A Unified Approach to Modeling, Verifying, and Improving the Manufacturability of Mechanical Assemblies

by

J. Michael Gray

B. S., Mechanical Engineering
Brigham Young University, 2002

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

June 2004

Signature of Author____
Department of Mechanical Engineering
May 7, 2004

Certified by____
Dr. Daniel E. Whitney
Senior Research Scientist
Center for Technology, Policy and Industrial Development
Senior Lecturer, Department of Mechanical Engineering
Thesis Supervisor

Accepted by____
Prof. Ain A. Sonin
Chairman, Departmental Committee on Graduate Students

A Unified Approach to Modeling, Verifying, and Improving the Manufacturability of Mechanical Assemblies

by

J. Michael Gray

Submitted to the Department of Mechanical Engineering
In Partial Fulfillment of the Requirements for the Degree of
Master of Science in Mechanical Engineering

ABSTRACT

The goal of a design engineering organization is to design products that satisfy customers. Reaching this objective is dependent, among other things, on five parameters: the customer expectations, the target percentage of satisfied customers, the nominal performance of the design, the variability in the manufacturing processes, and the sensitivity of the design performance to such variability.

This work presents a unified methodology that is amendable to computer implementation for modeling these five parameters for products that are primarily mechanical in nature. The validity of this methodology is subject to five major assumptions: the nominal performance of the design matches the performance expected by the customer, the set of customer expectations can be represented completely by a set of geometric relationships and tolerances between features in the assembly, the degradation in product performance is due solely to quantifiable variability or mean shift in the assembly geometry, the variability in each geometric relationship is independent of the variability in any other geometric relationship, and any compliant parts in the assembly can be accurately modeled as sets of rigid parts connected with linearly-compliant joints.

The assembly model is developed using a combination of Screw Theory, Network Theory, Homogeneous Transformation Matrices, and Probability Theory. It is shown how this model can be used to verify the manufacturability of a mechanical assembly design. It is also shown how the model and the results obtained from it can be used to improve the level of manufacturability of a design if it is found to be unacceptably low. To validate the effectiveness and accuracy of the methodology, an automated version written for Matlab® was used to model and analyze the manufacturability of an engine valvetrain. The results of this case study are presented and compared to results using existing industry-standard tools. Several suggestions for improving the manufacturability of the valvetrain are also proposed and discussed.

Thesis Supervisor: Daniel E. Whitney
Title: Senior Research Scientist

ACKNOWLEDGEMENTS

I would like to express my enormous appreciation to Dan Whitney for his patience and guidance during these last two exciting and eventful years in my life. His mentorship, constructive comments, and collaborative spirit have truly brought out the best in me. He allowed me the flexibility to pursue research that most interested me and that is something I will never forget.

A large debt is also due to Ford® Motor Company for supporting me financially through the Ford-MIT Alliance and for all the employees that facilitated my work. Specifically, I want to thank Craig Moccio, Agus Sudjianto, Tim Bohr, Chris Hocking, Dan Jakobcic, and Mike Vargo for their help with the valvetrain case study and for their constructive comments regarding the methodology.

This work could not have been possible without the encouragement, comic relief, and insights, provided by my friends, officemates, MIT staff, and research associates. I want to especially thank Jagmeet Arora, Ion Freeman, Gennadiy Goldenshteyn, Tom Speller, Mo-Han Hsieh, and Peng Zhang. I also want to recognize CIPD staff members Chris Magee, Nils Nordal, Kathleen Sullivan, and Susan MacPhee for all the support they gave me during these last two years. In addition, while I have never met either of these individuals, I would like to thank Trevor Davies and Gaurav Shukla whose work served as inspiration for the theory developed in this work.

I also want to acknowledge the support of my parents, brothers, and extended family. They have been a sounding board for many of my ideas. I particularly want to recognize the unwavering support, devotion, and love of my wife, Niki. Every step of the way during this challenging endeavor she has always been there to cheer me up and fill me with happiness. There is no one else in the world like her. I am also grateful for my son Adam, who is an enormous blessing to me and has been bringing a smile to my face since the moment he was born.

Finally, I want to proclaim my unshakable faith in our Heavenly Father and His Son Jesus Christ, who not only provided us all with a plan whereby we might have eternal happiness, but have been ever at my side guiding and protecting me, throughout this and all experiences in my life. Everything good that has happened in my life is due to their infinite love and mercy and I am eternally grateful for that.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

1. INTRODUCTION

1.1 Motivation and Background

One of the objectives of a design engineering organization is to design products that meet the expectations of as many customers as possible. This can often be achieved via a top-down design process in which high-level customer expectations are cascaded into a more specific set of product architecture specifications that are then mapped further onto a list of functional requirements. Different methods for accomplishing this are presented by [Clausing], [Mantripragada and Whitney, 1998], [Pahl and Beitz], [Shukla], [Slocum], [Suh], [Ulrich and Eppinger], and others. A functional requirement is partially characterized by defining a target for one of the explicit measurable characteristics of the product (e.g. a certain top-speed, power rating, weight, or efficiency) and each functional requirement can be tied directly through the cascading process to one or more customer expectations. The set of functional requirements should capture not only the expectations of a "typical" customer, but should be defined so that it takes into account the variety of customers and usage scenarios for the product. Thus functional requirements provide a quantifiable way for engineering to measure the performance of a product relative to the expectations of the customers.

In addition to a target (e.g. the engine must be designed to deliver 200 HP), each functional requirement usually has a tolerance or limit that reflects the willingness of the customer to accept a product whose performance is less than the expectation (e.g. the engine must be designed to deliver no less than 195 HP). A single product is said to “meet a functional requirement” if the actual value of a functional requirement is equal to the target or within the tolerance of the functional requirement (e.g. the actual engine delivers 197 HP which is above the minimum of 195 HP). A single satisfied customer is someone who buys a single product that meets all the functional requirements.

There are two main factors that cause a customer not be satisfied. First, there can be a difference between the nominal or theoretical performance of the design and the performance expected by the customer. Discrepancies of this type come from the failure or inability of the engineering organization to produce a design, even in its nominal form, that meets the functional requirements. Second, there can be a difference between the actual performance of the manufactured product and the theoretical performance of the design. Such discrepancies are due to the interaction between the inherent variability in the available manufacturing processes capable of converting the design into an actual product and the sensitivity of the design to such variability. This means it is possible for two customers with identical expectations and usage scenarios to buy the same product and have one satisfied but not the other. For this reason, discussions regarding this second factor are usually framed by looking at the percentage of satisfied customers rather that the satisfaction of any single customer. A design for a single product is considered a success if the actual percentage of satisfied customers is above a required target percentage. This target percentage is usually set by looking at the needs and goals of the enterprise.

Given the discussion above, the objective of a design engineering organization is to design products that, when fabricated and assembled using the available manufacturing processes, meet all the functional requirements in the required target percentage of cases. Reaching this objective depends, among other things, on the following five parameters:

- **Functional Requirements Targets and Tolerances**
- **Target Percentage of Satisfied Customers**
- **Nominal Performance of Design**
- **Sensitivity of Design to Manufacturing Process Variability**
- **Variability in Available Manufacturing Processes**

Reaching the design objective requires an organization that can efficiently and effectively deal with all the parameters listed above. For each product such organizations must, among many other things, be able to verify the extent to which the design objective will be met and remedy any shortcomings before the product is released to the market. A major goal of this work is to develop a methodology that supports such verification efforts. The list of parameters above suggests the following analogous set of remedies:

- **Lower Functional Requirement Targets/Increase Tolerances**
- **Lower Target Percentage of Satisfied Customers**
- **Increase Performance of Nominal Design**
- **Reduce Sensitivity of Design to Manufacturing Process Variability**
- **Increase Capability of Available Manufacturing Processes**

In exploring and applying the remedies above, the engineering organization will need to engage in both internal dialogue and interface with other organizations within the enterprise (e.g. marketing and manufacturing). It is therefore important, among other things, that the engineering organization is able to assess the benefits, risks, and costs associated with each potential remedy and make the appropriate design changes. A second major goal of this work is to create a methodology that supports such activities.

1.2 Scope

The validity of the methodology presented in this thesis is subject to the following five major assumptions:

- **The Nominal Design Performance Matches the Customer Expectations**
- **All Functional Requirements Can Be Expressed Geometrically**
- **All Manufacturing Variability Can Be Expressed Geometrically**
- **All Variability in Geometry is Independent**
- **All Compliant Parts Can be Modeled as Sets of Rigid Parts Connected by Linearly-Compliant Joints**

The first assumption is that the performance of the nominal design matches the expectations of the customer. For this to be true, the engineering organization must always be able to deliver a nominal design whose performance is theoretically sufficient. As such, this work does not provide any insight into what an engineering organization should do to create a competent nominal design and does not provide any means for verifying that the performance of a nominal design has indeed hit the functional requirement targets.

In short, the first assumption eliminates entirely one of the two main factors that can cause a customer to become dissatisfied: the discrepancy between the nominal performance of the design and the target performance expected by the customer. What remains is how the second factor, the interaction between the inherent variability in the available manufacturing processes and the sensitivity of the design to such variability, relates to the required target percentage of satisfied customers. In this work the relationship between this second factor and the required target percentage of satisfied customers will be referred to as the **manufacturability** of the design. A design with a high level of manufacturability is one that either has relatively low sensitivity to the variability in the manufacturing processes, is made using manufacturing processes with relatively low variability, has a relatively low required target percentage of satisfied customers, or some combination of the three.

The second assumption is that it is possible to map all functional requirement targets onto an equivalent set of nominal one-dimensional geometric relationships within the assembly (e.g. distances or angles between features or parts in an assembly). Further, it must also be possible to map all tolerances associated with the functional requirements onto a set of equivalent limits on these relationships within the assembly. In order for such a mapping to take place, all the details of the nominal design must be known (e.g. the number of parts, geometry, joints).

The third assumption is that all variability in the available manufacturing process can be expressed in geometric terms. Implicit in this assumption is that the current capabilities of the available manufacturing processes that will carry out the various fabrication and assembly processes required can be measured and are known.

Together, the third and fourth assumptions insure that all the parameters and performance measures associated with the design of a product are not only defined, but are effectively "normalized". This means that they all will have either units of angular rotation (e.g. radians or degrees) or units of length (e.g. inches or millimeters). This also insures that the methodology developing in this work is context-free and can be applied to a variety of different assemblies. Figure 1.1 shows an example of a process that might be used to convert a qualitative customer need into an equivalent set of geometric relationships and tolerances.

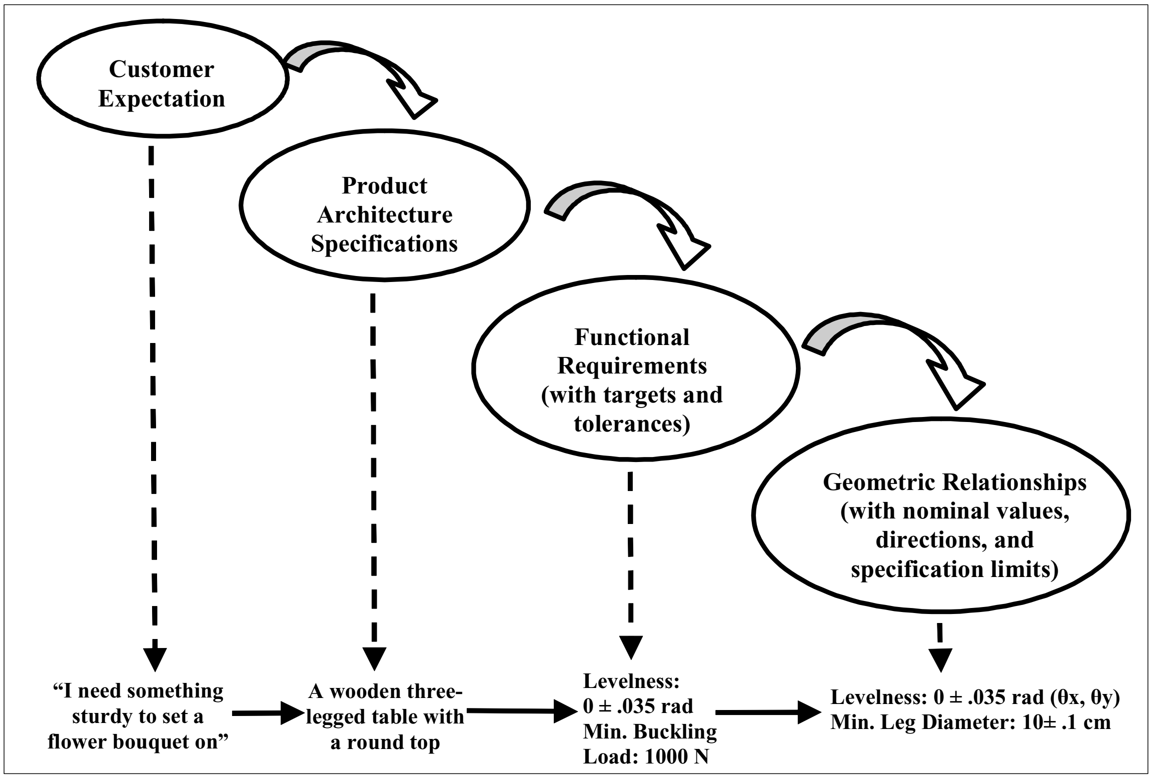

Customer Expectation
Product Architecture Specifications
Functional Requirements (with targets and tolerances)
Geometric Relationships (with nominal values, directions, and specification limits)
"I need something sturdy to set a flower bouquet on"
A wooden three-legged table with a round top
Levelness: 0 ± .035 rad Min. Buckling Load: 1000 N
Levelness: 0 ± .035 rad (θx, θy) Min. Leg Diameter: 10± .1 cm

Figure 1.1. Cascading Customer Needs into Geometric Relationships

It is essential for the successful implementation of this methodology that the engineering organization posses the ability to create competent nominal designs and convert all important aspects of the design and the customer expectations into geometric terms as shown in Figure 1.1. This requires knowledge of the parameters that govern design performance and how these parameters can be tied to the geometric relationships within the product. For a discussion on this subject see [Whitney].

The last two assumptions are that the variability in one geometric relationship is independent of the variability in any other geometric relationship and that compliant parts can be modeled as sets of rigid parts connects by linearly-compliant joints. These two assumptions enable the use of several mathematical methods that would have otherwise been impractical or inaccurate.

1.3 Thesis Goals

The goal of this work is to develop a methodology, specifically applicable to products that are primarily mechanical in nature, that enables design engineering organizations to verify the manufacturability of an assembly and make improvements if necessary. This methodology consists of three major components:

- **Creating a Accurate Quantitative Model of a Mechanical Assembly Design**
- **Using the Model to Verify Manufacturability**
- **Using the Model to Support Manufacturability Improvement Efforts**

1.4 Prior Research

Many approaches have been proposed to analytically model assemblies so that their nominal performance and manufacturability can be assessed and improved. Researchers such as [Adams], [Chang and Perng], [De Fazio et al], [Gui and Mäntylä], [Shukla], [Söderberg and Johannesson], and others have proposed featured-based models that do not require the details of the design such as geometry to be known but provide ways to interrelate the parts to each other and assess the viability of the design in high-level ways. Such models are very useful for comparing concepts early in the product design process and detecting so-called tolerance chains or Datum Flow Chains but are not capable of providing precise quantitative statements regarding manufacturability.

Other models capture the assembly process in addition to the assembly design. One such model presented by [Hu], based on what is called Stream-of-Variation Theory, characterizes the assembly process as a series of stages that each introduces some variability into the assembly. The manufacturability of the final assembly is determined by looking at the combined effect of all the variability that is introduced during the assembly process. [Camelio, Hu, and Ceglarek], [Ceglarek and Shi], [Lee, Long, and Hu], [Shiu, Ceglarek, and Shi], and others have built on this theory to include variability in the geometry and stiffness of compliant parts using Finite Element Analysis, or FEA, and all types of variability distributions using Monte Carlo simulation. Since the combination of FEA and Monte Carlo simulation is computationally infeasible in many cases, [Lui, and Hu] presented an alternate and more efficient approach called the Method of Influence Coefficients. [Mantripragada and Whitney, 1998 and 1999] presented a related model for assemblies of rigid parts using a state transition approach taken from Control Theory where each assembly operation is treated as a state change that adds to the variability of the assembly. [Kern and Thornton] also present a process-based model that allows for a closed-form solution using two mathematical functions called the “**DeltaP**” “and “**SigmaP**” functions.

