line that passes through their center points is perpendicular to them. Instructions are provided for guiding the designer in selecting constraints that are non-redundant.


1 constraint from each of the 4 disks that are all skew with respect to each other

(Note: The fourth constraint shouldn't, however, lie in the orthogonal ribbon space created by the other three skew constraints)

Figure 8.51: Fifth sub-constraint space of Case 4, Type 4

8.3.5 Case 4, Type 5:

This section describes the unique constraint space with its sub-constraint spaces for a system with the freedom space described in **Section 8.1.1** and shown in **Figure 8.5**. For completeness, it is shown again here in **Figure 8.52**. In this freedom space, the pure rotational freedom line is parallel to screw lines on a common plane. A pure translation also points in a direction that is not perpendicular to this plane.

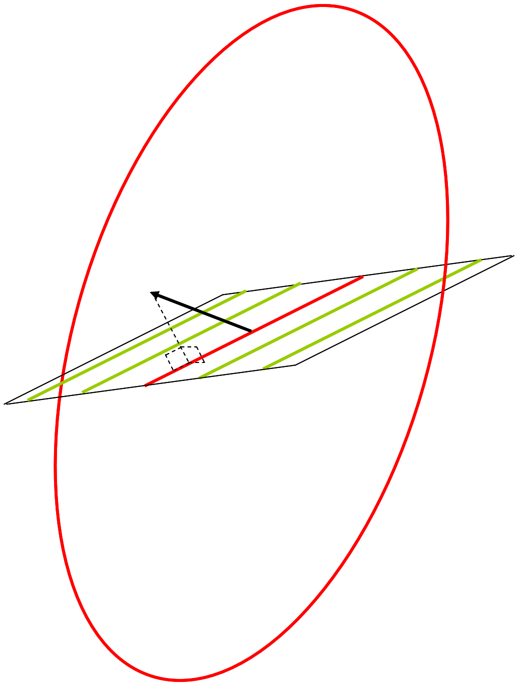


Figure 8.52: Freedom space of Case 4, Type 5

In order to determine this freedom space's complementary constraint space, Blanding's Rule of Complementary Patterns may be applied to determine every constraint line that intersects the pure rotational freedom line and the pure rotational hoop at least once. If these two pure rotational twists are complemented by constraint lines, one can also deduce that the same constraint lines will also complement the screw lines since the screws are linear combinations of these two pure rotational twists. The resulting constraint space is shown in **Figure 8.53**. The constraint space consists of an infinite number of disks of constraint lines. The plane of each disk is parallel to the plane of the other disks, but the line that intersects all the disks through their center points is not perpendicular to the planes of these disks. If this line were perpendicular to the plane of these disks, the constraint space would belong to Case 4, Type 4.

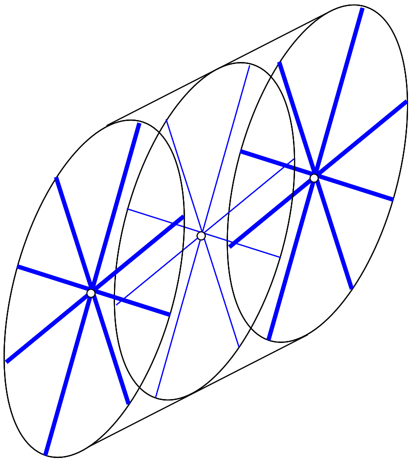


Figure 8.53: Constraint space of Case 4, Type 5

Figure 8.54 shows how the freedom and constraint spaces of this system fit together. The line that intersects all the disks at their center points is coincident with the pure rotational freedom line in the freedom space.

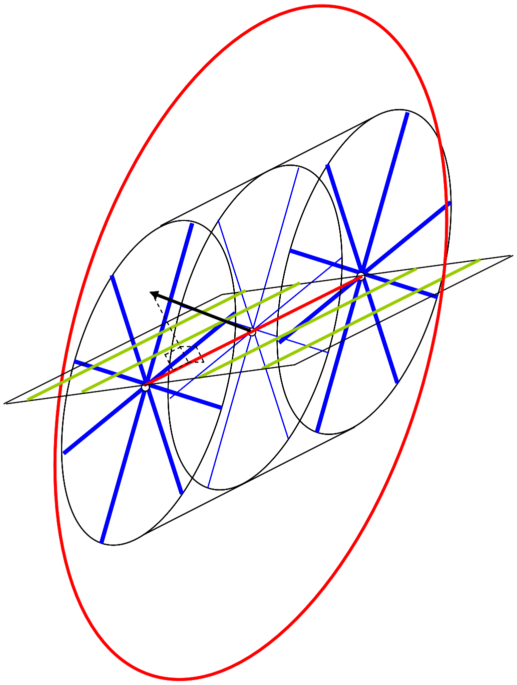


Figure 8.54: Freedom space (red and green) and constraint space (blue) of Case 4, Type 5 together.

There are five sub-constraint spaces within this constraint space. The first is shown in **Figure 8.55**. It consists of two planar constraint sets of parallel lines. The parallel constraint lines that lie on the vertical plane intersect the intersection line of the two planes at an angle of α as shown in the figure. The parallel constraint lines that lie on the horizontal plane intersect the intersection line of the two planes at an angle of β also shown in the figure. Both of these angles must be greater than zero degrees and less than 180 degrees. If one of these angles equals zero or 180 degrees, the space belongs to Case 4, Type 2. If both angles simultaneously equal zero or 180 degrees, the space belongs to Case 3, Type 5. If these two angles simultaneously equal 90 degrees, the space belongs to Case 4, Type 4. The angle, θ, between the two planes of parallel constraint lines must also be greater than zero degrees and less than 180 degrees. If θ equals zero or 180 degrees, the space becomes Case 3, Type 1. Instructions for choosing the non-redundant constraints from this sub-constraint space are included in the figure.

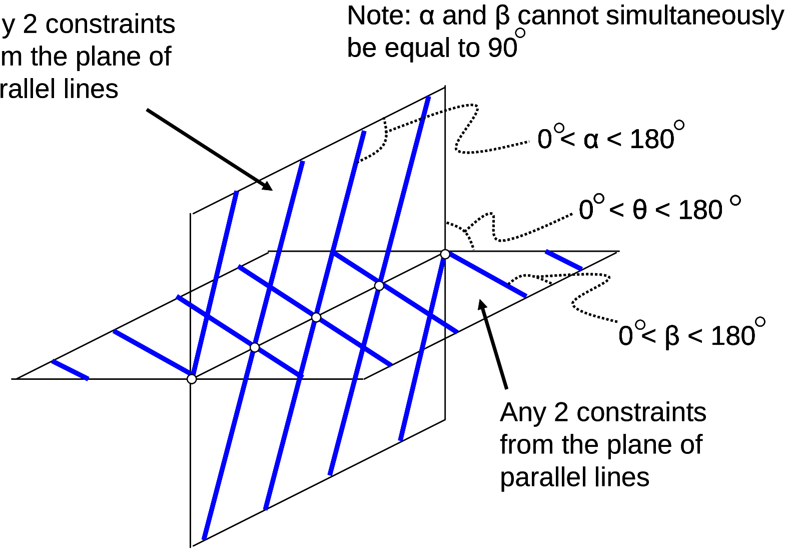

Any 2 constraints from the plane of parallel lines
Note: α and β cannot simultaneously be equal to 90°
0° < α < 180°
0° < θ < 180 °
0° < β < 180°
Any 2 constraints from the plane of parallel lines

Figure 8.55: First sub-constraint space of Case 4, Type 5.

The second sub-constraint space is shown in **Figure 8.56**. It consists of two constraint set disks. The planes of these disks are parallel and separated by a non-zero distance, *d*. The instructions for choosing non-redundant constraints are also included in the figure.

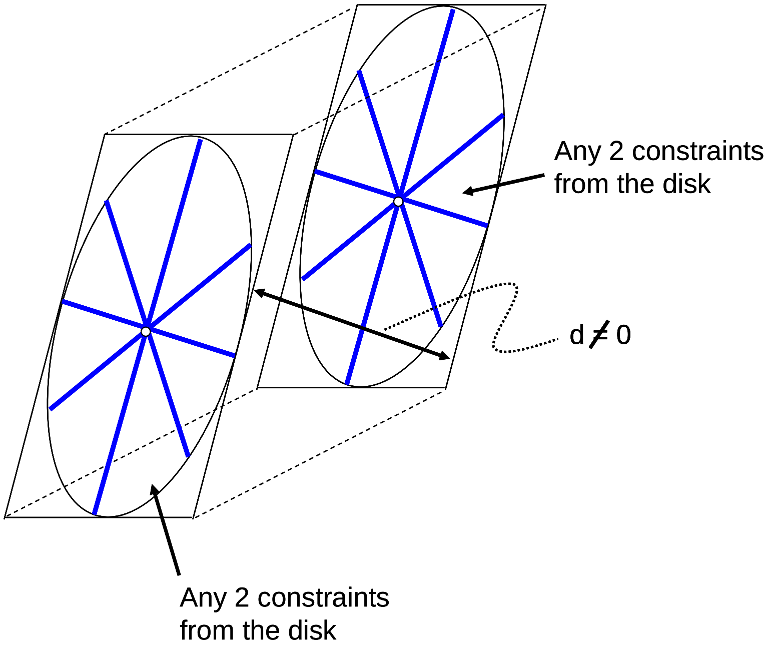


Figure 8.56: Second sub-constraint space of Case 4, Type 5.

The third sub-constraint space is shown in **Figure 8.57**. It consists of three constraint sets, a disk and two planes of parallel constraint lines. The parallel constraint lines that lie on the vertical plane intersect the intersection line of the two planes at an angle of α. The parallel constraint lines that lie on the horizontal plane intersect the intersection line of the two planes at an angle of β. Both of these angles must be greater than zero degrees and less than 180 degrees. They may also not simultaneously both equal 90 degrees. The plane of the disk is perpendicular to the vectors that result from taking the cross product of constraint orientation vectors ($\vec{f}$) from the planes. Instructions for choosing the four non-redundant constraints from this sub-constraint space are shown in the figure.

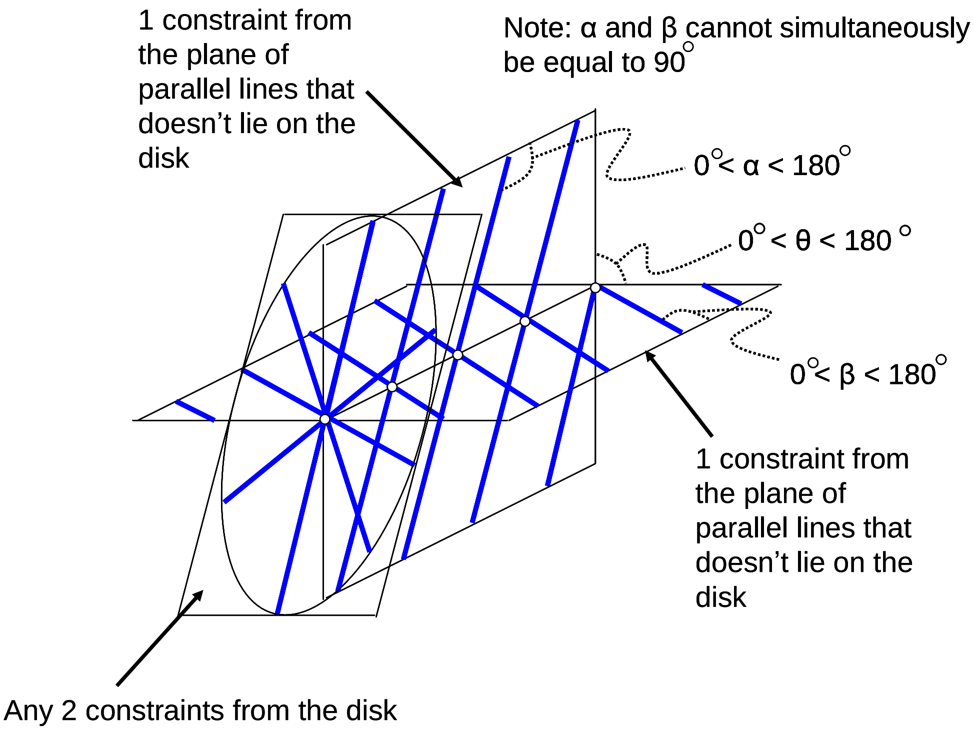

1 constraint from the plane of parallel lines that doesn’t lie on the disk
Note: α and β cannot simultaneously be equal to 90°
0°< α < 180°
0°< θ < 180 °
0°< β < 180°
1 constraint from the plane of parallel lines that doesn’t lie on the disk
Any 2 constraints from the disk

Figure 8.57: Third sub-constraint space of Case 4, Type 5

The fourth sub-constraint space is shown in **Figure 8.58**. It consists of three constraint sets, two disks and a plane containing parallel lines. The planes of the disks are parallel and are separated by a non-zero distance, *d*, as shown in the figure. The parallel lines on the plane are parallel to the planes of these disks. Instructions for choosing the non-redundant constraints are also shown.

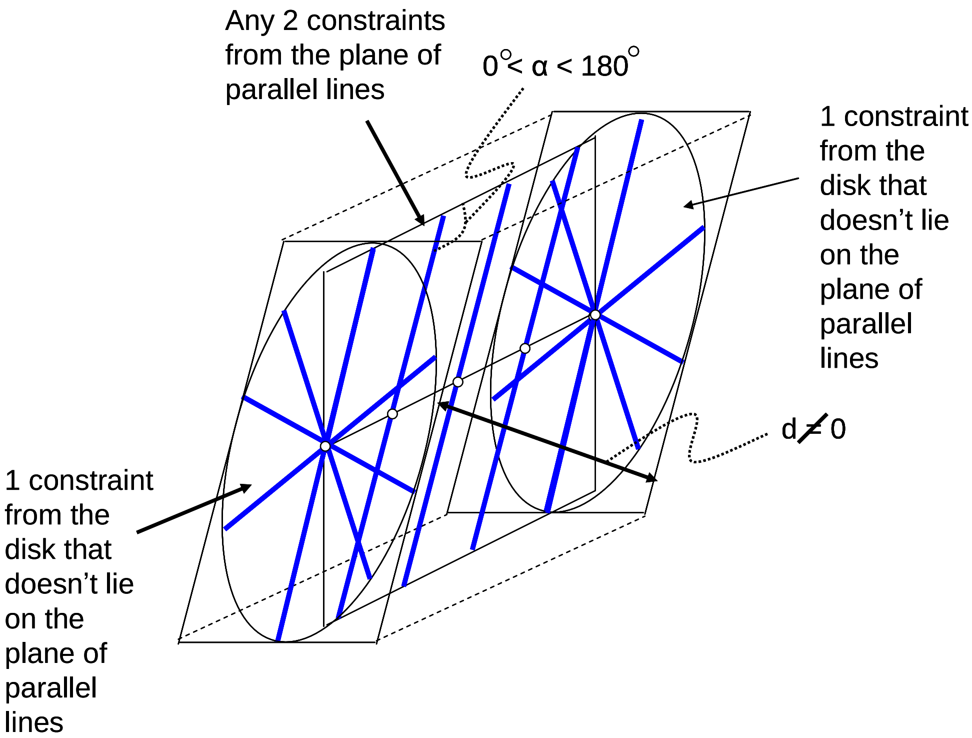

Any 2 constraints from the plane of parallel lines
0° < α < 180°
1 constraint from the disk that doesn't lie on the plane of parallel lines
d ≠ 0
1 constraint from the disk that doesn't lie on the plane of parallel lines

Figure 8.58: Fourth sub-constraint space of Case 4, Type 5

The fifth sub-constraint space is shown in **Figure 8.59**. It consists of four constraint set disks. The planes of these four disks are parallel and the dashed black line that passes through their center points is not perpendicular to them. Instructions are provided for guiding the designer in selecting constraints that are non-redundant.

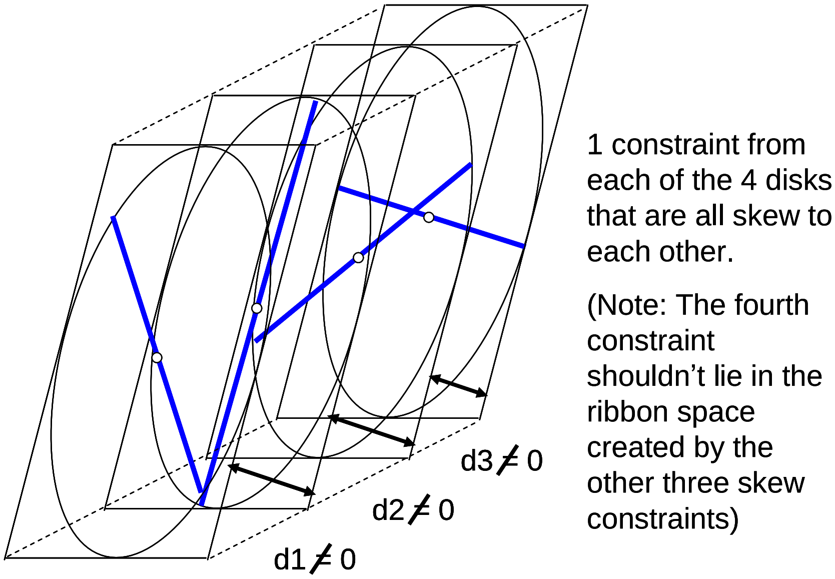

1 constraint from each of the 4 disks that are all skew to each other.
(Note: The fourth constraint shouldn't lie in the ribbon space created by the other three skew constraints)
d3 ≠ 0
d2 ≠ 0
d1 ≠ 0

Figure 8.59: Fifth sub-constraint space of Case 4, Type 5

8.3.6 Case 4, Type 6:

This section describes the unique constraint space with its sub-constraint spaces for a system with the freedom space described in **Section 8.1.1** and shown in **Figure 8.4**. For completeness it is shown again here in **Figure 8.60**. In this freedom space only a single pure rotational freedom line at infinity shown as a pure rotational hoop exists. A plane of parallel screws of equal pitch value is perpendicular to the direction of this pure translation.

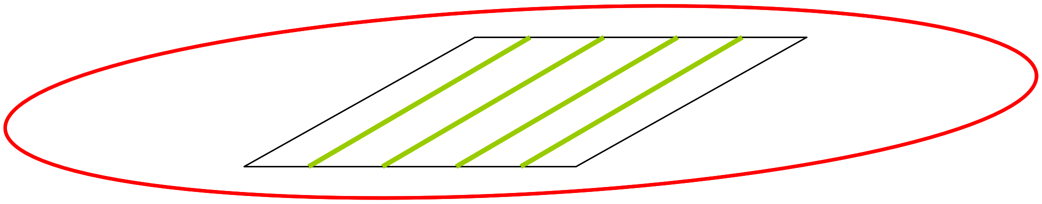


Figure 8.60: Freedom space of Case 4, Type 6

Finding this system's complementary constraint space is not as easy as it has been for past types. For this system there aren't two independent pure rotational freedom lines to apply Blanding's Rule of Complementary Patterns to for finding the constraint lines. It is known that every constraint line does intersect the pure rotational hoop at infinity and, therefore, must lie on planes with normal vectors that are parallel to the normal vector of the hoop. But these constraint lines must also complement the planar set's parallel screws in accordance with **Equation (3.13)** from **Chapter 3**. The constraint space that complements this freedom space is, therefore, shown in **Figure 8.61**. It consists of an infinite number of planar constraint sets each of which contains an infinite number of parallel constraint lines. A middle plane of parallel constraint lines exists that is coincident with the plane of screws from the freedom space. The parallel constraint lines on this plane are orthogonal to the screw lines. The orientation angle, θ, with respect to the direction of the screw lines (dashed black line) for the parallel constraint lines separated from the middle plane by a distance, *d*, may be solved using **Equation (3.13)** for a given pitch value, *p*, of the screws from the freedom space. The planes infinitely far away from the middle plane each contain parallel constraint lines that approach being parallel with respect to each other and are orthogonal to the parallel constraint lines on the middle plane. Seen another way, the constraint space of this system is essentially an infinite number of hyperbolic paraboloids stacked on top of each other. The particular constraint space shown in **Figure 8.61** is for a freedom space of screws with a negative pitch value. This observation must be the case since the constraint lines form right-handed ribbons.

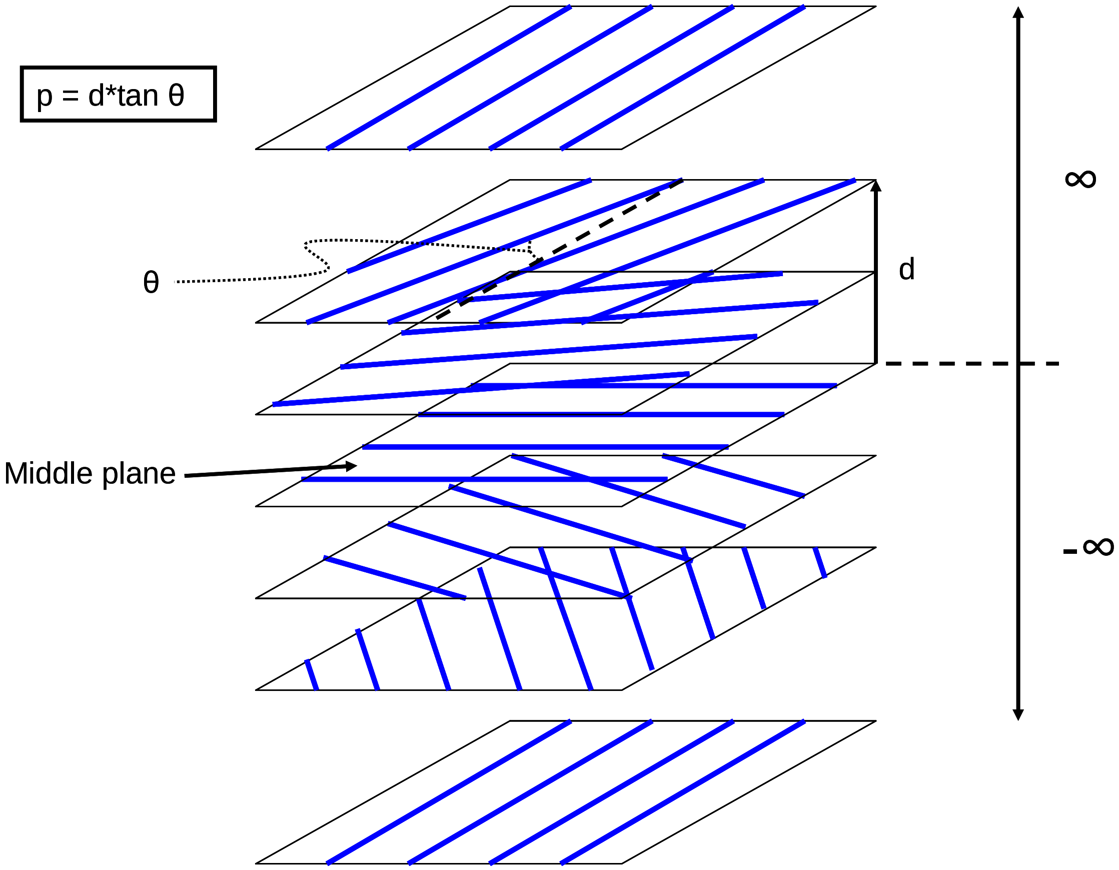


Figure 8.61: Constraint space of Case 4, Type 6

Figure 8.62 shows how the freedom and constraint spaces of this system fit together. The plane of parallel screws from the freedom space is coincident with the middle plane of parallel constraint lines from the constraint space.

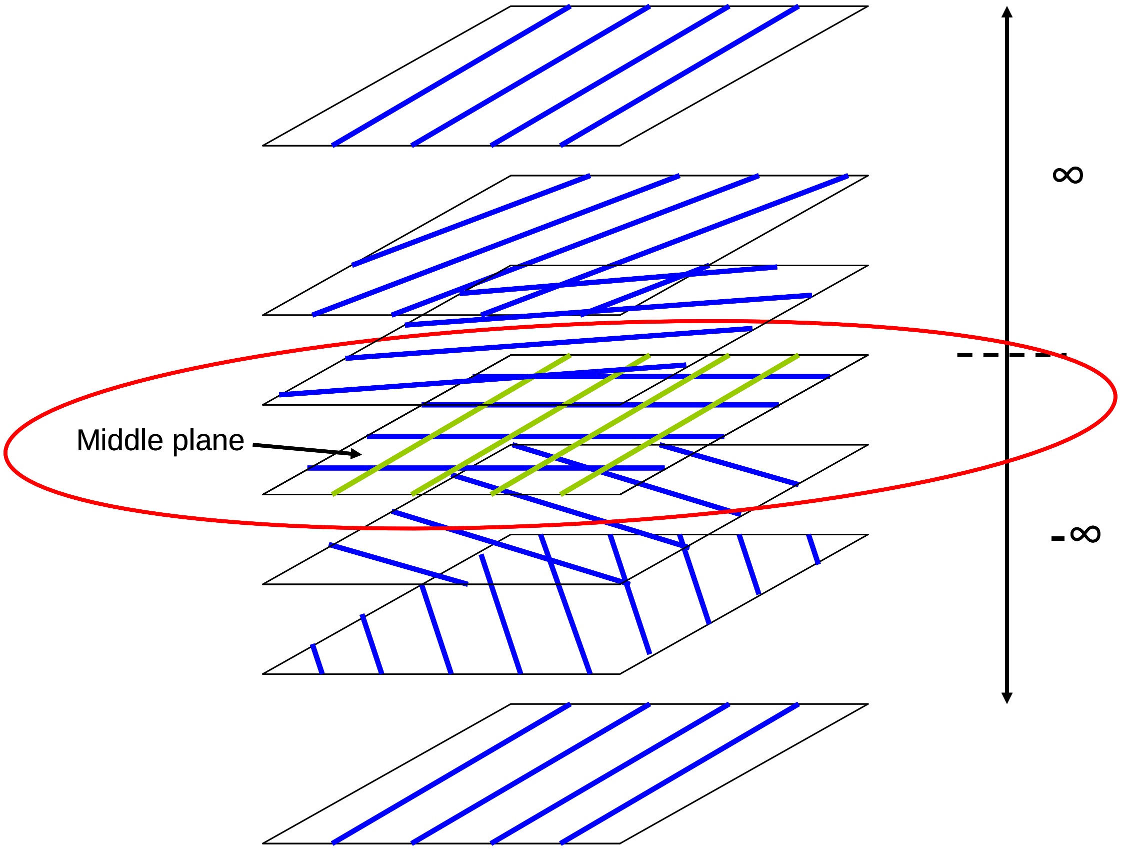


Figure 8.62: Freedom space (red and green) and constraint space (blue) of Case 4, Type 6 together.

Two sub-constraint spaces exist within this constraint space. The first is shown in **Figure 8.63**. It consists of three planar constraint sets of parallel constraint lines. Instructions for choosing the non-redundant constraints from this sub-constraint space are included in the figure.

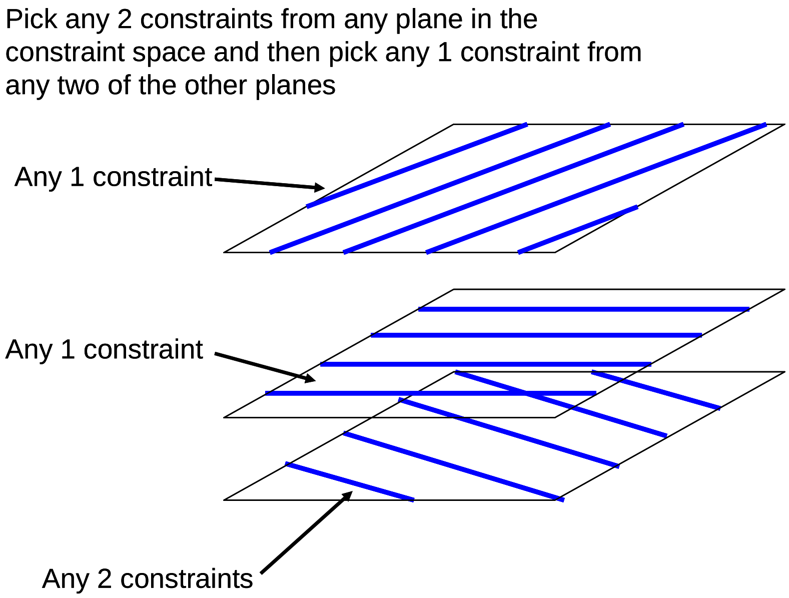

Pick any 2 constraints from any plane in the constraint space and then pick any 1 constraint from any two of the other planes
Any 1 constraint
Any 1 constraint
Any 2 constraints

Figure 8.63: First sub-constraint space of Case 4, Type 6

The second sub-constraint space is shown in **Figure 8.64**. It consists of four planar constraint sets of parallel constraint lines. Instructions for choosing the non-redundant constraints are included in the figure.

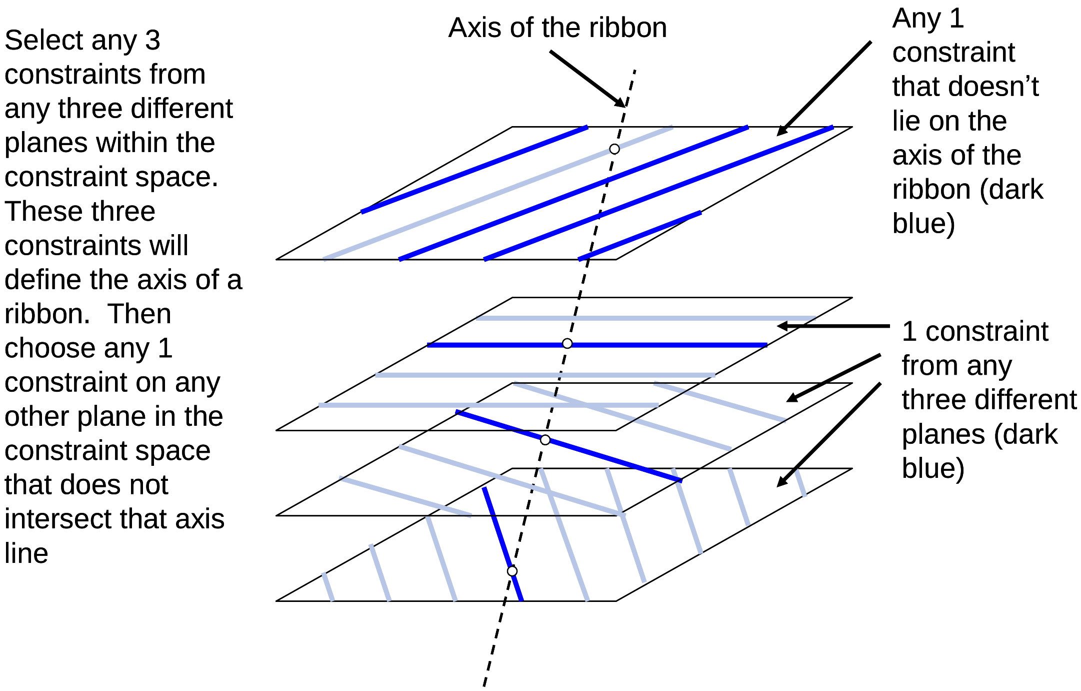

Select any 3 constraints from any three different planes within the constraint space. These three constraints will define the axis of a ribbon. Then choose any 1 constraint on any other plane in the constraint space that does not intersect that axis line
Axis of the ribbon
Any 1 constraint that doesn't lie on the axis of the ribbon (dark blue)
1 constraint from any three different planes (dark blue)

Figure 8.64: Second sub-constraint space of Case 4, Type 6

8.3.7 Case 4, Type 7:

This section describes the unique constraint space with its sub-constraint spaces for a system with the freedom space described in **Section 8.1.2** and shown in **Figure 8.10**. For completeness, it is shown again here in **Figure 8.65**. This freedom space is a cylindroid of twists with a principal generator that is a pure rotational freedom line. The rest of the twists within the cylindroid are screws with finite non-zero pitch values.

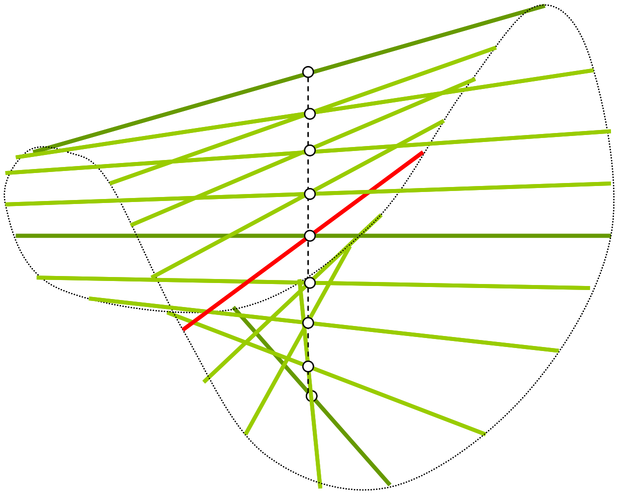


Figure 8.65: Freedom space of Case 4, Type 7

In order to determine this freedom space's complementary constraint space, note that apply Blanding's Rule of Complementary Patterns alone will not be sufficient. This is because there is only a single pure rotational freedom line within the freedom space and in order to find the system's complete constraint space, two independent twists from within the cylindroid must be found. Consider the cylindroid's two principal generators. Blanding's Rule of Complementary Patterns suggests that every constraint line must simultaneously intersect the pure rotational freedom principal generator at least once. These constraint lines must also satisfy **Equation (3.13)** for the other screw principal generator. The resulting constraint space that satisfies both of these conditions is shown in **Figure 8.66**.

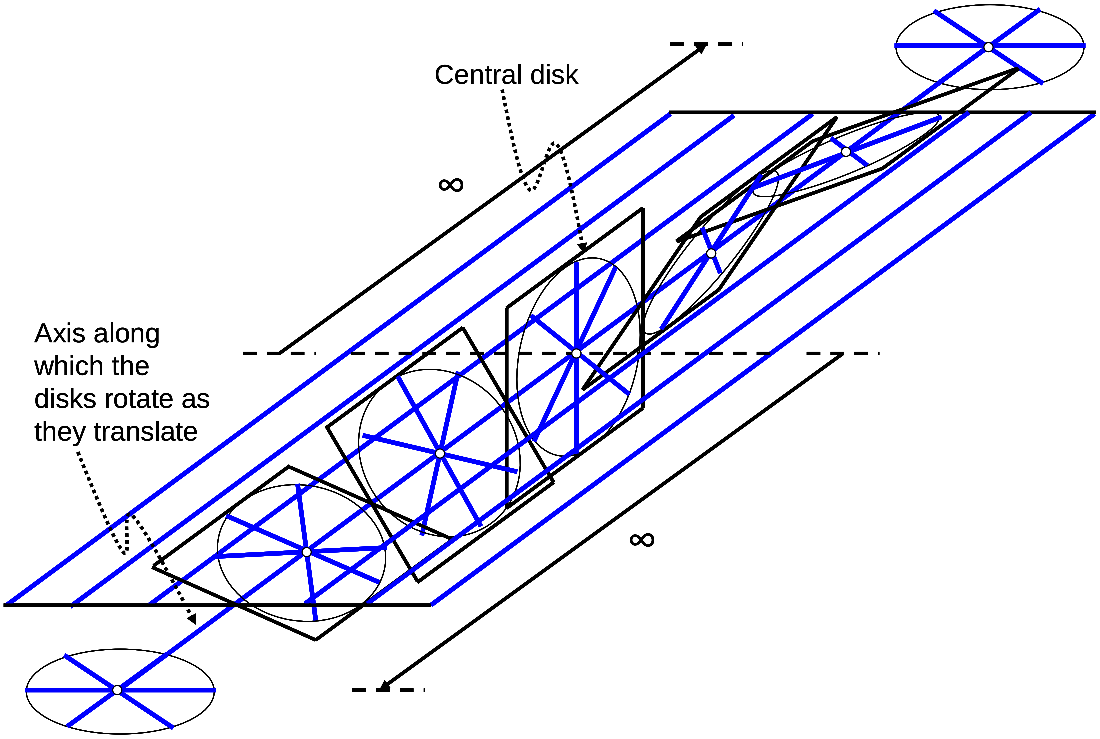


Figure 8.66: Constraint space of Case 4, Type 7

The constraint space consists of an infinite number of constraint disks and a plane of parallel constraint lines. The disks translate as they rotate along an axis that lies on the plane of parallel lines. The rate that the disks translate as they rotate depends on the pitch of the principal generator screw within the freedom space. These disks behave exactly like the lines that lie on the surface of orthogonal ribbons discussed in **Chapter 7**. One can, therefore, deduce from the findings in **Appendix F** that the double derivative of the rate that the disks translate as they rotate with respect to their position along the axis that they're translating along is a constant, *K*, that relates to the pitch, *p*, of the principal generator screw as **Equation (F.8)**. It is also important to note that every point along the axis of the rotating disks is a center point for a single disk that lies on a unique plane. The constraint space's central disk lies on a plane that is orthogonal to the plane of parallel constraint lines, and the disks that are infinitely far away from this central disk lie on planes that approach the plane of the parallel constraint lines as shown in the figure.

Figure 8.67 shows how the freedom and constraint spaces of this system fit together. The axis line of the rotating disks within the constraint space is coincident with the pure rotational principal generator line within the freedom space. The other screw principal generator line is orthogonal to the plane of the central disk within the constraint space and intersects it at its center point.

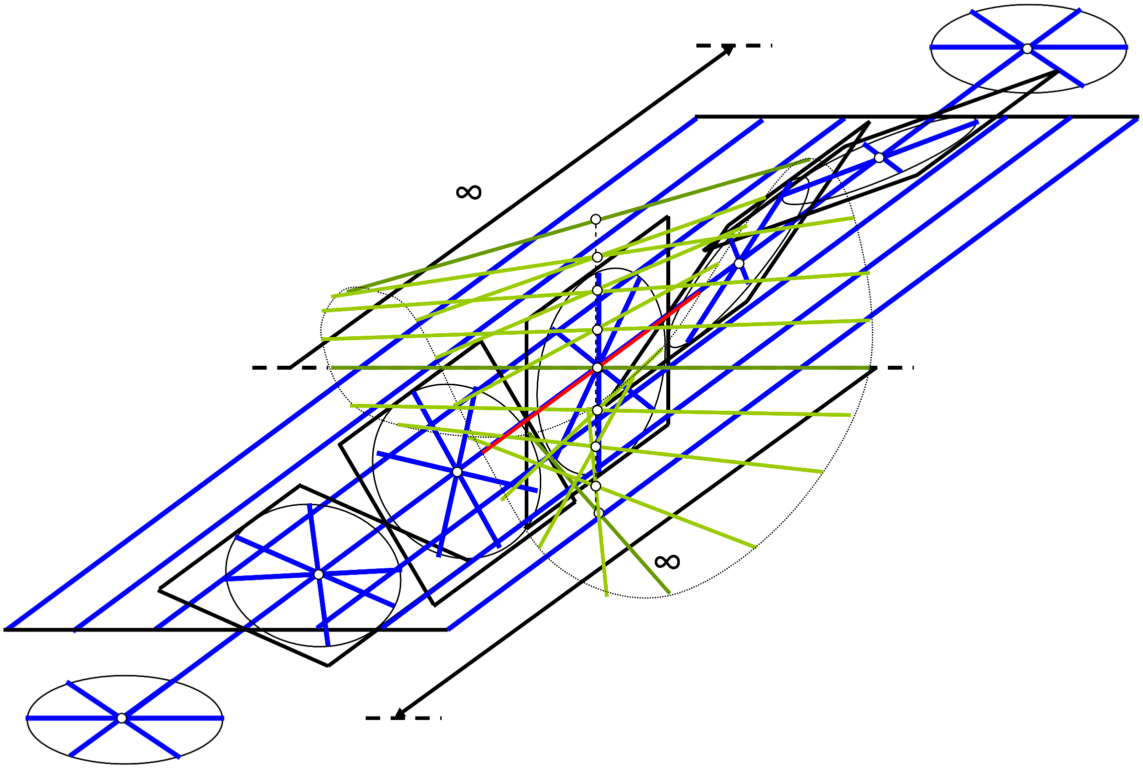


Figure 8.67: Freedom space (red and green) and constraint space (blue) of Case 4, Type 7 together.

Five sub-constraint spaces exist within this constraint space. The first is shown in **Figure 8.68**. It consists of a planar constraint set of parallel lines and two constraint set disks. These disks may be any two disks from within the constraint space of the system and don't have to be the two disks arbitrarily represented in the figure. Instructions for choosing the non-redundant constraints from this sub-constraint space are included in the figure.

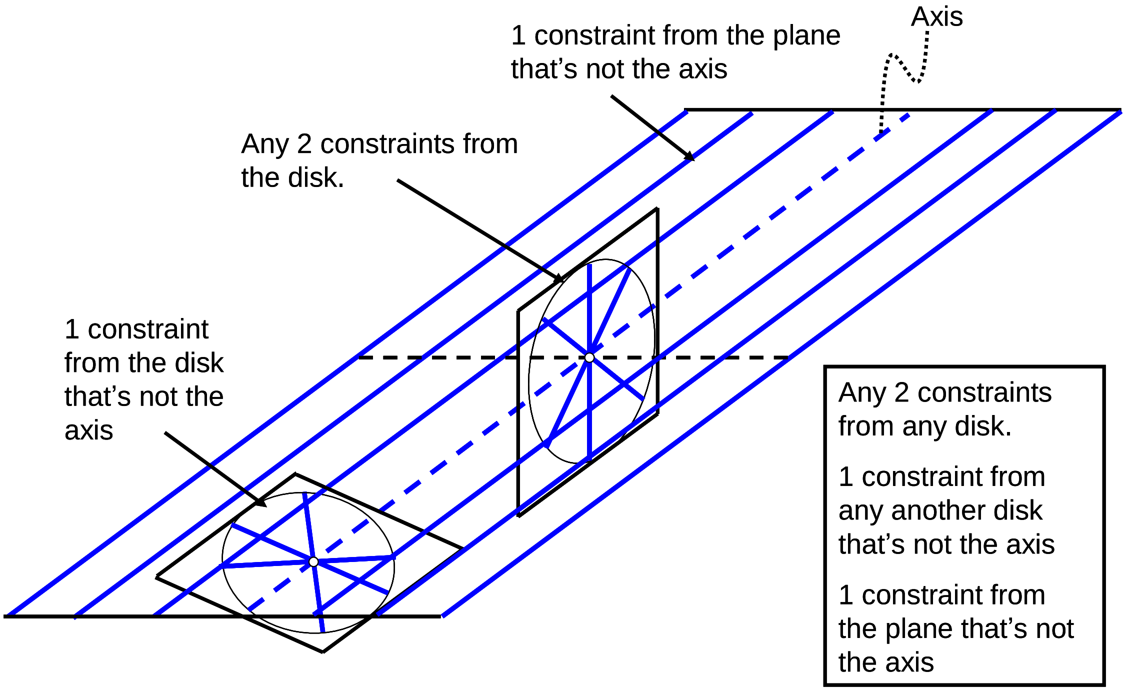

Axis
1 constraint from the plane that's not the axis
Any 2 constraints from the disk.
1 constraint from the disk that's not the axis
Any 2 constraints from any disk.
1 constraint from any another disk that's not the axis
1 constraint from the plane that's not the axis

Figure 8.68: First sub-constraint space of Case 4, Type 7

The second sub-constraint space is shown in **Figure 8.69**. It is identical in geometry to the first sub-constraint space but has different instructions for choosing a different set of non-redundant constraints and is, therefore, its own sub-constraint space. Again, any two disks from within the constraint space may be selected by the designer and not just the two disks shown in the figure. Instructions for choosing the non-redundant constraints are given in the figure.

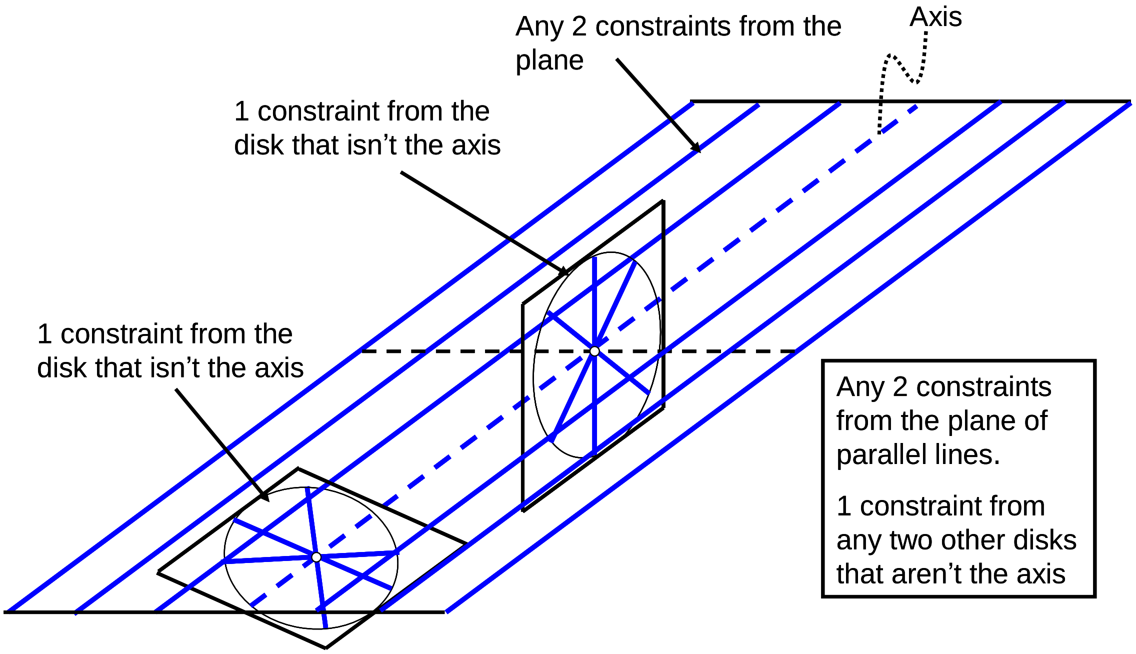

Any 2 constraints from the plane
Axis
1 constraint from the disk that isn't the axis
1 constraint from the disk that isn't the axis
Any 2 constraints from the plane of parallel lines.
1 constraint from any two other disks that aren't the axis

Figure 8.69: Second sub-constraint space of Case 4, Type 7

The third sub-constraint space is shown in **Figure 8.70**. It consists of a planar constraint set of parallel lines and three constraint set disks. These disks may be any three disks from within the constraint space of the system and don't have to be the three disks arbitrarily represented in the figure. Instructions for choosing the four non-redundant constraints from this sub-constraint space are shown in the figure.

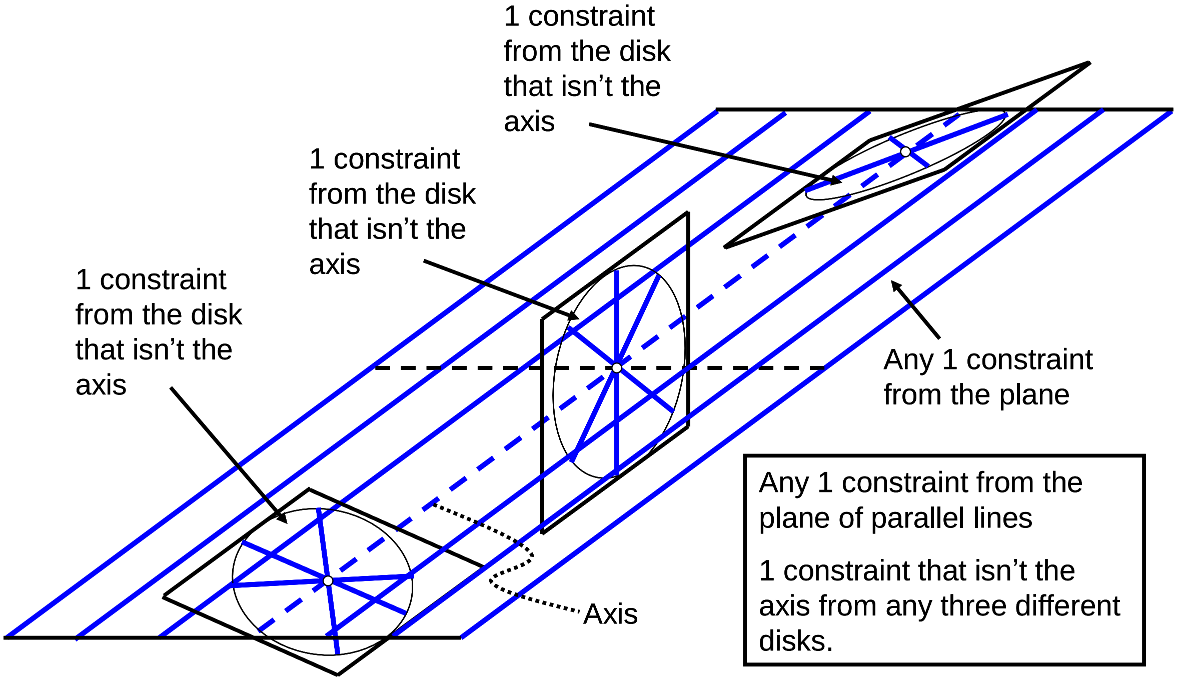


Figure 8.70: Third sub-constraint space of Case 4, Type 7

The fourth sub-constraint space is shown in **Figure 8.71**. It consists of three constraint set disks. These disks may be any three disks from within the constraint space of the system and don't have to be the three disks arbitrarily represented in the figure. Instructions for choosing the non-redundant constraints are given in the figure.

