Feature Based Analysis of Selective Limited Motion in Assemblies

by

Jeffrey D. Adams

A.P.E. Engineering Snow College, 1995

B.S. Mechanical Engineering University of Utah, 1996

SUBMITTED TO THE DEPARTMENT OF MECHANICAL ENGINEERING IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

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

FEBRUARY 1998

© 1998 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.

Signature of Author: _	Department of Mechanical Engineering January 14, 1998
Certified by:	,
	Daniel E. Whitney
	Senior Research Scientist
	Thesis Supervisor
Accepted by:	
	Ain A. Sonin
	Chairman Department Committee on Graduate Students

APR 271999 ARCHIVES

Feature Based Analysis of Selective Limited Motion in Assemblies

by

Jeffrey D. Adams

Submitted to the Department of Mechanical Engineering on January 14, 1998 in Partial Fulfillment of the Requirements for the Degree of Master of Science in Mechanical Engineering

Abstract

Motion Limit Analysis (MLA) is a tool used to support the top down design and analysis of assemblies. The purpose of MLA is to provide mathematical models of assembly features from which the ability of a feature to position one part relative to another or others in space can be calculated. Positioning comprises calculation of which of the six degrees of freedom of a rigid part are constrained, and which are capable of allowing rigid body motion relative to the parts it is assembled to, assuming those parts are rigid and do not move. If rigid body motion is allowed, the quantitative amount of motion is of interest. A user of this theory is able to obtain two major types of information about an assembly:

- 1. Knowledge of the directions and quantitative amounts of possible motions of a part that is being added to an assembly at a given assembly station via connection of a defined set of assembly features.
- 2. Knowledge of whether or not the defined feature set over-, under-, or fully-constrains the location and orientation of the part.

The ability to calculate the rigid body motions of a part being added to an assembly is important for using in-process adjustments during assembly to precisely establish key assembly dimensions. Also, locational overconstraint of parts can lead to assemblability problems or require deformation of parts in order to complete assembly.

MLA uses the mathematics of screw theory to model the ability of mechanical assembly features to allow or constrain rigid body motion in six degrees of freedom. Seventeen feature types have been modeled.

The MLA software is a part of a suite of software programs used to do assembly analysis. It is used to choose the assembly features that realize the interconnections between parts and it performs constraint calculations about the properties of the chosen set of features. These results provide information to the user and input to other analysis programs.

Thesis Supervisor: Daniel E. Whitney

Title: Senior Research Scientist

Biographical Note

Jeffrey D. Adams was born in Manti, Utah in 1971. He graduated from Manti High School in 1989 as Valedictorian. Jeff attended Snow College in Ephraim, Utah on a Bertleson Scholarship and graduated in 1995 with an Associates of Pre-Engineering Degree. He went on to the University of Utah in Salt Lake City, Utah as a Kennecott Scholar and a Department Scholar in the Department of Mechanical Engineering. He graduated 2 of 493 in 1996 with a Bachelors of Mechanical Engineering Degree. Jeff is now attending the Massachusetts Institute of Technology as a Tau Beta Pi Fellow. He will graduate in February, 1998 with a Masters of Science in Mechanical Engineering degree.

Acknowledgements

As I submit my Masters thesis I wish to thank the many people who have helped and supported me throughout the time I was working on this degree. First to my wife Anna, who has been so supportive in taking care of our home life and of our son Scott so that I might have the time required to complete this degree. Next to Dr. Daniel E. Whitney, Ph.D., who served as my research advisor. His advice, leadership, and suggestions have been invaluable as I struggled to find a suitable thesis topic and bring it to fruition. Also to Dr. Ramakrishna Mantripragada, Ph.D. who was my research partner throughout my time here at MIT. His foresight and suggestions helped me to develop Motion Limit Analysis to the point it is today. Thanks to Tim Cunningham for providing me data on the example of Chapter 8 and giving support and suggestions. A special thanks is due to Dr. Ranjit Konkar who developed many of the basic algorithms used in Motion Limit Analysis. His explanations helped me realize the potential applications of the algorithms to assembly analysis. I am also grateful for his providing me documentation of his As Dr. Whitney often says, "Assembly is more than just putting parts together." I have learned through experience that this statement is true. It is one of the most valuable lessons I have learned at MIT, one which I hope to put to good use in my career as an engineer.

The research described in this document was supported by the U.S. Airforce Grant F33615-94-C-4429 under the MIT Fast and Flexible Manufacturing Project.

Contents

ABSTRACT	3
BIOGRAPHICAL NOTE	4
ACKNOWLEDGEMENTS	
CONTENTS	5
LIST OF FIGURES AND TABLES	9
GLOSSARY OF ACRONYMS	11
CHAPTER 1 INTRODUCTION	13
1.1 MOTIVATION	14
1.1.1 Datum Flow Chains	14
1.1.2 State Transition Models of Mechanical Assembly Processes	15
1.2 Preview	16
CHAPTER 2 ASSEMBLY ORIENTED DESIGN	19
2.1 AOD OVERVIEW	19
2.1.1 Benefits and Limitations of the Approach	21
2.2 DATUM FLOW CHAINS	22
2.3 STATE TRANSITION MODELS OF ASSEMBLY	27
2.4 ASSEMBLY DESIGN USING AOD AND MLA	29
2.5 SUMMARY	30
CHAPTER 3 PRIOR WORK	31
3.1 ASSEMBLY ANALYSIS	31
3.2 FEATURE MODELS AND FEATURE BASED DESIGN	32
3.3 SCREW THEORY KINEMATICS	34
3.4 ALTERNATE APPROACHES TO APPLIED KINEMATIC ANALYSIS	35
3.5 KINEMATIC MODELS OF FEATURES	36
3.6 Summary	37
CHAPTER 4 SCREW THEORY	39

4.1 DEFINITION OF TERMS	39
4.2 APPLICATION OF SCREW THEORY IN DESCRIBING KINEMATIC JOINTS	43
4.3 INTERPRETATION OF RESULTANT TWISTMATRICES AND WRENCHMATRICES	44
4.4 Summary	46
CHAPTER 5 FEATURE MODELS	47
5.1 Variation Absorption at Feature Sites	47
5.1.1 Explicit Variation Absorption	48
5.1.2 Implicit Variation Absorption	50
5.1.3 Variation Absorption Capabilities of Features	52
5.2 TWISTMATRIX REPRESENTATION OF FEATURES	52
5.2.1 Role of 4x4 Transform Matrices	52
5.2.2 Nomenclature and Conventions	54
5.2.3 Freedom Zones and Absorption Zones	55
5.2.4 Single Mating Features	
5.3 SUMMARY	
CHAPTER 6 MOTION LIMIT ANALYSIS	69
6.1 Scope of Application	70
6.2 MOTION LIMIT VECTORS	70
6.3 DEFINITION OF ALGORITHMS	71
6.3.1 Prose Description of the Twistmatrix Intersection Algorithm	71
6.3.2 Mathematical Description of the Twistmatrix Intersection Algorithm	71
6.3.3 Interpretation of the Resultant Twistmatrix	72
6.3.3.1 Prose Description of the Point Algorithm	73
6.3.3.2 Mathematical Description of the Point Algorithm	
6.3.3.3 Prose Description of the Row ID Algorithm	74
6.3.4 Prose Description of the MLA Algorithm and Geometric Reasoning	
6.3.4.1 Rotational DOFs	75
6.3.4.1.1 Search For Axis Which Allows the Maximum Possible Rotation	78
6.3.4.1.2 Search Within A Linear Space	
6.3.4.1.3 Limits on the Scalar Parameter t	
6.3.4.1.4 Search Within A Planar Space	
6.3.4.2 Translational DOFs	
6.3.5 Mathematical Implementation of MLA Algorithm	
6.3.5.1 Rotational DOF's	
6.3.5.1.1 Determination of the Space of Possible Axes	
6.3.5.1.2 Finding θmax For a Given Rotational Axis	
6.3.5.1.3 Search Within A Linear Space	86

6.3.5.1.4 Search Within A Planar Space	87
6.3.5.2 Translational DOF's	87
6.3.5.3 Calculating the Components of the Motion Limit Vector	87
6.3.6 Redundant Constraint Analysis	
6.3.7 Prose Description of the Constraint Analysis Algorithm	89
6.3.8 Mathematical Description of the Constraint Analysis Algorithm	89
6.3.9 Interpretation of a Resultant Wrenchmatrix	90
6.4 COMMENTARY ON ANALYSIS RESULTS	90
6.5 EXAMPLES	91
6.5.1 Example 1: Hole and Slot Combination	91
6.5.2 Example 2: Fixturing of Sheet Metal Parts	95
6.6 Summary	98
CHAPTER 7 MLA SOFTWARE PROGRAM	99
7.1 USING THE MLA SOFTWARE	101
7.1.1 Example Assembly	103
7.1.2 Inputs	105
7.1.3 Assumptions	107
7.1.4 Calculations	108
7.1.5 Outputs	109
7.2 MODELING AND ANALYSIS CONSIDERATIONS	112
7.3 CLASS STRUCTURE	112
7.3.1 Part Class	113
7.3.2 Feature Class	
7.3.3 TwistMat Class	114
7.3.4 Transform Class	115
7.3.5 Array Class	115
7.4 LOGICAL PROGRAM FLOW	115
7.5 SUMMARY	118
CHAPTER 8 DESIGN AND ANALYSIS EXAMPLE	119
8.1 NEED	119
8.2 HORIZONTAL STABILIZER ASSEMBLY	120
8.3 ASSEMBLY ORIENTED DESIGN APPROACH	121
8.3.1 KC Identification	121
8.3.2 DFC Design	121
8.3.3 Variation Absorption Strategy and Feature Design	122
8.3.4 MLA Analysis	126

8.4 CONCLUSION	128
8.5 SUMMARY	129
CHAPTER 9 CONCLUSIONS	131
9.1 CONCLUSIONS	131
9.2 FUTURE WORK	131
BIBLIOGRAPHY / REFERENCES	133
APPENDIX A SOFTWARE DOCUMENTATION	137
APPENDIX B ANALYSIS TRANSCRIPTS	139
Chapter 7 Analysis Example	139
CHAPTER 8 ANALYSIS EXAMPLE	148

List of Figures and Tables

FIGURE 2.1 ASSEMBLY ORIENTED DESIGN FLOWCHART	20
Figure 2.2 Two Candidate DFCs for a Simple 3-Part Assembly	25
FIGURE 4.1 DIAGRAM OF TWIST INTERPRETATION	40
FIGURE 4.2 THE WRENCH RESULTING FROM AN APPLIED FORCE SYSTEM	41
FIGURE 5.1 Skin Panel Assembly Showing Explicit Variation Absorption	49
FIGURE 5.2 DFC FOR SKIN PANEL ASSEMBLY	49
FIGURE 5.3 SHEET METAL ASSEMBLY SHOWING IMPLICIT VARIATION ABSORPTION	51
FIGURE 5.4 RELATIONSHIP BETWEEN PART AND FEATURE-LEVEL COORDINATE FRAMES	54
FIGURE 6.1 DIAGRAM OF THE INTERPRETATION OF A TWIST	73
FIGURE 6.2 ILLUSTRATION OF ROTATIONAL MOTION ALLOWED BY COMBINATIONS OF FEATURES	76
FIGURE 6.3 GEOMETRY FOR CALCULATION OF R	77
FIGURE 6.4 CALCULATION OF LIMITS ON THE SCALAR PARAMETER T	81
FIGURE 6.5 THETA VS. T FOR TWO SHAPES OF FREEDOM ZONES	82
FIGURE 6.6 HOLE-SLOT COMBINATION ASSEMBLY	92
FIGURE 6.7 DFC FOR HOLE-SLOT COMBINATION ASSEMBLY	92
FIGURE 6.8 THREE-PART SHEET METAL ASSEMBLY	96
Figure 6.9 Possible Fixture Redesign to Eliminate Overconstraint	98
FIGURE 7.1 FLOWCHART OF SOFTWARE USED IN AOD	100
FIGURE 7.2 FILENAMES OF INFORMATION SHARED WITH MLA	101
FIGURE 7.3 SCHEMATIC OF EXAMPLE ASSEMBLY	104
FIGURE 7.4 DFC FOR EXAMPLE ASSEMBLY	104
FIGURE 7.5 SCHEMATIC OF PART 'RIB'	105
FIGURE 7.6 DEFINITION OF PART LOCATIONS	106
Figure 7.7 Feature Selection and Definition	107
FIGURE 7.8 CALCULATION FLOW FOR MATES AND CONTACTS ANALYSIS	108
FIGURE 7.9 SAMPLE OUTPUT FOR FULLY CONSTRAINED PARTS	109
FIGURE 7.10 SAMPLE OUTPUT FOR AN UNDER-CONSTRAINED PART	110
FIGURE 7.11 SCHEMATIC OF A LINEAR SPACE OF SOLUTIONS	111
FIGURE 7.12 FLOWCHART DIAGRAMING CALCULATION ROUTINES	116
FIGURE 7.13 FLOWCHART OF LINEAR AND PLANAR SOLUTION SPACE SEARCHES	118
FIGURE 8.1 EXPLODED VIEW OF THE MAIN TORQUE BOX OF A HORIZONTAL STABILIZER	120
FIGURE 8.2 BOTTOM VIEW OF AN UPPER SKIN SUBASSEMBLY [40]	120
FIGURE 8.3 ASSEMBLY KEY CHARACTERISTICS FOR THE UPPER SKIN ASSEMBLY [41]	121

FIGURE 8.4 DATUM FLOW CHAIN FOR THE UPPER SKIN ASSEMBLY	122
FIGURE 8.5 POSSIBLE LOCATIONS FOR VARIATION ABSORPTION [41]	123
FIGURE 8.6 ASSEMBLY FEATURE DESIGN FOR THE UPPER SKIN ASSEMBLY	125
FIGURE 8.7 SIZE LIMITATIONS ON TEMPORARY SLOTTED-HOLE FEATURES	128
TABLE A. 1 LISTING OF MLA SOFTWARE FILES	138

Glossary of Acronyms

AOD Assembly Oriented Design

CAD Computer Aided Design

DFC Datum Flow Chain

dof Degree of Freedom

FCS Feature Coordinate System

GCS Global Coordinate System

GD&T Geometric Dimensioning and Tolerancing

ISA Instantaneous Screw Axis

KC Key Characteristic

MLA Motion Limit Analysis

MLV Motion Limit Vector

PCS Part Coordinate System

Chapter 1 Introduction

Motion Limit Analysis (MLA) is a tool used to support the top down design and analysis of assemblies. The purpose of MLA is to provide mathematical models of assembly features from which the ability of a feature to position one part relative to another in space can be calculated. Positioning comprises calculation of which of the six degrees of freedom of a rigid body are constrained, and which are capable of allowing rigid body motion. If rigid body motion is allowed, the quantitative amount of motion is of interest in order to calculate possible adjustments that can be made to part positions during the act of assembly. Adjusting part positions during assembly can allow a manufacturer to counteract the effects of manufacturing and thermal variations in the parts, which may otherwise compromise the dimensional accuracy of the assembly. A user of this theory is able to obtain two major types of information about an assembly:

- Knowledge of the directions and quantitative amounts of possible part motions of a
 part that is being added to an assembly at a given assembly station via connection of a
 defined set of assembly features.
- 2. Knowledge of whether or not the defined feature set over-, under-, or fully-constrains the location and orientation of the part.

MLA is part of an assembly design methodology called 'assembly oriented design' (AOD) [1]. AOD uses a number of assembly analysis tools in order to help an assembly designer plan out and analyze candidate assembly schemes prior to having detailed

knowledge of the geometry of the parts involved. The activities supported by the AOD tools include dimensional control planning, selection of assembly features, analysis of part constraint and motion freedom (described in this thesis), assembly sequence analysis, and tolerance propagation. In this way, many assembly schemes can be inexpensively proven for their ability to deliver the important characteristics of the final product. This ability is important for a product design group to improve quality while shortening the product development time. The tools used in AOD, including MLA, are best applied to product design, but can also be used to analyze an existing assembly and suggest areas for improvement.

1.1 Motivation

This thesis was motivated by the need to provide analytical and computational support to the ideas of Datum Flow Chains and State Transition Models of Mechanical Assembly Processes which are being developed by researchers at MIT. These are two of the major elements of the AOD methodology. This section will give a brief description of the two ideas in the context of how this work supports them. For detailed descriptions of the MIT work, the reader is referred to references [1] through [4].

1.1.1 Datum Flow Chains

The datum flow chain (DFC) graph provides a method by which an assembly designer can lay out the locational and dimensional constraint plan inherent in an assembly design. The important dimensional, functional, and performance attributes of an assembly are embodied in a set of Key Characteristics (KCs) that the assembly must deliver in order to satisfy the customer of the product. The datum flow chain represents a dimensional constraint and location plan that maps dimensional relationships between parts in an assembly. The DFC is designed to fulfill the requirements of the set of KCs by positioning each part in an assembly with the required accuracy. A DFC is "a directed acyclic (a graph with no cycles) graphical representation of an assembly with nodes representing the parts and arcs representing mates between them." [2] The connections

(mates) between parts represented by an arc in a DFC are physically realized by a set of mating features that connect the parts. Each DFC link is thus realized by specifying one or more mating features that satisfy the conditions of positioning and part immobilization associated with that link. The first motivation of this work is therefore to provide a library of mating features which are mathematically defined as to their ability to accurately position and immobilize the two parts that are connected by each.

1.1.2 State Transition Models of Mechanical Assembly Processes

In a state transition model of assembly, the assembly process is modeled as a multi-stage linear dynamic system. At any stage during the assembly process, the state of the system is given by a 6×1 vector that contains the actual deviation of the position and orientation of the part that is being added to the assembly at that stage, with respect to the nominal position and orientation of the part. Depending on the nature of the assembly, adjustments of the part's position and orientation may be allowed at some stages in the assembly process in an attempt to return the state of the system to zero, i.e. no deviation. The ability to do this depends on the assembly features that are chosen to connect the parts in the assembly together. Therefore, the second motivation of this work is to provide models of assembly features that characterize them as to their ability to allow position and orientation adjustments.

The state transition model can be used for two types of analyses. First is tolerance analysis of assemblies that allow the types of adjustments mentioned above. The model is used to calculate the statistical distribution of part positions and compare them to the nominal positions that would result if there were no variations present. The second type of analysis addresses the reverse problem where an assembly decomposition exists, and the designer wishes to synthesize a set of assembly features that will allow an optimal set of in-process adjustments. This is accomplished using an optimal control algorithm that calculates the desired amount of adjustment at each stage in the assembly process. The designer must then design an assembly feature set that will produce the needed adjustment freedoms. Both of these types of analysis require calculation of the

quantitative amount of positional and orientational adjustment that a set of assembly features will allow. The third motivation of this thesis is then to provide this calculation ability.

1.2 Preview

Chapter 2 presents the ideas of assembly oriented design (AOD) of which motion limit analysis is a part. The theory of the method is introduced and illustrated with a flowchart. The benefits and limitations of the approach are then outlined. Some of the major theoretical components of the methodology are presented and their connection to MLA is explained. Finally an overview of using MLA with the AOD framework is given.

Chapter 3 presents prior research work that has been done in many research areas that relate to the topic of motion limit analysis. A brief description is given of each researcher's work along with an explanation of the similarities or differences between that work and MLA.

Chapter 4 provides definitions of the terms used in classical screw theory and also explains a standard representation of screws in matrix form. Set operations used for manipulating groups of screws are then defined. It is then shown briefly how screw theory can be applied to describe kinematic joints. Lastly, the physical interpretation of a resultant twistmatrix or wrenchmatrix calculation is given.

Chapter 5 extends the discussion of how screw theory can be applied to kinematics by giving a twistmatrix mathematical representation for seventeen kinds of assembly features which span the space of possible kinematic motions that can be allowed by physical features. First, the variation absorption capabilities of assembly features are discussed. Then the role of 4×4 homogeneous transform matrices in describing feature locations on parts is described. The method in which 4×4 transforms are used to calculate twistmatrices for individual features is then presented. Following the definition of some nomenclature and conventions, models for 17 types of assembly mating features, that span the set of possible methods of degree of freedom constraint between two

distinct rigid parts, are presented. The models are classified by the number of independent relative motions they would allow between two distinct parts connected by each. The definitions of a freedom zone and absorption zone are also given.

Chapter 6 describes the theory behind the operation of MLA. An overview of MLA is given along with a discussion of its scope of application. The main algorithms used in MLA are then presented in both prose and mathematical form. Two simple examples are then worked in detail in order to illustrate the method.

Chapter 7 documents the operation of the MLA software program. The nature of the program and the environment it operates in are first explained. Step by step instructions on how to use the program are then given and illustrated with an example. The assumptions used in the calculations are also elucidated. Next an explanation of the data structures used to store part and feature data are given. The logical flow of the calculation routine finishes the chapter.

Chapter 8 presents a design and analysis example showing how MLA analysis is used within the assembly oriented design framework and demonstrating the type of results that are obtained. The example assembly is an upper skin assembly, which is part of an aircraft horizontal stabilizer assembly. The assembly oriented design approach is used to define and analyze this assembly with particular attention to the MLA analysis component. The results are presented and discussed.

Chapter 2 Assembly Oriented Design

This chapter will explain the concepts of assembly oriented design (AOD) of which MLA is a part. The chapter will illustrate the methods of AOD, give examples of its use, and describe how MLA contributes to the theory and fits in to the implementation.

2.1 AOD Overview

AOD uses a number of assembly analysis tools in order to help an assembly designer plan out and analyze candidate assembly schemes prior to having detailed knowledge of the geometry of the parts involved. In this way, many assembly schemes can be inexpensively tested for their ability to deliver the important characteristics of the final product. This ability is important for a product design group to improve quality while shortening the product development time. The tools used in AOD are best applied to product design, but can also be used to analyze an existing assembly and suggest areas for improvement. The flowchart in Figure 2.1 below is a representation of the current state of knowledge regarding assembly oriented design. More detail about the elements in Figure 2.1 may be found in Chapter 7.

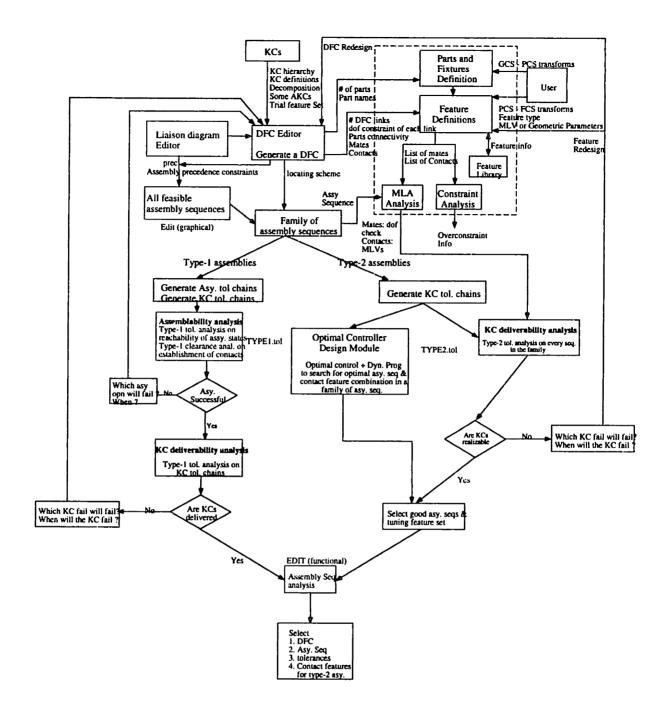


Figure 2.1 Assembly Oriented Design Flowchart

Motion Limit Analysis is in the Dashed-Line Box

Assembly oriented design is a top-down design methodology used to perform assembly design and analysis at early stages of the product development process. The flowchart shows that the top level input to the method is a set of Key Characteristics (KCs). These KCs define the important aspects of the product that are crucial to customer satisfaction. The DFC editor is then used to design a DFC to meet the requirements of the KCs.

Designing a DFC necessitates a decomposition of the assembly into distinct parts. Only the geometrical locations of interfaces between parts need be specified in detail. This is done using the parts and fixtures definition module of MLA. The DFC along with a liaison diagram represent the connections between parts in the assembly. These two diagrams provide inputs to assembly sequence generation software that generates the set of all feasible assembly sequences. In order for the user to answer the questions in the assembly sequence generator, s/he must have some rough idea of the way the parts fit together in order to identify escape directions for the parts. Thus an approximate trial set of assembly features must be present before the final set is chosen and analyzed using MLA. Assembly sequence is also an input to MLA, thus there is some circularity involved with establishing a trial set of features, generating assembly sequences, and specifying and analyzing a more exact feature set. This loop may have to be iterated a few times before the set of features and sequences satisfies the designer. The sequences are analyzed by following modules to determine which are best able to support the logic of the DFC in achieving the requirements set forth by the KCs. The DFC also provides inputs to MLA such that assembly feature sets can be chosen. Once a feature set is established, the models are passed to tolerance analysis algorithms that analyze the performance of the chosen design. Iteration may be required to find a suitable combination of DFC, assembly sequence, tolerances, and contact features that will deliver the set of KCs. The modules outlined by the dashed box belong to motion limit analysis, and are the main topic of this thesis. Each arrow crossing the dashed lines indicates information that is being shared between MLA and the other modules of the methodology.

2.1.1 Benefits and Limitations of the Approach

The overriding benefit of the AOD approach is that it is a structured top-down design approach, which includes supporting analysis methods and tools. This gives the designer a system view of the product and a way to trace design and production problems up or down the assembly levels in order to find their root cause. Because detailed geometry is not needed in order to perform the various analyses, the method can be applied early in

the design process and still yield accurate and useful results. The system thus allows designers to consider assembly related issues in the first stages of design, and fix problems and inconsistencies before they rear their ugly heads during prototyping, rampup, production, or service.

The biggest limitation of the approach as currently developed is that it assumes that all bodies in the assembly are rigid. Extending the ability of the approach to analyze compliant parts would greatly enhance the applicability of the method.

The following sections will give more detailed descriptions of some of the modules shown in the flowchart of Figure 2.1 in order to provide context for the following discussions of MLA.

2.2 Datum Flow Chains

The concept of Datum Flow Chains (DFCs) was introduced in Chapter 1. The concepts of DFC and MLA are intimately related. Thus, in order to provide a better understanding of the context in which the work of this thesis lies, a more complete description of the DFC is presented here. A full exposition of the subject is given in [4].

Most modern CAD systems are "part-centric" in that they are adept at helping designers create detailed geometric models of single parts on the computer screen. The datuming scheme used to create the part and features on it is normally determined by finding the scheme that allows the part to be most easily created geometrically. Often these geometric datums do not correspond to the assembly datums of the assembly into which the part fits. Careful assembly design involves coordinating assembly-level and part-level datums in a top-down way. Thus there is a need for a computer-based tool which allows designers to implement a top-down design process. That is, to start at an assembly or final product level, define datuming schemes consistent with the important aspects of that product or assembly, and coordinate those datums down the assembly tree to the part level. The DFC provides the method to do this.

A DFC is "a directed acyclic (a graph with no cycles) graphical representation of an assembly with nodes representing the parts and arcs representing mates between them." [2] DFCs can contain two types of arcs. A "mate", represented as a solid directed line, represents a connection between parts in which the part at the tail of the arrow defines the location of the part at the head of the arrow by constraining all or some of its degrees of freedom (dof). A number is associated with each mate that indicates how many dofs are constrained by that mate. Thus, the vectorial sum of the constrained dofs associated with incoming arrows to a part should always be six in order that the mates constrain all six degrees of freedom of the part. MLA is capable of carrying out this check. The second type of arc is termed a "contact". A contact connection between parts is redundant location-wise and is normally used to add additional support or strength to an assembly after the parts have been located via the mates. A DFC can be drawn with or without contacts shown. When contacts do appear, they are shown as dashed lines with no direction. Therefore, assembly is the process of using the mates between parts to define their locations, and then, if desired, fastening contacts to provide additional strength and support. The choice of which joints between parts should be mates and which should be contacts is an integral part of assembly design using the DFC. The locating scheme defined by the chain of mates in a DFC is the datuming scheme and tolerance chain for the assembly.

Assembly design using the DFC consists of six steps:

- 1. Defining the Key Characteristics (KCs) for the assembly. A KC can be a dimension that must be tightly controlled, a weight or strength specification, a performance requirement, or some other important characteristic of the product which, if not delivered, will affect the satisfaction of the customer of the product.
- 2. Partitioning the assembly into discrete parts and designing a DFC to establish a datuming and locating scheme for the assembly that is consistent with the KCs that were chosen in step 1.
- 3. Establishing a trial assembly feature set in sufficient detail to establish part escape direction which are required for generating assembly sequences.
- 4. Geometrically defining the locations of parts and of interfaces between parts.

- 5. Choosing assembly features that connect the parts across those interfaces that realize the locating scheme dictated by the DFC and determine the direction and amount of allowed motions or variation absorptions. This includes:
 - a) Deciding which of the 6 dofs of the part being located will be constrained by each feature.
 - b) Choosing geometric parameters in order to size the feature or place limits on the motions allowed by it.
 - c) Placing and orienting each feature.
- 5. Analyzing the DFC and feature choices to see how well they deliver the KCs.

Based on the results of the analysis, the designer may choose to iterate through some or all of the preceding steps. This section of this thesis is concerned with step 2. MLA is used to accomplish steps 3, 4, and some of 5. These concepts will be fully developed in later chapters of the thesis. Step 1, KC definition, will not be explained here, but the reader is referred to [35-37] for a full development.

The concept of designing a DFC (step 2) is best illustrated using an example. A simple assembly with one degree of adjustment freedom and two DFC representations for it are shown in Figure 2.2 below.

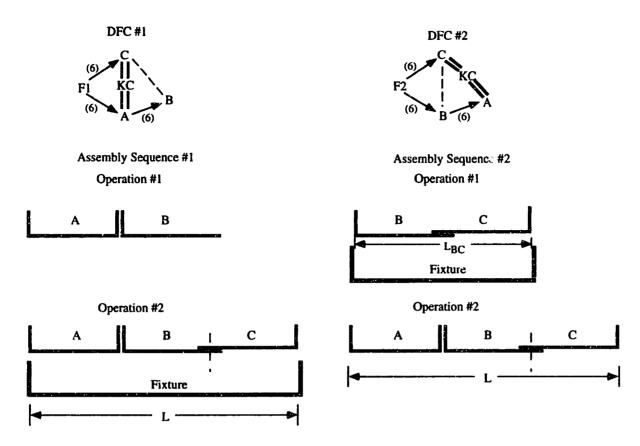


Figure 2.2 Two Candidate DFCs for a Simple 3-Part Assembly

The two DFCs represent different schemes for locating the parts within this assembly, and also define different assembly sequences. The KC for this assembly is the dimension between parts A and C, which is the overall length (L) of the assembly. Note that because this figure is shown in 2-D, it is assumed that the parts butt up against a surface which is parallel to the plane of the page which prohibits translations in to or out of the page and also constrains any associated rotations.

For DFC#1, the mate between parts A and B is established first. Thus A locates B and constrains its 6 degrees of freedom. Next a fixture (F) is used to locate parts A and C and constrain all of their dofs. Therefore the fixture establishes the KC. The slip-plane overlap connection between parts B and C is a contact. It is fastened last and used to lock in the dimensional relationship between part C and the subassembly A-B. This DFC and assembly sequence deliver the dimension L despite any length variations that may exist in the parts. This is so because of the adjustability provided by the slip plane feature between parts B and C. Within limits defined by the amount of overlap that is allowed

between B and C, the parts may have length variations without compromising the dimension L.

For DFC#2, a fixture (F) is used to set the locations of parts B and C and the joint between them is fastened. Part A is then located by part B. Length variations on parts B and C are absorbed in the slip plane contact between them. Thus L_{BC} can be set exactly. The variation in the overall length L will then be equal to the variation of the length of part A.