Other approaches to modeling compliant assemblies have been proposed. [Chang and Gossard] developed a model for compliant assemblies based on the first principles of static analysis of structures where the stiffness of the parts and joints are obtained using

FEA. [Bihlmaier] has demonstrated how FEA can be combined with spectral analysis for modeling and analyzing compliant assemblies. [Donald and Pai], [Fasse], and [Fasse and Zhang] use a lumped-parameter model where the stiffness is localized at the joints between parts and the part themselves are modeled as collections of compliantly-connected rigid parts. [Villarreal and Asada] introduce the concept of buffer zones to model assemblies containing parts with distributed, not localized, compliance (e.g. thin sheet metal). [Ramana and Rao] present an in-depth review of all such techniques as applied to sheet metal assemblies.

Other analytical assembly modeling techniques such as the offset method by [Requicha], the vector-based method of [Wirtz] and built upon by [Yau], and others have arisen to address the modeling of tolerances on geometric dimensions. One such scheme, based on kinematics and torsors that was developed by [Clement et al], [Clement and Derochers], [Derochers], [Rivest, Fortin, and Morel], [Marguet], and others, is particularly relevant to this thesis. In this approach, kinematic constraints are imposed on the motion of the parts by converting each part into a Technologically and Topologically Related Surface, or TTRS. The TTRSs are then related to each other using Graph Theory [Maxwell and Reed]. Variation torsors are added to the model to simulate the effects of dimensional variability. Typically, TTRS-based formulations are limited to rigid parts connect by lower-order kinematic pairs. Manufacturability is usually expressed using a worst-case or root-sum-square approach (root-sum-square, or RSS, usually assumes that all the variables are independent and normally distributed).

[Laperriére and Lafond] and [Laperriére, Ghie, and Derochers] built on the TTRS model using "virtual joints" to locate features in space and represent variability and Jacobian matrices to find the sensitivities of the performance measures to this variability. Assemblies whose parts are joined to each other such that they are over-constrained (i.e. the kinematic constraints between the parts are such that locked-in stresses will arise if the geometry of the parts deviates from its nominal definition) are dealt with using a pseudo-inverse operation that provides a best-fit linear approximation. While computationally attractive, the pseudo-inverse does not always give accurate results. This is because it does not take into account the compliance of the over-constrained parts.

[Chase, Gao, Magleby, Sorenson] and [Gao, Chase, and Magleby] have also developed a kinematic approach where parts are related to each other using kinematic constraints. Dimensional variations, based on tolerances of both form and size, are treated as small kinematic adjustments. Vector loops are used to connect the various parts and create a complete assembly model. Manufacturability is estimated using a worst-case or RSS approach that relies on the linearized sensitivities derived from the kinematic assembly model. For over-constrained assemblies, the efficient but sometimes inaccurate pseudo-inverse operation is used.

Another group of kinematic approaches utilizes Screw Theory, a six-dimensional vector algebra for representing forces and velocities [Ball]. [Konkar] and [Konkar and

Cutkosky] use Screw Theory to create instantaneous models that can be incremented to determine whether assemblies have full cycle mobility. [Davies and Primrose] and [Davies, 1981, 1983a, 1983b, 1983c, 1995, and 2000] combine Screw Theory and Network Theory to create generalized instantaneous kinematic assembly models that can be used to determine the presence of any excess over-constraints (e.g. the potential for locked-in stress) or under-constraints (e.g. the parts can move relative to each other). Due to the instantaneous first-order nature of Screw Theory, these models only provide a "snapshot" of an assembly and must be updated to reflect different configurations of the assembly. Also, none of these technique model compliant parts. [Davidson and Shah] and [Davidson Mujezinović, and Shah] show how Screw Theory can be used in connection with Minkowski sums to find the worst-case envelope for critical geometric relationships in an assembly given dimensional variability. [Tischler and Samuel] introduce variation as infinitesimal screws to determine the effect of slop in spatial linkages on their performance and precision. [Fasse], [Fasse and Zhang], [Patterson and Lipkin, 1993a], and [Patterson and Lipkin, 1993b] combine Screw Theory with localized joint stiffness and compliance matrices to create models for representing robots and other dynamic systems.

Another line of research involves methods for carrying out what is called tolerance analysis and tolerance synthesis (or allocation). [Chase and Parkinson] and [Hong and Chang] provide detailed reviews of these methods. Generally, tolerance analysis methods (see [Varghese, Braswell, Wand, and Zhang] [Sacks and Joskowicz] for examples) look at whether the tolerances assigned to the dimensions or other parameters of a design will results in an assembly that is manufacturable. On the other hand, tolerance synthesis approaches (see [Bjørke], [Chase, Greenwood, Loosli, and Hauglund], [Ngoi and Min], and [Terry]) are intended to find the best or lowest-cost set of dimensional tolerances given the nature of the design and the cost-performance tradeoffs of the available manufacturing processes. Tolerance synthesis methods are often difficult to implement in practice because creating analytical functions that capture the cost-performance tradeoffs of manufacturing processes can be difficult. [Ding, Ceglarek, Jin, and Shi] present an assembly process-oriented tolerance synthesis method specifically applicable to sheet metal assemblies.

Many of the modeling techniques above depend in some form on Homogeneous Transformation Matrices, or HTMs, that are extremely useful for analytically relating parts, features, and coordinate frames to each other in 3-D space. HTMs were first used by [Denavit and Hartenberg] to represent kinematic linkages and then by [Ahuja and Coons] and others for displaying CAD geometry on a computer screen. [Ashiagbor, Liu, and Nnaji], [Paul], [Lee and Gossard], [Johannesson and Söderberg], [Sodhi and Turner], [Srikanth and Turner], [Whitney], and others have used them for creating generalized models of robots and assemblies that can also be used to capture their behavior in the presence of dimensional variability using Monte Carlo simulation, linearization of sensitivities, or related techniques. [Huang, Whitehouse, and Chetwynd] and others have derived specialized models that are only applicable to a specific subset of assemblies.

1.5 Modeling Approach Used In This Thesis

This work uses the kinematic formulation based on Screw Theory and Network Theory developed by [Davies and Primrose] and [Davies, 1981, 1983a, 1983b, 1983c, 1995, and 2000] to nominally model mechanical assemblies and detect excess over-constraints and under-constraints. The ability to model compliant parts is added by treating compliant parts as sets of rigid parts connected by linearly-compliant joints similar to the techniques used in [Fasse], [Fasse and Zhang], [Patterson and Lipkin, 1993a], and [Patterson and Lipkin, 1993b]. Chains of HTMs and virtual joints [Laperriére and la Fond] are used to locate all features, parts, and the joints between parts relative to a global reference frame [Whitney]. Combined use of Screw Theory and HTMs enables the modeling of assemblies with closed-loops (e.g. mechanisms and linkages), open-loops (e.g. robot arms and end effectors), or both.

A technique similar to that presented by [Gao, Chase, and Magleby] is used to model the directions of the dimensional variability and determine the sensitivity of the design performance to variability in those directions. Comparing the method in [Gao, Chase, and Magleby] to the method presented in this thesis there are several key differences. First, the model in this work has the added capability of handling assemblies with compliant parts. Second, the pseudo-inverse function is not used when analyzing over-constrained assemblies and therefore the results are generally more accurate. Finally, the dimensional variability is modeled more accurately by using a full distribution approach based on traditional Probability Theory [Drake] instead of a second moment approach where only the mean and standard deviation of the variability are used.

The manufacturability of the assembly design with respect to each customer expectation is expressed in this work as a Process Capability Index, or Cpk, which is obtained using the Most Probable Point method [Melchers], a technique that was first used in the field of civil engineering to asses and predict the reliability of structures like bridges and buildings. The contributions of the variability in the dimensions to the variability in the assembly performance measures are also obtained via the MPP method. The MPP method is not only computationally very efficient (since it does not depend on Monte Carlo simulation) but it is also quite accurate (since it does not depend on linearized sensitivities). Exploiting these facts, several methods are presented in this thesis for optimizing the nominal values of the dimensions to maximize manufacturability, an option that is not possible with many other assembly modeling techniques.

1.6 Thesis Organization

The remainder of this thesis is divided into seven chapters. Chapter 2 provides a framework for discussion by defining a set of terminology and how it is to be interpreted in this work. Chapter 2 also takes a closer look at the various characteristics of mechanical assemblies and how these characteristics affect assembly behavior. Chapter 2 concludes with a brief discussion of what aspects of assemblies are and are not included in the methodology presented.

Chapter 3 discusses in detail the modeling procedure that is used to create a quantitative representation of a mechanical assembly design that accurately models its nominal and variable properties. This procedure is divided into two phases: the Element Identification & Location Phase and the Element Quantification & Integration Phase.

Chapter 4 reveals how the model developed in Chapter 3 can be used to verify the manufacturability of an assembly. The verification process is composed of a suite of analysis techniques, namely Motion/Constraint, Sensitivity, Performance, Contribution, and Coupling Analyses.

Chapter 5 discusses how the results from Chapter 3 and Chapter 4 can be used to support efforts to improve the manufacturability of the design. These efforts are divided into three categories: System-Level, Parameter-Level, and Tolerance-Level Strategies.

Chapter 6 shows how this methodology was applied to the design of a Roller Finger Follower, or RFF, Valvetrain on a V-8 Ford® Engine. The scope was limited by looking at a single customer expectation: the level of unwanted vibration caused by the engine. This expectation was mapped geometrically to the rotation angle of the cam sprocket, since any error that angle causes undesirable and noticeable vibration. Application of the methodology revealed that the combination of high variability in the manufacturing processes and high sensitivity of the design to that variability resulted in an assembly with low manufacturability. As validation of the methodology, these results were compared to predictions using other industry-standard tools. Chapter 6 concludes by using the assembly model and the results it provided to suggest several ways to improve the manufacturability of the valvetrain.

Finally, Chapter 7 presents a summary of the work presented in this thesis and highlights the strengths and weaknesses of the methodology. Based on this discussion, some suggestions are made for future work. These suggestions are divided into three groups: Improving Existing Capabilities, Adding New Capabilities, and Improving Computer Implementation. The first group includes those changes that would improve the existing capabilities of the methodology so that it is more robust and generally applicable. The second group contains suggestions for adding capabilities to the methodology so that it can be used earlier in the design process. If possible, the methodology would not only serve as an evaluative tool, but also as a driver of design. The third category discusses how the accessibility of the methodology could be increased through improved computer implementation and user interface.

2. CHARACTERIZING MECHANICAL ASSEMBLIES

The goal of this chapter is to provide a foundation upon which a sound methodology for modeling, verifying, and improving the manufacturability of mechanical assemblies can be based. In the first section of this chapter, a set of terminology and concepts as understood and used in this thesis are presented. In the second section, a discussion is developed that addresses many of the characteristics that dictate assembly behavior. Finally, a summary of the chapter is presented including comments on what characteristics of assemblies are and are not addressed in this work.

2.1 Terminology

To eliminate the possibility for misinterpretation, it is important to define some basic terminology as it will be used in this work.

Coordinate Frame: A set of three orthogonal axes that forms the fundamental theoretical building block of mechanical design. In this work assemblies are treated as sets of interrelated coordinate frames. Coordinate frames can be located relative to each other in six independent directions. Calling the three orthogonal axes $\boldsymbol{X}$, $\boldsymbol{Y}$, and $\boldsymbol{Z}$, the directions can be labeled $\boldsymbol{\theta x}$, $\boldsymbol{\theta y}$, $\boldsymbol{\theta z}$, $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$. Directions $\boldsymbol{\theta x}$, $\boldsymbol{\theta y}$, and $\boldsymbol{\theta z}$ represent rotations about the $\boldsymbol{X}$, $\boldsymbol{Y}$, and $\boldsymbol{Z}$ axes respectively and directions $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$ represent translations along those same axes respectively. In general, these directions need not be aligned with the coordinate axes $\boldsymbol{X}$, $\boldsymbol{Y}$, and $\boldsymbol{Z}$ of a coordinate frame.

Feature: The fundamental physical building block of mechanical design. [De Fazio et al, 1993] define a feature as "any geometric or non-geometric attribute of a discrete part whose presence or dimensions are relevant to the product's or part's function, manufacture, engineering, analysis, use, etc, or whose availability as a primitive or operation facilitates the design process". In this work a feature is more narrowly defined as a geometric element (e.g. a surface, pin, slot, hole, end-effector) that is assigned a single coordinate frame and is important either because, 1) is interfaces with another feature or, 2) is critical to the function and/or performance of the assembly.

Dimension: A geometric relationship between two features in a single coordinate direction (e.g. an angle between two planes in the $\boldsymbol{\theta y}$ direction or a distance between the axes of two holes in the $\boldsymbol{x}$ direction).

Intra-Part Relationship: A set comprised of from one to six serially-linked dimensions that uniquely determines the position of one feature (and its coordinate frame) relative to another feature (and its coordinate frame).

Part: A set of features interrelated via a set of intra-part relationships.

Global Origin: A coordinate frame of an assembly that is used as the common reference for describing the location of all features in an assembly. Coordinates relative to the global origin are referred to as assembly coordinates.

Local Origin: Any coordinate frame that is used as the reference for describing the location of one or more features. Coordinates corresponding to a local origin are referred to as local coordinates. The local origin used to locate all the features on a part is called the part origin and the corresponding coordinates are called part coordinates.

Base Feature: The feature (and coordinate frame) associated with the global origin. This could be a plane, point, pin, hole, etc.

Base Part[1]**:** The part containing the base feature [Whitney, Mantripragada, Adams, and Rhee].

Key Characteristic (KC): An important relationship between two features in an assembly. [Thornton 1999] defines a KC as "[something] whose variation from nominal significantly impacts the final cost, performance [including quality], or safety of a product". In this work, a KC is more narrowly defined as a geometric relationship between two features along a single coordinate direction that has a nominal value and a specification limit. This limit can be either an upper or lower specification limit but not both. KCs in this work represent the expectations of the customer. A KC is "delivered" or "met" if the target percentage of customers is satisfied with respect to the customer expectations that the KC represents.

Mate[2]**:** A relationship between two parts, or more specifically two features on those parts, which allows relative motion in some directions and constrains it in others.

Variability: The deviation from nominal in a measurable quantity as seen in a group of theoretically identical entities that is sometimes quantified via a variance or standard deviation [Drake]. Here the measurable quantities are the dimensions or the performance-related KCs and the group of entities is a batch of assemblies all based on the same design and fabricated and assembled using the same processes. In this work the sole source of fluctuation in the KCs is the fluctuation in the dimensions due to the limited capabilities of the fabrication processes that actually set them. The variability of all other parameters is assumed to be zero.

[1] In some of the literature, the first part in an assembly sequence is referred to as the base part [Whitney].

[2] [Whitney] defines a mate as a joint between two parts that participates in insuring that a KC is delivered.

Manufacturability: The ease in which a design can be turned into a physical product [Bralla]. In this work, it is used as a descriptor of an assembly's design that indicates the extent to which the design, given the capabilities of the available manufacturing processes, will satisfy the required target percentage of customers. Manufacturability is measured by comparing the level of variability seen in the performance-related KCs relative to their respective specification limits. Increasing the specification limits, decreasing the variability of the performance-related KCs (by decreasing the sensitivity of the design to dimensional variability), or increasing the capabilities of the fabrication process will all increase manufacturability.

Figure 2.1 uses a three-legged table as an example to illustrate some of the terms mentioned above.

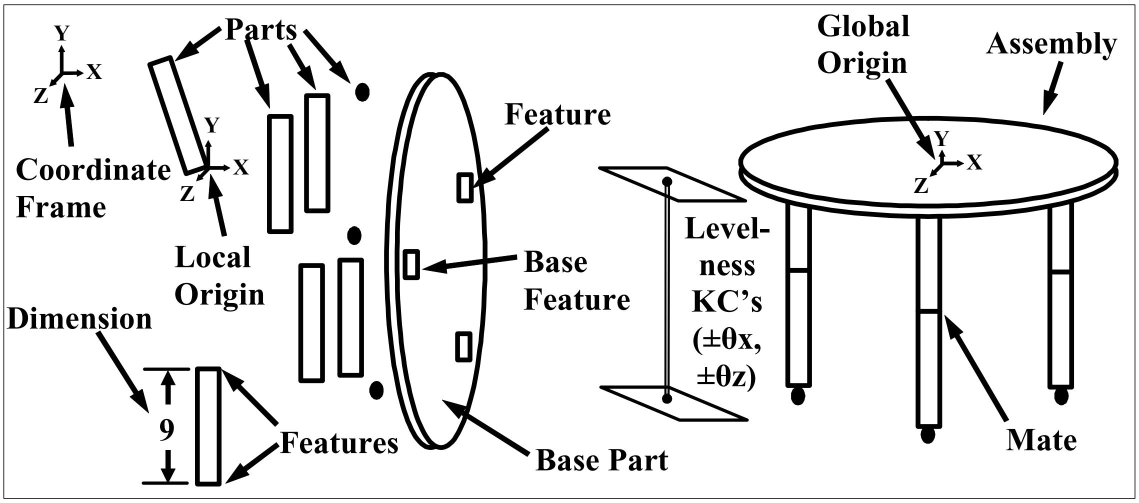


Figure 2.1. Visualization of Some Terminology Using a Table as an Example

2.2 Characteristics of Assemblies

There are a host of factors related to assembly behavior and in this work they have been divided into four categories:

- **Design**
- **Manufacture**
- **Use Environment**
- **Performance Measures**

The factors within the design category and some of the factors in the manufacture and use environment categories determine the nominal, or theoretically designed, behavior of an assembly. The remaining factors in the manufacture and use environment categories determine how much an actual assembly's behavior will deviate from nominal. The performance measures category discusses how customer expectations are quantified. These four categories and the factors they contain (as discussed below) are not mutually exclusive or collectively exhaustive. Nonetheless, they provide a useful context for the methodology presented in this work.