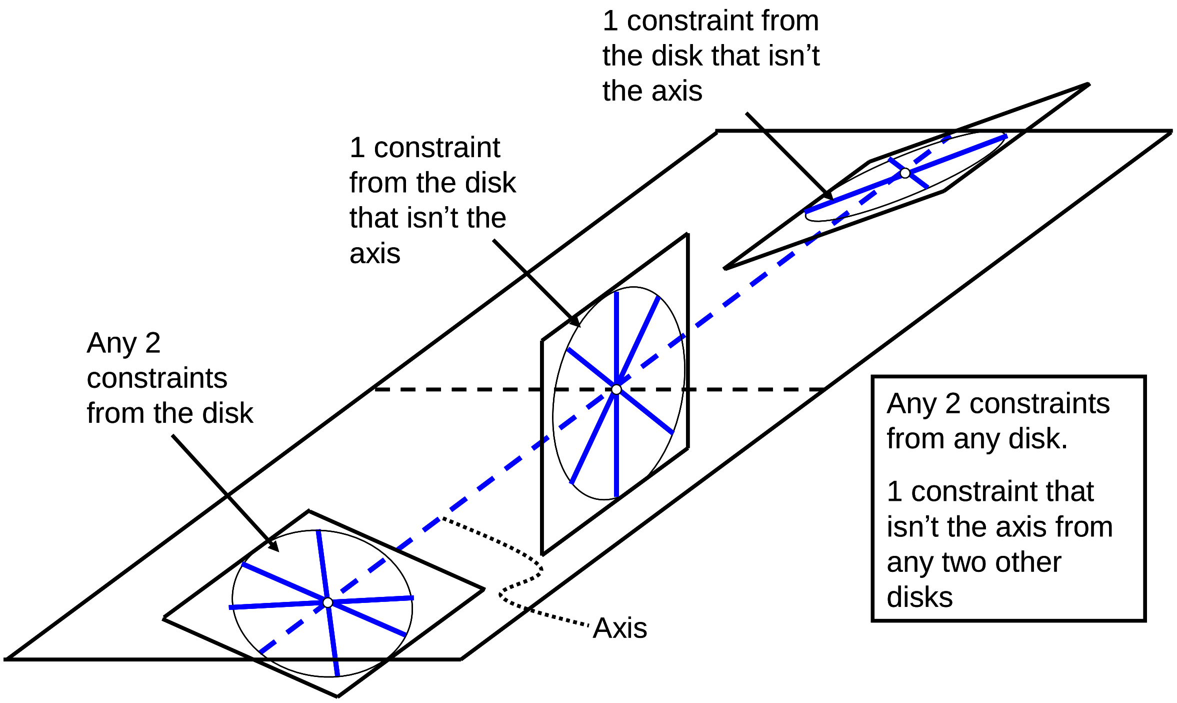


Figure 8.71: Fourth sub-constraint space of Case 4, Type 7

The fifth sub-constraint space is shown in **Figure 8.72**. It consists of four constraint set disks. These disks may be any four disks from within the constraint space of the system and not just the four disks arbitrarily represented in the figure. Instructions are provided in the figure for guiding the designer in selecting constraints that are non-redundant. Note also that if each of the four constraints is selected from the same location within its respective disk, one of these constraints will be redundant since all four constraints will belong to a ribbon space and will lie on the surface of a hyperbolic paraboloid from Case 3. If, for instance, every constraint line selected is at the location within its disk such that it is perpendicular to the axis line of the constraint space, the four constraints selected will belong to an orthogonal ribbon space.

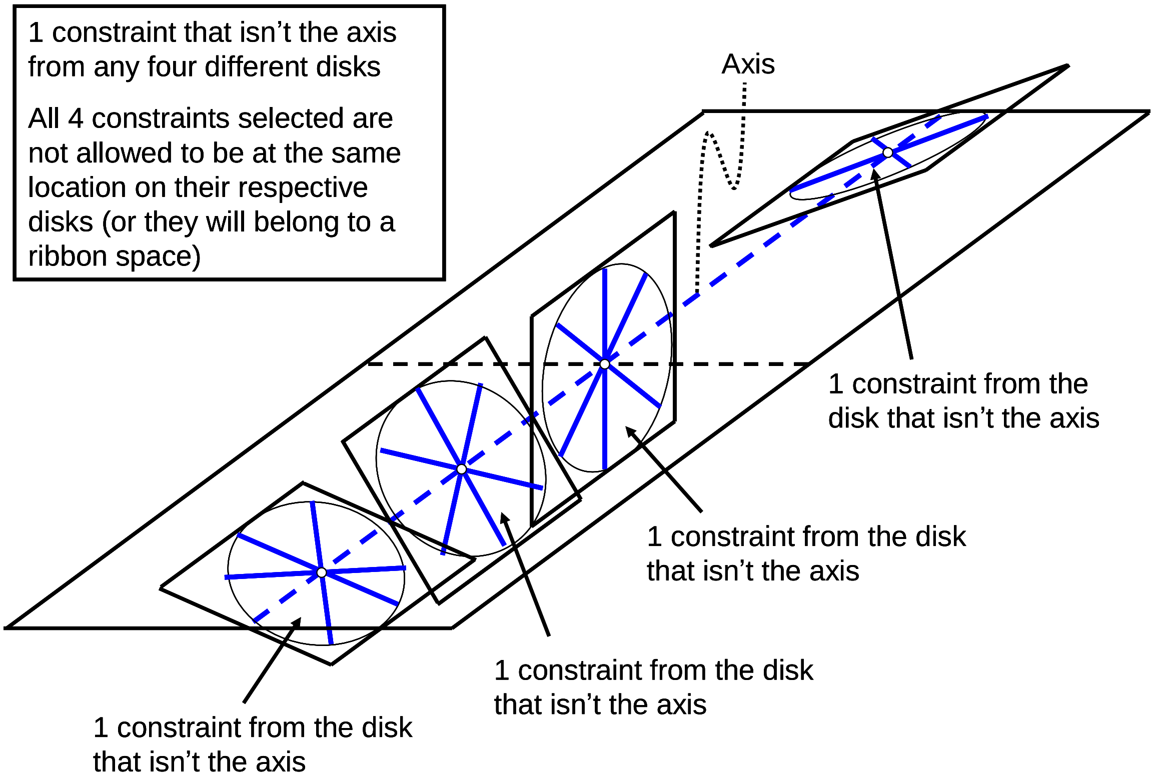

1 constraint that isn't the axis from any four different disks
All 4 constraints selected are not allowed to be at the same location on their respective disks (or they will belong to a ribbon space)
Axis
1 constraint from the disk that isn't the axis
1 constraint from the disk that isn't the axis
1 constraint from the disk that isn't the axis
1 constraint from the disk that isn't the axis

Figure 8.72: Fifth sub-constraint space of Case 4, Type 7

8.3.8 Case 4, Type 8:

This section describes the unique constraint space with its sub-constraint spaces for a system with the freedom space described in **Section 8.1.2** and shown in **Figure 8.8**. For completeness, it is shown again here in **Figure 8.73**. This freedom space is a disk of screws that all have the same finite, non-zero pitch value.

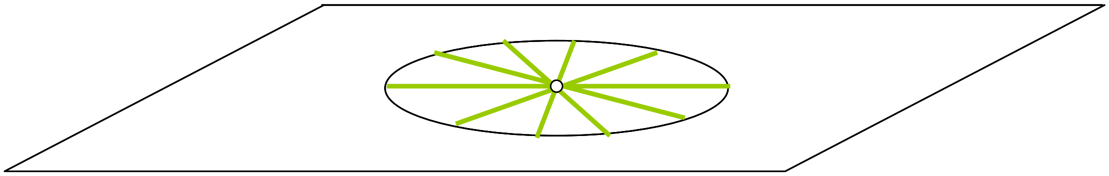


Figure 8.73: Freedom space of Case 4, Type 8

In order to determine this freedom space's complementary constraint space, note that there are no pure rotational freedom lines to apply Blanding's Rule of Complementary Patterns to. One must, therefore, rely on a completely different approach for finding this system's constraint space. In a way, however, this system's constraint space has already been found. Recall from **Section 7.2.1** in **Chapter 7** that **Equation (7.1)** relates the pitch of a twist that complements a disk of constraint lines to parameters that define its position with respect to that disk. From symmetry, the same argument presented in that section is used for determining the reverse scenario of a constraint line that complements a disk of twists that all have the same finite, non-zero pitch value, *p*. This reverse scenario is shown in **Figure 8.74** (Note how this figure compares with **Figure 7.13** from **Chapter 7**). From the conclusions in **Chapter 7**, one may deduce that only a constraint line that intersects the plane of the disk of screws at an angle of, α, that is also perpendicular to one of these screw lines and is a distance *L* away from the center point of this disk will complement the system's freedom space according to **Equation (7.1)** which is given again here as

$$p = L\tan\alpha\,. \tag{8.22}$$

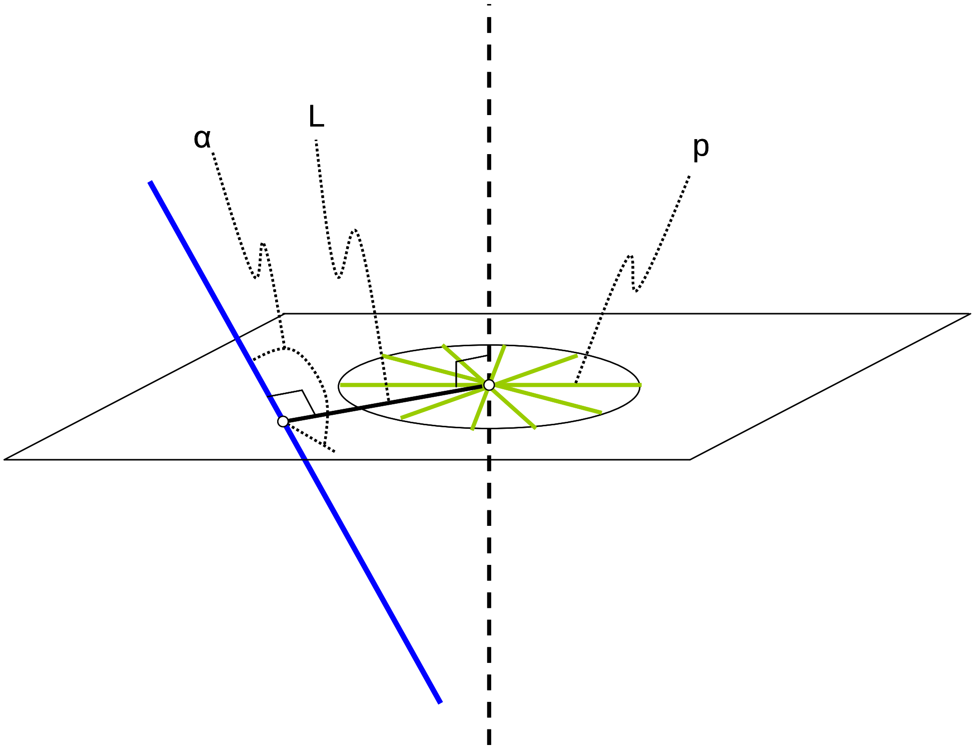


Figure 8.74: The parameters for a constraint line that complements a disk of screws that have the same pitch value, *p*.

Since the disk of screws is symmetric about the dashed black line shown in **Figure 8.74**, note that every constraint line on the surface of a circular hyperboloid with a central circular cross-sectional radius of *L* will exist. In fact, since the pitch, *p*, of the disk of screws is a constant value for every freedom space of this case and type, **Equation (8.22)** suggests that at every distance of *L* away from the center point of this disk, there is a unique circular hyperboloid of constraint lines with fixed and equal angles, α, that will exist within the system's constraint space. The constraint space of this system is, therefore, an infinite number of nested hyperboloids. The inner most hyperboloid has a radius, L, equal to zero and is in fact a single constraint line that is perpendicular to the disk of screws and intersects them at their center point. The hyperboloid's angle, α, changes as the radius, *L*, gets larger. As this radius approaches infinity, the circular hyperboloid gradually collapses onto a plane of constraint lines. This concept is shown in **Figure 8.75**.

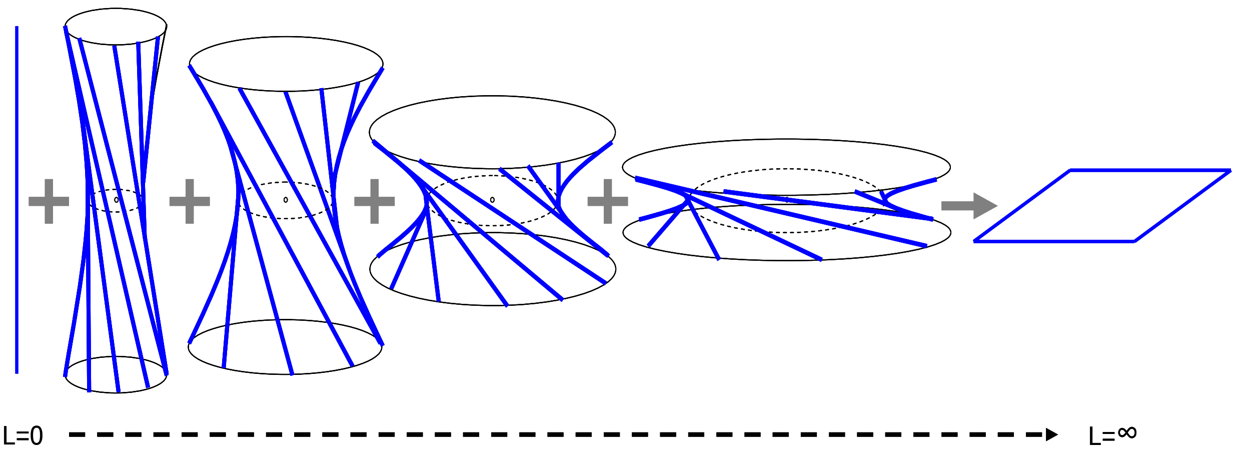


Figure 8.75: Nested circular hyperboloids within the constraint space of Case 4, Type 8 for different values of *L* where *L* is the radius of the hyperboloid's smallest circular cross-section.

Note, however, that the plane of constraint lines on the right side of the figure is not a part of this constraint space since *L* never actually reaches infinity. The complete constraint space of this case and type may, therefore, be represented as shown in **Figure 8.76**.

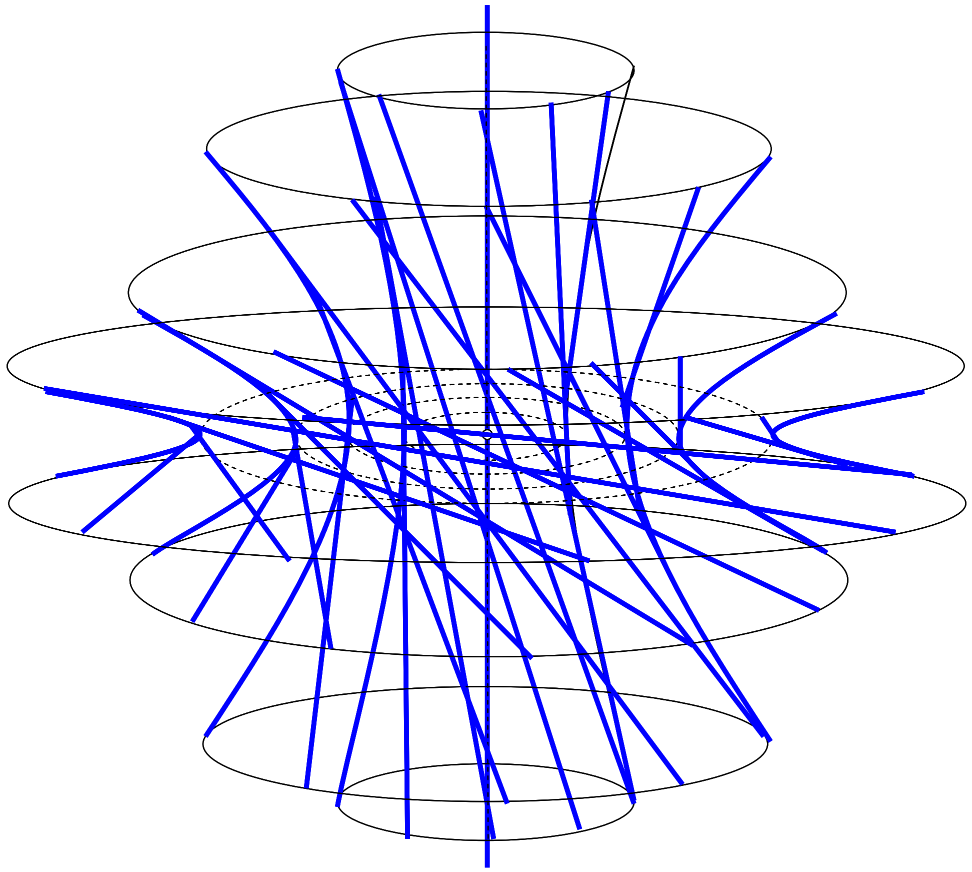


Figure 8.76: Constraint space of Case 4, Type 8

Another way to look at this constraint space is to consider an infinite number of orthogonal ribbons that rotate as they translate toward the central, vertical constraint line. The axes of these infinite ribbons are coincident with the screw lines within the disk of the freedom space. Part of one of these ribbons is shown in **Figure 8.77**. The constraint space of this case and type, therefore, consists of both an infinite number of circular hyperboloids and an infinite number of hyperboloic paraboloids (where *a*=*b* in **Equation (6.1)**).

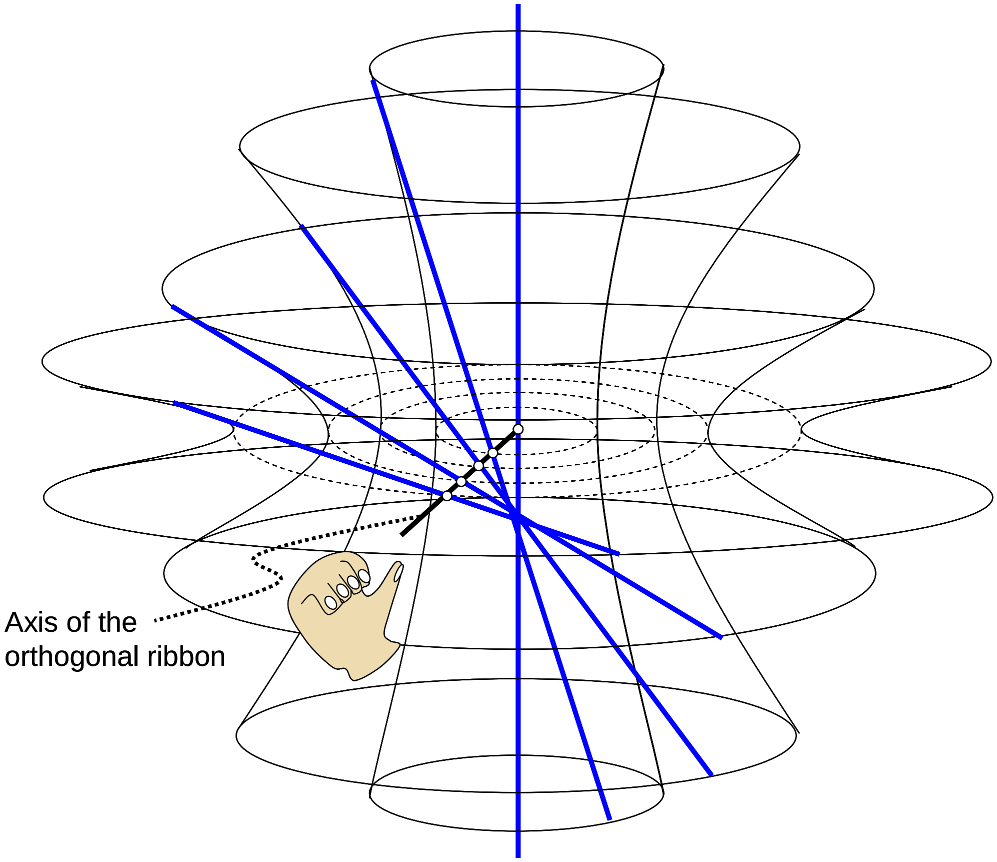


Figure 8.77: An example of a single orthogonal ribbon of constraint lines that exists within the constraint space of Case 4, Type 8.

Note that the orthogonal ribbon in the figure is a right-handed ribbon. The reason for this fact is that it exists within the constraint space of a system with a freedom space that consists of a disk of screws that have negative pitch values. Moreover, note that every hyperboloid shown in the constraint space in **Figure 8.76** is also a right-handed hyperboloid. In fact, because such nested hyperboloids are made up of right-handed ribbons, the author decided to name these hyperboloids right-handed hyperboloids. If the pitch values of the screws within the freedom space disk had been positive, the orthogonal ribbons and the circular hyperboloids within the constraint space would have both been left-handed.

An equation can now be derived that describes the complete constraint space of Case 4, Type 8. Recall from **Chapter 7** that the equation for a circular hyperboloid is **Equation (7.6)**. If **Equation (8.22)** is substituted into **Equation (7.6)**, one finds that for a given distance, *L*, from the central point of the disk of screws, a nested circular hyperboloid will exist that contains an infinite ruling of constraint lines on its surface that is described by

$$\frac{x^2+y^2}{L^2}-\frac{z^2}{p^2}=1, \tag{8.23}$$

where p is the pitch of the disk of screws from the freedom space. The equation of the entire constraint space is, therefore, **Equation (8.23)** for all real values of L where each value corresponds to a single circular hyperboloid.

Figure 8.78 shows how the freedom and constraint spaces of this system fit together. The central, vertical constraint line within the constraint space is perpendicular to the plane of the disk of screws from the freedom space and intersects it at its central point. The plane of the nested hyperboloids' smallest circular cross-sections is the same plane as the disk of screws.

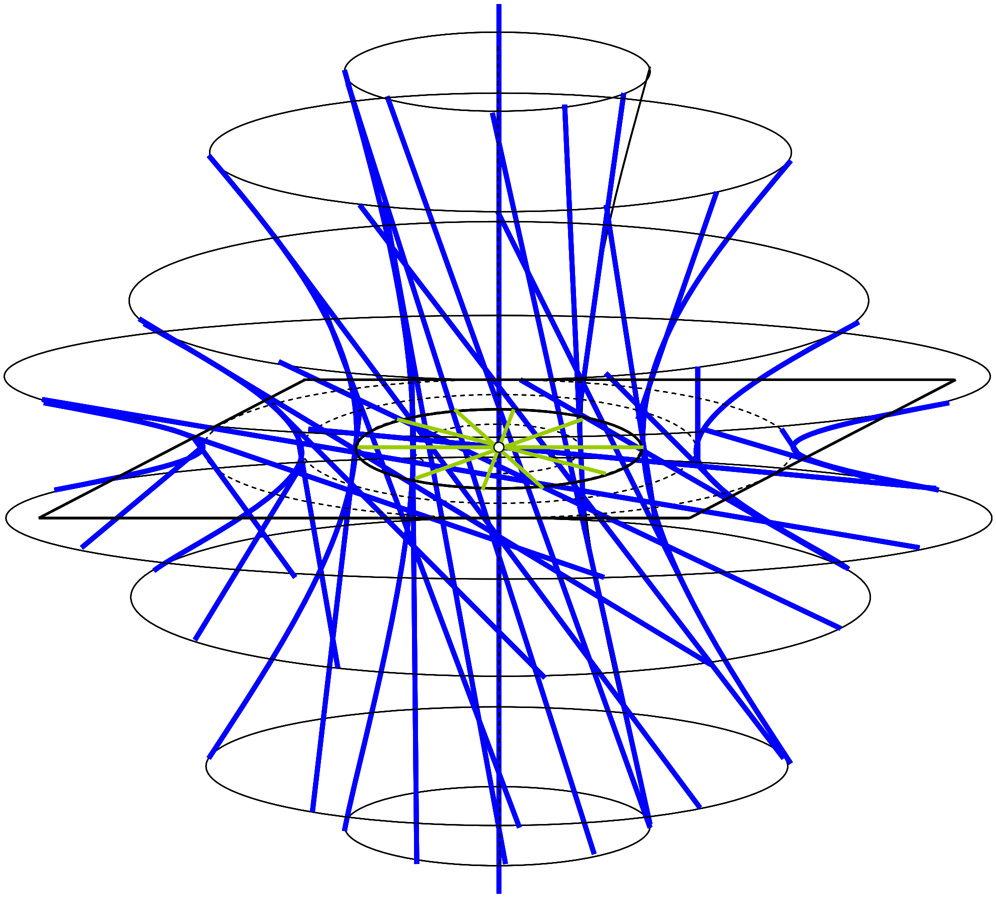


Figure 8.78: Freedom space (green) and constraint space (blue) of Case 4, Type 8 together.

Three sub-constraint spaces exist within this constraint space. All three sub-constraint spaces look identical to their constraint space. Each, however, has its own set of unique instructions for guiding the designer in selecting every possible set of non-redundant constraints from within the constraint space of the system. The first is shown in **Figure 8.79**.

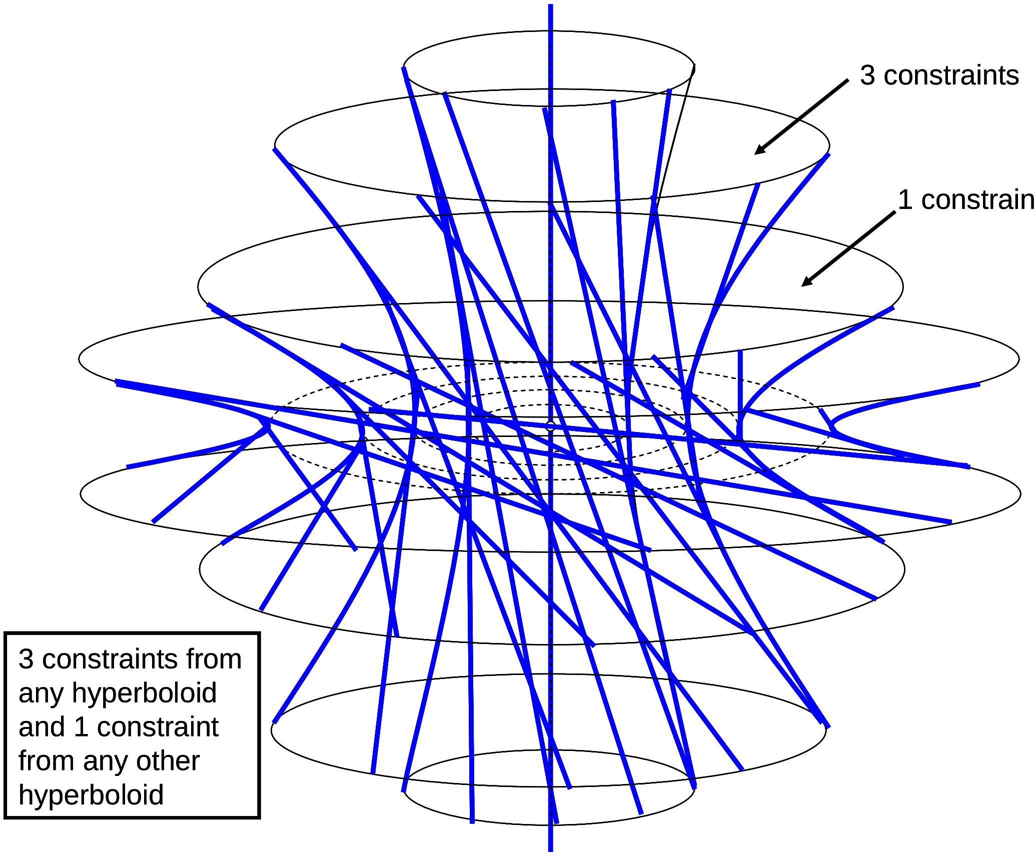

3 constraints
1 constraint
3 constraints from any hyperboloid and 1 constraint from any other hyperboloid

Figure 8.79: First sub-constraint space of Case 4, Type 8

The second sub-constraint space is shown in **Figure 8.80**.

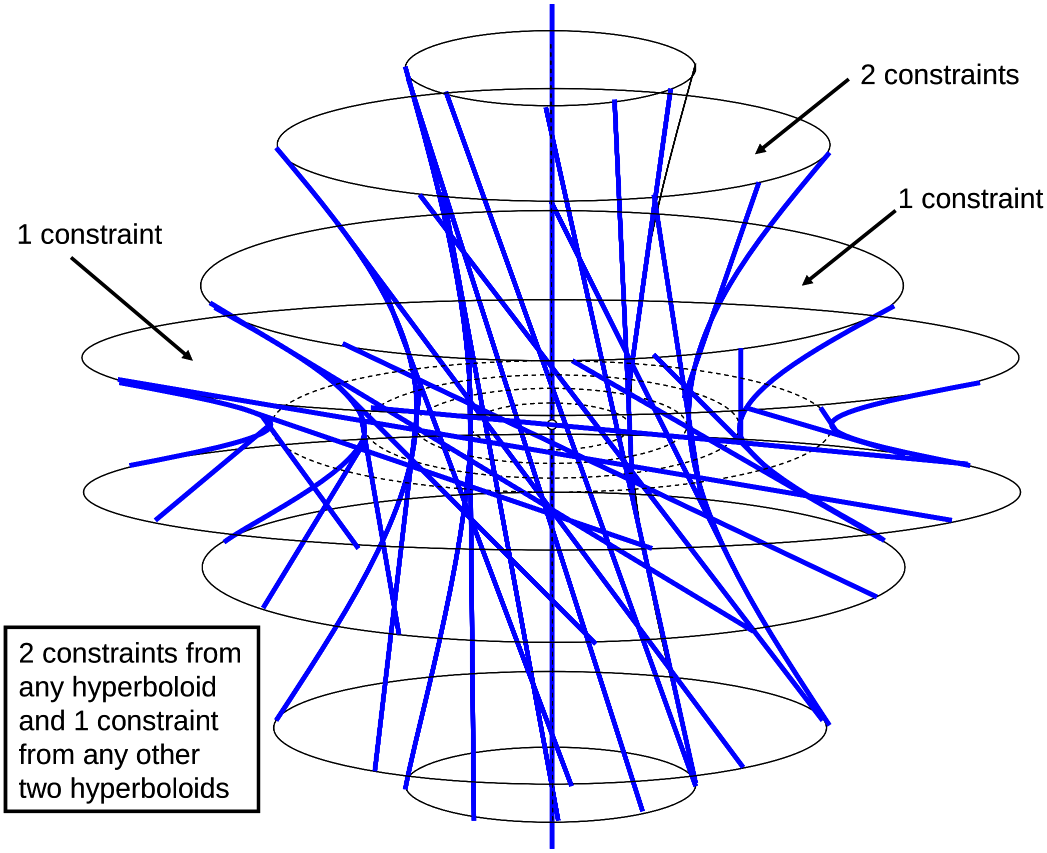

2 constraints
1 constraint
1 constraint
2 constraints from any hyperboloid and 1 constraint from any other two hyperboloids

Figure 8.80: Second sub-constraint space of Case 4, Type 8

The third sub-constraint space is shown in **Figure 8.81**.

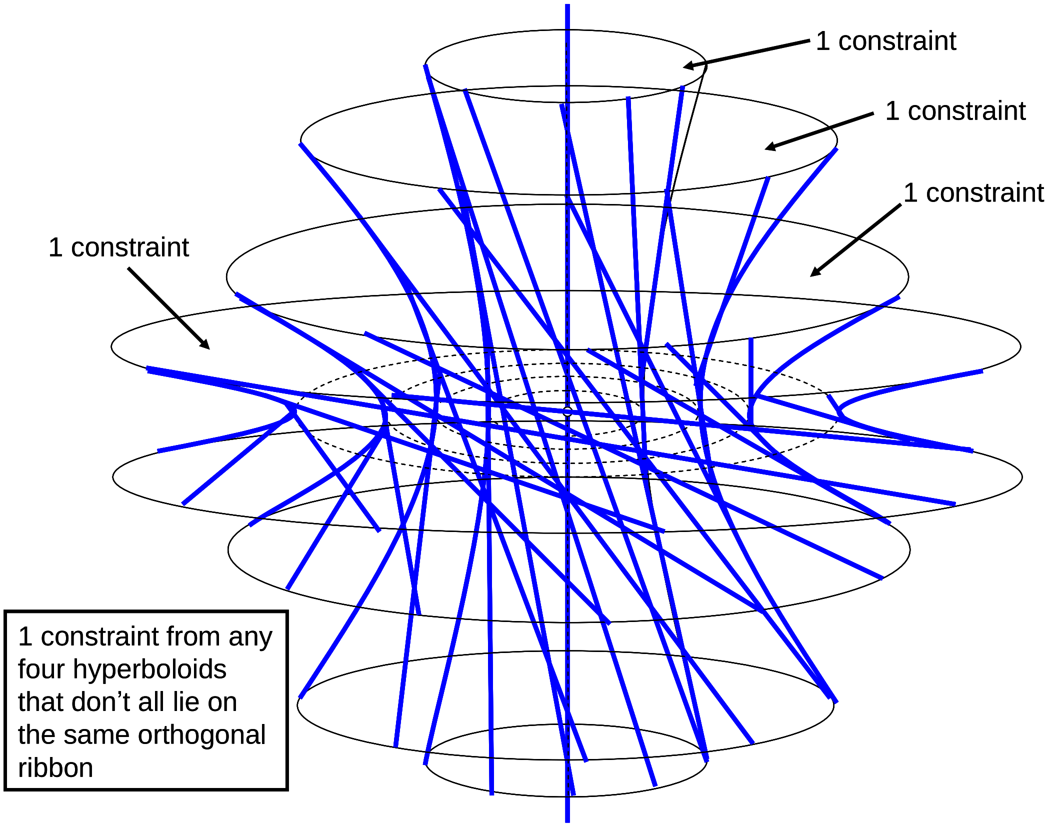

1 constraint
1 constraint
1 constraint
1 constraint
1 constraint from any four hyperboloids that don't all lie on the same orthogonal ribbon

Figure 8.81: Third sub-constraint space of Case 4, Type 8

8.3.9 Case 4, Type 9:

This section describes the unique constraint space with its sub-constraint spaces for a system with the freedom space described in **Section 8.1.2** and shown in **Figure 8.12**. For completeness, it is shown again here in **Figure 8.82**. This freedom space is a cylindroid of pure screws.

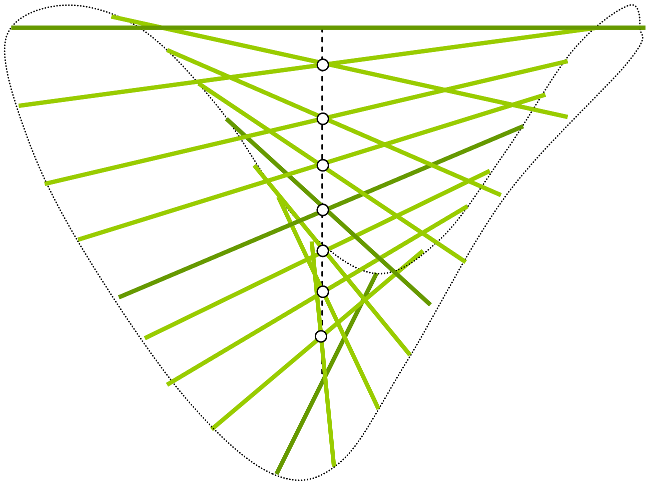


Figure 8.82: Freedom space of Case 4, Type 8

In order to determine this freedom space's complementary constraint space, note that there are no pure rotational freedom lines to apply Blanding's Rule of Complementary Patterns to. One must, therefore, rely on a different approach for finding this system's constraint space. If all the constraint lines are found that complement the cylindroid's two principal generators (or any other two independent twists within the cylindroid), the system's complete constraint space will have been found since every twist within the cylindroid is a linear combination of its two principal generators. Suppose the two screw principal generators are oriented along the x- and y-axes of a coordinate system. Recall that principal generators always intersect at orthogonal angles. Now recall from **Figure 3.9** in **Chapter 3** that a constraint line that intersects a screw line at an orthogonal angle is permitted by **Equation (3.12)**. One may, therefore, know that any line that intersects and is orthogonal to the principal generator along the y-axis that also simultaneously satisfies **Equation (3.13)** for the principal generator along the x-axis will be a constraint line within the system's constraint space. These constraint lines make up an orthogonal ribbon with an axis along the y-axis as shown in **Figure 8.83**. Note that the ribbon of constraint lines is a right-handed orthogonal ribbon. This is because the principal generator screw along the x-axis has a negative pitch value for this figure.

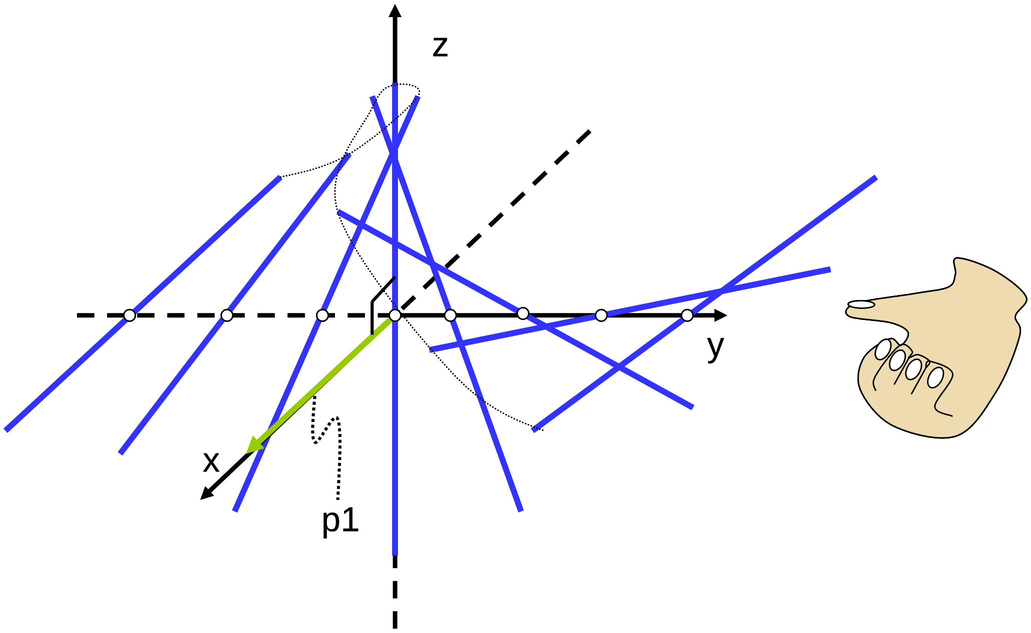


Figure 8.83: An orthogonal ribbon within the constraint space of Case 4, Type 9 that complements a cylindroid of pure screws with principal generators aligned along the x- and y-axes.

Also any line that intersects and is orthogonal to the principal generator along the x-axis that also simultaneously satisfies **Equation (3.13)** for the principal generator along the y-axis will be a constraint line within the system's constraint space. These constraint lines also make up an orthogonal ribbon with an axis along the x-axis as shown in **Figure 8.84**. Note that the ribbon of constraint lines is again a right-handed orthogonal ribbon. This statement is true because the principal generator screw along the y-axis also has a negative pitch value for this figure.

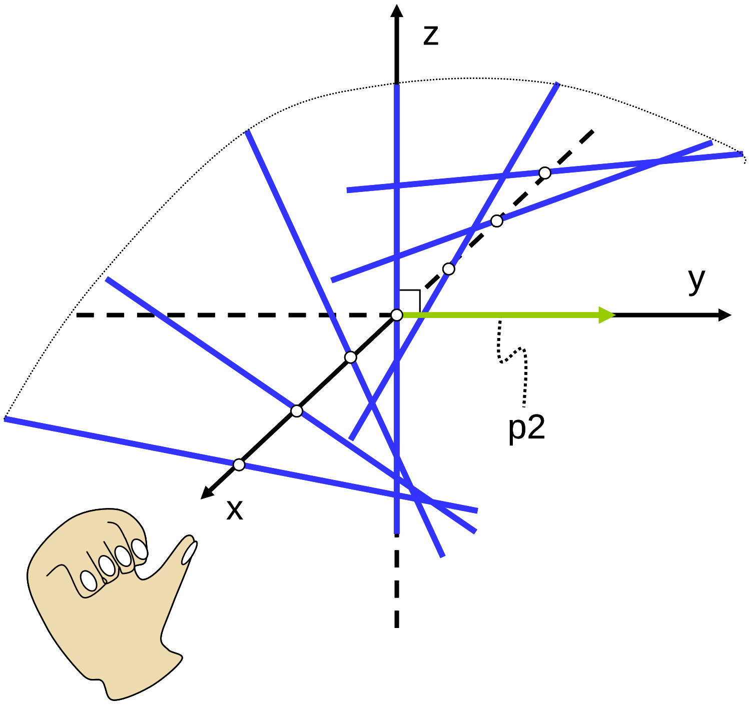


Figure 8.84: Another orthogonal ribbon within the constraint space of Case 4, Type 9 that also complements a cylindroid of pure screws with principal generators aligned along the x- and y-axes.

Recall also that the screw principal generators within the cylindroid of this system's freedom space must have pitch values that are not equal to each other but are of the same sign. They must also have non-zero and finite values. With these facts in mind, one may deduce that the two orthogonal ribbons shown above that exist within the constraint space of this system are either both right-handed or both left-handed orthogonal ribbons. Furthermore, one may deduce that the pitch of the ribbons, or the rate that the lines translate as they rotate along their axes, must also be different.

If the pitch values of both principal generators were equal, the cylindroid of the freedom space would collapse into a disk of screws and that these two orthogonal ribbons would translate as they rotate with the same rate as do the infinite orthogonal ribbons within the constraint space of Case 4, Type 8. In fact, if the pitches of these two principal generators were equivalent, this case and type would become Case 4, Type 8. It can, therefore, be intelligently hypothesized that the complete constraint space for this case and type is a series of nested elliptical hyperboloids that contain an infinite number of orthogonal ribbons that translate as they rotate at different rates toward a central constraint line similar to the constraint space of Case 4, Type 8. The hyperboloids would, in fact, have to be elliptical if these ribbons had different pitches. The two ribbons drawn in **Figure 8.83** and **Figure 8.84** are shown again within these nested elliptical hyperboloids in **Figure 8.85** and **Figure 8.86** to help the reader visualize the actual geometry of the constraint lines. Note also that the axes of these ribbons and, therefore, the principal generators within the system's freedom space are coincident with the major and minor axes of the smallest elliptical cross-sections within the nested elliptical hyperboloids.

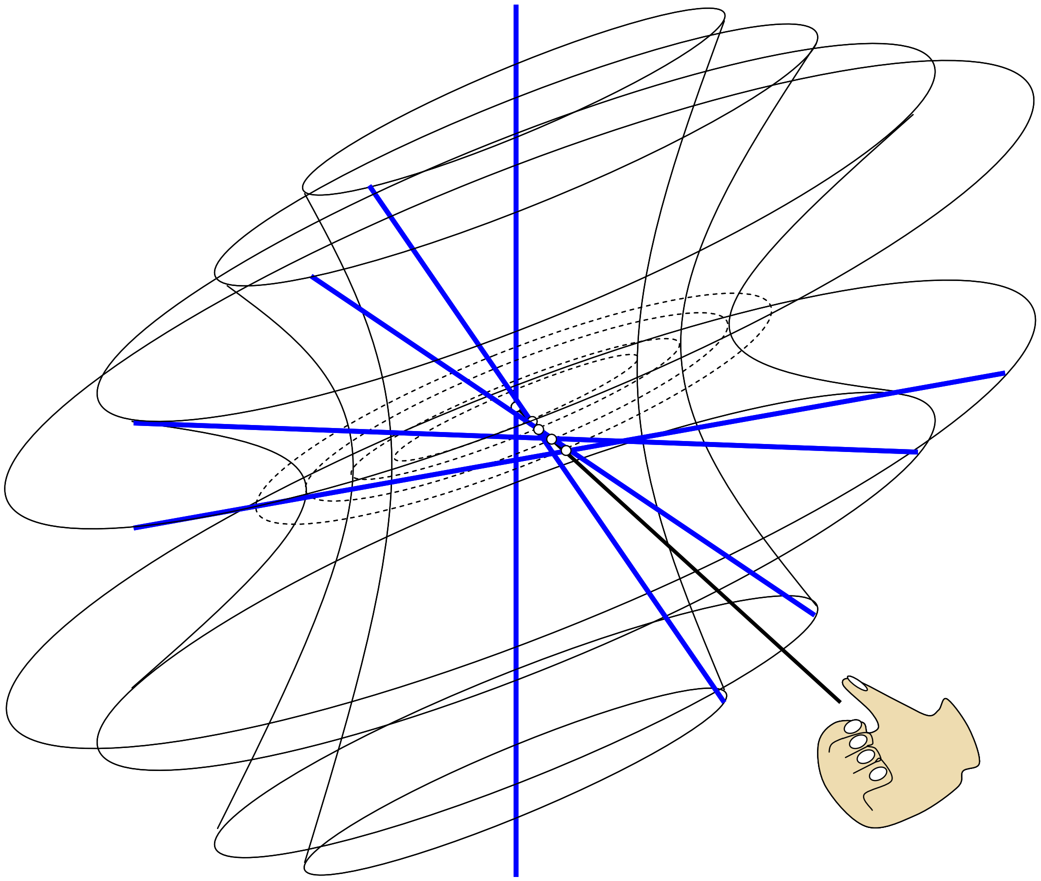


Figure 8.85: The same orthogonal ribbon from Figure 8.83 shown within nested elliptical hyperboloids. The axis of this ribbon (thick black) is the minor axis of the smallest elliptical cross-sections of the hyperboloids (dashed black) and was the y-axis from Figure 8.83.