Both DFCs employ a fixture and contain three mates and one contact. Also in each case the contact is used to absorb variations in the parts. The difference is in where and how the fixture is used, and when the contact is established. DFC#1 was clearly designed more in line with the KC that was defined for the assembly. This can be seen qualitatively. A quantitative analysis of the two situations was done in [2] with the following results. The standard deviation (SD) associated with length L as established using DFC#1 was 0.564. This corresponds to L being within the tolerances defined in [2] 97.36% of the time. Using DFC#2 resulted in a SD of 0.8183 with a KC delivery probability of 75.08%. [3] This example will be revisited in Chapter 6 where MLA analysis will be applied to reveal more information about the assembly.

Note that the DFC is not unique. For each assembly there can be many DFCs. Thus a DFC must be carefully designed according to the demands of the KCs. It is not a property of the assembly and cannot be simply "written down" by inspection.

One aspect of the relationship between the DFC and MLA can be understood in the context of this example by realizing that the range of overlaps allowed by the contact feature between parts B and C will affect how much part length variation can be absorbed by that feature. Using the location of that feature on parts B and C, and some geometric parameters defining the zone of allowable overlaps, MLA will calculate limits on part length variations that can be absorbed at the feature.

2.3 State Transition Models of Assembly

In a state transition model of assembly, the assembly process is modeled as a multi-stage linear dynamic system. A 4×4 homogeneous transform [5] is used to define the nominal position and orientation of each part in an assembly. At any stage in the assembly process, the state of the assembly is defined by a 6×1 vector that contains the total deviation in position and orientation of each of the six degrees of freedom of the last part that was added to the assembly. In other words, it contains the difference between a part's actual position and orientation and the nominal position and orientation given by the 4×4 transform.

Two distinct types of assemblies have been defined, called Type-1 and Type-2 [3]. For a Type-1 assembly, the parts are rigid and all of the assembly features on each part are mates as defined above and are created during fabrication. The act of assembly consists of connecting the parts together by mating the assembly features together. Due to the fact that the mating features in this type of assembly are fabricated prior to assembly, the variation in the final assembly dimension is just the stackup of the variation associated with each part that contributes to a given dimension. Typically, no in-process adjustment of part locations is allowed during assembly of Type-1 assemblies.

Type-2 assemblies are typified by aircraft fuselage and automotive body parts, where some or all of the mating features are created during the assembly process. This in-situ fabrication of assembly features results in a set of assembly features whose exact dimensional locations are unique to each assembly. Often, fixtures are used to pid parts relative to each other in the desired orientation where drilling, welding, riveting, or some other process can be performed to fasten the parts together. This type of assembly process allows the opportunity for making in-process adjustments of part positions in order to counteract the effects of part variations and deliver key dimensions more precisely. If in-process adjustments are desired, the designer must carefully specify a set of features that will reliably locate the part while allowing adjustment motion to occur in the direction(s) of interest. In this sense, it would be helpful to a designer if s/he could design a feature set on a part and quickly evaluate whether or not that set provides

positioning and adjustability as intended. MLA provides this capability with automated evaluation of a set of mating features as to their constraint and adjustment properties.

The state transition model of a Type-2 assembly is given by the following equation with associated definitions [2]:

$$\tilde{X}(k+1) = A(k)\tilde{X}(k) + B(k)\tilde{U}(k) + F(k)\tilde{w}(k) \tag{1}$$

where

 $\tilde{w}(k)$: 6×1 vector describing the variation associated with the part being assembled at the k^{th} assembly station, expressed in local part coordinates

 $\tilde{X}(k+1)$: 6×1 vector describing the total variation accumulated after the k^{th} assembly station, defined and measured in the base coordinate frame for the KC chain.

 $\tilde{U}(k)$: 6×1 vector describing the absorption zone of the contact feature as the assembly control input,

and

A(k): Identity matrix

F(k): 6×6 matrix that transforms the variation associated with the incoming part at the kth assembly station from part k's coordinate frame to the base coordinate frame of the KC chain.

B(k): 6×6 matrix that transforms $\tilde{U}(k)$ from the coordinates of part k to the base coordinates for the KC chain.

The values in the vector $\tilde{U}(k)$ are determined by the set of mating features used to locate the part that is added at the k^{th} assembly station. $\tilde{U}(k)$ defines the limits of motion in 6 dof for the part added at the k^{th} station. Calculating $\tilde{U}(k)$ for each part in an assembly based on a chosen feature set is another capability of MLA.

2.4 Assembly Design Using AOD and MLA

This section will give some high level discussion of using MLA within the AOD methodology. The past two sections described two of the important concepts of AOD and showed how MLA supports them analytically. Thus one can see that MLA is used to choose sets of assembly features and place them on parts to realize the connections in an assembly.

In choosing a feature or features to realize a DFC link, the designer must first consider which of the six dofs that link will constrain. A feature or set of features is then chosen and oriented to constrain only the desired dofs. Remember that dof constraint is a vector quantity. The number of dofs constrained by a feature set is a vector sum of the dofs constrained by each. Chapter 5 will present models of all the feature choices in MLA, shown in local feature coordinates. The designer must remember to orient the feature correctly in order to constrain the desired part coordinate dofs. Features are chosen to realize both mate connections and contact connections.

The nominal objective of the mate connections is to constrain all six dofs of the part toward which the DFC arrows are pointing. If less that six dofs are constrained the part will be able to undergo rigid body motion. This case may be desired for a kinematic assembly, or if adjustments are to be made to the part's position during assembly. The latter is called explicit variation absorption and will be explained in more detail in Chapter 5.

Contact connections between parts exist nominally to provide additional strength and support to the assembly after the mates have determined part locations. Due to manufacturing variations, contact features can never be placed exactly. Thus contact features are usually chosen to constrain only the dofs required to provide strength and leave others unconstrained in order to reduce the consequences of misplacing the feature. This is called implicit variation absorption and will also be expounded in Chapter 5.

After features are chosen for all mates and contacts in the DFC, MLA performs an analysis and provides the user information as to the ability of the chosen feature set to accomplish the desired constraint, and calculates limits on explicit and implicit variation absorption. Each of these topics will be discussed fully in later chapters.

2.5 Summary

This chapter explained the method of assembly oriented design. The theory of the method was introduced and illustrated with a flowchart. The benefits and limitations of the approach were outlined. Some of the major theoretical components of the methodology were presented and their connection to MLA was explained. Finally an overview of using MLA with the AOD framework was given.

Chapter 3 Prior Work

This chapter will present prior research work done in several areas relating to the topic of this thesis. The descriptions are in no way meant to be a comprehensive coverage of the associated field of study. Rather they discuss research that is similar in nature to that presented in this work and provide references for those interested.

3.1 Assembly Analysis

Assembly analysis is a topic that has only received serious scientific study during the past 25 to 30 years. Although there are many aspects of this topic that could be discussed, only a few works will be described here because of their direct applicability to this thesis.

In 1987, Dr. Daniel E. Whitney and a colleague Dr. Thomas De Fazio published a paper on the generation of all possible mechanical assembly sequences [6]. The paper outlines a method which simplifies the techniques of Bourjault [7] in presenting questions to a user which, when answered, provide rules to prune the set of all possible combinations of parts given by a liaison diagram into feasible assembly sequences. Their technique was later implemented as an automated computer tool [8] which generates the appropriate questions from knowledge of the liaison diagram, asks them to the user, and generates and displays the set of feasible assembly sequences. That program is part of the suite of computer tools used to implement AOD. Two more researchers, Gilbert and Jastrzebski, developed statistical models of geometric variations in assemblies using matrix transform methods [9]. In [9], Gilbert has developed matrix representations of the standardized

dimensional and form tolerances outlined in the ANSI Y14.5 geometric dimensioning and tolerancing (GD&T) standard. Gilbert attaches localized coordinate frames to surfaces, axes, or points of interest and uses the GD&T specification to define a zone within which that coordinate frame must lie in order to satisfy the tolerance specification. This work has application to MLA in that MLA employs matrix methods for calculation, attaches local coordinate frames to features, and defines motion limit zones that define the locus of locations of the feature coordinate frame similar to the tolerance zones specified by Gilbert.

3.2 Feature Models and Feature Based Design

Many researchers have been interested in the topic of feature based design. Notables among them are Jami Shah and Marti Mantyla. [10, 11, 12] Their work discusses the use of feature based design as a means to simplify design and CAD work, and to store relevant information about each feature such as geometric properties, kinematic properties, and manufacturing information. They propose using an object-oriented programming style to create parameterized versions of features on parts. This approach has been borrowed for MLA. Using their system, a user is able to choose a feature from a library and place the feature on a part. A few parameters are then specified to define the geometric and processing properties of the feature. This approach replaces the "electronic drawing" nature of many CAD systems. The feature model takes care of the details of geometric construction and provides a more natural design tool for the user. Much of their discussion of features is done at a high level excepting a detailed method for implementing a feature based CAD system. They mention the existence of motion limits on unconstrained degrees of freedom, but provide no method for the calculation of those limits.

Other important works in the field include De Fazio et. al., Anderson and Chang, Mascle et. al., and Kim and Wu. De Fazio [13] and his group at the Charles Stark Draper Lab developed a system in which they added feature based computational capability to the IDEASTM CAD modeling program. In this system, the user first designs all parts in an

assembly by adding features to the basic geometry of a part. The feature information is used by downstream applications to do assembly analysis (including generation of all possible assembly sequences), process planning, and a cost analysis. provides a design tool whose scope is extended beyond the capability of the Shah and Mantyla CAD system. Anderson and Chang [14] developed a system called AMPS which is focused on the link between design and process planning. Using AMPS consists of six steps: feature refinement, process selection, tool selection, process sequencing, and NC cutter path generation. This system is unique in that the process planning step also includes fixture planning. Tolerances are included on features to aid in semi-automated process and tool selection. Mascle, Jabbour, and Maranzana [15] have developed a system called SCAP (Systeme de Caracteristiques d'Assemblage de Produits - Product Assembly Features System) which adds geometric, functional, and technological information to a liaison diagram. The added information attempts to automatically answer many of the assembly sequence constraint equations that are asked explicitly to the user in the assembly sequence software mentioned above. It considers processing steps and other factors in the determination of feasible assembly sequences. It makes no assertions about the kinematic properties of assembly features. Kim and Wu propose a standard feature representation scheme in which features are classified as to the number of kinematic degrees of freedom they constrain when used to connect two parts [16, 17]. They discuss describing part mating constraints before, during, and after assembly. Their model defines 17 types of constraints coming from 28 possible cases because some of the cases are the same with an axis rotated. A four-digit code is used to represent each type of mating condition. The 17 types thus span the set of all possible constraint configurations. They use the before, during, and after assembly constraint conditions to derive compliance control strategies for designing automated means of assembling the parts that are modeled.

MLA borrows from the systems described above a few key concepts that are common to most of them. First, MLA employs an object-oriented approach in which feature information is stored as a unique data type containing information used by downstream calculation modules. Second, a combination of user input and computation are required

to obtain the desired results. Third, feature location and orientation are defined by a local feature coordinate system (FCS) which is related to a part coordinate system (PCS) and a global coordinate system (GCS) through homogeneous transforms. Lastly, MLA uses the standard coordinate systems for feature definition proposed by Kim and Wu [16, 17].

3.3 Screw Theory Kinematics

Applying screw theory to do kinematic computations is a well developed field whose roots began in 1900 with Ball's defining work on screw theory [18]. An outline of screw theory and its application to this work is provided in Chapter 3. In 1966, Waldron [19] was the first to apply screw theory to the problem of determining the relative degrees of freedom (dof) between any two bodies in a mechanism. Waldron derived the series and parallel laws of mechanisms along with "simple" and "complex" joints, which he used to create formulas for calculating the mobility (relative dof) of any link in a mechanism. Waldron showed that previous mobility formulas based on different mathematics were not accurate in all cases and pointed out their limitations. He then extended the theory to the manufacture of mechanisms by determining the number of accurate operations required to manufacture a machine based on the mechanism in question. In 1978, Hunt further developed the application of screw theory by identifying 22 cases of the reciprocal screw systems put forth by Ball [20]. Hunt applied the screw theory techniques to mechanism synthesis rather than analysis. He described the stationary position and uncertainty characteristics of mechanisms. J.E. Baker published a series of papers in the early 1980's in which he applied the methods of Waldron with a few refinements to many types of complex mechanisms [21 - 23]. Two of Baker's major contributions to the field were the development of a screw system algebra which was used to solve for the mobility and degree of redundancy in both planar and spatial mechanisms, and the investigation of determining motion extrema in linkages. This last concept is directly applicable to the research presented in this thesis where the physical limits of motion in assemblies, which are modeled as kinematic mechanisms, are calculated. T.H. Davies was also an active researcher in the field, publishing a series of papers on linkage analysis in the early 1980's. Davies took a graph theory approach to

mechanisms and showed how Kirchoff's well known loop and node equations could be applied to mechanism analysis [24]. Using that theory, Davies developed matrix algebra based formulae for determining the degrees of mobility and redundancy in planar and spatial mechanisms [25 - 27]. He also explored the reciprocal nature of wrenches and screws to determine the rate of work done by wrenches acting on a mechanism, and shows how stresses can be locked into redundantly constrained mechanisms.

3.4 Alternate Approaches to Applied Kinematic Analysis

Other researchers have developed methods of kinematic analysis using methods other than traditional screw theory. Woo and Freudenstein [28] transform traditional screw coordinates into Plucker line coordinates in order to facilitate their method of calculation. The basic output of the paper is an algorithm that allows one to analyze (simulate) the movements of mechanisms by applying infinitesimal twists about each joint in the mechanism and summing the motions. It contains a table of joint (feature) representations in line coordinates that corresponds to those presented in Chapter 5 of this thesis using screw coordinates.

A second approach taken by Kramer [29] is concerned mainly with determining how a chosen set of assembly features will position a part with respect to others in an assembly. Kramer calls this degree of freedom analysis. The computerized method allows the user to develop an assembly model by connecting parts together via traditional kinematic joints (revolute, linear,...). The analysis is done one part at a time and involves virtually mating one of the joints, and revolving or translating the part along the unconstrained dofs until it is in a position to mate another of its joints. This is done until either all joints are mated or all 6 dofs have been constrained. It is possible that this method could be applied within the analysis framework of this thesis. It would be used to determine if a chosen set of assembly features could position the parts they are placed on in the desired manner and with the desired accuracy. This is a good possibility for future work in the field of motion limit analysis.

A third and completely different approach mathematically uses the theory of convex cones to make kinematic calculations similar to those described above. An example is Hirai and Asada [30].

3.5 Kinematic Models of Features

By creating kinematic models of assembly features, researchers have been able to apply kinematic theory to assembly analysis. These applications borrow greatly from the kinematic analysis methods described in the last section, but have the distinction of applying the methods to more practical problems in robotics and assembly analysis. Mason and Salisbury use screw theory to characterize the nature of different types of contacts between robot gripper hands and objects [31]. Each type of contact is considered with friction absent and present. Twist and wrench systems describing each contact situation are derived. The models are used to select a robot hand design depending on the types and directions of forces that the hand can impart to objects. Ohwovoriole and Roth extend traditional screw theory by deriving two new types of screw systems which they call "repelling" and "contrary" screws [32]. These systems arise when a basic assumption in reciprocal systems of constant contact between bodies in the mechanism is violated. The new systems describe the situations in which motion causes two bodies in a mechanism to separate (repelling) or penetrate (contrary) each other. It is shown that if one wishes to assemble two parts, the parts can only move along twists that are either reciprocal or repelling in order for assembly to proceed. More recently, a Ph.D. student at Stanford, Ranjit Konkar, based his thesis on creating screw system representations of assembly mating features and using the methods of screw theory to determine the number of relative degrees of freedom between any two parts in an assembly [33, 34]. Konkar details the screw system representation of six basic assembly features which correspond to traditional kinematic joints such as prismatic and revolute joints. He also outlines a computational procedure for implementing the calculation. These two elements were lacking in most of the works on kinematics and kinematic feature models described above. Konkar also treats the synthesis problem of deducing a joint type and geometry from the screw system representation of a desired dof

capability. Konkar developed an assembly analysis program called "Assembly Editor" which implements the calculations described above. The focus of Dr. Konkar's thesis, and of papers he wrote on the subject, is on the computational efficiency of his store and reuse method of handling the program data.

MLA extends Konkar's work by defining 17 types of assembly features. It uses the same calculation algorithm to determine the available dof's of any part in an assembly, and then extends the analysis to the calculation of the actual quantitative motion capability of the part along those dof's.

3.6 Summary

This chapter presented prior research work that has been done in many research areas that relate to the topic of Motion Limit Analysis. A brief description was given of each researcher's work along with an explanation of the similarities or differences between that work and MLA. Chapter 4 will present detailed mathematical models of 17 types of assembly features which span the set of possible methods of degree of freedom constraint between two distinct rigid parts.

Chapter 4 Screw Theory

Motion Limit Analysis uses the mathematics of screw theory in order to calculate the possible degrees of freedom of any part in an assembly subject to the constraints placed upon it by the features that connect it to other parts. A brief summary of screw theory as it applies to this work is given below.

4.1 Definition of Terms

According to Chasle's theorem, any motion or physical displacement of a rigid body can be reproduced as a rotation of the body about a unique line in space and a translation along that same line. The pitch of this helicoidal motion is the ratio of translation to rotation. The line about and along which the motion takes place is called the Instantaneous Screw Axis (ISA). For infinitesimal displacements, the first order geometric properties of the motion are completely determined by the screw axis and the pitch. In order to determine the actual displacement of the body under the action of the screw, the amount of infinitesimal rotation about the ISA must be specified. This quantity is called the amplitude of the screw. Six quantities are required to define the axis and pitch of the screw. The definitions given below are paraphrased from [18, 32, and 33].

Screw: A screw is an ordered 6-tuple that may represent either a twist or a wrench (which will be defined below). Thus the context has to be given in order to interpret a screw. The first triplet represents a line vector with which is associated a unique line in

space. The second triplet represents a free vector which is not confined to a specific line of action but whose direction only is important. A unique point in space is associated with this triplet.

The physical interpretation of a screw depends upon whether it is being used to represent a twist or a wrench.

Twist: A twist is a screw which describes to first order the instantaneous motion of a rigid body. The first triplet represents the angular velocity of the body with respect to a global reference frame. The second triplet represents the velocity, in the global reference frame, of that point on the body or its extension that is instantaneously located at the origin of the global frame. Thus the unique line associated with the first triplet is the axis of rotation, the ISA, and the unique point associated with the second triplet is the point on the body, or as extension, which is at that instant located at the origin of the global reference frame. A twist is often written as a row vector of the form:

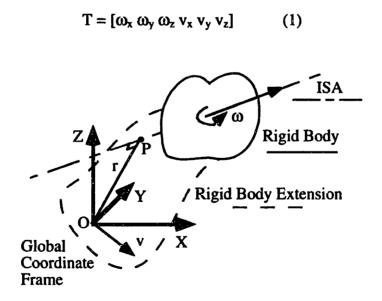


Figure 4.1 Diagram of Twist Interpretation

The interpretation of the twist is shown in Figure 4.1. The second triplet \mathbf{v} is calculated by taking the cross product of the angular velocity $\boldsymbol{\omega}$ and the vector \mathbf{r} which extends from

the origin of the global coordinate frame to a point P on the ISA. The length of r is the perpendicular distance from the ISA to the origin.

Wrench: Poinsot's principle provides the motivation for the definition of a wrench. It says that any set of forces and couples applied to a body can be reduced to a set comprising of a single force acting along a specific line in space and a pure couple acting in a plane perpendicular to that line.

A wrench is a screw which describes the resultant force and moment of a force system acting on a rigid body. The first triplet describes the resultant force in a global reference frame. The second triplet represents the resultant moment of the force system about the origin of the global frame. A wrench is also written as a row vector and takes the form:

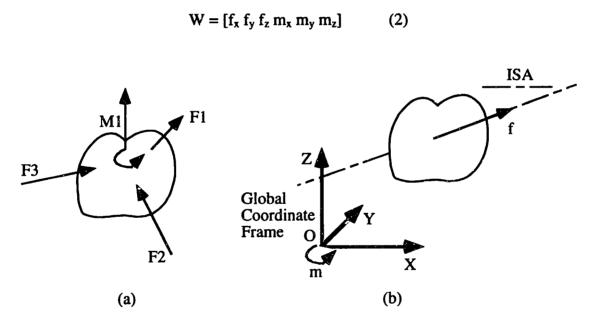


Figure 4.2 The Wrench Resulting From an Applied Force System

Suppose a set of forces and moments are applied to a rigid body as in Figure 4.2a. The wrench that describes that system is the resulting force **f** and moment about the origin **m** shown in Figure 4.2b. Note that the line of action of the resulting force defines the instantaneous screw axis of the wrench. Using the notation of Figure 4.2, the following relations are used to calculate **f** and **m**:

$$\mathbf{f} = \Sigma F_i$$
 $i = 1, ..., n$
 $M_i = r_i \times F_i$ $i = 1, ..., n$ (3)
 $\mathbf{m} = \Sigma M_i$ $i = 1, ..., n$

where r_i is the perpendicular distance from the PCS to the line of action of F_i.

Space of Screws: A screwspace is the set of all screws attainable from linear combinations of a given set of screws. Thus the terms twistspace and wrenchspace are defined from this term.

Reciprocal Screws: Two screws (a wrench and a twist) are mutually reciprocal if their virtual coefficient is zero. In physical terms, this implies that the wrench does no work in moving along the path represented by the twist. This allows one to compute the twists that a body may undergo while in contact with other bodies without breaking the contact. It also allows one to compute the wrenches that the contact may support while the body moves in accordance with a known screw. This is known as the dual nature of reciprocal screws. One can be calculated from knowledge of the other. Mathematically, if T represents the twistspace and W represents the wrenchspace, then the relation ToW = 0 holds for the screw system. ToW is the virtual coefficient. The calculation of the intersection of screws employs the dual concept.

Union of Screws: The union of multiple screws is obtained by concatenating the individual screws into a matrix with each screw occupying a row. Thus if $s_1, s_2,...,s_n$ are the individual screws, the union is given by:

$$Union(s_1, s_2, ..., s_n) = \begin{bmatrix} s_1 \\ s_2 \\ ... \\ s_n \end{bmatrix}$$
 (4)

A screwmatrix is thus an $n\times6$ matrix, where n is the number of independent screws represented in the matrix. When the screws are twists or wrenches the union is called a twistmatrix or wrenchmatrix respectively.

Intersection of Screws: The intersection of different sets of screws is the set of screws common to all the sets. The intersection is computed as a double reciprocal. Let S_1 , S_2 ,..., S_n be the different screwsets where each S_i is a set. Then the intersection is given by:

$$S_{intersection} = Reciprocal \left(\bigcup_{i=1}^{n} Re \, ciprocal(S_i) \right)$$
 (5)

The function Reciprocal(S) is a combination of two operations: 1) computing the nullspace of the screwmatrix of the screw set S, and 2) "flipping" the first three elements of the result with the last three. "Flipping" is defined in this context as exchanging the columns of the matrix according to the following pattern: 1 becomes 4, 2 becomes 3, 3 becomes 6, 4 becomes 1, 5 becomes 2, and 6 becomes 3. The flipping operation is performed to preserve the positions of the line vectors and the free vectors in a screw, i.e. the first triplet represents the line vector and the second triplet represents the free vector. Thus to compute the intersection of a set of twistmatrices, the reciprocal of each twistmatrix is computed to obtain a set of wrenchmatrices. The union of the wrenchmatrices is obtained by gathering the wrenches into one matrix. The reciprocal of this matrix is the intersection of the original set of twistmatrices.

4.2 Application of Screw Theory in Describing Kinematic Joints

Any joint between two parts allows a set of independent relative motions between the parts, and/or is also able to transmit a set of forces and couples from one part to the next. Each independent motion can be represented instantaneously as a twist. Thus if three independent motions are allowed by a joint, three twists are sufficient to describe the set. Concatenation of the twists provides a twistmatrix which spans the space of instantaneous motions allowed by the joint. Similarly, a wrenchmatrix can be found that describes the set of forces and torques that can be transmitted by the joint provided that physical

connection between the parts is not broken. The rank of a twistmatrix is equal to the connectivity of the joint it represents. The connectivity is the number of relative degrees of freedom between two bodies. For example, a revolute joint in kinematics allows one relative rotational degree of freedom about the joint axis connecting two members. Therefore, a twist could be written which describes the motion capability of one member relative to the other. The ISA of the twist would be concurrent with the axis of revolution of the revolute joint. Only one dof is allowed, and so a twistmatrix of rank one, containing one twist, is sufficient to describe the joint. Assuming the revolute joint is closed, it can transmit forces in three independent directions and couples in two independent planes. Thus five wrenches are necessary to describe the force transmission ability of the joint. The corresponding wrenchmatrix will be of rank five.

The example points out an important fact about the twistspace and wrenchspace of a given joint. They are reciprocal spaces. No joint can allow a motion and transmit a force in the same direction. Similarly, a couple cannot be transmitted by a joint about an axis of allowed rotation. Thus, if the twistmatrix of a joint is known, its wrenchmatrix can be calculated as the reciprocal of the twistmatrix and vice versa. This also implies that for any twistmatrix of rank n, the corresponding reciprocal wrenchmatrix will be of rank 6-n.

If a part is connected to other parts by more that one joint, the resultant motion that is allowed by the combination of joints is the logical intersection of the individual twistmatrices that define each joint. This is called the resultant twistmatrix, and it is computed using the intersection of screws formula presented above.

4.3 Interpretation of Resultant Twistmatrices and Wrenchmatrices

Suppose that a resultant twistmatrix was calculated by intersecting a set of several twistmatrices describing the joints connecting a part to others in an assembly. This matrix would contain the logical intersection of twistmatrices describing the joints. This matrix would have the form of:

$$R = \begin{bmatrix} \omega_{1x} & \omega_{1y} & \omega_{1z} & v_{1x} & v_{1y} & v_{1z} \\ \omega_{2x} & \omega_{2y} & \omega_{2z} & v_{2x} & v_{2y} & v_{2z} \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$
(6)

The number of rows of R is the number of relative degrees of freedom between the part being analyzed and the parts it is connected to. Each row is a twist describing an independent degree of freedom. The first triplet ω of the row is a unit vector describing the direction of an axis about which the part could rotate. The second triplet \mathbf{v} is the velocity that a point at the origin of the global coordinate frame would have if the part were rotating about ω with an angular velocity of unit magnitude. For a twist to appear in the resultant twistmatrix, it means that all of the joints connecting the part to others will allow the motion described by that twist. This is what is meant by the logical intersection of the joint twistmatrices.

A twist representing a pure translation is one in which the ω vector is 0. Thus the twist looks like:

$$\begin{bmatrix} 0 & 0 & 0 & \nu_x & \nu_y & \nu_z \end{bmatrix} \tag{7}$$

Suppose now that we took the reciprocal of each of the twists representing unconstrained dofs in Equation 6 above. We would obtain a set of wrenchmatrices describing all of the forces and torques that could be applied to the part in question via the joints that connect it to the other parts. In other words the wrenchmatrices would represent all of the degrees of freedom that were constrained by the various joints. Taking the intersection of these wrenchmatrices in a similar manner as was used for the twistmatrices produces a resultant wrenchmatrix W. W has the form of:

$$W = \begin{bmatrix} f_{1x} & f_{1y} & f_{1z} & m_{1x} & m_{1y} & m_{1z} \\ f_{2x} & f_{2y} & f_{2z} & m_{2x} & m_{2y} & m_{2z} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$
(8)

For a wrench to appear in W, this means that the constrained degree of freedom it represents is common to all joints. In other words, each joint is attempting to constrain that particular degree of freedom. Thus, the rows of W describe the over-constrained or redundantly constrained degrees of freedom of the part being analyzed. The first triplet **f** would give the direction of an over-constrained linear motion and the second triplet **m** would give the axis of an over-constrained rotation. This is analogous to saying that more that one joint could impart a force to the part in the direction of **f** or a torque about the axis **m**.

4.4 Summary

This chapter provided definitions of the terms used in classical screw theory and also explained a standard representation of screws in matrix form. Set operations used for manipulating groups of screws were then defined. It was shown briefly how screw theory could be applied to describe kinematic joints. Lastly, the physical interpretation of a resultant twistmatrix or wrenchmatrix calculation was given. Chapter 5 will extend the discussion of how screw theory can be applied to kinematics by giving a twistmatrix mathematical representation for seventeen kinds of assembly features which span the space of possible kinematic motions that can be allowed by physical features.

Chapter 5 Feature Models

In order to make computer-based calculations using assembly features, a mathematical model that can sufficiently describe the properties of the features that are of interest must be developed. MLA requires feature models that characterize the various features as to their ability to constrain the various dofs of the parts they connect. The model must also support calculations of geometric motions as limited by the features. This motion characterizes the variation absorption capability of each feature. This chapter will give mathematical definitions of assembly features that posses these capabilities. As such, it provides the foundation for the following discussion in Chapter 6 of Motion Limit Analysis.

5.1 Variation Absorption at Feature Sites

One important result of feature modeling is the ability to calculate possible variation absorption that can occur at feature sites. It is important to note that adjustability (variation absorption) at local feature sites can occur in two ways, explicitly and implicitly. These concepts will be explained in this section to provide context for the feature models that follow.

 $\leq i$

5.1.1 Explicit Variation Absorption

In explicit absorption, the part retains one or more degrees of freedom after being added to an assembly but before all fastening of mates has taken place. A Motion Limit Vector (MLV) is a 6×1 vector that contains the numerical limits on the unconstrained motions. The MLV pair (one for positive direction motions and one for negative) resulting from analysis of the mates completed so far will describe the unconstrained dofs and place limits on them. The part may then undergo rigid body motion along the unconstrained dofs, within the limits set by the MLV, until the person or machine doing the assembly is satisfied with its position. This knowledge would generally be obtained by an in-process measurement. In this case, the measurement is acting as a "virtual mate" which is constraining the unconstrained dofs of the part. This type of virtual mate will be shown on a DFC as a directed arrow going between the nodes with a smaller rotated arrow on top of it (see Figure 5.2 below). It looks similar to the symbol for a variable resistor in a circuit diagram. The mates, then contacts, to other parts would then be fastened in order to lock in the position of the part and provide strength and support. The exact position of the contact features would have to be adjustable in order to conform to the current position of the part.

An example of this type of process would be the adding of an airplane fuselage skin panel to an assembly of other similar panels. The situation is shown in Figure 5.1 below.

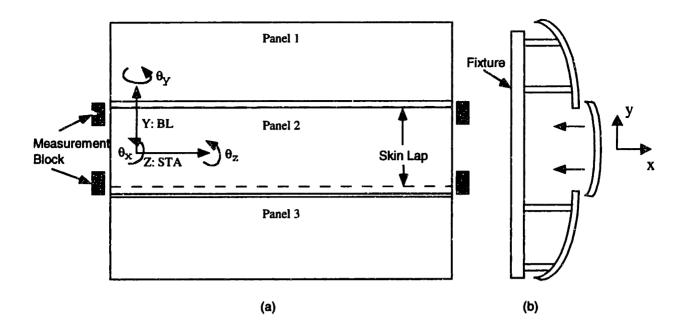


Figure 5.1 Skin Panel Assembly Showing Explicit Variation Absorption

In this assembly there are three skin panels. Two panels, 1 and 3, have already been located on the fixture that holds the assembly prior to fastening. The fixture contains measurement blocks with respect to which the position of the skin panels are measured in order to establish their correct position. Panel 2 is placed on top of panels 1 and 3 and makes a plate-plate lap joint mate with each of them (Fig. 5.1b). These mates together constrain only three degrees of freedom of panel 2. This panel is still free to undergo planar motion by translating in Y or Z or rotating about a vector aligned with the X-axis. Using the virtual mate notation, the DFC representing the process is shown in Figure 5.2.