2.2.1 Design

All the factors in the design category are determined during the design process. Each is responsible in part for the behavior of the assembly. The following list contains the most important factors in this category relevant to this work:

- ***Part-Status***
- ***Loop-Status***
- ***Mates***
- ***Nominal Part Properties***
- ***Constraint-Status***

A brief discussion of each now follows.

Part-Status: The part-status is the term used in this work to refer to the connectivity of the assembly and is related to topology as described in the fields of Graph Theory [Maxwell and Reed] and Network Theory [Seshu and Reed] and [Davies, 1995]. At a general level, the part-status of an assembly is defined by specifying the number of parts an assembly has and the way in which those parts are connected to each other via the mates. Consider the three tables in Figure 2.2, each consisting of ten identical parts: one top, three braces, three posts, and three feet. Since the parts are connected in a different way in each case (even though the parts are the same), each assembly has a different part-status and may also have a different behavior (e.g. ability to support external loads).

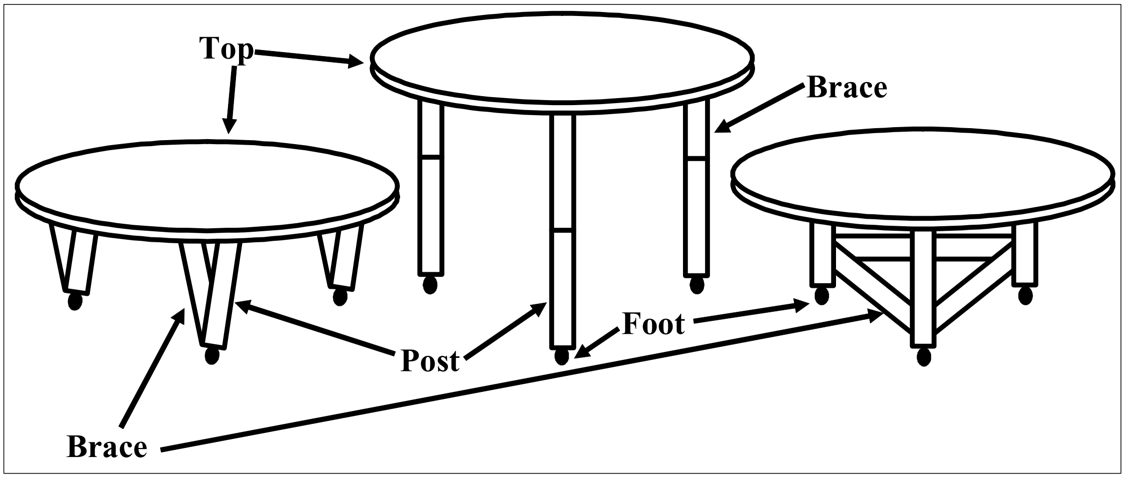


Figure 2.2. Three Tables Each with a Different Part-Status

Loop-Status: The loop-status is the dual of part-status and is an equivalent but alternate way to describe the topology or connectivity of an assembly. Loop-status is described by defining the number and nature of loops an assembly has and the set of parts, or more specifically features, each loop contains. There are only two types of loops:

- **Open-Loops**
- **Closed-Loops**

An open-loop is defined here as a group of connected features, mates, and intra-part relationships that includes the base feature (designated as the beginning of the loop), a feature that is connected to only one other feature via either a mate or an intra-part relationship (designated as the end of the loop), and the set of features, mates, and intra-part relationships one encounters on a continuous path that connect the beginning and end of the loop. A robot gripper (left side of Figure 2.3) is an example of an assembly with an open-loop with the gripper feature at the end of the open loop and the base feature at the beginning of the open-loop.

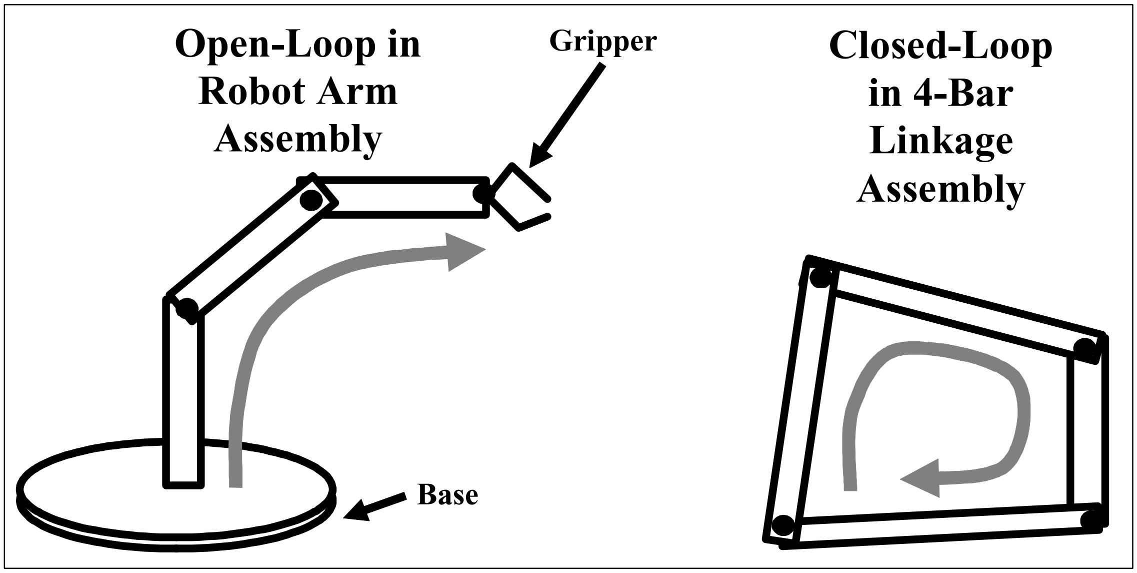

Open-Loop in Robot Arm Assembly
Gripper
Base
Closed-Loop in 4-Bar Linkage Assembly

Figure 2.3. Examples of an Open-Loop and Closed-Loop

A closed-loop is defined here as a group of connected features, mates, and intra-part relationships where each feature in the group is related via mates or intra-part relationships to no more or less than two other features in the group. A four-bar linkage (right side of Figure 2.3) is an example of an assembly with a closed-loop with the lengths of the bars serving at the intra-part relationships and the revolute joints serving as the mates. Figure 2.4 shows the three tables in Figure 2.2 inverted with some of the open-loops and closed-loops identified in each case. Notice that while each table consists of identical parts, each has a different loop-status and this may result in differences in behavior.

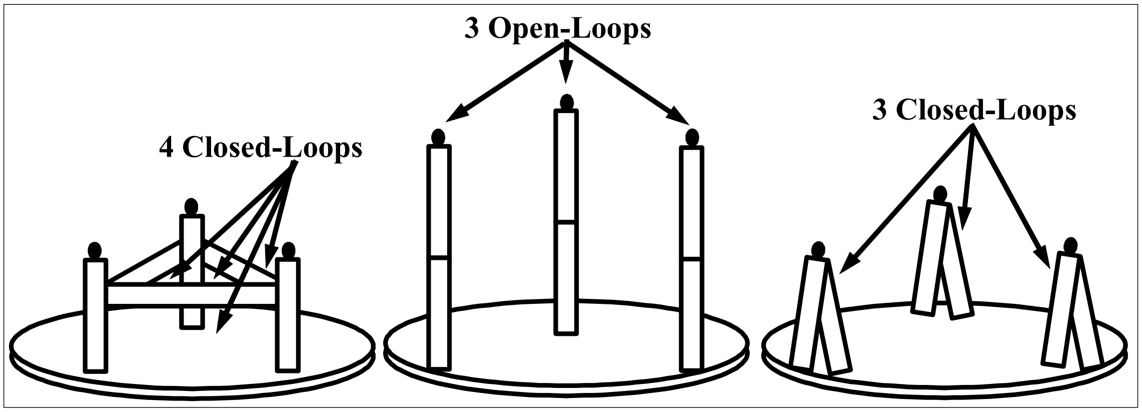


Figure 2.4. Identifying the Open-Loops and Closed-Loops in Assemblies

Mates: As mentioned above, a mate is a relationship between two parts, or more specifically, two features on different parts. Assemblies have the following two types of mates (one type which is divided further into two subcategories):

- **Real**
  - **Non-Compliant**
  - **Compliant**
- **Virtual**

A real mate is an interface that actually exists in the real assembly between two features on separate parts. Real mates are divided into two subcategories: non-compliant mates and compliant mates. A non-compliant mate is defined here as an interface between two features on separate parts (e.g. a pin in a matching hole) that completely prohibits the relative motion between the parts in from zero to six coordinate directions and completely allows relative motion in the remaining coordinate directions without storing any energy. Conversely, a compliant mate is defined here as an interface between two features on separate parts (e.g. a spring-loaded hinge) that allows relative motion in from one to six coordinate directions and can transmit forces between the parts in at least one of those same directions while storing energy at the same time. Real mates affect assembly behavior because they dictate the way in which parts interact with each other.

Virtual mates, such as those used by [Laperriére and Lafond], do not actually exist in an assembly and as such do not affect its behavior in any way. Rather, virtual mates are added solely for analysis purposes and their value will be shown in Chapter 4. An assembly has one virtual mate for each open-loop which connects the base feature to the feature at the end of the open-loop but does not store energy or enforce constraint in any coordinate direction. Instead, a virtual mate allows total unrestrained motion of the end of an open-loop relative to the base feature in all six coordinate directions.

Every mate (real or virtual) can be thought of as a serial chain of six elements and each element falls into one of the following four categories:

- **Degree-of-Freedom (DOF)**
- **Degree-of-Constraint (DOC)**
- **Degree-of-Stiffness (DOS)**
- **Degree-of-Virtuality (DOV)**

A Degree-Of-Freedom, or DOF (the abbreviation DOF refers to either a single Degree-of-Freedom or multiple Degrees-of-Freedom), is described by specifying a direction, a nominal value, and a pitch. The DOF direction is a vector (expressed in coordinates local to that DOF) along or about which the real mate completely allows the motion of one part relative to its mating part but does not transmit forces or store energy. In other words, it is assumed in this work that DOF are frictionless and have zero stiffness. The DOF nominal value is simply the distance or angle between the two coordinate frames of the mating features in the direction of the DOF. The DOF pitch describes the type of motion allowed by the DOF and can be zero, infinite, or any positive real number. A DOF with zero pitch describes a pure rotational motion. A DOF with infinite pitch describes a pure translational motion. A DOF with finite pitch describes a helical motion [Ball].

A Degree-Of-Constraint, or DOC (the abbreviation DOC refers to either a single Degree-of-Constraint or multiple Degrees-of-Constraint), is described by specifying a direction, a nominal value, and a pitch. The DOC direction is described by a vector (expressed in coordinates local to that DOC) along or about which the real mate can transmit a force (or moment) from one part to its mating part but does not allow motion or store energy. In other words, it is assumed in this work that DOC are rigid (i.e. DOC have infinite stiffness). The DOC nominal value is simply the distance or angle between the two coordinate frames of the mating features in the direction of the DOC. The DOC pitch describes the type of force (or moment) transmitted by the DOC and can be zero, infinite, or any positive real number. A DOC with zero pitch describes a pure moment. A DOC with infinite pitch describes a pure force. A DOC with finite pitch describes a force/moment couple.

A Degree-Of-Stiffness (DOS) (the abbreviation DOS refers to either a single Degree-of-Stiffness or multiple Degrees-of-Stiffness), is described by specifying a direction, a nominal value, a pitch. A DOS direction is a vector (expressed in coordinates local to that DOS) along or about which a compliant mate can store energy, allow relative motion, and transmit forces between parts. The DOS nominal value is simply the distance or angle between the two coordinate frames of the mating features in the direction of the DOS. The DOS pitch can be zero, infinite, or any positive real number. A DOS with zero pitch describes a torsional spring. A DOS with infinite pitch describes a linear spring. A DOS with finite pitch describes a torsional spring coupled to a linear spring. A compliant mate can have from one to six DOS and the relationship between the motion allowed and forces transmitted by these DOS is contained in a stiffness matrix for the mate. In this work all compliant mate stiffnesses are taken to be linear.

A Degree-Of-Virtuality, or DOV (the abbreviation DOV refers to both a single Degree-of-Virtuality or multiple Degrees-of-Virtuality) is just like a DOF, except it is a sub-element of virtual mate instead of a real mate (real mates cannot have DOV). A DOV always has a nominal value of zero, a pitch that is either zero or infinite, and a direction that is always aligned with one of coordinate axes of the feature at the end of an open-loop (more will be said later about the direction of DOV).

It will also be assumed that the number and direction of DOF, DOC and DOS of all real mates in an assembly are constant. It is also assumed that the DOF, DOC, and DOS of all mates are **bidirectional**. For a DOF, DOC, or DOS to be bidirectional, one of the corresponding conditions must hold:

- **For DOF, motion in both the positive and negative sense must be allowed**
- **For DOC, motion in both the positive and negative sense must be prohibited**
- **For DOS, motion in both the positive and negative sense must store energy**

The first condition means that mates such as one-way clutches are not permitted. The second condition requires that mates such as the bottom of a flat block resting on a level planar surface must not only prevent the block from moving normally toward the surface, but also must prohibit its motion normally away from the surface. The last condition means that compliant mates such as spring-loaded hinges must store energy when either opened or closed. Mates that are not bidirectional are often called unidirectional or one-sided [Whitney] and are not covered in this work. Virtual mates do not exist in the actual assembly but, since by definition they do not constrain motion in any direction, they are inherently bidirectional.

Some additional properties, **form closure** and **force closure**, are sometimes used to describe bidirectional DOC. DOC exhibiting form closure are bidirectional because of the mating features alone (e.g. the DOC of a pin in a hole with no clearance) [Blanding]. DOC exhibiting force closure are bidirectional because of an external force (e.g. gravity or a spring) [Green]. In this work, all DOC exhibiting force closure are assumed to retain their bidirectionality under all circumstances. In other words, features that are in contact remain in contact. In reality, whether a DOC exhibiting force closure is bidirectional or unidirectional is situation dependent. [Green] describes an **effector** as an element (e.g. spring or gravity) that insures that a DOC exhibiting force closure remains bidirectional at all times.

The number of real mates in an assembly, and the DOF, DOC, and DOS each real mate has both affect assembly behavior. As evidence, compare the behavior of the two tables in Figure 2.5. Both have the same parts-status and loop-status but since the mates allow different DOF, they exhibit different behavior.

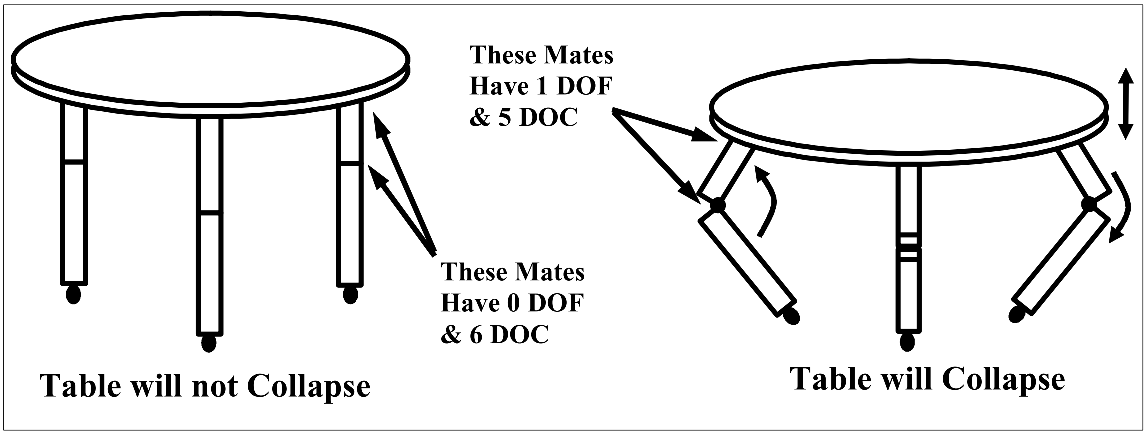


Figure 2.5. Two Tables Only Differing in Type of Mates

Nominal Part Properties: During the design phase, parts are assigned many nominal properties. These properties, combined with the mates, part-status, and loop-status, determine in large part the behavior of the assembly. In this work, nominal part properties will be divided into the following three categories:

- **Nominal Geometry**
- **Nominal Stiffness**
- **Other Nominal Properties**

Nominal part geometry in this work refers specifically to the set intra-part relationships that uniquely (i.e. the part geometry must not be over-specified) and completely (i.e. the part geometry must not be under-specified) describes the location of all features in a part. Each intra-part relationship is composed of a chain of serially-linked dimensions and each dimension is described by specifying a nominal value and a direction (expressed in coordinates local to that dimension). Nominal part geometry can and most likely will affect assembly behavior. As an example, imagine a table similar to one of the ones in Figure 2.2 except that the design has been modified such that one leg is much shorter than the other two legs. The levelness of this table will be much different than the levelness of the original table. This discrepancy is completely due to the nominal geometry and not the part-status, loop-status, or mates.