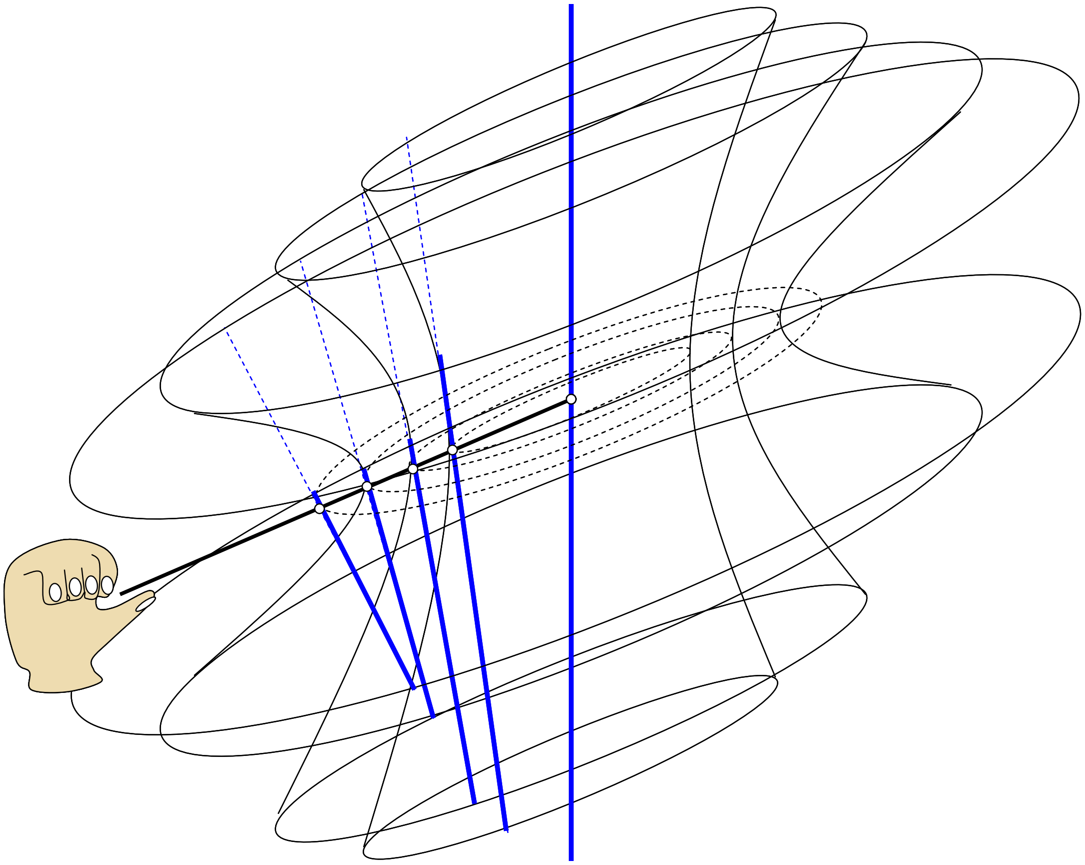


Figure 8.86: The same orthogonal ribbon from Figure 8.84 shown within nested elliptical hyperboloids. The axis of this ribbon (thick black) is the major axis of the smallest elliptical cross-sections of the hyperboloids (dashed black) and was the x-axis from Figure 8.84.

Appendix N mathematically proves that the hypothesis is correct. The complete constraint space of Case 4, Type 9 is indeed an infinite number of nested elliptical hyperboloids as shown in **Figure 8.87**.

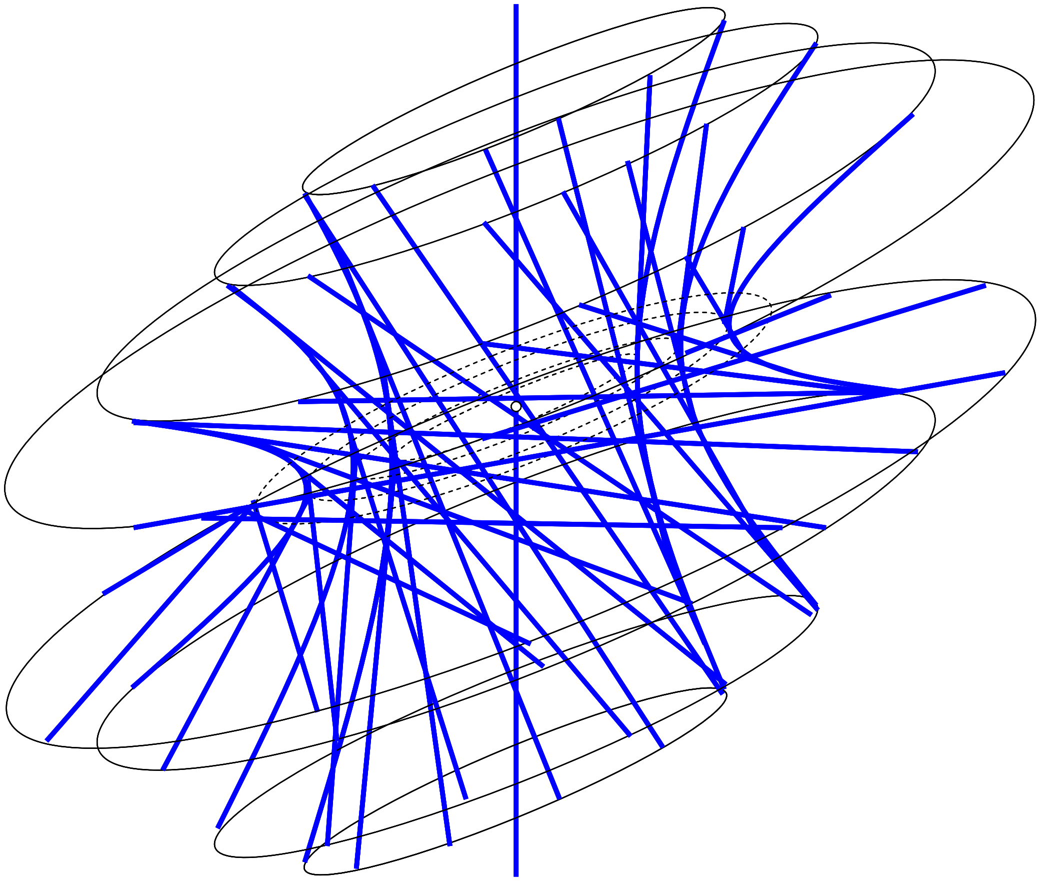


Figure 8.87: Constraint space of Case 4, Type 9

The minor axis of the smallest elliptical cross-section of each nested hyperboloid will arbitrarily be chosen to mathematically describe this constraint space. Recall also that this axis is the y-axis using the previous coordinate system. The distance along this axis from the center point of the elliptical hyperboloids is *b*. When *b* equals zero, the corresponding elliptical hyperboloid is a single vertical constraint line. As *b* increase, the constraint lines within the corresponding elliptical hyperboloids begin to approach a flat plane, which is perpendicular to the central vertical constraint line. This concept is shown in **Figure 8.88**. This plane of constraint lines is not a part of the constraint space since *b* never actually reaches infinity.

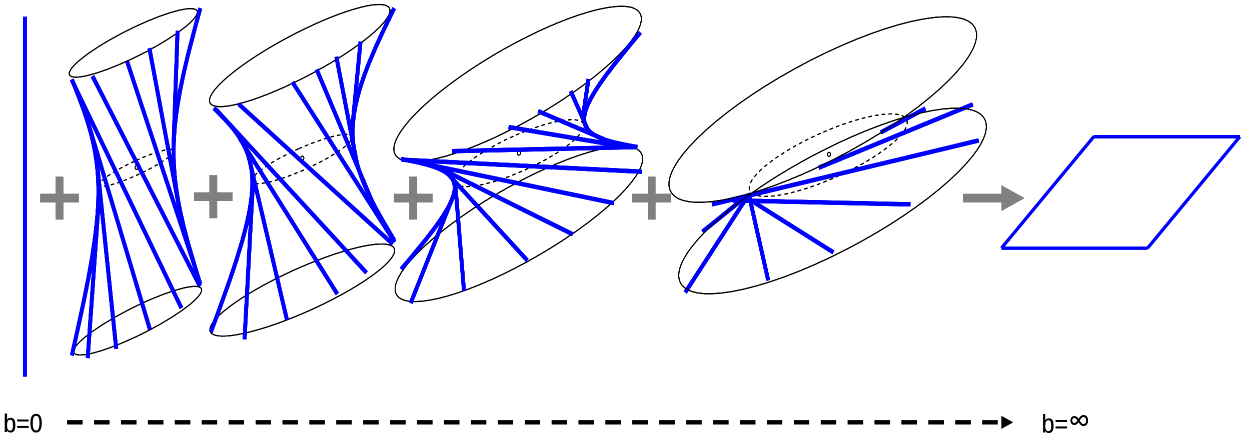


Figure 8.88: Nested elliptical hyperboloids within the constraint space of Case 4, Type 9 for different values of *b* where *b* is the minor axis of the hyperboloid's smallest elliptical cross-section.

If **Equation (N.1)** and **Equation (N.4)** are substituted from **Appendix N** into **Equation (7.7)** from **Chapter 7**, one finds that the equation for a single elliptical hyperboloid within the constraint space of Case 4, Type 9 is

$$\left(\frac{p_1}{p_2}\right)x^2+y^2=b^2\left(\frac{z^2}{p_1 p_2}+1\right), \tag{8.24}$$

where p_1 and p_2 are the pitch values of the principal generators from the cylindroid of the system's freedom space. The equation for the system's complete constraint space of nested elliptical hyperboloids is, therefore, **Equation (8.24)** for all real values of *b* where each value corresponds to a single elliptical hyperboloid with a minor axis of *b*.

Note that if p_1 is set equal to p_2 and *a* and *b* are set equal to *L*, the system would become Case 4, Type 8. This expectation is confirmed since **Equation (8.24)** does become **Equation (8.23)** when these values are equated.

Figure 8.89 shows how the freedom and constraint spaces of this system fit together. The principal generators from the cylindroid of pure screws are coincident with the major and minor axes of the smallest elliptical cross-sections within the nested elliptical hyperboloids of the constraint space.

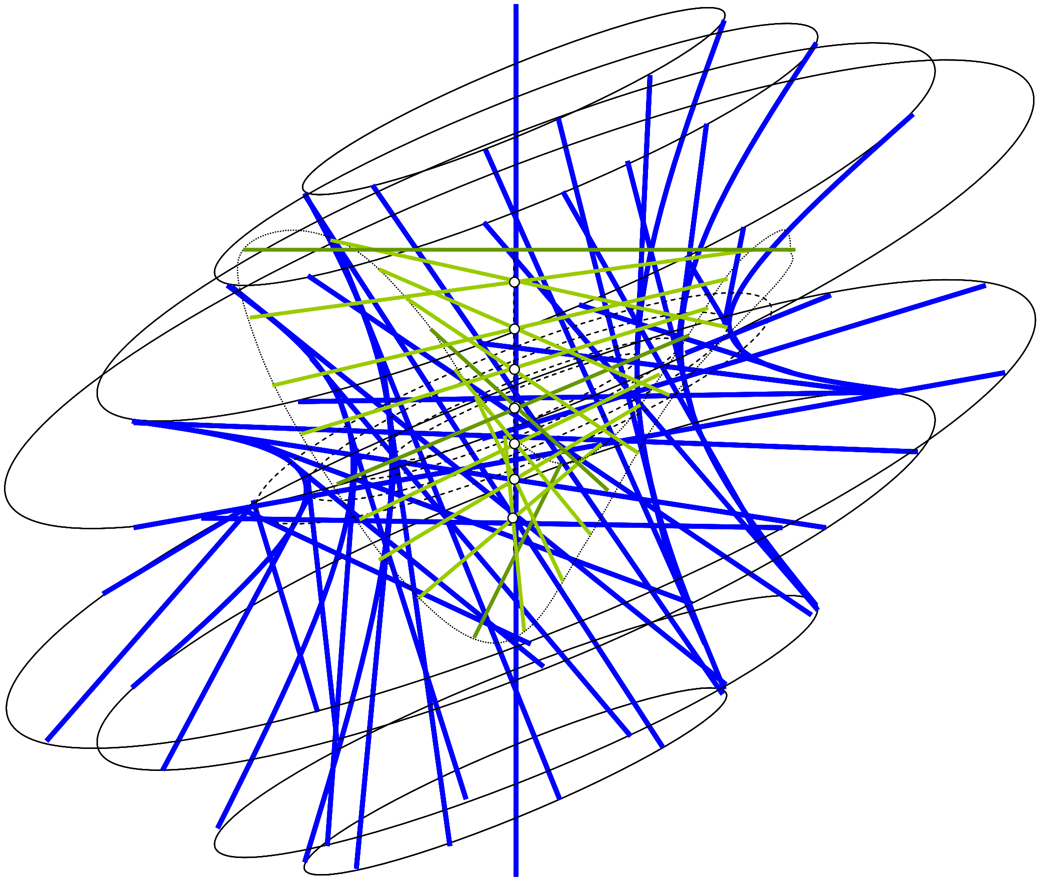


Figure 8.89: Freedom space (green) and constraint space (blue) of Case 4, Type 9 together.

Four sub-constraint spaces exist within this constraint space. All four sub-constraint spaces look identical to the constraint space. Each, however, has its own set of unique instructions for guiding the designer in selecting every possible set of non-redundant constraints from within the constraint space of the system. The first is shown in **Figure 8.90**.

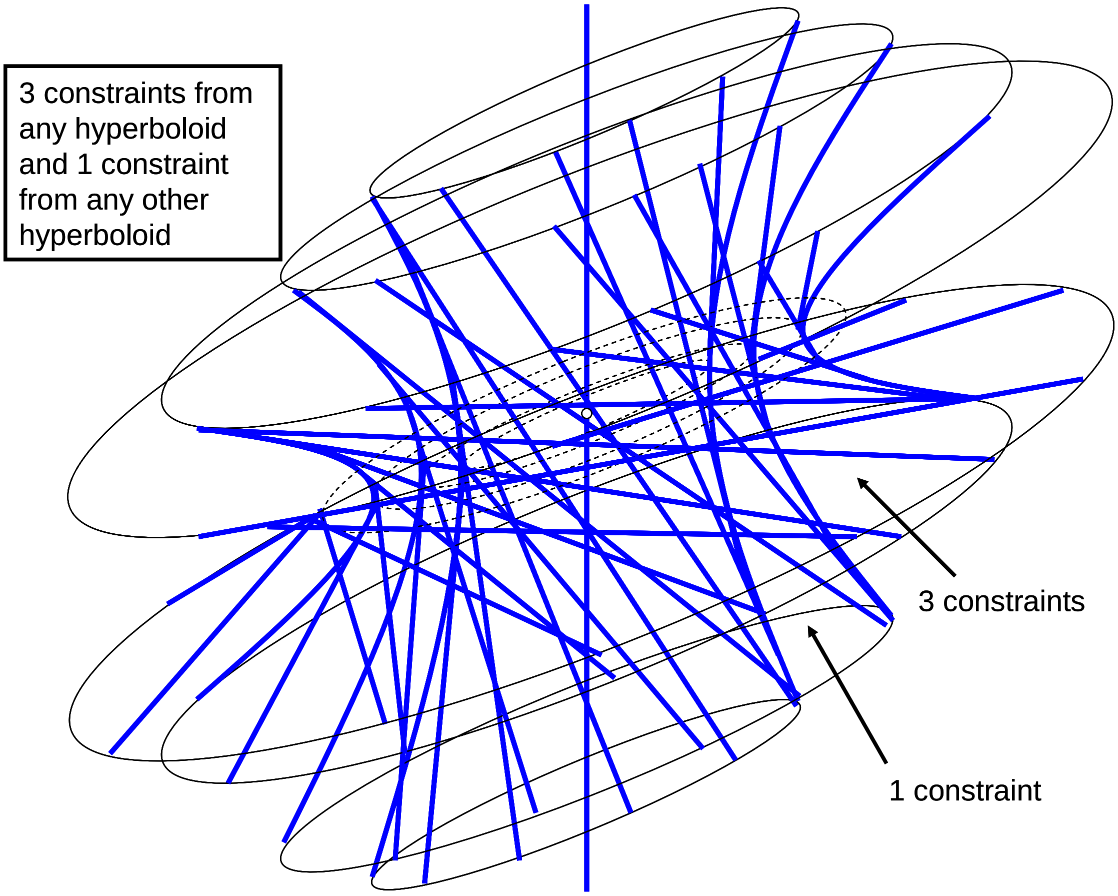

3 constraints from any hyperboloid and 1 constraint from any other hyperboloid
3 constraints
1 constraint

Figure 8.90: First sub-constraint space of Case 4, Type 9

The second sub-constraint space is shown in **Figure 8.91**.

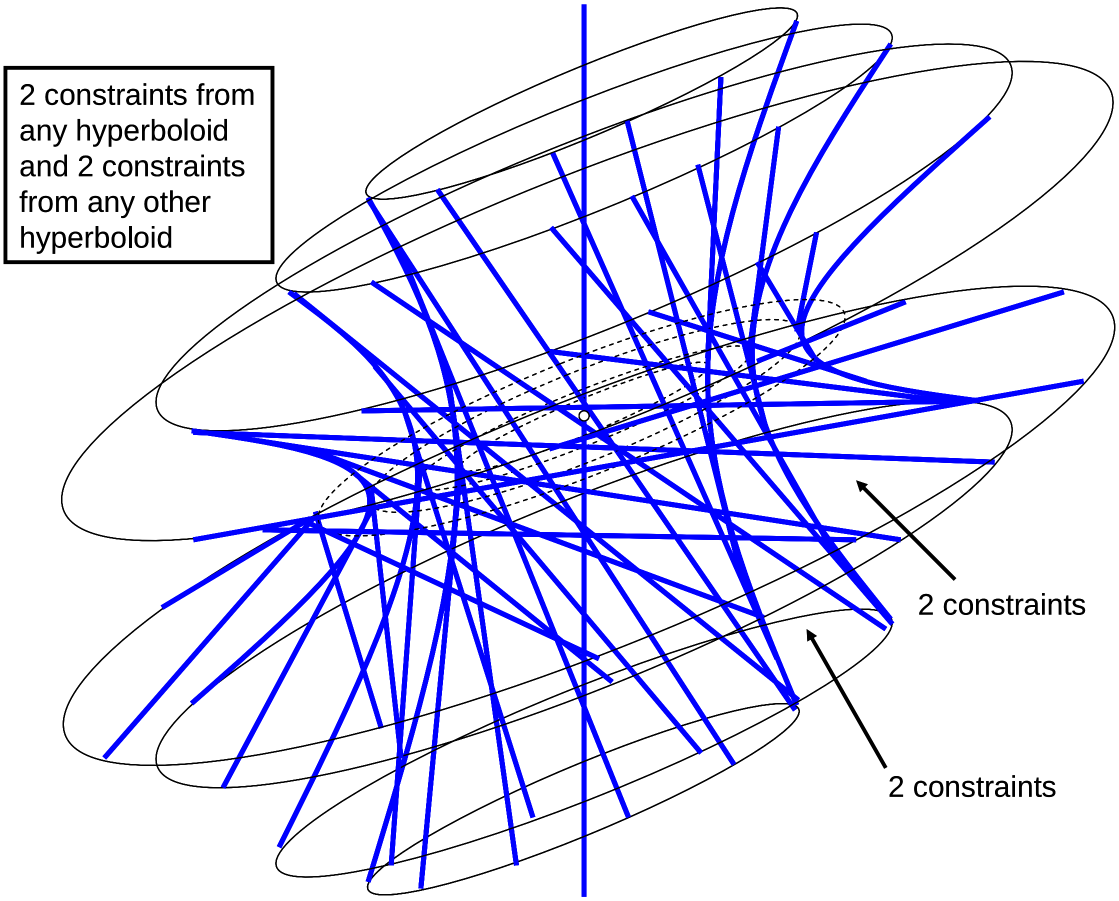

2 constraints from any hyperboloid and 2 constraints from any other hyperboloid
2 constraints
2 constraints

Figure 8.91: Second sub-constraint space of Case 4, Type 9

The third sub-constraint space is shown in **Figure 8.92**.

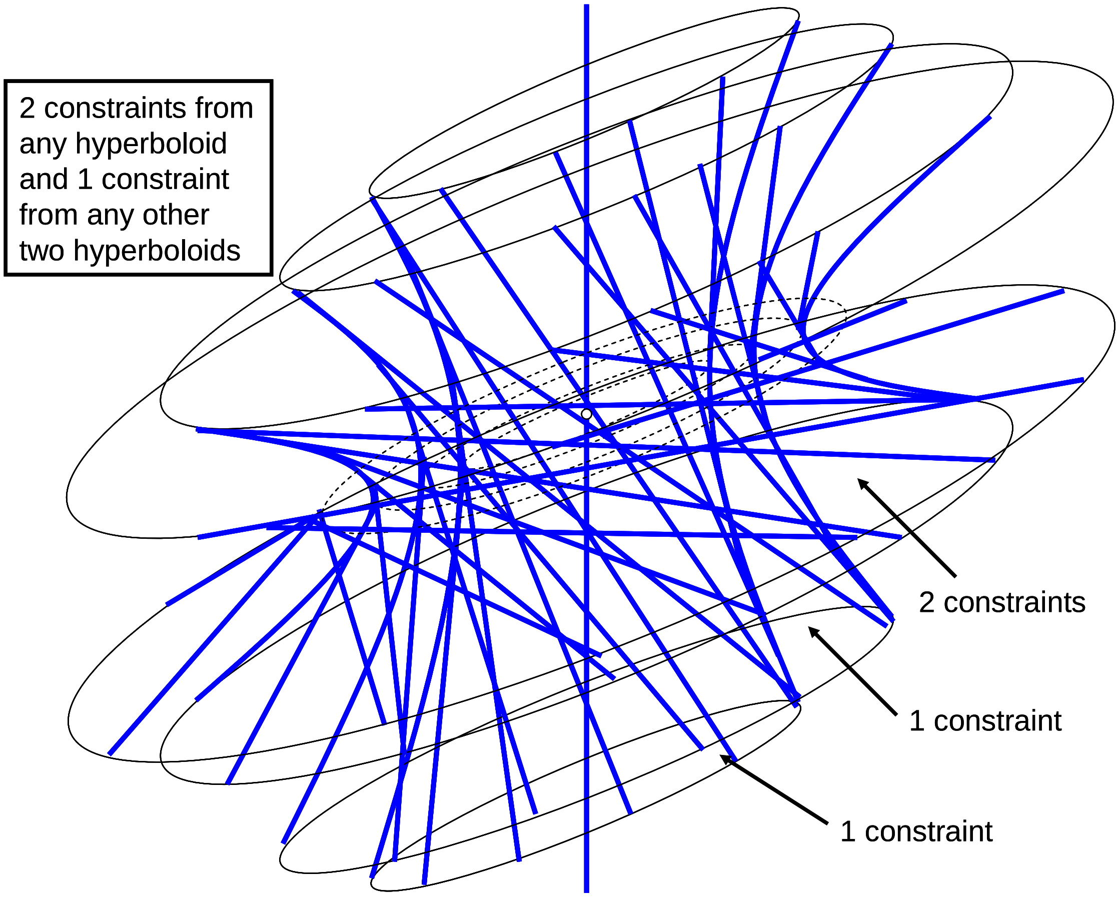

2 constraints from any hyperboloid and 1 constraint from any other two hyperboloids
2 constraints
1 constraint
1 constraint

Figure 8.92: Third sub-constraint space of Case 4, Type 9

The fourth sub-constraint space is shown in **Figure 8.93**.

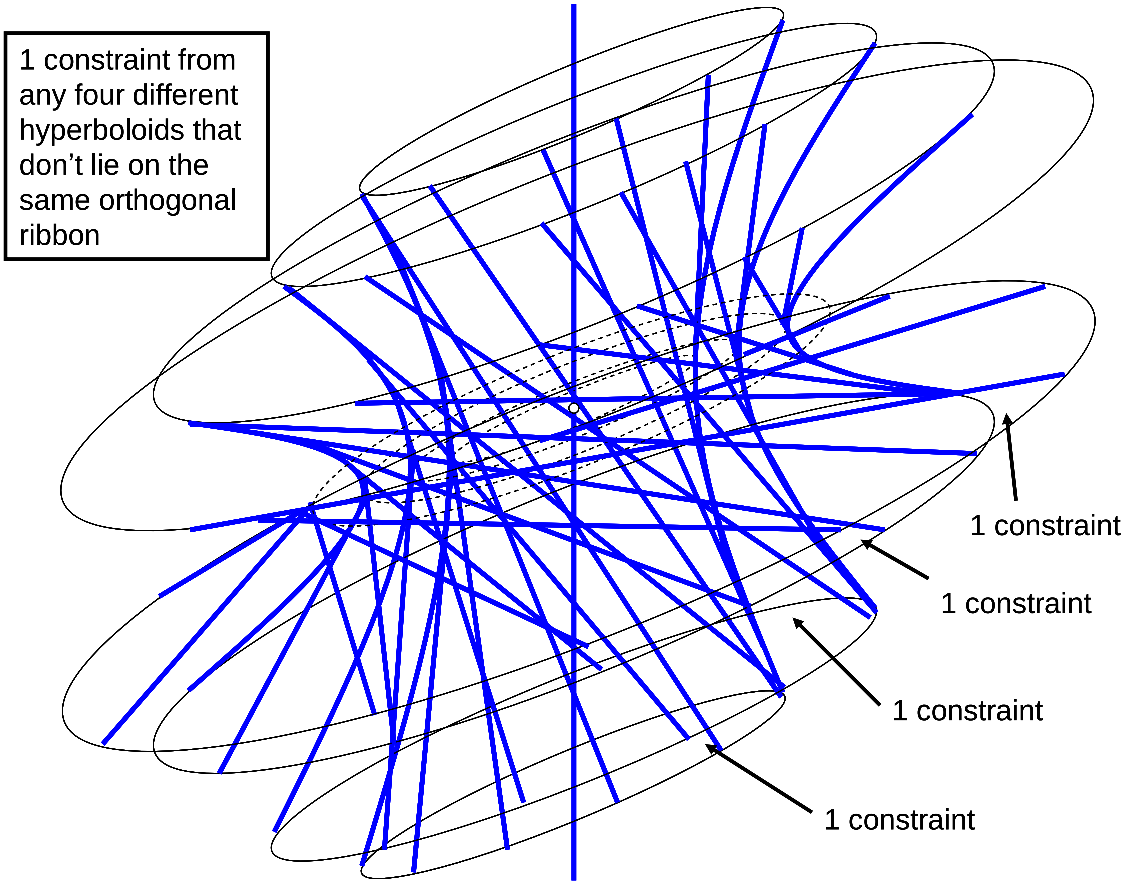

1 constraint from any four different hyperboloids that don't lie on the same orthogonal ribbon
1 constraint
1 constraint
1 constraint
1 constraint

Figure 8.93: Fourth sub-constraint space of Case 4, Type 9

8.4 Case 5:

This section describes the fifth case of 6. The fifth case consists of all systems that contain five non-redundant constraints. Using **Equation (2.1)** from **Chapter 2**, therefore, one deduces that the freedom spaces within Case 5 contain only a single independent twist. It was previously concluded that since there are only three fundamentally different twists (pure rotations, screws, and pure translations), there must only be three types within this case. In this section, these three freedom spaces will be presented briefly and their unique constraint spaces will then be determined. The concept of sub-constraint space and how it applies to Case 5 will be discussed at the end of this section as well.

8.4.1 Case 5, Type 1:

This section describes the unique constraint space of a system with a freedom space that consists of a single pure rotational freedom line. This freedom space is shown in **Figure 8.94**.

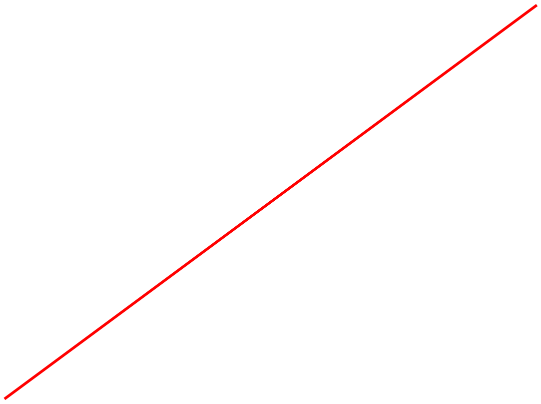


Figure 8.94: Freedom space of Case 5, Type 1

Since this freedom space consists entirely of pure rotational freedom lines, Blanding's Rule of Complementary Patterns may be applied to determine its constraint space. Every line that intersects the freedom line at least once is an allowable constraint line within the constraint space. This constraint space is shown in **Figure 8.95**. It consists of an infinite number of spherical constraint sets. Each of these spherical sets contains every constraint line that intersects a single point along the freedom line. Every point along the freedom line is the central point of a single spherical constraint set. The constraint space also contains a box constraint set that consists of every constraint line in three-space that is parallel to the freedom line.

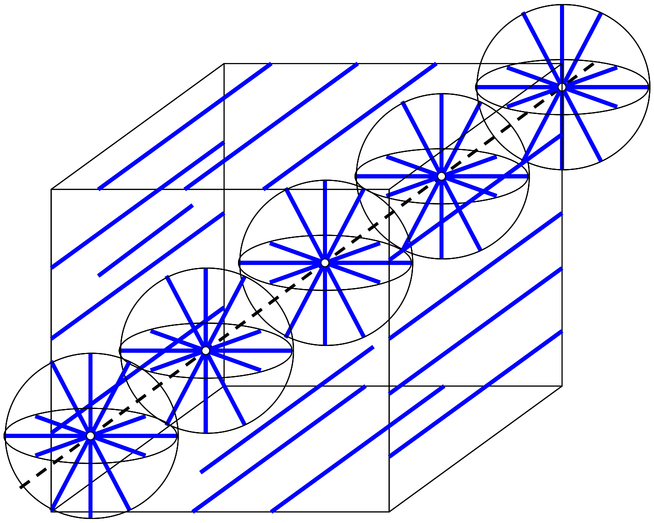


Figure 8.95: Constraint space of Case 5, Type 1

Figure 8.96 shows how the freedom and constraint spaces fit together. The freedom line intersects the center point of every spherical constraint set and is parallel to every constraint line in the box constraint set.

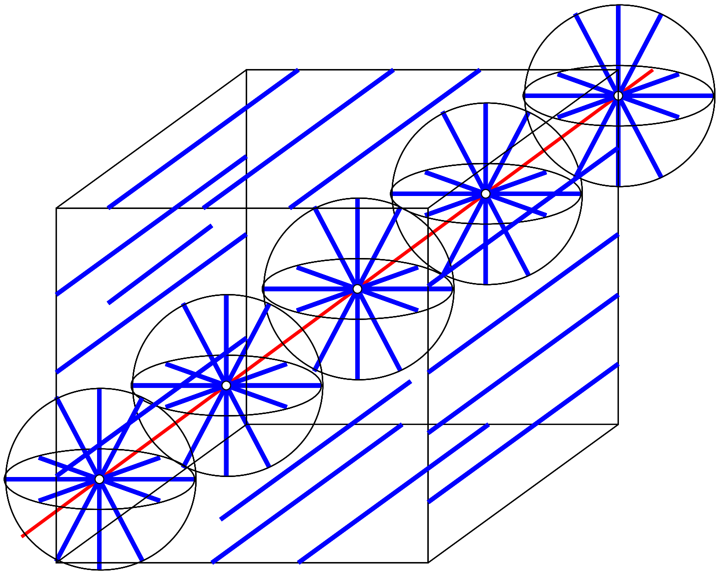


Figure 8.96: Freedom space (red) and constraint space (blue) of Case 5, Type 1 together

8.4.2 Case 5, Type 2:

This section describes the unique constraint space of a system with a freedom space that consists of a single screw line. This freedom space is shown in **Figure 8.97**. The screw has a non-zero finite pitch value.

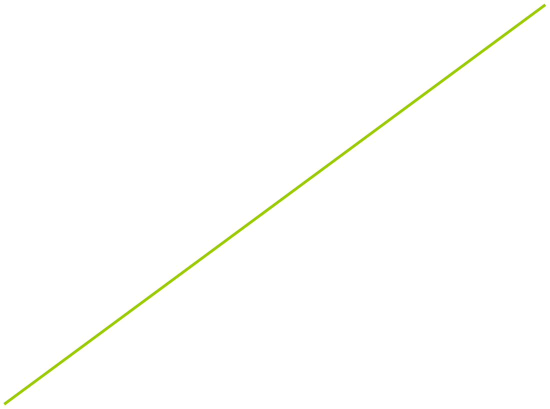


Figure 8.97: Freedom space of Case 5, Type 2

Since this freedom space consists of a single screw, **Equation (3.13)** may be used to determine the system's constraint space. This constraint space is shown in **Figure 8.98**. The constraint lines may be represented as blue lines that are tangent to the surface of a cylinder with a radius of *d* as shown in the figure. The pitch value, *p*, of the freedom space's screw determines the angle, θ, of these constraint lines at designated points on the surface of every cylinder for all real values of *d*.

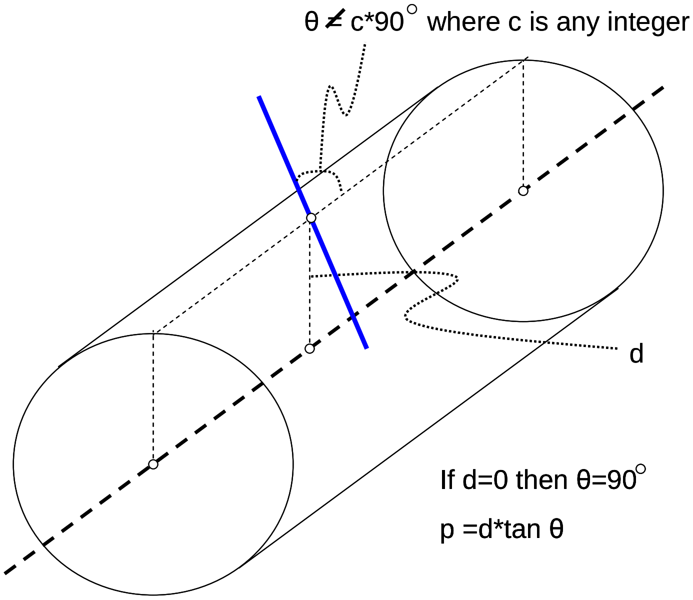


Figure 8.98: Constraint space of Case 5, Type 2

Figure 8.99 shows how the freedom and constraint spaces fit together. The screw line is coincident with the dashed black line in **Figure 8.98**.

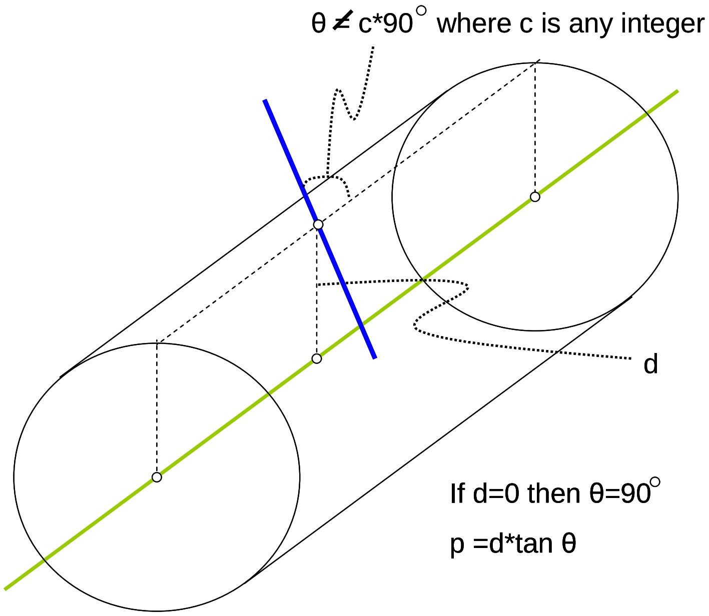


Figure 8.99: Freedom space (green) and constraint space (blue) of Case 5, Type 2 together

8.4.3 Case 5, Type 3:

This section describes the unique constraint space of a system with a freedom space that consists of a single pure translation. This freedom space is shown in **Figure 8.100** as a pure rotational hoop with a normal vector that points in the direction of the translational motion.

Figure 8.100: Freedom space of Case 5, Type 3

Since this freedom space consists of a pure rotational freedom line at infinity, Blanding's Rule of Complementary Patterns may be applied to find the system's constraint space by finding every line that intersects it at least once. This constraint space is shown in **Figure 8.101**. It consists of an infinite number of stacked planar constraint sets that are all parallel to each other. Each of these planes contains every constraint line that lies on its surface.

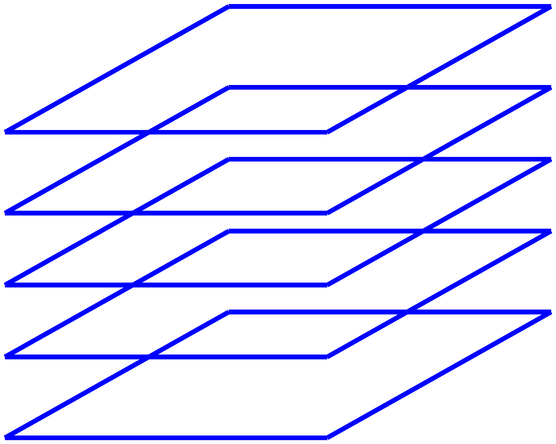


Figure 8.101: Constraint space of Case 5, Type 3

Figure 8.102 shows how the freedom and constraint spaces fit together. The normal vector of the pure rotational hoop is parallel to the normal vectors of the planar constraint sets.

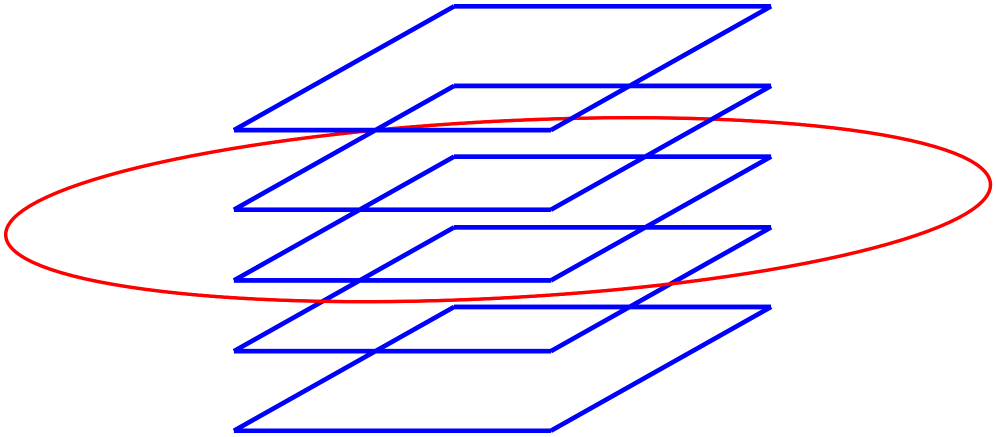


Figure 8.102: Freedom space (red) and constraint space (blue) of Case 5, Type 3 together

8.4.4 Sub-constraint Space of Case 5

This section describes how the concept of sub-constraint space applies to the constraint spaces of Case 5. In short, the sub-constraint spaces of Case 5 are the constraint and sub-constraint spaces of Case 4.

To better understand why this is true, the reader must recognize that every freedom space within Case 4 that contains the type of twist within the freedom space of interest from Case 5 will have a constraint space that is contained within the constraint space of that particular type from Case 5. This must be true because the constraint space of the freedom space of interest from Case 5 contains every constraint line that complements its particular twist while every constraint space of every freedom space that contains the same twist from Case 4 also complements that twist. If this were not true, the twist would not exist within that freedom space from Case 4. The only difference between the Case 5 constraint space and the constraint spaces of Case 4 that contain the same twist within their freedom spaces, is that the Case 4 constraint spaces complement more twists than the single twist that the constraint space from Case 5 complements. The freedom spaces of Case 4 types contain two independent twists while the constraint spaces of Case 5 types contain only a single independent twist. The designer can, therefore, use the sub-constraint spaces of any constraint space within Case 4 with a unique freedom space that contains the twist of interest from the Case 5 freedom space to determine the first four non-redundant constraints of the system. Then the designer can select any constraint line that does not lie within the complete constraint space of the type he/she chose from Case 4 that does lie within the constraint space of the Case 5 freedom space in order to determine the fifth and final non-redundant constraint of the system.

The author is aware that this concept appears extremely complicated and confusing to a first time reader. In actuality, however, it is really quite simple when the explanation is accompanied with an example. An extensive example of this concept will, therefore, be given in the last case study in **Chapter 10**. The reader is, therefore, encouraged to withhold frustration until carefully studying this example before prematurely yielding to confusion.

8.5 Case 6:

This section describes the sixth case of 6. The sixth case consists of all systems that contain 6 non-redundant constraints. Using **Equation (2.1)** from **Chapter 2**, therefore, one deduces that the freedom spaces within Case 6 contain no twists. Furthermore, there is only a single type within this case since a system that is fully constrained and fixed cannot move and, therefore, has only one freedom space containing nothing.

8.5.1 Case 6, Type 1:

This section describes the unique constraint space of a system with an empty freedom space. Since this system's freedom space is empty, its complementary constraint space consists of every constraint line in three-space. Once a system has 6 non-redundant constraints, any other constraint selected from anywhere will result in the same empty freedom space and will, therefore, be redundant.

The freedom and constraint spaces of Case 6, Type 1 are shown in **Figure 8.103**.

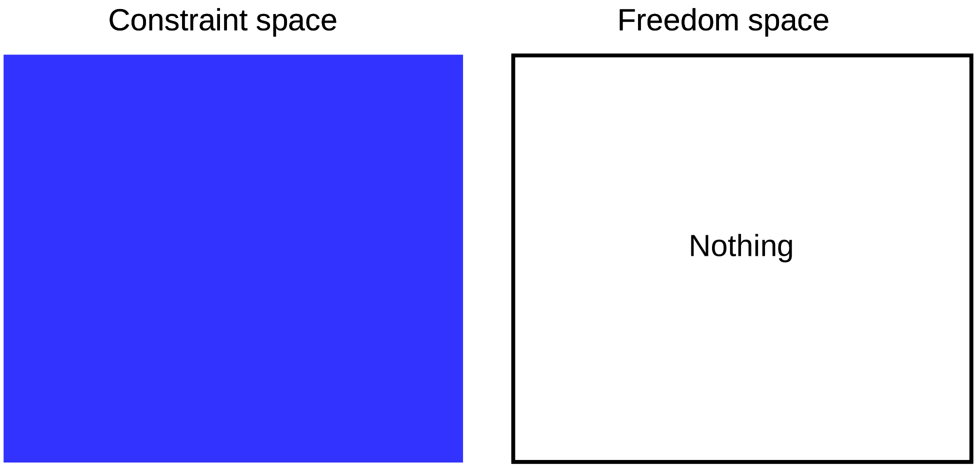


Figure 8.103: Freedom space (white) and constraint space (blue) of Case 6, Type 1

In order to select 6 non-redundant constraints from within this constraint space the designer must again rely on sub-constraint spaces. The sub-constraint spaces of Case 6 are the constraint spaces of Case 5, and the constraint and sub-constraint spaces of Case 4. The concept is similar to Case 5's sub-constraint spaces described in the previous section. The designer first uses the sub-constraint spaces from a particular constraint space within Case 4 to select the first four non-redundant constraints of the system. Then he/she selects the fifth non-redundant constraint from the desired Case 5 constraint space that does not lie within the selected Case 4 constraint space. Then the sixth non-redundant constraint is selected from within the Case 6 constraint space (anywhere) that does not lie within the selected Case 5 constraint space. Again, this concept will make more sense after the example from **Chapter 10**.

CHAPTER 9:

“Systems Symmetry”

This chapter summarizes and compares the cases and types found in the previous two chapters and discusses significant observations relating to patterns of symmetry within their freedom and constraint spaces. Essentially the purpose of this chapter is to tie up loose ends and provide the reader with a “big picture” of the complete theory necessary for a full understanding of the FACT method. A new, special case (Case 0) will also be presented in the context of a broader theory that mathematically describes all possible kinematic systems in three-space, not just systems constrained by ideal compliant flexures.

9.1 Symmetry Within Cases 1 Through 6

This section reviews the 6 cases and their respective types described in **Chapter 7** and **Chapter 8**. In this chapter, the author will only briefly review figures of these spaces assuming the reader has already carefully studied their geometries in previous chapters. Seen together, a surprising symmetry is observed in the types found in Cases 1 through 3 and Cases 4 through 6.

Figure 9.1, **Figure 9.2**, **Figure 9.3**, **Figure 9.4**, **Figure 9.5**, and **Figure 9.6** all review the types within Cases 1 through 6.