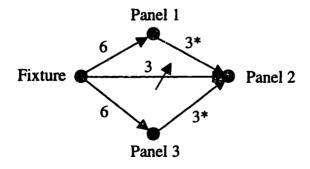
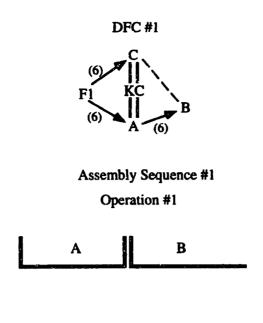


Figure 5.2 DFC for Skin Panel Assembly

The 3's marked with a * represent the same dofs, so panels 1 and 3 together constrain only 3 dofs of panel 2. The next step is for the workers to adjust the position of panel 2 until the measurements between it and the various measurement blocks are within specifications for the assembly. Holes are drilled between parts 1 and 2, and 3 and 2 along the skin lap joints. Temporary fasteners are then installed in the holes to lock in the position of panel 2 until automated drilling and riveting can take place. This example represents explicit variation absorption because panel 2 undergoes rigid body motion in adjusting its position with respect to the measurement blocks. The measurements could actually be modeled as virtual assembly features whose MLVs contained limits equal to the engineering tolerances on the measurements. Variation absorption takes place in this example because panel 2 iss positioned correctly with respect to panels 1 and 3 despite the inherent variations in the size and shape of panel 2.

5.1.2 Implicit Variation Absorption

Implicit variation absorption takes place only at **contacts** in situations where the mates fully constrain the location of the parts. To illustrate this, let's revisit assembly sequence #1 for the simple three-part assembly presented in Section 2.2. The parts are shown in Figure 5.3 below.



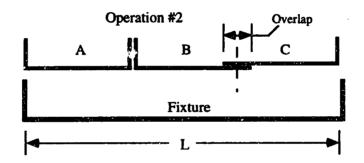


Figure 5.3 Sheet Metal Assembly Showing Implicit Variation Absorption

After the subassembly A-B has been completed, it is placed in a fixture and part C is added to the assembly. The fixture constrains all six dofs of both the subassembly A-B and of part C. Nonetheless, variation absorption has taken place. It happened at the plate-plate lap joint contact feature between parts B and C. The overlap distance for this joint has some limit given by the MLVs for the feature. Within these limits, length variations in parts A, B, and C are absorbed at this site by varying the amount of overlap. The overlap amount is not set explicitly by causing either the subassembly A-B or part C to undergo rigid body motion. Rather the lap amount is the resultant of positioning the parts with respect to the fixture, and the absorption takes place implicitly. Thus in this case, the MLVs represent the limits on the lengths of parts A, B, and C rather than limits on rigid body motion.

O

5.1.3 Variation Absorption Capabilities of Features

The above discussion of implicit and explicit variation absorption provides some basis for a discussion of the variation absorption capabilities of assembly features. Computationally speaking, implicit and explicit variation absorption can be thought of in the same way. In choosing a feature, the designer has to think about what type of relative motion a given feature will allow between the two parts it connects. For explicit variation absorption, this motion will be rigid body motion. For implicit variation absorption, the motion can be thought of as the motion created if the part were to expand or contract, or alternatively the motion of varying the exact feature location on the part. The motion will only take place along dofs that are unconstrained by the feature. Thus the designer will choose and orient a feature such that the degrees of freedom that s/he wishes to control are constrained and those that are intended to allow motion (absorb variation) are not. In this sense, each feature can act as both a mate and a contact depending on which direction one is concerned with. MLA requires the designer to place limits on the quantitative amount of motion allowed along the unconstrained dofs by either setting geometric parameters to size the feature or specifying numerical limits. In this way the direction and amount of variation absorption can be calculated for each part according to the choice of features.

5.2 Twistmatrix Representation of Features

As described in the last Chapter 4, twistmatrices can be used to represent the motion properties of kinematic joints. By modeling assembly features as kinematic joints, the twistmatrix representation and each type of joint can be used in kinematic computations. This chapter will present explicit models of 17 types of assembly features. The 17 features span the set of possible combinations of dof constraint of rigid body objects. Each type may have one or more real-world physical realizations.

5.2.1 Role of 4x4 Transform Matrices

Each of the features presented in section 5.2.4 below is represented in a local coordinate frame based on the feature. In order for the matrices representing the twists of each

feature to be combined to obtain a twistmatrix describing a pan, all of the features must be defined in the part coordinate frame. The 4×4 homogeneous transform provides the tool to accomplish this purpose [5]. Assume that a revolute joint, f1, with one rotational degree of freedom (RDOF) allowed about its z-axis is placed somewhere in space with respect to a part-level coordinate frame O. Let the 4×4 transform that describes the location and orientation of f1's coordinate frame in part coordinates be denoted by F. F is a partitioned matrix of the following form:

$$F = \begin{bmatrix} A \mid d \\ O \mid 1 \end{bmatrix} \tag{9}$$

where A is a 3×3 rotation matrix, \mathbf{d} is a 3×1 displacement vector, and \mathbf{O} is a 1×3 row vector of zeros. The unit angular velocity vector $\mathbf{\omega}$ that describes the allowable joint rotation in its own coordinate frame is the vector $[0\ 0\ 1]^T$. The orientation of this vector in the part frame is given by the matrix multiplication of A and $\mathbf{\omega}$. Let this product be denoted by $\bar{\mathbf{\omega}}$. The vector is transposed after multiplication such that $\bar{\mathbf{\omega}}$ is a 1×3 vector. The origin of O and the origin of f1 are located on the same rigid part. Thus the twistmatrix representation of the feature f1 will describe the relative angular velocity of the part and the relative linear velocity of the point at the origin of O with respect to the part to which the one under consideration is connected via the feature. The vector \bar{r} that describes the position of f1 with respect to O is also needed to compute the twistmatrix. \bar{r} is contained within the 4×4 transformation matrix F as the displacement vector \mathbf{d} . Thus, using this nomenclature, the twistmatrix of f1 is given by:

$$T_{f1} = \left[\vec{\omega} : \vec{v}\right] \tag{10}$$

where
$$\vec{\omega} = (A\omega)^T$$
,
 $\vec{v} = \vec{r} \times \vec{\omega}$, and
 $\vec{r} = d^T$.

These part and coordinate frame relationships are illustrated in Figure 5.4 below.

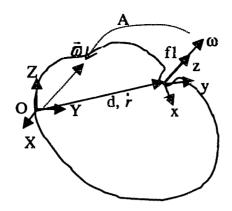


Figure 5.4 Relationship Between Part and Feature-Level Coordinate Frames

Figure 5.4 shows that \mathbf{d} is the translation vector that transforms the vector $\boldsymbol{\omega}$ from the feature level coordinates of f1 to the part coordinate system O. A expresses the difference in orientation between O and f1. Thus multiplying $\boldsymbol{\omega}$ by A will convert the direction of $\boldsymbol{\omega}$ from f1's coordinate system to an equivalent direction expressed with respect to O. This method of transforming vectors defined in a local coordinate frame to that of a defining part or global frame will be used for all features presented below. Whether the vectors are angular rotation vectors $\boldsymbol{\omega}$ or translational direction vectors \boldsymbol{k} , the transformation is the same.

5.2.2 Nomenclature and Conventions

Each feature below is shown in its nominal mating configuration. The positive z-axis of the feature should always be pointing in the nominal mating direction. By arbitrary choice, the y-axis points in the direction of translational freedom for features with only one translational dof. For features with two translational dofs, the y-axis points in what is considered the primary absorption direction. For cases where this does not apply, axis direction assignments are arbitrary but adhered to as convention for each case. Symbols such as "Px" indicate the amount of freedom in the positive x direction. Similarly, "Nx" denotes the negative x direction. For unconstrained rotational degrees of freedom, the

limits are taken as π to $-\pi$ such that any angular position may be reached within this range. Frictional forces are considered negligible in restraining part motion.

In each case, one part in the mating pair is taken to be immobile and is denoted by the attached ground symbol . The title of each feature tells how many degrees of freedom it constrains if used singly to connect two parts. The parameters in brackets are the required information that the user must input in order to characterize the feature aside from nominal location and axis orientation. These parameters are often obtained from the specific geometry of the pair of parts being mated. For example in a peg-hole mate, the hole is immobile and the nominal coordinate frame is placed on the cylindrical axis of the peg and centered lengthwise (z direction). Thus, limits are placed on the amount of z-axis motion freedom of the peg such that the peg does not leave the hole and become unmated. It is assumed that all features are at their nominal size and shape. For example in a pin-slot combination, the pin is the same diameter as the width of the slot such that no motion is allowed along the short axis of the slot.

5.2.3 Freedom Zones and Absorption Zones

Freedom zones and absorption zones are calculated in the same way, but have different interpretations and uses. Each is a zone within which the local feature coordinate frame is allowed to reside. Both can be used to absorb variation. Variation absorption occurs when the nominal part relationships established between the two parts connected by the feature can be maintained despite part variations due to manufacturing, thermal expansion, or other errors. This can occur either explicitly or implicitly as explained in Section 5.1 above. In this sense, a freedom zone represents the explicit variation absorption capability of a feature, and an absorption zone represents the implicit variation absorption capability. Thus, the freedom zone for each feature is a zone of allowable rigid body motion, and an absorption zone represents the ability of a feature to implicitly absorb variation. The freedom and absorption zones are shown in the figures below, where practical, by a graph where the shaded region represents the locus of allowable feature location values along respective axes. Each zone is also represented as

a pair of 6×1 vectors. The first vector represents motion freedom in positive axis directions while the second represents freedom in negative axis directions. The first three elements of each vector represent the x, y, and z directions of freedom respectively. The second three elements represent rotations about the three axes and appear in the order of θ_x , θ_y , and θ_z . This ordering of dofs is reversed with respect to the twistmatrix representation of translation and rotation. This was done in order to agree with the notation of the $\tilde{U}(k)$ vector of the State Transition Models of Assembly Processes presented in Chapter 2. The 6×1 vectors are called Motion Limit Vectors, and will be explained in more detail in Chapter 6.

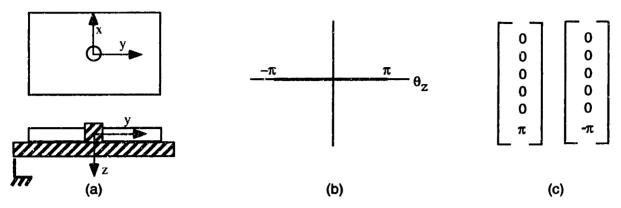
5.2.4 Single Mating Features

The series of figures below presents each primitive feature type with its nominal axis directions, freedom zone characterization, and twistmatrix representation. The features are categorized as to the number of unconstrained degrees of freedom they allow if acting independently between two rigid parts. The quantities in brackets represent user supplied geometric parameters which are needed to calculate the vectors representing the motion limit zone for the feature. Some may argue that what is here called a mating feature is actually a pair of mating features. The 17 features span the set of possible methods of degree of freedom constraint between two distinct rigid parts. A similar set of features were developed by Kim and Wu and are presented in [16, 17].

Feature 1 has zero dofs unconstrained, and thus its twistmatrix representation is trivial:

$$T_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{11}$$

2) Plate Pin in Through Hole (1dof) [Null]

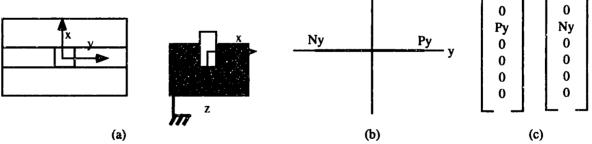


The revolute joint represented by Feature 2 is characterized by its line of rotation which is simply the local angular velocity vector ω . Its twistmatrix is given as:

$$T_2 = \left[\vec{\omega} : \vec{v}\right] \tag{12}$$

where, as above:
$$\vec{\omega} = (A\omega)^T$$
,
 $\vec{v} = \vec{r} \times \vec{\omega}$, and $\vec{r} = d^T$.





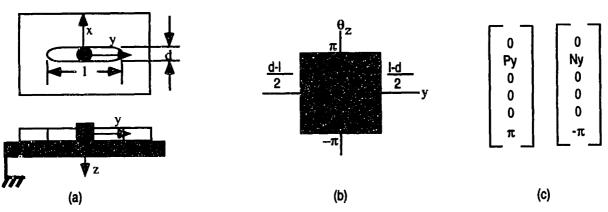
Feature 3 allows only translational motion. For rigid bodies, all points on the body have the same velocity such that the location of the feature with respect to the part coordinate frame does not matter. Thus the feature is characterized by the translation direction vector **k**. **k** is the vector of direction cosines of the translation direction in the local coordinate frame, which is the y-direction by convention. The twistmatrix of Feature 3 is given by:

$$T_3 = \left[\mathbf{O} : \vec{v}\right] \tag{13}$$

where,
$$\vec{v} = (Ak)^T$$
,
and $\mathbf{O} = (0, 0, 0)$.

The same procedure for creating \vec{v} is followed as was used to create $\vec{\omega}$ from A and ω above.

4) Plate Slotted Pin Joint (2 dof) [l,d]



For the plate slotted pin joint, the following formulas are used to calculate Py and Ny.

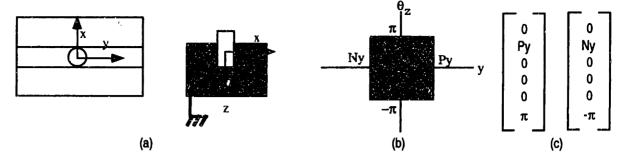
$$Py = \frac{l-d}{2}, Ny = \frac{d-l}{2} \tag{14}$$

The twistmatrix for Feature 4 is a combination of a rotational dof and a translational dof.

The vectors are formed similar to the above cases and are given by:

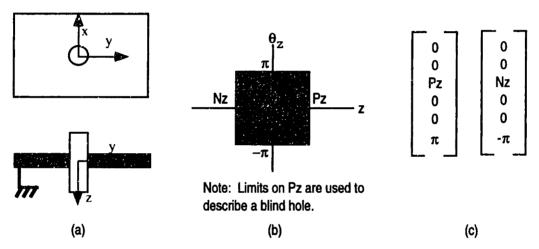
$$T_4 = \begin{bmatrix} \vec{\omega} : \vec{v}_1 \\ \mathbf{O} : \vec{v}_2 \end{bmatrix}$$
 (15)
where $\vec{v}_1 = \vec{r} \times \vec{\omega}$, and $\vec{v}_2 = (\mathbf{Ak})^{\mathrm{T}}$.

5) Prismatic Slot, Round Peg (2dof) [Py, Ny]



Feature 5 has a twistmatrix identical to that of Feature 4.

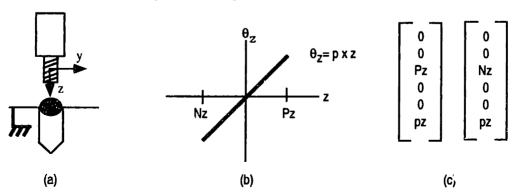
6) Round Peg in a Through or Blind Hole (2dof) [Pz, Nz]



Feature 6 is a basic cylindrical joint where the axis of rotation and the translational axis are the same. This axis is characterized by the vector ω . The twistmatrix is:

$$T_6 = \begin{bmatrix} \vec{\omega} : \vec{v} \\ \mathbf{O} : \vec{\omega} \end{bmatrix} \tag{16}$$

7) Threaded Joint (2dof coupled) [Pz, Nz, p]

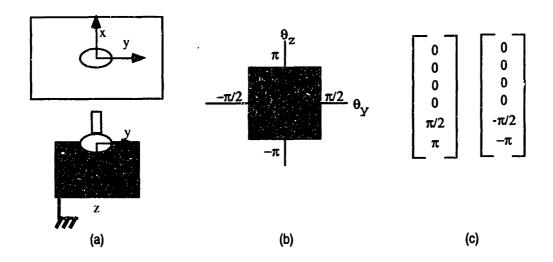


Feature 7 is a basic helical joint where 'p' represents the pitch of the threads in the units of rad/inch. The rotation and translation that it allows are coupled and are represented by the twistmatrix:

$$T_{7} = \left[\vec{\omega} : \vec{v}\right] \tag{17}$$

where
$$\vec{v} = \vec{r} \times \vec{\omega} + p\vec{\omega}$$
.

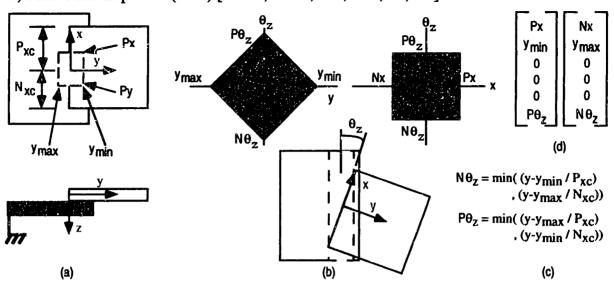
8) Elliptical Ball and Socket (2dof) [Null]



Feature 8 allows two rotations and no translations. Let ω_1 and ω_2 represent the two rotational axes. The twistmatrix is given by:

$$T_8 = \begin{bmatrix} \vec{\omega}_1 : \vec{v}_1 \\ \vec{\omega}_2 : \vec{v}_2 \end{bmatrix} \tag{18}$$

9) Plate-Plate Lap Joint (3dof) [Ymax, Ymin, Pxc, Nxc, Px, Nx]



The lightly shaded area in (a) represents the allowable location of the coordinate frame for the lapping part as determined by the limits y_{max} , y_{min} , Px, and Nx. Following convention, Px and Nx represent the limits on x-direction motion of the overlapping plate. y_{max} represents the y position of the plate edge at the maximum allowable amount of part overlap. Similarly, y_{min} represents the minimum amount of overlap that is needed to maintain joint integrity. These two parameters define a zone in the y direction within which the edge of the lapping part must be maintained in order to meet joint specifications. This zone determines the maximum amount that the part can rotate about the z-axis as shown (b) where the part is rotated to its maximum limit. The dimension from the feature axis (about which rotation occurs) and the corners of the part becomes important in determining this maximum rotation angle. The symbols P_{xc} and N_{xc} are used for the dimensions to the corner in the positive and negative x-direction respectively. The equation for the sin of the angle is:

$$\sin(\theta_z) = \frac{y - y_{\min}}{\sqrt{P_{xc}^2 + (y - y_{\min})^2}}$$
 (19)

By assuming that $y-y_{min}$ is much smaller that P_{xc} , we can use a small angle approximation where θ_z is linearly related to y and given by:

$$N\theta_{z} = \min\left(\frac{y - y_{\min}}{P_{xc}}, \frac{y - y_{\max}}{N_{zc}}\right)$$
 (20)

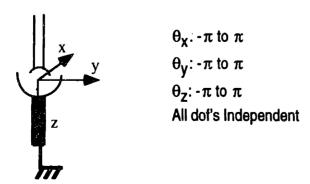
Thus the angle in the negative θ_z direction is given by the minimum of the two quantities in parenthesis depending mainly on the relative magnitudes of P_{xc} and N_{xc} . An expression for $P\theta_z$ is similar and is given in (c). Note that y in the above equations is the actual y position of the edge of the overlapping plate measured in the feature coordinate system. Because we are attempting to calculate the maximum possible value of $N\theta_x$, the quantity y in equations 19 and 20 is assumed to be zero. Therefore it is not a user supplied input. It is assumed that the y direction is the direction that is critical for defining joint integrity and is the major axis of variation absorption. Hence, allowable motion in the x direction is assumed to be driven by a desire for the part edges to align or other similar criteria. As such, the x-position of the plate within the limits of Px to Nx has no correlation to the allowed θ_z rotation. The two quantities are thus represented as independent in (b).

The planar joint is characterized by the vector normal to its surface. This is the same vector ω which characterizes the allowed rotation. The twistmatrix of Feature 9 is thus:

$$T_9 = \begin{bmatrix} \vec{\omega} : \mathbf{O} \\ \mathbf{O} : \vec{v}_1 \\ \mathbf{O} : \vec{v}_2 \end{bmatrix}$$
 (21)

where \vec{v}_1 = any vector perpendicular to $\vec{\omega}$, and $\vec{v}_2 = \vec{\omega} \times \vec{v}_1$.

10) Spherical Joint (3dof) [Null]



A spherical joint is characterized by the point at its center about which all rotations are allowed. Denoting this point in part coordinates by $[c_1, c_2, c_3]$, the twistmatrix is:

$$T_{10} = \begin{bmatrix} \vec{\omega}_1 : \vec{v}_1 \\ \vec{\omega}_2 : \vec{v}_2 \\ \vec{\omega}_3 : \vec{v}_3 \end{bmatrix}$$
 (22)

where

$$\vec{\omega}_1 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

$$\vec{\omega}_2 = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$$

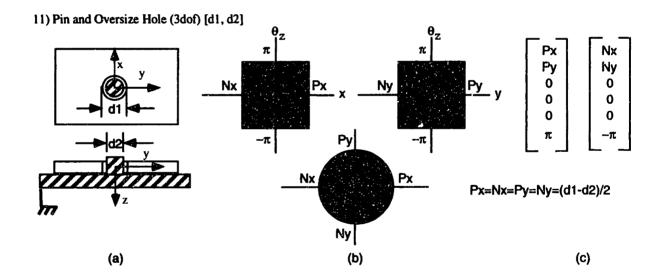
$$\vec{\omega}_3 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$$

$$\vec{v}_1 = \vec{r} \times \vec{\omega}_1$$

$$\vec{v}_2 = \vec{r} \times \vec{\omega}_2$$

$$\vec{v}_3 = \vec{r} \times \vec{\omega}_3$$

$$\vec{r} = \begin{bmatrix} c_1 & c_2 & c_3 \end{bmatrix}$$



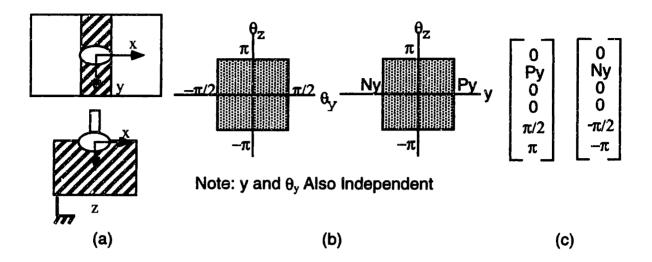
The pin and oversize hole is similar to a plate-plate lap joint (Feature 9) for small motions. Thus it has the same twistmatrix representation:

$$T_{11} = \begin{bmatrix} \vec{\omega} : \mathbf{O} \\ \mathbf{O} : \vec{v}_1 \\ \mathbf{O} : \vec{v}_2 \end{bmatrix}$$
 (23)

where \vec{v}_1 = any vector perpendicular to $\vec{\omega}$, and

$$\vec{v}_2 = \vec{\omega} \times \vec{v}_1$$
.

12) Elliptical Ball in Cylindrical Trough

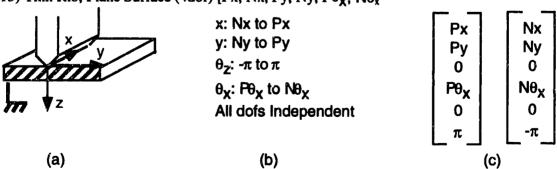


Feature 12 is a combination of two rotational dofs and one translation which are all independent. Thus if **k** again represents the direction of translational motion in local coordinates, the twistmatrix of Feature 12 is:

$$T_{12} = \begin{bmatrix} \vec{\omega}_1 : \vec{v}_1 \\ \vec{\omega}_2 : \vec{v}_2 \\ \mathbf{O} : \vec{v}_3 \end{bmatrix}$$
 (24)

where \vec{v}_1 and \vec{v}_2 are defined as usual, and $\vec{v}_3 = (Ak)^T$.

13) Thin Rib, Plane Surface (4dof) [Px, Nx, Py, Ny, $P\theta_X$, $N\theta_x$

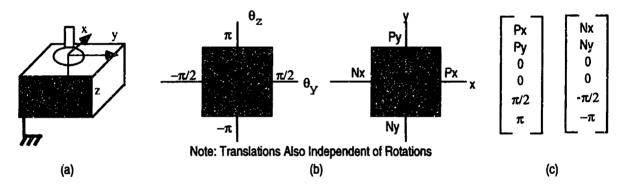


 $P\theta_x$ and $N\theta_x$ are dependent on the included angle of the point of the rib. This joint can be seen as a combination of a planar and an x-axis revolute joint. By concatenating the twistmatrices for these joints, the twistmatrix for Feature 13 is:

$$T_{13} = \begin{bmatrix} \vec{\omega}_x : \vec{v}_1 \\ \vec{\omega}_z : \mathbf{O} \\ \mathbf{O} : \vec{v}_2 \\ \mathbf{O} : \vec{v}_3 \end{bmatrix}$$
 (25)

where \vec{v}_1 is defined as usual, $\vec{v}_2 = (Ak_x)^T$, and $\vec{v}_3 = (Ak_y)^T$.

14) Ellipsoid on Plane Surface (4dof) [Px, Nx, Py, Ny]

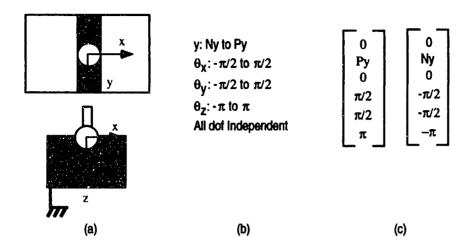


Feature 14 is very similar to feature 13 except that the in-plane rotation is about the y-axis instead of the x-axis. Thus the twistmatrix is:

$$T_{14} = \begin{bmatrix} \vec{\omega}_{y} : \vec{v}_{1} \\ \vec{\omega}_{z} : \mathbf{O} \\ \mathbf{O} : \vec{v}_{2} \\ \mathbf{O} : \vec{v}_{3} \end{bmatrix}$$
 (26)

where \vec{v}_1 is defined as usual, $\vec{v}_2 = (Ak_x)^T$, and $\vec{v}_3 = (Ak_y)^T$.

15) Spherical Ball in Cylindrical Trough (4dof) [Py, Ny]



Feature 15 is a combination of a spherical and prismatic joint. By concatenating those individual twistmatrices, the twistmatrix for Feature 15 is:

$$T_{15} = \begin{bmatrix} \vec{\omega}_1 : \vec{v}_1 \\ \vec{\omega}_2 : \vec{v}_2 \\ \vec{\omega}_3 : \vec{v}_3 \\ \mathbf{O} : \vec{v}_4 \end{bmatrix}$$
 (27)

where

$$\vec{\omega}_1 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

$$\vec{\omega}_2 = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$$

$$\vec{\omega}_3 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$$

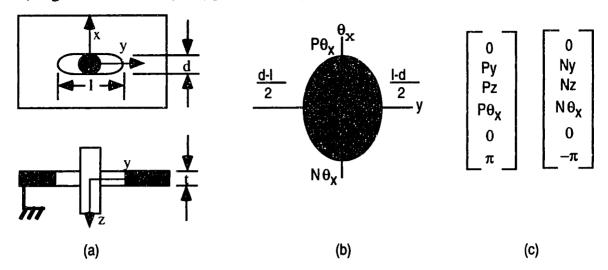
$$\vec{v}_1 = \vec{r} \times \vec{\omega}_1$$

$$\vec{v}_2 = \vec{r} \times \vec{\omega}_2$$

$$\vec{v}_3 = \vec{r} \times \vec{\omega}_3$$

$$\vec{r} = \begin{bmatrix} c_1 & c_2 & c_3 \end{bmatrix}$$
and
$$\vec{v}_4 = (Ak_y)^T$$
.

16) Peg in a slotted hole (4dof) [l, d, t, Pz, Nz]



Feature 16 is a pin-slot joint without the associated planar pair of Feature 4. The amount of pin rotation allowed about the x-axis depends on the y-position of the peg in the slot, the diameter of the peg, and the thickness of the plate containing the slot. Assuming that the pin rotates about the x-axis as shown in the figure, the equation for computing θ_x is:

$$\theta_x = \frac{\pi}{2} - \alpha - \beta \tag{28}$$

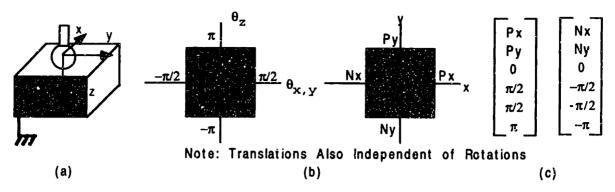
where
$$\tan \alpha = \frac{t}{l-2|y|}$$
 and $\sin \beta = \frac{d}{\sqrt{t^2 + (l-2|y|)^2}}$

The twistmatrix for Feature 16 is given by:

$$T_{16} = \begin{bmatrix} \vec{\omega}_1 : \vec{v}_1 \\ \vec{\omega}_2 : \vec{v}_2 \\ \mathbf{O} : \vec{v}_3 \\ \mathbf{O} : \vec{v}_4 \end{bmatrix}$$
 (29)

where $\vec{v}_3 = (Ak_y)^T$ and $\vec{v}_4 = (Ak_z)^T$.

17) Sphere on Plane Surface (5dof) [Px, Nx, Py, Ny]



Feature 17 is a combination of a spherical and planar joint. The twistmatrix is:

$$T_{17} = \begin{bmatrix} \vec{\omega}_1 : \vec{v}_1 \\ \vec{\omega}_2 : \vec{v}_2 \\ \vec{\omega}_3 : \mathbf{O} \\ \mathbf{O} : \vec{v}_3 \\ \mathbf{O} : \vec{v}_4 \end{bmatrix}$$
(30)

where the rotations are defined as in Feature 15, $\vec{v}_3 = (Ak_x)^T$, and $\vec{v}_4 = (Ak_y)^T$.

5.3 Summary

This chapter provided the mathematical foundation for MLA by detailing the twistmatrix representation of mating features. First, the variation absorption capabilities of assembly features was discussed. Then the role of 4x4 homogeneous transform matrices in describing feature locations on parts was described. The method in which 4x4 transforms are used to calculate twistmatrices for individual features was also presented. Following the definition of some nomenclature and conventions, models for 17 types of assembly mating features that span the set of possible methods of degree of freedom constraint between two distinct rigid parts were presented. The models were classified by the number of independent relative motions they would allow between two distinct parts connected by each. The definitions of a freedom zone and absorption zone were also given. Chapter 6 will present in detail the method of Motion Limit Analysis including descriptions in both prose and mathematical form of the algorithms used for calculation.

Chapter 6 Motion Limit Analysis

Motion Limit Analysis (MLA) is a tool used to support the top down design and analysis of assemblies. The purpose of MLA is to provide mathematical models of assembly features from which the ability of a feature to position one part relative to another in space can be calculated. Positioning comprises calculation of which of the six degrees of freedom of a rigid body are constrained, and which are capable of allowing rigid body motion. If rigid body motion is allowed, the quantitative amount of motion is of interest. A user of this theory is able to obtain two major types of information about an assembly:

- 1. Knowledge of the directions and quantitative amounts of possible motions of a part that is being added to an assembly at a given assembly station via connection of a defined set of assembly features.
- 2. Knowledge of whether or not the defined feature set over-, under-, or fully-constrains the location and orientation of the part.

MLA has been implemented as a software tool. The following sections will describe the application of motion limit analysis, give the definition of some terms, and give descriptions of the algorithms that are used to accomplish the calculations. Chapter 7 will give a detailed description of the software itself. Note that when a set of features on a part is analyzed, a separate analysis is done for the mates and the contacts. Thus all of the following discussion applies to both cases where sometimes the list of features will be mates and other times it will be contacts.