Nominal stiffness is a measure of the rigidity of part in the presence of applied loads. Sometimes it is defined in terms of two or more features on a part and how those features move relative to each other when known, yet balanced, forces are applied to those features. Nominal stiffness is dependent, among other things, on the nominal geometry (e.g. a thicker plate generally has higher stiffness) and the nominal properties of the material that the part is composed of (e.g. rubber has relatively low stiffness compared to steel).

There are many assemblies whose behavior is affected by the stiffness of one or more of their members (e.g. wheel suspension, garage door, car fender made of sheet metal) and such assemblies are often referred to as compliant assemblies [Goldenshteyn]. To illustrate this point, imagine the how the ability of one of the tables in Figure 2.2 to resist applied loads would differ if the legs were made out of soft rubber instead of wood or steel. Even if the assembly experiences no external forces during assembly or use, the nominal stiffness of the parts might still influence assembly behavior (e.g. its sensitivity to dimensional variability).

As mentioned in Chapter 1, compliant parts are modeled in this work as sets of two or more smaller rigid parts connected by one or more linearly-compliant mates, each with one or more DOS [Donald and Pai]. Such an approach creates what is often called a lumped-parameter approximation that in many cases is a sufficiently accurate representation of the actual situation. While this work gives no recommendation as to what the ideal lumped-parameter approximation is for a given part or what error is introduced by adopting such an approach, there are many techniques cited in the literature (see [Howell], [Fasse and Zhang], and [Patterson and Lipkin, 1993b]). Parts with distributed stiffness (e.g. a flimsy sheet metal plate), where neither the assumption of infinite stiffness nor lumped finite stiffness is valid are not covered in this work. [Heubner, Dewhirst, Smith, and Byrom], [Shiu, Ceglarek, and Shi], and [Villarreal and Asada], and others discuss methods for modeling such parts. Also, due to the definition adopted in this work for compliant mates, parts with nonlinear stiffness, even in a lumped finite stiffness sense, are not covered in this work.

In addition to nominal stiffness and geometry, there are many other nominal part properties whose values are determined during design and affect assembly behavior. Among these are material strength, mass, density, surface finish, fatigue resistance, thermal conductivity, electric resistivity, emmisivity, thermal expansivity, and viscosity, just to name a few. For example, the behavior of an aircraft wing is not just dependent on its shape and stiffness, but also its weight, strength, and fatigue resistance. In this work, all such nominal part properties can only be modeled if they can be expressed in terms of the geometry or the compliance of the parts (e.g. express the mass of a part as a function of the part's dimensions).

Constraint-Status: The constraint-status is a property of the nominal design that indicates the nature of the relative motion allowed between parts and forces transmitted between parts in an assembly via their mates. The constraint-status of an assembly is dependent on its part-status, loop-status, its mates, and the nominal properties of its parts [Whitney]. At the highest level, the constraint-status of an assembly falls into one of the following four categories:

- **Properly-Constrained**
- **Under-Constrained**
- **Over-Constrained**
- **Both Over-Constrained & Under-Constrained**

Properly-constrained assemblies are those in which each and every part is uniquely located in all six coordinate directions by one or more mates it shares with other parts in the assembly [Kriegel]. Under-constrained assemblies are those in which there is at least one DOF of excess relative motion allowed between at least one pair of parts [Phillips, 1984]. Over-constrained assemblies or are those in which at least two parts share mates that enforce one or more of the same DOC [Whitney]. Finally, it is possible for an assembly to be both over-constrained and under-constrained at the same time [Davies, 1983b]. Loops and individual parts can also be referred to as properly-constrained, under-constrained, over-constrained, or both over-constrained and under-constrained (the only exception is that open-loops cannot be over-constrained). Individual mates exhibiting form closure can also be over-constrained as described by [Slocum] but such over-constraints are not included in this work.

Figure 2.6 shows four tables, each with a different constraint-status: one properly-constrained, one under-constrained, one over-constrained, and one both over-constrained and under-constrained (each post in the over-constrained tables is being located in the six coordinate directions by three different mates: one it shares with the top and two it shares with the two braces).

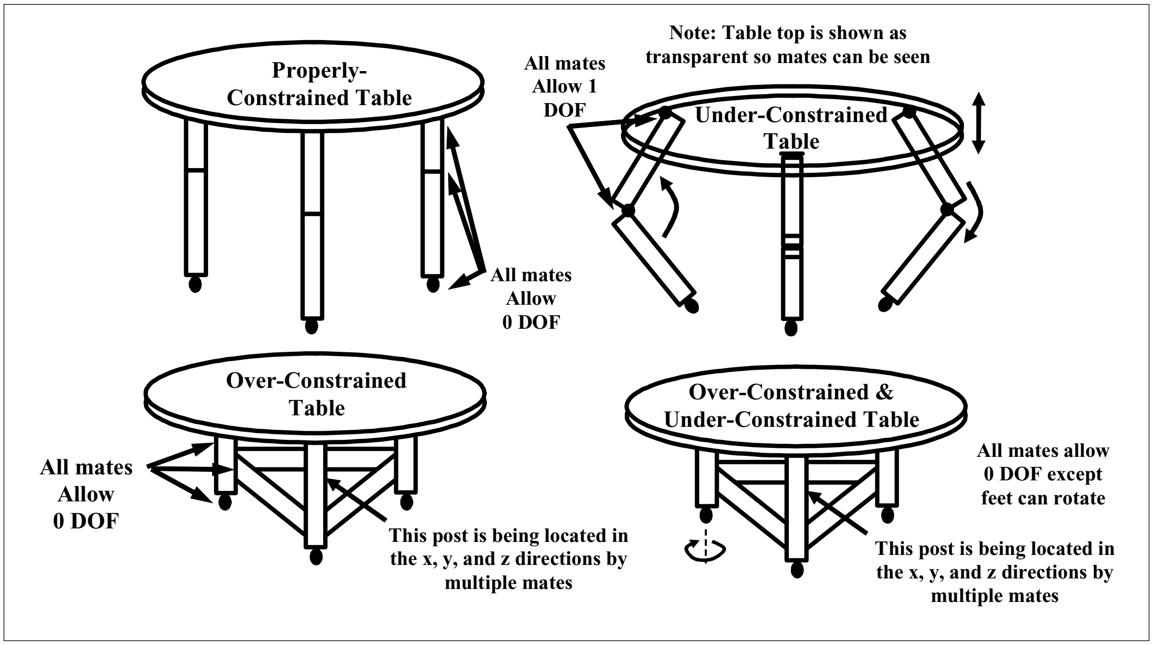

Properly-Constrained Table
All mates Allow 0 DOF
Note: Table top is shown as transparent so mates can be seen
All mates Allow 1 DOF
Under-Constrained Table
Over-Constrained Table
All mates Allow 0 DOF
This post is being located in the x, y, and z directions by multiple mates
Over-Constrained & Under-Constrained Table
All mates allow 0 DOF except feet can rotate
This post is being located in the x, y, and z directions by multiple mates

Figure 2.6. Four Tables Each with a Different Constraint-Status

Problems can arise when unintended over-constraints or under-constraints exist in an assembly. For example, if the under-constraints in the upper-right table in Figure 2.6 were accidental, the table may unexpectedly collapse during use. If the over-constraints in the bottom tables in Figure 2.6 were accidental, the table may be much harder to assemble than anticipated, especially if the parts are not fabricated to the exact nominal dimensions (which they won't be). There are many examples in the literature of such mistakes (see [Kriegel], [Hart-Smith], and [Sweder and Pollack]).

Due to the fact that under-constrained assemblies (and assemblies that are both over- and under-constrained) allow relative motion between their parts, a single snapshot of how the parts are connected is not sufficient to completely describe the assembly. In this work, this is overcome by choosing a finite set of discrete **configurations** of the assembly. Two configurations of the same assembly are distinguished from each other by looking at the locations of the parts relative to each other as expressed in the nominal values of the DOF of the mates. Figure 2.7 shows two possible configurations for the under-constrained table in the upper right corner of Figure 2.6. These two configurations are distinguished from each other by the comparing the angles between the braces and posts. Such an approach is often called an incremental approach and has been used by [Konkar] and [Konkar and Cutkosky] to analyze and synthesize assemblies.

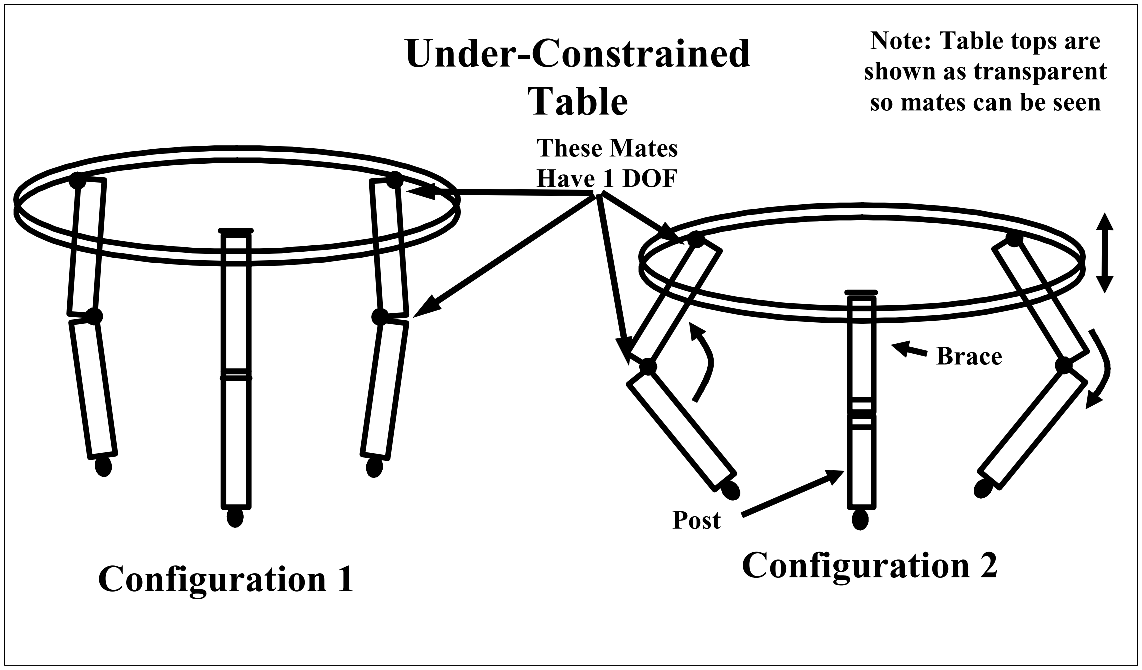


Figure 2.7. Two Different Configurations of the Same Table

2.2.2 Manufacture

Equally as important as the design of the assembly, is the process that is used to physically create it. Three main areas of the physical creation process are the following:

- ***Part Assembly***
- ***Part Fabrication***
- ***Part Inspection***

Each of these will now be briefly discussed.

Part Assembly: Part assembly is the process designed to take parts and connect them together into a single assembly. The following two aspects of part assembly can potentially affect assembly behavior (one of which is further divided into two categories):

- **Fixtures**
  - **Holding Fixtures**
  - **Locating Fixtures**
- **Adjustments**

Here a fixture is defined as anything that facilitates the process of relating parts to each other (e.g. a clamp, aligning jig, measurement tool, etc) [Campbell]. In this work two types of fixtures are identified: holding fixtures and locating fixtures. Holding fixtures merely facilitate the process of joining parts together by placing them in close proximity to each other. Due to their passive role, holding fixtures are not involved in any way in the permanent locations of the parts relative to each other and so in the context of this work cannot affect assembly behavior.

Locating fixtures, on the other hand, do take an active part in setting geometric relationships between features on different parts and therefore can affect assembly behavior. Locating fixtures are just like parts in every way except they do not exist in the final completed assembly. As such, the term “part” in this work will refer to either a part or a locating fixture. While locating fixtures may also hold parts secure, they more importantly enforce temporarily one or more geometric relationships between two or more parts. Before the locating fixtures are removed, a joining process is used to make these geometric relationships permanent so that when the locating fixtures are removed, the relative locations of the parts as set by the fixtures remains.

When locating fixtures are used, the performance of the assembly is dependent as much on the properties of the locating fixtures as it is on the properties of the parts. As an example, compare the two different ways to assemble legs for a table as shown in Figure 2.8. The one on the left uses a holding fixture whereas the one on the right uses a locating fixture. It is clear that the straightness and length of the legs that come out of the assembly process on the right are dependent on the fixture whereas the properties of the legs assembled using the process on the left are independent of the fixture.

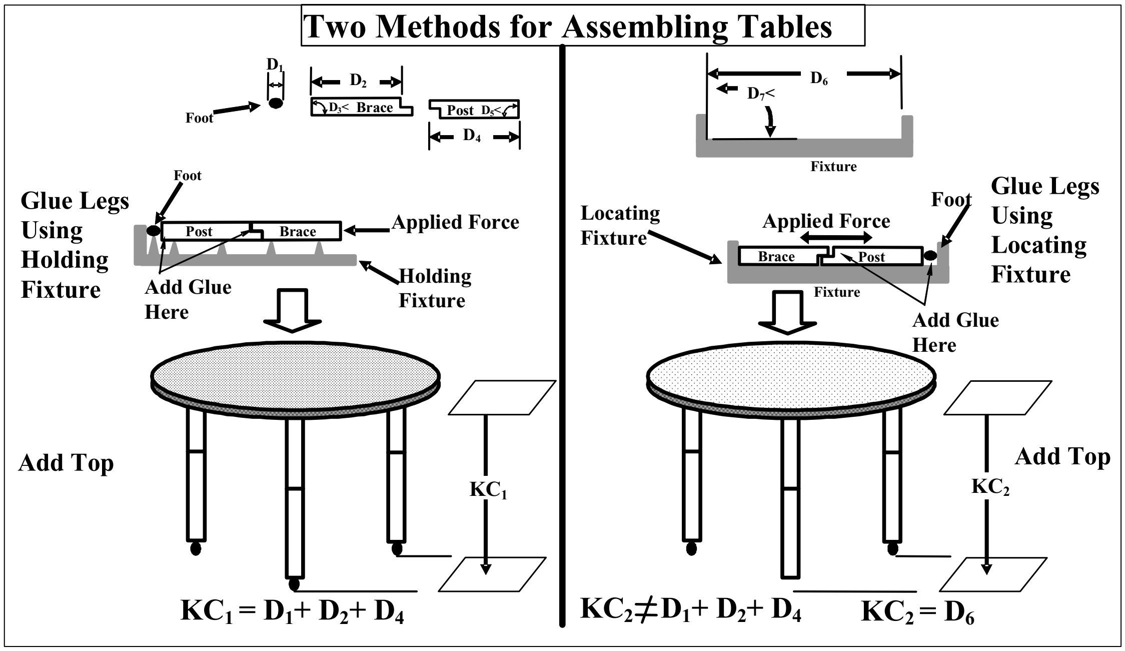


Figure 2.8. Comparing a Holding Fixture to a Locating Fixture

Adjustments are the other aspect of part assembly that is relevant to this work. Adjustments are basically a combination of an inspection process, a calibration process, and a joining process [Adams]. First, a relationship between two features on different parts is measured and then it is adjusted to be within a certain target range by exploiting excess under-constraints that were intentionally left in the assembly during design. Once the relationship is set, a joining process (e.g. welding or bonding) is used to secure that geometric relationship and remove the excess under-constraints. Whether using the human eye or a precision measurement tool, all adjustments serve the identical purpose as locating fixtures. For this reason, the term "locating fixture" in this work will refer to a locating fixture or an adjustment (this also means that the term "part" will refer to a part, a locating fixture, or an adjustment).