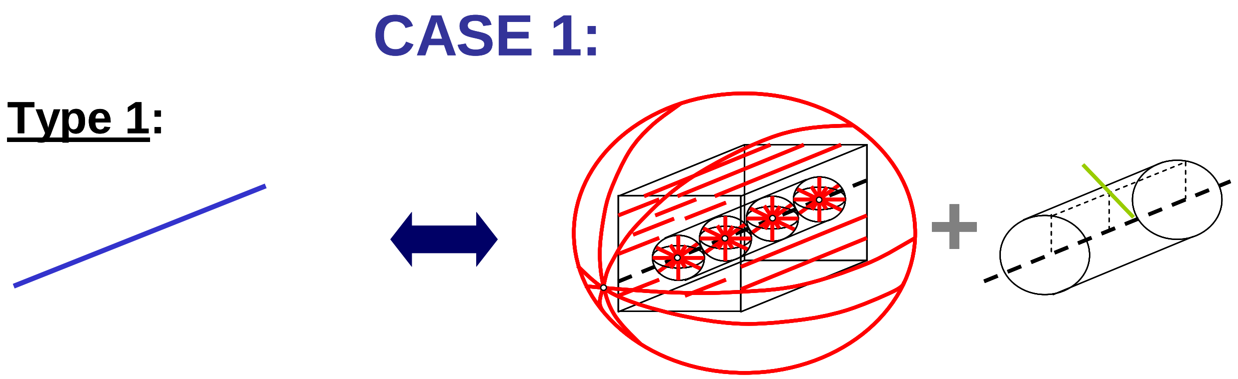


Figure 9.1: Case 1 types (constraint space on the left and freedom space on the right of the arrow)

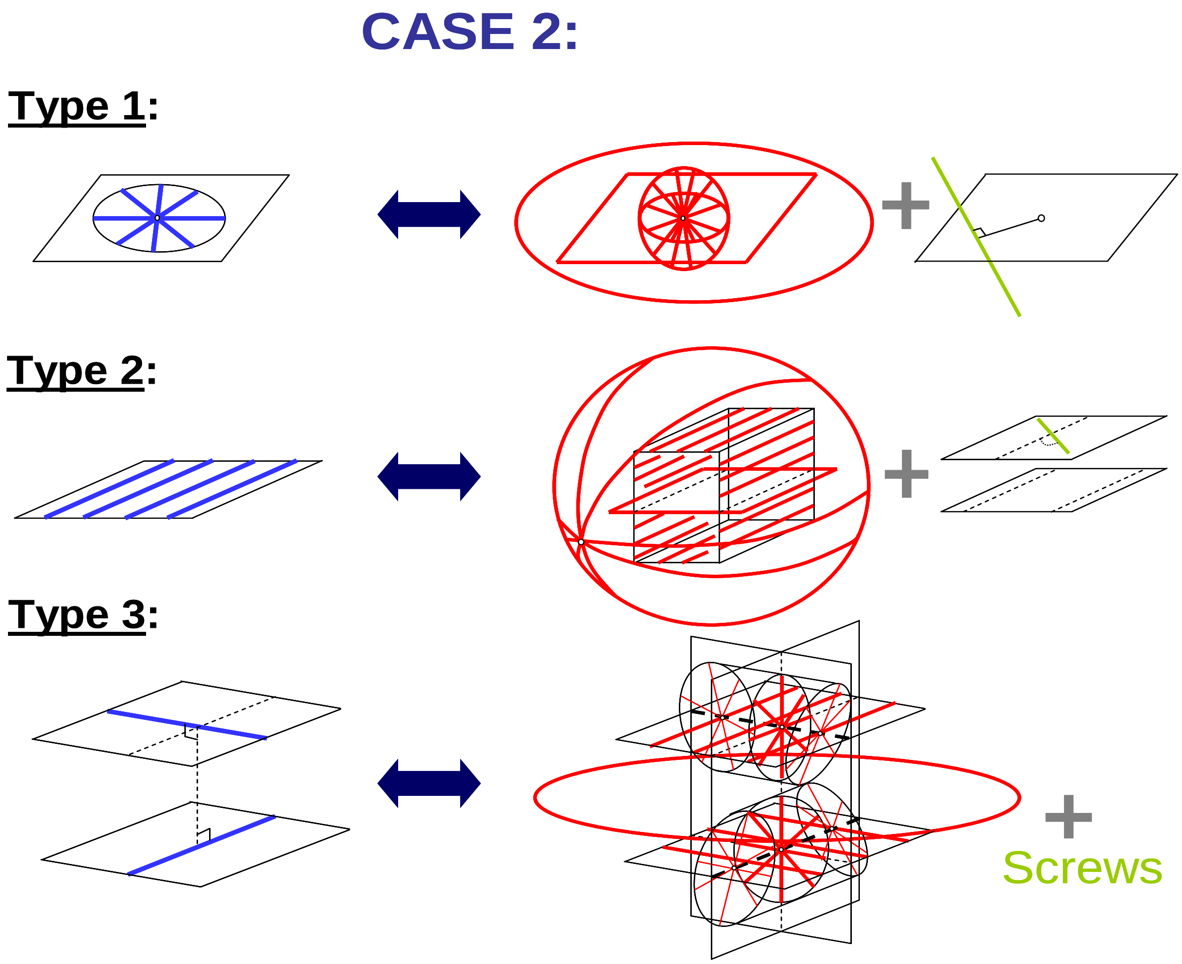


Figure 9.2: Case 2 types (constraint space on the left and freedom space on the right of the arrow)

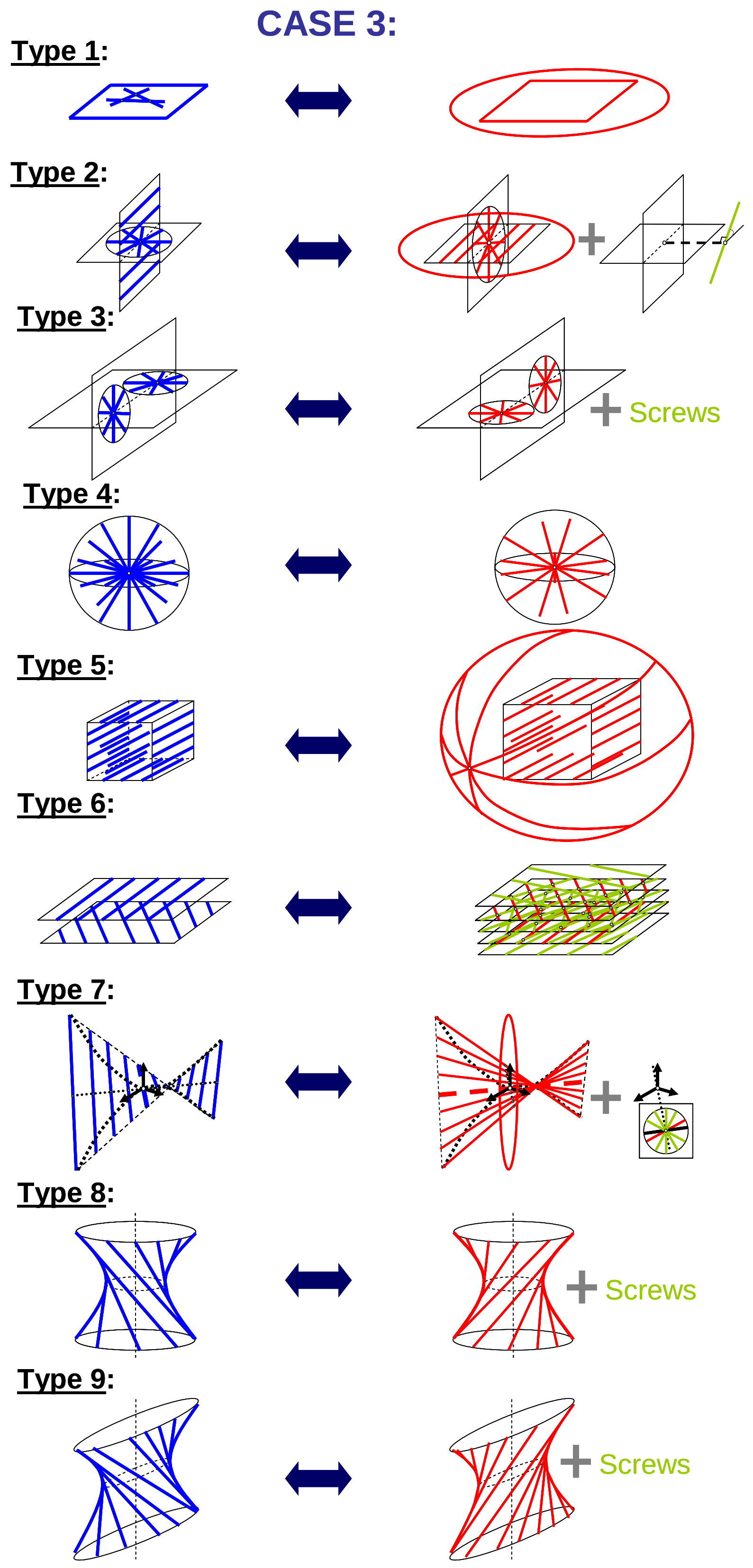


Figure 9.3: Case 3 types (constraint space on the left and freedom space on the right of the arrow)

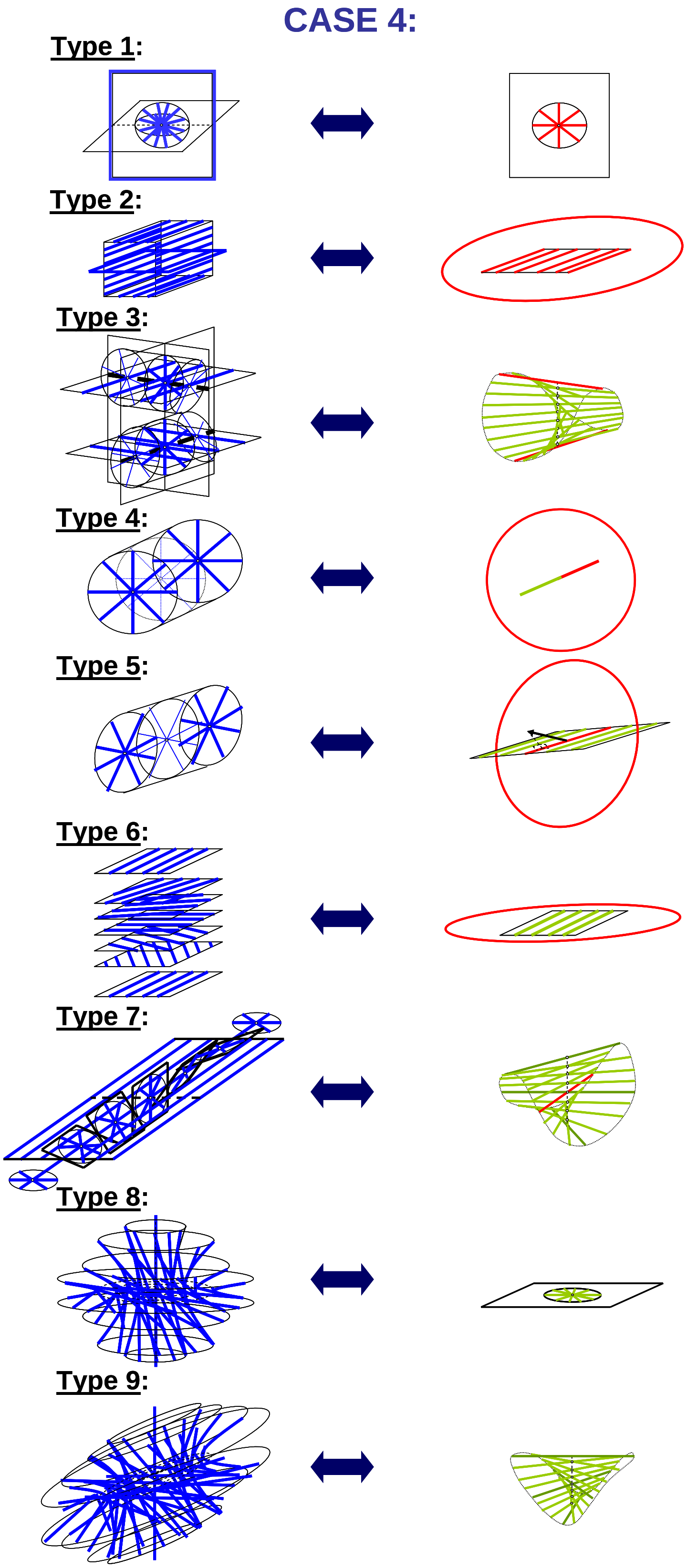


Figure 9.4: Case 4 types (constraint space on the left and freedom space on the right of the arrow)

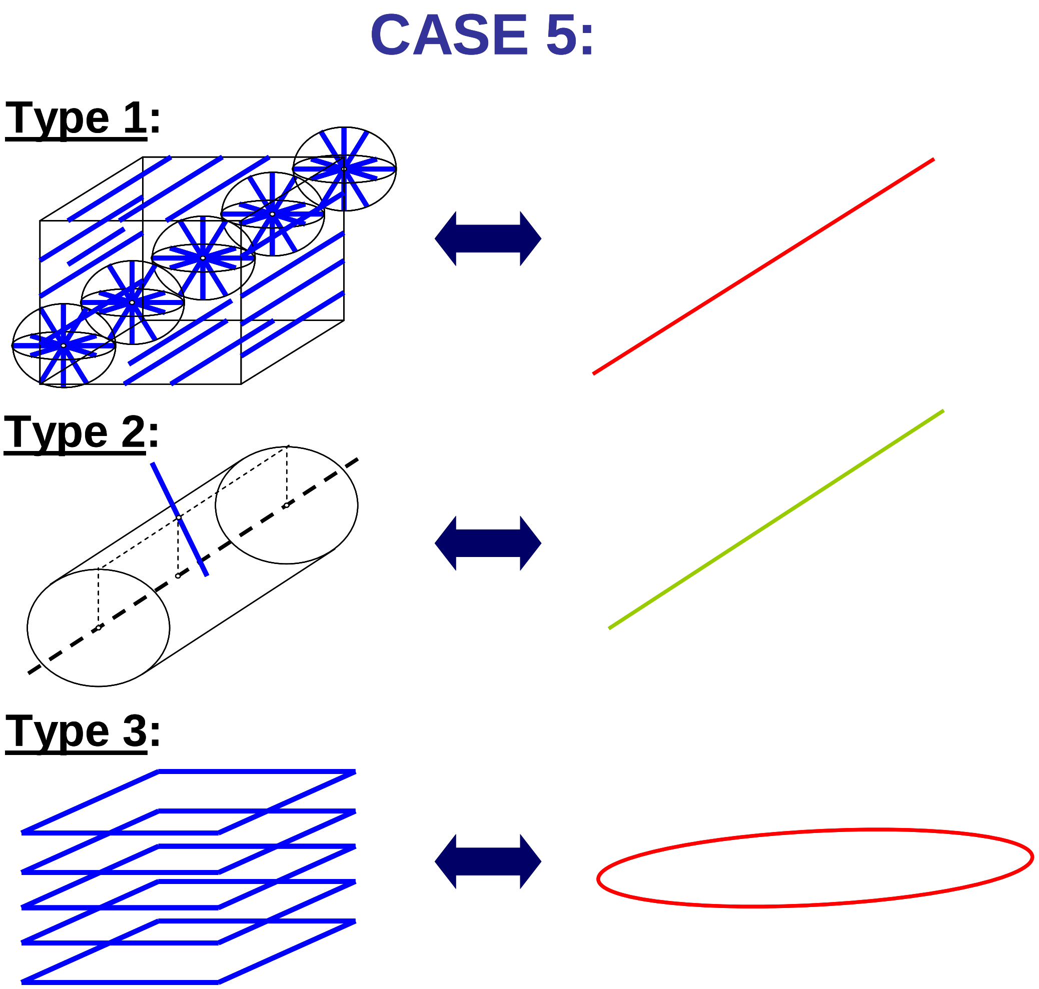


Figure 9.5: Case 5 types (constraint space on the left and freedom space on the right of the arrow)

CASE 6:

Type 1:

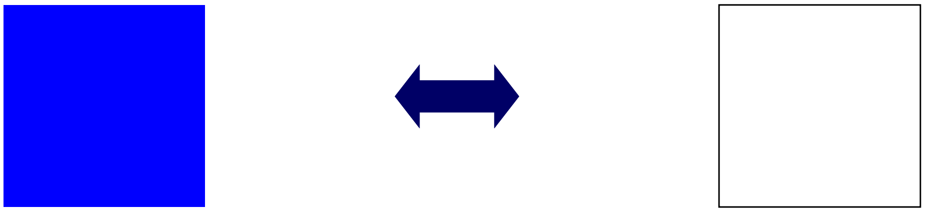


Figure 9.6: Case 6 types (constraint space on the left and freedom space on the right of the arrow)

Observe the increasing number of types in the first three cases. Case 1 has one type, Case 2 has three types, and Case 3 has 9 types. In contrast, note the decreasing number of types within the last three cases. Case 4 has 9 types, Case 5 has three types, and Case 6 has one type. The symmetry within the cases is clearly shown in **Figure 9.7**.

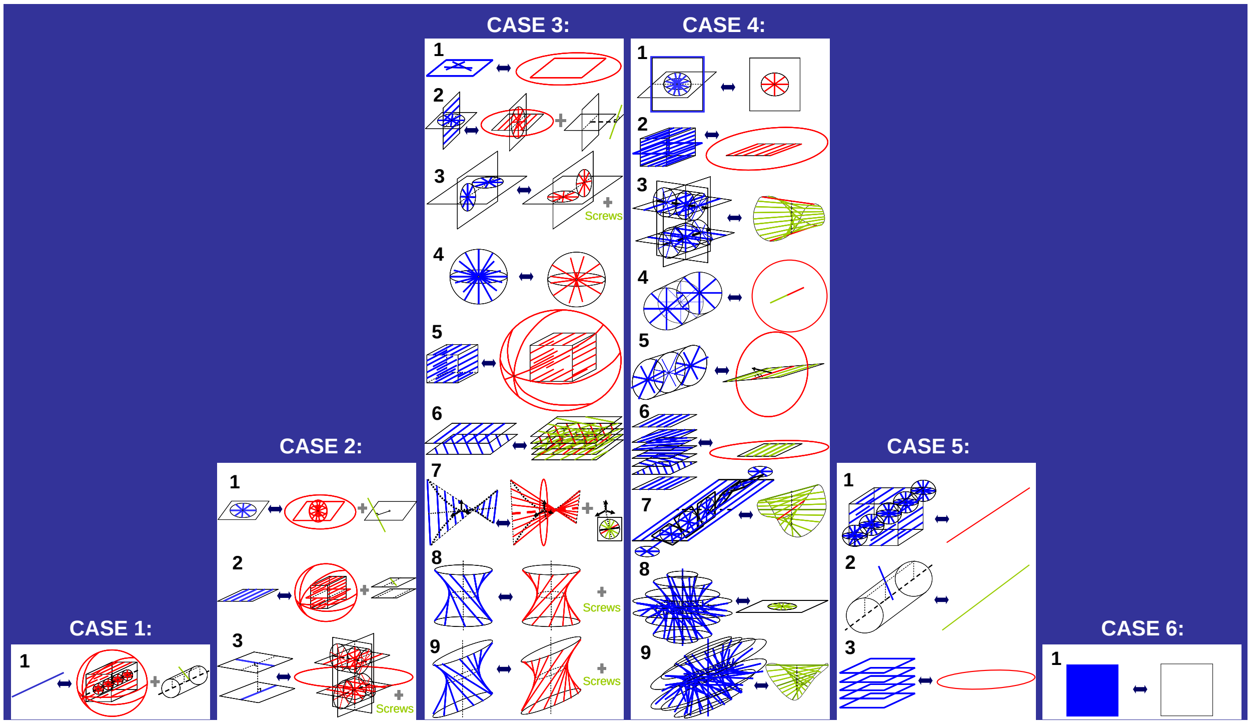


Figure 9.7: Every case and type for all systems of ideal constraints[4]

[4] Merlet [41] and McCarthy [32] classify lines similarly in their works

The symmetry noted above is pleasing but surprising to the author. The author would have expected symmetry within systems that don’t only include ideal constraints capable of only imparting axial forces like the flexure systems studied thus far. The reason for the expected symmetry within these types of systems will be shown in the next section.

9.2 Proof of Symmetry

This section reviews some basic concepts of screw theory discussed in **Chapter 3** and uses these concepts to prove that systems containing all types of constraints are symmetric about Case 3.

Systems that contain all types of constraints instead of just systems that contain ideal constraints only will now be considered. In other words, this section will discuss systems consisting of constraints modeled as wrenches that are allowed to have any real values of q instead of systems consisting of constraints modeled as wrenches that must have *q* values of zero where *q* is defined in **Section 3.2** from **Chapter 3**. Thus far wrenches with *q* values of zero have been represented as blue lines in three-space. Now wrenches with infinite *q* values will be represented as purple lines. Also wrenches with *q* values that are finite and non-zero will be represented as brown lines. This convention is shown in **Table 9.1**.

Table 9.1: Wrench names and line colors for different categories of *q*

***q* Value**	**Name of Wrench**	**Color of Wrench Line**
$q = 0$	Constraint Line	🟦
$q = \infty$	Pure Torque	🟪

$q \neq 0 \neq \infty$	Coupled Force and Torque	━━━━━━━━

Thus far wrenches with q values equal to zero have been called constraint lines. They are only capable of imparting axial forces on the objects they constraint and are analogous with pure rotational freedom twists with pitch values equal to zero. Wrenches with infinite q values are called pure torques. They are only capable of imparting torques about their wrench lines and are analogous to pure translational twists with infinite pitch values. Pure torques could just as well be represented by blue constraint line hoops with infinite radii that are orthogonal to their wrench lines since pure translations are represented by pure rotational hoops as discussed in **Chapter 4**. Wrenches with finite non-zero q values represent constraints that impart both axial forces with coupled torques about their wrench lines. They are analogues to screws with finite non-zero pitch values. Compare **Table 9.1** with **Table 3.1** from **Chapter 3**.

Recall now that a general wrench vector, $\vec{W}$, for all types of constraints for all values of q may be expressed as

$$\vec{W} = \begin{bmatrix} \vec{f} & (\vec{r} \times \vec{f}) + q\vec{f} \end{bmatrix} = \begin{bmatrix} f_x & f_y & f_z & \tau_x & \tau_y & \tau_z \end{bmatrix}, \tag{9.1}$$

where its parameters are defined in **Section 3.2** of **Chapter 3**. Every possible constraint space from Case n may be expressed as a wrench matrix, $[W]$, that contains n rows of independent wrench vectors from within the particular constraint space as shown by

$$[W] = \begin{bmatrix} \vec{W}_1 \\ \vec{W}_2 \\ \vdots \\ \vec{W}_n \end{bmatrix}. \tag{9.2}$$

Recall also that a general twist vector, $\vec{T}$, for all types of degrees of freedom for all pitch values, p, may be expressed as

$$\vec{T} = [\vec{w} \quad (\vec{c}\times\vec{w}) + p\vec{w}] = [w_x \quad w_y \quad w_z \quad v_x \quad v_y \quad v_z], \tag{9.3}$$

where its parameters are defined in **Section 3.1** of **Chapter 3**. Every possible freedom space from Case *n* may, therefore, be expressed as a twist matrix, $[T]$, that contains 6-*n* rows of independent twist vectors from within the particular freedom space as shown by

$$[T] = \begin{bmatrix} \vec{T}_1 \\ \vec{T}_2 \\ \vdots \\ \vec{T}_{6-n} \end{bmatrix}. \tag{9.4}$$

Suppose now a new operator, $\Re$, is defined that switches the first three columns of an $m\times 6$ matrix with the last three columns such that

$$\Re\begin{bmatrix} A & B & C & D & E & F \\ G & H & I & J & K & L \\ M & N & O & P & Q & R \end{bmatrix} = \begin{bmatrix} D & E & F & A & B & C \\ J & K & L & G & H & I \\ P & Q & R & M & N & O \end{bmatrix}, \tag{9.5}$$

for an arbitrary 3×6 matrix. Using **Equation (3.12)** from **Chapter 3**, therefore, any constraint space's relationship with its unique freedom space can be represented as

$$[W]\bullet(\Re[T])^T = [0], \tag{9.6}$$

or as

$$[T]\bullet(\Re[W])^T = [0], \tag{9.7}$$

where $[0]$ is an appropriately sized matrix filled with all zero values.

Suppose now an arbitrary constraint space is defined for a system, (*a*), with a wrench matrix $[W_a]$. Using **Equation (9.6)** its complementary unique freedom space could be represented by a twist matrix $[T_a]$. Now suppose one wished to define another system, (*b*), with a freedom space that looks geometrically identical to the last system's constraint space. In other words,

$[T_b]=[W_a]$ or the rows of $[T_b]$ are made to be independent linear combinations of the rows of $[W_a]$. The only visual difference between the constraint space of system (*a*) and the freedom space of system (*b*) is that every line that is blue in system (*a*) is shown as red in system (*b*), every line that is purple in system (*a*) is shown as black in system (*b*), and every line that is brown in system (*a*) is shown as green in system (*b*). According to **Equation (9.6)** and **Equation (9.7)**, system (*b*)'s constraint space matrix, $[W_b]$, would either be equal to system (*a*)'s freedom space matrix, $[T_a]$, or its rows would be independent linear combinations of the rows within $[T_a]$. In other words, system (*b*)'s constraint space would look geometrically identical to system (*a*)'s freedom space where the red lines would be drawn as blue lines, the black lines would be drawn as purple lines, and the green lines would be drawn as brown lines.

Every Case *N* constraint space, therefore, looks geometrically identical to some Case *6-N* freedom space and every Case *N* freedom space looks geometrically identical to some Case *6-N* constraint space. This means that systems containing all types of constraints are symmetric about Case 3.

Traces of this conclusion are seen within the freedom and constraint spaces of the systems that contain only ideal constraints (*q*=0) shown in **Figure 9.1** through **Figure 9.6**. Note the geometric similarities in Case 1, Type 1's freedom space and Case 5, Type 1's constraint space as well as Case 1, Type 1's constraint space and Case 1, Type 1's freedom space. Also note the geometric similarities in Case 2, Type 1's freedom space and Case 4, Type 1's constraint space as well as Case 2, Type 1's constraint space and Case 4, Type 1's freedom space. Also note the geometric similarities in Case 2, Type 2's freedom space and Case 4, Type 2's constraint space as well as Case 2, Type 2's constraint space and Case 4, Type 2's freedom space. Also note the geometric similarities in Case 2, Type 3's freedom space and Case 4, Type 3's constraint space as well as Case 2, Type 3's constraint space and Case 4, Type 3's freedom space. The only geometric difference between these spaces is that the freedom spaces show every type of twist for all pitch values while the constraint spaces only show wrenches with zero *q* values.

Note also that every freedom space within Case 3 looks almost identical in shape to its complementary constraint space. This observation is true because Case 3 is symmetric about itself. For this reason, all the doubly ruled surfaces show up as constraint and freedom space pairs within Case 3. If wrenches for all *q* values where determined for every system within Case 3, one would find that every freedom space within Case 3 is exactly identical in shape to its complementary constraint space.

The reader may wonder what case is symmetric to Case 6 if all cases are symmetric about Case 3. The answer is given in the next section.

9.3 Case 0:

This section describes a special extra case called Case 0. This case consists of all systems with zero or no non-redundant constraints. In other words, it consists of systems of free standing bodies that are not constrained by anything. Since there is only one way to not constrain an object, it is evident that this case has only a single type. From the conclusion in **Section 9.2**, this type's freedom space would be expected to be similar to the constraint space of Case 6, Type 1 and this type's constraint space would be expected to be similar to the freedom space of Case 6, Type 1 since Case 6 is symmetric with Case 0 about Case 3.

9.3.1 Case 0, Type 1:

This section describes the unique freedom space of a system with an empty constraint space. Since this system's constraint space contains no constraints, its complementary freedom space consists of every twist in three-space. In other words, its freedom space includes every pure rotational freedom line (red), every screw with every finite non-zero pitch value (green), and every pure rotational hoop that may be represented by a sphere of pure translations that point in all directions (black). This should be obvious since a free standing object may, by definition, move with any motion possible being completely unrestricted.

The constraint space of Case 0, Type 1 is shown in **Figure 9.8** and the freedom space of Case 0, Type 1 is shown in **Figure 9.9**.

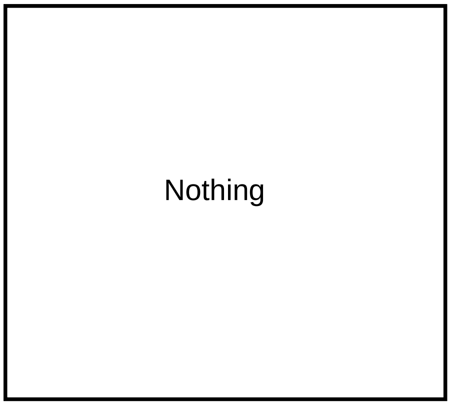


Figure 9.8: Constraint space (white) of Case 0, Type 1

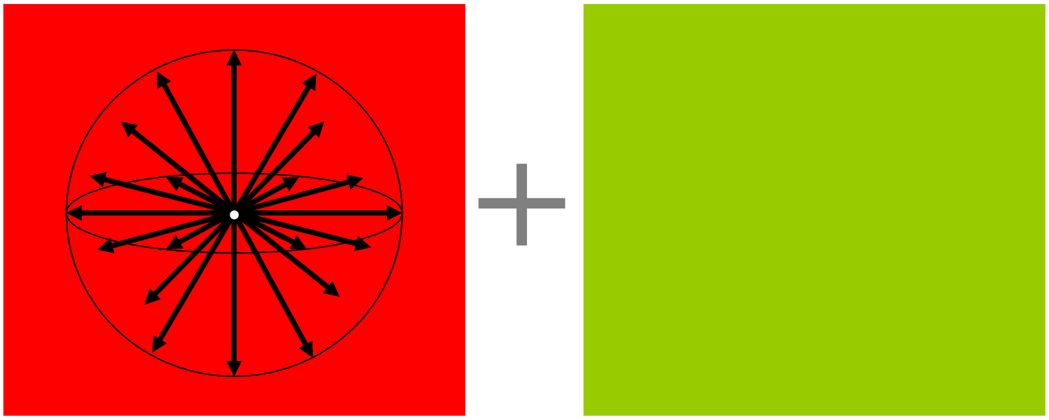


Figure 9.9: Freedom space (red, green, and black) of Case 0, Type 1

9.4 More Types Within Each Case

Now that systems that include all types of wrenches are being considered, one must also suppose that many of the constraint spaces found in past chapters would include more lines that were not accounted for because they initially contained only ideal constraints with zero *q* values (The new lines would be brown and purple according to the convention). Furthermore, one should also suspect the existence of other new types within many of the cases. This section discusses these types that are not accounted for.

To understand why more types exist that are not accounted for within systems that consist of all types of wrenches, consider the freedom space of Case 4, Type 7 shown in **Figure 9.4**. It consists of a cylindroid of screws (green) with a single pure rotational freedom line (red). From symmetry one would expect an identical constraint space to exist within Case 2 that consists of a single ideal constraint line (blue) on the surface of a cylindroid of wrenches with finite, non-zero *q* values (brown). This constraint space is, however, no where to be found in **Figure 9.2**. The reason for this is that, although this constraint space does consist of two independent wrenches and, therefore, does belong to Case 2, the constraint space does not contain two non-redundant ideal constraint lines but only one.

Since all the freedom spaces of Cases 4, 5, and 6 were found in **Chapter 8**, and since they all include every type of twist possible (pure rotations, pure translations, and screws), the number of types that must exist within those cases can be known even without a full understanding of what their complementary unique constraint spaces look like. Recall from **Section 8.1** of **Chapter 8** that Case 4 would actually have 10 types for systems that include all types of wrenches where the tenth type's freedom space is a disk of pure translations. Recall also that Case 5 has three types and Case 6 has only one type.

From symmetry, therefore, the complete constraint spaces that include all types of wrenches for Cases 0, 1, and 2 are known. They will be identical to the freedom spaces of Cases 6, 5, and 4 respectively. Case 0 must have one type, Case 1 must have three types (one for each type of wrench), and Case 2 must have 10 types. The author does, however, not know how many types Case 3 would have for systems that include all types of wrenches. It is certain that there would be more than the 9 types found in **Chapter 7** for systems of ideal constraints, but more research is necessary to determine exactly how many more and what these new types would look like. There are a finite number of these new types and their freedom spaces would look identical to their complementary constraint spaces since Case 3 is symmetric about itself.

To give the reader an idea of what some of these new spaces would look like, the author has found two of the new types within Case 3 for systems that include all types of wrenches. They are shown in **Figure 9.10**. One type consists of a spherical constraint space of wrenches with finite, non-zero *q* values with a complementary spherical freedom space of screws with finite, non-zero pitch values. All the wrenches have the same *q* values in the constraint space and all the screws have the same pitch values in the freedom space and both of these values are also equivalent with each other. The second new type consists of a spherical constraint space of pure torques with infinite *q* values and a complementary spherical freedom space of pure translations with infinite pitch values. Case 3, therefore, has at least 11 types.

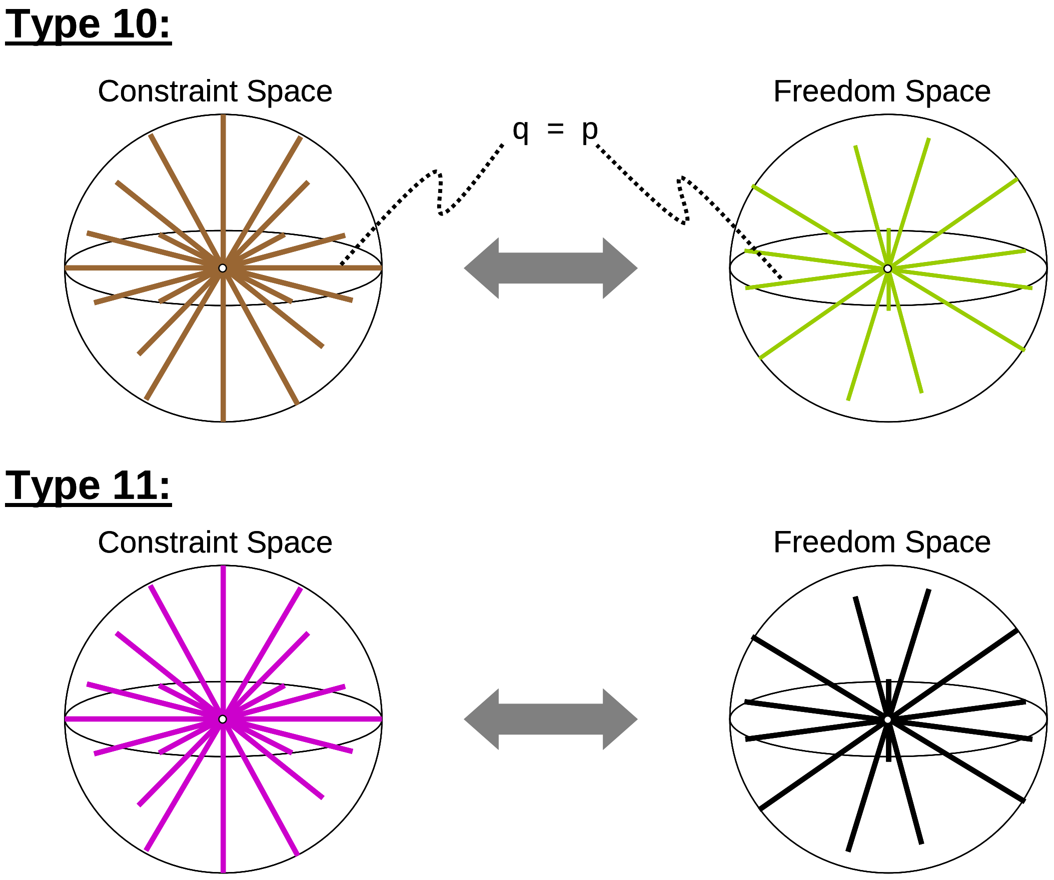


Figure 9.10: Two new types within Case 3 for systems that include all types of wrenches

The results of this chapter's discussion are shown in **Table 9.2**. This table demonstrates the proven symmetry about Case 3 among the number of types within each case for systems that include all types of wrenches. It also compares these systems' symmetry with the unexpected symmetry observed within systems of ideal constraints studied in the rest of this thesis.

Table 9.2: The number of types within each case for systems consisting of all types of wrenches (*q*=any real number) and systems consisting of only ideal constraints (*q*=0). The number of types shown in red demonstrates the symmetry observed in this thesis for flexure systems with ideal constraints.

***q* Values**	**Case 0**	**Case 1**	**Case 2**	**Case 3**	**Case 4**	**Case 5**	**Case 6**
q=any real number	1	3	10	?	10	3	1
q=0	1	1	3	9	9	3	1

It should now be emphasized that although more types do exist outside the 26 types discussed and described in the rest of this thesis, those 26 types shown in red in **Table 9.2** are the only types with any useful spaces for designing flexure systems. In fact, those spaces alone fully describe all the possible kinematics and constraint topologies for most real world systems because most real world systems consist of constraints that are best modeled as ideal. For this reason, the author has not pursued the study of systems that include other types of constraints further.

CHAPTER 10:

"FACT Design Method"

This chapter describes how flexure systems may be designed using Freedom and Constraint Topology (FACT). Essentially it demonstrates the utility of the theory developed thus far. The 6 step FACT design method is first presented and discussed followed by three comprehensive design case studies.

10.1 Six Steps of FACT

This section introduces the 6 steps of the FACT design method. Each step will be discussed in detail in subsequent subsections. The 6 steps are shown in **Figure 10.1**.

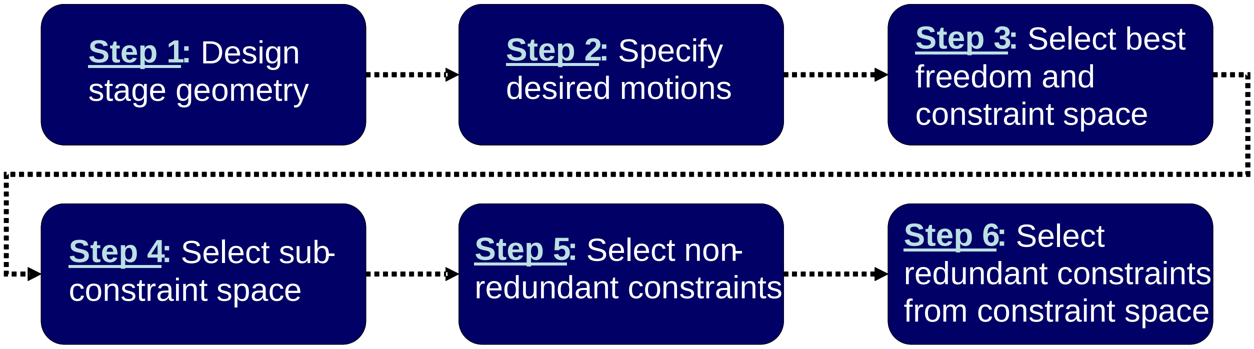

Step 1: Design stage geometry
Step 2: Specify desired motions
Step 3: Select best freedom and constraint space
Step 4: Select sub-constraint space
Step 5: Select non-redundant constraints
Step 6: Select redundant constraints from constraint space

Figure 10.1: 6 steps of the FACT design method for designing flexure systems

10.1.1 <u>Step 1:</u> Design Stage Geometry

The first step of the FACT design method is to design flexure system's stage. This step has no restrictions since the allowable motions or degrees of freedom of a system are completely independent of the shape, orientation, or size of its stage as was discussed in **Chapter 2**. The designer, therefore, has the freedom to design any imaginable shape on any scale as long as it is possible to manufacture. The size of the stage designed determines the system's characteristic length. This length imposes restrictions on the minimum length of the system's constraints and largely determines how far apart the constraints should be located for greatest stability. There will be more discussion of this concept in later sections.

10.1.2 Step 2: Specify Desired Motions

The second step of the FACT design method is to specify the flexure system's desired motions, i.e. freedom topology, or degrees of freedom. In other words, the designer determines what twists the stage should be able to move with. There are no restrictions on this step either. Any pure rotation, any pure translation, and any screw with any pitch value may be specified in any location and in any orientation in three-space.

10.1.3 Step 3: Select Best Freedom and Constraint Space

The third step of the FACT design method is to select the most appropriate freedom and constraint space pair or type from within the most appropriate case based on the twists specified in Step 2. At least one of the 26 possible freedom spaces will always contain the twists specified in Step 2. Those that do contain all of the specified twists will often contain many more twists that are not desired by the designer. The best or most appropriate freedom space for any given group of specified twists is defined as the space that contains all the specified twists with as few extra, unwanted twists as possible. In many cases, the designer will have to make due with a flexure system that may move with undesired motions, but the fewer undesired motions the flexure system has, the easier it will be to control such that it only moves with the desired motions specified from Step 2.

Selecting the best freedom and constraint space pair from the 26 available types is in many instances a difficult task for a novice designer when presented with an arbitrary group of twists from Step 2. Fortunately, a program may be written which reliably performs this step independent of the designer. In short, therefore, once the designer has simply determined what he/she wants from Steps 1 and 2, a computer tool may be utilized to provide him/her with the best constraint space that contains every possible answer for achieving the desired motions with as few undesired motions as possible.

10.1.4 Step 4: Select Sub-constraint Space

The fourth step of the FACT design method is to select a desired sub-constraint space from within the constraint space determined in Step 3. If the constraint space determined in Step 3 belongs to Case 1, 2, or 3, the sub-constraint space is simply the constraint space and no choice needs to be made. If, however, the constraint space belongs to Case 4, there will be a number of sub-constraint spaces to choose from. If the constraint space from Step 3 belongs to Case 5, a constraint space from Case 4 that belongs within that constraint space must be chosen as well as a sub-constraint space within that Case 4 constraint space. If the constraint space from Step 3 belongs to Case 6, a constraint space from Case 5 must be selected as well as a constraint space from Case 4 that lies within that Case 5 constraint space. Finally a sub-constraint space from that Case 4 constraint space must also be selected.

The selection of sub-constraint spaces is a design decision that narrows down the possible solutions or constraint topologies of the flexure system being designed. The decision should be made based on geometric constraint considerations, symmetry, or balanced stiffness requirements of the final flexure system. The location and geometry of the ground with respect to the stage, for instance, would largely determine which sub-constraint space to select. Some sub-constraint spaces, for example, may not contain constraint lines that pass through the ground and stage. Selecting different sub-constraint spaces will have different consequences on the flexure system's final appearance and performance. Each sub-constraint space should, therefore, be thoughtfully considered before a final decision is made.

10.1.5 Step 5: Select Non-redundant Constraints

The fifth step of the FACT design method is to select appropriate non-redundant constraints from the sub-constraint space chosen in Step 4. The number of non-redundant constraints that need to be selected is the system's case number. Instructions are included with every sub-constraint space that guide the designer in appropriately selecting the non-redundant constraints. If the system is from Case 5, the fifth non-redundant constraint will be a constraint that exists within its constraint space but that does not exist within the constraint space of the Case 4 constraint space from which the sub-constraint space was selected. This will also be true for a system from Case 6. In this case, however, the sixth non-redundant constraint will be a constraint that does not lie within the Case 5 constraint space chosen but that does lie within the constraint space of Case 6. An example of this concept is provided in the last case study of this chapter.

Non-redundant constraints chosen from constraint sets should be as far apart from each other as possible. At very least they should be a characteristic length apart as determined by the size of the stage of the flexure system designed in Step 1. The reason for this is that the farther apart constraints are situated, the greater stability the flexure system will have. This statement is true because external disturbance torques will best be resisted as the resisting torque moment arms are increased.

Furthermore, constraints chosen should be able to be long enough to act as ideal constraints in that they are very compliant in directions perpendicular to their axes, but short enough so as to not buckle under realistic axial loads. They should at least be a characteristic length long and must span between ground and the stage of the flexure system. These considerations should also influence the designer in selecting appropriate sub-constraint spaces in Step 4.

It is important to note, that once Step 5 is complete, the flexure system is non–redundantly constrained and therefore the stage will move with the desired degrees of freedom. In some instances, the designer could stop here having achieved his/her objective.

10.1.6 Step 6: Select Redundant Constraints

The sixth step of the FACT design method enables the designer to select redundant constraints from the system's constraint space. This step is sometimes desirable if symmetry, increased stiffness or load capacity is necessary. By adding redundant constraints, the flexure system's kinematics will not change, but sometimes the designer can achieve more robust designs by adding the extra constraints. Adding redundant constraints such that the mechanism will be symmetric, for instance, will make the system impervious to thermal expansion errors. Also, systems redundantly constrained can afford some constraint failure without loosing kinematic performance. Furthermore, suppose the designer wishes to add stiffness to the system without making the existing constraints shorter or thicker. This is achieved by adding redundant constraints.

10.2 Three Case Studies

This section provides three practical examples of flexure systems designed using the FACT design method. The first of these case studies is the design of a compliant spherical ball joint, the second is the design of a compliant probe for a five axis STM, and the third is the design of a three-dimensional compliant rotary flexure.

10.2.1 Compliant Spherical Ball Joint

This section demonstrates the design of a compliant spherical ball joint using the FACT design method. Before proceeding, however, it must be emphasized that the main purpose of this section is not to introduce a new mechanism, but rather to demonstrate how quickly an effective flexure system design may be conceived using the FACT design steps for a Case 3 system.

Ball joints are useful machine elements on any scale. They consist of a linkage with a ball on one end fitted inside a cup-like casing on the end of a second linkage. This type of joint prevents translations while allowing three independent, orthogonal rotations. Ball joints are common in nature as hip joints. A traditional ball joint is shown in **Figure 10.2**.

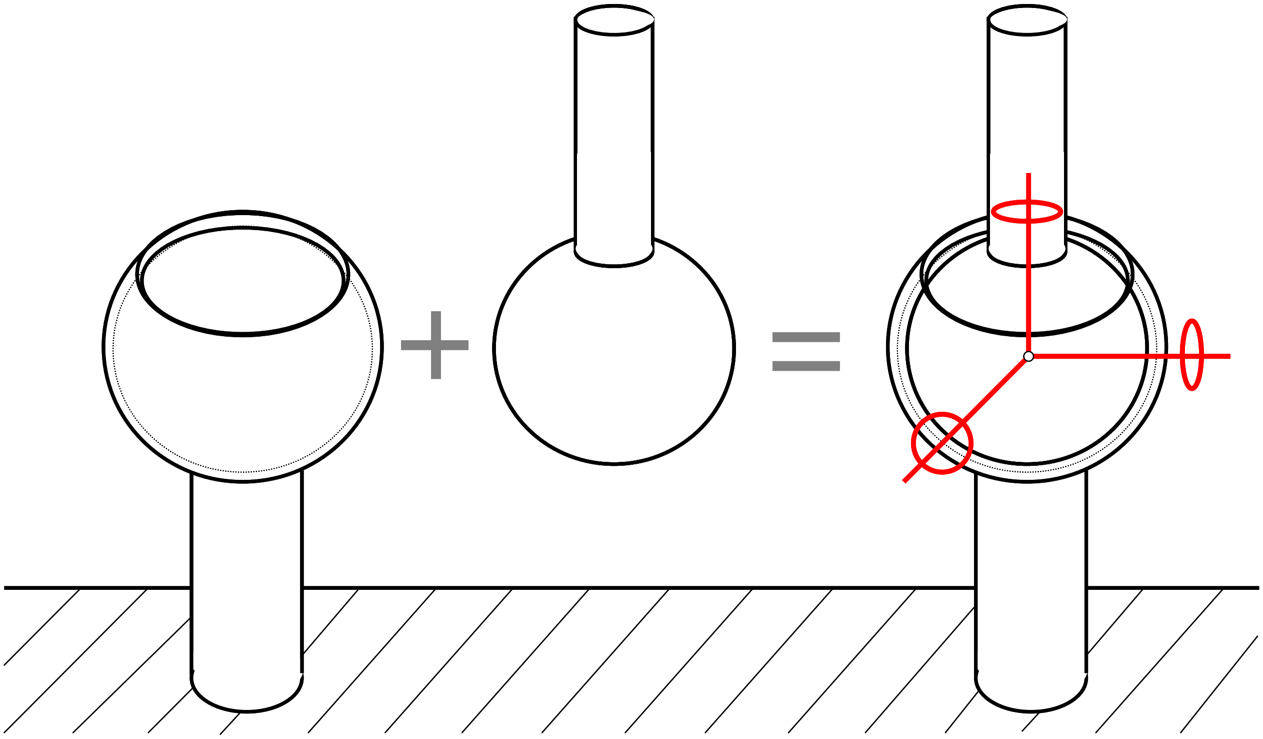


Figure 10.2: Spherical ball joint

Traditional spherical ball joints consist of two parts that experience wear as they rub against each other and generate heat through friction. If a compliant version of a spherical ball joint could be developed, the problems of heat generation, friction and wear would largely be eliminated and a sufficient range of motion could be achieved for small, micro- or nano-devices. In most cases, these motions would be repeatable with atomic precision so long as the device is not actuated beyond its elastic region.

The 6 steps of the FACT design method will now be applied to develop a compliant version of the spherical ball joint. Step 1 requires the designer to first design the spherical ball joint's stage. Since this stage may be any shape, the same stage from the traditional spherical ball joint will be used. This is shown in **Figure 10.3**.

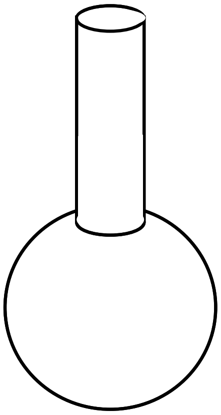


Figure 10.3: Stage of the compliant spherical ball joint designed from Step 1

Step 2 requires the designer to now specify the degrees of freedom he/she wishes this stage to move with. In order to imitate the kinematics of the traditional spherical ball joint, three orthogonal, independent pure rotations are chosen that all intersect at the center of the sphere as shown in **Figure 10.4**.

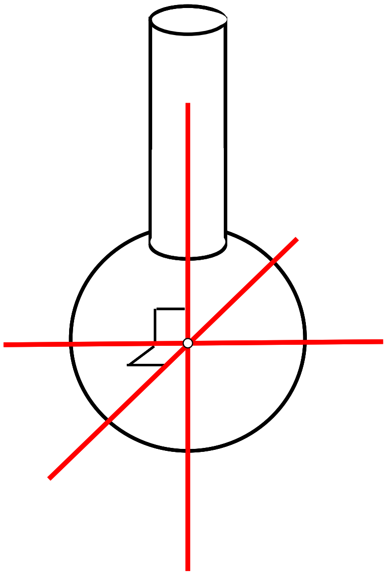


Figure 10.4: Desired degrees of freedom specified from Step 2

Step 3 requires the designer to select the most appropriate freedom and constraint space pair for this system. Since it consists of three independent twists as specified from Step 2, one recognizes that this system belongs to a freedom space from Case 3. One should also recognize that the most appropriate type within Case 3 for this system is Type 4. The most appropriate freedom and constraint space pair for this system is shown in **Figure 10.5**. Note that this system will not only move with three orthogonal intersecting pure rotations as specified in Step 2, but will move with any pure rotational freedom line that intersects the center point of the sphere.

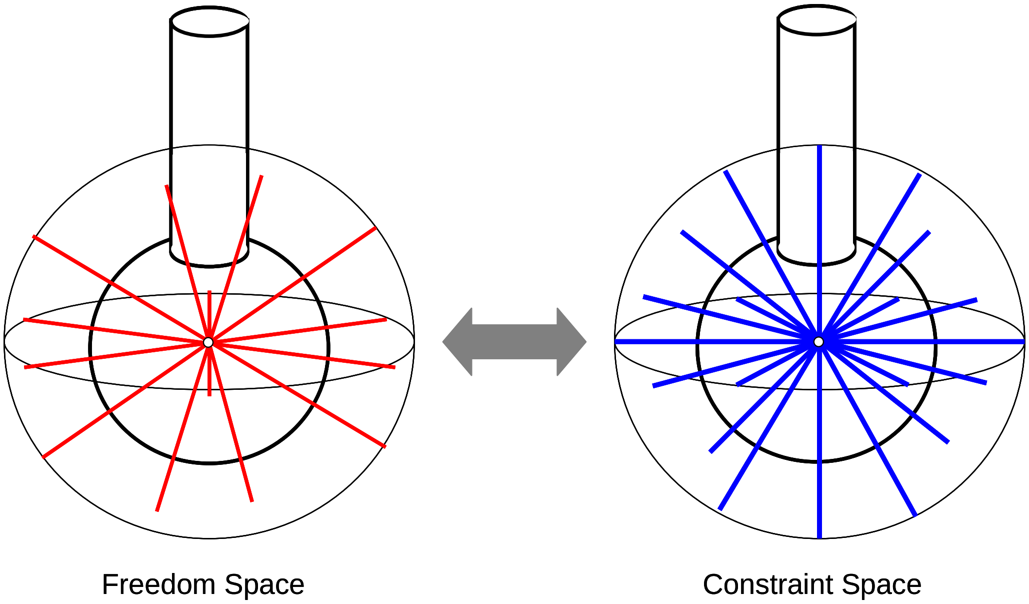


Figure 10.5: Case 3, Type 4 selected for Step 3 given the desired degrees of freedom from Step 2

Step 4 requires the designer to now choose a sub-constraint space for the system. Since this system belongs to Case 3, however, there is only one sub-constraint space to choose from. This sub-constraint space is shown in **Figure 10.6**. It includes instructions to the designer for selecting constraints that are non-redundant.