6.1 Scope of Application

As was described in the Introduction chapter, MLA was developed in order to provide computational support to assembly layout and analysis methods under development by Mr. Ramakrishna Mantripragada. Thus it is specifically tailored to the needs, notation, and conventions of those methods. Nonetheless, the methods used to implement MLA are general and it can be used to analyze any assembly which uses the mating features described in Chapter 5 to connect parts together.

6.2 Motion Limit Vectors

A Motion Limit Vector (MLV) is a 6×1 vector in which the first three rows represent numerical limits on translational motion in three independent directions given in whatever the working units for the assembly are, and the last three rows represent limits on rotational motion, in units of degrees, about axes aligned with the same three independent directions. The pair of MLVs associated with each assembly feature was given in the figures in Chapter 5. This pair is an inherent property of the feature. The vectors represent the limits on part motion that would result if the feature were used singly to locate one part with respect to another. The numerical entries in these vectors are sometimes user specified and other times calculated automatically from geometric parameters describing the feature. The main purpose of MLA is to combine the effects of several sets of MLVs associated with the features that are being used to connect one part to others, and calculate the net pair of MLVs that describe the resulting motion properties of the part as a whole. In the context of the results produced by the MLA software, this resultant pair represents the amount that the position and orientation of a part which is being added to an assembly at a given assembly station can be adjusted in order to adjust out the effects of part variations. Because there is a resultant MLV describing motion capability in both the positive and negative directions, there can be bilateral limits on the possible adjustment of the part, which limits need not be symmetric.

6.3 Definition of Algorithms

This section contains both prose and mathematical descriptions of the algorithms for obtaining a resultant twistmatrix from a set of feature defining twistmatrices and then converting that resultant twistmatrix into a pair of motion limit vectors. The mathematical descriptions are provided to help the reader better understand the theory underlying MLA, and to provide some guidance to the operation of the computer code used to implement MLA if that is of interest.

6.3.1 Prose Description of the Twistmatrix Intersection Algorithm

As was explained in the chapter on screw theory, the resultant twistmatrix obtained by combining the effects of two or more assembly features is given by the logical intersection of the twistmatrices of each feature. This algorithm was developed by Konkar [33]. To compute the intersection of a set of twistmatrices, the reciprocal of each twistmatrix is computed to obtain a set of wrenchmatrices. The reciprocal is calculated by computing the null space of the given matrix and then flipping the result. The flip operation is done by exchanging elements 1, 2, and 3 with elements 4, 5, and 6 respectively in each row of the null space matrix. This flip operation is done in order to maintain the location, within the matrix, of the line vectors and the free vectors of the screw in each row. The union of the wrenchmatrices is obtained by gathering the wrenches into one matrix, WU. The reciprocal of this matrix is the intersection of the original set of twistmatrices, i.e. the resultant twistmatrix.

6.3.2 Mathematical Description of the Twistmatrix Intersection Algorithm

Many of the matrix manipulations and calculations required by the MLA program are done using MATLAB™ software¹. The commands shown in bold below are MATLAB commands. The author wrote a small user-defined function in MATLAB to implement the flip operation.

¹ MATLAB is a trade mark of The Math Works corporation.

Step 1: For each feature on the part, find the associated wrenchspace (Wi) of wrenches reciprocal to the twistmatrix (Ti) of the feature. This is done by:

- a) wi = null(Ti),
- b) wti = wi' where 'denotes transpose,
- c) Wi = flip(wti). The flip operation is done by exchanging columns according to the following pattern: 1 becomes 4, 2 becomes 5, 3 becomes 6. Columns 4, 5, and 6 become 1,2, and 3 respectively.
- Step 2: Collect all the Wi matrices for each feature into a matrix called WU which is the union of the individual W's.
 - a) WU = [W1; W2; ...; Wn]

Step 3: Obtain the row reduced echelon form of the WU matrix by:

a) W = rref(WU)

Step 4: Obtain the resultant twistmatrix from the combined action of all part features by:

- a) t = null(W)
- b) tt = t'
- c) Twist = flip(tt)

6.3.3 Interpretation of the Resultant Twistmatrix

Section 4.3 introduced the basic interpretation of the resultant twistmatrix. This section will extend that explanation by providing descriptions of the calculation algorithms used to provide interpretations of a twistmatrix which are more readily understood than presenting the twistmatrix in matrix form.

A resultant twistmatrix is of the form:

$$R = \begin{bmatrix} \omega_{1x} & \omega_{1y} & \omega_{1z} & v_{1x} & v_{1y} & v_{1z} \\ \omega_{2x} & \omega_{2y} & \omega_{2z} & v_{2x} & v_{2y} & v_{2z} \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$
(31)

Each row represents an independent degree of freedom of the part being analyzed. If the degree of freedom is a pure translation, the elements of the ω vector are all zero. The interpretation of a pure translation is simple. The vector \mathbf{v} is a unit magnitude velocity vector pointed in the direction of an allowed translation. Because rotations and translations are coupled, the interpretation of a row representing an allowed rotation is a bit more complicated. Figure 4.1 is repeated here as Figure 6.1.

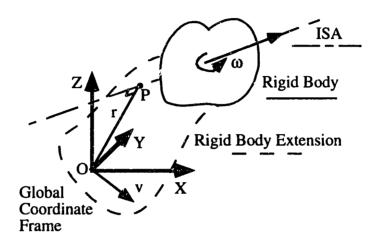


Figure 6.1 Diagram of the Interpretation of a Twist

The vector \mathbf{v} is calculated as the cross-product of $\boldsymbol{\omega}$ and \mathbf{r} where \mathbf{r} is the coordinate vector of a point P which lies on the instantaneous screw axis (ISA) of the twist. Thus \mathbf{r} points from the defining coordinate frame to the point P. Figure 6.1 calls this frame a 'Global Coordinate Frame'. In the case of MLA the defining frame is the part-level coordinate frame. Thus for a twist where $\boldsymbol{\omega}$ and \boldsymbol{v} are given, we are interested in finding a point P which lies on the ISA axis. This calculation is of interest to locate the rotation axis in space so as to quantify the motion of the part. P is found using the 'point' algorithm described next.

6.3.3.1 Prose Description of the Point Algorithm

Because \mathbf{v} is calculated as a cross-product of ω and \mathbf{r} , two of the three elements of this equation are known. The task is to solve the cross-product equation for \mathbf{r} which is the coordinate vector of the point P. This solution will be a type of 'inverse cross-product'

involving ω and v which will yield r. Depending on the elements of ω , the solution for r is calculated differently. The details are given in Section 6.3.3.2.

6.3.3.2 Mathematical Description of the Point Algorithm

If none of the elements of ω are zero, \mathbf{r} is perpendicular to the ISA and is given by the following matrix equation:

$$r = \begin{bmatrix} 0 & \boldsymbol{\varpi}_z & -\boldsymbol{\varpi}_y \\ -\boldsymbol{\varpi}_z & 0 & \boldsymbol{\varpi}_x \\ \boldsymbol{\varpi}_y & -\boldsymbol{\varpi}_x & 0 \end{bmatrix}^{-1} \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}$$
(32)

If any element of ω is zero, the perpendicular cannot be solved for directly. Even so a point on the ISA can be found according to which element of ω is not zero. Thus \mathbf{r} will point to some point P on the ISA and is calculated accordingly as follows [33]:

if
$$\omega_z \neq 0$$
 then $\mathbf{r} = [-v_y/\omega_z, v_x/\omega_z, 0]$,
if $\omega_z = 0$ and $\omega_y \neq 0$ then $\mathbf{r} = [v_z/\omega_y, 0, -v_x/\omega_y]$, and (33)
if $\omega_y = \omega_z = 0$ and $\omega_x \neq 0$ then $\mathbf{r} = [0, -v_z/\omega_x, v_y/\omega_x]$.

6.3.3.3 Prose Description of the Row ID Algorithm

The row id algorithm is used to classify the rows of a resultant twistmatrix as rotations or translations. The logic of this algorithm is simple. If the row represents a pure translation, all of the elements of the first triplet will be zero. Else the row represents a rotation. Thus the algorithm searches the first three elements of each row to see if every element is zero and if so it classifies the row as a translation, otherwise the row is classified as a rotation. This algorithm is important for later calculations where the analysis differs in its treatment of translations and rotations. We will forego the mathematical description of this algorithm because of its simplicity. The interested reader can look at the function within the definition of the TwistMat class which is included in Appendix A.

6.3.4 Prose Description of the MLA Algorithm and Geometric Reasoning

The twistmatrix intersection analysis results in an $n\times 6$ matrix where n is the number of instantaneous degrees of freedom that the part has when constrained by the specified set of features. Having obtained and analyzed that resultant twistmatrix, the next objective of the analysis is to calculate how much part motion is possible along each of those degrees of freedom before reaching the motion limits of the features. The result of this analysis will be two 6×1 motion limit vectors for each independent dof, which quantify the maximum part motion in the positive and negative directions respectively from the nominal configuration.

The input to the algorithm is a resultant twistmatrix, and two MLVs for each feature that was defined by the user upon feature specification. Rotational and translational dofs of the part can be analyzed separately. This is true because the result will be motion limit vectors which represent the maximum rotation and/or translation that the part may undergo. The vectors imply nothing about how that motion is obtained, whether it is purely rotation or translation or a combination. Thus the three rotational components of each resultant MLV for the part will be the components of the vector whose direction is parallel to an allowed axis of rotation, and whose magnitude equals the maximum rotational angle allowed about that axis. The translational components represent the maximum allowed displacement of the part coordinate frame in each direction.

6.3.4.1 Rotational DOFs

For a part constrained by features to be able to rotate about a given axis ω , the assembly features must allow translational motion in a direction that is mutually perpendicular to the axis of rotation and the vector from that axis to the origin of the feature. The situation is shown in Figure 6.2 below.

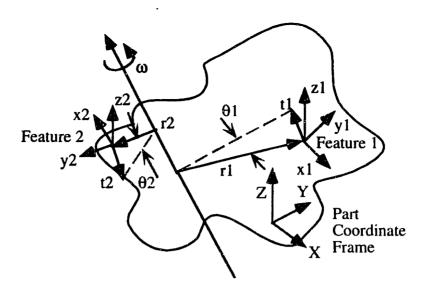


Figure 6.2 Illustration of Rotational Motion Allowed by Combinations of Features

Using the notation of the figure, Feature 1 has to allow motion along the vector t1 which is mutually perpendicular to r1 and ω . The same holds for feature 2. It is assumed that linear features such as a slot possess a slight curvature such that part rotation is allowed until the connecting pin reaches the end of the slot. Since the twistmatrix only gives the direction of ω , the first step in the algorithm is to calculate a point through which ω passes in order to establish its location in space. Then the r vectors must be calculated. Each r vector has to satisfy three conditions:

- 1. **r** is perpendicular to ω , (i.e. $\mathbf{r} \bullet \omega = 0$)
- 2. r passes through the origin of the feature coordinate frame,
- 3. r intersects the ω axis.

Figure 6.3 below illustrates the method of calculation of r.

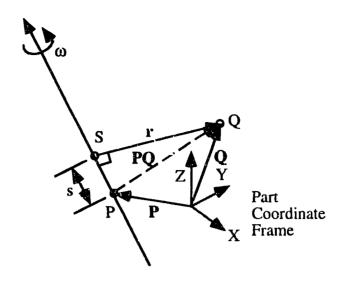


Figure 6.3 Geometry for Calculation of r

From knowledge of the twist T which describes the allowed rotation, the coordinates of a point P which lies on ω can be calculated. This calculation was shown in Section 6.3.3.1. Q is the point at the origin of a given feature coordinate frame expressed in part coordinates. From points P and Q the vector \mathbf{PQ} is calculated. The projection of this vector in the direction of ω is the scalar distance s. Knowing the direction of ω , the coordinates of P, and the distance s allows one to calculate the coordinates of point S which lies at the base of the desired vector \mathbf{r} . Thus \mathbf{r} is the vector from S to Q. This process is repeated to find the \mathbf{r} vector for each feature $(\mathbf{r_i})$.

With ω and $\mathbf{r_i}$ established, $\mathbf{t_i}$ can be calculated. From this calculation, we are only interested in the direction of $\mathbf{t_i}$ and not its magnitude. The true magnitude of $\mathbf{t_i}$ should be equal to the maximum amount of motion that the feature will allow in that direction. The amount of motion allowed in the feature level \mathbf{x} , \mathbf{y} , and \mathbf{z} directions is known. Thus by knowing the orientation of $\mathbf{t_i}$ with respect to the ith feature coordinate frame, the magnitude of $\mathbf{t_i}$ can be calculated. The vectors $\mathbf{t_i}$ and $\mathbf{r_i}$ are then used to calculate the angle θ_i . This angle represents the amount that the part can rotate about the ω axis subject to the constraints placed upon it by feature i. The maximum angle that the part can rotate about ω is the minimum of the θ_i . This quantity is denoted by θ_i max. θ_i max can then be broken down into its part level components of θ_i , θ_i , and θ_i . These are the

quantities needed to fill the rotational part of the positive motion limit vector. The analysis is then repeated for rotation about ω in the negative direction. The results fill the rotational part of the negative motion limit vector. Note that if there are two or more RDOFs, the analysis must be repeated for both positive and negative rotations about the other ω axes. The resultant θx , θy , and θz of each of these analyses must then be compared to the corresponding quantity calculated from rotation about the first axis and the maximum of the quantities entered into the Motion Limit Vector for that part. The maximum is taken because each of the RDOFs is independent and we are in search of the maximum possible rotation.

6.3.4.1.1 Search For Axis Which Allows the Maximum Possible Rotation

It must be recognized that in many situations the ω vector contained in the resultant twistmatrix is not the only possible axis of rotation. There may be a one- or twodimensional infinity of parallel vectors representing possible rotation axes. algorithm just described calculates the motion capability of the part about one ω axis. The ω axis that allows the maximum possible motion must be searched for within the space of possible axes. The ω axis contained in the resultant twistmatrix is dependent on the location of the part coordinate frame if there is more that one possible axis. This is so because of the input of the **d** vector from the 4×4 transform matrix in calculating the twistmatrix for each feature. Thus, to quantify the space of possible ω axes, the twist analysis is done twice with the part coordinate frame in two different locations. If the resultant ω axes are the same (i.e. pass through the same point), it is known that there is only one possible axis. If the axes are different, the twist analysis must be done again with the part coordinate frame in a third location. If the ω axis resulting from this analysis lies on the line passing through the first two it is known that the line defines the space of possible axes. If not, the space of possible axes is the plane perpendicular to the three ω axes. Once the space is quantified, the search for the axis that allows the maximum motion is conducted within that space.

The point-direction equation of a line in 3D space with direction **b** passing through the point **p** can be expressed in vector form as: $\mathbf{v} = \mathbf{p} + \mathbf{tb}$. Where t is a scalar parameter and \mathbf{v} is the vector of coordinates of any point on the line. Thus by varying the value of t, the coordinates of any point on the line can be found. The two point form of a line passing through points **a** and **p** is expressed as: $\mathbf{v} = (1-t)\mathbf{p} + t\mathbf{a}$. This form will be used to check whether or not the third axis location in the above search algorithm lies on the line passing through the first two. Suppose the axis points of the three analyses are respectively the points **p**, **a**, and **v**. If the three points are co-linear, the following equations must hold:

$$\frac{v_x - p_x}{a_x - p_x} = \frac{v_y - p_y}{a_y - p_y} = \frac{v_z - p_z}{a_z - p_z} = t$$
 (34)

If any of the differences in the denominators of these equations are zero, the difference in the numerator must also be zero for the relation to hold true. The denominators cannot all be zero though because this implies that **a** and **p** are the same point which is not allowed. If it were the case, it would have been detected at the first step of the algorithm and the analysis would not have arrived at this point.

If the three points do not satisfy the conditions of co-linearity, the space of possible axes must be a plane. The plane is perpendicular to the direction of the ω axis and passes through the point **p**. One can construct a vector α that is perpendicular to ω . The components of α are: $[1/\omega_x \cdot 2/\omega_y \ 1/\omega_z]$. This is true if none of the components of ω are zero. If one or two of them are zero, the corresponding entry in α is set to one and the other entries in α corresponding to non-zero entries in ω are set to zero. For example, if $\omega = [1, 1, 0]$ the algorithm would produce $\alpha = [0, 0, 1]$. Thus $\alpha \cdot \omega = 0$. A basis for describing the plane is completed by finding a third vector β which also lies in the plane and is perpendicular to both α and ω . This vector is simply the cross product $\beta = \alpha \times \omega$. Thus any point in the plane can be described by a linear combination of α and β . The equation of a point v in the plane becomes:

$$\mathbf{v} = \mathbf{p} + \mathbf{s}\alpha + \mathbf{t}\beta,\tag{35}$$

where s and t are scalar parameters that may be varied.

6.3.4.1.2 Search Within A Linear Space

The search for the axis about which maximum rotational motion is allowed becomes the search for the maximum of a function with independent variable t and dependent variable θ . Each value of t represents a unique axis, within the linear space, about which a rotation of magnitude θ is allowed. To begin, three points on the function are known. They are points obtained by the analysis of each of the three positions of the part coordinate frame. The three points are stored in an array of points. More points are added to this array by varying the parameter t and using $\mathbf{v} = \mathbf{p} + \mathbf{t}\mathbf{b}$ to calculate more points that lie on the line whose direction is given by the vector \mathbf{b} . The limits on t are calculated according to the type and positioning of the features. The calculation of these limits will be given in the next section. When the array is filled, the magnitude, θ , of the rotation allowed about the axis passing through each point is calculated and stored in another array. θ is the minimum of the rotations allowed by each mated feature on the part about the axis in question. This second array of θ s is then searched for the maximum magnitude of θ . The point corresponding to this value of θ identifies the axis within the space which allows the maximum possible rotation.

6.3.4.1.3 Limits on the Scalar Parameter t

The limits on the scalar parameter t will determine the length of the linear space that is searched for possible rotation axes. Remember that this space is being searched for the axis which allows the maximum rotation. The maximum rotation depends on the distance of the axis from the feature limiting the rotation, and the shape of the freedom zone of the feature. The calculation situation is shown in Figure 6.4 below.

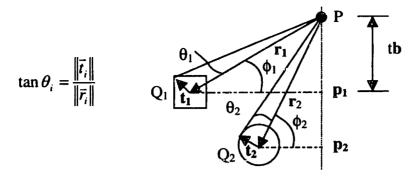


Figure 6.4 Calculation of Limits on the Scalar Parameter t

There are three possibilities for the shape of the freedom zone: rectangular, circular, and linear. The maximum value of θ_i for any zone will occur when the magnitude of t_i is maximized and the magnitude of \mathbf{r}_i is minimized. For all cases, the minimum value of \mathbf{r}_i occurs when the point P is coincident with the point \mathbf{p}_i . For the rectangular zone, θ_{max} occurs when \mathbf{t} points to the corner of the zone. Thus this relation sets the ratio between the length $\mathbf{t}\mathbf{b}$ and the distance from \mathbf{p} to the center of the zone, i.e. θ_{max} will occur when those two lengths are in the same ratio as the sides of the rectangular freedom zone. For a circular shaped freedom zone, θ_{max} occurs when $\mathbf{t}\mathbf{b} = \mathbf{0}$. \mathbf{b} is a unit vector. Thus the length $\mathbf{t}\mathbf{b}$ is equal to \mathbf{t} . A plot of θ vs. \mathbf{t} for a square and circular freedom zone is shown in Figure 6.5 to illustrate the behavior described in the last few sentences. For this plot, the square zone has sides of length 0.5 units and the circular zone has a diameter of 0.5 units.

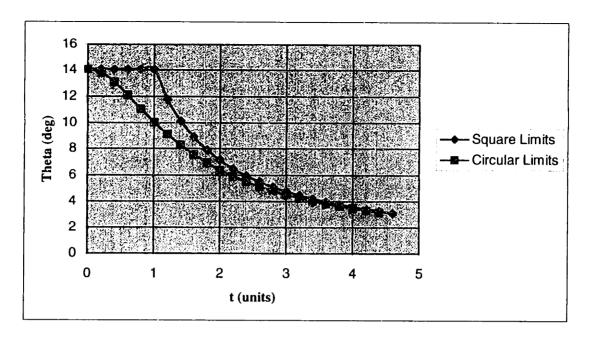


Figure 6.5 Theta vs. t For Two Shapes of Freedom Zones

If the zone is linear, say for a slotted-hole feature, the linear space of axes will be perpendicular to the line of the zone and cross through it at the center of the feature. In this situation, θ_{max} occurs about the axis passing through the center of the feature.

In the linear and circular cases, the magnitude of \mathbf{t} does not change with the position of point P. To simplify the calculations we will represent a rectangular zone as a circular zone with radius equal to half the diagonal of the rectangle, i.e. \mathbf{t}_{max} . Thus the dependence on the magnitude of \mathbf{t} is removed from the calculation, and θ_{max} will always occur at the point where the magnitude of \mathbf{r} is minimized. Thus the scalar parameter \mathbf{t} must be big enough for the linear space to encompass all of the points \mathbf{p}_i from every feature. The following mathematical formula will calculate the needed value of scalar \mathbf{t} .

$$t\vec{b} \ge \max(\vec{r_i} \bullet \vec{b})$$
 (36)

Thus the point which gives the minimum magnitude of \mathbf{r}_i will always be included in the search space.

6.3.4.1.4 Search Within A Planar Space

The space equation now has two independent variables s and t, and a dependent variable θ . Similar to the linear space search, an array full of points is created and the magnitude of the allowed rotation θ about the axis passing through each point is calculated. In the planar case, the points are arranged in a grid and are obtained by varying the parameters s and t in equation 35. The limits on s and t are both ± 5 . Again the values of θ are stored in an array which is searched for the maximum value of θ . The point associated with this value of θ denotes the axis about which the maximum rotation can occur.

6.3.4.2 Translational DOFs

For a rigid body undergoing pure translation, the velocity and displacement of every point on the body is the same. Thus to calculate the translational part of the motion limit vector, it is sufficient to choose the minimum of the motions allowed by each feature in each direction to fill the vector.

6.3.5 Mathematical Implementation of MLA Algorithm

Input: Two 6×1 limit vectors (PL_1 , NL_1), (PL_2 , NL_2), ..., (PL_N , NL_N) for each feature and a $n\times6$ resultant twistmatrix T, where N is the number of mate or contact features on the part and n is the number of unconstrained degrees of freedom, i.e. the rank of T. The rowspace of T spans locus of possible motions. Rotational and translational dofs are calculated separately. The row id algorithm of Section 6.3.3.3 is applied to identify rows as translation or rotation and the following algorithms are applied accordingly.

6.3.5.1 Rotational DOF's

Calculating θ_{max} for each rotational dof consists of quantifying the space of possible rotation axes, finding θ_{max} for each axis within the space, and searching the space for the overall θ_{max} .

6.3.5.1.1 Determination of the Space of Possible Axes

1. Move part axis to location (1,1,1) relative to original axis. This is done by post-multiplying the F matrices describing the location of each feature by the matrix:

$$F_{m} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
 (37)

- 2. Redo resultant twistmatrix analysis using new origin point.
- 3. Calculate the point through which each resultant rotation axis passes using the point algorithm of section 5.3.3.2. This gives \mathbf{p}_i .
- 4. Transform the coordinates of each point back into the original part coordinates:

$$\mathbf{p}_{oi} = \mathbf{p}_{i} - [1, 1, 1]$$
 (38)

5. Check each allowed rotation axis to see if if passes through the same point as the axis in the same direction obtained from the first analysis.

$$\mathbf{p}_{i} = \mathbf{p}_{oi} ? \tag{39}$$

If $\mathbf{p}_i = \mathbf{p}_{oi}$ there is only one rotation axis allowed. θ max is already known and can be resolved into its respective components (Section 5.3.5.3). If $\mathbf{p}_i \neq \mathbf{p}_{oi}$, the origin is moved to (2,2,2) relative to the original axis and the twistspace calculation is done again.

6. Again transform the points resulting from this third analysis back into original part coordinates:

$$\mathbf{p}_{oi} = \mathbf{p}_{i} - [2,2,2].$$
 (40)

For each axis, check to see if \mathbf{p}_1 , \mathbf{p}_2 , and \mathbf{p}_3 lie on the same line:

$$\frac{v_x - p_x}{a_x - p_x} = \frac{v_y - p_y}{a_y - p_y} = \frac{v_z - p_z}{a_z - p_z} = t ? \quad (41)$$

where $\mathbf{p}_1 = \mathbf{p}$, $\mathbf{p}_2 = \mathbf{a}$, and $\mathbf{p}_3 = \mathbf{v}$.

7. If the relationship is true, search the linear space defined by the three points which is:

$$\mathbf{v} = \mathbf{p}_1 + \mathbf{tb}, \quad (42)$$

where $\mathbf{b} = (\mathbf{p}_3 - \mathbf{p}_1) / || \mathbf{p}_3 - \mathbf{p}_1||$.

8. If the relationship is false, search the planar space defined by the three points which is:

$$\mathbf{v} = \mathbf{p}_{\perp} + \mathbf{s}\alpha + \mathbf{t}\beta, \qquad (43)$$

where $\alpha = [1/\omega_x - 2/\omega_y 1/\omega_z]$, and $\beta = \omega \times \alpha$

6.3.5.1.2 Finding Omax For a Given Rotational Axis

1. For each RDOF $(\omega_x \omega_y \omega_z v_x v_y v_z)$ find the point **p** through which the ISA passes (point algorithm).

$$\begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \begin{bmatrix} 0 & \omega_z & -\omega_y \\ -\omega_z & 0 & \omega_x \\ \omega_y & -\omega_x & 0 \end{bmatrix}^{-1} \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}$$
 (44)

Special Cases: If any of the three members of the ω vector are zero, the point is calculated according to the following:

$$\mathbf{p} = [-v_y/\omega_z \ v_x/\omega_z \ 0]^T \text{ if } \omega_z \neq 0, \text{ or}$$

$$\mathbf{p} = [\ v_z/\omega_y \ 0 \ -v_x/\omega_y]^T \text{ if } \omega_y \neq 0 \text{ and } \omega_z = 0, \text{ or}$$

$$\mathbf{p} = [\ 0 \ -v_z/\omega_x \ v_y/\omega_x]^T \text{ if } \omega_x \neq 0 \text{ and } \omega_y = \omega_z = 0.$$
(45)

 ω vectors can now be represented in direction-point form.

- 2. For each feature (i = 1 to n), construct vectors \mathbf{r}_i that are perpendicular to the ISA and pass through the origin of each feature coordinate frame. This is done by:
 - a) $\mathbf{PQ}_i = \mathbf{Q}_i \mathbf{P}$
 - b) $\omega \bullet \mathbf{PQ}_i = \mathbf{s}_i$
 - c) $S_i = P + s_i \omega$ (46)
 - d) $\mathbf{r}_i = \mathbf{Q}_i \mathbf{S}_i$
- 3. For each \mathbf{r}_i , obtain \mathbf{t}_i lying in the plane of \mathbf{r}_i and perpendicular to it.

$$\vec{t}_i = \bar{\omega} \times \vec{r}_i \qquad (47)$$

4. Normalize t_i.

$$\hat{t}_i = \frac{\bar{t}_i}{\|\bar{t}_i\|} \tag{48}$$

5. Calculate the magnitude of t_i using the motion limit vectors for the feature.

$$t_{ix} = (\hat{t}_i \bullet \hat{t}_i) P x_i$$

$$t_{iy} = (\hat{t}_i \bullet \hat{j}_i) P y_i$$

$$t_{iz} = (\hat{t}_i \bullet \hat{k}_i) P z_i$$

$$|t_i| = \sqrt{t_{ix}^2 + t_{iy}^2 + t_{iz}^2}$$
(49)

6. Calculate the maximum rotation angle about the ISA that is allowed by t_i using the formula:

$$\tan \theta_i = \frac{\|\vec{r}_i\|}{\|\vec{r}_i\|} \qquad S \qquad (50)$$

7. Choose the minimum of all θ_i as the maximum rotation allowed by the features.

$$\theta \max = \min(\theta_i) \tag{51}$$

6.3.5.1.3 Search Within A. Linear Space

Use equation 41 that gives the equation of the linear space to generate an array full of points in the space between the limits of t = 10 to t = -10. Find the θ_j for each axis point and corresponding to each feature using the algorithm of Section 5.3.5.1.2. At each point, choose the minimum of the rotations allowed by each feature as θ_i .

$$\theta_{i} = \min(\theta_{j}) \tag{52}$$

Choose the maximum of the θ_i to be θ_{max} .

$$\theta_{\text{max}} = \max(\theta_{\text{i}}) \tag{53}$$

6.3.5.1.4 Search Within A Planar Space

Use equation 42 that gives the equation of the planar space to generate a 10 x 10 unit grid of points in the plane and store them in an array. Find the θ_j for each axis point and corresponding to each feature using the algorithm of Section 5.3.5.1.2. At each point, choose the minimum of the rotations allowed by each feature as θ_i .

$$\theta_i = \min(\theta_i)$$
 (54)

Choose the maximum of the θ_i to be θ_{max} .

$$\theta_{\text{max}} = \max(\theta_i)$$
 (55)

6.3.5.2 Translational DOF's

1. For each TDOF (0 0 0 v_x v_y v_z), the motion available in each direction is computed by the dot product of the TDOF with each translational freedom allowed by a given feature. The freedom allowed by a given feature in a given direction is the product of the scalar motion limit contained in the MLV for that direction (e.g. Px_i) and the appropriate component direction vector contained in the twistmatrix description of that feature (e.g. v_{ix}). Denoting the resultant TDOF vector by v_r , the translation allowed by each feature in each direction is computed by:

$$dx_i = (\mathbf{v}_i \bullet \mathbf{i}_i) Px_i, dy_i = (\mathbf{v} \bullet \mathbf{j}_i) Py_i, dz_i = (\mathbf{v} \bullet \mathbf{k}_i) Pz_i.$$
 (56)

6.3.5.3 Calculating the Components of the Motion Limit Vector

1. It is possible that there are θ_{max} values associated with rotations about three independent axes. Thus the overall θ_{max} for the whole part is the minimum of the respective θ_x , θ_y , and θ_z components of θ_{max} in each direction. Thus:

$$\theta_x = \min(\theta \max_i(\omega_x)), \quad \theta_y = \min(\theta \max_i(\omega_y)), \quad \text{and} \quad \theta_z = \min(\theta \max_i(\omega_z)).$$
 (57)

2. Check that θx , θy , and θz are smaller that the corresponding rotational limits $P\theta x$, $P\theta y$, and $P\theta z$ in the MLV for each feature. If limits $P\theta x$, $P\theta y$, or $P\theta z$ are smaller, its value must replace the corresponding calculated rotational limit in the MLV.

- 3. These θx , θy , and θz are the rotational components of the part's motion limit vector.
- 4. The overall entries for the translational part of the motion limit vector are the minimum of the freedoms allowed by each feature. Thus:

$$Px = min(dx_i)$$
, $Py = min(dy_i)$, and $Pz = min(dz_i)$, $i = 1...n$. (58)

The entire process is repeated for rotations and translations in the negative direction to calculate the components of the negative direction MLV.

6.3.6 Redundant Constraint Analysis

The redundant constraint analysis is concerned with finding the intersection of the set of wrenches acting on a part corresponding to the features that connect the part to others in the assembly. If this matrix is not empty, it means that all of the features on a part are trying to constrain a certain degree of freedom of the part, i.e. the wrench is common to all of the features. This situation may cause assemblability or dimensional control problems when the part is assembled. Thus this analysis provides a constraint check for the designer and points out possible problem areas.

The need for detection of overconstraint is generally not taught in an undergraduate engineering curriculum, but rather realized by engineers when they encounter assembly problems [42]. The value of this analysis is shown by the following quote from [38]. The assembly in question is the 1994½ Ford Windstar Minivan. The problem described could have been avoided if the assembly had been modeled and analyzed using redundant constraint analysis at the design stage.