The way in which an assembly is physically put together is referred to as the assembly's **type-status** [Whitney]. The type-status of an assembly is unrelated to the actual number of assembly steps used or the number of parts. Instead it depends on whether or not locating fixtures are used during assembly. The type status of any assembly falls into one of the following two categories:

- **Type-I**
- **Type-II**

Assemblies that are assembled without any locating fixtures are categorized as Type-I. The three-legged table shown on the left side of Figure 2.8 is an example of Type-I assembly. In such cases the locations of all the parts in the assembly are completely determined by the mates they share with other parts in the assembly. Assemblies requiring one or more locating fixtures are categorized as Type-II [Mantripragada and Whitney, 1998]. The table shown on the right side of Figure 2.8 is an example of Type-II assembly. For Type-II assemblies, care must be taken to properly design, fabricate, inspect, and model not only the parts, but the fixtures and adjustments processes as well.

After each joining process involving a locating fixture, the assembly assumes a new state. The number of states an assembly has depends on the number of locating-fixture-dependent joining processes the assembly undergoes. A single fixture-dependent joining process begins when the assembly is placed in one or more locating fixtures and ends when the assembly is removed from at least one of those fixtures. During those two events multiple joining processes may actually occur but it is considered as a single process for the purposes of this work.

In general, the total number of states an assembly has is equal to one more than the number of fixture-dependent joining processes. This means that by definition Type-I assemblies only have one state and Type-II assemblies will always have more than one state. Further, each state of an assembly that is under-constrained (or both under-constrained and over-constrained) must be described in terms of one or more configurations.

To handle to various states in a Type-II assembly, concepts such as the Stream of Variation developed by [Camelio, Hu, and Ceglarek] and [Hu] and a State Transition approach presented by [Mantripragada and Whitney, 1997] have been proposed as methods for interrelating analytically the various assembly states of Type-II assemblies and account for the propagation of variability from state to state.

If an assembly state in a Type-II assembly is over-constrained, there exists the possibility that the locating-fixture-dependent joining process that separates that state from the next state will lock in stresses that will cause the newly mated parts to deform and change location relative to each other in an effort to find a lower energy state once the parts are mated and the locating fixture(s) is/are removed. This phenomenon is called **springback** and it not addressed in this work. For more on springback see [Hu], [Camelio, Hu, and Ceglarek], [Goldenshteyn], and [Liu and Hu].

Part Fabrication: When part designs are released for fabrication and parts are actually made, the properties of these physical parts will always differ, even if only slightly, from the nominal design due to the limited capabilities of the fabrication processes. For this reason dimensions are often assigned tolerances or tolerance zones [Foster]. Understanding the relationship between the nominal assembly design and the variability in the part fabrication processes is critical to understanding assembly behavior. In the case of the three-legged tables in Figure 2.2, if the geometry of the nine actual parts that make up the table legs deviates from the specified nominal geometry, the table might not be level when placed on a level floor even if the design dictates that is should be in theory. Part fabrication not only includes processes such as casting and machining, but any process that alters the properties of the parts. This could even be a process that inadvertently deforms the parts (e.g. a conveyor that dents the parts as it transfers them from one station to another).

In this work it is assumed that only those fabrication processes that set the dimensions of the parts exhibit variability. All other fabrication processes, including those that determine the stiffness of the compliant mates, are assumed to exhibit no variability. Further, even though dimensional variability may cause stiffness variability (e.g. a thicker beam will have a higher stiffness), such interactions are not covered in this work. This means that the stiffnesses of the compliant mates remain unchanged at their nominal values as contained in the corresponding stiffness matrices. If it is essential that the effect of variability in the compliant mate stiffnesses be addressed, the only alternative presented here is to choose two or three designs differing only in the stiffness values of the compliant mates and create and compare the models and results obtained from those models for each case. Unless this is done, the variability in the KCs will be dependent solely on the variability in the dimensions.

As mentioned in Chapter 1, the variability of any one dimension is assumed to be independent of the variability of any other dimension. A suggestion for eliminating the need for this assumption will be discussed in Chapter 7.

In reality, dimensional variability can have many adverse effects on assembly behavior. Sometimes such variability can cause a change in loop-status and/or constraint-status. Notice in Figure 2.9 how the loop-status and constraint-status of a table changes in if the shaded braces are much longer than designed. Also notice that the post in the lower right changes from being properly constrained to over-constrained.

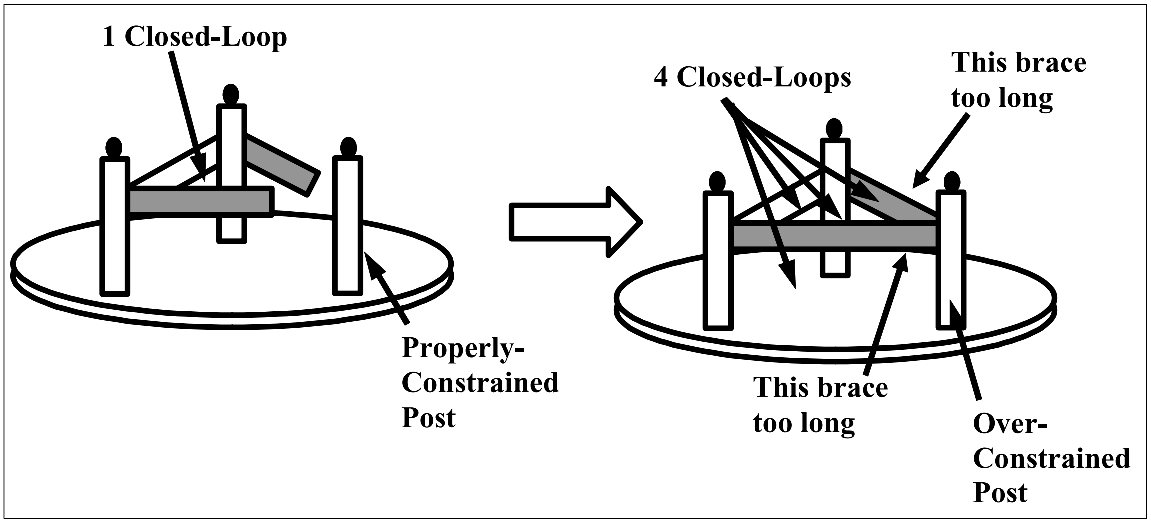


Figure 2.9. Effect of Variability on an Assembly's Part-Status & Constraint-Status

Sometimes dimensional part variability can cause a change in the number of relative DOF or DOC between two parts. Consider the block in a fixture shown in Figure 2.10. The block is designed to have one DOF with respect to the fixture. If the block is too long, it may suddenly have zero DOF with respect to the fixture. Also notice how the constraint-status of the assembly changes from being under-constrained to over-constrained.

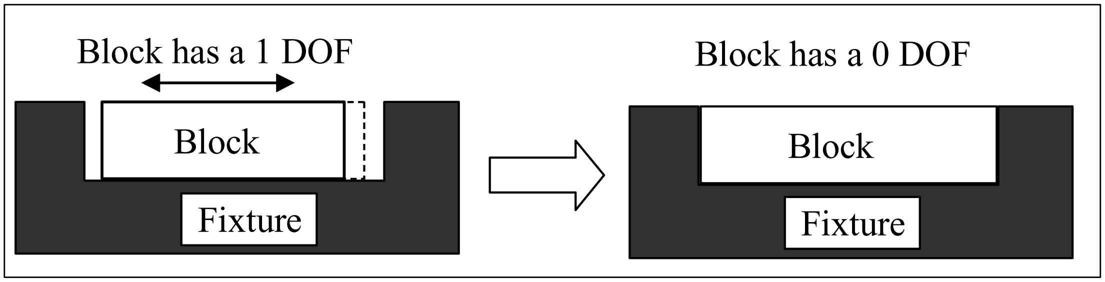


Figure 2.10. Effect of Variability on the DOF Allowed By a Mate

In this work it is assumed that dimensional variability does not cause the loop-status or constraint-status of an assembly to change. If there exists a significant risk that the loop-status, constraint-status, or the number and/or direction of the DOF, DOC or DOS of the mates of an assembly might change due to the dimensional variability, one or more KCs should be added to guard against such undesirable changes [Whitney].

A dimension whose value is uncertain (due to the limited fabrication process that sets it) can be modeled as a continuous random variable. Random variables are commonly defined in terms of two functions, the probability density function, or PDF, and the cumulative distribution function, or CDF [Drake]. The CDF and PDF of a random variable are related so by knowing one, the other can be obtained, either mathematically or numerically. Rather than define the exact value of a random variable, a PDF defines the probability density for each value of the random variable. The area under any PDF function is always one but in general the value returned by an arbitrary PDF can be any non-negative real number. A CDF is the integral of the PDF and defines the probability that the variable will take on a value that is less than or equal to a certain value. The CDF always return a value between zero and one [Drake].

While in theory a random variable can be assigned any PDF (as long as it integrates to one), in practice random variables are usually assigned PDFs that can be parameterized mathematically. A PDF is parameterized by specifying the following two things:

- **Distribution Type**
- **Distribution Parameter(s)**

The distribution type specifies the general “shape” of the PDF. Normal, Poisson, and Uniform are all examples of distribution types (many others exist) and each has its own unique properties. Figure 2.11 shows the PDF and CDF for these distributions. With each distribution type there are an infinite number of actual distributions, but all have the same general “shape”.

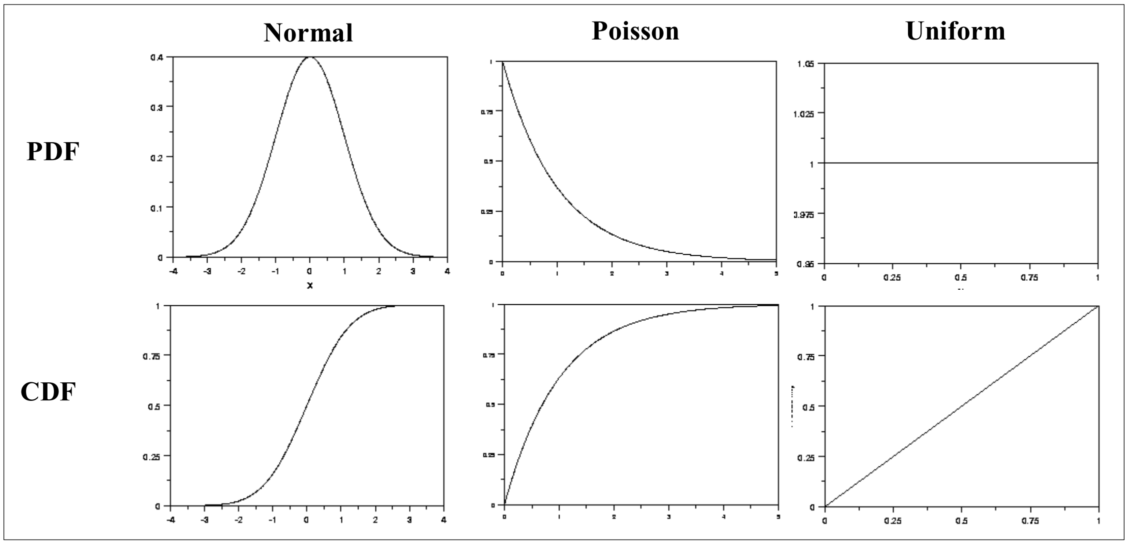


Figure 2.11. The PDF and CDF for the Normal, Poisson, and Uniform Distributions

Distribution parameters are what distinguish one distribution of the same type with another distribution of that same type. For example, in the case of distributions that are of the Normal type, the mean value, generically denoted $\boldsymbol{m_x}$, and the standard deviation, generically denoted $\boldsymbol{\sigma_x}$, of those distributions distinguish them from each other. If two distributions are both Normal and both have the same mean and standard deviation, then they are identical. For distributions of the Uniform type, the characteristic parameters are the upper and lower limits on the variable, denoted generically, $\boldsymbol{ul_x}$ and $\boldsymbol{ll_x}$. These are just two of the many distribution types that exist.

In practice, the PDF and CDF functions for a dimension-setting fabrication process are usually obtained by measuring a sample of dimensions set by that fabrication process, plotting them, and choosing the distribution type (e.g. Rayleigh, Gamma, Uniform, Normal, etc) and associated parameters that result in a distribution that most closely matches the data.

Part Inspection: When a batch of parts is fabricated, there will always be variability in the properties of the parts in the batch. Often the properties of these parts are then inspected (either one by one or in some other manner) to see if they actually have values at or near the nominal design values. Sometimes parts are discarded or "reworked" based on knowledge gained from this inspection process. Discarding or reworking parts can alter the shape of a PDF function (e.g. an inspection process might narrow the "shape" of a distribution or truncate its tails). Since inspection processes alter the PDF and CDF, they can influence or reduce the variability exhibited by the KCs, which is precisely why they are usually carried out in the first place. Imagine for example, a three-legged table manufacturing facility that includes a station where all the legs were inspected and those that were longer than a certain length, denoted ***l***, are discarded. Finished tables leaving this manufacturing facility could never be taller than ***l*** plus the thickness of the tabletop.

It is important to note that some inspection processes, like those based on zones derived from Geometric Dimensioning & Tolerancing, or GD&T, can create dependencies between variables [Foster]. The effects of such inspection processes are not included in this work but are addressed in the literature. For example [Davidson and Shah] and [Davidson, Mujezinović, and Shah] look at the cases where GD&T is applied to rectangular and round faces respectively and [Whitney and Jastrzebski] suggest a multivariate statistical model to deal with the dependencies between tolerances of size and orientation of a surface.

2.2.3 Use Environment

The third are has to do with the environment in which the assembly is used or assembled. The following three subcategories in this area are addressed here:

- ***Applied Forces & Moments***
- ***Temperature***
- ***Other Use Factors***

Each of these subcategories will now be discussed.

Applied Forces and Moments: Applied forces and moments can be either static or dynamic and both types can affect assembly behavior. Sometimes these applied forces are intentionally applied to the assembly and are controlled so that they align with one or more intentionally designed-in under-constraints, as is the case with the turning of a crankshaft in an engine by the forces of gasoline combustion. Other times these forces are not foreseen and become aligned with unintentionally designed-in under-constraints, like in the example of the collapsing three-legged table shown in Figure 2.5. While including the effects of externally applied forces is not discussed in detail in this work, a suggestion for incorporating static external forces will be made at the end of Chapter 3.

Temperature: Both the geometry and stiffness of parts are often dependent on temperature. KCs can also be dependent on temperature. One could imagine that all of the tables in Figure 2.2 might get slightly taller if their temperature increased and the legs expanded in response. In this work, the effects of temperature are only considered indirectly. This is accomplished by first assuming that the nominal part geometry is defined for a specific temperature (e.g. room temperature). Then, knowing the coefficient of thermal expansion for the materials of the parts, the variability in temperature can be modeled as an equivalent variability in part geometry [Paz and Leigh]. While no more mention of temperature effects will be made here, it should be assumed when dimensional variability is discussed, that the effects of temperature can be included in the manner just described. The dependence of other part properties on temperature (e.g. stiffness or fracture strength) is also not considered. If a customer's expectations are related to temperature, such expectations should be expressed using geometrically equivalent KCs.

Other Use Environment Factors: In addition to applied forces and temperature, there are many other environmental factors that effect assembly behavior. Among these are humidity, pressure, atmospheric composition, exposure to light, exposure to sound, just to name a few. While ideally a modeling technique would be able to capture all these interactions, this methodology can only capture them if they can be expressed in terms of the part geometry or part stiffness.

2.2.4 Performance Measures

In general there are numerous ways that assembly performance can be measured. In this work, the only way in which performance is measured is via the KCs. As defined above, a KC, or Key Characteristic, is a one-dimensional geometric relationship between two features with a nominal value and an upper or lower specification limit. Under this definition, a one-dimensional geometric relationship with both an upper and lower limit must be represented by two separate KCs and a two-dimensional geometric relationship (e.g. the distance between two features in both $\boldsymbol{x}$ and $\boldsymbol{y}$ directions) with both an upper and lower limits in each direction must be represented by four KCs.

Any two features that are related by a KC are called a **KC feature pair**. In general a KC feature pair can related by anywhere from zero to twelve KCs (up to six directions and up to two limits in each direction). In this work the following two types of KCs are defined (one of which is further divided into two categories):

- **Performance**
    - **Clearance**
    - **Alignment**
- **Linking**

Performance KCs are one-dimensional relationships between features that act as quantitatively measurable surrogates for those attributes or functions of an assembly that are important to the customer (e.g. cost, power, efficiency, etc). All performance KCs have a nominal value and a specification limit that represent respectively the expectation of the customer and their willingness accept substandard performance. Here, the manufacturability of an assembly will always be measured by comparing the variability in the value of the performance KCs relative to their specification limits. In other words, two identical assemblies may have different levels of manufacturability if they have different sets of performance KCs (differing either in the features they relate, their nominal values, or their specification limits).