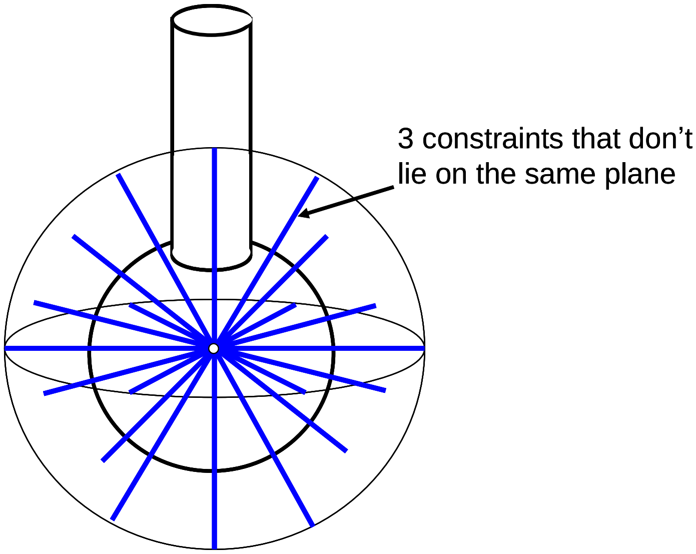


Figure 10.6: Sub-constraint space selected from Step 4

Step 5 requires the designer to select three non-redundant constraints from the sub-constraint space chosen from Step 4. Recall that the most stable system possible will be the system whose constraint lines are as far from each other as possible. This is achieved by selecting a tripod of constraint lines from the sphere that are all orthogonal to each other. One possible solution is, therefore, shown in **Figure 10.7**.

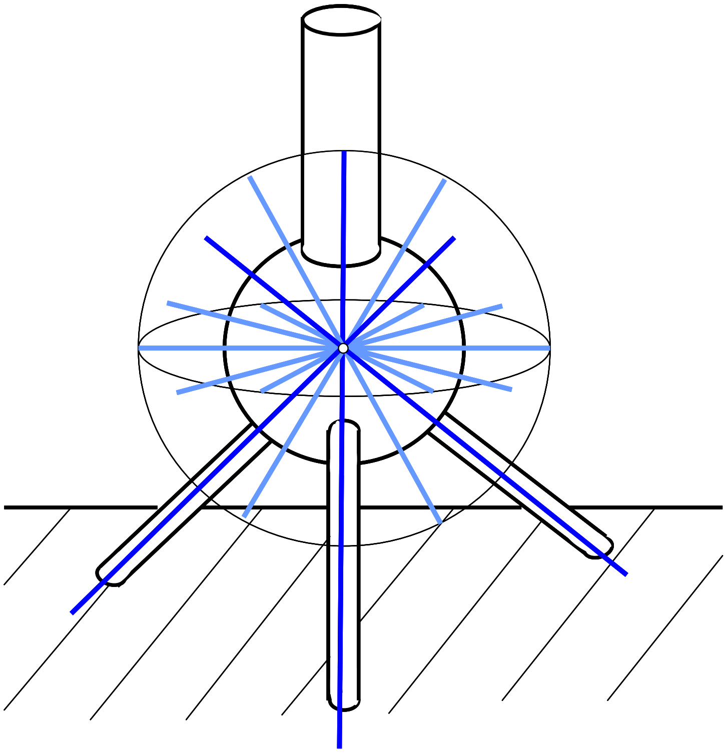


Figure 10.7: Selecting non-redundant constraints from Step 5 that are as far apart as possible (dark blue lines are the constraint lines selected)

At this point, a compliant spherical ball joint has successfully been designed that moves with the exact same degrees of freedom as a traditional two piece spherical ball joint. One could now proceed to Step 6 by selecting more redundant constraints from the constraint space of the system, but because increased symmetry, load capacity, and stiffness are not absolutely necessary for the purposes of this mechanism, the design process will stop here. The final design is shown in **Figure 10.8** next to a traditional spherical ball joint.

This flexure system could also find applications in precision optics. Its topology could be used to constrain a lens such that it may only move about a fixed focal point.

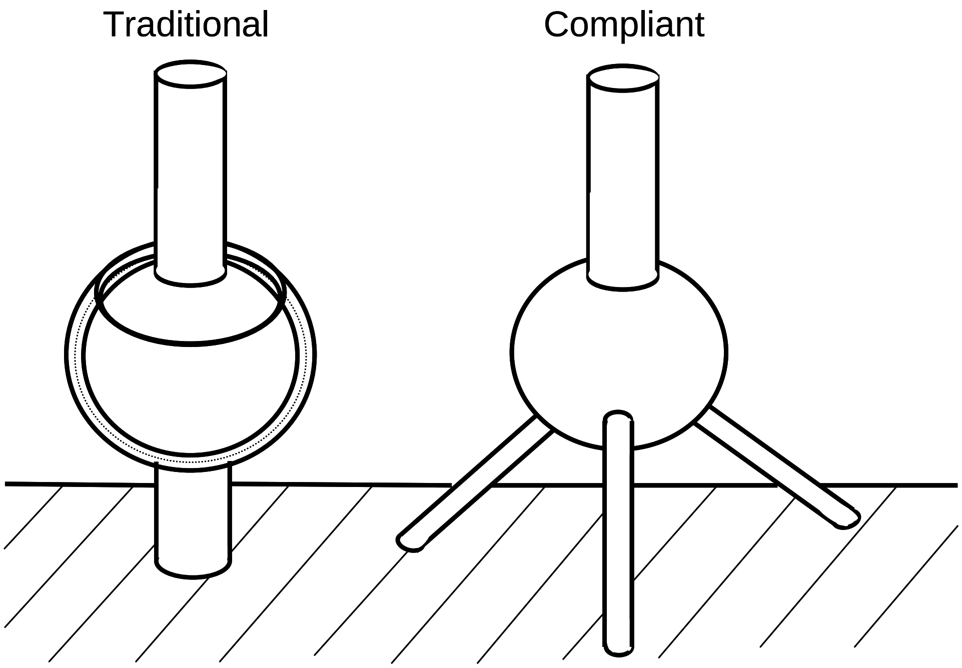


Figure 10.8: Traditional and compliant versions of a spherical ball joint

10.2.2 Compliant Probe For a Five Axis STM

This section demonstrates the design of a compliant probe used to achieve five axes of scanning for a Scanning Tunneling Microscope (STM). The purpose of this section is to demonstrate FACT for a Case 4 system.

Traditional STMs create images of surfaces using a three degree of freedom scanning system [42]. A probe is held fixed in the head of the microscope suspended above the sample being imaged. The sample rests on top of a piezo actuator that provides the three degrees of freedom—x, y, and z translations. Electrons are tunneled through the STM's probe and into the sample being imaged across a gap with a distance that is exponentially related to the tunneling current. A control system is used to maintain this gap distance by maintaining a constant tunneling current as the sample raster scans back and forth under the probe while the probe is tracking the sample's surface. The image of the surface is constructed from the feedback control voltage on the vertical z-piezo element as it corrects for changing surface-to-tip distance.

Three primary sources of error occur when imaging with a traditional three axis STM. The first error is caused by geometric incompatibilities between the probe and the surface being scanned. Often times the probe is not small enough or is not properly oriented to fit in or around cracks and dips in the surface. The second error is caused by the fact that the tunneling current doesn't always tunnel through the axis of the probe. If the probe is against a wall inside a dip on the sample, the electrons may discharge laterally. The third error is caused by the probe's tip bending due to electrostatic forces while scanning.

If two extra degrees of freedom were added to the scanning process (pitch and yaw of the probe) such that the STM becomes a five axis STM instead of a traditional three axis STM, all three imaging errors would be reduced and the microscope's resolution and accuracy would be improved [43]. This is the case because adding pitch and yaw to the probe would more easily allow the probe to be as orthogonal to the sample's surface as possible. The probe could reach places it couldn't normally reach with the extra degrees of freedom. The resulting decrease in imaging error is demonstrated in **Figure 10.9**.

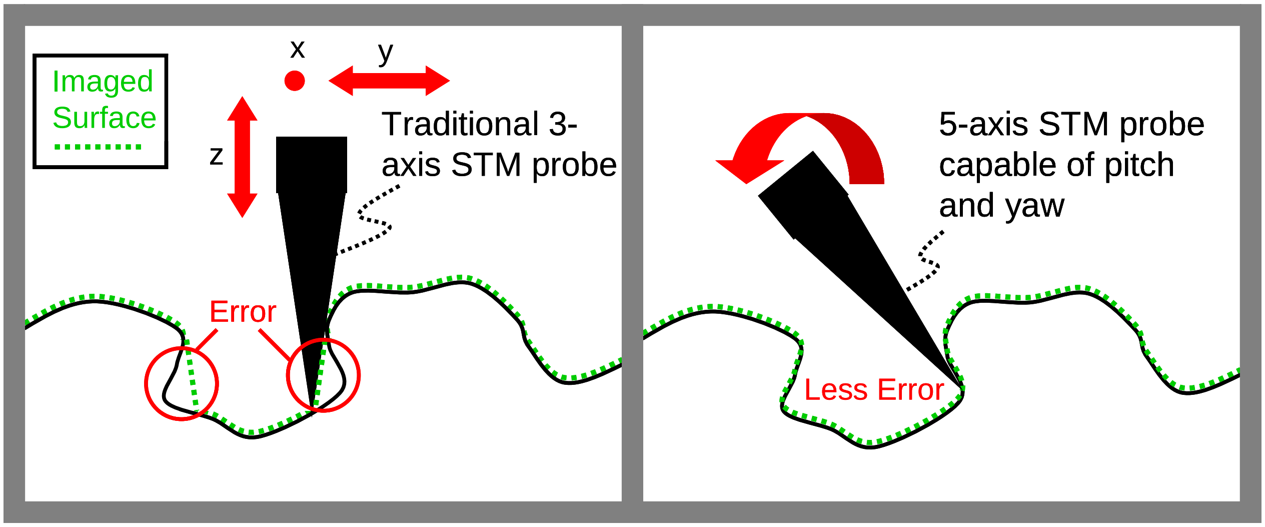

Imaged Surface
x
y
z
Traditional 3-axis STM probe
Error
5-axis STM probe capable of pitch and yaw
Less Error

Figure 10.9: Image quality is improved when an STM capable of five axis scanning is implemented

Since the piezo actuator on which the sample rests successfully achieves the three necessary translational degrees of freedom in traditional STMs, the two extra degrees of freedom will be added to the STM's probe for the design of the new five axis STM. In what follows, therefore, a compliant probe capable of pitch and yaw will be designed using FACT.

Step 1 requires the designer to first design the probe's stage geometry. The probe holder stage shown in **Figure 10.10** is the stage design that will be used for this example.

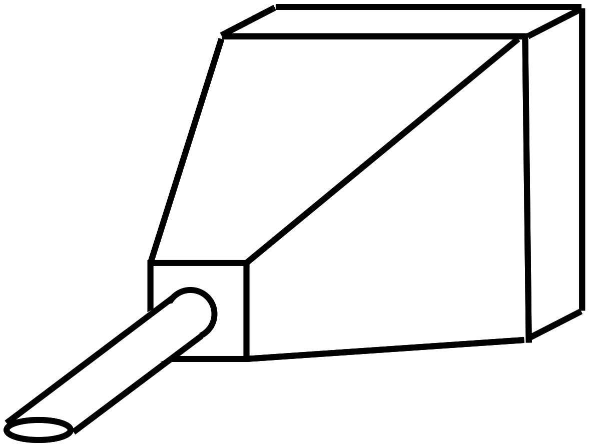


Figure 10.10: Stage geometry of the compliant STM probe designed from Step 1

Step 2 requires the designer to now specify the desired degrees of freedom the stage is to move with. As indicated earlier, pitch and yaw are the system's desired degrees of freedom. These degrees of freedom are independent orthogonal intersecting pure rotations that yield a disk of pure rotations as shown in **Figure 10.11**. This disk of pure rotations allows the tip of the probe to sweep out the inside of a spherical dimple in the surface of a sample being scanned.

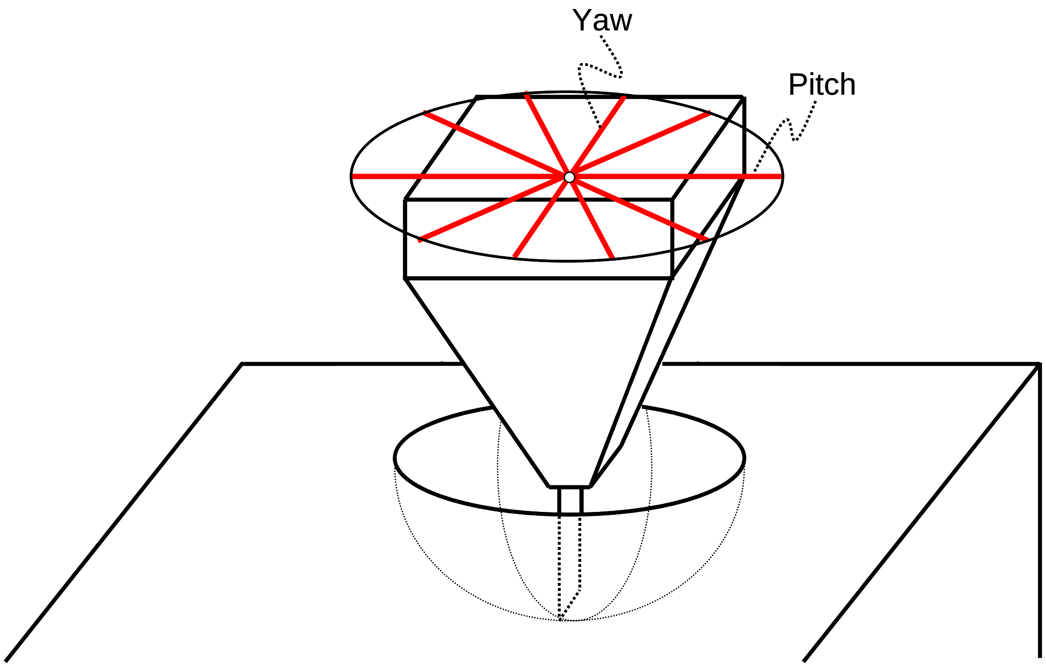


Figure 10.11: Desired degrees of freedom specified from Step 2 include the pitch and yaw of the STM probe such that its tip may sweep out a spherical dimple in the sample's surface

Step 3 requires the designer to select the most appropriate freedom and constraint space pair for this system. Since it consists of two independent pure rotational twists, this system must belong to Case 4. The most appropriate type within this case is Case 4, Type 1 since its freedom space consists of a disk of pure rotations. This freedom and constraint space pair is shown in **Figure 10.12**.

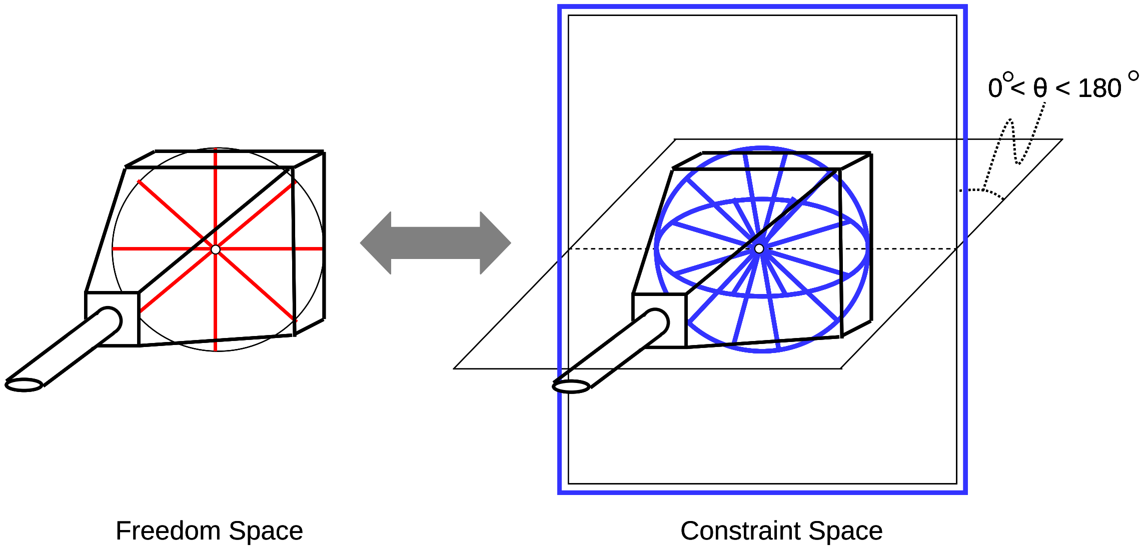


Figure 10.12: Case 4, Type 1 selected for Step 3 given the desired degrees of freedom from Step 2

Step 4 requires the designer to now choose a sub-constraint space for the system. Recall from **Section 8.3.1** in **Chapter 8** that Case 4, Type 1 has four sub-constraint spaces. They are shown in **Figure 10.13** without their instructions for choosing the non-redundant constraints due to the fact that there is not room for instructions in the figure. These instructions are, however, given in **Figure 8.22**, **Figure 8.23**, **Figure 8.24**, and **Figure 8.25** from **Chapter 8**.

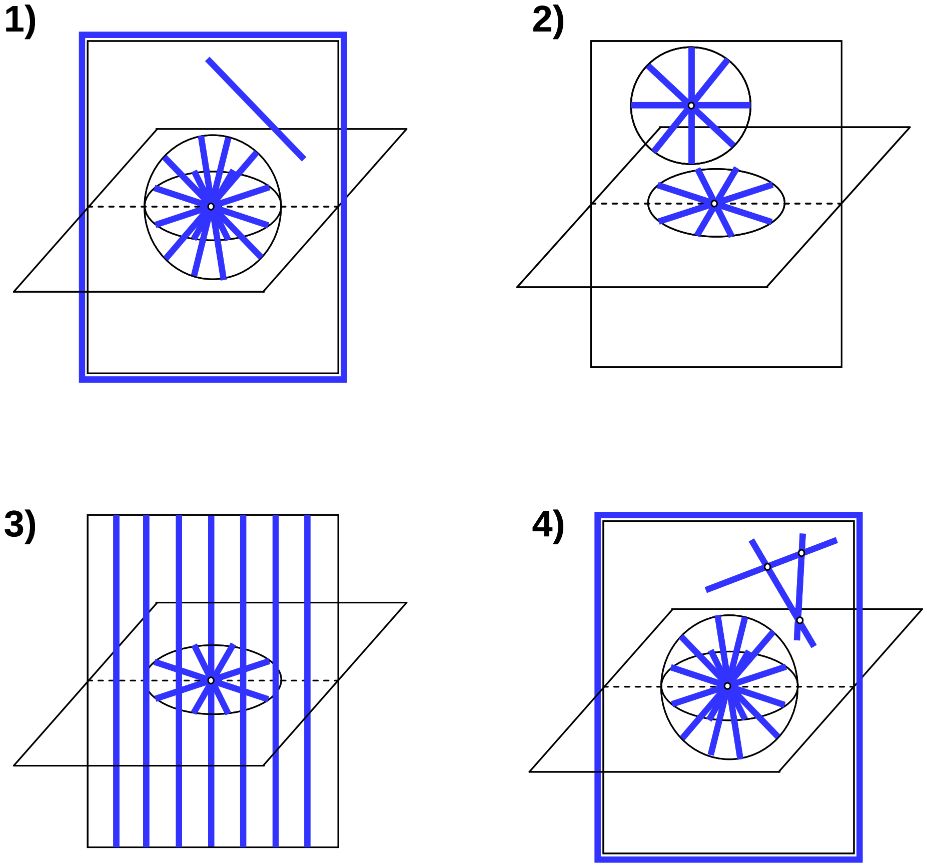


Figure 10.13: Four sub-constraint spaces of Case 4, Type 1. The third one will be chosen for Step 4.

Although any of the four sub-constraint spaces within Case 4, Type 1 could produce functioning constraint topologies, the third sub-constraint space shown in **Figure 10.13** will produce a sufficiently symmetric and robust design with constraints that fit within the geometry of the STM head. This third sub-constraint space will, therefore be selected for Step 4.

Step 5 requires the designer to select four non-redundant constraints from this chosen sub-constraint space. Two of these non-redundant constraints are to be chosen from the planar set of constraint lines. In order to optimize stability, these two constraints must attach to the probe's stage but be as far apart from each other as possible. Two other non-redundant constraints are also to be selected from the disk constraint set. These two constraints will be most stable if they are orthogonal since this orientation also places them as far apart from each other as possible. A feasible solution is, therefore, shown in **Figure 10.14**.

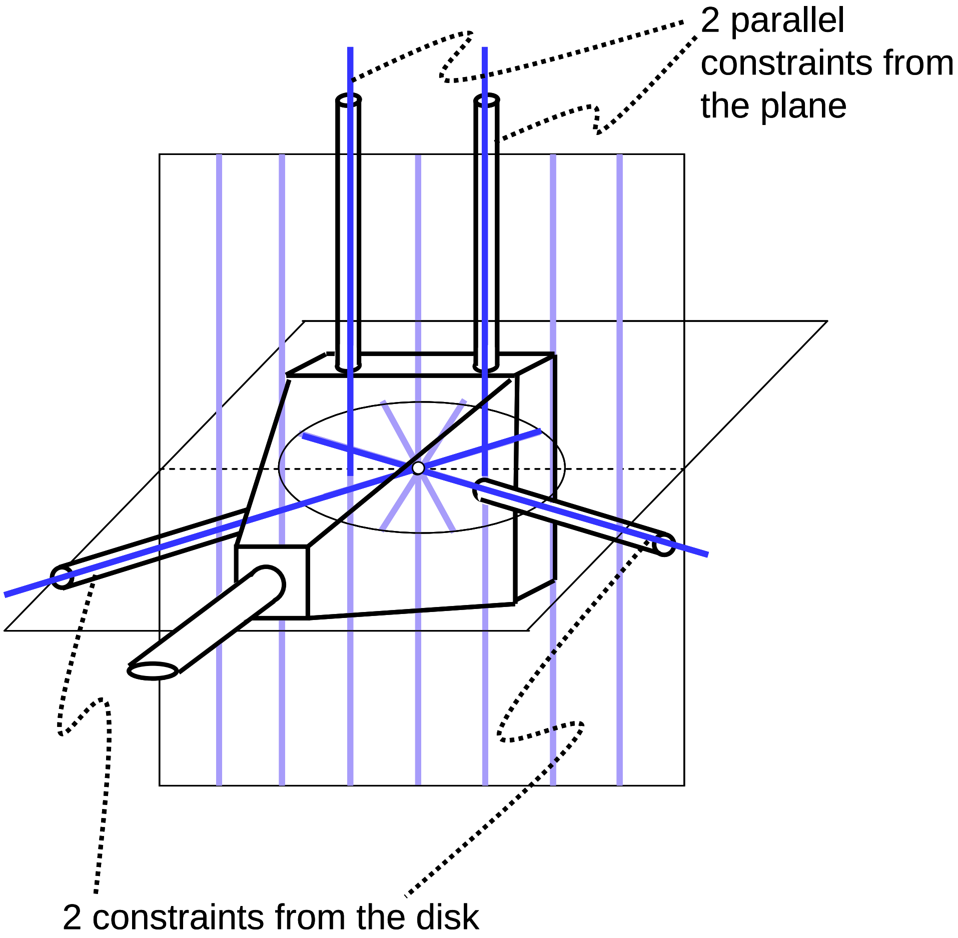

2 parallel constraints from the plane
2 constraints from the disk

Figure 10.14: Selecting non-redundant constraints from Step 5 that are as far apart as possible (dark blue lines are the constraint lines selected)

At this point, a compliant STM probe has been designed that is capable of pitch and yaw degrees of freedom. This probe is shown in **Figure 10.15** with its kinematics shown in red as pure rotational freedom lines.

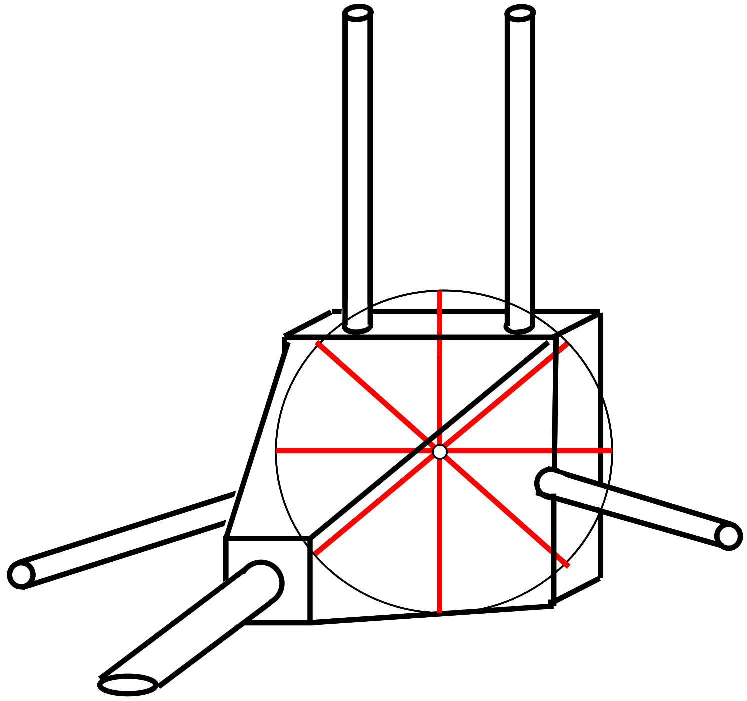


Figure 10.15: Non-redundantly constrained compliant STM probe with its kinematics shown in red

Although the probe designed thus far moves with the desired degrees of freedom from Step 1, it is not yet a symmetric mechanism. If the probe were to experience a fluctuation in temperature, the constraints attached to it would expand and contract moving the probe with undesired and uncontrolled motions. Step 6 will, therefore, be followed by selecting redundant constraints from the system's constraint space to correct for these thermal expansion errors. Four more constraints are chosen from the system's constraint space to add symmetry to the probe as shown in **Figure 10.16**.

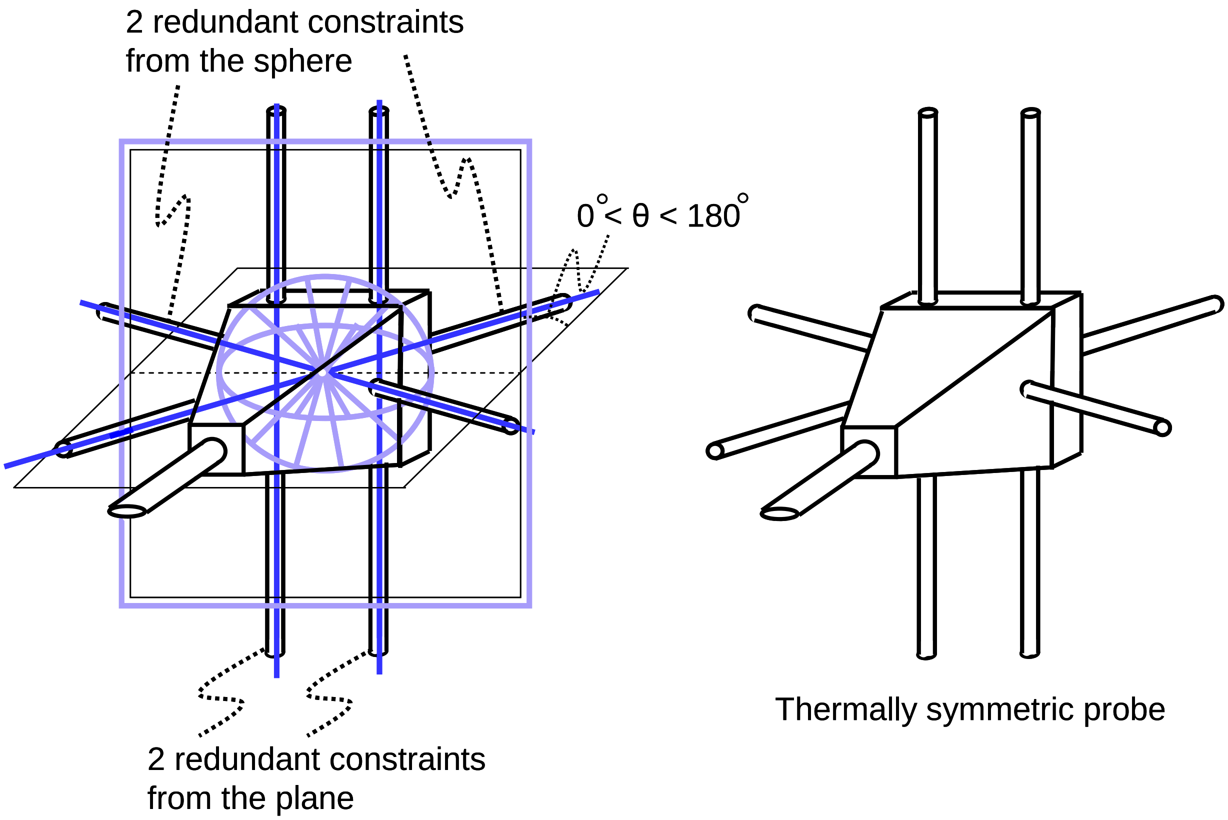


Figure 10.16: Redundant constraints are selected from the system's constraint space to create a thermally symmetric probe for Step 6

A solid model of the final STM probe capable of pitch and yaw is shown in **Figure 10.17**. An exploded view of the STM's parts is also shown in the figure. The outer rectangle is held fixed inside the STM's head while the probe is free to move. Two actuators would need to be attached to the probe's stage to independently actuate its pitch and yaw motions.

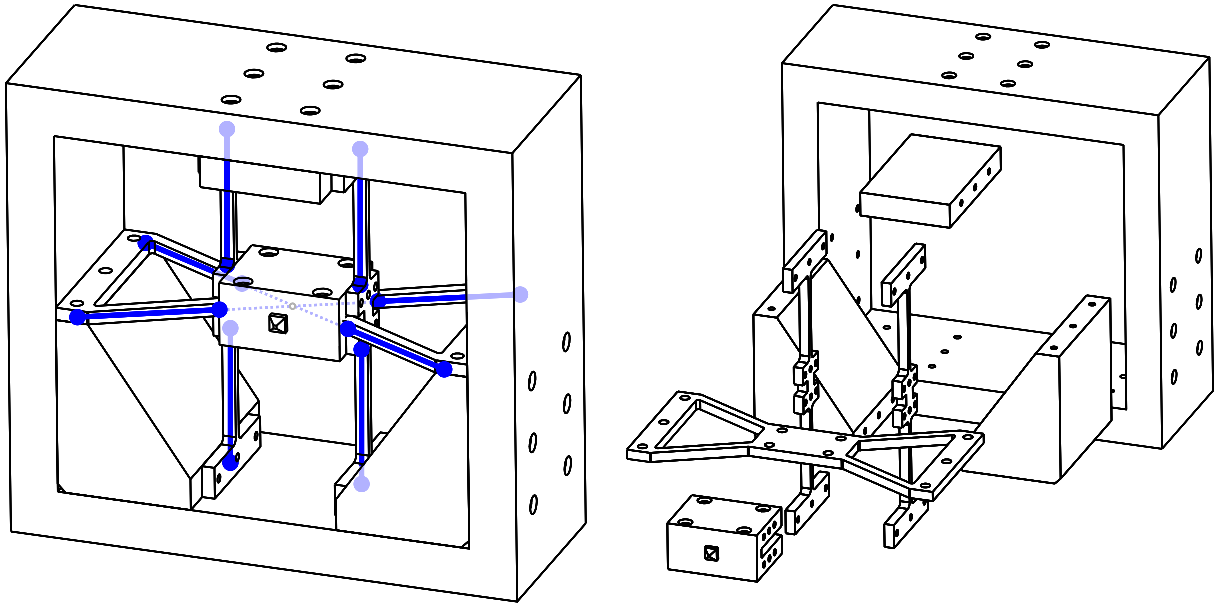


Figure 10.17: Solid model of a compliant probe for a five axis STM

10.2.3 Three-dimensional Compliant Rotary Flexure

This section demonstrates the design of a three-dimensional compliant rotary flexure using the FACT design method. The purpose of this section is to demonstrate FACT for a Case 5 system.

Rotational motion is important to many applications. Some possible applications at the micro- or nano-scale include hinges, levers, drills, or vibration energy harvesters. Compliant rotary flexures are not a new invention. Planar compliant rotary flexures are commonly used for achieving precise rotational motions about an instant center with an axis that is perpendicular to the plane of the flexure. Three-dimensional compliant rotary flexures are more complex and difficult to design. Such flexures may be preferable to planar rotary flexures under certain circumstances due to geometric incompatibilities or space requirements. In this section a three-dimensional rotary flexure will be designed that could not easily be conceived without using the FACT design method.

Step 1 requires the designer to first design the mechanism's stage. The stage of the rotary flexure designed for this example is shown in **Figure 10.18**.

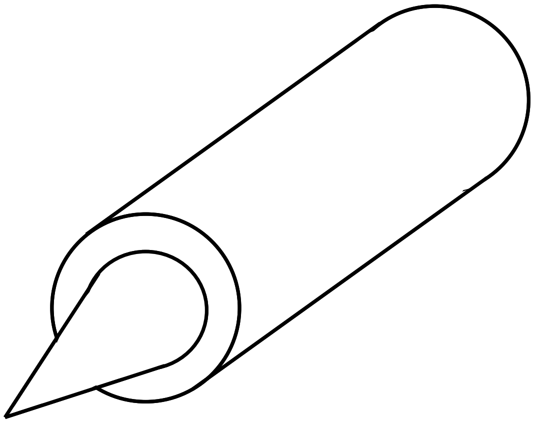


Figure 10.18: Stage of the compliant rotary flexure designed from Step 1

Step 2 requires the designer to now specify the degrees of freedom he/she wishes this stage to move with. As indicated earlier, the objective of the rotary flexure is to rotate about its axis. This degree of freedom may be represented by a single pure rotational freedom line as shown in **Figure 10.19**.

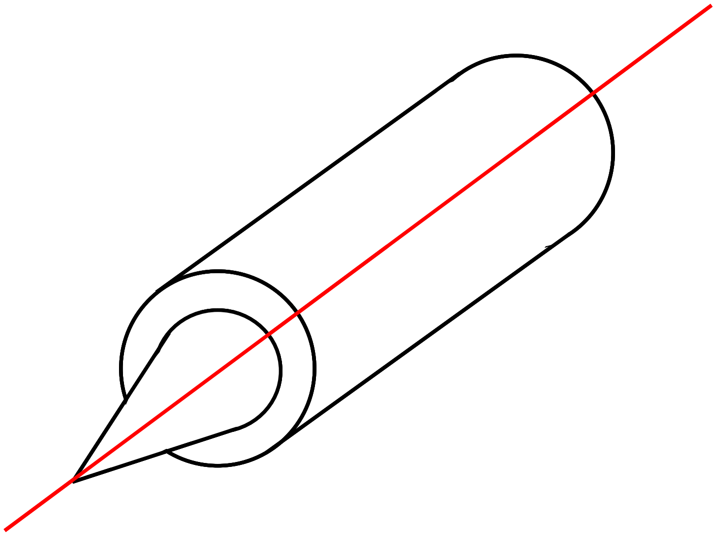


Figure 10.19: Desired rotational degree of freedom specified from Step 2.

Step 3 requires the designer to select the most appropriate freedom and constraint space pair for this system. Since it consists of a single pure rotational twist, this system must belong to Case 5. The most appropriate type within this case is Case 5, Type 1. This freedom and constraint space pair is shown in **Figure 10.20**.

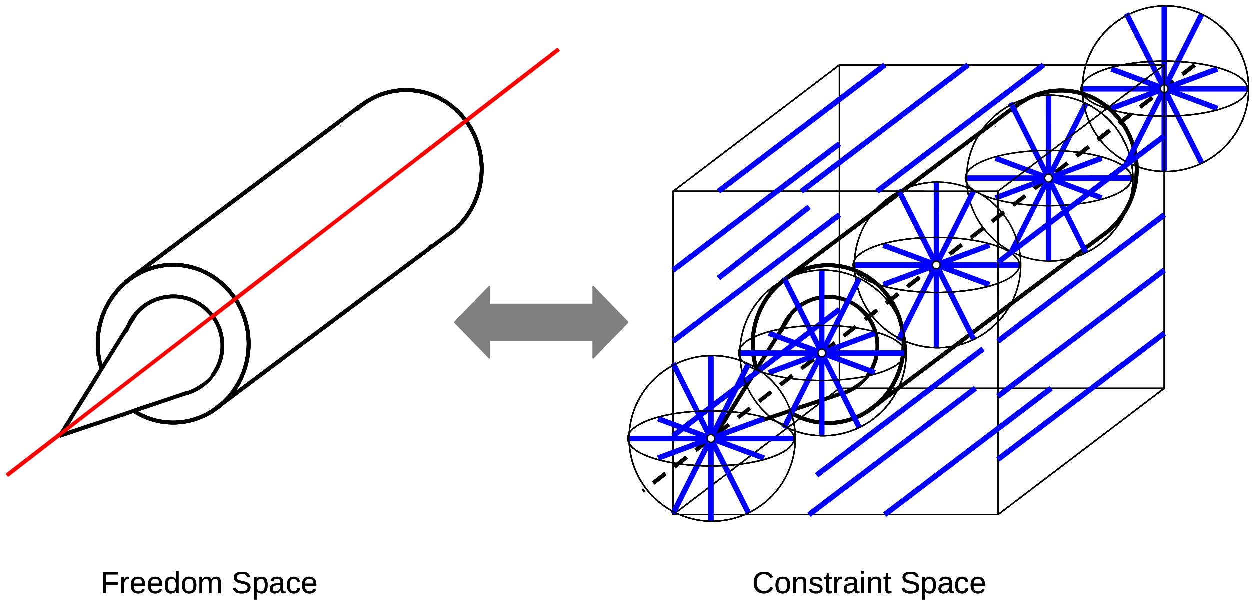


Figure 10.20: Case 5, Type 1 selected for Step 3 given the desired degrees of freedom from Step 2

Step 4 requires the designer to now choose a sub-constraint space for the system. To achieve this task one must first select a constraint space from Case 4 with a unique freedom space that contains a pure rotational freedom line. Suppose one chooses Case 4, Type 7 for this purpose. These spaces are shown in **Figure 10.21**. Note also that the pure rotational freedom line (red) within the freedom space of Case 4, Type 7 is coincident with the pure rotational freedom line within the freedom space of Case 5, Type 1.

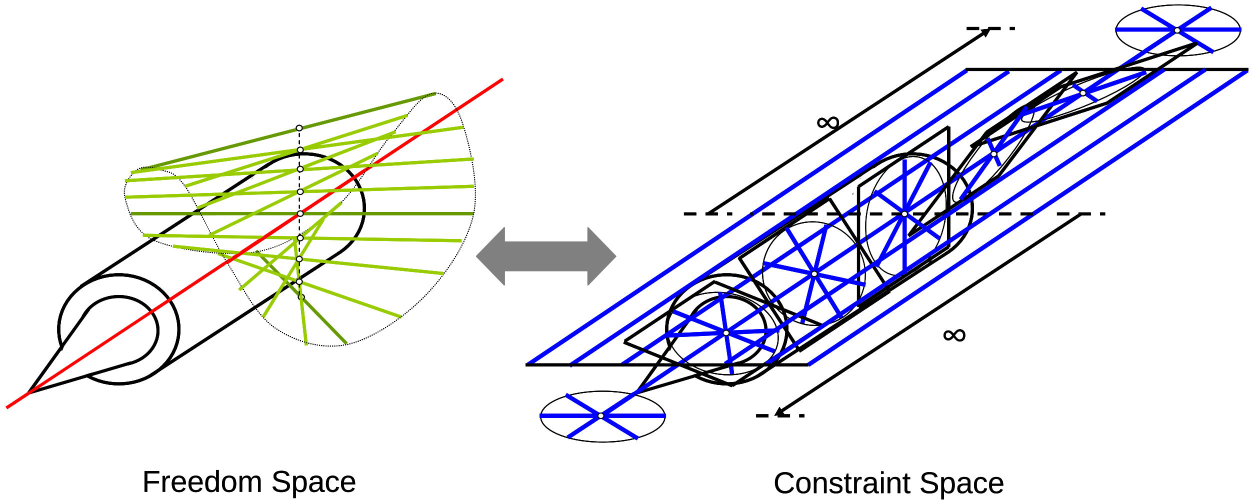


Figure 10.21: Case 4, Type 7 selected within Step 4 for finding a sub-constraint space for the system.

Now that the constraint space of Case 4, Type 7 has been selected, recall from **Section 8.3.7** in **Chapter 8** that this constraint space has five sub-constraint spaces. These sub-constraint spaces are shown in **Figure 10.22** without their instructions for choosing the non-redundant constraints due to the fact that there is no room for instructions in the figure. These instructions are, however, given in **Figure 8.68**, **Figure 8.69**, **Figure 8.70**, **Figure 8.71**, and **Figure 8.72** from **Chapter 8**.

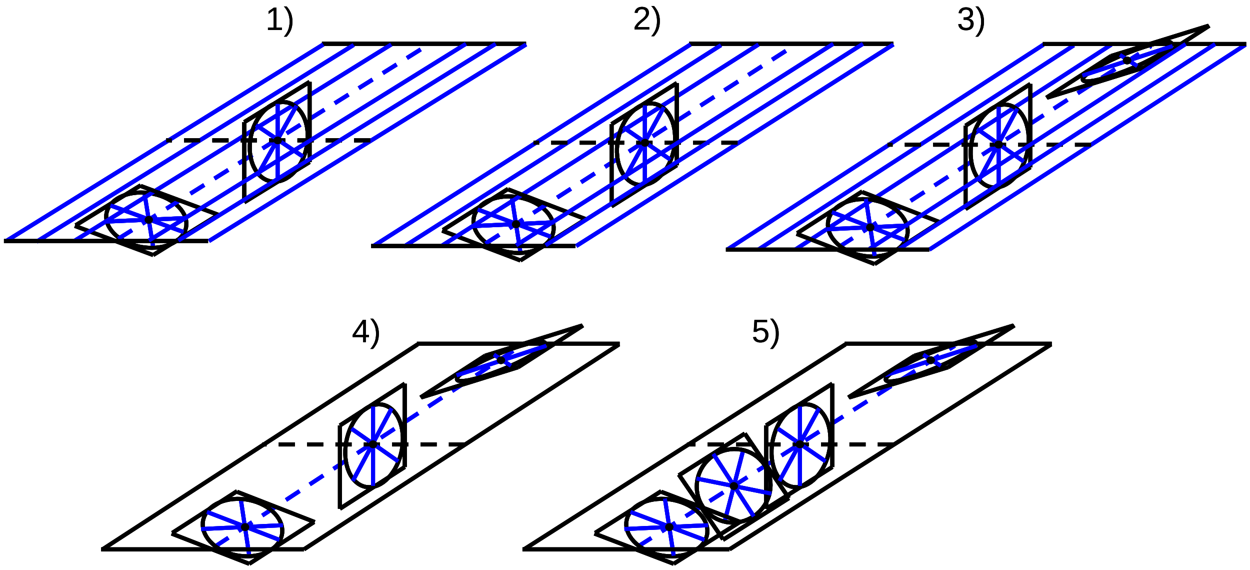


Figure 10.22: Five sub-constraint spaces of Case 4, Type 7. The first one will be chosen for Step 4.

In order to complete Step 4, one of these sub-constraint spaces must be chosen. The first sub-constraint space is selected for this example.

Step 5 requires the designer to select non-redundant constraints from the chosen sub-constraint spaces. The first sub-constraint space within Case 4, Type 7 selected from Step 4 requires the designer to select four non-redundant constraints. Two of these non-redundant constraints should be selected from within one of the disks within the system's constraint space. One of the non-redundant constraints should be selected from another disk within the system's constraint space as long as it is not the axis of the disks (dashed blue). The other non-redundant constraint should be selected from the plane of parallel constraint lines as long as it is also not the axis of the disks. These instructions are followed to select the first four non-redundant constraints as shown in **Figure 10.23**.

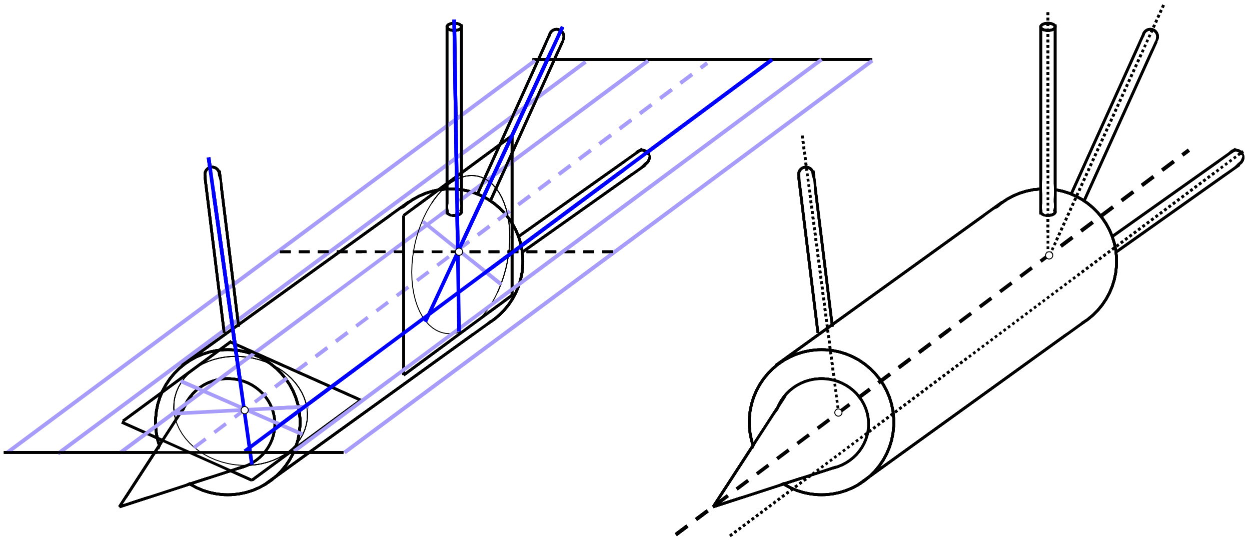


Figure 10.23: Selecting four non-redundant constraints from the first sub-constraint space of Case 4, Type 7 for Step 5 (dark blue lines are the constraint lines selected)

Step 5, however, has not yet been completed. Only four of five non-redundant constraints have been selected and, therefore, the rotary flexure is expected to move with two independent twists instead of only one. This fact is already known because the kinematics of the rotary flexure thus far is described by the freedom space of Case 4, Type 7 as shown in **Figure 10.24**.

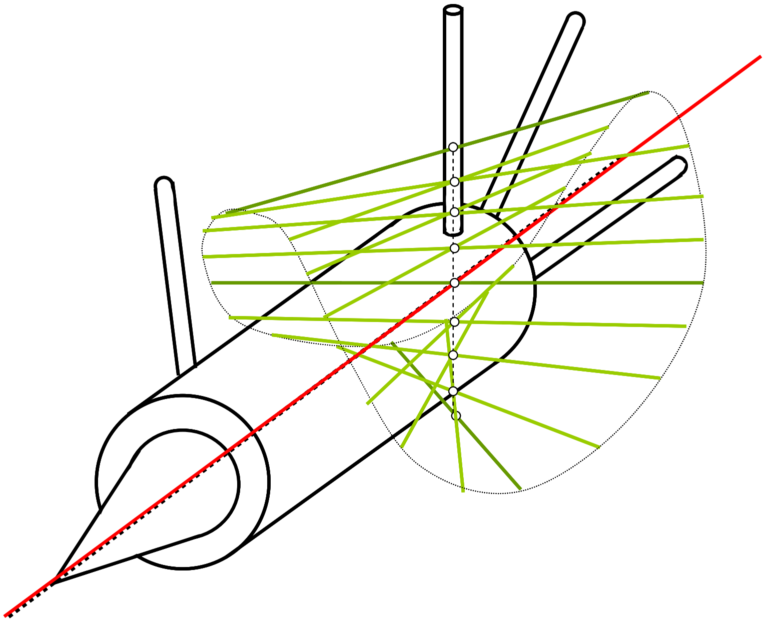


Figure 10.24: The kinematics of the current design with four non-redundant constraints is described by the freedom space of Case 4, Type 7.