The seat installation process was having difficulty achieving a consistent clearance to trim panels. The model demonstrated that the seat was over located by four sized holes and operator dependant for installation. By increasing the size of two mounting holes and elongating a slot in a third stanchion the remaining sized hole became the four way locator. The operator was instructed to start with this hole. Surprisingly, consistent installation was achieved by increasing tolerances for three holes.

This example shows that overconstraint happens in real world assemblies and causes problems for companies. It also demonstrates that some engineers are unaware of this problem, as the author was surprised that resizing the holes and removing the overconstraint improved the installation process. The DFC for this assembly would be rooted on the sized-hole feature. It's interesting that the assembly instructions instruct the operator to start with this hole. It verifies one of the rules of a DFC, which states that mates should be made before contacts.

6.3.7 Prose Description of the Constraint Analysis Algorithm

The constraint analysis algorithm is very similar to the algorithm for computing a resultant twistmatrix. For the constraint analysis, we are interested in the intersection of all wrenches acting on a part via the assembly features. Following the resultant twistmatrix algorithm, the first step would normally be to start with the set of wrenchmatrices for a part and take the reciprocal of each to obtain a set of twistmatrices characteristic of the feature connections. In this case the twistmatrices are already known. Thus the first step is skipped, and the union of the twistmatrices is obtained by collecting the individual twistmatrices into one large matrix TU. The reciprocal of this matrix is the desired intersection of the wrenches acting on the part.

6.3.8 Mathematical Description of the Constraint Analysis Algorithm

Step 1: Collect the individual twists into a union matrix called TU.

a)
$$TU = [T1;T2;...;Tn]$$

Step 2: Obtain the row reduced echelon form of the TU matrix by:

a) T = rref(TU)

Step 3: Calculate the reciprocal of T by:

- a) w = null(T)
- b) wt = w'
- c) Wrench = flip(wt)

6.3.9 Interpretation of a Resultant Wrenchmatrix

A wrenchmatrix has the form of:

$$[f_x f_y f_z m_x m_y m_z]$$
 (59)

Non-zero entries for the f terms indicate overconstraint of a translation, while non-zero m terms indicate overconstraint of a rotation. The direction of overconstraint is the direction of the vector, f or m. The constraint analysis results are presented to the designer as a caution. Having the part overconstrained is not a guarantee that problems will be encountered. Often the overconstraint can be removed by changing the feature design. This was shown by the Windstar seat example cited above. The solution to the seat installation problem was to provide more clearance on all but one of the holes in order to remove the overconstraint.

6.4 Commentary on Analysis Results

The above discussion is very involved. This section will provide a summary of the important ways to interpret the results given by MLA.

First one can understand the constraint situation of a part by examining the ranks of the resulting twistmatrix (T) and wrenchmatrix (W). If rank(T) > 0, the part is underconstrained. If rank(W) > 0, the part is overconstrained. If rank(T) = 0 and rank(W) = 0 the part is fully constrained. This is the ideal solution. Note that a part can be both underconstrained and overconstrained at the same time. This is because constraint is a vector quantity of each dof of the part. For example, consider two flat plates lying one atop the other and connected by a pin-hole feature and a pin-slot feature. If the long axis of the slot is oriented perpendicular to the line between the two features, the top plate is free to rotate about the pin-hole feature, the rank of T is 1, and the assembly is underconstrained. Nonetheless, because of the assumption that features must stay mated, both features constrain the translation that would separate the two plates along with the two rotations in the plane of the flat mating surfaces. Thus the rank of W

is 3 and the assembly is overconstrained. This example will be worked in detail in Section 6.5.1 below.

Second, if a part is underconstrained by its mates, explicit variation absorption can occur along the unconstrained directions. The amount of variation that can be absorbed is given by the MLV resulting from the analysis of the **incoming mates** that position the part. The amount of implicit variation that can be absorbed is given by the MLV resulting from the analysis of the **contacts** to other parts. The implicit variation results are also passed as inputs to the Type-2 tolerance analysis algorithm shown in the flowchart of Chapter 2.

The AOD methodology does not use all of the analytical capability of MLA. For example, there is currently no provision for doing tolerance or assemblability analysis for the case of explicit variation absorption. For example, MLA could analyze the motions allowed when features with clearances are mated. AOD currently cannot use these results. The results may possibly be used to obtain quantitative assemblability results. Thus the abilities of MLA can point toward future development directions for the AOD methodology.

6.5 Examples

The MLA analysis procedure is shown through two simple examples below.

6.5.1 Example 1: Hole and Slot Combination

Figure 6.6 shows a two part assembly consisting of two flat plates joined by two features. One feature is a pin-through hole combination, and the other is a pin-slot combination.

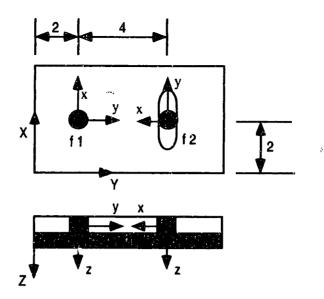


Figure 6.6 Hole-Slot Combination Assembly

The DFC for this two part assembly is shown in Figure 6.7.

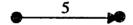


Figure 6.7 DFC for Hole-Slot Combination Assembly

The plate on top is incompletely constrained. By inspection we can see that the remaining dof is a rotation about the pin axis at feature f1. This result will be proven by the analysis. With the dimensions and orientations shown in the figure the F matrices (homogeneous transforms) for each feature are:

$$F1 = \begin{bmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad F2 = \begin{bmatrix} 0 & 1 & 0 & 2 \\ -1 & 0 & 0 & 6 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{60}$$

Following the definitions of hole and slot features, the twistmatrices are:

$$T1 = \begin{bmatrix} 0 & 0 & 1 & 2 & -2 & 0 \end{bmatrix} \qquad T2 = \begin{bmatrix} 0 & 0 & 1 & 6 & -2 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \end{bmatrix} \tag{61}$$

Applying the resultant twistmatrix algorithm to T1 and T2 yields the following matrices:

$$W1 = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1/3 & 8/9 & -2/9 & 2/9 & 0 \\ 0 & -2/3 & -2/9 & 5/9 & 4/9 & 0 \\ 0 & 2/3 & 2/9 & 4/9 & 5/9 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad W2 = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & .8944 & 0 & .4472 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$
(62)

Note that w1, w2, wt1, and wt2 are not shown here for clarity, but they can easily be deduced from W1 and W2. Concatenating W1 and W2 gives WU. W is WU expressed in row-reduced echelon form:

The resultant twist is obtained by calculating the reciprocal of W. The null space of W is:

$$t = \begin{bmatrix} -2/3 \\ 2/3 \\ 0 \\ 0 \\ 0 \\ -1/3 \end{bmatrix} \tag{64}$$

Thus after transposing t, applying the **flip** operation, and multiplying by the constant 3, the result is:

$$Result = \begin{bmatrix} 0 & 0 & 1 & 2 & -2 & 0 \end{bmatrix}$$
 (65)

This resultant twistmatrix can be interpreted as follows: the first triplet, ω , gives the unit vector of the axis of an allowed rotation, or in other words the vector of the unit magnitude angular velocity that the part could have. In this case it is [0 0 1] which means that rotation is allowed about an axis aligned with the PCS z-axis. The second triplet, v, contains information about the location of that axis in the PCS. Applying the point algorithm to this twistmatrix gives: P = (2,2,0). This is the location of the center of the hole feature (f1) in the PCS. The result is thus that the top-plate is able to rotate about the axis of the hole feature. This result can be predicted by physical reasoning. Thus the result is verified.

The MLA algorithm is now applied to determine the amount of possible rotation. This will depend on the length of the slot feature. It is assumed that the pin has a small clearance relative to the slot such that rotation can occur until the pin reaches the end of the slot. By choosing a slot with l = 0.4 units and d = 0.2 units, MLA gives the following results:

$$ppl = [0\ 0\ 0\ 0\ 1.4321]$$
 (66)

$$pnl = [0\ 0\ 0\ 0\ 1.4321] \tag{67}$$

ppl and pnl stand for part positive limit and part negative limit vectors respectively. Thus the result of the analysis is that the part can rotate 1.4231° in either direction about the axis of feature f1 located at the point (2,2,0).

The constraint analysis yields the following results:

$$T = \begin{bmatrix} 0 & 0 & 1 & 0 & -2 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$
 (68)

The reciprocal of T is the common wrenchspace. It is calculated to be:

$$Wrench = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 2 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$
 (69)

The first row indicates overconstraint of a combination of a displacement in the y-direction and a rotation about the z-axis. This shows that an error in the y-distance between the two features could cause difficulty in assembling the part onto the two pins. The second row indicates overconstraint of position in the z-direction, while the third and fourth rows show overconstraint of rotations about the x- and y-axes respectively. These constraints come about because both features include a planar mate between the parts as well as the mate to the pin.

6.5.2 Example 2: Fixturing of Sheet Metal Parts

Figure 6.8 below shows the three-part sheet metal assembly introduced in Chapter 2. The parts are assumed to be infinite in the x direction. This example will analyze the last step in the assembly sequence which is the assembly of part C to the right-hand side of the fixture and to the subassembly A-B which has previously been located on the fixture. The analysis is done in part coordinates marked by the origin point P3 and will consist of mating features 9 to 10 and 8 to 11.

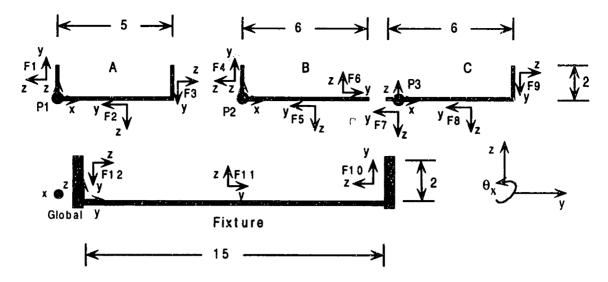


Figure 6.8 Three-Part Sheet Metal Assembly

All mating features are of the planar type and allow three degrees of freedom. The F matrices for feature 8 and 9 are:

$$F8 = \begin{bmatrix} -1 & 0 & 0 & 2.5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad F9 = \begin{bmatrix} 0 & 0 & 1 & 5 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{70}$$

Following the definitions for the twistspaces of planar features, the twistmatrices are:

The corresponding wrenchspaces are:

Concatenating the wrenchspaces into W and calculating the nullspace of W yields:

$$t = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{73}$$

After applying the transpose and flip operations, the result is:

$$Result = [0 \ 0 \ 0 \ 0 \ 1 \ 0]$$
 (74)

The result is an allowed translation in the y-direction which is the correct result. It is correct because no y-direction constraint was modeled explicitly. This can be changed in two ways. First one could place another feature on the edge of part C that mated to a vertical surface on the fixture, for example, that constrained the y-direction translation. The second would be to use the MLVs specified for one or the other of the existing planar mate features and constrain the y-direction translation to be zero. For example, if when using MLA, one specified the MLV for Feature 8 to be:

$$pl8 = [0.1\ 0\ 0\ 0\ 0\ 180] \tag{75}$$

The MLA program gives the result that the part cannot move.

The constraint analysis yields:

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$
 (76)

The reciprocal of this gives the wrenchspace intersection as:

$$Wrench = \begin{bmatrix} 0 & 0 & 0 & 0 & -1 & 0 \end{bmatrix}$$
 (77)

This is an overconstraint of the moment about the y-axis. This means that both features 8 and 9 are attempting to control the angular position of the part about the y-axis. This could create problems if for example the angle of the part was not 90°. One or the other of the planar surfaces would not be able to mate completely with the fixture. This result suggests a possible fixture redesign that would eliminate the overconstraint. One possibility is to use the design shown in Figure 6.9 below where the left and right hand sides of the fixture use thin rib features which control the x-position of the part without constraining the rotation about y.

Figure 6.9 Possible Fixture Redesign to Eliminate Overconstraint

The two examples presented in Section 6.5 were intended to be simple in order to demonstrate the basic principles of operation of the algorithms used in MLA. Chapter 8 will present a design and analysis example where the results are not so intuitively obvious.

6.6 Summary

This chapter described the theory behind the operation of MLA. An overview of MLA was given along with a discussion of its scope of application. The main algorithms used in MLA were then presented in both prose and mathematical form. Two simple examples were then worked in detail in order to illustrate the method.

Chapter 7 MLA Software Program

MLA has been implemented as a software program running on a Unix workstation. The program was written in C++ and takes advantage of the object-oriented programming style of C++. The software was written to work in conjunction with several other assembly analysis programs that have been written by other researchers in the past few years. Figure 7.1 shows a flowchart detailing the various software programs and the ways information is shared between them. The following list of references will point the reader to more detailed information on each program: DFC Editor [4], Tolerance Analysis [3], SPAS, PRED, LSG, and EDIT [6, 8].

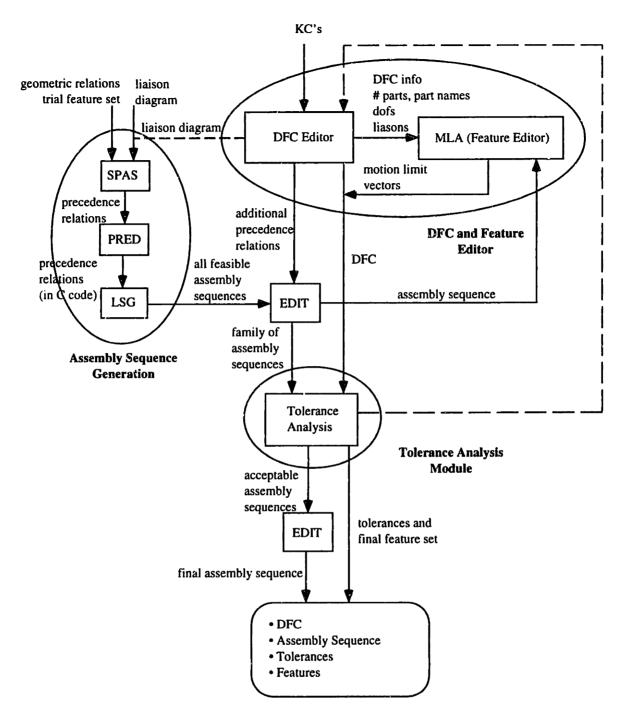


Figure 7.1 Flowchart of Software Used in AOD

This chapter will focus on two major themes: the steps required to use the software, and the flow of the program. With this information, the interested reader should be able to understand the C++ code listed in Appendix A and gain a detailed knowledge of the software.

7.1 Using the MLA Software

Using the MLA software, a user performs four basic steps:

- 1. Define the location and orientation of all parts in the assembly under study.
- 2. Choose assembly features to physically realize the connections between parts that are inferred from the DFC diagram of the assembly.
- 3. Define the location and orientation of these assembly features on each part.
- 4. Specify geometric parameters defining the feature and / or place numerical limits on the motions that each feature will allow acting individually.

Before the above steps can be performed, various files which are output by the DFC editor software and the assembly sequence software must be present in the local directory where MLA resides. Figure 7.2 is a cutout of the AOD flowchart presented in Chapter 2 and shows the various files, what type of information is contained in each, and where each comes from.

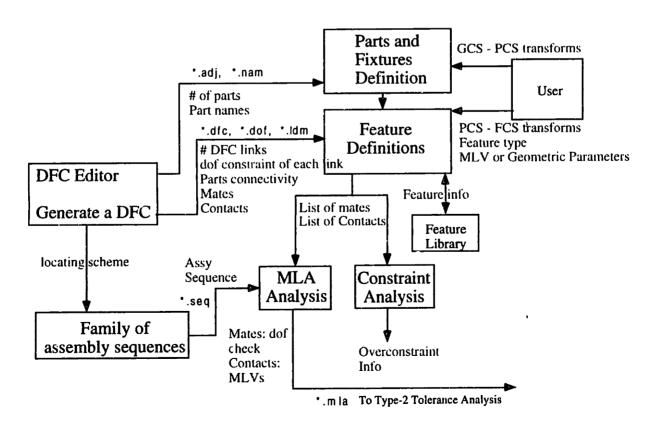


Figure 7.2 Filenames of Information Shared With MLA

Each file has the same name but different file extensions. The file name, referred to as <assyname>, is the name of the assembly and is input while using the DFC editor software. Similarly, all files that are output by MLA use <assyname> and add their own extensions according to which type of file is being output. The files that need to be present are:

<assyname>.dfc: This matrix contains information about how parts are connected together according to the chosen DFC design.

<assyname>.adj: The relevant information from this file is the number of parts in the assembly which is stored in the first line of the file.

<assyname>.nam: This is an ordered list of the names of each part. The ordering corresponds to the ordering of the rows in the <assyname>.dfc file, i.e. one row represents each part in the same order as the names are stored.

<assyname>.ldm: This is a matrix representation of the liason diagram. It is compared to the .dfc matrix to see which links between parts are mates and which are contacts.

<assyname>.seq: This is a list of the assembly sequence for the assembly. It consists of an ordered column of numbers corresponding to the id numbers of the parts in sequence.

With these files in place, the MLA software is invoked using a command line argument by typing:

mla <assyname>

The <assyname> command line argument is required in order for the program to find the <assyname>.* files listed above. If one is analyzing the same assembly many times, it is often helpful to create an input file for the assembly that automates the user input part of the program. This file is normally names <assyname>.in. It can be created using any

text editor, and should contain each user input that MLA asks for, in order, with each line of program input on a separate line in the file. Also, the user may want to redirect the program output from the screen to an output file such that the output may be printed or analyzed more easily. This can be done by just adding a command line argument redirecting the output to the file <assyname>.out. Thus if both the input and output file suggestions were taken, the program would be called using the following command:

mla <assyname> < <assyname>.in > <assyname>.out

The following sections will describe how to use the software and demonstrate the technique using sample screen dumps from the actual running software.

7.1.1 Example Assembly

The example assembly that will be analyzed is a simple four-part assembly in which the fourth part is not completely constrained and may be adjusted in an explicit manner. The first three parts (Base, Surface, and Pin) form a rigid immobile frame to which the fourth part (Rib) will be added. Assume that the pin is spring loaded such that assembly is possible. A simple schematic of the assembly is shown in Figure 7.3 below. The global coordinate frame is shown. A transcript of the computer session used to analyze this assembly is shown in Appendix B where all inputs and feature definitions are shown.

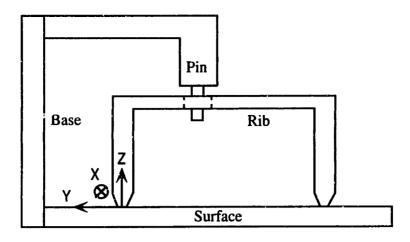


Figure 7.3 Schematic of Example Assembly

A DFC for this assembly is shown in Figure 7.4. Note that the numbers on the arrows locating part 'Rib' add to 6 which would seem to indicate that 'Rib' is fully constrained. This is not so because the addition of degrees of freedom is a vectorial add not a scalar adding of numbers. Thus because Z and θ_y are constrained by both links, the total number of independent dofs constrained for part 'Rib' is 4.

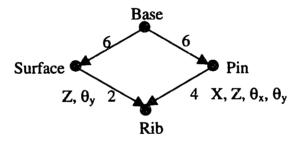


Figure 7.4 DFC for Example Assembly

The goal of this exercise is to calculate the motion capability of the fourth part called 'Rib'. A schematic drawing of part 'Rib' is shown in Figure 7.5 below. The small dashed rectangles in the top view represent the freedom zones of the thin rib – plane surface features. The sizes of the zones are determined by the designer and are set during feature specification. A number of criteria such as part interference, strength requirements, etc. may dictate the specification.

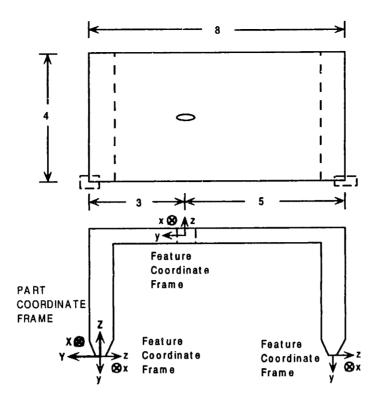


Figure 7.5 Schematic of Part 'Rib'

The PCS for this part is shown in bold. MLA will now be used to analyze the assembly.

7.1.2 Inputs

MLA asks for several types of input from the user in order to model the assembly. First one must define the location of each part in the assembly with reference to a global coordinate frame. The user will be prompted to input the information to create the 4x4 homogeneous transform that describes each part's position in the following format:

$$X Y Z \theta_x \theta_y \theta_z$$

These parameters locate the part coordinate system (PCS) in space by starting from the origin of the global frame, translating through 3-D space according to the X Y Z vector, and then rotating about the origin of the PCS according to θ_x θ_y θ_z to obtain the desired orientation. Figure 7.6 shows a screen dump of this first input step for the example assembly.

Figure 7.6 Definition of Part Locations

Next the user chooses assembly features to physically realize the mate and contact connections within the assembly. For each DFC link or contact the user is prompted for the number of features that will be used to realize the link. It then iterates through the feature selection loop that number of times to allow the user to choose each feature. The various feature choices are presented in a list and are categorized by the number of dofs they constrain. Once a feature is chosen, the user needs to place the feature on the part. This is done similar to placing the part relative to the global coordinate frame as in step 1. The user enters the X Y Z θ_x θ_y θ_z to create the transform from the PCS to the local feature coordinate system (FCS). The orientation of the FCS is important to the forthcoming calculations. The proper coordinate system directions for each feature type are shown in the figures in Chapter 5 where the features were defined. According to the type of feature chosen, the program then prompts for geometric or motion limit information regarding the feature. The program then carries out the calculations. The process of choosing and defining a feature is shown in Figure 7.7 Lelow. One of the rib to surface features for the part 'rib' is being defined.

```
Fig. 1. A promperty party countsee, and important and Language and important at Language and the second for each and the second for each and features and made and to use to realize important for the transplant of the transplant of the transplant of the second pages.

Fig. 2. A primarity was Primarity to be a few and transplant of the primarity of the primarity
```

Figure 7.7 Feature Selection and Definition

7.1.3 Assumptions

- The most basic assumption of the analysis is that the parts are all rigid bodies. Thus modeling of compliant parts is not now within the capability of the program.
- It is assumed that any allowed motion within the assembly does not break the physical contact between the parts.
- The act of assembly consists of adding a part to an assembly, possibly adjusting its' position, and fastening it immediately. Thus, motion limit vectors that are calculated by MLA describe the motion capability of a part that is being added to an assembly, before fastening, with respect to other immobile parts.

7.1.4 Calculations

The program performs calculations on two distinct types of entities, mates and contacts. The mates analysis is done to determine if the incoming mates to a part are sufficient to fully determine its location and constrain all of its degrees of freedom. Note that a single part may have DFC mate links which are incoming from other parts and DFC mate links which are outgoing that are connected when this part is used to locate another part. We are only concerned with the incoming links for each part. If a part is not fully constrained by its mates, the program calculates the **explicit** variation absorption ability of the part and outputs it as a motion limit vector (see Section 5.1.1). The calculation flow for each part in the assembly is illustrated in Figure 7.8.

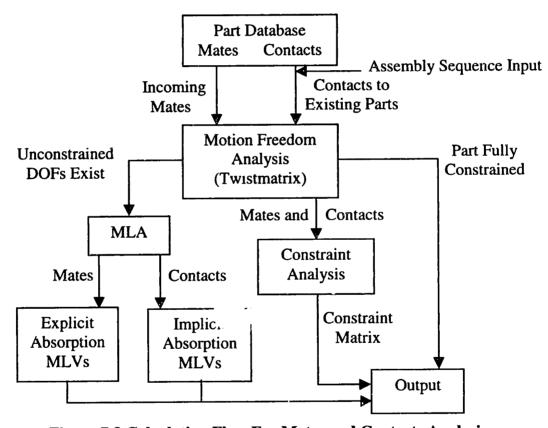


Figure 7.8 Calculation Flow For Mates and Contacts Analysis

Contact analysis is similar to mates analysis computationally, but has a different interpretation for the results. The goal of contact analysis is to determine the amount of variation that can be absorbed **implicitly** at the contact interfaces between parts (see Section 5.1.2). Similar to mates, a single part may have contacts to several other parts. Assembly sequence is here used as an input such that for each part, only the contacts to

parts that already have been assembled at the stage that the current part is assembled are analyzed. It is assumed that if a contact exists between the part being assembled and any part that has already been assembled, the contact is made when the current part is assembled. If unconstrained dofs result from the contact analysis, it does not mean the part can move. Remember the part is already constrained by the mates. The contacts analysis results in a motion limit vector that indicates the part's implicit variation absorption capability.

7.1.5 Outputs

MLA sends some outputs to the screen for the benefit of the user and stores other outputs in a file that will be accesses by the type II tolerance analysis program. Figure 7.9 shows the output for parts 'base' and 'surface' of the example assembly. These parts are rigidly placed and are identified as such by the program.

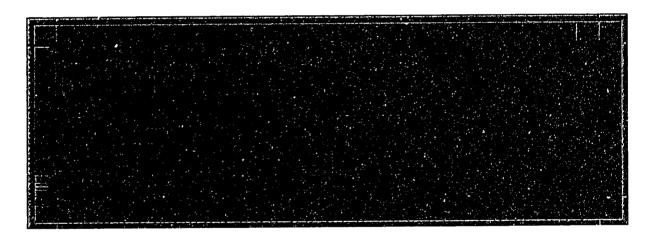


Figure 7.9 Sample Output for Fully Constrained Parts

The interesting part of MLA is the results it gives for parts which are not fully constrained. In the example assembly, this is the part 'Rib'. Figure 7.10 shows the program output regarding the part 'Rib'.

```
Part Trib is not completely constrained by the choice of mate.

Part Trib is not completely constrained by the choice of mate.

Resultant twistmatriz of mate analysis:

t:

0 0.1 -3 0 0.

0 0 0 0 1 0.

color 1 0.

trl = 1 vtl = 0.1 vol = -3.0 0

trl = 1 vtl = 0.1 vol = -3.0 0

trl = 0 vtl = 0.1 vol = -3.0 0

trl = 0 vtl = 0.1 vol = -3.0 0

Aris to vtl = 0.1 0

Points on axis 1 coincident? = 0

Ince test [] = 1

Axis 1:

Positive rotation occurs about the point: 3.95979 -3 3,85979

Feature number 1 is the 1 mitting feature for positive rotations.

Negative rotation occurs about the point: 3,95979 -3 3,95979

Feature number 1 is the 1 mitting feature for negative rotations.

Megative rotation occurs about the point: 3,95979 -3 3,95979

Feature number 1 is the 1 mitting feature for negative rotations.

Motion Limit Vectors for part rib:

ppl = [0 0.1 0.9 0.2.2184]

ppl = [0 0.1 0.9 0.2.2184]
```

Figure 7.10 Sample Output for an Under-Constrained Part

Because the mates in a DFC are nominally supposed to fully constrain the part, the program alerts the user that it is not the case for this part. The resultant twistmatrix and constraint matrix are then displayed. The resultant twistmatrix shows that two dofs are unconstrained. First is a rotation about the z-axis and second a translation in the y-direction. Note that all results are expressed in the coordinates of the part being analyzed. The constraint matrix shows overconstraint of a rotation about the y-axis. Next are the results of the test for coincident rotation points. A zero indicates false. Thus the program tests to see if the solutions lie in a linear space. The next output indicates that there is a linear space of solutions (1 = true). The location of the axis on that line and the resulting motion limit vectors for the part are then output. Note that the point about which negative rotation occurs need not be the same as for positive rotations because the freedom zones for each feature can be non-symmetric. In this case, the limits were defined symmetrically, and thus the points and the MLV are the same for both directions. Figure 7.11 shows a schematic of the solution situation. The center-line is the space of

linear solutions, and the point P is the location of a possible rotation axis. The output also indicates that feature 1 limits the part's rotation. The determination of which feature is feature 1 can be seen on an earlier output screen where the program lists each part and each feature on it. In this case, feature 1 is labeled f1 in Figure 7.11 below. Again the dashed rectangles represent the freedom zone of the feature.

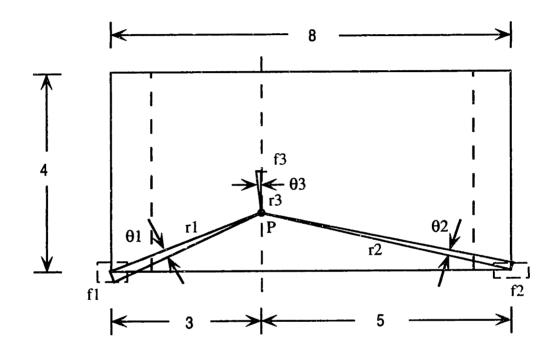


Figure 7.11 Schematic of a Linear Space of Solutions

MLA also outputs a file containing a list of positive direction MLVs for each part which are ordered the same as the parts in the *.nam and *.dfc files. The name of this output file is <assyname>.mla. It is read by the type II tolerance analysis software. This software is not coded to use bilateral limits on part motion. This is why only the positive direction MLVs are output and the motions are assumed symmetric.

7.2 Modeling and Analysis Considerations

Often a designer will specify multiple assembly features to realize the connection between two parts in an assembly. This is especially true if the parts are large or compliant. At times the designer intends only a portion of the features to act as the mates, while the others act as contacts. Because of the nature of the MLA calculation algorithm, this is not permitted. For any connection between parts, there may be multiple feature realizations of the mate or multiple feature realizations of the contact. What is not allowed is a mate and a contact between the same two parts. If the designer wishes to model such a situation, separate analyses involving only mates and only contacts must be done. The analysis of the mates will provide a check of the location scheme. The analysis of the contacts will yield MLVs associated with the motions allowed by the combined actions of the contacts which may be used afterward in doing Type-2 tolerance analysis. Both analyses will yield overconstraint information.

The user should note that because a computer number contains a finite number of digits, the resultant twistmatrix or constraint analyses may return results which have numbers such as 3.245234e-17 as one of the entries of the matrix. These types of numbers should be regarded as zeros. Robustness routines have been added to the computer code such that numbers like this are converted to zeros before further calculation, but these routines do not change the displayed number.

7.3 Class Structure

The MLA software is written in C++ and takes advantage of the object-oriented programming style that is available when using that language. Thus the database of information about the parts in the assembly is arranged into classes. Each class can be considered as a user defined data type with its own properties and definitions of operations. This section will present the member variables of each class and describe how each is used to store data describing an assembly. Each class also contains a few member functions which are used to perform such operations as printing all of the member variables to the screen. These functions will not be described here. The detailed class definitions can be found in Appendix A.

It was noted earlier that MATLABTM software is used to make many of the matrix calculations done in MLA. Thus, the definitions of the data structure for storing matrices adhere to the conventions of that software. Details of the 'Matrix' data structure can be found in the "External Interface Guide" manual that accompanies all versions of MATLABTM.

7.3.1 Part Class

The Part class is the main data structure used to store data about parts in the assembly. It is defined in the file Mpart.h. This class contains the following member variables:

- name: character string containing the name of the part which was input using the DFC
 editor.
- id: integer containing the id number of the part. It is assigned in the same order as the parts are listed by row in the *.dfc matrix
- ppl, npl: these are the motion limit vectors for the part. ppl is what is placed in the
 *.mla output file. They are vectors of doubles of length 6.
- Mates, Contacts: these are arrays of type Feature which contain the list of features used to realize the mate and contact links of the DFC for this part.
- Mates2, Mates3, Contacts2, Contacts3: these are also arrays of Features which are
 used by the linear and planar search algorithms. They contain the same features as
 Mates and Contacts, but the features are described in a displaced coordinate system.
 (see section 6.3.4.1.2)

7.3.2 Feature Class

The Feature class encapsulates all of the data regarding a mating feature into one compact data structure. It is defined in the file Mfeature.h. This class contains the definitions of all 17 types of features and is used to construct a feature of the correct type when it is chosen by the user. This class contains the following member variables:

• T: this is a variable of type TwistMat which stores the twistmatrix representation of the feature.