Performance KCs are further divided into clearance KCs and alignment KCs. Clearance KCs are binary in the sense that as long as the value is above a certain value, the functional requirement(s) related to that KC are met without any degradation in performance. The specification limit for a clearance KC is a fixed positive number and as such the larger the nominal value of a clearance KC, the more variability it can exhibit without affecting performance. Clearance KC's are similar in some respects to the "larger-is-better" quality characteristics presented by [Taguchi, 1992]. Conversely, alignment KCs are not binary which means that any deviation from nominal represents a degradation in performance and any deviation outside the specification limit is deemed unacceptable. Alignment KC's are similar in some respects to the "nominal-is-better" quality characteristics discussed by [Taguchi, 1992].

Linking KCs are used only to connect together different assembly states of the same assembly. The number of assembly states an assembly has depends on the assembly process as discussed in the manufacture section above. Linking KCs differ from performance KCs in that they do not reflect the expectations of the customer directly but serve as vehicles for stringing together the various states of a Type-II assembly (Type-I assemblies do not have linking KCs). Therefore, linking KCs will have a nominal value and a direction, but no specification limit.

As one final note, it is important to realize that for assemblies containing one or more under-constrained (or both over-constrained and under-constrained) assembly states, it is possible that the KCs for the under-constrained state(s), may not be all defined in the same configuration. In general, one need only specify enough configurations for an under-constrained assembly state to capture all the KCs.

2.3 Summary

When modeling assemblies, lack of attention to any of the factors discussed above, including those not addressed by this thesis, can lead to inaccurate conclusions. Table 2.1 shows the factors discussed above divided into three categories. The category on the left includes all the factors affecting the nominal assembly behavior. The middle category includes all the factors that influence the variability in assembly behavior. The category on the right lists those factors not considered in this work. Caution should be taken whenever applying the methodology presented in this thesis to situations where the factors in the third category are present and deemed significant.

Nominal Factors Included in this Work	Influences on Variability Included in this Work	Factors Not Included in this Work
Part-Status	Dimension-Setting Fabrication Processes	Unidirectional Mates
Loop-Status		Mate Friction
Mate Properties	Part Inspection Processes	Distributed Part Stiffness
Part Geometry	Stiffness-Setting Fabrication Processes (lightly)	Non-Linear Part Stiffness
Part Stiffness		Nominal Part Mass
Constraint-Status		Other Nominal Part Properties
Assembly Process		Variability in Other Nominal Part Properties
Fixtures & Adjustements		
Type-Status		Dynamic Applied Forces
KCs		Other Time-Dependent Factors
Temperature (indirectly)		Variability in Applied Forces
Static Applied Forces (lightly)		Other Environmental Factors
		Variability in Other Environmental Factors
		Springback
		Dependent Dimensional Variability

Table 2.1. Characteristics of Assemblies Relevant to This Work

3. MODELING MECHANICAL ASSEMBLIES

In the previous chapter, a list of important factors that affect assembly behavior was presented and discussed. In this chapter, a unified approach for modeling assemblies that captures much of this behavior is presented. This methodology is composed of the following two phases:

- **The Element Identification & Location Phase**
- **The Element Quantification & Integration Phase**

The first half of this chapter describes the Element Identification & Location Phase. During this phase the various elements of an assembly that are relevant to manufacturability are identified and located relative to a global reference frame. The important assembly elements include: assembly states, assembly configurations, parts, features, mates, intra-part relationships, closed-loops, open-loops, and KC-feature pairs. The second section of this chapter presents the Element Quantification & Integration Phase. During this phase, each of the elements is represented analytically and these various analytical representations are integrated into a coherent, unified matrix-based assembly model. The model is partitioned into two matrices such that one matrix captures all the nominal properties of the assembly and the other matrix captures the variability properties of the assembly. The final section of this chapter discusses the nature of the model and compares it to the traditional matrix-based model used in static analysis of structures.

To facilitate the understanding of the modeling procedure employed here, after each task is explained, its application will be illustrated using a fingernail clipper (see Figure 3.1). To eliminate complexity that only complicates understanding, the file and rivet (used to attach the file) that most common fingernail clippers have will not be discussed. This leaves only three distinct entities in its final assembled state: the cutter, the handle, and the pin[3].

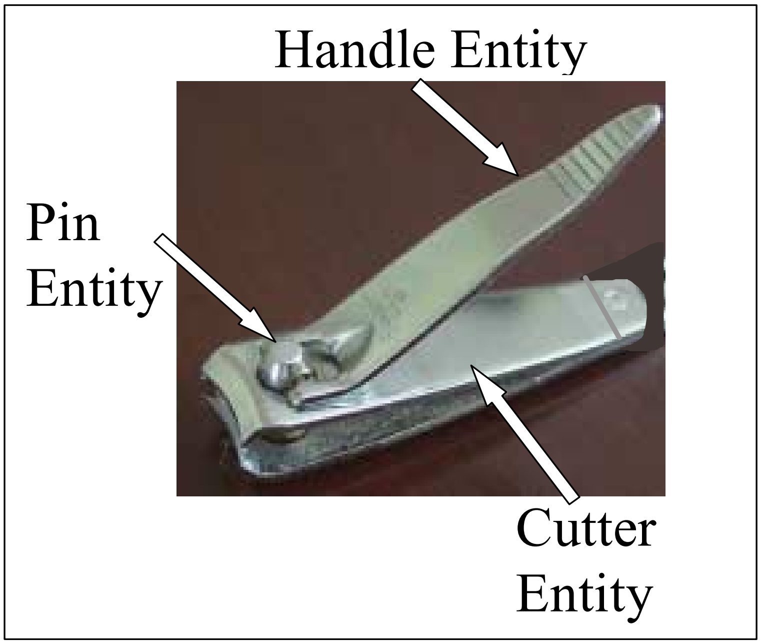


Figure 3.1. Photo of Fingernail Clipper with Parts Labeled

3.1 Element Identification & Location Phase

The first phase of the modeling methodology is to identify all the elements of the assembly that are relevant to the manufacturability of the assembly. Figure 3.2 shows a flow diagram for this phase. Each box corresponds to a task that must be completed and

[3] the term "entity" is used instead of "part" because the term "part" has a specific meaning in this work.

the arrows between the boxes indicate the order in which they should be completed. The numbers in parentheses indicate which subsection of Section 3.1 addresses the corresponding task.

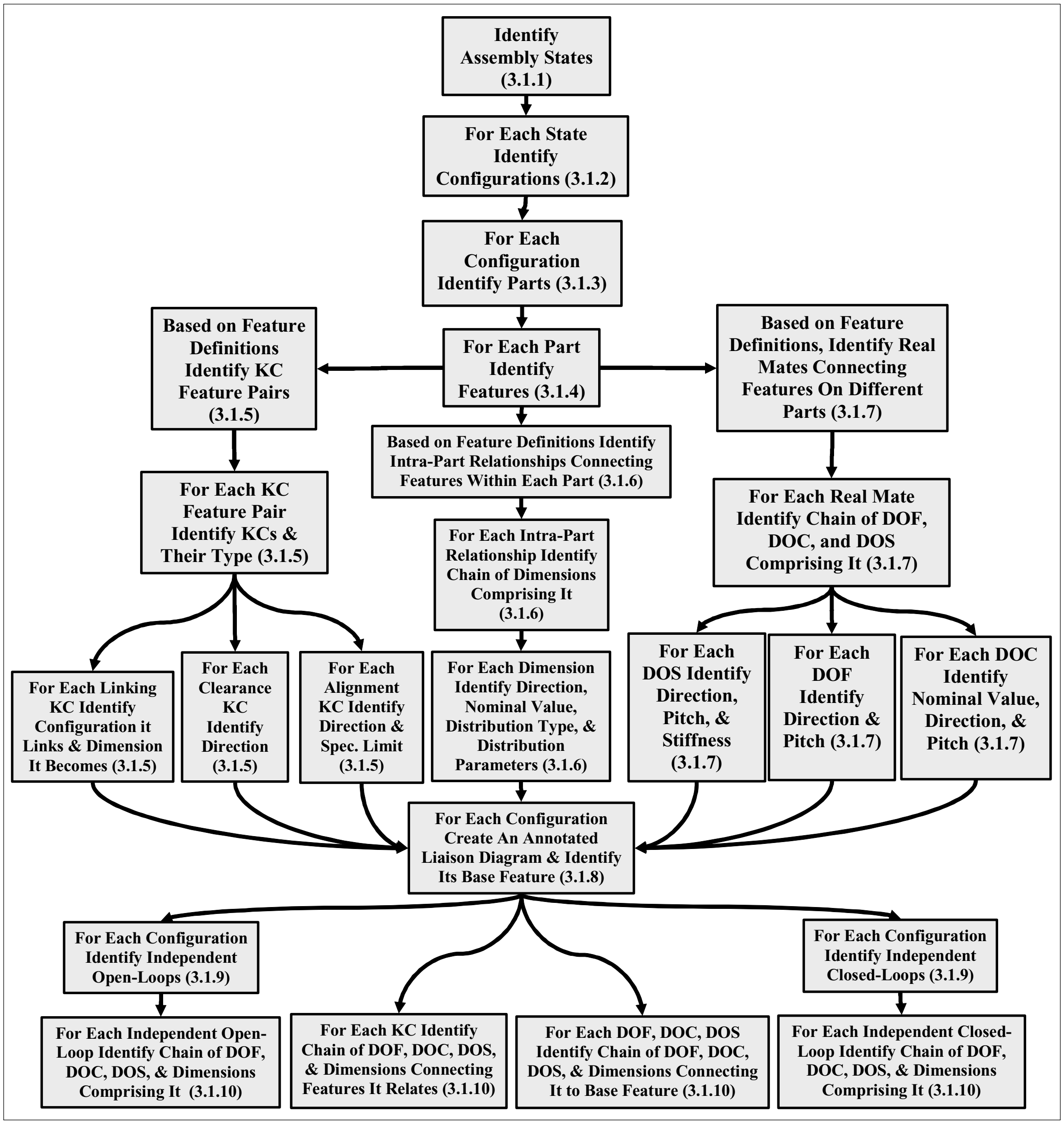

Identify Assembly States (3.1.1)
For Each State Identify Configurations (3.1.2)
For Each Configuration Identify Parts (3.1.3)
Based on Feature Definitions Identify KC Feature Pairs (3.1.5)
For Each Part Identify Features (3.1.4)
Based on Feature Definitions, Identify Real Mates Connecting Features On Different Parts (3.1.7)
Based on Feature Definitions Identify Intra-Part Relationships Connecting Features Within Each Part (3.1.6)
For Each KC Feature Pair Identify KCs & Their Type (3.1.5)
For Each Intra-Part Relationship Identify Chain of Dimensions Comprising It (3.1.6)
For Each Real Mate Identify Chain of DOF, DOC, and DOS Comprising It (3.1.7)
For Each Linking KC Identify Configuration it Links & Dimension It Becomes (3.1.5)
For Each Clearance KC Identify Direction (3.1.5)
For Each Alignment KC Identify Direction & Spec. Limit (3.1.5)
For Each Dimension Identify Direction, Nominal Value, Distribution Type, & Distribution Parameters (3.1.6)
For Each DOS Identify Direction, Pitch, & Stiffness (3.1.7)
For Each DOF Identify Direction & Pitch (3.1.7)
For Each DOC Identify Nominal Value, Direction, & Pitch (3.1.7)
For Each Configuration Create An Annotated Liaison Diagram & Identify Its Base Feature (3.1.8)
For Each Configuration Identify Independent Open-Loops (3.1.9)
For Each Configuration Identify Independent Closed-Loops (3.1.9)
For Each Independent Open-Loop Identify Chain of DOF, DOC, DOS, & Dimensions Comprising It (3.1.10)
For Each KC Identify Chain of DOF, DOC, DOS, & Dimensions Connecting Features It Relates (3.1.10)
For Each DOF, DOC, DOS Identify Chain of DOF, DOC, DOS, & Dimensions Connecting It to Base Feature (3.1.10)
For Each Independent Closed-Loop Identify Chain of DOF, DOC, DOS, & Dimensions Comprising It (3.1.10)

Figure 3.2. Element Identification Phase

Before explaining how to complete these tasks, a brief discussion on notation is in order. Throughout this thesis, a lower-case variable (a variable may be more than a single character) without brackets will always refer to a quantity of elements (e.g. dtt or $m_{i,j}$). An upper-case variable without brackets will always refer to an actual element (e.g. A_i or $M_{i,j,k}$). A bracketed variable will always refer to either a vector or matrix of values assigned to a set of elements (e.g. $[Dn]$, $[KCn]_{i,j}$, or $[d^l]_{i,j,k}$). A bracketed variable with one or two subscripts inside the brackets will always refer to a specific value within a vector or matrix (e.g. $[Cd_i]$ or $[H_{l,n}]_{i,j,k}$). A bracketed variable with subscripts inside the brackets that are separated by colons refer to a sub-matrix of the vector or array (e.g. $[f^l_{1:3,3:4}]_{i,j,k}$ would be a $(3\text{x}2)$ matrix containing the first three entries in the third and fourth columns of matrix $[f^l]_{i,j,k}$). Subscripts in parenthesis inside the brackets of a bracketed variable refer to the size of the vector or matrix (e.g. $(k \text{ x } l)$ in $[G_{(k \text{ x } l)}]_{i,j}$). Finally, empty matrices are denoted by $[\emptyset]$, matrices of all zeros are denoted by $[0_{(i \text{x} j)}]$, where i and j are the number of rows and columns respectively, and identity matrices are denoted by $[I_{(i\text{x}i)}]$ where i is the size of the matrix.

3.1.1 Identifying Assembly States

As mentioned in Chapter 2, the number of states, denoted a, of an assembly is equal to one more than the number of locating-fixture-dependent joining processes. For any assembly, its states can be labeled generically as A_i such that the set of all assembly states for that assembly is A_1, A_2,…, A_a. In the case of the fingernail clipper, careful study reveals that the cutter entity is created by a single welding process. Since welding is a fixture-dependent joining process and that is the only such process used in its manufacture, the fingernail clipper must have two assembly states ($\boldsymbol{a} = 2$). These states will be labeled $\boldsymbol{A_1}$ and $\boldsymbol{A_2}$. $\boldsymbol{A_1}$ is the state of the assembly before welding and $\boldsymbol{A_2}$ is the state after welding and when the fingernail clipper is completely assembled. Because it has more than one state, the fingernail clipper is a Type-II assembly.

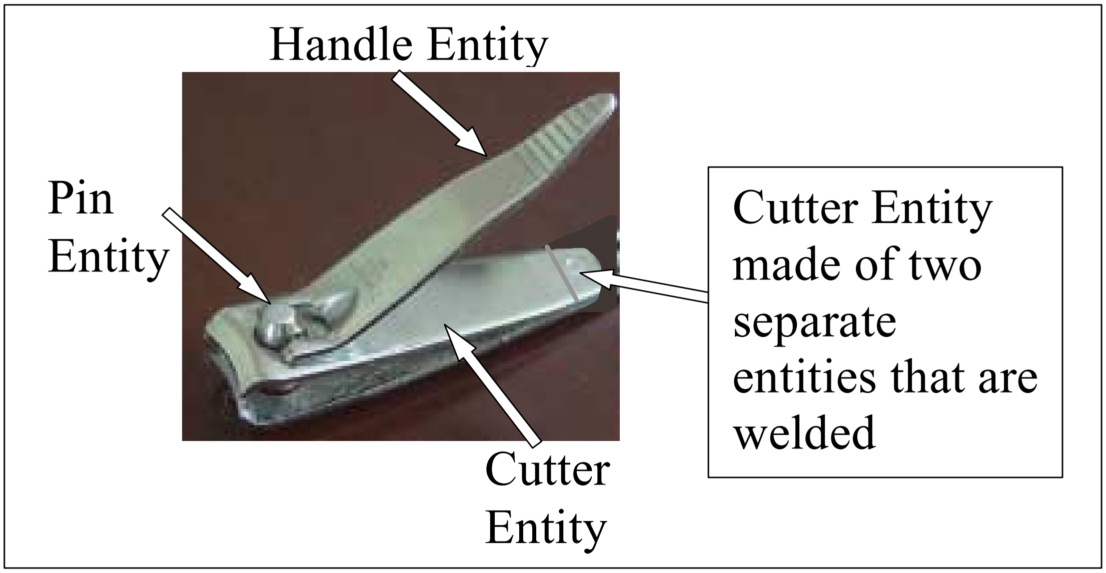


Figure 3.3. Photo of Fingernail Clipper Showing Weld Location

3.1.2 Identifying Configurations

Every assembly state has at least one configuration and for those assembly states that are under-constrained there can exist more than one. The configurations for an under-constrained assembly state should be chosen so that all the KCs relevant to that assembly state are included in at least one of the configurations. In general, the total number of configurations, $\boldsymbol{bt}$, for an assembly is determined by using the following equation:

$$bt = \sum_{i=1}^{a} b_i$$

Equation 3.1

In Equation 3.1, ***a*** is the number of assembly states and $\boldsymbol{b_i}$ is the number of configurations in the $\boldsymbol{i}^{\text{th}}$ assembly state. $\boldsymbol{b_a}$ would therefore be the number of configurations in the final assembly state. Individual configurations themselves are labeled with a double subscripted ***B*** such that $\boldsymbol{B_{i,j}}$ is the $\boldsymbol{j}^{\text{th}}$ configuration in the $\boldsymbol{i}^{\text{th}}$ assembly state. For example, $\boldsymbol{B_{4,2}}$ is the second configuration of the fourth assembly state and $\boldsymbol{b_4}$ is the number of configurations in the fourth assembly state. Analysis of the fingernail clipper reveals state $\boldsymbol{A_1}$ has a single configuration and state $\boldsymbol{A_2}$ has four configurations ($\boldsymbol{b_1}$ = 1, $\boldsymbol{b_2}$ = 4, and ***bt*** = 5). All five configurations are shown schematically in Figure 3.4.