In order to properly select the fifth and final non-redundant constraint, one must first recognize that since the freedom space of Case 5, Type 1 lies within the freedom space of Case 4, Type 7, the constraint space of Case 4, Type 7 must lie within the constraint space of Case 5, Type 1 as shown in **Figure 10.25**. Note that each disk within the constraint space of Case 4, Type 7 lies within a single corresponding sphere within the constraint space of Case 5, Type 1. Note also that the plane of parallel lines within the constraint space of Case 4, Type 7 lies within the box of parallel lines within the constraint space of Case 5, Type 1.

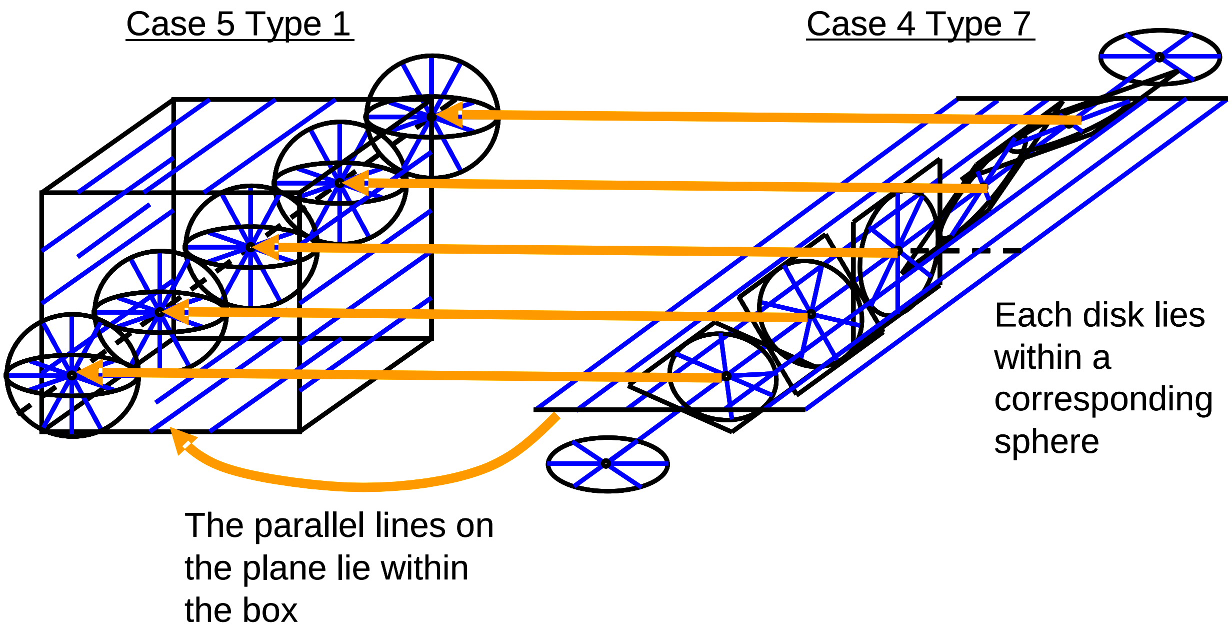

Case 5 Type 1
Case 4 Type 7
Each disk lies within a corresponding sphere
The parallel lines on the plane lie within the box

Figure 10.25: The constraint space of Case 4, Type 7 lies within the constraint space of Case 5, Type 1

If the fifth constraint were to be selected from within the constraint space of Case 4, Type 7, it would, by definition, be redundant. Furthermore, if the fifth constraint is selected such that it lies anywhere outside the constraint space of Case 4, Type 7, it will be non-redundant. But in order for this non-redundant constraint to complement the freedom space of Case 5, Type 1, it must lie within the constraint space of Case 5, Type 1 as well. The conclusion is, therefore, drawn that the fifth non-redundant constraint must lie within the constraint space of Case 5, Type 1 but must not lie within the constraint space of Case 4, Type 7. The fifth non-redundant constraint selected for this example satisfies these conditions and is shown in **Figure 10.26**. Once this constraint is selected, Step 5 is complete and all five non-redundant constraints have been selected.

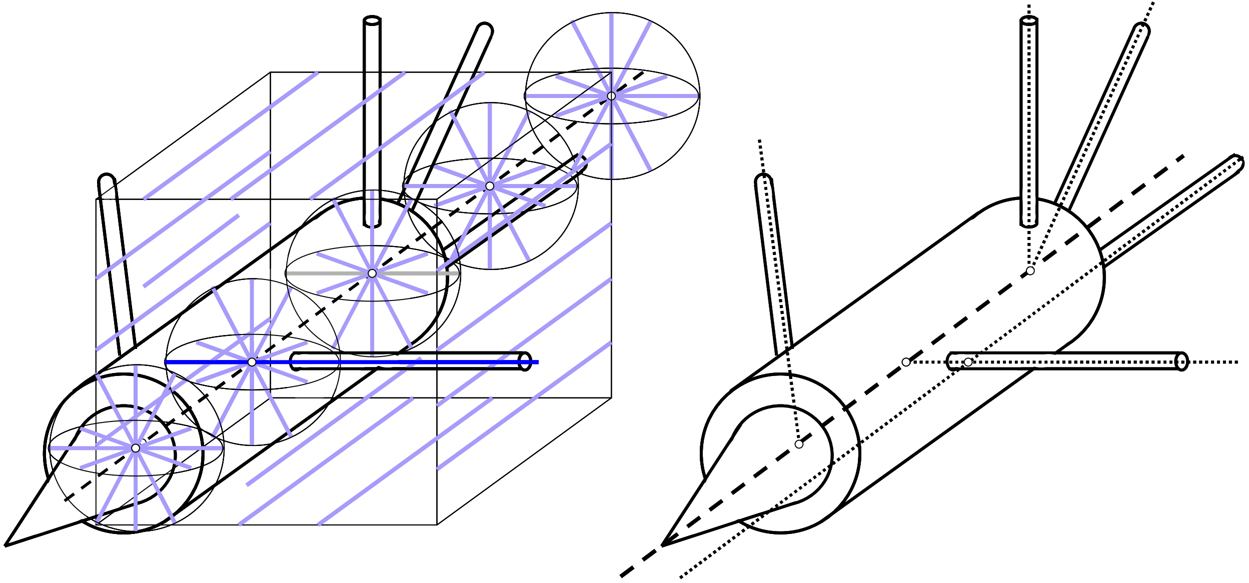


Figure 10.26: Selecting the fifth non-redundant constraint from the constraint space of Case 5, Type 1 that doesn't lie within the constraint space of Case 4, Type 7 to complete Step 5 (dark blue line is the constraint line selected)

At this point, a compliant rotary flexure has successfully been designed with a single rotational degree of freedom. It is shown in **Figure 10.27** with its kinematics shown in red.

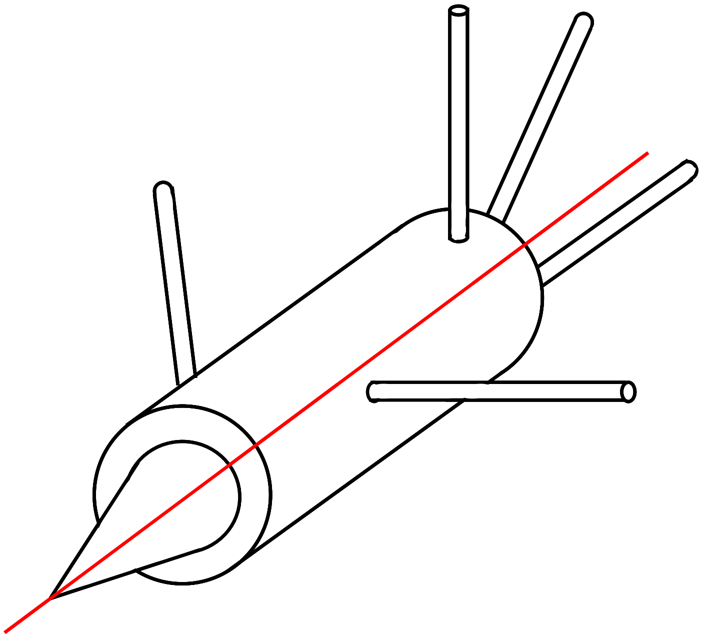


Figure 10.27: Non-redundantly constrained compliant rotary flexure

One could now proceed to Step 6 by selecting redundant constraints from the constraint space of Case 5, Type 1 to add symmetry, stiffness, load capacity, etc, but for this example, the process will stop here. The reason for this is that the objective for this example has already been achieved. The design proposed above is not very practical and would probably never be used as an actual rotary flexure but it was designed this way to help the reader understand how Steps 4 and 5 of the FACT design method work for Case 5 flexure systems, and to show the reader how spaces may efficiently be selected to produce non-intuitive and functioning designs that could not easily be conceived without using FACT.

CHAPTER 11:

“Conclusion”

This chapter reviews the purpose, importance, and impact of this research. This chapter also summarizes the accomplishments of this research and discusses future work for advancing the FACT design method.

11.1 Purpose, Importance and Impact

The purpose of this thesis is to learn how to represent every possible freedom and constraint topology in three dimensions to form a framework called FACT that allows designers, novice or expert, to create any parallel, multi-axis flexure system.

The importance of this research is that flexure system designers can be confident that the final design selected will most optimally achieve the desired design requirements. FACT embodies every possible design solution for parallel flexure systems and thus the designer is able to visualize and consider every possible solution from the beginning of the design process before selecting an optimal design for a specific application.

The impact of FACT is that it improves the design processes for small-scale flexure systems and precision machines that require complex three-dimensional motion. The demand for low-cost, precision machines that are capable of multi-axis motion increases with the advance of micro- and nanotechnologies. These technologies find applications in modern consumer products such as memory storage devices, flat panel TVs, and fiber optic devices. The ability to achieve complex mechanical motions on the nano-scale is also important for helping physicists and scientists understand natural laws on that scale.

11.2 Accomplishments

This section reviews some of the major accomplishments of this thesis:

(1) An effective method for visually representing constraints and degrees of freedom in three-space has been developed. The relationship between a system's degrees of freedom and its complementary constraints has also qualitatively and quantitatively been described using Douglass Blanding's Rule of Complementary Patterns, screw theory, and projective geometry. Screw theory was implemented in order to understand and describe coupled degrees-of-freedom and projective geometry proved useful in visually representing pure translations. Before the creation of FACT, flexure system designers were not able to easily design flexures that were capable of moving with coupled motions.

(2) From this research it was found that the complete freedom and constraint topologies of any system may be visually displayed as ruled surfaces and volumes that contain an infinite number of freedom and constraint lines. These spaces are called freedom and constraint spaces. Every system will have a unique pair of freedom and constraint spaces. Common surfaces in mathematics such as the hyperbolic paraboloid, the hyperboloid, and the cylindroid were found to be fundamental building blocks of many freedom and constraint spaces. Furthermore, these spaces have been mathematically described and parameterized. These spaces allow designers to rapidly identify non-intuitive flexure system design concepts.

(3) A qualitative and quantitative understanding of the difference between redundant and non-redundant constraints has also been developed and applied to the creation of spaces within a system's constraint space that directs designers in selecting appropriate non-redundant constraints. These spaces are called sub-constraint spaces. Using these spaces designers can satisfy stiffness and symmetry design requirements without altering the mechanism's motions. Before the creation of FACT, constraint redundancy was not well understood and thus could not be implemented to improve flexure system design.

(4) From this research it was found that a finite number of ways exist to visually represent all possible solutions for every parallel flexure system in three-space. More specifically, for flexure

systems there are 26 constraint and freedom space pairs within 6 cases where the number of each case corresponds to the number of non-redundant constraints within the system. A system is fixed when it contains 6 non-redundant constraints. The 26 pairs or types were displayed in **Chapter 9** and are shown again here in **Figure 11.1**. Note the symmetry within the cases. Cases 1 and 6 have one pair of spaces, Cases 2 and 5 have three pairs of spaces, Cases 3 and 4 have nine pairs of spaces. Before the creation of FACT, designers did not understand that every possible parallel flexure system exists within a finite number of spaces. Designers incorrectly assumed that an infinite number of ways exist for constraining a stage. Through this research all the ways a stage can be constrained have been discovered and organized into 26 spaces.

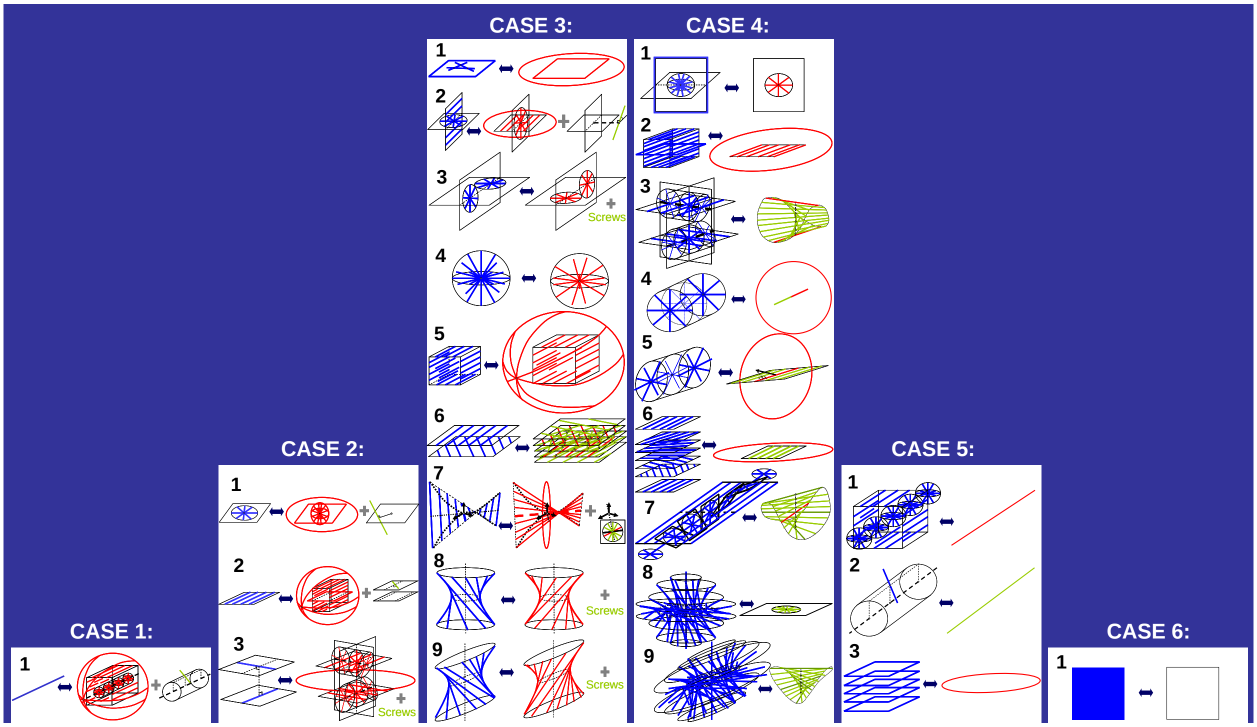


Figure 11.1: Every case and type for all flexure systems

(5) The FACT design method for the synthesis of parallel flexure systems has been created. This method consists of 6 steps. FACT is capable of designing every possible parallel flexure system capable of any possible motions in three-space. The steps include:

Step 1: Design stage geometry
Step 2: Specify desired motions
Step 3: Select best freedom and constraint space
Step 4: Select sub-constraint space
Step 5: Select non-redundant constraints
Step 6: Select redundant constraints

11.3 Future Work

This section describes future research efforts to further FACT.

(1) The work done thus far contains the kinematic solutions for all possible flexure systems. The elasto-mechanics of systems has, however, not been addressed in this thesis. More work needs to be done in guiding the designer in selecting appropriate constraints from sub-constraint spaces and in determining their optimal lengths and thicknesses to achieve desired system stiffnesses. The dynamics of these flexure systems including modal and vibration analysis must also be investigated and integrated into the FACT design tool as well as steps to avoid constraints from buckling.

(2) Constraints with alternate geometries such as coils could potentially be modeled as wrenches with non-zero *q* values. If this were true, the designer would have access to more freedom and constraint space pairs. More research would, therefore, need to be done to determine how many types would exist within Case 3 and to add the extra types that would also exist within Cases 1 and 2 for systems containing wrenches of all real *q* values. The existing constraint spaces found in this thesis would also need to include these new constraints.

(3) Once the flexure system's stage has been constrained such that it may only move with the desired degrees of freedom, actuators must be attached to it to actuate these degrees of freedom. Certain actuator locations and orientations are better than others for controlling these motions. Research should be done to guide the designer in optimally placing and attaching these actuators.

(4) So far FACT is only capable of designing flexure systems that consist of a single rigid stage constrained by long slender beams grounded at one end. Research could be done, however, to extend these principles to the design of such mechanisms stacked in series or in parallel. This concept is shown in **Figure 11.2**.

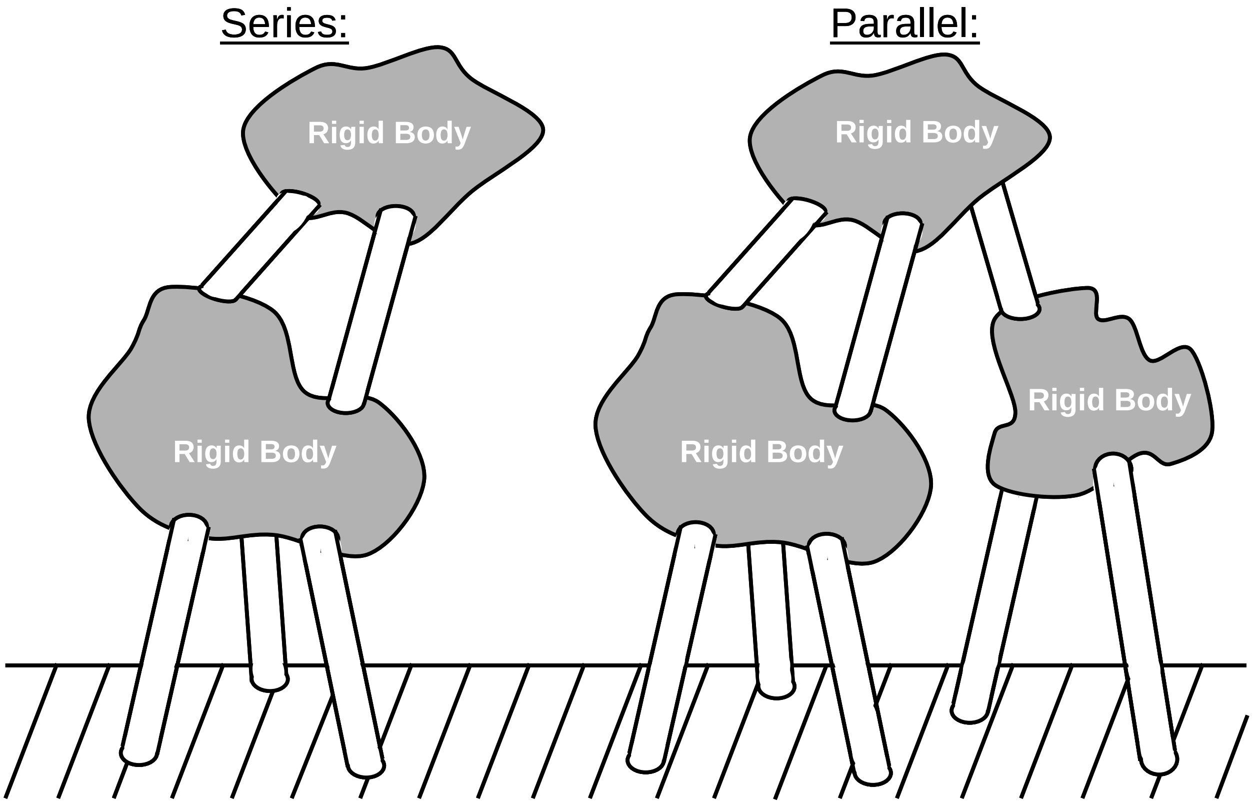


Figure 11.2: Parallel flexure systems stacked in series and in parallel

(5) The FACT design method must be integrated into a virtual reality design tool that allows the designer to view and select constraints within a three dimensional "user friendly" environment.

The completion of these tasks will make FACT an even more powerful design tool for the synthesis of precision flexure systems.

APPENDIX A:

“Proof of Twist and Wrench Relationship”

This appendix explains how **Equation (3.12)** is simplified to **Equation (3.13)** from **Chapter 3**.

Equation (3.12) may be simplified to

$$(\vec{c}-\vec{r})\bullet(\vec{w}\times\vec{f})+(p+q)(\vec{f}\bullet\vec{w})=0\,. \tag{A.1}$$

Recalling that q always equals zero for constraints used in flexure systems and referring to the parameters and their relationships shown in **Figure A.1**, **Equation (A.1)** simplifies to

$$|(\vec{c}-\vec{r})|\,|(\vec{w}\times\vec{f})|\cos\phi+p\,|\vec{f}|\,|\vec{w}|\cos\theta=0\,. \tag{A.2}$$

If the shortest distance line, d, is considered positive moving from the wrench to the twist, the skew angle, θ, will be positive as defined in **Figure A.1** according to the right-hand rule. From **Figure A.1** note that

$$|(\vec{c}-\vec{r})|\cos\phi=-d\,. \tag{A.3}$$

If **Equation (A.3)** is plugged into **Equation (A.2)** and the magnitude of the cross product of the $\vec{w}$ and $\vec{f}$ vectors are solved for, **Equation (A.2)** is simplified to

$$-d\,|\vec{w}|\,|\vec{f}|\sin\theta+p\,|\vec{\mathrm{w}}|\,|\vec{f}|\cos\theta=0\,, \tag{A.4}$$

which may further be simplified to **Equation (3.13)**.

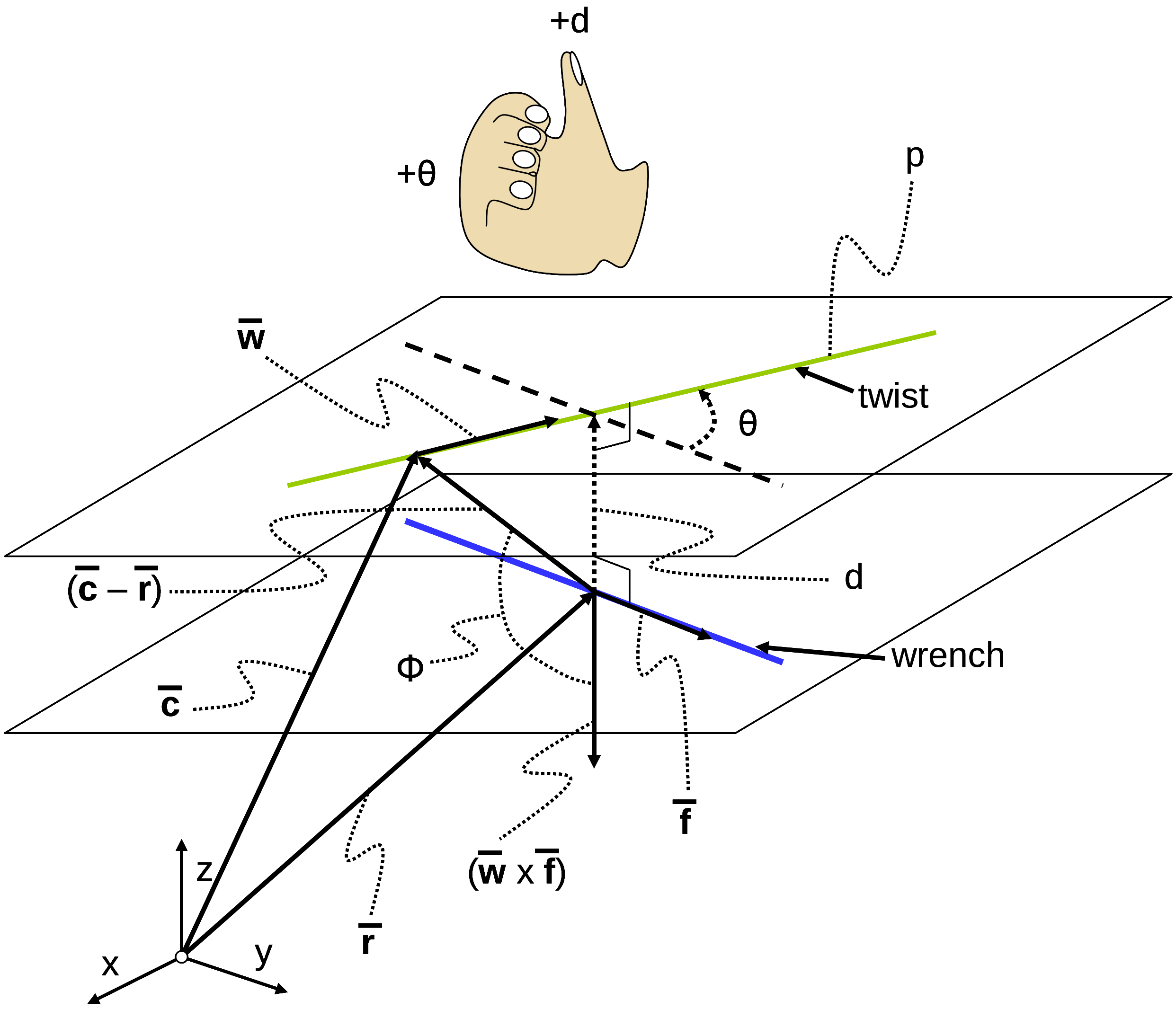


Figure A.1: The parameters and relationships needed to simplify Equation (A.1)

APPENDIX B:

"A Second Pitch Equation"

Equation (3.13) relates the pitch of a twist to the shortest distance between that twist and the wrench as well as the skew angle between these lines. This appendix finds another equation for the pitch of a twist in terms of that twist's orientation and location vectors as well as the wrench's orientation and location vectors. This second pitch equation is used in **Chapter 7** to prove **Equation (7.1)**.

Using **Figure B.1**, the shortest distance, *d*, between an arbitrary twist line and an arbitrary wrench line may be expressed as

$$d = \frac{\left(\vec{f} \times \vec{w}\right)}{\left|\left(\vec{f} \times \vec{w}\right)\right|} \bullet \left(\vec{c} - \vec{r}\right). \qquad \text{(B.1)}$$

Using the definition of a dot product and applying the parameters shown in **Figure B.1** note also that

$$\cos\theta = \frac{\vec{f} \bullet \vec{w}}{\left|\vec{f}\right|\left|\vec{w}\right|}. \qquad \text{(B.2)}$$

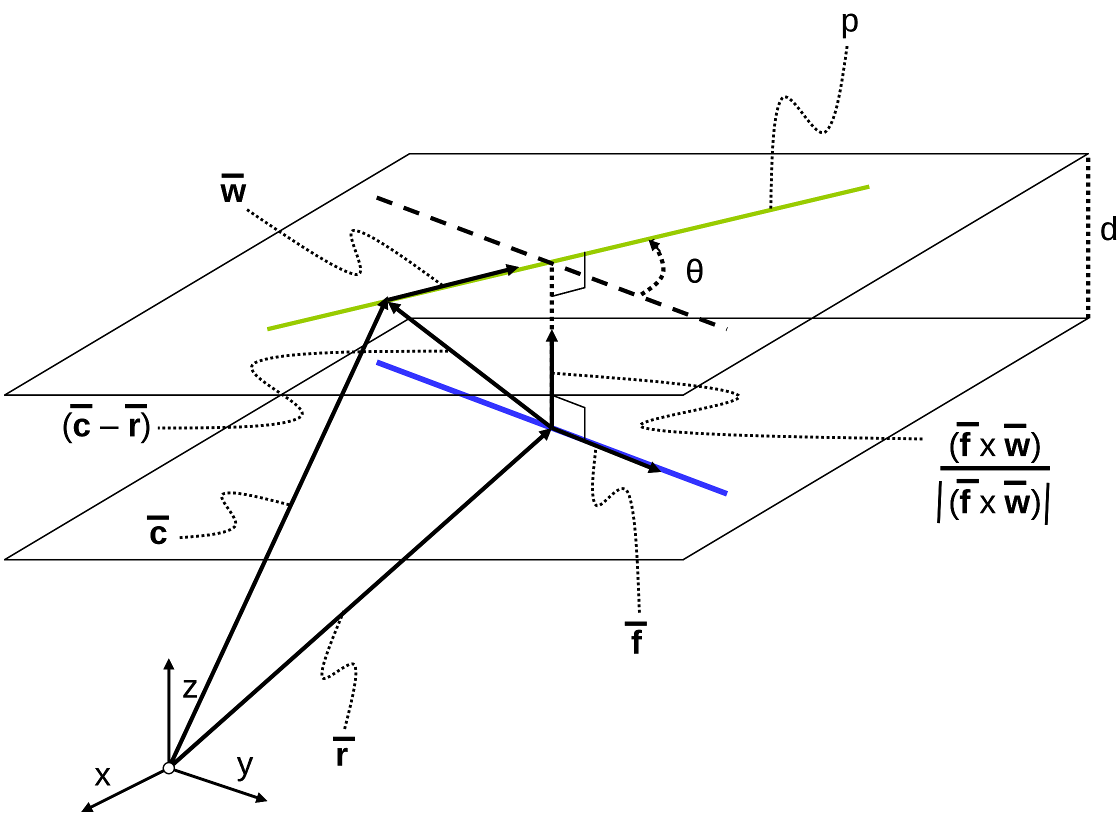


Figure B.1: Arbitrary twist and wrench lines with the parameters necessary for finding *d* and cos*θ* in terms of their orientation and location vectors.

The tangent of the skew angle, *θ*, can now be found by applying **Equation (B.2)** to construct the triangle shown in **Figure B.2**. From this triangle, it is clear that

$$\tan\theta = \frac{\sqrt{1-\left(\dfrac{\vec{f}\bullet\vec{w}}{\left|\vec{f}\right|*\left|\vec{w}\right|}\right)^2}}{\left(\dfrac{\bar{f}\bullet\vec{w}}{\left|\vec{f}\right|*\left|\vec{w}\right|}\right)} \tag{B.3}$$

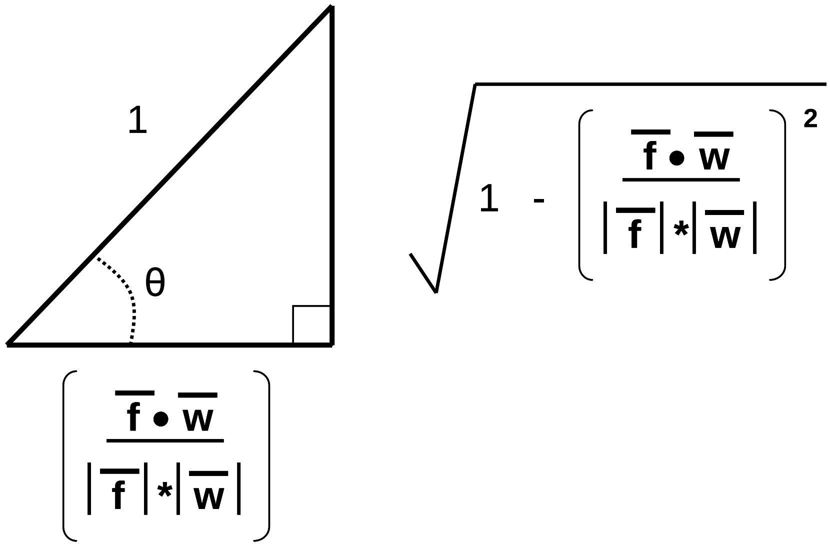

1
θ
$\sqrt{1 - \left(\frac{\bar{f} \bullet \bar{w}}{|\bar{f}|*|\bar{w}|}\right)^{2}}$
$\left(\frac{\bar{f} \bullet \bar{w}}{|\bar{f}|*|\bar{w}|}\right)$

Figure B.2: Triangle constructed from Equation (B.2)

Equation (3.13) can now be applied to find the second pitch equation by multiplying **Equation (B.1)** to **Equation (B.3)**. The pitch, *p*, of this arbitrary twist in terms of the orientation and location vectors, $\vec{r}$, $\vec{f}$, $\vec{c}$, and $\vec{w}$ is, therefore,

$$p = \left(\frac{\left(\vec{f} \times \vec{w}\right)}{\left|\left(\vec{f} \times \vec{w}\right)\right|} \bullet \left(\vec{c} - \vec{r}\right) \right) * \frac{\sqrt{1 - \left(\frac{\vec{f} \bullet \vec{w}}{\left|\vec{f}\right| * \left|\vec{w}\right|} \right)^{2}}}{\left(\frac{\vec{f} \bullet \vec{w}}{\left|\vec{f}\right| * \left|\vec{w}\right|} \right)} \tag{B.4}$$

APPENDIX C:

“A Point and Two Skew Lines”

This appendix contains the mathematical proof for locating the single line that intersects a point and two skew lines in three-space. The concepts discussed in this appendix are used in **Chapter 7** for determining complementary lines within ribbon sets.

For any pair of skew lines, every point in three-space will be intersected by a single line that also intersects these two lines. This is true unless the point is on one of the skew lines in which case an infinite disk of lines will both intersect that point and the two skew lines. If the point is not on one of the skew lines but lies on one of their parallel planes, the line that intersects it and the two skew lines (either in finite space or at infinity) will lie on that plane and will be parallel to the other skew line on the opposing parallel plane. If the point lies somewhere above, between, or below the two parallel planes of the two skew lines, there will be a single line that intersects both it and the two skew lines. This line points in a direction that will now be solved for.

To solve for this orientation vector, $\vec{w}$, the point of interest, $\vec{c}$, and the two skew lines must first be mathematically defined. Suppose one wished to define the first skew line using a location vector, $\vec{r}_1$, and an orientation vector, $\vec{f}_1$, and one wished to define the second skew line with a location vector, $\vec{r}_2$, and an orientation vector, $\vec{f}_2$. These vectors are shown in **Figure B.1**. The line that intersects both the point of interest, $\vec{c}$, and the two skew lines lies on a plane with a normal vector, $\vec{n}$, of

$$\vec{n} = \vec{f}_2 \times \left(\vec{c} - \vec{r}_2\right). \tag{C.1}$$

The equation of this plane also shown in **Figure B.1** is

$$n_x x + n_y y + n_z z = h, \tag{C.2}$$

where *h* can be solved for using a point on this plane. Since it is known that $\vec{c}$ is a point on this plane by definition,

$$h = \vec{n} \bullet \vec{c}. \tag{C.3}$$

The parameter *b* is defined to be a scalar value such that the vector given by

$$\vec{r}_1 + \vec{f}_1 b = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \tag{C.4}$$

will be a point that lies on that plane as well. If **Equation (C.3)** and **Equation (C.4)** are plugged into **Equation (C.2)**, *b* can be solved for as

$$b = \frac{(\vec{n} \bullet \vec{c}) - (\vec{n} \bullet \vec{r}_1)}{(\vec{n} \bullet \vec{f}_1)}. \tag{C.5}$$

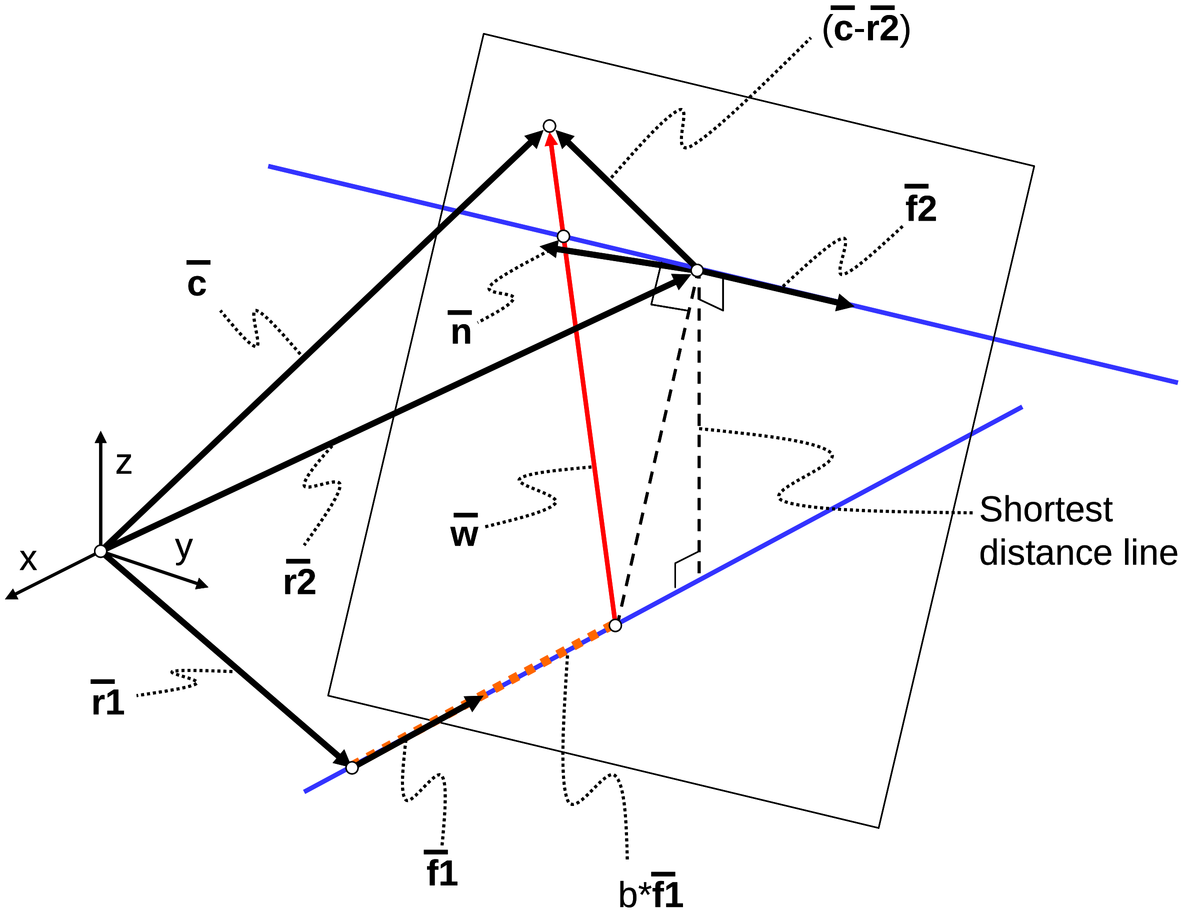


Figure C.1: Two skew lines (blue) with their location and orientation vectors, $\vec{r}$ and $\vec{f}$, and the orientation vector, $\vec{w}$, (red) of the line that intersects both of these skew lines and a point $\vec{c}$.

The unit vector, $\vec{w}$, that points in the direction of the line that passes through point $\vec{c}$ and intersects the two skew lines can now be solved for as

$$\vec{w} = \frac{\vec{c} - \left(\vec{r}_1 + b\vec{f}_1\right)}{\left|\vec{c} - \left(\vec{r}_1 + b\vec{f}_1\right)\right|}. \tag{C.6}$$

Since the location and orientation vectors, $\vec{c}$ and $\vec{w}$, of the line that intersects both the point of interest and the two general skew lines have been found, the desired line of interest has been found.

A function has been written using MATLAB that finds this line's orientation vector given any point and any two skew lines. It is provided below:

```
function [w] = PointSkewLine(r1,f1,r2,f2,pt)
%Input: Two skew lines each with a location vector, r, and
%and an orientation vector, f, and a point, pt.
%Output: Orientation vector, w, of the
%line that intersects both lines and the point.
%A possible location vector of this line is c.
c = pt;
planeNdir = cross(f1,f2); %vector pointing normal to the plane
planeN = planeNdir/sqrt(dot(planeNdir,planeNdir)); %unit vector of this direction
h1 = dot(planeN,r1);
h2 = dot(planeN,r2);
if(dot(cross(f1,(c-r1)),cross(f1,(c-r1))) == 0 || dot(cross(f2,(c-r2)),cross(f2,(c-r2))) == 0)
   disp('Error: Point lies on one of the skew lines')
elseif(dot(planeN,c) == h1)       %if the point is on plane 1:
   wdir = f2;
elseif(dot(planeN,c) == h2)       %if the point is on plane 2:
   wdir = f1;
else                              %point is not one either plane
   ndir = cross(f2,(c-r2));
   n = ndir/sqrt(dot(ndir,ndir));
   h = dot(n,c);
   b = (h-dot(n,r1))/dot(n,f1);
   rint = r1+(b*f1);
   wdir = c-rint;
end
w = wdir/sqrt(dot(wdir,wdir));  %make the w direction a unit vector
```

APPENDIX D:

“Drawing Ribbon Space”

This appendix contains the MATLAB code necessary for generating the ribbon spaces shown in **Figure 7.68**. This code is also capable of generating any other freedom and constraint spaces that contain only three independent skew pure rotations and three independent skew ideal constraints.

The code consists of three functions. One of these functions is given in **Appendix C** and the other two functions are given below. These functions require the user to input three skew constraint lines in the form of three different location vectors, $\vec{r}$, and three different orientation vectors, $\vec{f}$. The user must also specify how many lines should be drawn within each space and how long each line segment should be. The program then finds freedom lines by treating one of the skew constraint lines as a series of points and by applying the principles discussed in **Appendix C**. Three of these freedom lines are then used to find the rest of the constraint lines using the same principles. The lines are plotted point by point.

The two new functions are given as:

Function (1):

```
function PlotLine(r1,f1,len,color)
%Function plots a line
%Input: r1 = location vector of the line
%       f1 = orientation vector of the line
%       len = length of the line
%       if color = 1 the line is blue
%       if color = 2 the line is red
f1 = f1/sqrt(dot(f1,f1));
plot3(r1(1),r1(2),r1(3))
for t = -(len/2):0.05:(len/2);
    hold on;
    line = r1 + f1*t;
    if(color == 1)
        plot3(line(1),line(2),line(3),'b');
    elseif(color == 2)
        plot3(line(1),line(2),line(3),'r');
    end
end
```

Function (2):

```
function SkewFreeConst(r1,f1,r2,f2,r3,f3,range,inc,len)
%Function takes three skew constraints and plots their freedom space (red)
%and constraint space (blue)
%range = determines how far out the lines in the spaces go
%inc = determines how large the increment is between each line
%len = determines how long each line is
%Make the constraint orientation vectors f unit vectors
f1 = f1/sqrt(dot(f1,f1));
f2 = f2/sqrt(dot(f2,f2));
f3 = f3/sqrt(dot(f3,f3));
%Find three pure rotational freedom lines
c1 = r2+f2*-0.1;
w1 = PointSkewLine(r1,f1,r3,f3,c1);
c2 = r2+f2*0;
w2 = PointSkewLine(r1,f1,r3,f3,c2);
c3 = r2+f2*0.1;
w3 = PointSkewLine(r1,f1,r3,f3,c3);
%Make the freedom line orientation vectors w unit vectors
w1 = w1/sqrt(dot(w1,w1));
w2 = w2/sqrt(dot(w2,w2));
w3 = w3/sqrt(dot(w3,w3));
%Plot the Freedom Space
for t = -range:inc:range
    c = r2 + f2*t;
    w = PointSkewLine(r1,f1,r3,f3,c);
    PlotLine(c,w,len,2);
end
%Plot the Constraint Space
for t = -range:inc:range
    r = c2 + w2*t;
    f = PointSkewLine(c1,w1,c3,w3,r);
    PlotLine(r,f,len,1);
end
```

APPENDIX E:

“Complementary Ribbon Spaces”

This appendix explores the relationship between orthogonal freedom ribbon sets and their complementary orthogonal constraint ribbon sets discussed in **Chapter 7**. More specifically, a study is conducted using MATLAB code that determines the pitches of complementary ribbon spaces and how they vary along their respective axes. A constant is determined that fully describes any orthogonal ribbon.

Seven functions were used to conduct this study but only six will be provided at the conclusion of this section. The seventh function is given and discussed in **Appendix C**.

The first two functions that will be discussed are the “RightHandRibbon” and “LeftHandRibbon” functions. Both functions require the input of two skew lines that are skew with respect to each other and with respect to a reference line along the z-axis. These lines are orthogonal to and intersect the y-axis. The function requires the user to input each line’s distance, *d*, away from the origin along the y-axis and each line’s skew angle, *g*, with respect to the reference line. This convention is shown in **Figure E.1**. The “RightHandRibbon” function requires the user to select positive skew angles and appropriate distances along the y-axis that will produce right handed ribbons. The “LeftHandRibbon” function requires the user to select negative skew angles and appropriate distances along the y-axis that will produce left handed ribbons. This convention will always produce orthogonal ribbons with axes that are coincident with the shortest distance line of the three original skew lines. This shortest distance line will always be the y-axis.

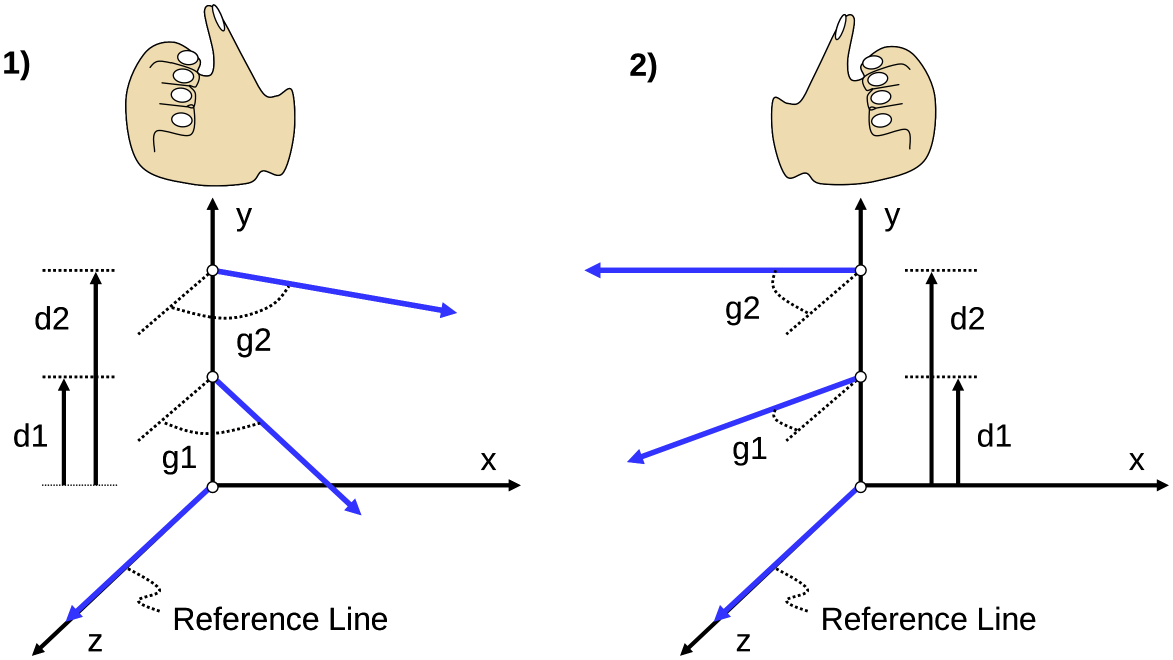


Figure E.1: (1) "RightHandRibbon" function's convention for right-handed ribbon input parameters. (2) "LeftHandRibbon" function's convention for left-handed ribbon input parameters.

Once these three skew lines have been received by either function, the principles of **Appendix C** are used to find two new skew lines that are complementary to the three skew lines by treating one of them as a series of points in the midst of two skew lines. Then the same principles are again applied to individual points along the y-axis using these two new skew lines. The rest of the lines that lie within the right- or left-handed ribbon are located using this approach. The skew angle with respect to the reference line of each of these lines is plotted versus their respective position along the y-axis. The pitch of the resulting ribbon is then plotted along the ribbon's axis by calculating the inverse derivative of this plot. The derivative of this pitch with respect to position along the y-axis is then calculated and plotted versus position along the ribbon's axis. Finally the derivative of this function is calculated with respect to position along the y-axis. This derivative is determined to be a constant value.

Note that the double derivative of every orthogonal ribbon's pitch will always be a constant that is unique to that particular ribbon. This constant will always be a positive value for right-handed ribbons and it will always be a negative value for left-handed ribbons. This constant will be equivalent in magnitude but opposite in sign for complementary ribbons.