- F: this is a variable of type Transform which stores the 4x4 homogeneous transform that describes the location and orientation of this feature on the part that it has been placed on.
- type: this is a character denoting the feature type. The letter stored here corresponds to the letter used to choose this feature type from the list of features during feature definition.
- pa: this is a integer array of length two which stores the id numbers of the parts that are connected by this feature. If the feature is used to describe a mate, the ordering of the numbers in pa correspond to the DFC location scheme, i.e. pa[0] is at the tail of the arrow and locates the part whose id is stored in pa[1].
- pl, nl: these are arrays of type double of length six. They store the MLVs describing the motion limits of this feature.

7.3.3 TwistMat Class

The TwistMat class stores a single twistmatrix. Its structure is driven by the data storage conventions of MATLAB™ as described earlier. It is defined in the file MTwistMat.h. This class contains the following member variables:

- t: this is a pointer to a Matrix data structure which is used to store the numerical elements of the twistmatrix.
- Rr1, rr2, rr3, tr1, tr2, tr3: These are integers which are used to store the row number of each row in the twistmatrix which represents a rotation or translation. Thus rr1 stands for 'rotation row 1' and tr1 stands for 'translation row 1'. Thus if the first row in a twistmatrix is identified to represent a rotation, the value of rr1 would be 1. It is important to note that the default value of each of these variables is 7. This is used by the program to know how many rotations and translations are represented in each twistmatrix. Because there are a maximum of six rows in a twistmatrix, these variables can never be assigned a value of 7 by the row identification routine. Thus these variable are used to check the existence of a particular dof.

• om1, om2, om3, vo1, vo2, vo3, vt1, vt2, vt3: these variables are pointers to Matrix data structures and are used to subdivide the twistmatrix into individual 1x3 vectors. 'om' represents 'omega' an angular velocity vector, 'vo' represents 'velocity omega' which is the translational velocity that occurs when a rigid body rotates, and 'vt' represents 'velocity translation' which are just translational velocity vectors.

7.3.4 Transform Class

The Transform class stores the elements of a 4x4 homogeneous matrix transform using the MATLAB™ Matrix data structure. It is defined in the file Mtransform.h. This class contains only one member variable:

 f: this is a pointer to a Matrix data structure which is used to store the numerical elements of the transform matrix. This matrix is constructed from the X Y Z θ_x θ_y θ_z input parameters described above.

7.3.5 Array Class

The Array class was written by Matt Wall [39] and is used extensively throughout the program. This class allows the user to create array of almost any data type. For example, in the Part class, Mates is an array of objects of type Feature made using the Array class. The benefit of the array class is that it contains certain useful member functions. First, the arrays are dynamically allocating, meaning that an array allocates more memory to itself as it needs it. This feature makes memory management more efficient. Second, the array keeps track of how many entries it has and can produce that information through the member function size(). The use of this class is straightforward, but crucial to understanding the operation of the entire MLA program. The interested reader is referred to the listing of Array.h in Appendix A which provides instructions for using the class.

7.4 Logical Program Flow

The routines used in MLA to get input from the user are very simple, and the interested reader should refer to Appendix A for a listing of the code. This section will present a

flowchart that describes the calculation routines. The chart will then be described in the context of the example that is being used throughout this chapter.

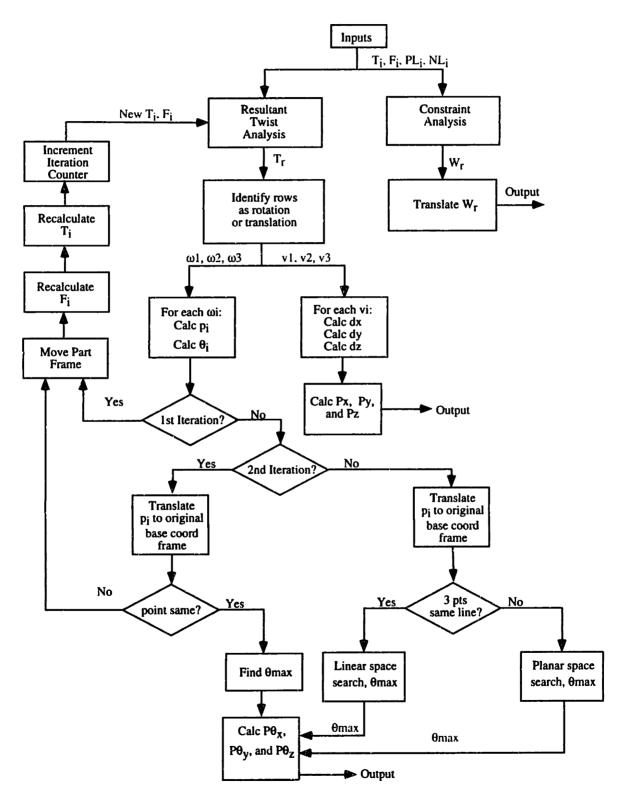


Figure 7.12 Flowchart Diagraming Calculation Routines

The inputs at the top of the page can either be a list of mates or contacts on a part. If it is a list of mates, the list contains only the mates that locate the part, i.e. incoming arrows in the DFC. If the list contains contacts, only the contacts with parts that already exist in the assembly are included in the list. The example of this chapter contains no contacts. The algorithm is run for each part in the assembly. The input list in each case would be a list of the assembly features associated with incoming arrows to the part. Note that part 'Base' has no incoming arrows, and this analysis is skipped as the part is assumed fully constrained. For the part 'Rib', we know that there is a linear space of solutions. Thus after two iterations through the resultant twistmatrix loop, the solutions from the two iterations would be compared to see if the resultant rotation point was the same. This was shown in Figure 7.10 where the program outputs: 'Points on axis 1 coincident? = 0'. The points were not coincident, and thus the loop is executed again to obtain a third solution. The three solutions are then compared to see whether or not they lie on one line. In this case they do, and the linear space is searched for the axis that allows the maximum rotation. If the three points were not collinear, a planar space search would be executed. Flowcharts outlining the two searches are shown in Figure 7.13 below.

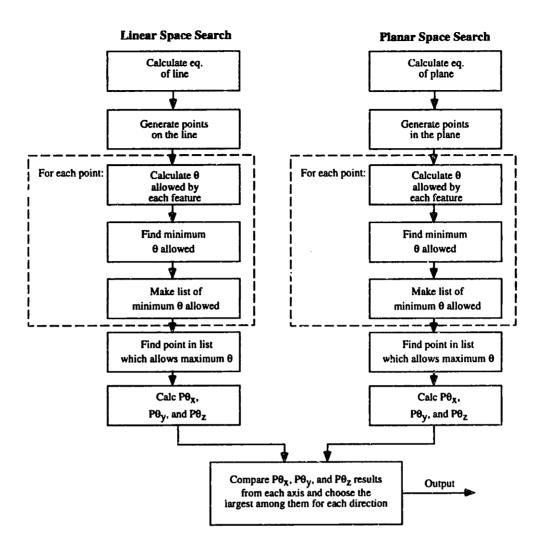


Figure 7.13 Flowchart of Linear and Planar Solution Space Searches

The maximum translation calculation and constraint analysis are performed only once for each part, and the results output.

7.5 Summary

This chapter documented the operation of the MLA software program. The nature of the program and the environment it operates in were explained. Step by step instructions on how to use the program were given and illustrated with an example. The assumptions used in the calculations were also elucidated. Next an explanation of the data structures used to store part and feature data was given. The logical flow of the calculation routine finished the chapter.

Chapter 8 Design and Analysis Example

This chapter will detail a real world design and analysis example. The purpose is to demonstrate the usefulness of MLA within the framework of assembly oriented design and present the types of results that are possible. The example is a subassembly of an aircraft horizontal stabilizer. This example is taken from several papers and theses on assembly design. For a more detailed explanation of the assembly and the business context that surrounds it, the reader is referred to [1, 2, 4, 40, 41].

8.1 Need

Traditional methods of assembling large compliant assemblies such as aircraft fuselage or wing sections and automotive ody parts use large dedicated fixtures to establish the relative positions of parts before fastening takes place. Manufacturers of these types of assemblies are seeking assembly methods that are more flexible and able to assemble a wide variety of assemblies while using a minimum of expensive fixtures. This type of process is commonly referred to in industry as 'fixture-less' or 'hole-to-hole' assembly. A fixture-less process relies on pre-fabricated assembly features on the parts for locating each part in the assembly. MLA has the potential to assist assembly designers in realizing more flexible processes by allowing rapid design and analysis of assembly features. Using MLA together with the other tools of AOD as presented in Chapter 2, represents a powerful method of rapid and inexpensive generation and evaluation of assembly designs.

8.2 Horizontal Stabilizer Assembly

Figure 8.1 below shows an exploded view of the main torque box of a horizontal stabilizer [1].

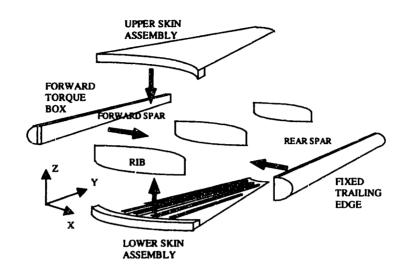


Figure 8.1 Exploded View of the Main Torque Box of a Horizontal Stabilizer

The main torque box carries the majority of the loads within the assembly. The top-level requirements for this assembly are set by strength and aerodynamics specifications. This study will focus on evaluation of the assembly of the upper and lower skin subassemblies. Because these two assemblies are nearly identical, the upper skin assembly will be used here with the results applying equally to both assemblies. Figure 8.2 shows an upper skin subassembly from below and identifies the various parts. These assemblies range from 30 to 60 feet long and 5 to 10 feet wide.

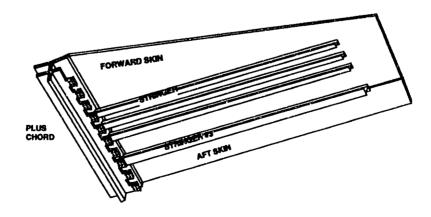


Figure 8.2 Bottom View of an Upper Skin Subassembly [40]

The plus chord is the main structural component used to attach the upper skin to the other components of the main torque box and to the aircraft. The skin itself consists of two parts, the forward and aft skin, which are spliced together by stringer #3. Stringers 4-11 (some of which are shown) attach to the plus chord and the forward skin, while stringers 1 and 2 (not shown) attach to the plus chord and aft skin.

8.3 Assembly Oriented Design Approach

8.3.1 KC Identification

Following the AOD approach outlined in Chapter 2, the first task is to define the KCs for the assembly. Again these KCs are driven by strength and aerodynamics requirements. A full exposition of the KC definition process for this assembly is given in [41]. Figure 8.3 shows the two most critical KCs for the upper skin assembly.

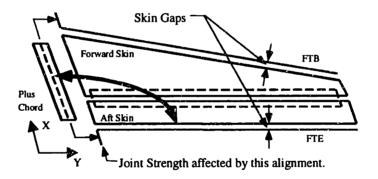


Figure 8.3 Assembly Key Characteristics for the Upper Skin Assembly [41]

8.3.2 DFC Design

The DFC design for the assembly should be focused on establishing a locating scheme that will deliver the KCs. For this assembly, that means that the assembly designer should seek to explicitly control the angle between the plus chord and the edge of the aft skin as well as the positions of both skins such that that gaps between them and other components are consistent. Mantripragada and Whitney [4] detail the evaluation of different DFCs for this assembly. In this case, the DFC that was most desirable for

delivering the KCs was not realizable because of assembly constraints. The final DFC chosen for this assembly is shown in Figure 8.4.

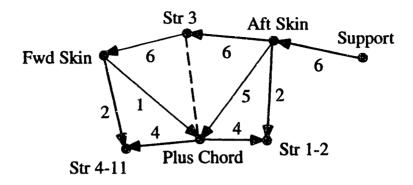
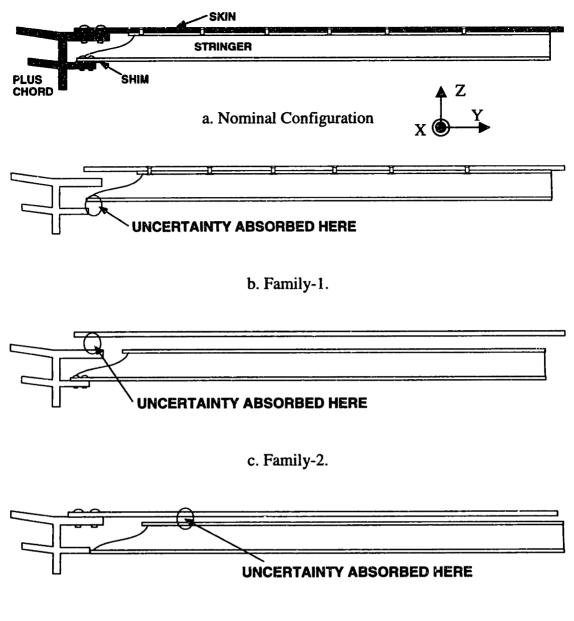


Figure 8.4 Datum Flow Chain for the Upper Skin Assembly

8.3.3 Variation Absorption Strategy and Feature Design

This is the point in the process where MLA is applied. A set of assembly features has to be chosen to realize the logic represented in the DFC. This choice of features is not trivial and often not obvious. Iteration is generally required in order to arrive at a suitable set of assembly sequence and features. The choice of assembly features is affected by the assembly sequence in that one needs to use assembly features on parts that already exist in the assembly to locate the next part to be added. The exception to this statement is a process where a fixture is used to hold the part in place and only contacts exist between it and other parts. There are two major decisions involved with choosing assembly features. First is choosing the features that will realize the important mates in the DFC and contribute directly to delivering the KCs. Second is choosing where, how, and in what direction(s) to absorb variation within the assembly. This is done by either defining an adjustable dof in the DFC and using explicit variation absorption, or by defining contact features that allow some ambiguity of feature position or part shape at certain locations without affecting the integrity of the resulting assembly. Figure 8.5 presents illustrations of different options for absorbing uncertainty within the upper skin assembly.



d. Family-3.

Figure 8.5 Possible Locations for Variation Absorption [41]

The choice of the location in which to absorb uncertainty affects not only feature design, but also assembly sequence. Thus the options are called families because each represents a subset (family) of all possible assembly sequences that could be used to build the assembly. Casting Figure 8.5 in the light of assembly feature selection indicates that for Family 1, the designer would choose some type of contact feature to realize the connection between the plus chord and the stringers. One possibility is a simple planar joint, called a plate-plate lap joint in Chapter 5 (feature #9). This choice would allow

absorption of positional variations in both the fore-aft and inboard-outboard directions as well as variations in the angular orientation between the two parts. The configuration of Family 1 was chosen for the connection between the plus chord and stringer 3. This joint is a contact as shown on the DFC (Fig. 8.4). A shim may be required in some cases to absorb z-direction variation.

Family 3 was chosen for the connection between the plus chord and the other stringers. This family can be implemented in an interesting way, which exploits the properties of the slotted-hole feature to both provide locational constraint and allow implicit variation absorption. The feature set that was chosen was to use a slotted-hole feature (feature 4. Ch. 5) to connect the stringer to the skin, and a pin-hole feature with no planar mate (feature 5, Ch. 5) to connect the stringer and the plus chord. Lets take a closer look at the constraint situation for these stringers in order to understand why this feature set was chosen. First, one must consider that this assembly is 30 to 60 feet long in the ydirection. Thermal expansion would present problems in trying to align full size hole features if they were used along the length of the stringers. Thus slots were chosen in order to absorb the thermal expansion induced variation while maintaining the ability to mate the features. Family 3 dictates that the feature between the plus chord and the stringer should be the primary locator for the stringer with variation being absorbed at the stringer-skin interface. It is simple to absorb x- and y-direction variation at this interface, but z-direction absorption would require shimming the entire length of the stabilizer. Thus, the stringer / plus-chord interface feature was chosen such that z-direction variation could be absorbed at this site. The pin-hole feature (Feature 6, page 52) constrains four degrees of freedom. In the part coordinates of the stringer these are x, y, θ_x , and θ_y (see Figs. 8.1 and 8.5). Z-direction variation is absorbed using a shim at this interface. This mate is made first and leaves two unconstrained dofs. When the slotted-hole feature between the stringer and the skin is mated, the rotation about the pin axis, θ_z , is constrained. The z-direction is also established when the plane of the stringer and skin meet. Thus the stringer is fully located. Because of the planar mate between the skin and stringer, the slotted-hole feature also constrains θ_x and θ_y as well as x. Thus, technically the assembly is overconstrained in three dofs. Nonetheless, the critical variation

absorption directions of y and z are not overconstrained. Both absorptions are of the implicit type. The overconstraint serves as a caution to the designer and does not mean that the location scheme is flawed.

Figure 8.6 shows a diagram of the parts and a set of assembly features that will deliver the key dimensions of the assembly.

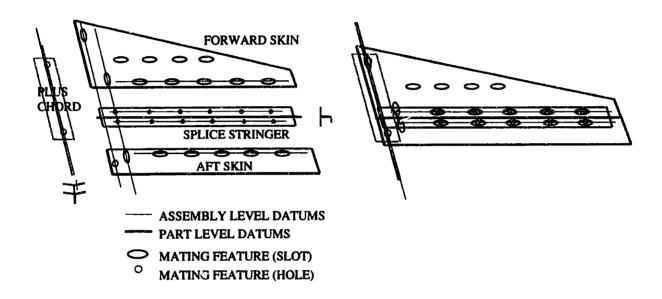


Figure 8.6 Assembly Feature Design for the Upper Skin Assembly

Note that only one of the slotted-hole features used along the length of the stabilizer serves as the mate. The others are contacts and exist to provide strength to the assembly until final fastening takes place.

The strategy of using slots for compliant part assembly is employed extensively in the car industry for locating body panels to fixtures. It has not seen much use in the aircraft industry because the holes in the skins are machined. Machining a slotted hole is time consuming and hard on the machining tools. The auto industry pays no penalty for making slotted holes because all of the holes are punched into the sheet metal. It is just as easy to punch a slotted hole as a round one. Slotted-hole features have one more benefit, they are able to accommodate the rigid body assumption made for MLA analysis.

In other words, parts are not necessarily rigid as assumed by MLA. By using slotted holes, a MLA analysis will be more realizable in the real world because the slotted-holes relax the assumption.

8.3.4 MLA Analysis

The upper skin assembly with the feature design shown above was modeled and analyzed using the MLA software. A transcript of the analysis session is contained in Appendix B. The analysis verified that each of the parts in the assembly was fully constrained by its mates. This analysis is a check of the designer's DFC logic and feature selection. If no explicit variation absorption capability is intended, the program should verify that the parts are fully constrained. If not, the designer should re-examine the DFC and / or feature selection. MLA verified that the logic of the DFC and the features shown in Figure 8.6 were sufficient to fully constrain the assembly. The constraint analysis returns results that prove useful for helping a designer spot critical assembly areas where form or other tolerances may need to be specified in order to ensure assemblability. In this case, the constraint matrix returned for each part was identical.

Constraint =
$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$
 (78)

The first row indicates overconstraint of translations in the z-direction (top-bottom of wing). The second row indicates overconstraint of rotations about the x-axis (fore-aft), and the third row indicates overconstraint of rotations about the y-axis (inboard-outboard). These three constraints arise beczuse the parts meet together on a plane. An assumption of MLA is that parts cannot separate in the assembly direction (z). Thus, because all features involve the planar mate, more than one feature is constraining the planes to stay mated. The rotational constraints arise from the same planar mate and work to keep the planes mated. In terms of tolerancing, the result indicates that if the planar surfaces are not flat, mating in the z-direction may be difficult. Thus the designer may consider placing a flatness specification on the undersides of the skin, on the

stringer, or on the flanges of the plus chord. A straightness specification on the stringers is another option. Because real aircraft parts are compliant and violate the rigid body assumption of MLA, such specifications would likely not be necessary. If a gap occurred because the surfaces did not completely mate, a clamp would be used to force the parts together before fastening. This also depends on the size of the gap. Nonetheless, the analysis has alerted the designer to this critical interface and leaves the action to the judgement of the designer.

The contact analysis is used to give an indication of how much variation, and in what directions, can be absorbed implicitly at the locations of the contact features. As noted in Section 7.2, contact and mate links between the same two parts cannot be analyzed concurrently. Because the feature design contains both mates and contacts between some parts, a separate contact analysis was performed (not shown in Appendix B). The numerical results of a contact analysis often depend on the amount of motion freedom that the designer allows each feature when specifying the MLV for the feature.

There were two main types of contacts analyzed for this assembly. The first were slotted-hole features to connect the skins to stringers. The length of the slot was chosen to be 0.14" while the diameter of the hole and the width of the slot were 0.10". The analysis yielded the following MLVs for the slotted holes:

$$ppl = pnl = [0\ 0.02\ 0\ 0\ 0]$$
 (79)

The result was symmetric and indicated that 0.02 units of variation in the y-direction could exist at each feature location without prohibiting assembly. A plate-plate lap joint was used to absorb variation between the stringers and the lower scallop of the plus chord (See Figure 8.5b). The limits on x and y-direction motion were chosen to be symmetric with a value of 0.2 units. The stringer was sized to be an inch wide, which makes the inputs Pxc and Nxc used to specify a planar feature equal to 0.5 each. The resultant MLV for each stringer was:

$$ppl = pnl = [0.2 \ 0.2 \ 0 \ 0 \ 0.966]$$
 (80)

The results are as expected, indicating absorption freedom in the x-y plane and the ability to absorb approximately 1° of angular error. The constraint analysis of the contacts gave the same results as were obtained in the mates analysis.

For the case of the upper skin assembly, the slotted-hole features between the skins and stringers, and between the skins and plus chord cannot be permanent features. Aircraft structures are riveted. The integrity of a riveted joint depends on the fact that the riveted fastener completely fills the hole it is inserted into. Thus the slotted-hole features are temporary and are used just to locate the parts prior to final fastening. They must be drilled out and replaced with a round hole that can be riveted. This situation places a limit on the size of slot that can be specified. A schematic is shown in Figure 8.7 below.

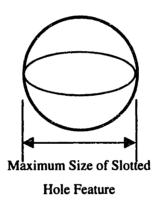


Figure 8.7 Size Limitations on Temporary Slotted-Hole Features

8.4 Conclusion

The horizontal stabilizer assembly analysis demonstrated some of the useful results that are provided to the assembly designer when using MLA. First is a check of the feature design chosen to implement a DFC. The designer can see right away whether or not the feature set is sufficient to constrain the parts as desired. Second the results of the constraint analysis highlight possible assemblability problem areas where the designer may want to take appropriate tolerancing action to ensure a proper assembly. Third, the program provides variation absorption information. In this case the results were fairly intuitive. MLA's value is not completely derived by the direct results it provides. MLA provides information about an assembly to the Type-2 tolerance analysis algorithm which

is crucial to obtaining tolerance analysis results. MLA was designed to work as part of an integrated system, and the value of its results are best based on the value of the results that the system as a whole provides.

8.5 Summary

This chapter presented a design and analysis example showing how MLA analysis is used within the assembly oriented design framework and demonstrated the type of results that are obtained. The example assembly was the upper skin assembly which is part of an aircraft horizontal stabilizer assembly. The assembly oriented design approach was used to define and analyze this assembly with particular attention to the MLA analysis component. The results were presented and discussed.

Chapter 9 Conclusions

This chapter will present conclusions regarding the content of this thesis along with an explanation of possibilities for future work.

9.1 Conclusions

This thesis has presented the Assembly Oriented Design approach to designing mechanical assemblies with specific emphasis on the theory and implementation of Motion Limit Analysis. The value of MLA was shown to be its ability to provide information about the constraint of parts in an assembly subject to the choice of assembly features used to connect the various parts. The MLA software program was written to implement the theories of MLA and works in concert with several other analysis programs which together provide the tools necessary to implement the AOD approach.

9.2 Future Work

Chapter 2 mentioned that the next major step in advancing the AOD method is to be able to analyze compliant parts. A possibility for doing this would be to use a simple FEA type model to relate the positions of the feature-level coordinate frames to the coordinate frame of the part they reside on. The same method could be used to relate the part-level coordinate frames to the global coordinate frame. A linear model should be used in order to enable the application of linear algebra to the analysis.

A way to improve the implementation of MLA would be to link it to the database of a CAD system such that the 4x4 transforms locating parts in the GCS and features in the PCS could be generated automatically. This would eliminate the most cumbersome and time consuming step of using MLA. Along the same lines, all of the modules used for AOD analysis presented in Figure 7.1 could be linked together such that all programs are run from a single window and all data transfer is automatic.

Currently, the AOD method does not make use of all of the capabilities of MLA. For example, MLA can analyze incremental motions allowed by small clearances between feature sizes. This analysis may have application to the quantification of assemblability between two parts. Thus the abilities of MLA could point toward directions for further development of the AOD methodology.

The mechanism synthesis method of Konkar [33, 34] may be used to automatically synthesize feature sets from the results of a Type-2 optimal control analysis. The theory would need to be extended from the 6 types of assembly features defined by Konkar to the 17 types defined in MLA.

Lastly, there is the possibility of implementing an approach taken by Kramer [29] to determine how a chosen set of assembly features will position a part with respect to others in an assembly. Kramer calls this degree of freedom analysis. The computerized method allows the user to develop an assembly model by connecting parts together via traditional kinematic joints (revolute, linear,...). The analysis is done one part at a time and involves virtually mating one of the joints, and revolving or translating the part along the unconstrained dofs until it is in a position to mate another of its joints. This is done until either all joints are mated or all 6 dofs have been constrained. This approach could be used to determine if a chosen set of assembly features could position the parts they are placed on in the desired manner and with the desired accuracy.

Bibliography / References

- [1] R. Mantripragada, T. Cunningham, and D. E. Whitney, "Assembly Oriented Design: A new approach to designing assemblies," presented at the Fifth IFIP WG 5.2 Workshop on Geometric Modeling in Computer-Aided Design, Airlie, Virginia, USA, 1996.
- [2] R. Mantripragada and D. E. Whitney, "State Transition Models of Mechanical Assembly Processes," submitted to IEEE Transactions on Robotics and Automation, 1997.
- [3] R. Mantripragada and D. E. Whitney, "Modeling and Controlling Variation Propagation in Mechanical Assemblies using State Transition Models," submitted to IEEE Transactions on Robotics and Automation, 1997.
- [4] R. Mantripragada and D. E. Whitney, "The Datum Flow Chain: A Systematic Approach to Assembly Design and Modeling," submitted to Research in Engineering Design, 1997.
- [5] R. Paul (1981) Robot Manipulators: Mathematics, Programming, and Control. The Computer Control of Robot Manipulators, The MIT Press, Cambridge, MA, pp. 9-40.
- [6] T.L. De Fazio and D.E. Whitney, "Simplified Generation of All Mechanical Assembly Sequences," *IEEE Journal of Robotics and Automation*, Vol. RA-3, No. 6, December 1987, pp. 640-658.

- [7] A. Bourjault, "Contribution a une approche methodologique de l'assemblage autornatise: Elaboration automatique des sequences operatiores," Thesis to obtain Grade de Docteur es Sciences Physiques at L'Universite de Franche-Comte, Nov. 1984.
- [8] D.F. Baldwin, T.E. Abell, M.M. Lui, T.L. De Fazio, and D.E. Whitney, "An Integrated Computer Aid for Generating and Evaluating Assembly Sequences for Mechanical Products," *IEEE Transactions on Robotics and Automation*, Vol. 7, No. 1, February 1991, pp. 78-94.
- [9] D.E. Whitney, O. L. Gilbert, and M. Jastrzebski, "Representation of Geometric Variations Using Matrix Transforms for Statistical Tolerance Analysis in Assemblies," Research in Engineering Design, (1994) 6: 191-210.
- [10] J.J. Shah and M. Mantyla (1995) Parametric and Feature-Based CAD/CAM: Concepts, Techniques, and Applications, John Wiley and Sons, New York, NY.
- [11] J.J. Shah and B.C. Zhang, "Attributed Graph Model for Geometric Tolerancing," ASME: Advances in Design Automation, DE-Vol. 44-2, 1992, pp. 133-140.
- [12] J.J. Shah and M.T. Rogers, "Assembly Modeling as an Extension of Feature-Based Design," Research in Engineering Design, (1993) 5: 218-237.
- [13] T.L. De Fazio, et. al., "A Prototype of Feature-Based Design for Assembly," Journal of Mechanical Design, December 1993, Vol. 115, pp. 723-734.
- [14] D.C. Anderson and T.C. Chang, "Geometric Reasoning in Feature-Based Design and Process Planning," *Computers and Graphics*, Vo. 14, No. 2, pp. 225-235.
- [15] C. Mascle, T. Jabbour, and R. Maranzana, "Assembly Features for Mechanical Product Data," <u>Proceedings of the 1997 IEEE International Symposium on Assembly and Task Planning</u>, Marina del Rey, CA, August 1997, pp. 218-223.
- [16] M.G. Kim and C.H. Wu, "Modeling of Part-Mating Strategies for Automating Assembly Operations for Robots," *IEEE Transations on Systems, Man, and Cybernetics*, Vol. 24, No. 7, July 1994, pp. 1065-1074.
- [17] M.G. Kim and C.H. Wu, "A Formal Part Mating Model for Generating Compliance Control Strategies of Assembly Operations," in the <u>Proceedings of The 1990 IEEE International Conference on Systems, Man, and Cybernetics</u>, Los Angeles, CA, Nov. 1990.

- [18] R. S. Ball, <u>A Treatise on the Theory of Screws</u>, Cambridge University Press, Cambridge, 1900.
- [19] K.J. Waldron, "The Constraint Analysis of Mechanisms," *Journal of Mechanisms*, Vol. 1, (1966) pp. 101-114.
- [20] K.H. Hunt, Kinematic Geometry of Mechanisms, Clarendon Press, Oxford, 1978.
- [21] J.E. Baker, "Screw System Algebra Applied to Special Linkage Configurations," Mechanism and Machine Theory, Vol. 15, 1980, pp. 255-265.
- [22] J.E. Baker, "On Relative Freedom Between Links in Kinematic Chains with Cross-Jointing," *Mechanism and Machine Theory*, Vol. 15, 1980, pp. 397-413.
- [23] J.E. Baker, "On Mobility and Relative Freedoms in Multiloop Linkages and Structures," *Mechanism and Machine Theory*, Vol. 16, No. 6, 1981, pp. 583-597.
- [24] T.H. Davies, "Kirchoff's Circulation Law Applied to Multi-Loop Kinematic Chains," *Mechanism and Machine Theory*, Vol. 16, 1981, pp. 171-183.
- [25] T.H. Davies, "Mechanical Networks I: Passivity and Redundancy," *Mechanism and Machine Theory*, Vol. 18, No. 2, 1983, pp. 95-101.
- [26] T.H. Davies, "Mechanical Networks II: Formulae for the Degrees of Mobility and Redundancy," *Mechanism and Machine Theory*, Vol. 18, No. 2, 1983, pp. 103-106.
- [27] T.H. Davies, "Mechanical Networks III: Wrenches on Circuit Screws," Mechanism and Machine Theory, Vol. 18, No. 2, 1983, pp. 107-112.
- [28] L. Woo and F. Freudenstein, "Application of Line Geometry to Theoretical Kinematics and the Kinematic Analysis of Mechanical Systems," *Journal of Mechanisms*, Vol. 5, 1970, pp. 417-460.
- [29] G.A. Kramer, Solving Geometric Constraint Systems: A Case Study in Kinematics, MIT Press, Cambridge, MA, 1992.
- [30] S. Hirai and H. Asada, "Kinematics and Statics of Manipulation Using the Theory of Polyhedral Convex Cones," submitted to the *International Journal of Robotics Research*.
- [31] M.T. Mason and J.K. Salisbury (1985) Robot Hands and the Mechanics of Manipulation, MIT Press, Cambridge, MA.
- [32] M.S. Ohwovoriole and B. Roth, "An Extension of Screw Theory," *Journal of Mechanical Design*, Vol. 103, October 1981, pp. 725-735.