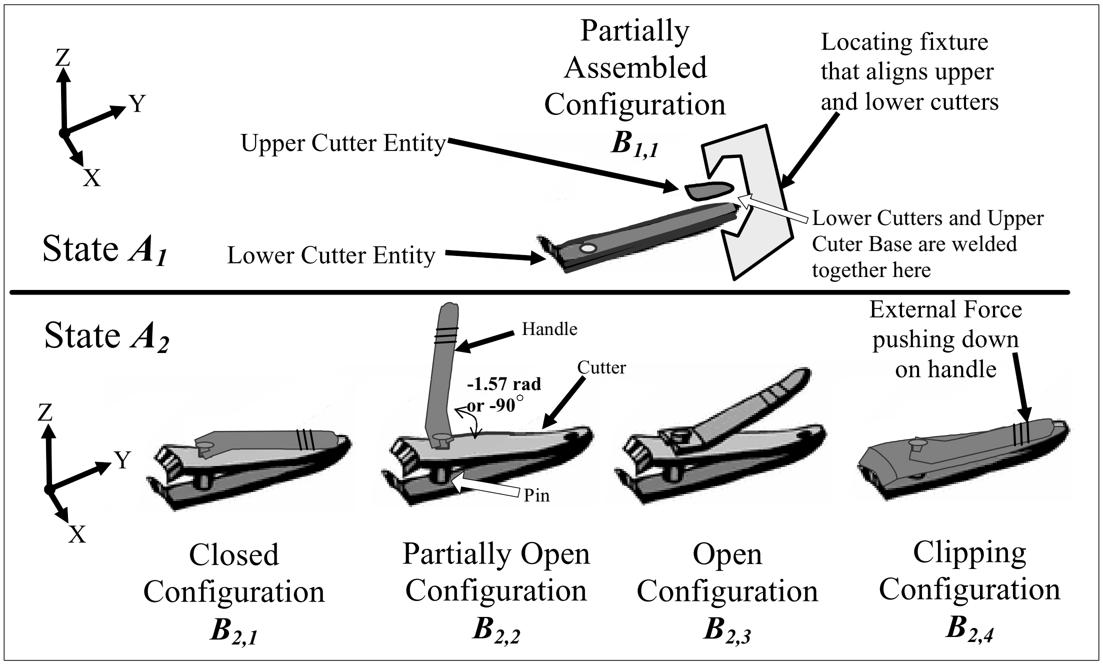

Z
Y
X
Partially Assembled Configuration
$B_{1,1}$
Locating fixture that aligns upper and lower cutters
Upper Cutter Entity
Lower Cutters and Upper Cuter Base are welded together here
State A_1
Lower Cutter Entity
State A_2
Handle
Cutter
-1.57 rad or -90°
External Force pushing down on handle
Z
Y
X
Pin
Closed Configuration
$B_{2,1}$
Partially Open Configuration
$B_{2,2}$
Open Configuration
$B_{2,3}$
Clipping Configuration
$B_{2,4}$

Figure 3.4. Different Configurations and Assembly States of the Fingernail Clipper

All that is known about $\boldsymbol{B_{1,1}}$ right now is that it contains two entities (whose exact natures have not yet been defined) that are placed in a welding fixture. $\boldsymbol{B_{2,1}}$, $\boldsymbol{B_{2,3}}$, $\boldsymbol{B_{2,3}}$, and $\boldsymbol{B_{2,4}}$ correspond to the closed, partially open, open, and clipping configurations of a typical fully-assembled fingernail clipper. To illustrate all the aspects of this methodology, it is only necessary to look at one configuration from each state so $\boldsymbol{B_{1,1}}$ and $\boldsymbol{B_{2,2}}$ have been chosen. The remaining configurations will not be discussed further.

3.1.3 Identifying Parts

An arbitrary configuration $\boldsymbol{B_{i,j}}$ in an assembly contains $\boldsymbol{p_{i,j}}$ rigid parts that are each labeled using a triple subscripted $\boldsymbol{P}$ such that $\boldsymbol{P_{i,j,k}}$ is the $\boldsymbol{k}^{\text{th}}$ part in the $\boldsymbol{j}^{\text{th}}$ configuration in the $\boldsymbol{i}^{\text{th}}$ assembly state. $\boldsymbol{P_{1,1,3}}$ would therefore be the third part in the first configuration of the first assembly state[4]. Taking all factors into consideration for the case of the fingernail clipper, there are actually three parts that require modeling in $\boldsymbol{B_{1,1}}$, and four parts that require modeling in $\boldsymbol{B_{2,2}}$ ($\boldsymbol{p_{1,1}} = 3$, $\boldsymbol{p_{2,2}} = 4$). Figure 3.5 shows schematics of $\boldsymbol{B_{1,1}}$ and $\boldsymbol{B_{2,2}}$ with the parts labeled.

[4] It is important to note that each locating fixture or adjustment process counts as a part and compliant entities must be treated as multiple parts connected by compliant mates.

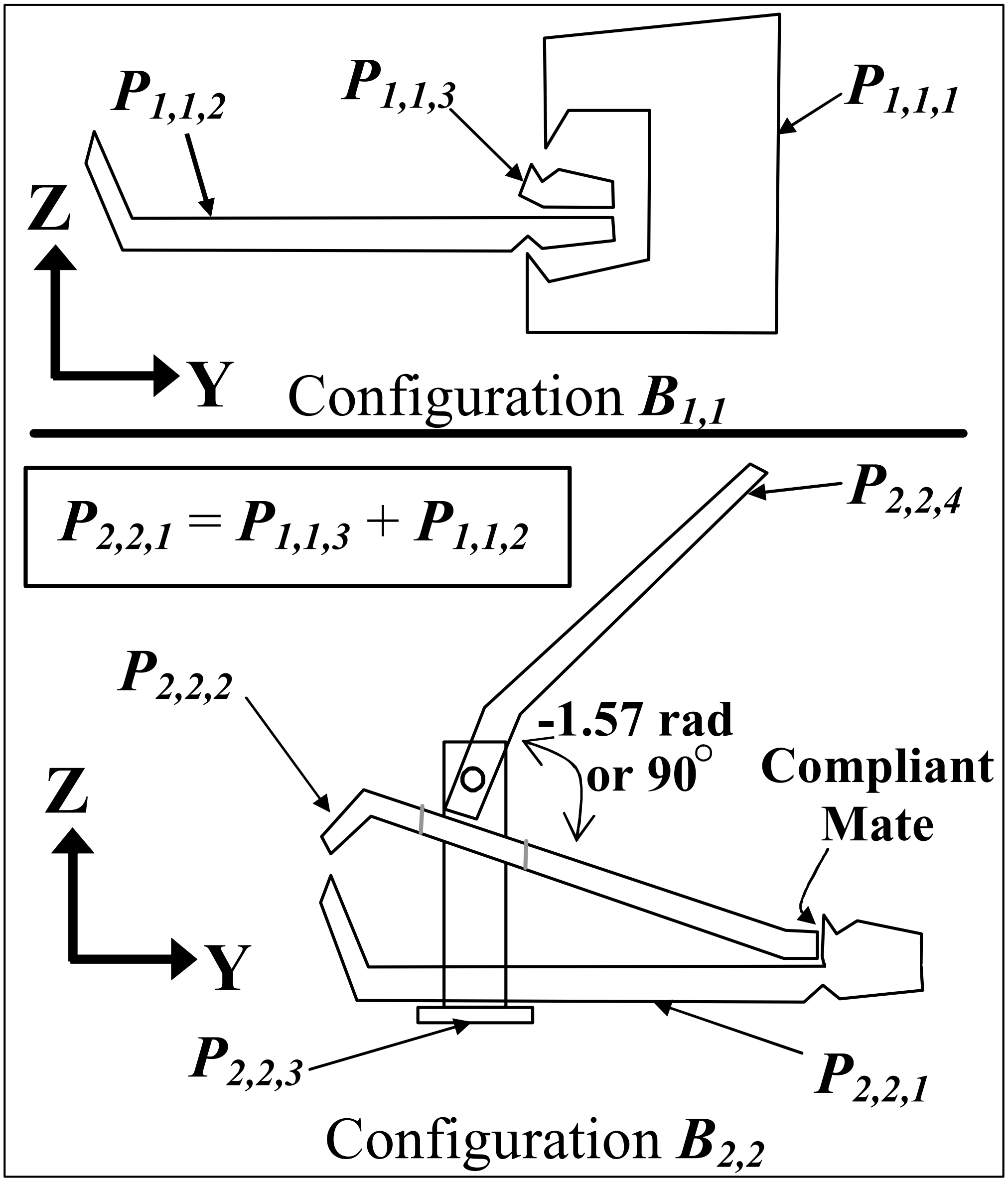


Figure 3.5. Schematics of Configurations $B_{1,1}$ and $B_{2,2}$ of the Fingernail Clipper

$\boldsymbol{B_{1,1}}$ consists of three parts: a welding fixture labeled $\boldsymbol{P_{1,1,1}}$, a lower cutter labeled $\boldsymbol{P_{1,1,2}}$, and an upper cutter base labeled $\boldsymbol{P_{1,1,3}}$. $\boldsymbol{B_{2,2}}$ consists of four parts: a cutter base labeled $\boldsymbol{P_{2,2,1}}$, an upper cutter body labeled $\boldsymbol{P_{2,2,2}}$, a pin labeled $\boldsymbol{P_{2,2,3}}$, and a handle labeled $\boldsymbol{P_{2,2,4}}$. Via the welding operation, the lower cutter ($\boldsymbol{P_{1,1,2}}$) and the upper cutter base ($\boldsymbol{P_{1,1,3}}$) in $\boldsymbol{B_{1,1}}$ become the cutter base ($\boldsymbol{P_{2,2,1}}$) in $\boldsymbol{B_{2,2}}$. The "cutter entity" in $\boldsymbol{B_{2,2}}$ is modeled as two parts, a cutter base ($\boldsymbol{P_{2,2,1}}$) and an upper cutter body ($\boldsymbol{P_{2,2,2}}$) that are joined by a compliant mate. This is necessary to capture the inherent compliance of the "cutter entity" (think of the rebounding nature of a fingernail clipper). For now this is all that will be said about the mates since they will be discussed in detail later.

3.1.4 Identifying Features

An arbitrary configuration $\boldsymbol{B_{i,j}}$ in an assembly contains $\boldsymbol{e_{i,j}}$ features that are each contained in one of the $\boldsymbol{p_{i,j}}$ parts. Features are each labeled using a triple subscripted $\boldsymbol{E}$ such that $\boldsymbol{E_{i,j,k}}$ is the $\boldsymbol{k}^{\text{th}}$ feature in the $\boldsymbol{j}^{\text{th}}$ configuration in the $\boldsymbol{i}^{\text{th}}$ assembly state. $\boldsymbol{E_{2,3,2}}$ would therefore be the second feature in the third configuration of the second assembly state. A feature is defined because it either mates with a feature on another part or is used in evaluating the manufacturability of the assembly. Each feature must also be assigned a coordinate frame. Analysis of fingernail clipper reveals that $\boldsymbol{B_{1,1}}$ has six features and $\boldsymbol{B_{2,2}}$ has eleven features ($\boldsymbol{e_{1,1}}$ = 6 and $\boldsymbol{e_{2,2}}$ = 11). Table 3.1 is a table providing a description of the seventeen features for the fingernail clipper. Schematics of the individual parts in $\boldsymbol{B_{1,1}}$ and $\boldsymbol{B_{2,2}}$ are shown respectively in Figure 3.6 and Figure 3.7 with their features identified and coordinate frames assigned.

Feature	Description
$E_{1,1,1}$	lower lip on Fixture (mates to Lower Cutter) (base feature for $B_{1,1}$)
$E_{1,1,2}$	upper lip on Fixture (mates to Upper Cutter Base)
$E_{1,1,3}$	notch on Lower Cutter (mates to Fixture)
$E_{1,1,4}$	top surface on Lower Cutter (mates to Upper Cutter Base)
$E_{1,1,5}$	bottom surface on Upper Cutter Base (mates to Lower Cutter)
$E_{1,1,6}$	notch on Upper Cutter Base (mates to Fixture)
$E_{2,2,1}$	notch on Cutter Base (base feature for $B_{2,2}$)
$E_{2,2,2}$	top lip on Cutter Base (mates compliantly to Upper Cutter Body)
$E_{2,2,3}$	lower blade on Cutter Base
$E_{2,2,4}$	hole on Cutter Base (mate to Pin)
$E_{2,2,5}$	head of Pin (mates to Cutter Base)
$E_{2,2,6}$	slot in Pin (mates to Handle)
$E_{2,2,7}$	rod of handle (mates to Pin)
$E_{2,2,8}$	fulcrum tip of Handle (mates to Upper Cutter Body)
$E_{2,2,9}$	top surface of Upper Cutter Body (mates to Handle)
$E_{2,2,10}$	back end of Upper Cutter body (mates compilantly to Cutter Base)
$E_{2,2,11}$	upper blade on Upper Cutter Body