This may be shown by using the next two functions. Their names are ComplementaryRight2Left and ComplementaryLeft2Right. These functions have the same input parameters as the functions discussed previously. The ComplementaryRight2Left function receives three skew lines for a right-handed ribbon, calls the "RightHandRibbon" function using these lines and then uses the principles of **Appendix C** to find three new skew lines that are complementary to these original three lines. These new skew lines are then redefined to correspond with the convention shown in **Figure E.1** such that their shortest distance line is coincident with the y-axis. Finally the "LeftHandRibbon" function is called using these three new skew lines to create the complementary left-handed ribbon. The ComplementaryLeft2Right function performs this same procedure by receiving three skew lines that form a left-handed ribbon and then creates its complementary right-handed ribbon.

The final two functions are used to calculate derivatives. Their names are "pitch" and "der_pitch". The code for these functions is provided at the end of this appendix.

The following is an example. Suppose one enters three skew lines that create a right-handed ribbon with a center line that is the reference line along the z-axis by typing "ComplementRight2Left(5,(pi/4),10,((pi/2)-atan(.5)),30)" in the command window of MATLAB (These three lines where chosen so that the plots would be centered at zero). **Figure E.2** contains the skew angles versus position along the ribbon's axis for lines within the right-handed ribbon (top) and its complementary left-handed ribbon (bottom). Note their symmetry.

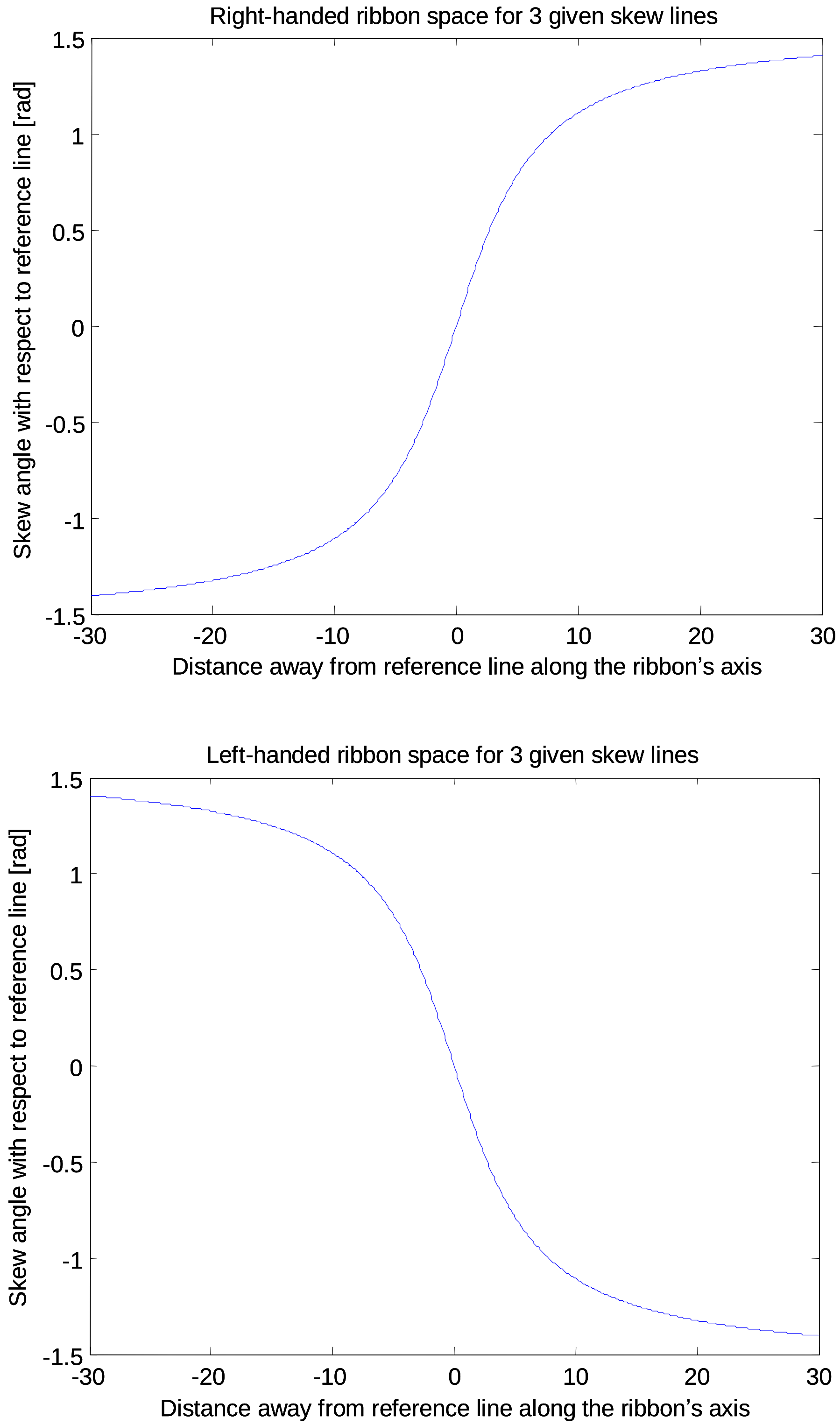
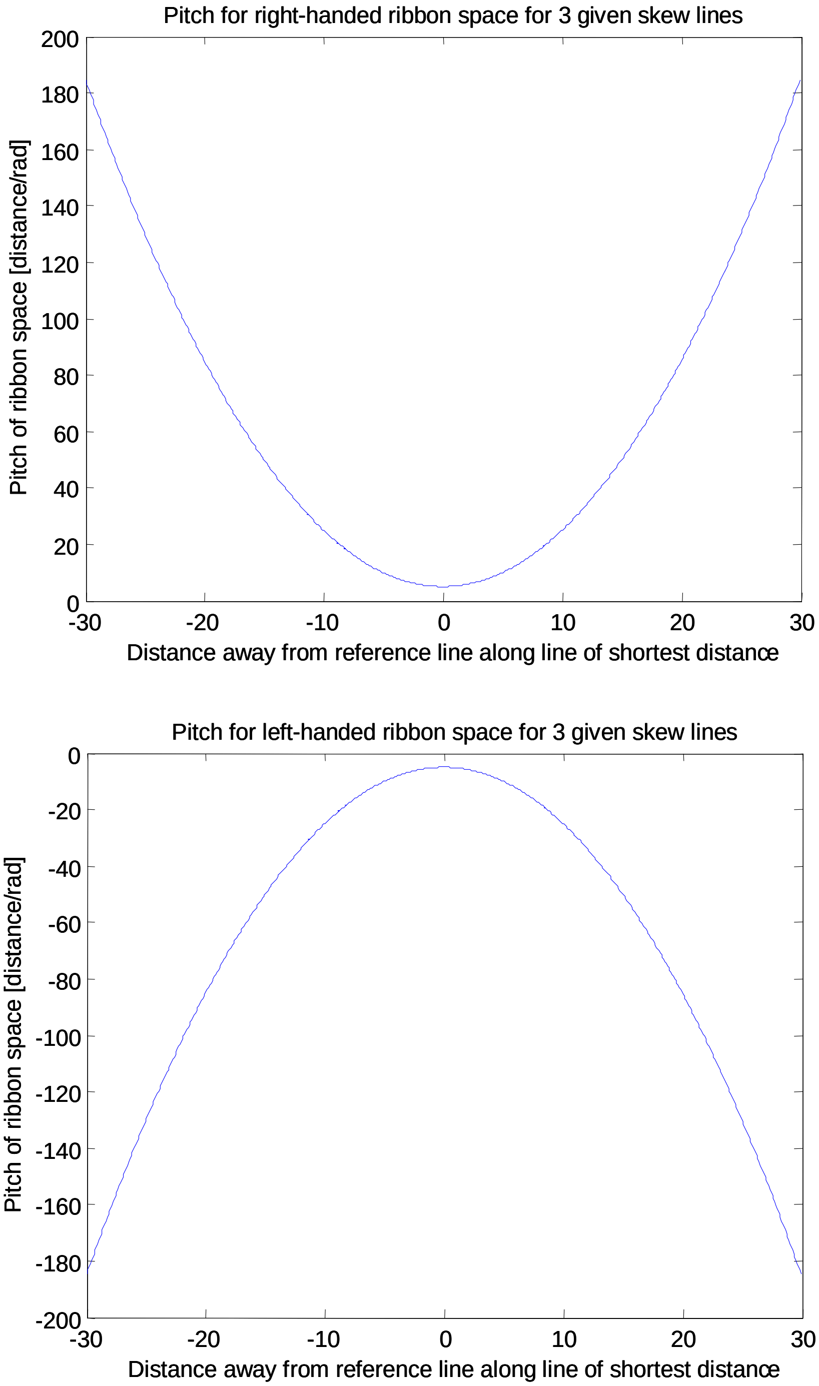


Figure E.2: Complementary ribbon spaces with their lines' skew angles plotted against their position along the y-axis

Each ribbon’s pitch values are plotted versus position along the ribbon’s axis. These plots are given in **Figure E.3**. Note that the complementary ribbons’ pitch values are equal and opposite at corresponding locations along the ribbons’ axes.



Pitch for right-handed ribbon space for 3 given skew lines

Pitch for left-handed ribbon space for 3 given skew lines

Figure E.3: Complementary ribbon spaces with their pitch values plotted against position along the y-axis

Figure E.4 provides the plots of the derivatives of these complementary ribbon pitch plots versus position along the ribbons' axes. Note that both of these plots are linear.

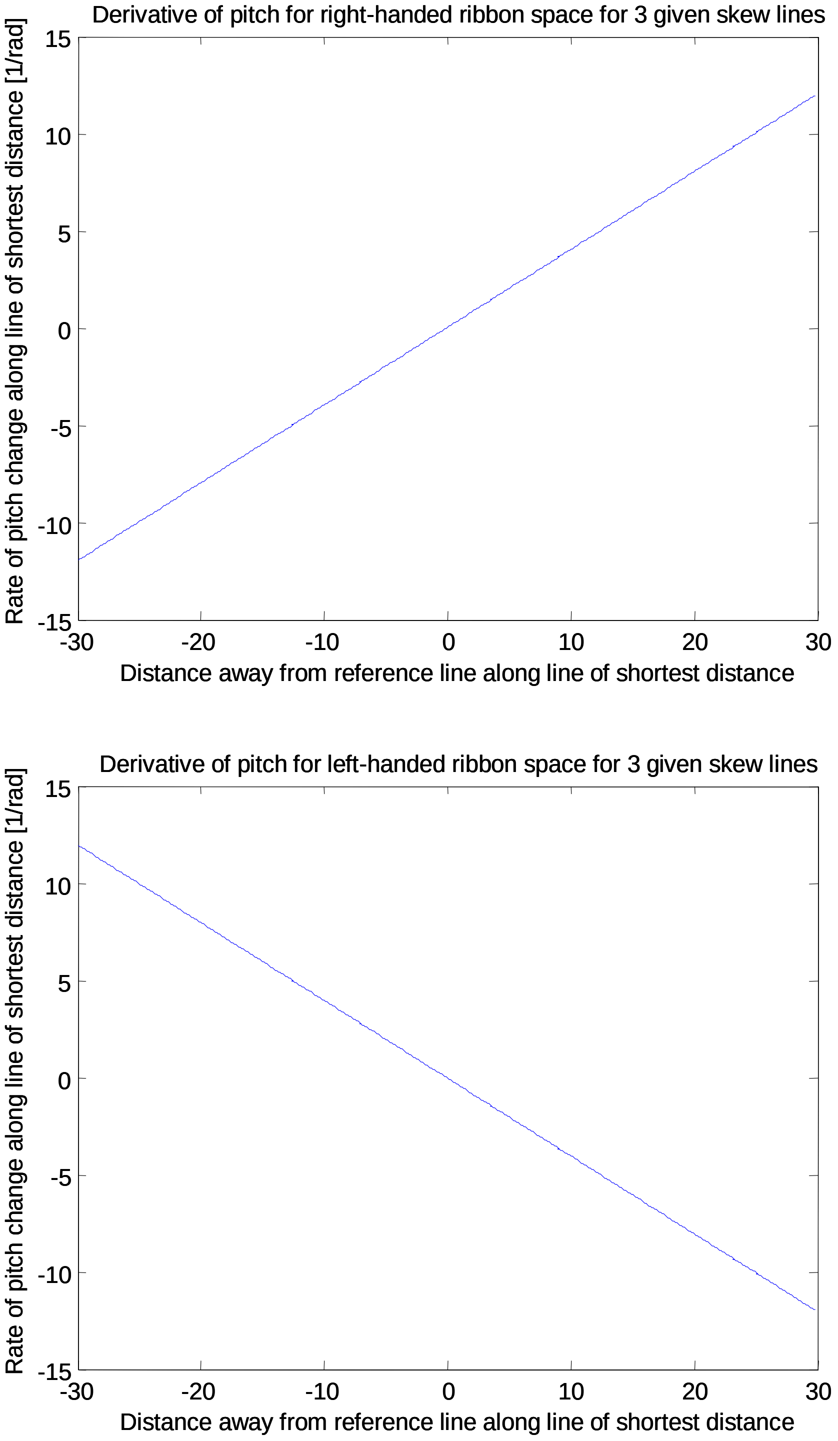


Figure E.4: Derivative of the complementary ribbons' pitches plotted against position along the ribbons' axes.

Running the function of this example will also return these lines in the MATLAB command window:

```
Right_sweep_angle = 2.8113
Right_slope = 0.4000
Left_sweep_angle = 2.8113
Left_slope = -0.4000
```

The “Right_slope” and “Left_slope” variables are the derivatives of the plots given in **Figure E.4** with respect to position along the shortest distance lines or axes of each ribbon. Note that these values are constant and have equal magnitudes (0.4) but opposite signs. Note also that the pitch’s double derivative constant is positive for the right-handed ribbon but it is negative for the complementary left-handed ribbon. The code provided in this appendix allows one to find this characteristic constant for any orthogonal ribbon in three-space.

The “Right_sweep_angle” and “Left_sweep_angle” variables are also provide by MATLAB. These variables are calculated from the difference of the last line’s skew angle calculated at one end of the ribbon with the skew angle of the first line calculated at the other end of the ribbon determined by the range parameter specified by the user when the “ComplementaryRight2Left” function was called (for this example the range was set to 30). In other words, it is the absolute value of the last angle subtracted from the first angle from the plots given in **Figure E.2**. Both of these variables approach 180 degrees or π (since the program uses radians) as the number of lines calculated within the ribbon is made larger. Theoretically, if this range parameter was set to infinity, these variables would both equal π. This finding confirms the hypothesis that all orthogonal ribbons will have a 180 degree twist that occurs at their center point.

The MATLAB functions are provided:

Function (1):

```
function RightHandRibbon(d1,g1,d2,g2,range)
%Finds all lines in a right handed ribbon space by plotting the angle vs.
%position of the lines inside the space along the y-axis
%with respect to a reference line along the z-axis.  The other two
%skew lines are assumed to intersect the y-axis and lie in planes
%parallel to the x-z plane.  This function also finds the pitch of the
%ribbon and plots it as well as the pitches derivative.  The constant double
%derivative of the pitch is displayed as is the amount the skew angle has
%changed over the range of lines tested.
%Input: d1 = shortest distance between reference line and first input line
%       g1 = skew angle between reference line and first input line [rad]
%       d2 = shortest distance between reference line and second input line
%       g2 = skew angle between reference line and second input line [rad]
%       range = distance up and down from origin along y-axis
%Constraints: 0<g1<pi  and 0<g2<pi but g1<g2
r0 = [0 0 0];  %reference line location
f0 = [0 0 1];  %reference line direction
r1 = [0 d1 0];                %middle skew line location
f1 = [sin(g1) 0 cos(g1)];   %middle skew line direction
r2 = [0 d2 0];                %top skew line location
f2 = [sin(g2) 0 cos(g2)];   %top skew line direction
pt1 = r0 + f0;              %first  point on the reference skew line
pt2 = r0 + 2*f0;            %second point on the reference skew line
%find two lines that intersect all three skew lines
c1 = pt1;
w1 = PointSkewLine(r1,f1,r2,f2,pt1);
c2 = pt2;
w2 = PointSkewLine(r1,f1,r2,f2,pt2);
t = [-range:0.1:range];    %location on the y-azis
count = 1;
%find the skew angles for the first half of the ribbon
for i = -range:0.1:-0.1;
   r = [0 i 0];
   f = PointSkewLine(c1,w1,c2,w2,r);
   %check to make sure every line is orthogonal to ribbon's axis
   if(abs(f(2)) >= 1.0e-06) %This should never happen since ribbon is orthogonal
      disp('Error1');
   end
   %a is the skew angle measure from reference line to the line of interest
   if(abs(f(3))<=1.0e-06 && f(1)>0)
      a(count) = -pi/2;
   elseif(abs(f(3))<=1.0e-06 && f(1)<0)
      a(count) = -pi/2;
   elseif(f(3)>0 && abs(f(1))<=1.0e-06)
      a(count) = 0;
```

```
    elseif(f(3)<0 && abs(f(1))<=1.0e-06)
        a(count) = 0;
    elseif(f(3)>0 && f(1)>0)
        a(count) = atan(f(1)/f(3))-pi/2;
    elseif(f(3)>0 && f(1)<0)
        a(count) = atan(abs(f(3))/abs(f(1)));
    elseif(f(3)<0 && f(1)>0)
        a(count) = atan(abs(f(3))/abs(f(1)))-pi/2;
    elseif(f(3)<0 && f(1)<0)
        a(count) = atan(abs(f(1))/abs(f(3)))-pi/2;
    else                     %This will happen if f is a zero vector
        disp('Error2');
    end
    count = count+1;
end
%find the skew angles for the second half of the ribbon
for k = 0:0.1:range;
    r = [0 k 0];
    f = PointSkewLine(c1,w1,c2,w2,r);
    if(abs(f(2)) >= 1.0e-06)
        disp('Error1');
    end
    if(abs(f(3))<=1.0e-06 && f(1)>0)
        a(count) = pi/2;
    elseif(abs(f(3))<=1.0e-06 && f(1)<0)
        a(count) = pi/2;
    elseif(f(3)>0 && abs(f(1))<=1.0e-06)
        a(count) = 0;
    elseif(f(3)<0 && abs(f(1))<=1.0e-06)
        a(count) = 0;
    elseif(f(3)>0 && f(1)>0)
        a(count) = atan(f(1)/f(3));
    elseif(f(3)>0 && f(1)<0)
        a(count) = -atan(abs(f(1))/abs(f(3)));
    elseif(f(3)<0 && f(1)>0)
        a(count) = atan(abs(f(3))/abs(f(1)))-pi;
    elseif(f(3)<0 && f(1)<0)
        a(count) = atan(abs(f(1))/abs(f(3)));
    else
        disp('Error2');
    end
    count = count+1;
end
figure(1);
plot(t,a);
xlabel('Distance away from reference line along the ribbons axis');
```

```
ylabel('Skew angle with respect to reference line [rad]');
title('Right handed ribbon space for 3 given skew lines');
%Display the amount the skew angle has changed over the range of lines
Right_sweep_angle = a(count-1)-a(1)
figure(2)
pit = pitch(t,a);
title('Pitch for right handed ribbon space for 3 given skew lines');
figure(3);
Right_slope = der_pitch(t,pit)
title('Derivative of pitch for right handed ribbon space for 3 given skew lines');
```

Function (2):

```
function LeftHandRibbon(d1,g1,d2,g2,range)
%Finds all lines in a left handed ribbon space by plotting the angle vs.
%position of the lines inside the space along the y-axis
%with respect to a reference line along the z-axis.  The other two
%skew lines are assumed to intersect the y-axis and lie in planes
%parallel to the x-z plane.  This function also finds the pitch of the
%ribbon and plots it as well as the pitches derivative.  The constant double
%derivative of the pitch is displayed as is the amount the skew angle has
%changed over the range of lines tested.
%Input: d1 = shortest distance between reference line and first input line
%       g1 = skew angle between reference line and first input line [rad]
%       d2 = shortest distance between reference line and second input line
%       g2 = skew angle between reference line and second input line [rad]
%       range = distance up and down from origin along y-axis
%Constraints: -pi<g1<0  and -pi<g2<0 but g2<g1
r0 = [0 0 0];  %reference line location
f0 = [0 0 1];  %reference line direction
r1 = [0 d1 0];                %middle skew line location
f1 = [sin(g1) 0 cos(g1)];   %middle skew line direction
r2 = [0 d2 0];                %top skew line location
f2 = [sin(g2) 0 cos(g2)];   %top skew line direction
pt1 = r1 + f1;              %first  point on the middle skew line
pt2 = r1 + 2*f1;            %second point on the middle skew line
%find two lines that intersect all three skew lines
c1 = pt1;
w1 = PointSkewLine(r0,f0,r2,f2,pt1);
c2 = pt2;
w2 = PointSkewLine(r0,f0,r2,f2,pt2);
t = [-range:0.1:range];    %location on the y-azis
count = 1;
for i = -range:0.1:-0.1;
   r = [0 i 0];
   f = PointSkewLine(c1,w1,c2,w2,r);
   if(abs(f(2)) >= 1.0e-06)     %This should theoretically never happen
      disp('Error1');
   end
   if(abs(f(3))<=1.0e-08 && f(1)>0)
      a(count) = pi/2;
   elseif(abs(f(3))<=1.0e-08 && f(1)<0)
      a(count) = pi/2;
   elseif(f(3)>0 && abs(f(1))<=1.0e-08)
      a(count) = 0;
   elseif(f(3)<0 && abs(f(1))<=1.0e-08)
      a(count) = 0;
   elseif(f(3)>0 && f(1)>0)
      a(count) = atan(f(1)/f(3));
   elseif(f(3)>0 && f(1)<0)
      a(count) = atan(abs(f(3))/abs(f(1)))+pi/2;
   elseif(f(3)<0 && f(1)>0)
      a(count) = atan(abs(f(3))/abs(f(1)))+pi/2;
   elseif(f(3)<0 && f(1)<0)
      a(count) = atan(abs(f(1))/abs(f(3)));
   else                  %This will happen if f is a zero vector
      disp('Error2');
   end
   count = count+1;
end
for k = 0:0.1:range;
   r = [0 k 0];
   f = PointSkewLine(c1,w1,c2,w2,r);
   if(abs(f(2)) >= 1.0e-06)     %This should theoretically never happen
      disp('Error1');
   end
   if(abs(f(3))<=1.0e-08 && f(1)>0)
      a(count) = -pi/2;
   elseif(abs(f(3))<=1.0e-08 && f(1)<0)
      a(count) = -pi/2;
   elseif(f(3)>0 && abs(f(1))<=1.0e-08)
      a(count) = 0;
   elseif(f(3)<0 && abs(f(1))<=1.0e-08)
      a(count) = 0;
   elseif(f(3)>0 && f(1)>0)
      a(count) = atan(f(1)/f(3))-pi;
   elseif(f(3)>0 && f(1)<0)
      a(count) = atan(abs(f(3))/abs(f(1)))-pi/2;
   elseif(f(3)<0 && f(1)>0)
      a(count) = atan(abs(f(3))/abs(f(1)))-pi/2;
   elseif(f(3)<0 && f(1)<0)
      a(count) = atan(abs(f(1))/abs(f(3)))-pi;
```

```
    else                         %This will happen if f is a zero vector
        disp('Error2');
    end
    count = count+1;
end
figure(4);
plot(t,a);
xlabel('Distance away from reference line along the ribbons axis');
ylabel('Skew angle with respect to reference line [rad]');
title('Left handed ribbon space for 3 given skew lines');
%Display the amount the skew angle has changed over the range of lines
Left_sweep_angle = a(1)-a(count-1)
figure(5);
pit = pitch(t,a);
title('Pitch for left handed ribbon space for 3 given skew lines');
figure(6);
Left_slope = der_pitch(t,pit)
title('Derivative of pitch for left handed ribbon space for 3 given skew lines');
```

Function (3):

```
function ComplementRight2Left(d1,g1,d2,g2,range);
%Given a right handed ribbon space this function finds its complementary left
%handed ribbon space
%Input: d1 = shortest distance between reference line and first input line
%       g1 = skew angle between reference line and first input line [rad]
%       d2 = shortest distance between reference line and second input line
%       g2 = skew angle between reference line and second input line [rad]
%       range = distance up and down from origin along y-axis
%Constraints: 0<g1<pi  and 0<g2<pi but g1<g2
RightHandRibbon(d1,g1,d2,g2,range);
r0 = [0 0 0];  %reference line location
f0 = [0 0 1];  %reference line direction
r1 = [0 d1 0];                %middle skew line location
f1 = [sin(g1) 0 cos(g1)];   %middle skew line direction
r2 = [0 d2 0];                %top skew line location
f2 = [sin(g2) 0 cos(g2)];   %top skew line direction
pt1 = r1 + 2*f1;              %first  point on the middle skew line
pt2 = r1 + 1*f1;            %second point on the middle skew line
pt3 = r1 + 0*f1;            %third point on the middle skew line
%find three lines that intersect all three skew lines
c1 = pt1;
w1 = PointSkewLine(r0,f0,r2,f2,pt1);
c2 = pt2;
w2 = PointSkewLine(r0,f0,r2,f2,pt2);
c3 = pt3;
```

```
w3 = PointSkewLine(r0,f0,r2,f2,pt3);
%Make sure all w's are pointing in the correct direction for a
%Left handed ribbon
g23 = -abs(acos(dot(w2,w3)/(sqrt(dot(w2,w2))*sqrt(dot(w3,w3)))));
g13 = -abs(acos(dot(w1,w3)/(sqrt(dot(w1,w1))*sqrt(dot(w3,w3)))));
ndir13 = cross(w1,w3);
n13 = ndir13/sqrt(dot(ndir13,ndir13));
d13 = abs(dot(n13,(c1-c3)));
ndir23 = cross(w2,w3);
n23 = ndir23/sqrt(dot(ndir23,ndir23));
d23 = abs(dot(n23,(c2-c3)));
LeftHandRibbon(d23,g23,d13,g13,range);
```

Function (4):

```
function ComplementLeft2Right(d1,g1,d2,g2,range);
%Given a left handed ribbon space this function finds its complementary right
%handed ribbon space
%Input: d1 = shortest distance between reference line and first input line
%       g1 = skew angle between reference line and first input line [rad]
%       d2 = shortest distance between reference line and second input line
%       g2 = skew angle between reference line and second input line [rad]
%       range = distance up and down from origin along y-axis
%Constraints: -pi<g1<0  and -pi<g2<0 but g2<g1
LeftHandRibbon(d1,g1,d2,g2,range);
r0 = [0 0 0];  %reference line location
f0 = [0 0 1];  %reference line direction
r1 = [0 d1 0];                %middle skew line location
f1 = [sin(g1) 0 cos(g1)];   %middle skew line direction
r2 = [0 d2 0];                %top skew line location
f2 = [sin(g2) 0 cos(g2)];   %top skew line direction
pt1 = r1 + 2*f1;             %first  point on the middle skew line
pt2 = r1 + 1*f1;           %second point on the middle skew line
pt3 = r1 + 0*f1;           %third point on the middle skew line
%find three lines that intersect all three skew lines
c1 = pt1;
w1 = PointSkewLine(r0,f0,r2,f2,pt1);
c2 = pt2;
w2 = PointSkewLine(r0,f0,r2,f2,pt2);
c3 = pt3;
w3 = PointSkewLine(r0,f0,r2,f2,pt3);
%Make sure all w's are pointing in the correct direction for a
%Right handed ribbon
g23 = abs(acos(dot(w2,w3)/(sqrt(dot(w2,w2))*sqrt(dot(w3,w3)))));
g13 = abs(acos(dot(w1,w3)/(sqrt(dot(w1,w1))*sqrt(dot(w3,w3)))));
ndir13 = cross(w1,w3);
```

```
n13 = ndir13/sqrt(dot(ndir13,ndir13));
d13 = abs(dot(n13,(c1-c3)));
ndir23 = cross(w2,w3);
n23 = ndir23/sqrt(dot(ndir23,ndir23));
d23 = abs(dot(n23,(c2-c3)));
RightHandRibbon(d23,g23,d13,g13,range);
```

Function (5):

```
function [p] = pitch(tt,a);
%Plots the pitch of a ribbon space vs position along the y-axis
%Input: a = skew angle relative to reference line
%       tt = postion along the y-axis
%Output: returns pitch, p
t=tt(1:length(tt)-1);
p=diff(tt)./diff(a);
plot(t,p);
xlabel('Distance away from reference line along line of shortest distance');
ylabel('Pitch of ribbon space [distance/rad]');
```

Function (6):

```
function [slope] = der_pitch(tt,p);
%Plots the pitch of a ribbon space vs position along the y-axis
%Input: p = pitch
%           tt = postion along the y-axis
%Output: return slope of line (double derivative of pitch), slope
for j = 1:(length(tt)-2)
    t(j) = tt(j);
end
for k = 1:(length(tt)-2);
    der_p(k) = (p(k+1)-p(k))/(tt(k+1)-tt(k));
end
plot(t,der_p);
xlabel('Distance away from reference line along line of shortest distance');
ylabel('Rate of pitch change along line of shortest distance [1/rad]');
len = length(t);
slope = (der_p(len)-der_p(round(len/2)))/(t(len)-t(round(len/2)));
```

APPENDIX F:

"Characteristic Screw's Pitch Related to the Ribbon's Pitch Double Derivative Constant"

This appendix mathematically proves **Equation (7.2)** which defines the relationship between the pitch of a ribbon's characteristic screw, *p*, and the double derivative constant of the ribbon's pitch, *K*, discussed in **Appendix E**.

Recall from **Equation (3.13)** in **Chapter 3** that the pitch of a screw, *p*, is defined as

$$p = d\tan\theta, \tag{F.1}$$

where *d* is the shortest distance between the screw line and a constraint line and where θ is the skew angle between these lines. Recall also that the constant *K* is the double derivative of a ribbon's pitch with respect to position along the ribbon's axis where the pitch of a ribbon, P_{ribbon}, is defined as the change in the position of constraint lines along the ribbon's axis over the change in the skew angle between these lines written as

$$P_{ribbon} = \frac{\Delta d}{\Delta\theta} = \frac{d(d)}{d\theta}, \tag{F.2}$$

For the characteristic screw of an orthogonal ribbon, the *d* variable from **Equation (F.1)** is equivalent to the *d* variable in **Equation (F.2)** since the shortest distance line between this screw and its constraint lines is synonymous with the axis of the ribbon. Although the skew angles θ are defined slightly differently within each of these two equations, the rate that they change along the ribbon's axis will be the same. One can, therefore, relate the characteristic screw's pitch with the constant *K* by reorganizing **Equation (F.1)** as

$$d = \frac{p}{\tan\theta}, \tag{F.3}$$

and taking its derivative with respect to *θ* in order to find the ribbon’s pitch using **Equation (F.2)**. One finds that

$$P_{ribbon} = -\frac{p}{(\sin\theta)^2}. \tag{F.4}$$

Using **Equation (F.1)** and the trigonometric identity shown in **Figure E.1**, one finds that

$$\sin\theta = \frac{p}{\sqrt{d^2 + p^2}}. \tag{F.5}$$

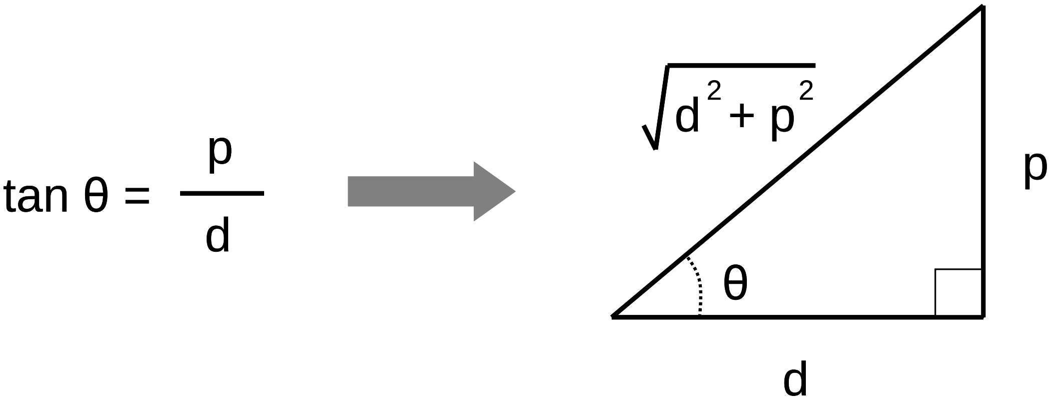


Figure F.8: Trigonometric identity created using Equation (F.1)

When **Equation (F.5)** is substituted into **Equation (F.4)**, an expression for the pitch of a ribbon with respect to position along the ribbon’s axis is found and given as

$$P_{ribbon} = -\frac{d^2}{p} - p. \tag{F.6}$$

Note that this function is parabolic. This is consistent with the plots shown in **Figure E.3** from **Appendix E**. If the derivative of this equation is taken with respect to *d*, one finds

$$\frac{dP_{ribbon}}{dd} = -\frac{2d}{p}\,. \tag{F.7}$$

This function is linear and is consistent the plots shown in **Figure E.4** from **Appendix E**. If the derivative of this equation is taken with respect to *d*, the constant *K* with respect to the characteristic screw's pitch, *p*, is found and given as

$$\frac{d^2 P_{ribbon}}{dd^2} = -\frac{2}{p} = K\,. \tag{F.8}$$

Equation (7.2) is obtained by rearranging **Equation (F.8)**.

Recall from **Appendix E** that the constant, *K*, that was found using the three skew constraint lines from the example was given as 0.4 for the right-handed orthogonal ribbon constraint set. **Equation (7.2)** suggests that this ribbon may be characterized by a screw that intersects and is orthogonal to its center line and its axis with a pitch of -5. **Equation (F.1)** can be used to confirm that a screw with a pitch of -5 does complement the three original skew constraint lines inputted by the user as "ComplementRight2Left(5,(pi/4),10,((pi/2)-atan(.5)),30)".

APPENDIX G:

"Hyperbolic Paraboloids Composed of Orthogonal Ribbon Sets Expressed in Terms of Characteristic Screw's Pitch"

This appendix mathematically proves **Equation (7.3)**, which describes the surface of a hyperbolic paraboloid composed of orthogonal ribbon freedom and constraint sets in terms of their characteristic screw's pitch.

First, the objective is to describe the constraint lines within a ribbon in terms of their location and orientation vectors, $\vec{r}$ and $\vec{f}$ respectively. **Figure G.1** helps clarify the geometry of the constraint lines within the orthogonal ribbon with respect to the characteristic screw. Using this figure, it is determined that a possible location vector, $\vec{r}$, for every constraint line in the orthogonal ribbon may mathematically be expressed as

$$\vec{r} = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \end{bmatrix} d \qquad \text{(G.1)}$$

where *d* is the vector's magnitude. This vector always points along the constraint ribbon's axis. Using the same figure and applying **Equation (3.13)**, it can also be determined that the orientation vector, $\vec{f}$, for every constraint line in the orthogonal ribbon in terms of its characteristic screw's pitch, *p*, is given as

$$\vec{f} = \begin{bmatrix} -\frac{p}{\sqrt{2}} & -\frac{p}{\sqrt{2}} & d \end{bmatrix}. \qquad \text{(G.2)}$$

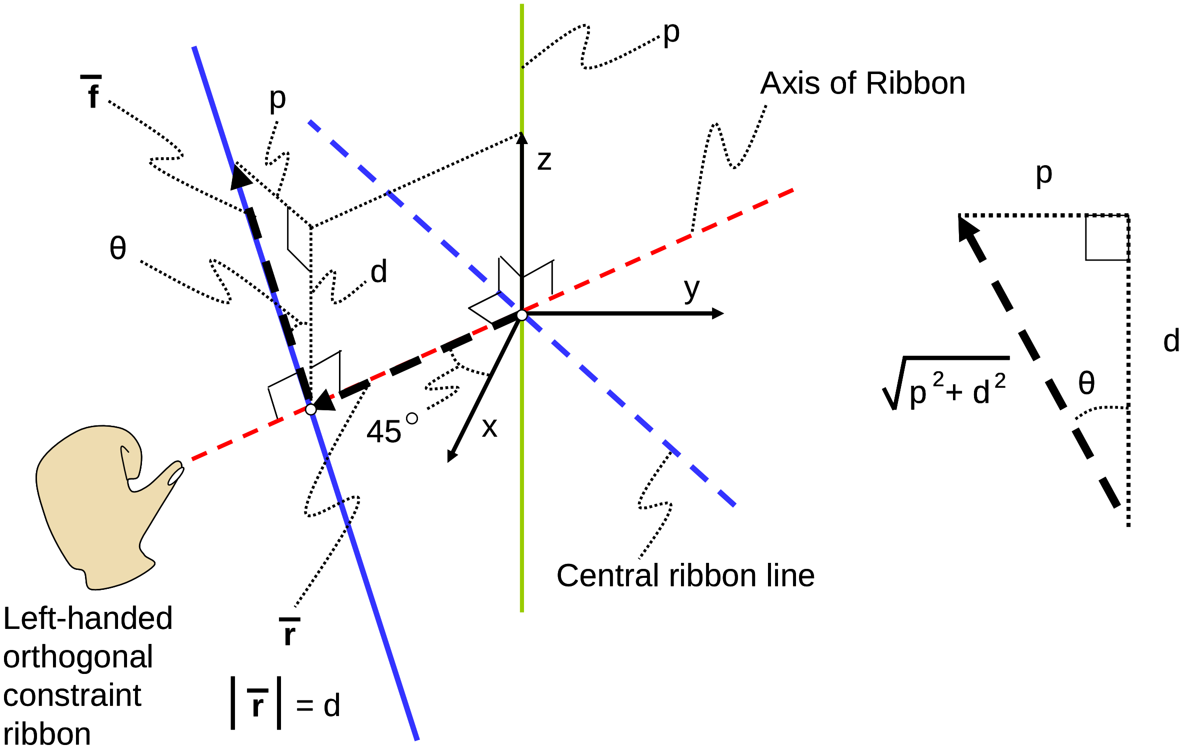


Figure G.1: Defining the location and orientation vectors, $\vec{r}$ and $\vec{f}$, for every constraint line (blue) within an orthogonal ribbon with its characteristic screw (green)

Since every constraint line must lie entirely on the surface of the hyperbolic paraboloid, note that the point

$$[x \quad y \quad z] = \vec{r} + \vec{f}\,t\,, \tag{G.3}$$

will also lie on the hyperbolic paraboloid where t is any real scalar value. If **Equation (G.1)** and **Equation (G.2)** are substituted into **Equation (G.3)**, one finds that

$$x = \left(\frac{d - pt}{\sqrt{2}}\right)$$

$$y = -\left(\frac{d + pt}{\sqrt{2}}\right) \tag{G.4}$$

$$z = dt\,.$$

If these values are substituted into the equation of a hyperbolic paraboloid given in **Chapter 6** as **Equation (6.1)**, and the *a* and *b* values for any two different pairs of *d* and *t* values are solved for, one finds that

$$a = b = \sqrt{2p} \ . \tag{G.5}$$

Substituting **Equation (G.5)** into **Equation (6.1)** proves that **Equation (7.3)** is in deed a true description of a hyperbolic paraboloid that contains complementary orthogonal ribbon sets in terms of their characteristic screw’s pitch.

APPENDIX H:

“Hyperbolic Paraboloids Composed of Non-orthogonal Ribbon Sets Expressed in Terms of Characteristic Screw’s Pitch”

This appendix mathematically proves **Equation (7.4)**, which relates the characteristic screw’s pitch of a hyperbolic paraboloid composed of non-orthogonal ribbon freedom and constraint sets to the *a* and *b* values used in the equation for the hyperbolic paraboloid given as **Equation (6.1)** from **Chapter 6**. **Equation (7.5)** is also verified as being the normal vector of the pure rotational hoop within the freedom space of Case 3, Type 7.

First, the objective is to describe the constraint lines within the non-orthogonal constraint ribbon in terms of their location and orientation vectors, $\vec{r}$ and $\vec{f}$ respectively. **Figure H.1** depicts a view of the non-orthogonal constraint ribbon looking down its characteristic screw along the z-axis. This view helps clarify the geometric relationship of the constraint lines within the ribbon. Using this figure one can determine that a possible location vector, $\vec{r}$, for every constraint line in the non-orthogonal ribbon may mathematically be expressed as

$$\vec{r} = \begin{bmatrix} \frac{a}{\sqrt{a^2+b^2}} & -\frac{b}{\sqrt{a^2+b^2}} & 0 \end{bmatrix} d \, , \qquad \text{(H.1)}$$

where *d* is the vector’s magnitude and *a* and *b* are the values used in the equation of a hyperbolic paraboloid given in **Chapter 6** as **Equation (6.1)**. The components of this vector were determined by noting the slopes of the asymptotic lines or axes of the complementary ribbons given in **Equation (6.2)**. This vector always points along the constraint ribbon’s axis.

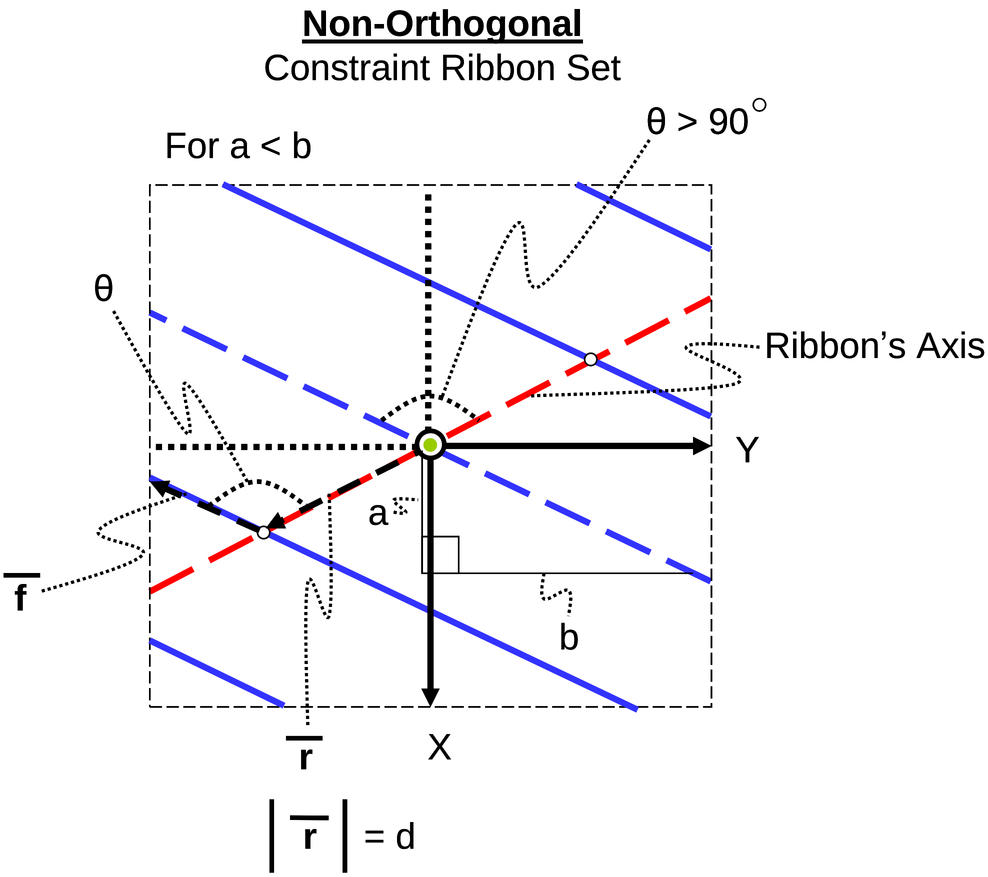


Figure H.1: Defining the location and orientation vectors, $\vec{r}$ and $\vec{f}$, for every constraint line (blue) within a non-orthogonal ribbon from a view looking down the characteristic screw (green) of the hyperbolic paraboloid (along the z-axis).

Note also that if *a*=*b*, the constraint ribbon is orthogonal and one would expect to find the same location vector as the location vector found in **Appendix G**. For this condition, note that **Equation (G.1)** does indeed equal **Equation (H.1)**. This observation validates the choice of location vector.

Finding the complete orientation vector, $\vec{f}$, is not clear from **Figure H.1** alone, however. From the figure, the x- and y-components of the vector may be determined, but its z-component, C_z, may not be. For now, this vector will be expressed as

$$\vec{f} = [-a \quad -b \quad C_z]. \tag{H.2}$$

To solve for this unknown z-component, C_z, within the constraint line's orientation vector, one must consider the characteristic screw of the constraint ribbon set that lies along the z-axis. This characteristic screw has a pitch value of *p* and has a location and orientation vector, $\vec{c}$ and $\vec{w}$, of

$$\begin{aligned} \vec{c} &= \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \\ \vec{w} &= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}. \end{aligned} \tag{H.3}$$

If the constraint line's vectors from **Equation (H.1)**, **Equation (H.2)**, and the characteristic screw's vectors from **Equation (H.3)** are substituted into the pitch equation given in **Appendix B** as **Equation (B.4)**, the characteristic screw's pitch, *p*, is found to be

$$p = \frac{2abd}{C_z\sqrt{a^2+b^2}}. \tag{H.4}$$

Using **Equation (H.4)**, the z-component, C_z, is solved for. The constraint line's complete orientation vector, $\vec{f}$, is therefore

$$\vec{f} = \begin{bmatrix} -a & -b & \dfrac{2abd}{p\sqrt{a^2+b^2}} \end{bmatrix}. \tag{H.5}$$

Note also that if *a*=*b*, the constraint ribbon would be orthogonal and one would expect to find the same orientation vector as the orientation vector found in **Appendix G**. For this condition, the vector given in **Equation (G.2)** does point in the same direction as the vector given in **Equation (H.5)**. This finding validates the selection of the orientation vector for non-orthogonal constraint ribbons.

Since every constraint line must lie entirely on the surface of the hyperbolic paraboloid, one can know that the point

$$\begin{bmatrix} x & y & z \end{bmatrix} = \vec{r} + \vec{f}\,t \tag{H.6}$$

will also lie on the hyperbolic paraboloid where t is any real scalar value. If **Equations (H.1)** and **Equations (H.5)** are substituted into **Equation (H.6)**, one finds that

$$x = \frac{ad}{\sqrt{a^2 + b^2}} - at$$

$$y = -\frac{bd}{\sqrt{a^2 + b^2}} - bt \tag{H.7}$$

$$z = \frac{2abdt}{p\sqrt{a^2 + b^2}} .$$

If these values are substituted into the equation for a hyperbolic paraboloid given in **Chapter 6** as **Equation (6.1)** and p is solved for in terms of the a and b values, **Equation (7.4)** is proven.

The normal vector, $\vec{n}$, of the pure rotational hoop may be found by taking the cross product of the orientation vector, $\vec{f}$, of the constraint lines given in **Equation (H.5)** and the orientation vector, $\vec{w}$, of the characteristic screw given in **Equation (H.3)**. The resulting vector of this cross product is given in **Equation (7.5)** from **Chapter 7**.

APPENDIX I:

“Finding the Screws of Case 3, Type 7 for Orthogonal Ribbon Sets”

This appendix proves that every twist within the freedom space of Case 3, Type 7 will exist within disks that lie on planes that are perpendicular to the axis of the freedom ribbon set (i.e. the central line of the constraint ribbon set) for the case of complementary orthogonal ribbon sets. The center points of these disks are intersected by the freedom ribbon set’s axis.

To begin the proof, consider three non-redundant skew constraint lines that produce orthogonal complementary ribbon sets. These three constraint lines are shown in **Figure I.1**. A characteristic screw is arbitrarily defined along the y-axis and is given a pitch of -1. The first constraint line is the central line within the constraint ribbon set and lies along the x-axis. The other two skew constraint lines intersect and are perpendicular to the z-axis and are positioned in such a way as to complement the ribbon’s characteristic screw. From **Chapter 7** it is known that these three skew constraint lines will produce a right-handed orthogonal constraint ribbon set with a complementary left-handed orthogonal freedom set whose axis is the central constraint line along the x-axis.