- [33] R. Konkar, "Incremental Kinematic Analysis and Symbolic Synthesis of Mechanisms," Ph.D. Dissertation, Stanford University, Stanford, CA 94305, USA, 1993.
- [34] R. Konkar and M. Cutkosky, "Incremental Kinematic Analysis of Mechanisms," Journal of Mechanical Design, Vol. 117, December 1995, pp. 589-596.
- [35] D. J. Lee, et.al., "Key Characteristics for Agile Product Development and Manufacturing," Agility Forum 4th Annual Conference Proceedings, March 1995.
- [36] A. Thornton, et.al., "Agile Customer-Supplier Relations in Design and Manufacture of Complex Mechanical Assemblies," Agility Forum 5th Annual Conference Proceedings, March 1996.
- [37] D.J. Lee and A. Thornton, "The Identification and Use of Key Characterisics in the Product development Process," ASME 8th Design Theory and Methodology Conference, Aug. 1996.
- [38] T.A. Sweder and J. Pollock, "Full Vehicle Variability Modeling," SAE Paper, Reprint #942334, SAE Inc., 1994.
- [39] M.B. Wall, "Array.h" C++ array class written February 9, 1996, MIT. Email: mbwall@mit.edu.
- [40] T.W. Cunningham, et.al, "Definition, Analysis, and Planning of a Flexible Assembly Process," ASME Japan/USA Symposium on Flexible Automation, Boston, USA, July 1996.
- [41] T.W. Cunningham, "Chains of Function Delivery: A Role for Product Architecture in Concept Design," Ph.D. Dissertation, Massachusetts Institute of Technology, February 1998.
- [42] J.M. Kriegel, "Exact Constrain Design," *Mechanical Engineering*, May 1995, pp. 88-90.

Appendix A Software Documentation

The appendix contains a listing of the source code of the various C++ files used to implement MLA. The programs are linked to some library files included with the MATLABTM software package. Please refer to the book "External Interface Guide" which is included with each version of MATLABTM software. Table A.1 lists the various files used in the program and gives a brief description of each. The files are contained on the accompanying disk.

File Description			
Header Files			
Array.h	Declaration and definition of Array class - contains user instructions		
Mpart.h	Declaration and definition of Part class		
Mfeature.h	Declaration and definition of Feature class		
MTwistMat.h	Declaration and definition of TwistMat class		
Mtransform.h	Declaration and definition of Transform class		
analyze.h	Declaration of MLA analysis routine		
MLAfunct.h	Declaration of various functions called by analyze.C		

C++ Files	
MLA.C	Main routine used for I/O
analyze.C	MLA calculation routine
MLAfunct.C	Definition of various functions called by analyze.C

Table A. 1 Listing of MLA Software Files

Appendix B Analysis Transcripts

This appendix gives transcripts of the computer sessions that were used to do the examples presented in Chapters 7 and 8. This is provided for the reader who is interested in examining all of the inputs, or is trying to learn to use the software.

Chapter 7 Analysis Example

```
Number of Parts = 4
Enter the translation and rotation parameters of the
global to part frame transform for part 'base' in the following form:
X Y Z THETAX THETAY THETAZ: 0 0 0 0 0
Enter the translation and rotation parameters of the
global to part frame transform for part 'surface' in the following
form:
X Y Z THETAX THETAY THETAZ: 0 0 0 0 0
Enter the translation and rotation parameters of the
global to part frame transform for part 'rib' in the following form:
X Y Z THETAX THETAY THETAZ: 0 0 0 0 0
Enter the translation and rotation parameters of the
global to part frame transform for part 'pin' in the following form:
X Y Z THETAX THETAY THETAZ: -2 2 6 0 0 0
DFC link 1 connects parts 'base' and 'surface'.
It constrains 6 degrees of freedom.
How many features do you want to use to realize this link? 1
Please choose feature 1 from the following list:
Choice Feature Name
                                            DOF constrained
       Prismatic Peg / Prismatic Hole
                                            6 XYZTxTyTz
                                           5 X Y Z Тх Ту
       Plate Pin in Through Hole
                                            5 X Z Tx Ty Tz
       Primatic Peg / Prismatic Slot
       Plate Pin in Slotted Hole
                                            4 X Z Tx Ty
```

```
4 X Z Tx Ty
        Round Peg / Prismatic Slot
6
                                             4 X Y Tx Ty
f
        Round Peg / Through or Blind Hole
                                             4 X Y Tx Ty
        Threaded Joint
q
        Elliptical Ball and Socket
                                             4 X Y Z Tx
h
        Plate-Plate Lap Joint
                                             3 Z Tx Tv
                                             3
                                               XYZ
        Spherical Joint
j
        Plate Pin in Oversize Hole
                                             3
                                               Z Tx Ty
k
        Elliptical Ball in Cylindrical Trough 3 X Z Tx
1
        Thin Rib / Plane Surface
                                             2
                                                Z Ty
m
                                             2
                                               Z Tx
        Ellipsoid on Plane Surface
n
        Spherical Ball in Cylindrical Trough 2 X Z
0
                                             2 X Ty
        Peg in Slotted Hole
р
        Spheroid on Plane Surface
                                             1 Z
p
```

Choice: a

Please input the data to create the homogeneous transform from the part To feature frame for this feature. Note, this is the transform from the part-level coordinate frame of the part 'base' to the location of this feature on that part.

X Y Z theta-x theta-y theta-z: 0 0 0 0 0

This feature type has no allowable motion.

DFC link 2 connects parts 'surface' and 'rib'.

It constrains 2 degrees of freedom.

How many features do you want to use to realize this link? 2

Please choose feature 1 from the following list:

Choice	Feature Name	DO	F constrained
а	Prismatic Peg / Prismatic Hole	6	X Y Z Tx Ty Tz
b	Plate Pin in Through Hole	5	Х Ү Z Тх Ту
C	Primatic Peg / Prismatic Slot	5	X Z Tx Ty Tz
đ	Plate Pin in Slotted Hole	4	X Z Tx Ty
e	Round Peg / Prismatic Slot	4	X Z Tx Ty
£	Round Peg / Through or Blind Hole	4	х ү тх ту
g	Threaded Joint	4	х ү тх ту
h	Elliptical Ball and Socket	4	X Y Z Tx
i	Plate-Plate Lap Joint	3	Z Tx Ty
j	Spherical Joint	3	XYZ
k	Plate Pin in Oversize Hole	3	Z Tx Ty
1	Elliptical Ball in Cylindrical Trough	3	X Z Tx
m	Thin Rib / Plane Surface	2	Z Ty
n	Ellipsoid on Plane Surface	2	Z Tx
0	Spherical Ball in Cylindrical Trough	2	X Z
р	Peg in Slotted Hole	2	х ту
q	Spheroid on Plane Surface	1	Z

Choice: m

Please input the data to create the homogeneous transform from the part to

feature frame for this feature. Note, this is the transform from the part-level coordinate frame of the part 'surface' to the location of this feature on that part.

X Y Z theta-x theta-y theta-z: 0 0 0 180 0 0

Please input the limits on X axis motion for this feature

Px Nx: .2 .2

Please input the limits on Y axis motion for this feature

Py Ny: .3 .3

Please input the angular limits (in degrees) on rotation about the X

and Z axes.

PthetaX NthetaX: 30 30 PthetaZ NthetaZ: 180 180

Please choose feature 2 from the following list:

Prismatic Peg / Prismatic Hole 6 X Y Z Tx Ty Tz Plate Pin in Through Hole 5 X Y Z Tx Ty Tz Primatic Peg / Prismatic Slot 5 X Z Tx Ty Tz Plate Pin in Slotted Hole 4 X Z Tx Ty Ex Round Peg / Prismatic Slot 4 X Z Tx Ty Ex Round Peg / Through or Blind Hole 4 X Y Tx Ty Ex Ty Ex Ty Ex	Choice	Feature Name	DO	F constrained
C Primatic Peg / Prismatic Slot 5 X Z Tx Ty Tz d Plate Pin in Slotted Hole 4 X Z Tx Ty e Round Peg / Prismatic Slot 4 X Z Tx Ty f Round Peg / Through or Blind Hole 4 X Y Tx Ty g Threaded Joint 4 X Y Tx Ty h Elliptical Ball and Socket 4 X Y Z Tx i Plate-Plate Lap Joint 3 Z Tx Ty j Spherical Joint 3 X Y Z k Plate Pin in Oversize Hole 3 Z Tx Ty l Elliptical Ball in Cylindrical Trough 3 X Z Tx m Thin Rib / Plane Surface 2 Z Ty n Ellipsoid on Plane Surface 2 Z Tx o Spherical Ball in Cylindrical Trough 2 X Z p Peg in Slotted Hole 2 X Ty	a	Prismatic Peg / Prismatic Hole	6	X Y Z Tx Ty Tz
d Plate Pin in Slotted Hole 4 X Z Tx Ty e Round Peg / Prismatic Slot 4 X Z Tx Ty f Round Peg / Through or Blind Hole 4 X Y Tx Ty g Threaded Joint 4 X Y Tx Ty h Elliptical Ball and Socket 4 X Y Z Tx i Plate-Plate Lap Joint 3 Z Tx Ty j Spherical Joint 3 X Y Z k Plate Pin in Oversize Hole 3 Z Tx Ty l Elliptical Ball in Cylindrical Trough 3 X Z Tx m Thin Rib / Plane Surface 2 Z Ty n Ellipsoid on Plane Surface 2 Z Tx o Spherical Ball in Cylindrical Trough 2 X Z p Peg in Slotted Hole 2 X Ty	b	Plate Pin in Through Hole	5	X Y Z Tx Ty
e Round Peg / Prismatic Slot 4 X Z Tx Ty f Round Peg / Through or Blind Hole 4 X Y Tx Ty g Threaded Joint 4 X Y Tx Ty h Elliptical Ball and Socket 4 X Y Z Tx i Plate-Plate Lap Joint 3 Z Tx Ty j Spherical Joint 3 X Y Z k Plate Pin in Oversize Hole 3 Z Tx Ty l Elliptical Ball in Cylindrical Trough 3 X Z Tx m Thin Rib / Plane Surface 2 Z Ty n Ellipsoid on Plane Surface 2 Z Tx o Spherical Ball in Cylindrical Trough 2 X Z p Peg in Slotted Hole 2 X Ty	С	Primatic Peg / Prismatic Slot	5	X Z Tx Ty Tz
f Round Peg / Through or Blind Hole 4 X Y Tx Ty g Threaded Joint 4 X Y Tx Ty h Elliptical Ball and Socket 4 X Y Z Tx i Plate-Plate Lap Joint 3 Z Tx Ty j Spherical Joint 3 X Y Z k Plate Pin in Oversize Hole 3 Z Tx Ty l Elliptical Ball in Cylindrical Trough 3 X Z Tx m Thin Rib / Plane Surface 2 Z Ty n Ellipsoid on Plane Surface 2 Z Tx o Spherical Ball in Cylindrical Trough 2 X Z p Peg in Slotted Hole 2 X Ty	đ	Plate Pin in Slotted Hole	4	X Z Tx Ty
Threaded Joint 4 X Y Tx Ty h Elliptical Ball and Socket 4 X Y Z Tx i Plate-Plate Lap Joint 3 Z Tx Ty j Spherical Joint 3 X Y Z k Plate Pin in Oversize Hole 3 Z Tx Ty l Elliptical Ball in Cylindrical Trough 3 X Z Tx m Thin Rib / Plane Surface 2 Z Ty n Ellipsoid on Plane Surface 2 Z Tx o Spherical Ball in Cylindrical Trough 2 X Z p Peg in Slotted Hole 2 X Ty	e	Round Peg / Prismatic Slot	4	Х Z Тх Ту
h Elliptical Ball and Socket 4 X Y Z Tx i Plate-Plate Lap Joint 3 Z Tx Ty j Spherical Joint 3 X Y Z k Plate Pin in Oversize Hole 3 Z Tx Ty l Elliptical Ball in Cylindrical Trough 3 X Z Tx m Thin Rib / Plane Surface 2 Z Ty n Ellipsoid on Plane Surface 2 Z Tx o Spherical Ball in Cylindrical Trough 2 X Z p Peg in Slotted Hole 2 X Ty	£	Round Peg / Through or Blind Hole	4	Х Ү Тх Ту
i Plate-Plate Lap Joint 3 Z Tx Ty j Spherical Joint 3 X Y Z k Plate Pin in Oversize Hole 3 Z Tx Ty l Elliptical Ball in Cylindrical Trough 3 X Z Tx m Thin Rib / Plane Surface 2 Z Ty n Ellipsoid on Plane Surface 2 Z Tx o Spherical Ball in Cylindrical Trough 2 X Z p Peg in Slotted Hole 2 X Ty	g	Threaded Joint	4	Х Ү Тх Ту
j Spherical Joint 3 X Y Z k Plate Pin in Oversize Hole 3 Z Tx Ty l Elliptical Ball in Cylindrical Trough 3 X Z Tx m Thin Rib / Plane Surface 2 Z Ty n Ellipsoid on Plane Surface 2 Z Tx o Spherical Ball in Cylindrical Trough 2 X Z p Peg in Slotted Hole 2 X Ty	h	Elliptical Ball and Socket	4	X Y Z Tx
k Plate Pin in Oversize Hole 3 Z Tx Ty l Elliptical Ball in Cylindrical Trough 3 X Z Tx m Thin Rib / Plane Surface 2 Z Ty n Ellipsoid on Plane Surface 2 Z Tx o Spherical Ball in Cylindrical Trough 2 X Z p Peg in Slotted Hole 2 X Ty	i	Plate-Plate Lap Joint	3	Z Tx Ty
1 Elliptical Ball in Cylindrical Trough 3 X Z Tx m Thin Rib / Plane Surface 2 Z Ty n Ellipsoid on Plane Surface 2 Z Tx o Spherical Ball in Cylindrical Trough 2 X Z p Peg in Slotted Hole 2 X Ty	j	Spherical Joint	3	X Y Z
m Thin Rib / Plane Surface 2 Z Ty n Ellipsoid on Plane Surface 2 Z Tx o Spherical Ball in Cylindrical Trough 2 X Z p Peg in Slotted Hole 2 X Ty	k	Plate Pin in Oversize Hole	3	Z Tx Ty
n Ellipsoid on Plane Surface 2 Z Tx o Spherical Ball in Cylindrical Trough 2 X Z p Peg in Slotted Hole 2 X Ty	1	Elliptical Ball in Cylindrical Trough	3	X Z Tx
o Spherical Ball in Cylindrical Trough 2 X Z p Peg in Slotted Hole 2 X Ty	m	Thin Rib / Plane Surface	2	Z Ty
p Peg in Slotted Hole 2 X Ty	n	Ellipsoid on Plane Surface	2	Z Tx
-	0	Spherical Ball in Cylindrical Trough	2	X Z
g Spheroid on Plane Surface 1 Z	p	Peg in Slotted Hole	2	Х Ту
-	đ	Spheroid on Plane Surface	1	Z

Choice: m

Please input the data to create the homogeneous transform from the part to feature frame for this feature. Note, this is the transform from the part-level coordinate frame of the part 'surface' to the location of this feature on that part.

X Y Z theta-x theta-y theta-z: 0 -8 0 180 0 0

Please input the limits on X axis motion for this feature

Px Nx: .15 .15

Please input the limits on Y axis motion for this feature

Py Ny: .35 .35

Please input the angular limits (in degrees) on rotation about the X

and Z axes.

PthetaX NthetaX: 15 15 PthetaZ NthetaZ: 180 180

DFC link 3 connects parts 'pin' and 'rib'.

It constrains 4 degrees of freedom.

How many features do you want to use to realize this link? 1

Please choose feature 1 from the following list:

Choice	Feature Name DOF constra	
a	Prismatic Peg / Prismatic Hole	6 XYZTxTyTz
b	Plate Pin in Through Hole	5 ХУ ZТхТ у
С	Primatic Peg / Prismatic Slot	5 X Z Тх Ту Тz
đ	Plate Pin in Slotted Hole	4 X Z Tx Ty
e	Round Peg / Prismatic Slot	4 X Z Tx Ty

```
£
        Round Peg / Through or Blind Hole
                                             4 X Y Tx Ty
                                             4 X Y Tx Ty
        Threaded Joint
g
h
        Elliptical Ball and Socket
                                             4 X Y Z Tx
        Plate-Plate Lap Joint
                                             3 Z Tx Ty
i
                                             3 X Y Z
        Spherical Joint
j
                                                Z Tx Ty
        Plate Pin in Oversize Hole
                                             3
k
        Elliptical Ball in Cylindrical Trough 3 X Z Tx
1
        Thin Rib / Plane Surface
                                             2 Z Ty
m
        Ellipsoid on Plane Surface
                                                Z Tx
n
                                             2 X Z
        Spherical Ball in Cylindrical Trough
0
        Peg in Slotted Hole
                                             2 X Tv
р
        Spheroid on Plane Surface
                                             1 Z
a
```

Choice: d

Please input the data to create the homogeneous transform from the part to feature frame for this feature. Note, this is the transform from the part-level coordinate frame of the part 'pin' to the location of this feature on that part.

X Y Z theta-x theta-y theta-z: 4 -5 -2 0 0 0

Input the length (1) of the slot and diameter (d) of the peg. 1 d: .4 .2

DFC link 4 connects parts 'base' and 'pin'.

It constrains 6 degrees of freedom.

How many features do you want to use to realize this link? 1

Please choose feature 1 from the following list:

Choice	Feature Name	DO	F constrained
a	Prismatic Peg / Prismatic Hole	6	X Y Z Tx Ty Tz
b	Plate Pin in Through Hole	5	X Y Z Tx Ty
С	Primatic Peg / Prismatic Slot	5	X Z Tx Ty Tz
đ	Plate Pin in Slotted Hole	4	X Z Tx Ty
е	Round Peg / Prismatic Slot	4	X Z Tx Ty
£	Round Peg / Through or Blind Hole	4	Х Ү Тх Ту
g	Threaded Joint	4	X Y Tx Ty
h	Elliptical Ball and Socket	4	X Y Z Tx
i	Plate-Plate Lap Joint	3	Z Tx Ty
j	Spherical Joint	3	XYZ
k	Plate Pin in Oversize Hole	3	Z Tx Ty
1	Elliptical Ball in Cylindrical Trough	3	X Z Tx
m	Thin Rib / Plane Surface	2	Z Ty
n	Ellipsoid on Plane Surface	2	Z Tx
0	Spherical Ball in Cylindrical Trough	2	X Z
р	Peg in Slotted Hole	2	х ту
q	Spheroid on Plane Surface	1	Z

Choice: a

Please input the data to create the homogeneous transform from the part to feature frame for this feature. Note, this is the transform from the part-level coordinate frame of the part 'base' to the location of this feature on that part.

X Y Z theta-x theta-y theta-z: 2 -3 6 0 0 0

This feature type has no allowable motion.

```
List of parts and their variables -----
Part Name: base
id#: 1
Motion Limit Vectors:
positive limits: 0 0 0 0 0 0
negative limits: 0 0 0 0 0 0
Part Transform:
f =
1 0 0 0
0 1 0 0
0 0 1 0
0 0 0 1
Mates:
T =
t:
0 0 0 0 0 0
F =
f =
1 0 0 0
0 1 0 0
0 0 1 0
0 0 0 1
type = a
pa = 1, 2
pl = 0, 0, 0, 0, 0, 0
n1 = 0, 0, 0, 0, 0, 0
T =
t:
0 0 0 0 0 0
F =
f =
1 0 0 2
0 1 0 -3
0 0 1 6
0 0 0 1
type = a
pa = 1, 4
p1 = 0, 0, 0, 0, 0, 0
n1 = 0, 0.0, 0, 0, 0
Contacts:
Part Name: surface
id#: 2
Motion Limit Vectors:
positive limits: 0 0 0 0 0 0
```

negative limits: 0 0 0 0 0 0

```
Part Transform:
f =
1 0 0 0
0 1 0 0
0 0 1 0
0 0 0 1
Mates:
T =
t:
0 0 0 0 0 0
F =
f =
1 0 0 0
0 1 0 0
0 0 1 0
0 0 0 1
type = a
pa = 1, 2
pl = 0, 0, 0, 0, 0, 0
n1 = 0, 0, 0, 0, 0, 0
T =
t:
1 0 0 0 0 0
0 0 -1 0 0 0
0 0 0 1 0 0
0 0 0 0 -1 0
F =
f =
1 0 0 0
0 -1 0 0
0 0 -1 0
0 0 0 1
type = m
pa = 2, 3
p1 = 0.2, 0.3, 0, 30, 0, 180
n1 = 0.2, 0.3, 0, 30, 0, 180
T =
t:
1 0 0 0 0 8
0 0 -1 0 0 0
0 0 0 1 0 0
0 0 0 0 -1 0
F =
f =
1 0 0 0
0 -1 0 -8
0 0 -1 0
```

```
0 0 0 1
type = m
pa = 2, 3
p1 = 0.15, 0.35, 0, 15, 0, 180
n1 = 0.15, 0.35, 0, 15, 0, 180
Contacts:
Part Name: rib
id#: 3
Motion Limit Vectors:
positive limits: 0 0 0 0 0 0
negative limits: 0 0 0 0 0 0
Part Transform:
f =
1 0 0 0
0 1 0 0
0 0 1 0
0 0 0 1
Mates:
T ==
t:
1 0 0 0 0 0
0 0 -1 0 0 0
0 0 0 1 0 0
0 0 0 0 -1 0
F =
f =
1 0 0 0
0 -1 0 0
0 0 -1 0
0 0 0 1
type = m
pa = 2, 3
p1 = 0.2, 0.3, 0, 30, 0, 180
n1 = 0.2, 0.3, 0, 30, 0, 180
T =
t:
1 0 0 0 0 8
0 0 -1 0 0 0
0 0 0 1 0 0
0 0 0 0 -1 0
F =
f =
1 0 0 0
0 -1 0 -8
0 0 -1 0
0 0 0 1
```

```
type = m
pa = 2, 3
p1 = 0.15, 0.35, 0, 15, 0, 180
n1 = 0.15, 0.35, 0, 15, 0, 180
T =
t:
0 0 1 -3 -2 0
000010
F =
f =
1 0 0 2
0 1 0 -3
0 0 1 4
0 0 0 1
type = d
pa = 4, 3
pl = 0, 0.1, 0, 0, 0, 3.14159
nl = 0, 0.1, 0, 0, 0, 3.14159
Contacts:
Part Name: pin
id#: 4
Motion Limit Vectors:
positive limits: 0 0 0 0 0 0
negative limits: 0 0 0 0 0 0
Part Transform:
f =
1 0 0 -2
0 1 0 2
0 0 1 6
0 0 0 1
Mates:
T =
t:
0 0 1 -5 -4 0
0 0 0 0 1 0
F =
f =
1 0 0 4
0 1 0 -5
0 0 1 -2
0 0 0 1
type = d
pa = 4, 3
p1 = 0, 0.1, 0, 0, 0, 3.14159
nl = 0, 0.1, 0, 0, 0, 3.14159
T =
```

```
t:
0 0 0 0 0 0
F =
f =
1 0 0 4
0 1 0 -5
0 0 1 0
0 0 0 1
type = a
pa = 1, 4
p1 = 0, 0, 0, 0, 0, 0
nl = 0, 0, 0, 0, 0, 0
Contacts:
Mates Analysis -----
Part 'base' is fully constrained by its mates.
Mates Motion Limit Vectors for part 'base':
ppl = [0 0 0 0 0 0]
pn1 = [0 \ 0 \ 0 \ 0 \ 0]
Part 'surface' is fully constrained by its mates.
Constraint Matrix:
The single feature constrains all dofs
Mates Motion Limit Vectors for part 'surface':
pp1 = [0 \ 0 \ 0 \ 0 \ 0]
pnl = [0 \ 0 \ 0 \ 0 \ 0]
Part 'rib' is not completely constrained by the choice of mates.
You may want to choose another feature set.
Resultant twistmatrix of mate analysis:
t:
0 0 1 -3 0 0
0 0 0 0 1 0
rr1 = 0 \text{ om}1 = 0 \text{ 0 1 vo}1 = -3 \text{ 0 0}
tr1 = 1 vt1 = 0 1 0
Constraint Matrix:
t:
0 0 0 0 1 0
tr1 = 0 vt1 = 0 1 0
Points on axis 1 coincident? = 0
line test 1? = 1
Axis 1:
Positive rotation occurs about the point: 3.95979 -3 3.95979
Feature number 1 is the limiting feature for positive rotations.
Negative rotation occurs about the point: 3.95979 -3 3.95979
Feature number 1 is the limiting feature for negative rotations.
```

```
Nates Motion Limit Vectors for part 'rib':

ppl = [0 0.1 0 0 0 2.2184]

pnl = [0 0.1 0 0 0 2.2184]

Part 'pin' is fully constrained by its mates.

Constraint Matrix:

The single feature constrains all dofs

Mates Motion Limit Vectors for part 'pin':

ppl = [0 0 0 0 0 0]

pnl = [0 0 0 0 0 0]
```

Chapter 8 Analysis Example

```
Number of Parts = 7
Enter the translation and rotation parameters of the global to part
frame transform for part 'Support' in the following form:
X Y Z THETAX THETAY THETAZ: 0 0 0 0 0 0
Enter the translation and rotation parameters of the global to part
frame transform for part 'Aft_Skin' in the following form:
X Y Z THETAX THETAY THETAZ: -4 0 .22 0 0 0
Enter the translation and rotation parameters of the global to part
frame transform for part 'Str_3' in the following form:
X Y Z THETAX THETAY THETAZ: 13.233 3.5 .22 0 0 0
Enter the translation and rotation parameters of the global to part
frame transform for part 'Fwd_Skin' in the following form:
X Y Z THETAX THETAY THETAZ: 17 0 .22 0 0 0
Enter the translation and rotation parameters of the global to part
frame transform for part 'Str_4_11' in the following form:
X Y Z THETAX THETAY THETAZ: 17.226 1 .22 0 0 0
Enter the translation and rotation parameters of the global to part
frame transform for part 'Plus_Chord' in the following form:
X Y Z THETAX THETAY THETAZ: 0 0 0 0 0
Enter the translation and rotation parameters of the global to part
frame transform for part 'Str_1_2' in the following form:
X Y Z THETAX THETAY THETAZ: 5 1 .22 0 0 0
```

DFC link 1 connects parts 'Support' and 'Aft_Skin'.

It constrains 6 degrees of freedom.

How many features do you want to use to realize this link? 1

```
Please choose feature 1 from the following list:
                                           DOF constrained
Choice Feature Name
а
       Prismatic Peg / Prismatic Hole
                                           6 X Y Z Tx Ty Tz
       Plate Pin in Through Hole
                                           5 X Y Z Tx Ty
b
       Primatic Peg / Prismatic Slot
                                          5 X Z Tx Ty Tz
C
                                          4 X Z Tx Ty
       Plate Pin in Slotted Hole
d
                                           4 X Z Tx Ty
       Round Peg / Prismatic Slot
e
       Round Peg / Through or Blind Hole 4 X Y Tx Ty
f
a
       Threaded Joint
                                          4 XYTxTy
                                       4 X Y Z Tx
      Elliptical Ball and Socket
h
                                          3 Z Tx Ty
      Plate-Plate Lap Joint
```

```
Spherical Joint
j
                                             3 X Y Z
        Plate Pin in Oversize Hole
                                             3 Z Tx Ty
k
1
        Elliptical Ball in Cylindrical Trough 3 X Z Tx
                                                Z Ty
        Thin Rib / Plane Surface
                                             2
m
n
        Ellipsoid on Plane Surface
                                             2
                                                z r_x
        Spherical Ball in Cylindrical Trough 2 X Z
0
        Peg in Slotted Hole
                                             2 X Ty
р
        Spheroid on Plane Surface
                                             1 Z
q
```

Choice: a

Please input the data to create the homogeneous transform from the part to feature frame for this feature. Note, this is the transform from the part-level coordinate frame of the part 'Support' to the location of this feature on that part.

X Y Z theta-x theta-y theta-z: -4 0 .22 0 0 0

This feature type has no allowable motion.

DFC link 2 connects parts 'Aft_Skin' and 'Str_3'.

It constrains 6 degrees of freedom.

How many features do you want to use to realize this link? 2

Please choose feature 1 from the following list:

Choice	Feature Name	DOF constrained			
a	Prismatic Peg / Prismatic Hole	6	X Y Z Tx Ty Tz		
b	Plate Pin in Through Hole	5	х у Z тх ту		
c	Primatic Peg / Prismatic Slot	5	X Z Tx Ty Tz		
đ	Plate Pin in Slotted Hole	4	х Z Тх Ту		
е	Round Peg / Prismatic Slot	4	Х Z Тх Ту		
f	Round Peg / Through or Blind Hole	4	Х Ү Тх Ту		
g	Threaded Joint	4	х ү тх ту		
h	Elliptical Ball and Socket	4	X Y Z Tx		
i	Plate-Plate Lap Joint	3	Z Tx Ty		
j	Spherical Joint	3	X Y Z		
k	Plate Pin in Oversize Hole	3	Z Tx Ty		
1	Elliptical Ball in Cylindrical Trough	3	X Z Tx		
m	Thin Rib / Plane Surface	2	Z Ty		
n	Ellipsoid on Plane Surface	2	Z Tx		
0	Spherical Ball in Cylindrical Trough	2	ХZ		
р	Peg in Slotted Hole	2	х ту		
đ	Spheroid on Plane Surface	1	Z		

Choice: d

Please input the data to create the homogeneous transform from the part to feature frame for this feature. Note, this is the transform from the part-level coordinate frame of the part 'Aft_Skin' to the location of this feature on that part.

X Y Z theta-x theta-y theta-z: 18.495 4 0 0 0 90

Input the length (1) of the slot and diameter (d) of the peg. $1 \ d: .26 \ .25$

Please choose feature 2 from the following list:
Choice Feature Name DOF constrained

```
Prismatic Peg / Prismatic Hole 6 X Y Z Tx Ty Tz
a
       Plate Pin in Through Hole
b
                                           5 X / Z Tx Tv
       Primatic Peg / Prismatic Slot
                                          5 X Z Tx Ty Tz
C
       Plate Pin in Slotted Hole
                                          4 X Z Tx Ty
d
                                          4 X Z Tx Ty
e
       Round Peg / Prismatic Slot
       Round Peg / Prismatic Slot 4 X 2 TX Ty
Round Peg / Through or Blind Hole 4 X Y Tx Ty
f
       Threaded Joint
                                           4 X Y Tx Ty
g
      Elliptical Ball and Socket
                                           4 XYZTx
h
       Plate-Plate Lap Joint
i
                                           3 Z Tx Ty
i
       Spherical Joint
                                           3 X Y Z
k
       Plate Pin in Oversize Hole
                                           3 Z Tx Ty
       Elliptical Ball in Cylindrical Trough 3 X Z Tx
1
       Thin Rib / Plane Surface
                                  2 Z Ty
m
       Ellipsoid on Plane Surface
n
                                           2 Z Tx
       Spherical Ball in Cylindrical Trough 2 X Z
0
       Peg in Slotted Hole
                                           2 X Ty
р
       Spheroid on Plane Surface
                                           1 Z
q
```

Choice: d

Please input the data to create the homogeneous transform from the part to feature frame for this feature. Note, this is the transform from the part-level coordinate frame of the part 'Aft_Skin' to the location of this feature on that part.