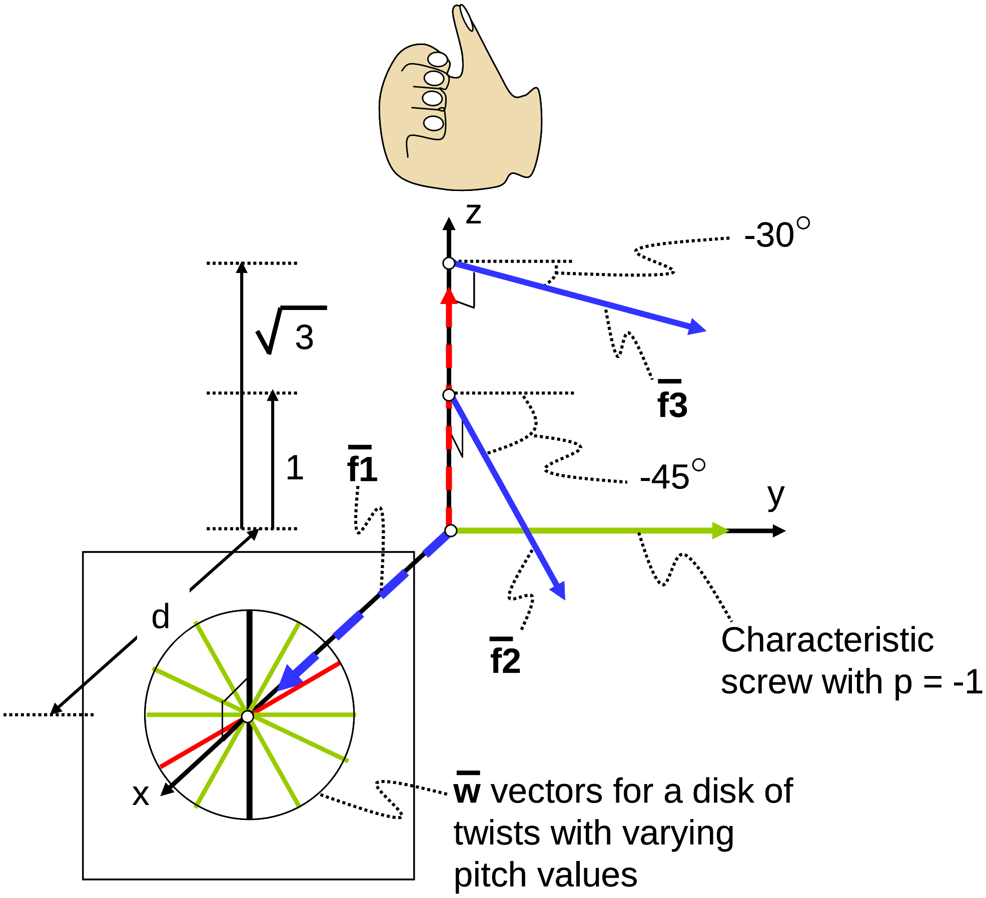


Figure I.1: Three skew constraint lines (blue) that produce a system of complementary orthogonal constraint and freedom ribbon sets with a characteristic screw's pitch of -1 (thick green). A disk of twists is shown that contains a pure rotational freedom line (red), a pure translational line (thick black), and an infinite number of screw lines (green).

One can verify the existence of this pure rotational freedom ribbon set and locate the screws of the system by using the mathematical approach described in **Chapter 3** of **Section 3.4.2**. This is done by first expressing the three non-redundant constraint lines as wrenches. From **Figure I.1** these three wrenches can be found and defined to be

$$\begin{aligned}\vec{W}_1 &= \begin{bmatrix}1 & 0 & 0 & 0 & 0 & 0\end{bmatrix}\\ \vec{W}_2 &= \begin{bmatrix}1 & 1 & 0 & -1 & 1 & 0\end{bmatrix}\\ \vec{W}_3 &= \begin{bmatrix}1 & \sqrt{3} & 0 & -3 & \sqrt{3} & 0\end{bmatrix}.\end{aligned} \tag{I.1}$$

A 3x6 wrench matrix may then be constructed using the three wrenches from **Equation (I.1)**. The null space of this matrix may then be calculated to find the complementary twists of the system. When this null space has been found and the $\vec{w}$ and $\vec{v}$ vectors have properly been switched within the resulting vectors to maintain the twist convention given in **Equation (3.1)**, the resultant twist vector of the system may be expressed as a linear combination of three independent twist vectors as

$$A\begin{bmatrix} 0 \\ -1 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} + B\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + C\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ -A \\ B \\ 0 \\ A \\ C \end{bmatrix} = \vec{T} \ , \tag{I.2}$$

where *A*, *B*, and *C* may be any real numbers. The 6×1 twist vector at the far right of **Equation (I.2)** is the complete mathematical representation of every possible twist for the system of three skew constraints. This resultant twist's rotational and translational velocity vectors, $\vec{w}$ and $\vec{v}$, are the following:

$$\begin{aligned} \vec{w} &= \begin{bmatrix} 0 & -A & B \end{bmatrix} \\ \vec{v} &= \begin{bmatrix} 0 & A & C \end{bmatrix}. \end{aligned} \tag{I.3}$$

Using these vectors and the definition of pitch given in **Equation (3.4),** one finds that every twist in the system must have a pitch, *p*, that equals

$$p = \frac{-A^2 + CB}{A^2 + B^2} . \tag{I.4}$$

It can also be concluded that every twist in the system must lie on planes that are parallel to or coincident with the y-z plane since the x-component of the twist's orientation vector, $\vec{w}$, is always zero. These planes are the parallel planes that the skew pure rotational freedom lines from the freedom ribbon set are expected to lie on.

If every allowable twist within the freedom space of this system also only exists within disks on these planes with center points that pass through the axis of the orthogonal freedom ribbon set (x-axis), one would expect the location vector

$$\vec{c} = \begin{bmatrix} d & 0 & 0 \end{bmatrix} \tag{I.5}$$

to be a possible location vector for every allowable twist within the system where *d* is any real number that corresponds to any location along the x-axis.

To check the validity of this claim, the twist's location vector, $\vec{c}$, given in **Equation (I.5)**, the twist's rotational and linear velocity vectors given in **Equation (I.3)**, and the twist's pitch given in **Equation (I.4)** are substituted into **Equation (3.5)** from **Chapter 3**. Three equations result after performing this substitution. They are

$$\begin{aligned} &0 = 0 \\ &A = -Bd + \frac{A^3 - ACB}{A^2 + B^2} \\ &C = -Ad + \frac{-A^2B + CB^2}{A^2 + B^2}. \end{aligned} \tag{I.6}$$

If the location vector, $\vec{c}$, given in **Equation (I.5)** is true for every twist within the freedom space of this system, all three equations from **Equation (I.6)** should always be true for any real values of *A*, *B*, *C*, and *d*. Clearly zero will always equal zero, but the last two equations from **Equation (I.6)** will each both independently simplify to

$$AB + A^2d + B^2d + AC = 0. \tag{I.7}$$

These last two equations could be subtracted from each other to again show that zero equals zero which will always be true. This proves that every twist line must pass through the x-axis or the axis of the orthogonal freedom ribbon set at least once.

Since it has been proven that every twist line must intersect the axis of the orthogonal freedom ribbon set and must lie on planes parallel to the parallel planes of the pure rotational freedom lines, it has been proven that every twist line lies within a disk like the one shown in **Figure I.1**.

Every such disk along the orthogonal freedom ribbon set's axis will contain a single pure rotational freedom line that corresponds to a line within the freedom ribbon set. These pure rotational freedom lines correspond to the twists that have zero pitch values and satisfy the condition

$$A^2 = CB. \tag{I.8}$$

This equation is derived by setting **Equation (I.4)** equal to zero.

Note also from the definition of the resultant twist given in **Equation (I.2)** that a pure translation will only exist within the freedom space of this system when A and B equal zero such that $\vec{w}$ becomes a zero vector. When this condition is satisfied, only one pure translation that points along the z-axis exists, which is the axis of the orthogonal constraint ribbon set.

Every disk of twists within the freedom space of complementary orthogonal ribbon sets will, therefore, contain a single pure rotational freedom line, a single pure translational line, and an infinite number of screws with pitch values that vary according to location within the disk. This conclusion is depicted in **Figure I.1**.

APPENDIX J:

"Equation for a Circular Hyperboloid"

This appendix proves **Equation (7.6)** as the equation for a circular hyperboloid in terms of the parameters L and α defined in **Figure 7.82** from **Chapter 7**.

To prove **Equation (7.6)**, the variable c in **Equation (6.3)** from **Chapter 6** must be solved for in terms of these desired parameters. To achieve this task, a point on the surface of a general circular hyperboloid must be located. This point is essentially any point along any constraint line that lies on the circular hyperboloid's surface. If one arbitrarily chooses the constraint line on the hyperboloid that also intersects the x-axis as shown in **Figure J.1**, one can define its location vector, $\vec{r}$, as

$$\vec{r} = \begin{bmatrix} L & 0 & 0 \end{bmatrix}, \tag{J.1}$$

where L is the radius of the hyperboloid's central circular cross-section. Using the relationships shown in **Figure J.1**, the orientation vector, $\vec{f}$, can also be determined for this constraint line in terms of α as

$$\vec{f} = \begin{bmatrix} 0 & \cos\alpha & \sin\alpha \end{bmatrix}. \tag{J.2}$$

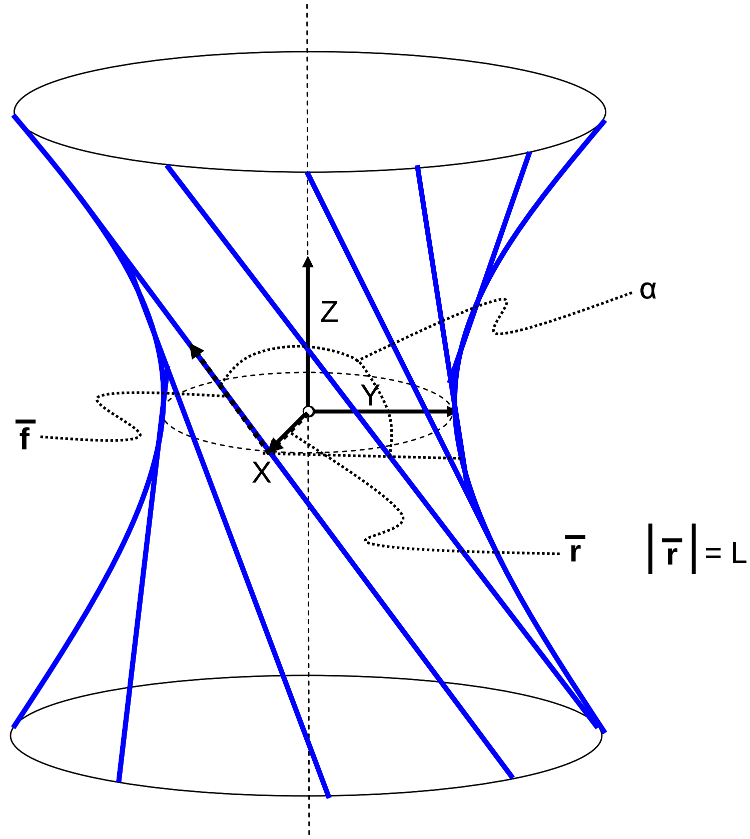


Figure J.1: Defining the location and orientation vectors, $\vec{r}$ and $\vec{f}$, for a constraint line (blue) within a circular hyperboloid

Since every point along this constraint line must lie entirely on the surface of the hyperboloid, one can know that the point

$$\begin{bmatrix} x & y & z \end{bmatrix} = \vec{r} + \vec{f}\, t\,, \tag{J.3}$$

will also lie on the hyperboloid where *t* is any real scalar value. If **Equation (J.1)** and **Equation (J.2)** are substituted into **Equation (J.3)**, one finds that

$$\begin{aligned} x &= L \\ y &= t\cos\alpha \\ z &= t\sin\alpha\,. \end{aligned} \tag{J.4}$$

If these values are substituted into the equation for a circular hyperboloid given in **Chapter 6** as **Equation (6.3)** and the *c* parameter is solved for, one finds that

$$c = L\tan\alpha\ . \tag{J.5}$$

Substituting this value into **Equation (6.3)** proves that **Equation (7.6)** is the true description of a circular hyperboloid in terms of *L* and *α*.

APPENDIX K:

“Equation for an Elliptical Hyperboloid”

This appendix proves **Equation (7.7)** and **Equation (7.8)** to be two possible equations that describe an elliptical hyperboloid in terms of the parameters *a*, *b*, α_1 and α_2 defined in **Figure 7.84** from **Chapter 7**.

To prove these equations, the variable *c* in **Equation (6.4)** from **Chapter 6** must be solved for in terms of the desired parameters. To achieve this task, a point on the surface of a general elliptical hyperboloid must be located. This point is essentially any point along any constraint line that lies on the elliptical hyperboloid’s surface. If the constraint line on the hyperboloid is chosen that also intersects the x-axis as shown in **Figure K.1**, its location vector, $\vec{r}_1$, can be defined as

$$\vec{r}_1 = \begin{bmatrix} a & 0 & 0 \end{bmatrix}, \tag{K.1}$$

where *a* is the length of the hyperboloid’s central elliptical cross-section’s major axis. Using the relationships shown in **Figure K.1**, the orientation vector, $\vec{f}_1$, can also be determined for this constraint line in terms of α_1 as

$$\vec{f}_1 = \begin{bmatrix} 0 & \cos\alpha_1 & \sin\alpha_1 \end{bmatrix}. \tag{K.2}$$

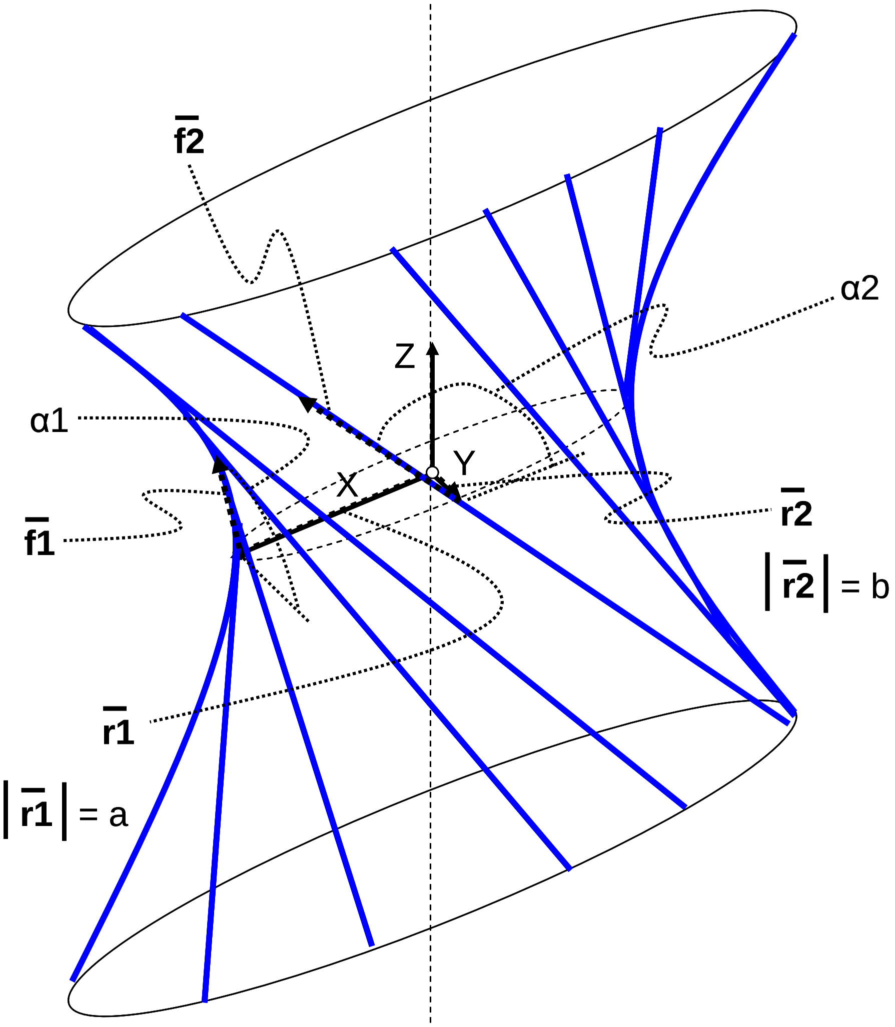


Figure K.1: Defining the location and orientation vectors, $\vec{r}$ and $\vec{f}$, for two constraint lines (blue) within an elliptical hyperboloid

Since every point along this constraint line must lie entirely on the surface of the hyperboloid, one can know that the point

$$[x_1 \quad y_1 \quad z_1] = \vec{r}_1 + \vec{f}_1 t\,, \qquad \text{(K.3)}$$

will lie on the hyperboloid where t is any real scalar value. If **Equation (K.1)** and **Equation (K.2)** are substituted into **Equation (K.3)**, one finds that

$$x_1 = a$$

$$y_1 = t\cos\alpha_1 \qquad \text{(K.4)}$$

$$z_1 = t\sin\alpha_1 .$$

If these values are substituted into the equation for an elliptical hyperboloid given in **Chapter 6** as **Equation (6.4)** and the *c* parameter is solved for, one finds that

$$c = b\tan\alpha_1 . \qquad \text{(K.5)}$$

Substituting this value into **Equation (6.4)**, **Equation (7.7)** is proven to be a true description of an elliptical hyperboloid in terms of *a*, *b*, and α_1.

Now **Equation (7.8)** is also proven to be true by choosing another point on the surface of the elliptical hyperboloid. If one chooses the constraint line on the hyperboloid that intersects the y-axis as shown in **Figure K.1**, its location vector, $\vec{r}_2$, is defined as

$$\vec{r}_2 = \begin{bmatrix} 0 & b & 0 \end{bmatrix}, \qquad \text{(K.6)}$$

where *b* is the length of the hyperboloid's central elliptical cross-section's minor axis. Using the relationships shown in **Figure K.1**, one can also determine the orientation vector, $\vec{f}_2$, for this constraint line in terms of α_2 as

$$\vec{f}_2 = \begin{bmatrix} -\cos\alpha_2 & 0 & \sin\alpha_2 \end{bmatrix}. \qquad \text{(K.7)}$$

Since every point along this constraint line must also lie entirely on the surface of the hyperboloid, the point

$$\begin{bmatrix} x_2 & y_2 & z_2 \end{bmatrix} = \vec{r}_2 + \vec{f}_2 t , \qquad \text{(K.8)}$$

will lie on the hyperboloid where *t* is any real scalar value. If **Equation (K.6)** and **Equation (K.7)** are substituted into **Equation (K.8)**, one finds that

$$
\begin{aligned}
x_2 &= -t\cos\alpha_2 \\
y_2 &= b \qquad\qquad \text{(K.9)} \\
z_2 &= t\sin\alpha_2 .
\end{aligned}
$$

If these values are substituted into the equation for an elliptical hyperboloid given in **Chapter 6** as **Equation (6.4)** and the *c* parameter is solved for, one finds that

$$c = a\tan\alpha_2 . \qquad \text{(K.10)}$$

Substituting this value into **Equation (6.4)** proves that **Equation (7.8)** is another true description of an elliptical hyperboloid in terms of *a*, *b*, and α_2.

APPENDIX L:

“Two Orthogonal Intersecting Twists Generate a Cylindroid”

This appendix proves that the linear combination of any two orthogonal, intersecting twists will result in an infinite number of twists that all lie on the surface of a cylindroid where the two original orthogonal twists are the cylindroid’s principal generators.

In order to prove that this statement is true, first assume that it is. If it is true, the location vector, $\vec{c}$, of every point along every twist line on the surface of the cylindroid will satisfy the equation of a cylindroid given in **Equation (6.5)** as

$$\begin{aligned} c_x &= r\cos\theta \\ c_y &= r\sin\theta \\ c_z &= -h\cos\theta\sin\theta\,. \end{aligned} \tag{L.1}$$

where *r*, *h*, and θ are defined in **Figure L.1**.

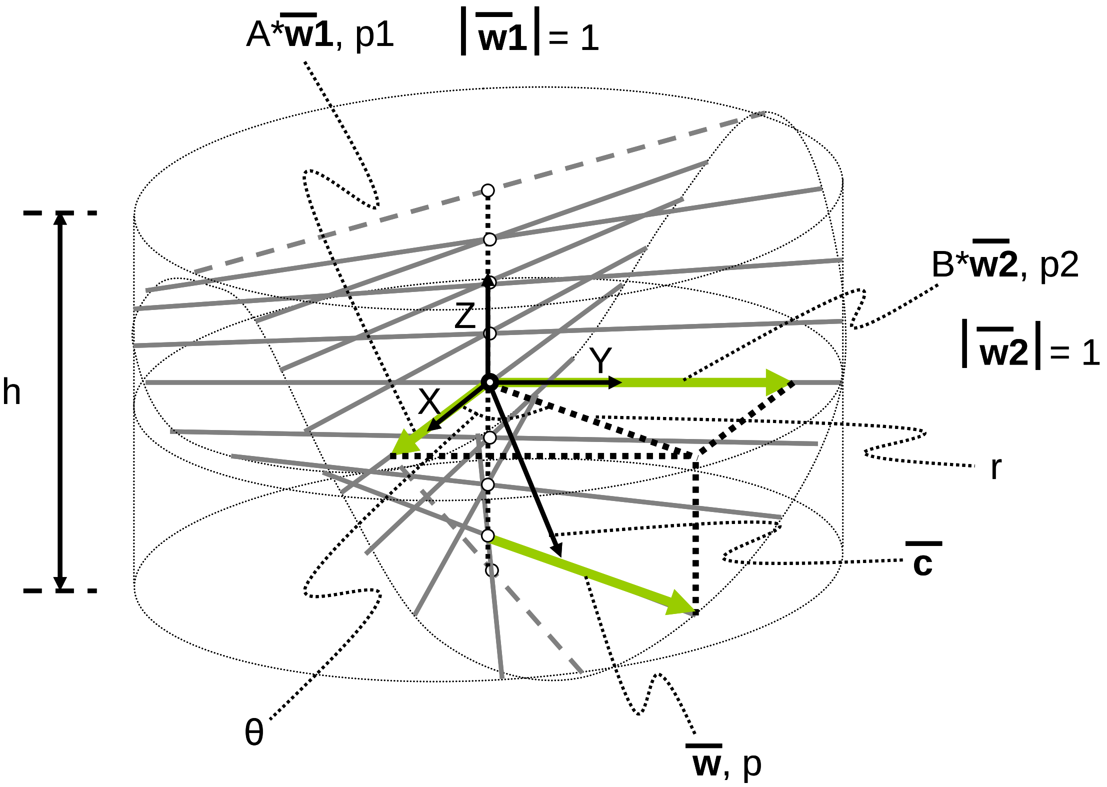


Figure L.1: Parameters defined for two orthogonal, intersecting twists. Every linear combination of these two twists results in another twist that lies on the surface of a cylindroid with principal generators that are the lines coincident with the two original twists.

Note also from **Figure L.1** that

$$\cos\theta = \frac{A}{\sqrt{A^2+B^2}}$$
$$\sin\theta = \frac{B}{\sqrt{A^2+B^2}}\,. \qquad \text{(L.2)}$$

If **Equation (L.2)** is plugged into **Equation (L.1)**, another expression for the location vector of every possible twist on the surface of the cylindroid is found to be

$$c_x = \frac{rA}{\sqrt{A^2 + B^2}}$$

$$c_y = \frac{rB}{\sqrt{A^2 + B^2}} \qquad \text{(L.3)}$$

$$c_z = -\frac{hAB}{A^2 + B^2}.$$

Recall from **Chapter 6** that a cylindroid's extreme generators are offset 45 degrees from its principal generators. The lower extreme generator of the cylindroid shown in **Figure L.1** is, therefore, expressed using the resultant twist given in **Equation (8.14)** by making its orientation vectors' components point 45 degrees away from either x or y axis by setting *A* and *B* equal to the same real number value. The lower extreme generator's $\vec{w}$ and $\vec{v}$ vectors are, therefore, expressed as

$$\vec{w}_{eg} = \begin{bmatrix} A & A & 0 \end{bmatrix}$$

$$\vec{v}_{eg} = \begin{bmatrix} Ap_1 & Ap_2 & 0 \end{bmatrix}. \qquad \text{(L.4)}$$

Recall also from **Chapter 6** and note from **Figure L.1** that a cylindroid's extreme generators are half the height, *h*, above and below the plane of the principal generators. The location vector, $\vec{c}$, for the lower extreme generator is, therefore, given as

$$\vec{c}_{eg} = \begin{bmatrix} A & A & -\frac{h}{2} \end{bmatrix}. \qquad \text{(L.5)}$$

If the pitch of the extreme generator is called p_{eg} and this value, **Equation (L.4)** and **Equation (L.5)** are plugged into **Equation (3.5)**, the resulting system of three equations may be solved to find the cylindroid's height, *h*, as

$$h = p_1 - p_2. \qquad \text{(L.6)}$$

This equation proves that the height of any cylindroid of twists is equal to the difference between the pitch values of its principal generators. If one plugs **Equation (L.6)** into **Equation (L.3)** and notes that the y-component of the location vector, c_y, in **Equation (L.3)** is equal to the x-component of the location vector, c_x, multiplied by *B*/*A* from the same equation, one must conclude that the location vector from **Equation (L.3)**, which was derived from **Equation (L.1)**, is equivalent to **Equation (8.17)**.

By finding these equations to be equal, the statement that was desired to be proven in this appendix has been proven. Since the general location vector derived from the linear combination of two orthogonal intersecting twists found in **Chapter 8** is equivalent to the general location vector for twists that lie on the surface of a cylindroid, it is known that every two orthogonal intersecting twists will produce a freedom space in the shape of a cylindroid where the two original twists are the cylindroid's principal generators. Note, however, from **Equation (L.6)** that if the pitch values of these twists are equal, the height of the cylindroid will be zero and the cylindroid will be a disk of twist lines.

In conclusion, note that the pitch value of the extreme generators may also be determined using the equations developed in this appendix. The pitch of the lower extreme generator, p_{eg}, is solved for by plugging **Equation (L.4)** and **Equation (L.5)** into **Equation (3.5)** and then plugging **Equation (L.6)** into one of the resulting equations. This pitch value is found to be

$$p_{eg} = \frac{p_1 + p_2}{2}. \qquad \text{(L.7)}$$

It may also be shown that the upper extreme generator has the same pitch value as the lower extreme generator. Note, therefore, that the upper and lower extreme generators' pitch values equal the average of the two principal generators' pitch values.

APPENDIX M:

“MATLAB Code for Drawing Case 4 Freedom Spaces”

This appendix explains and provides the MATLAB code for drawing any freedom space within Case 4 generated from any two independent twists. It provides some example plots created using this code.

The complete code consists of only four functions. These functions are provided at the end of this appendix. The main function is called “PlotLinComTwists”. In this function the user inputs two independent twist vectors, a number that determines how many twist lines will be drawn within the freedom space, and a number that determines how long each twist line should be drawn. The program then linearly combines the two independent twists given according to the user’s input specifications and plots the resulting twist lines. If the twist lines are determined to be pure rotations, they are plotted in red. If the twist lines are determined to be screws, they are plotted in green. If they are determined to be pure translations, they are plotted in black and located at the origin of the system. If enough twist lines are plotted, the spaces they occupy begin to resemble the freedom spaces found and described in **Section 8.1** of **Chapter 8**. In this way, the author was able to strategically enter twists into the program to discover, understand and visually verify every freedom space mathematically proven in **Section 8.1**.

Two examples of the utility of this program are shown below. Suppose, first, the user input these commands:

```
T1 = [1 0 0 0 0 0];
T2 = [0 1 0 0 10 0];
PlotLinComTwists(T1,T2,10,5);
```

The program then plots the freedom space shown in **Figure M.1**. This freedom space is a cylindroid of screws with a single pure rotational freedom line as one of its principal generators. The resulting freedom space is the same freedom space that was described and shown in **Figure 8.10** from **Chapter 8**.

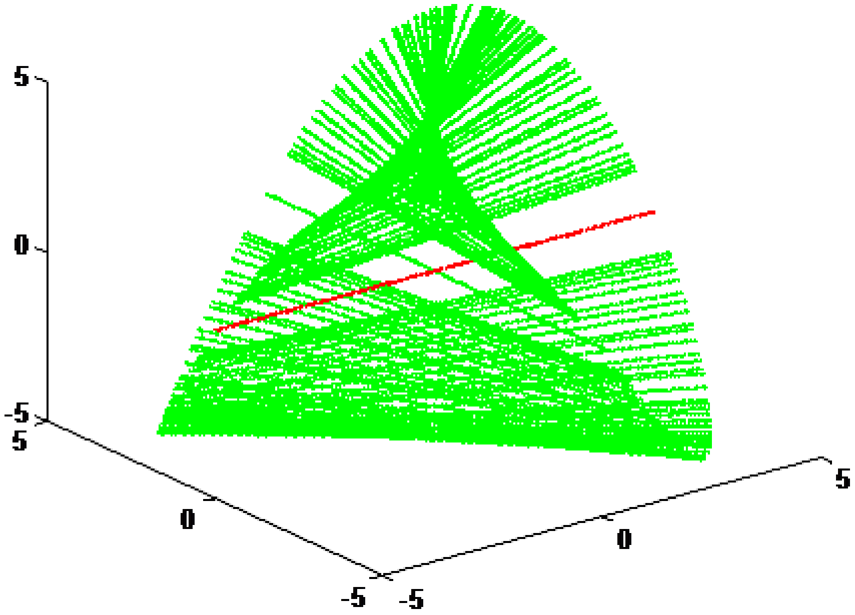


Figure M.1: Cylindroid freedom space generated using the MATLAB code of this appendix

Suppose now that the user input these commands:

```
T1 = [1 0 0 0 0 0];
T2 = [1 0 0 5 0 5];
PlotLinComTwists(T1,T2,10,5);
```

The program then plots the freedom space shown in **Figure M.2**. This freedom space is a plane of parallel screws with a single pure rotational freedom line as well as a pure translation that intersects the plane with a projected angle that is not 90 degrees. The resulting freedom space is the same freedom space that was described and shown in **Figure 8.5** from **Chapter 8**.

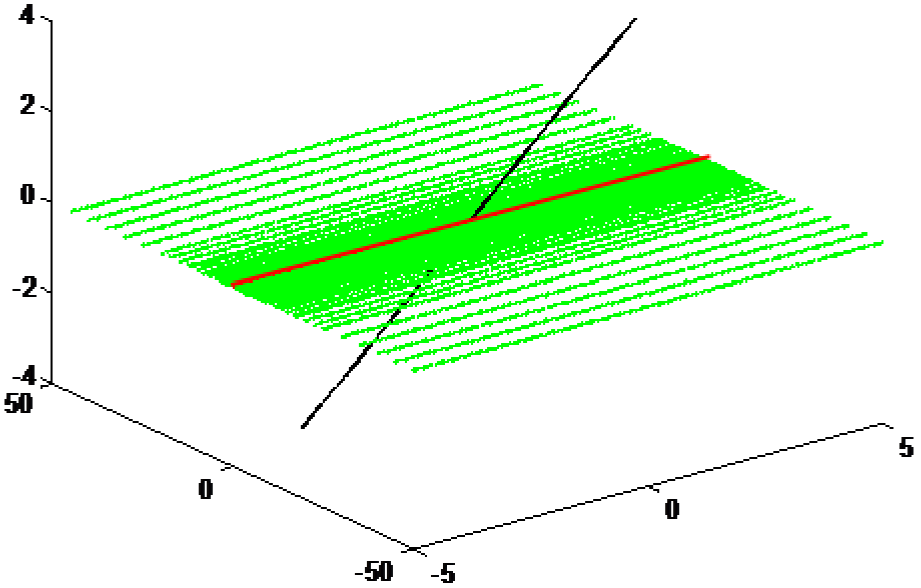


Figure M.2: Planar freedom space generated using the MATLAB code of this appendix

The four functions are provided below:

Funtion (1):

```
function PlotLinComTwists(T1,T2,num,range)
%Takes two twists and plots multiple ((2*num+1)^2) linear combinations of them to
%generate the complete freedom space.  Lines plotted have a length of 2*range
for k = -num:1:num;
   for h = -num:1:num;
      T = T1*k + T2*h;
      MainPlotTwist(T,range);
   end
end
```

Function (2):

```
function MainPlotTwist(T,range)
%Plots a twist T with a length of 2*range
[c w p] = TwistDecomp(T);
PlotTwist(c,w,p,range);
```

Function (3):

```
function [c w p] = TwistDecomp(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( T(1)==0 && T(2)==0 && T(3)==0 )
   %pure translation
   c = [0 0 0];
   w = [T(4) T(5) T(6)];  %Orientation vector defined
   p = 'Inf';
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
end
```

Function (4)

```
function PlotTwist(c,w,p,range)
%Function plots a twist
%If twist is a pure rotation it is red
%If twist is a pure translation it is black
%If twist is a screw it is green
w = w/sqrt(dot(w,w)); %Make orientation vector a unit vector
plot3(c(1),c(2),c(3))
for t = -range:0.05:range;
    hold on;
    line = c + w*t;
    if(p == 0)
        plot3(line(1),line(2),line(3),'r');
    elseif(p == 'Inf')
        plot3(line(1),line(2),line(3),'k');
    else
        plot3(line(1),line(2),line(3),'g');
    end
end
```

APPENDIX N:

“Nested Elliptical Hyperboloids as the Constraint Space of Case 4, Type 9”

This appendix proves that the constraint space of Case 4, Type 9 is an infinite number of nested elliptical hyperboloids shown in **Figure 8.87** from **Chapter 8**.

In **Section 8.3.9** it was shown that two orthogonal ribbons that are either both right-handed or both left-handed exist within the constraint space of Case 4, Type 9. These ribbons’ axes intersect at the origin and lie along the x- and y-axes. The two principal generators of the freedom space’s cylindroid of pure screws also lie along the x- and y-axes. It was established that if all the constraint lines could be found that complement these two principal generators, all the constraint lines will have been found that complement the entire freedom space and the complete constraint space of the system will, therefore, have been found. The two screw principal generators (green) and a single constraint line (blue) from each of the two orthogonal ribbons is shown in **Figure N.1**. The lengths, *a* and *b*, and the angles, α_1 and α_2, are labeled in the figure.

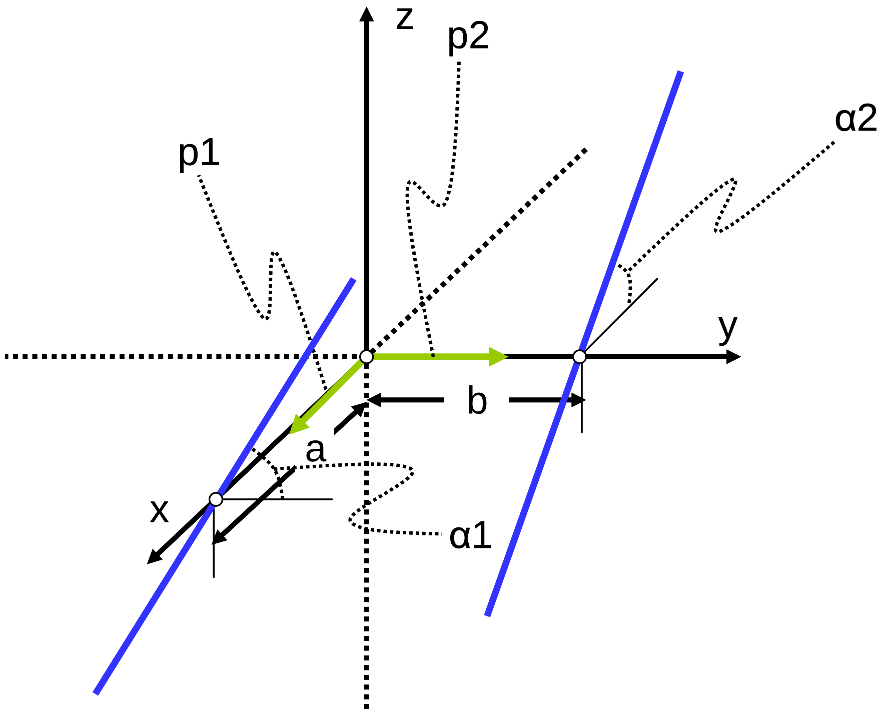


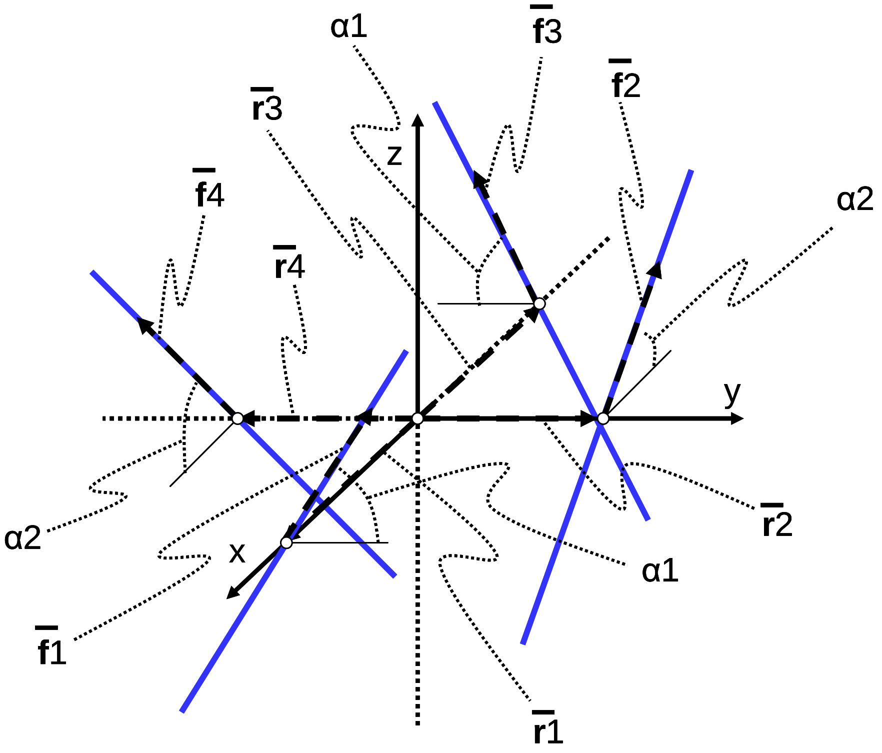
Figure N.1: Screw principal generators (green) and a single constraint line (blue) from each of the two orthogonal ribbons found within the constraint space of Case 4, Type 9 with key parameters labeled.

Note that the way **Figure N.1** is drawn, the principal generators' pitches are both positive since the constraint lines drawn belong to left-handed orthogonal ribbons.

From the parameters shown in **Figure N.1**, one can describe the two orthogonal ribbons found in **Chapter 8** using **Equation (3.13)** as

$$p_1 = b\tan\alpha_2$$
$$p_2 = a\tan\alpha_1 . \qquad \text{(N.1)}$$

If the complete constraint space of the system is an infinite number of nested elliptical hyperboloids and the constraint lines within the two orthogonal ribbons described by **Equation (N.1)** are part of this space, one should expect the linear combination of certain constraint lines from within these two orthogonal ribbons to produce constraint lines that lie on the surface of elliptical hyperboloids. More specifically, one would expect two constraint lines from each ribbon to lie on the surface of a single hyperboloid within the constraint space. The reason for this is that the elliptical hyperboloids within the constraint space are centered about the z-axis and, therefore, two and only two constraint lines the same distance away from the z-axis from each of the orthogonal ribbons will lay on the surface of a single elliptical hyperboloid. These four constraints are shown in **Figure N.2**.



$$|\bar{r}1| = |\bar{r}3| = a$$

$$|\bar{r}2| = |\bar{r}4| = b$$

Figure N.2: Four constraint lines (two from each orthogonal ribbon) that all lie on the surface of a single elliptical hyperboloid within the constraint space of Case 4, Type 9.

Using the parameters defined in **Figure N.2**, the location and orientation vectors, $\vec{r}$ and $\vec{f}$ respectively, can be defined for all four constraint lines as

$$\begin{aligned}
\vec{r}_1 &= \begin{bmatrix} a & 0 & 0 \end{bmatrix} \\
\vec{f}_1 &= \begin{bmatrix} 0 & \cos\alpha_1 & \sin\alpha_1 \end{bmatrix} \\
\vec{r}_2 &= \begin{bmatrix} 0 & b & 0 \end{bmatrix} \\
\vec{f}_2 &= \begin{bmatrix} -\cos\alpha_2 & 0 & \sin\alpha_2 \end{bmatrix} \\
\vec{r}_3 &= \begin{bmatrix} -a & 0 & 0 \end{bmatrix} \\
\vec{f}_3 &= \begin{bmatrix} 0 & -\cos\alpha_1 & \sin\alpha_1 \end{bmatrix} \\
\vec{r}_4 &= \begin{bmatrix} 0 & -b & 0 \end{bmatrix} \\
\vec{f}_4 &= \begin{bmatrix} \cos\alpha_2 & 0 & \sin\alpha_2 \end{bmatrix}.
\end{aligned} \tag{N.2}$$

If **Equation (N.2)** is used to produce a wrench matrix containing four rows of wrenches and Gaussian Elimination is then used to simplify this matrix, one finds

$$\begin{bmatrix} \cos\alpha_2 & 0 & 0 & -b\sin\alpha_2 & 0 & 0 \\ 0 & \cos\alpha_1 & 0 & 0 & -a\sin\alpha_1 & 0 \\ 0 & 0 & 2\sin\alpha_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2a\cos\alpha_1 - \dfrac{2b\sin\alpha_1}{\tan\alpha_2} \end{bmatrix}. \tag{N.3}$$

From **Chapter 7** it was established that every hyperboloid always only consists of three independent constraint lines. In order to find the correct relationship between *a*, *b*, α_1, and α_2 such that the four constraint lines from the two orthogonal ribbons will lie on the surface of the same hyperboloid, therefore, one must make sure the final pivot in the matrix from **Equation (N.3)** is zero such that one of the four constraints will always be redundant. If the last pivot, therefore, is set equal to zero and the resulting equation is rearranged, one finds

$$a\tan\alpha_2 = b\tan\alpha_1. \tag{N.4}$$

This result was found in **Appendix K** from **Equation (K.5)** and **Equation (K.10)** when an equation was being searched for that would describe the surface of an elliptical hyperboloid. In other words, as long as the parameters *a*, *b*, α_1, and α_2 for the four constraint lines from the two orthogonal ribbons found in **Section 8.3.9** satisfy **Equation (N.4)**, the linear combination of these four constraint lines will result in constraint lines that lie on the surface of an elliptical hyperboloid.

Note that if *a* equals *b*, $\tan\alpha_1$ must equal $\tan\alpha_2$ according to **Equation (N.4)** and that the principal generators of the freedom space must have equivalent pitch values according to **Equation (N.1)**. If this is the case, the system will become Case 4, Type 8.

REFERENCES

[1] Slocum, A., “FUNdaMENTALS of Design Topic Topic 3 FUNdaMENTAL Principles,” MIT 2.007 lecture notes, 2005.

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

[3] Cavanagh, R., “2. Instrumentation and Metrology for Nanocharacterization”, Report of the National Nanotechnology Initiative Workshop, January 27-29, 2004, Gaithersburg, MD.

[4] Chen S. C , Culpepper M. L., Design of a Six–axis Micro–scale Nanopositioner—μHexFlex. Prec Eng 2006;30:314–24

[5] Culpepper M. L., Anderson G., Design of a Low–cost Nano–manipulator which Utilizes a Monolithic, Spatial Compliant Mechanism. Prec Eng 2004;28:469–82.

[6] 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.

[7] Dagalakis N., Amatucci E., Kinematic Modeling of a 6 DOF Tri–stage Micro–positioner. Proceedings of the 16th Annual ASPE Meeting, November, 2001.

[8] Nomura T., and Suzuki R., Six–axis Controlled Nano–meter–order Positioning Stage for Micro–fabrication. Nanotechnology 1992;3(1):21–8.

[9] Gao P., Swei S., A Six DOF Micro–manipulator Based on Piezoelectric Translators. Nanotechnology 1999;10:447–52.

[10] McInroy J.E., Hamann J.C., Design and Control of Flexure Jointed Hexapods. IEEE Transactions in Robotics and Automation 2000;16:372–81.

[11] Zago L., Genequand P., and 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.

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

[13] 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.

[14] White, J. R., Slocum, A. H., “The Nanogate: Nanoscale Flow Control” PhD Dissertation, Massachusetts Institute of Technology, 2003

[15] Hargreaves, C., Phys. Lett. A (Netherlands) 30a, 491, 1969

[16] Howell, L. L., *Compliant Mechanisms*. John Wiley & Sons, Inc., New York, 2001

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

[18] Hale, L. C., “Principles and Techniques for Designing Precision Machines.” PhD Dissertation, Massachusetts Institute of Technology, 1999.

[19] Kennedy J. A., Howell L. L., Greenwood W., Compliant high–precision E–quintet ratcheting (CHEQR) mechanism for safety and arming devices. Prec Eng 2007;31(1):13–21.

[20] Cannon B. R., Lillian T. D., Magleby S. P., Howell L. L., Linford M. R., A compliant end–effector for microscribing. Prec Eng 2005;29(1):86–94

[21] Frecker, M. I., Ananthasuresh G. K., Nishiwaki N., Kikuchi N., Kota S., Topological Synthesis of Compliant Mechanisms Using Multi–criteria Optimization. J Mech Design 1997;119:238–45.

[22] Sigmund O., On the Design of Compliant Mechanisms Using Topology Optimization. Mech Struct Mach 1997;25(4):495–526.

[23] Joo, J., Kota, S., Kikuchi, N. “Topological Synthesis of Compliant Mechanisms Using Linear Beam Elements.” Mech. Struct. & Mach., 28(4) 245-280 (2000).

[24] Timm, R. W., “Visual-based Methods in Compliant Mechanism Optimization.” Masters Thesis, Massachusetts Institute of Technology, 2006

[25] Maxwell, J. C., “General considerations concerning scientific apparatus” in ‘The Scientific Papers of James Clerk Maxwell’ ed W. D. Niven, Dover Press, 1890. Reprinted from the Handbook to the Special Loan Collection of Scientific Apparatus, 1876.

[26] Evans, C., *Precision Engineering: an Evolutionary View*. Cranfield Press, 1989.

[27] Blanding, D. L., *Exact Constraint: Machine Design Using Kinematic Principles*. ASME Press, New York, 1999.

[28] Ball, R. S., *A Treatise on the Theory of Screws*, The University Press, Cambridge, England, 1900.

[29] Phillips, J., *Freedom in Machinery: Volume 1, Introducing Screw Theory*, Cambridge University Press, New York, NY, 1984.

[30] Phillips, J., *Freedom in Machinery: Volume 2 Screw Theory Exemplified*, Cambridge University Press, New York, NY, 1990.

[31] Kumar, V., *MEAM 520 notes: The theorems of Euler and Chasles*

[32] McCarthy, J. M., *Geometric Design of Linkages*, Springer Press, 2000.

[33] Stolfi, J., *Oriented Projective Geometry A Framework for Geometric Computations*, Acamemic Press, Inc., 1991.

[34] Coxeter, H. S. M., *Projective Geometry Second Edition*, Springer Press, 1987.

[35] Hunt, K. H., *Kinematic Geometry of Mechanisms*, Clarendon Press, Oxford, 1978.

[36] Weisstein, Eric W. "Hyperbolic Paraboloid." From *MathWorld*--A Wolfram Web Resource. http://mathworld.wolfram.com/HyperbolicParaboloid.html

[37] Weisstein, Eric W. "Hyperboloid." From *MathWorld*--A Wolfram Web Resource. http://mathworld.wolfram.com/Hyperboloid.html

[38] Weisstein, Eric W. "Plücker's Conoid." From *MathWorld*--A Wolfram Web Resource. http://mathworld.wolfram.com/PlueckersConoid.html

[39] Hilbert, D. and Cohn-Vossen, S., *Geometry and the Imagination.* New York: Chelsea, 1999.

[40] Nguyen, T., “Hyperbolic Paraboloid” Cluster 6: Mathematics.

[41] Merlet, J. P., “Singular Configurations of Parallel Manipulators and Grassmann Geometry,” *The International Journal of Robotics Research*, 8(5):45-56, 1989.

[42] Ichizli, V. M., Droba, M., Vogt, A., Tiginyanu, I. M., Hartnagel, H. L., “Peculiarities of the Scanning Tunneling Microscopy Probe on Porous Gallium Phosphide,” in Atomic Force Microscopy/Scanning Tunneling Microscopy 3, edited by Cohen S. H. and Lightbody M. L. Kluwer Academic/Plenum Publishers, 1999, pp. 158-164.

[43] Hopkins, J. B., “The Effect of Probe Tilt Angle on the Quality of Scanning Tunneling Microscope Measurements”, Massachusetts Institute of Technology, Cambridge MA, 2005.