X Y Z theta-x theta-y theta-z: -94.579 240 0 0 0 25.6

Input the length (1) of the slot and diameter (d) of the peg. 1 d: .14 .1

DFC link 3 connects parts 'Str_3' and 'Fwd_Skin'.

It constrains 6 degrees of freedom.

How many features do you want to use to realize this link? 2

Please choose feature 1 from the following list:

Choice	Feature Name	DOF constrained				
a	Prismatic Peg / Prismatic Hole	6	X Y Z Tx Ty Tz			
b	Plate Pin in Through Hole	5	х у с тх ту			
C	Primatic Peg / Prismatic Slot	5	X Z Tx Ty Tz			
d	Plate Pin in Slotted Hole	4	X Z Tx Ty			
e	Round Peg / Prismatic Slot	4	X Z Tx Ty			
f	Round Peg / Through or Blind Hole	4	х ү тх ту			
g	Threaded Joint	4	х у тх ту			
h	Elliptical Ball and Socket	4	X Y Z Tx			
i	Plate-Plate Lap Joint	3	Z Тх Ту			
j	Spherical Joint	3	XYZ			
k	Plate Pin in Oversize Hole	3	Z Тх Ту			
1	Elliptical Ball in Cylindrical Trough	3	X Z Tx			
m	Thin Rib / Plane Surface	2	Z Ty			
n	Ellipsoid on Plane Surface	2	Z Tx			
0	Spherical Ball in Cylindrical Trough	2	X Z			
p	Peg in Slotted Hole	2	Х Ту			
q	Spheroid on Plane Surface	1	2			

Choice: d

Please input the data to create the homogeneous transform from the part to feature frame for this feature. Note, this is the transform from the part-level coordinate frame of the part 'Str_3' to the location of this feature on that part.

X Y Z theta-x theta-y theta-z: 2.361 .5 0 0 0 90

Input the length (1) of the slot and diameter (d) of the peg. $1 \ d: .26 \ 25$

Please choose feature 2 from the following list:

Choice	Feature Name	DO	F constrained
a	Prismatic Peg / Prismatic Hole	6	X Y Z Tx Ty Tz
b	Plate Pin in Through Hole	5	X Y Z Tx Ty
С	Primatic Peg / Prismatic Slot	5	X Z Tx Ty Tz
đ	Plate Pin in Slotted Hole	4	X Z Tx Ty
e	Round Peg / Prismatic Slot	4	X Z Tx Ty
£	Round Peg / Through or Blind Hole	4	х ү тх ту
g	Threaded Joint	4	Х Ү Тх Ту
h	Elliptical Ball and Socket	4	X Y Z Tx
i	Plate-Plate Lap Joint	3	Z Tx Ty
j	Spherical Joint	3	X Y Z
k	Plate Pin in Oversize Hole	3	Z Tx Ty
1	Elliptical Ball in Cylindrical Trough	3	X Z Tx
m	Thin Rib / Plane Surface	2	Z Ty
n	Ellipsoid on Plane Surface	2	Z Tx
0	Spherical Ball in Cylindrical Trough	2	ΧZ
р	Peg in Slotted Hole	2	х ту
đ	Spheroid on Plane Surface	1	Z

Choice: d

Please input the data to create the homogeneous transform from the part to feature frame for this feature. Note, this is the transform from the part-level coordinate frame of the part 'Str_3' to the location of this feature on that part.

X Y Z theta-x theta-y theta-z: -110.722 236.5 0 0 0 25.6

Input the length (1) of the slot and diameter (d) of the peg. $1\ d:\ .14$.1

DFC link 4 connects parts 'Fwd_Skin' and 'Plus_Chord'.

It constrains 1 degrees of freedom.

How many features do you want to use to realize this link? 1

Please choose feature 1 from the following list:

Choice	Feature Name	DOF constrained				
a	Prismatic Peg / Prismatic Hole	6	X Y Z Tx Ty Tz			
b	Plate Pin in Through Hole	5	х у Z Тх Ту			
С	Primatic Peg / Prismatic Slot	5	X Z Tx Ty Tz			
d	Plate Pin in Slotted Hole	4	X Z Tx Ty			
e	Round Peg / Prismatic Slot	4	X Z Tx Ty			
£	Round Peg / Through or Blind Hole	4	х у тх ту			
g	Threaded Joint	4	х у тх ту			
h	Elliptical Ball and Socket	4	X Y Z Tx			
i	Plate-Plate Lap Joint	3	Z Tx Ty			

```
j
       Spherical Joint
                                            3 X Y Z
       Plate Pin in Oversize Hole
                                            3 Z Tx Ty
k
1
       Elliptical Ball in Cylindrical Trough 3 X Z Tx
       Thin Rib / Plane Surface
                                           2 Z Ty
m
       Ellipsoid on Plane Surface
                                            2 Z Tx
n
       Spherical Ball in Cylindrical Trough 2 X Z
0
       Peg in Slotted Hole
                                            2 X Ty
р
       Spheroid on Plane Surface
                                            1 Z
```

Choice: d

Please input the data to create the homogeneous transform from the part to feature frame for this feature. Note, this is the transform from the part-level coordinate frame of the part 'Fwd_Skin' to the location of this feature on that part.

X Y Z theta-x theta-y theta-z: 43 1 0 0 0 -90

Input the length (1) of the slot and diameter (d) of the peg. $1 \ d: .26 \ .25$

DFC link 5 connects parts 'Aft_Skin' and 'Plus_Chord'.

It constrains 5 degrees of freedom.

How many features do you want to use to realize this link? 1

Please choose feature 1 from the following list:

Choice	Feature Name	DOF constrained			
a	Prismatic Peg / Prismatic Hole	6	X Y Z Tx Ty Tz		
b	Plate Pin in Through Hole	5	X Y Z Tx Ty		
C	Primatic Peg / Prismatic Slot	5	X Z Tx Ty Tz		
d	Plate Pin in Slotted Hole	4	X Z Tx Ty		
e	Round Peg / Prismatic Slot	4	X Z Tx Ty		
£	Round Peg / Through or Blind Hole	4	х у тх ту		
g	Threaded Joint	4	Х Ү Тх Ту		
h	Elliptical Ball and Socket	4	X Y Z Tx		
i	Plate-Plate Lap Joint	3	Z Tx Ty		
j	Spherical Joint	3	XYZ		
k	Plate Pin in Oversize Hole	3	Z Тх Ту		
1	Elliptical Ball in Cylindrical Trough	3	X Z Tx		
m	Thin Rib / Plane Surface	2	Z Ty		
r:	Ellipsoid on Plane Surface	2	Z Tx		
0	Spherical Ball in Cylindrical Trough	2	ХZ		
р	Peg in Slotted Hole	2	х ту		
q	Spheroid on Plane Surface	1	Z		

Choice: b

Please imput the data to create the homogeneous transform from the part to feature frame for this feature. Note, this is the transform from the part-level coordinate frame of the part 'Aft_Skin' to the location of this feature on that part.

X Y Z theta-x theta-y theta-z: 1 1 0 0 0 0

Please input the angular limits (in degrees) on rotation about the pin axis. (Tz)

Positive Limit: 180 Negative Limit: 180 DFC link 6 connects parts 'Aft_Skin' and 'Str_1_2'.

It constrains 2 degrees of freedom.

How many features do you want to use to realize this link? 1

Please choose feature 1 from the following list:

Choice	Feature Name	DOF constrained				
a	Prismatic Peg / Prismatic Hole	6	X Y Z Tx Ty Tz			
b	Plate Pin in Through Hole	5	X Y Z Tx Ty			
c	Primatic Peg / Prismatic Slot	5	X Z Tx Ty Tz			
đ	Plate Pin in Slotted Hole	4	х и тх ту			
е	Round Peg / Prismatic Slot	4	Х Z Тх Ту			
f	Round Peg / Through or Blind Hole	4	х ү тх ту			
g	Threaded Joint	4	х у тх ту			
h	Elliptical Ball and Socket	4	X Y Z Tx			
i	Plate-Plate Lap Joint	3	Z Тх Ту			
j	Spherical Joint	3	X Y Z			
k	Plate Pin in Oversize Hole	3	Z Тх Ту			
1	Elliptical Ball in Cylindrical Trough	3	X Z Tx			
m	Thin Rib / Plane Surface	2	Z Ty			
n	Ellipsoid on Plane Surface	2	Z Tx			
0	Spherical Ball in Cylindrical Trough	2	ΧZ			
p	Peg in Slotted Hole	2	х ту			
q	Spheroid on Plane Surface	1	Z			

Choice: d

Please input the data to create the homogeneous transform from the part to feature frame for this feature. Note, this is the transform from the part-level coordinate frame of the part 'Aft_Skin' to the location of this feature on that part.

X Y Z theta-x theta-y theta-z: -104.047 240 0 0 0 25.6

Input the length (1) of the slot and diameter (d) of the peg. $1 \ d: .14 \ .1$

DFC link 7 connects parts 'Plus_Chord' and 'Str_1_2'.

It constrains 4 degrees of freedom.

How many features do you want to use to realize this link? 1

Please choose feature 1 from the following list:

Choice	Feature Name	DO:	F constrained
a	Prismatic Peg / Prismatic Hole	6	X Y Z Tx Ty Tz
b	Plate Pin in Through Hole	5	X Y Z Tx Ty
С	Primatic Peg / Prismatic Slot	5	X Z Tx Ty Tz
đ	Plate Pin in Slotted Hole	4	х z тх ту
e	Round Peg / Prismatic Slot	4	X Z Tx Ty
£	Round Peg / Through or Blind Hole	4	х ү тх ту
g	Threaded Joint	4	х ү тх ту
h	Elliptical Ball and Socket	4	X Y Z Tx
i	Plate-Plate Lap Joint	3	Z Tx Ty
j	Spherical Joint	3	XYZ
k	Plate Pin in Oversize Hole	3	Z Tx Ty
1	Elliptical Ball in Cylindrical Trough	3	X Z Tx
m	Thin Rib / Plane Surface	2	Z Ty
n	Ellipsoid on Plane Sunface	2	Z Tx

```
o Spherical Ball in Cylindrical Trough 2 X Z p Peg in Slotted Hole 2 X Ty Spheroid on Plane Surface 1 Z
```

Choice: f

Please input the data to create the homogeneous transform from the part to feature frame for this feature. Note, this is the transform from the part-level coordinate frame of the part 'Plus_Chord' to the location of this feature on that part.

X Y Z theta-x theta-y theta-z: 5 1 .22 0 0 0

Please input the limits on Z axis motion of the peg.

Pz Nz: .1 0

Please input the angular limits (in degrees) on rotation about the peg

axis. (Tz)

Positive Limit: 180 Negative Limit: 180

DFC link 8 connects parts 'Plus_Chord' and 'Str_4_11'. It constrains 4 degrees of freedom.

How many features do you want to use to realize this link? 1

Please choose feature 1 from the following list:

Choice	Feature Name	DOF constrained			
a	Prismatic Peg / Prismatic Hole	6	X Y Z Tx Ty T		
b	Plate Pin in Through Hole	5	х у Z Тх Ту		
С	Primatic Peg / Prismatic Slot	5	X Z Tx Ty Tz		
d	Plate Pin in Slotted Hole	4	X Z Tx Ty		
e	Round Peg / Prismatic Slot	4	Х Z Тх Ту		
f	Round Peg / Through or Blind Hole	4	Х Ү Тх Ту		
g	Threaded Joint	4	Х Ү Тх Ту		
h	Elliptical Ball and Socket	4	X Y Z Tx		
i	Plate-Plate Lap Joint	3	Z Tx Ty		
j	Spherical Joint	3	X Y Z		
k	Plate Pin in Oversize Hole	3	Z Tx Ty		
1	Elliptical Ball in Cylindrical Trough	3	X Z Tx		
m	Thin Rib / Plane Surface	2	Z Ty		
n	Ellipsoid on Plane Surface	2	Z Tx		
0	Spherical Ball in Cylindrical Trough	2	X Z		
р	Peg in Slotted Hole	2	х ту		
q	Spheroid on Plane Surface	1	Z		

Choice: f

Please input the data to create the homogeneous transform from the part to feature frame for this feature. Note, this is the transform from the part-level coordinate frame of the part 'Plus_Chord' to the location of this feature on that part.

X Y Z theta-x theta-y theta-z: 17.226 1 .22 0 0 0

Please input the limits on Z axis motion of the peg.

Pz Nz: .1 0

Please input the angular limits (in degrees) on rotation about the peg

axis. (Tz)

Positive Limit: 180

Negative Limit: 180

DFC link 9 connects parts 'Fwd_Skin' and 'Str_4_11'. It constrains 2 degrees of freedom. How many features do you want to use to realize this link? 1

Please choose feature 1 from the following list:

Choice	Feature Name	DO	F constrained
a	Prismatic Peg / Prismatic Hole	6	X Y Z Tx Ty Tz
b	Plate Pin in Through Hole	5	X Y Z Tx Ty
С	Primatic Peg / Prismatic Slot	5	X Z Tx Ty Tz
đ	Plate Pin in Slotted Hole	4	X Z Tx Ty
e	Round Peg / Prismatic Slot	4	X Z Tx Ty
£	Round Peg / Through or Blind Hole	4	Х Ү Тх Ту
g	Threaded Joint	4	х ү тх ту
h	Elliptical Ball and Socket	4	X Y Z Tx
i	Plate-Plate Lap Joint	3	Z Tx Ty
j	Spherical Joint	3	X Y Z
k	Plate Pin in Oversize Hole	3	Z Tx Ty
1	Elliptical Ball in Cylindrical Trough	3	X Z Tx
m	Thin Rib / Plane Surface	2	Z Ty
n	Ellipsoid on Plane Surface	2	Z Tx
0	Spherical Ball in Cylindrical Trough	2	X Z
p	Peg in Slotted Hole	2	х ту
q	Spheroid on Plane Surface	1	Z

Choice: d

Please input the data to create the homogeneous transform from the part to feature frame for this feature. Note, this is the transform from the part-level coordinate frame of the part 'Fwd_Skin' to the location of this feature on that part.

X Y Z theta-x theta-y theta-z: 30 240 0 0 0 25.6

Input the length (1) of the slot and diameter (d) of the peg. 1 d: .14 .1

Contact 1 connects parts 'Str_3' and 'Plus_Chord'.
How many features do you want to use to realize this link? 1
Please choose feature 1 from the following list:

Choice	Feature Name	DOF constrained					1	
a	Prismatic Peg / Prismatic Hole	6	Х	Y	Z	Тх	Ту	Tz
b	Plate Pin in Through Hole	5	Х	Y	Z	$\mathbf{T}\mathbf{x}$	Ту	
С	Primatic Peg / Prismatic Slot	5	X	Z	Тx	: T ₃	T	Z
đ	Plate Pin in Slotted Hole	4	X	Z	Т×	: T ₂	,	
е	Round Peg / Prismatic Slot	4	X	Z	Т×	: T ₃	7	
f	Round Peg / Through or Blind Hole	4	Х	Y	Т×	: T3	r	
g	Threaded Joint	4	X	Y	Т×	: T3	,	
h	Elliptical Ball and Socket	4	Х	Y	Z	Тx		
i	Plate-Plate Lap Joint	3	Z	T	c I	፞፞፞፞፞፞፞		
j	Spherical Joint	3	Х	Y	Z			
k	Plate Pin in Oversize Hole	3	Z	T	c I	Υ		
1	Elliptical Ball in Cylindrical Trough	3	X	Z	Т×	:		
m	Thin Rib / Plane Surface	2	Z	Ty	7			
n	Ellipsoid on Plane Surface	2	Z	T	•			

```
Spherical Ball in Cylindrical Trough 2 X Z
0
                                             2 X Ty
        Peg in Slotted Hole
        Spheroid on Plane Surface
Choice: i
Please input the data to create the homogeneous transform from the part
to feature frame for this feature. Note, this is the transform from the
part-level coordinate frame of the part 'Str_3' to the location of this
contact feature on that part.
X Y Z theta-x theta-y theta-z: -2.233 -2.5 -2.05 0 0 0
Input the maximum and minimum joint overlap (ymax, ymin),
the distance to the corners of the plate (Pxc, Nxc),
and the limits on + or - x direction motions (Px, Nx).
ymax ymin Pxc Nxc Px Nx: .2 .2 .5 .5 .2 .2
List of parts and their variables -----
Part Name: Support
id#: 1
Motion Limit Vectors:
positive limits: 0 0 0 0 0 0
negative limits: 0 0 0 0 0 0
Part Transform:
f =
1 0 0 0
0 1 0 0
0 0 1 0
0 0 0 1
Mates:
T =
t:
0 0 0 0 0
F =
£ =
1 0 0 -4
0 1 0 0
0 0 1 0.22
0 0 0 1
type = a
pa = 1, 2
pl = 0, 0, 0, 0, 0, 0
n1 = 0, 0, 0, 0, 0, 0
Contacts:
Part Name: Aft_Skin
id#: 2
Motion Limit Vectors:
positive limits: 0 0 0 0 0 0
```

```
negative limits: 0 0 0 0 0 0
Part Transform:
f =
1 0 0 -4
0 1 0 0
0 0 1 0.22
0 0 0 1
Mates:
T =
t:
0 0 0 0 0 0
F =
f =
1 0 0 0
0 1 0 0
0 0 1 0
0 0 0 1
type = a
pa = 1, 2
pl = 0, 0, 0, 0, 0, 0
nl = 0, 0, 0, 0, 0, 0
T =
t:
0 0 1 4 -18.495 0
0 0 0 -1 0 0
F =
f =
0 -1 0 18.495
1 0 0 4
0 0 1 0
0 0 0 1
type = d
pa = 2, 3
p1 = 0, 0.005, 0, 0, 0, 3.14159
n1 = 0, 0.005, 0, 0, 0, 3.14159
T =
t:
0 0 1 240 94.579 0
0 0 0 -0.432086 0.901833 0
F =
f =
0.901833 -0.432086 0 -94.579
0.432086 0.901833 0 240
0 0 1 0
0 0 0 1
type = d
```

```
pa = 2, 3
pl = 0, 0.02, 0, 0, 0, 3.14159
n1 = 0, 0.02, 0, 0, 0, 3.14159
T =
t:
0 0 1 1 -1 0
F =
f =
1 0 0 1
0 1 0 1
0 0 1 0
0 0 0 1
type = b
pa = 2, 6
p1 = 0, 0, 0, 0, 0, 180
n1 = 0, 0, 0, 0, 0, 180
T =
t:
0 0 1 240 104.047 0
0 0 0 -0.432086 0.901833 0
F =
f =
0.901833 -0.432086 0 -104.047
0.432086 0.901833 0 240
0 0 1 0
0 0 0 1
type = d
pa = 2, 7
p1 = 0, 0.02, 0, 0, 0, 3.14159
n1 = 0, 0.02, 0, 0, 0, 3.14159
Contacts:
Part Name: Str_3
id#: 3
Motion Limit Vectors:
positive limits: 0 0 0 0 0 0
negative limits: 0 0 0 0 0 0
Part Transform:
f =
1 0 0 13.233
0 1 0 3.5
0 0 1 0.22
0 0 0 1
Mates:
T =
0 0 1 0.5 -1.262 0
```

```
0 0 0 -1 0 0
F =
f =
0 -1 0 1.262
1 0 0 0.5
0 0 1 0
0 0 0 1
type = d
pa = 2, 3
p1 = 0, 0.005, 0, 0, 0, 3.14159
n1 = 0, 0.005, 0, 0, 0, 3.14159
T =
t:
0 0 1 236.5 111.812 0
0 0 0 -0.432086 0.901833 0
F =
f =
0.901833 - 0.432086 0 - 111.812
0.432086 0.901833 0 236.5
0 0 1 0
0 0 0 1
type = d
pa = 2, 3
pl = 0, 0.02, 0, 0, 0, 3.14159
nl = 0, 0.02, 0, 0, 0, 3.14159
T =
t:
0 0 1 0.5 -2.361 0
0 0 0 -1 0 0
F =
f =
0 -1 0 2.361
1 0 0 0.5
0 0 1 0
0 0 0 1
type = d
pa = 3, 4
p1 = 0, -12.37, 0, 0, 0, 3.14159
nl = 0, -12.37, 0, 0, 0, 3.14159
T =
t:
0 0 1 236.5 110.722 0
0 0 0 -0.432086 0.901833 0
F =
f =
0.901833 -0.432086 0 -110.722
0.432086 0.901833 0 236.5
```

```
0 0 1 0
0 0 0 1
type = d
pa = 3, 4
p1 = 0, 0.02, 0, 0, 0, 3.14159
n1 = 0, 0.02, 0, 0, 0, 3.14159
Contacts:
T =
t:
0 0 1 0 0 0
0 0 0 0 1 0
0 0 0 1 0 0
F =
f =
1 0 0 -2.233
0 \ 1 \ 0 \ -2.5
0 0 1 -2.05
0 0 0 1
type = i
pa = 3, 6
p1 = 0.2, 0.2, 0, 0, 0, 0.4
nl = 0.2, 0.2, 0, 0, 0, 0.4
Part Name: Fwd_Skin
id#: 4
Motion Limit Vectors:
positive limits: 0 0 0 0 0 0
negative limits: 0 0 0 0 0 0
Part Transform:
f =
1 0 0 17
0 1 0 0
0 0 1 0.22
0 0 0 1
Mates:
T =
t:
0 0 1 4 1.406 0
0 0 0 -1 0 0
F =
f =
0 -1 0 -1.406
1 0 0 4
0 0 1 0
0 0 0 1
type = d
pa = 3, 4
```

```
p1 = 0, -12.37, 0, 0, 0, 3.14159
n1 = 0, -12.37, 0, 0, 0, 3.14159
T =
t:
0 0 1 240 114.489 0
0 0 0 -0.432086 0.901833 0
F =
f =
0.901833 -0.432086 0 -114.489
0.432086 0.901833 0 240
0 0 1 0
0 0 0 1
type = d
pa = 3, 4
p1 = 0, 0.02, 0, 0, 0, 3.14159
nl = 0, 0.02, 0, 0, 0, 3.14159
t:
0 0 1 1 -43 0
0 0 0 1 0 0
F =
f =
0 1 0 43
-1 0 0 1
0 0 1 0
0 0 0 1
type = d
pa = 4, 6
pl = 0, 0.005, 0, 0, 0, 3.14159
n1 = 0, 0.005, 0, 0, 0, 3.14159
T =
t:
0 0 1 240 -30 0
0 0 0 -0.432086 0.901833 0
F =
f =
0.901833 -0.432086 0 30
0.432086 0.901833 0 240
0 0 1 0
0 0 0 1
type = d
pa = 4, 5
p1 = 0, 0.02, 0, 0, 0, 3.14159
n1 = 0, 0.02, 0, 0, 0, 3.14159
Contacts:
```

Part Name: Str_4_11

```
id#: 5
Motion Limit Vectors:
positive limits: 0 0 0 0 0 0
negative limits: 0 0 0 0 0 0
Part Transform:
f =
1 0 0 17.226
0 1 0 1
0 0 1 0.22
0 0 0 1
Mates:
T =
t:
0 0 1 0 0 0
0 0 0 0 0 1
F =
f =
1 0 0 0
0 1 0 0
0 0 1 0
0 0 0 1
type = f
pa = 6, 5
p1 = 0, 0, 0.1, 0, 0, 180
nl = 0, 0, 0, 0, 0, 180
T =
t:
0 0 1 239 -29.774 0
0 0 0 -0.432086 0.901833 0
F =
f =
0.901833 -0.432086 0 29.774
0.432086 0.901833 0 239
0 0 1 0
0 0 0 1
type = d
pa = 4, 5
p1 = 0, 0.02, 0, 0, 0, 3.14159
nl = 0, 0.02, 0, 0, 0, 3.14159
Contacts:
Part Name: Plus_Chord
id#: 6
Motion Limit Vectors:
positive limits: 0 0 0 0 0 0
negative limits: 0 0 0 0 0 0
Part Transform:
f =
```

```
1 0 0 0
0 1 0 0
0 0 1 0
0 0 0 1
Mates:
T =
t:
0 0 1 1 -60 0
0 0 0 1 0 0
F =
f =
0 1 0 60
-1 0 0 1
0 0 1 0.22
0 0 0 1
type = d
pa = 4, 6
p1 = 0, 0.005, 0, 0, 0, 3.14159
nl = 0, 0.005, 0, 0, 0, 3.14159
T =
t:
0 0 1 1 3 0
F =
f =
1 0 0 -3
0 1 0 1
0 0 1 0.22
0 0 0 1
type = b
pa = 2, 6
p1 = 0, 0, 0, 0, 180
nl = 0, 0, 0, 0, 0, 180
T =
t:
0 0 1 1 -5 0
0 0 0 0 0 1
F =
f =
1 0 0 5
0 1 0 1
0 0 1 0.22
0 0 0 1
type = f
pa = 6, 7
pl = 0, 0, 0.1, 0, 0, 180
nl = 0, 0, 0, 0, 0, 180
```

3

```
T =
t:
0 0 1 1 -17.226 0
0 0 0 0 0 1
F =
f =
1 0 0 17.226
0 1 0 1
0 0 1 0.22
0 0 0 1
type = f
pa = 6, 5
pl = 0, 0, 0.1, 0, 0, 180
n1 = 0, 0, 0, 0, 0, 180
Contacts:
T =
t:
0 0 1 0 0 0
0 0 0 0 1 0
0 0 0 1 0 0
F =
f =
1 0 0 11
0 1 0 1
0 0 1 -1.83
0 0 0 1
type = i
pa = 6, 3
p1 = 0.2, 0.2, 0, 0, 0, 0.4
n1 = 0.2, 0.2, 0, 0, 0, 0.4
Part Name: Str_1_2
id#: 7
Motion Limit Vectors:
positive limits: 0 0 0 0 0 0
negative limits: 0 0 0 0 0 0
Part Transform:
f =
1 0 0 5
0 1 0 1
0 0 1 0.22
0 0 0 1
Mates:
T =.
t:
0 0 1 239 113.047 0
0 0 0 -0.432086 0.901833 0
```

```
F =
f =
0.901833 -0.432086 0 -113.047
0.432086 0.901833 0 239
0 0 1 0
0 0 0 1
type = d
pa = 2, 7
p1 = 0, 0.02, 0, 0, 0, 3.14159
n1 = 0, 0.02, 0, 0, 0, 3.14159
T =
t:
0 0 1 0 0 0
0 0 0 0 0 1
F =
f =
1 0 0 0
0 1 0 0
0 0 1 0
0 0 0 1
type = f
pa = 6, 7
pl = 0, 0, 0.1, 0, 0, 180
n1 = 0, 0, 0, 0, 0, 180
Contacts:
Mates Analysis -----
Part 'Support' is fully constrained by its mates.
Mates Motion Limit Vectors for part 'Support':
[0 \ 0 \ 0 \ 0 \ 0] = [qq]
pnl = [0 \ 0 \ 0 \ 0 \ 0]
Part 'Aft_Skin' is fully constrained by its mates.
Constraint Matrix:
The single feature constrains all dofs
Mates Motion Limit Vectors for part 'Aft_Skin':
[0 \ 0 \ 0 \ 0 \ 0] = [qq]
pnl = [0 \ 0 \ 0 \ 0 \ 0]
Part 'Str_3' is fully constrained by its mates.
Constraint Matrix:
t:
0 0 1 0 0 0
0 0 0 1 0 0
0 0 0 0 1 -5.90177e-16
rr1 = 0 \text{ om1} = 0 \text{ 0 1 vo1} = 0 \text{ 0 0}
```

```
tr1 = 1 vt1 = 1 0 0
tr2 = 2 vt2 = 0 1 -5.90177e-16
Mates Motion Limit Vectors for part 'Str_3':
ppl = [0 0 0 0 0 0]
pn1 = [0 \ 0 \ 0 \ 0 \ 0]
Part 'Fwd Skin' is fully constrained by its mates.
Constraint Matrix:
t:
0 0 1 0 0 0
0 0 0 1 0 0
0 0 0 0 1 -1.11852e-15
rr1 = 0 om1 = 0 0 1 vo1 = 0 0 0
tr1 = 1 vt1 = 1 0 0
tr2 = 2 vt2 = 0 1 -1.11852e-15
Mates Motion Limit Vectors for part 'Fwd_Skin':
ppl = [0 0 0 0 0 0]
pnl = [0 \ 0 \ 0 \ 0 \ 0]
Part 'Str_4_11' is fully constrained by its mates.
Constraint Matrix:
t:
0 0 0 1 0 0
0 0 0 0 1 0
tr1 = 0 vt1 = 1 0 0
tr2 = 1 vt2 = 0 1 0
Mates Motion Limit Vectors for part 'Str_4_11':
ppl = [0 0 0 0 0 0]
pnl = [0 \ 0 \ 0 \ 0 \ 0]
Part 'Plus_Chord' is fully constrained by its mates.
Constraint Matrix:
t:
0 0 1 0 0 0
0 0 0 1 0 0
0 0 0 0 1 4.57967e-16
rr1 = 0 \text{ om1} = 0 0 1 \text{ vo1} = 0 0 0
tr1 = 1 vt1 = 1 0 0
tr2 = 2 vt2 = 0 1 4.57967e-16
Mates Motion Limit Vectors for part 'I us_Chord':
ppl = [0 0 0 0 0 0]
pnl = [0 \ 0 \ 0 \ 0 \ 0]
Part 'Str_1_2' is fully constrained by its mates.
Constraint Matrix:
t:
```

```
0 0 0 1 0 0
0 0 0 0 1 -1.48319e-16
tr1 = 0 vt1 = 1 0 0
tr2 = 1 vt2 = 0 1 -1.46319e-16
Mates Motion Limit Vectors for part 'Str_1_2':
ppl = [0 0 0 0 0 0]
[0\ 0\ 0\ 0\ 0] = Ing
Contacts Analysis -----
This set of contacts allows no motion.
Contacts Motion Limit Vectors for part 'Support':
[0 \ 0 \ 0 \ 0 \ 0] = [qq
pnl = [0 \ 0 \ 0 \ 0 \ 0]
This set of contacts allows no motion.
Contacts Motion Limit Vectors for part 'Aft_Skin':
ppl = [0 0 0 0 0 0]
pn1 = [0 \ 0 \ 0 \ 0 \ 0]
This set of contacts allows no motion.
Contacts Motion Limit Vectors for part 'Str_3':
[0 \ 0 \ 0 \ 0 \ 0] = [qq]
pnl = [0 \ 0 \ 0 \ 0 \ 0]
This set of contacts allows no motion.
Contacts Motion Limit Vectors for part 'Fwd_Skin':
pp1 = [0 \ 0 \ 0 \ 0 \ 0]
pn1 = [0 \ 0 \ 0 \ 0 \ 0]
This set of contacts allows no motion.
Contacts Motion Limit Vectors for part 'Str_4_11':
pp1 = [0 \ 0 \ 0 \ 0 \ 0]
pn1 = [0 \ 0 \ 0 \ 0 \ 0]
Resultant Twistmatrix of Contact Analysis:
t:
0 0 1 0 0 0
0 0 0 1 0 0
0 0 0 0 1 0
rr1 = 0 om1 = 0 0 1 vo1 = 0 0 0
tr1 = 1 vt1 = 1 0 0
tr2 = 2 vt2 = 0 1 0
Axis 1:
Positive rotation occurs about the point: 0 0 0
Feature number 2 is the limiting feature for positive rotations.
Negative rotation occurs about the point: 0 0 0
Feature number 2 is the limiting feature for negative rotations.
Contacts Motion Limit Vectors for part 'Plus_Chord':
pp1 = [0.2 \ 0.2 \ 0 \ 0 \ 0.966056]
pn1 = [0.2 \ 0.2 \ 0 \ 0 \ 0.966056]
This set of contacts allows no motion.
Contacts Motion Limit Vectors for part 'Str_1_2':
```

```
ppl = [0 0 0 0 0 0]
pnl = [0 0 0 0 0 0]
```