Table 3.1. Description of Features in the Fingernail Clipper

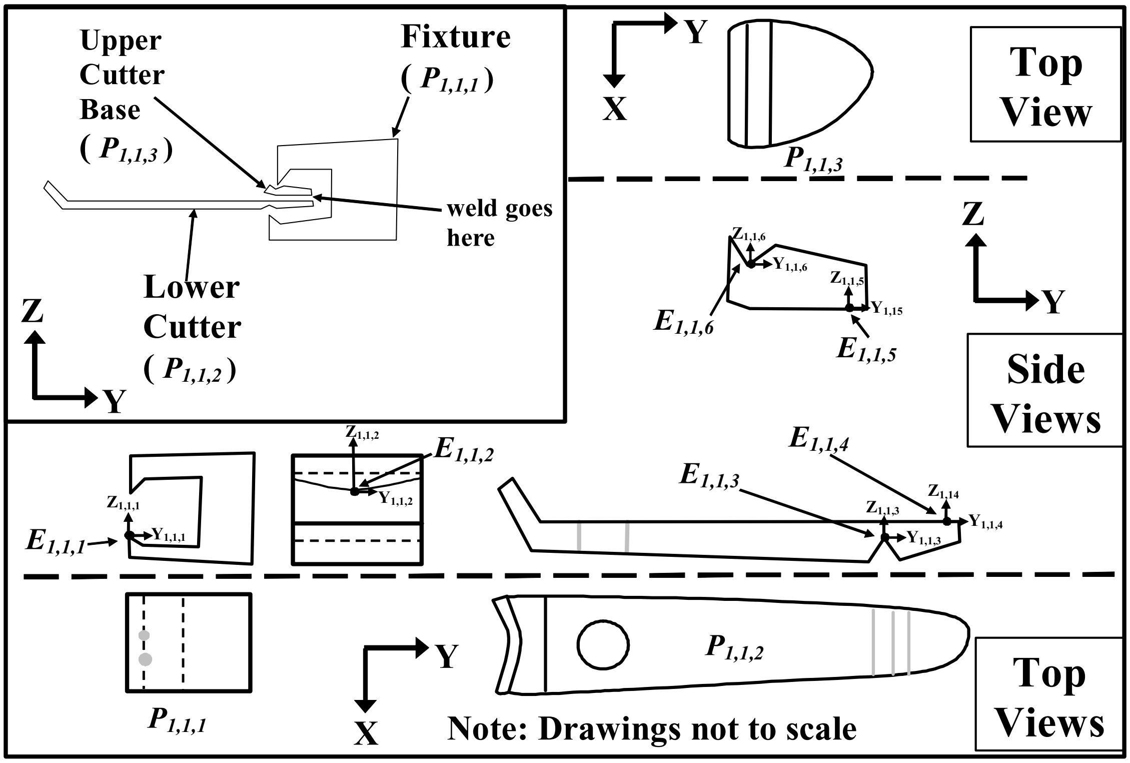


Figure 3.6. Features for Configuration $B_{1,1}$ of the Fingernail Clipper

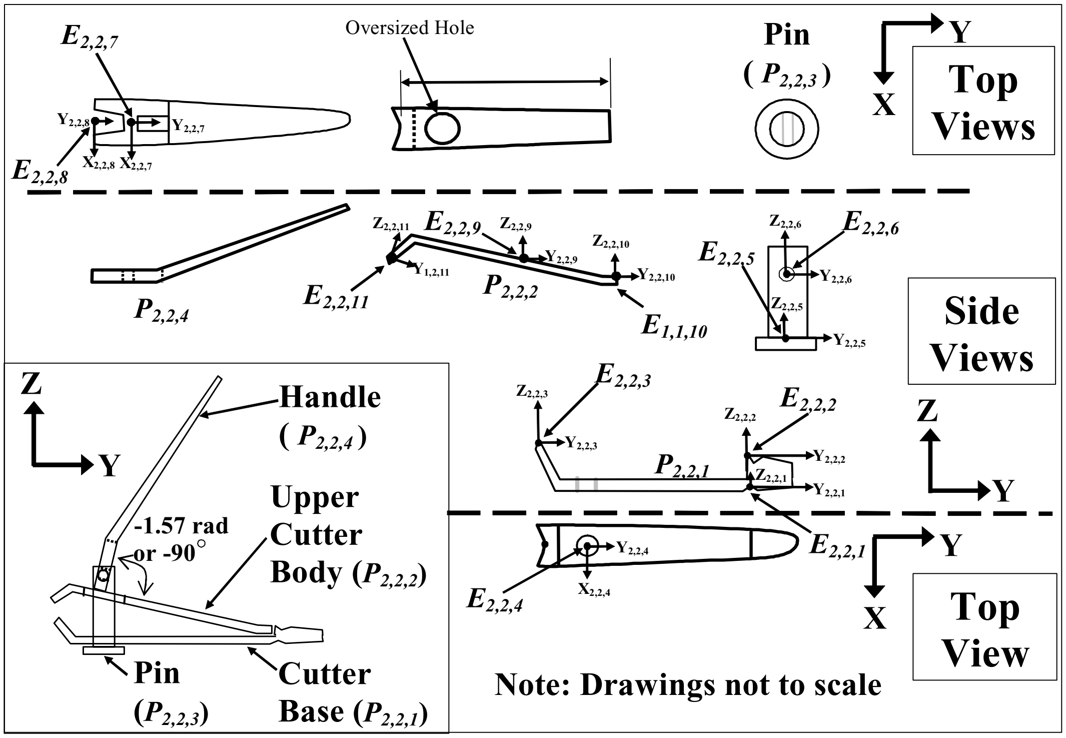


Figure 3.7. Features for Configuration $B_{2,2}$ of the Fingernail Clipper

3.1.5 Identifying KC Feature Pairs

An arbitrary configuration $\boldsymbol{B_{i,j}}$ in an assembly contains $\boldsymbol{h_{i,j}}$ KC feature pairs that relate features on different parts and geometrically represent the expectations of the customer. KC feature pairs are labeled using a triple subscripted $\boldsymbol{H}$ such that $\boldsymbol{H_{i,j,k}}$ is the $\boldsymbol{k}^{\text{th}}$ KC feature pair in the $\boldsymbol{j}^{\text{th}}$ configuration in the $\boldsymbol{i}^{\text{th}}$ assembly state. $\boldsymbol{H_{4,3,1}}$ would therefore be the first KC feature pair in the third configuration of the fourth assembly state. Analysis of fingernail clipper reveals that $\boldsymbol{B_{1,1}}$ has one KC feature pair labeled $\boldsymbol{H_{1,1,1}}$ and $\boldsymbol{B_{2,2}}$ has one KC feature pair labeled $\boldsymbol{H_{2,2,1}}$ ($\boldsymbol{h_{1,1}}$ = 1 and $\boldsymbol{h_{2,2}}$ = 1). $\boldsymbol{H_{1,1,1}}$ relates $\boldsymbol{E_{1,1,6}}$, the notch in the upper cutter base to $\boldsymbol{E_{1,1,3}}$, the notch in the lower cutter. $\boldsymbol{H_{2,2,1}}$ relates $\boldsymbol{E_{1,1,11}}$, the upper blade in the upper cutter body to $\boldsymbol{E_{2,2,3}}$, the lower notch in the cutter base. These relationships are shown in Figure 3.9.

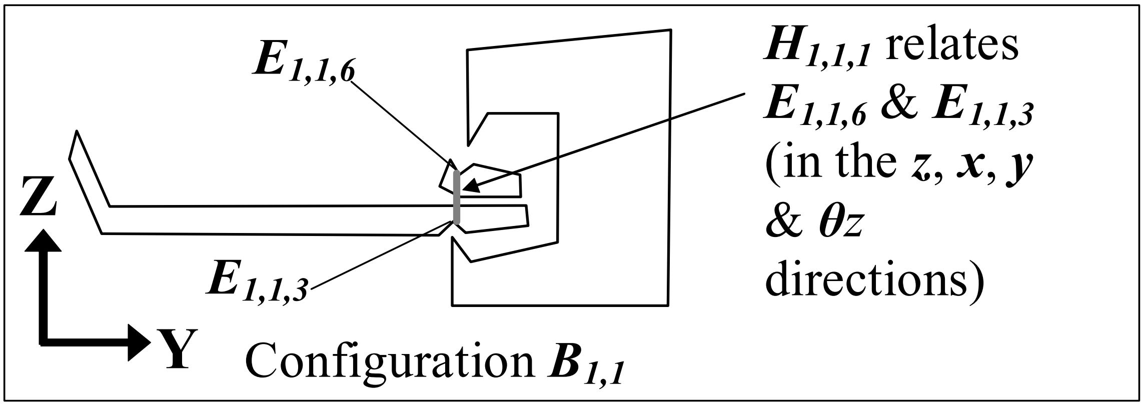


Figure 3.8. KC's For Configurations $B_{1,1}$ of the Fingernail Clipper

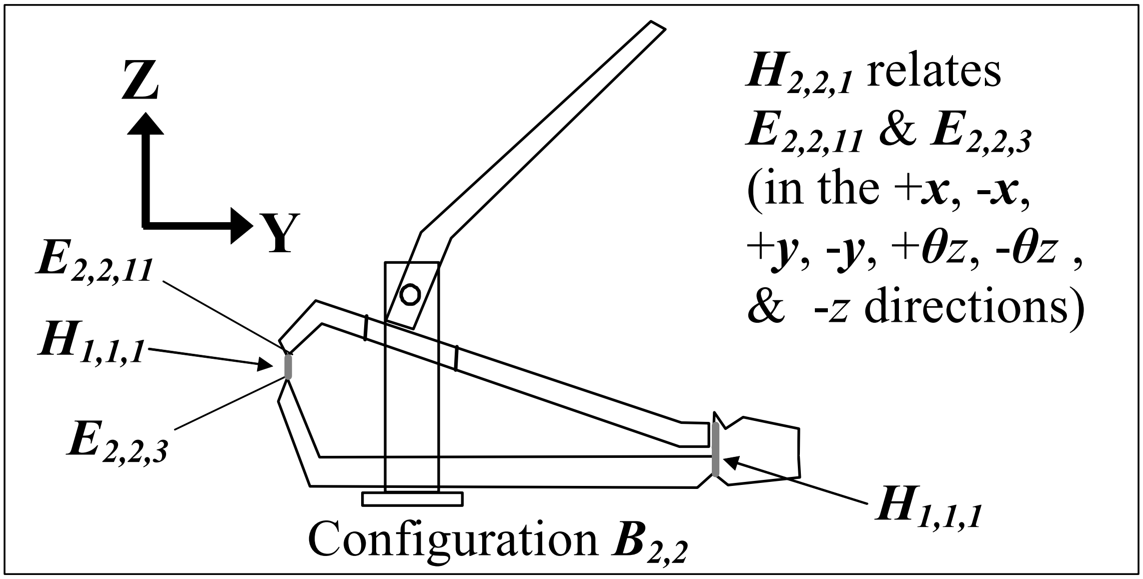


Figure 3.9. KC's For Configurations $B_{1,1}$ & $B_{2,2}$ of the Fingernail Clipper

An arbitrary KC feature pair, $H_{i,j,k}$ is composed of $kc_{i,j,k}$ KCs, where $kc_{i,j,k}$ is any number between one and twelve. Each KC can be a linking KC or a performance KC and each performance KC can be either an alignment KC a clearance KC. For an arbitrary configuration of an assembly, $B_{i,j}$, the total number of KCs in that configuration, $kct_{i,j}$, is computed as follows:

$$kct_{i,j} = \sum_{k=1}^{h_{i,j}} kc_{i,j,k}$$

$$for$$

$$i = 1,2,...,a$$

$$j = 1,2,...,b_i$$

Equation 3.2

The set of all nominal values for the KCs of an arbitrary configuration, $\boldsymbol{B_{i,j}}$, can be contained in a **(*kct*$_{i,j}$** x ***1*)** vector, **[*KCn*]$_{i,j}$** called the configuration-level nominal KC value matrix. The set of all KC directions (***θx*** = 1, ***θy*** = 2 , ***θz*** = 3, ***x*** = 4, ***y*** = 5, and ***z*** = 6 expressed relative to the coordinate frame local to the first feature in the KC feature pair) can be contained in a **(*kct*$_{i,j}$** x ***1*)** vector **[*KCd*]$_{i,j}$** called the configuration-level KC direction matrix. **[*KCn*$_k$]$_{i,j}$** is therefore the nominal value of the ***k***th KC of the ***j***th configuration in the ***i***th assembly state. Also, if **[*KCd*$_3$]$_{2,2}$** equals six, then the third KC in the second configuration of the second assembly state is a distance along the ***Z***-axis of a coordinate frame whose location is local to first feature in the corresponding KC feature pair.

The total number of KCs in an assembly is denoted ***kctt***. Since a KC can be present in more than one configuration, the sum of ***kct*$_{i,j}$** for all configurations does not in general yield ***kctt***. Explicitly this can be expressed as follows:

$$kctt \neq \sum_{i=1}^{a}\sum_{j=1}^{b_i} kct_{i,j}$$

Equation 3.3

The KCs for an entire assembly are labeled $\boldsymbol{KC_1}$, $\boldsymbol{KC_2}$,…, $\boldsymbol{KC_{kctt}}$. $\boldsymbol{KC_5}$ is therefore the fifth KC of an assembly and as stated above can be included in more than one configuration. The KCs in an entire assembly can be divided into two groups, ***kcl*** linking KCs and ***kcp*** performance KCs where:

$$kctt = kcl + kcp$$

Equation 3.4

For an arbitrary assembly, the set of all performance KC types (***alignment*** = 1 and ***clearance*** = 2) can be contained in a **(*kcp*** x ***1*)** vector **[*KCt*]** called the KC type matrix. The set of all performance KC directions (***θx*** = 1, ***θy*** = 2 , ***θz*** = 3, ***x*** = 4, ***y*** = 5, and ***z*** = 6 expressed relative to the coordinate frame local to the first feature in the KC feature pair) can be contained in a **(*kcp*** x ***1*)** vector **[*KCd*]** called the performance KC direction matrix. The set of all performance KC nominal values can be contained in a **(*kcp*** x ***1*)** vector **[*KCn*]** (angles should be expressed in radians not degrees) called the performance KC nominal value matrix. **[$\boldsymbol{KCn_i}$]** is therefore the nominal value of the ***i***th performance KC in the assembly. This means that if **[$\boldsymbol{KCd_8}$]** equals four, then the eighth performance KC in the assembly is a distance along the ***X***-axis of a coordinate frame whose location is local to first feature in the corresponding KC feature pair.

The set of all specification limits offsets for the performance KCs can be contained in a **(*kcp* x *1*)** vector, **[*KCl*]**. The set of all specification limits for the performance KCs can be contained in a **(*kcp* x *1*)** vector, **[*KCL*]** (note that linking KCs do not require specification limits) called the KC specification limit matrix where the ***i***th entry **[*KCL*$_i$]** is computed as follows:

$$[KCL_i] = \begin{cases} (\,[KCl_i] + [KCn_i]\,)\,if\ [KCt_i] = 1 \\ [KCl_i]\,if\ [KCt_i] = 2 \end{cases}$$

$$for$$

$$i = 1,2,...,kcp$$

Equation 3.5

Analysis of fingernail clipper reveals that $\boldsymbol{B_{1,1}}$ has four KCs and $\boldsymbol{B_{2,2}}$ has seven ($\boldsymbol{kct_{1,1}}$ = 4, $\boldsymbol{kct_{2,2}}$ = 7, and ***kctt*** = 11). Of the eleven KCs, four are linking KCs and seven are performance KCs (***kcl*** = 4 and ***kcp*** = 7). These eleven KCs are described in Table 3.2. $\boldsymbol{KC_1}$, $\boldsymbol{KC_2}$, $\boldsymbol{KC_3}$, and $\boldsymbol{KC_4}$ in $\boldsymbol{B_{1,1}}$ are linking KC's that collectively describe the alignment of the notch in the lower cutter relative to the notch in the upper cutter base in the ***z***, ***x***, ***y*** and ***θz*** directions after the welding operation as set by the fixture. These linking KCs will become dimensions in $\boldsymbol{B_{2,2}}$ (this will be discussed in more detail below). $\boldsymbol{KC_5}$, $\boldsymbol{KC_6}$, $\boldsymbol{KC_7}$, $\boldsymbol{KC_8}$, $\boldsymbol{KC_9}$, and $\boldsymbol{KC_{10}}$ are alignment KCs that describe the alignment in the +***x***, -***x***, +***y***, -***y***, +***θz*** and -***θz*** directions respectively between the upper blade on the upper cutter body and the lower blade on the cutter base in $\boldsymbol{B_{2,2}}$ (the “+” and “-“ signs indicate whether the performance KC has an upper or lower specification limit). $\boldsymbol{KC_{11}}$ is a clearance KC that describes the gap (in the -***z*** direction) between the upper and lower blades and has a lower specification limit of zero (as long as the two blades don't touch during the opening of the clippers, this $\boldsymbol{KC_{11}}$ is delivered).

KC Feature Pair	Two Features it Relates	Number and Type of KCs relating each KC feature pair	KCs each KC feature pair contains
$\boldsymbol{H_{1,1,1}}$	$\boldsymbol{E_{1,1,6}}$ **&** $\boldsymbol{E_{1,1,3}}$	4 Linking KCs, 0 Performance KCs	***Linking:*** $\boldsymbol{KC_1}$ (+z)**,** $\boldsymbol{KC_2}$ (+x)**,** $\boldsymbol{KC_3}$ (+y)**,** $\boldsymbol{KC_4}$ (+θz)
$\boldsymbol{H_{2,2,1}}$	$\boldsymbol{E_{2,2,11}}$ **&** $\boldsymbol{E_{2,2,3}}$	0 Linking KCs, 7 Performance KCs: 6 Alignment KCs & 1 Clearance KC	***Alignment:*** $\boldsymbol{KC_5}$ (+x)**,** $\boldsymbol{KC_6}$ (-x)**,** $\boldsymbol{KC_7}$ (+y)**,** $\boldsymbol{KC_8}$ (-y)**,** $\boldsymbol{KC_9}$ (+θz), $\boldsymbol{KC_{10}}$ (-θz) ***Clearance:*** $\boldsymbol{KC_{11}}$ (-z)

Table 3.2. List of KCs Derived from Each KC feature pair

Figure 3.10 shows **[*KCn*]**$_{1,1}$, **[*KCn*]**$_{2,2}$, **[*KCd*]**$_{1,1}$, **[*KCd*]**$_{2,2}$, **[*KCt*]**, **[*KCn*]**, **[*KCd*]**, **[*KCl*]** and **[*KCL*]** for the fingernail clipper example. While for this assembly, **[*KCn*]** equals **[*KCn*]**$_{2,2}$, this is not the case in general. The nominal KC values in **[*KCn*]**$_{1,1}$, **[*KCn*]**$_{2,2}$, **[*KCn*]** and the specification limits of the alignment KCs in **[*KCL*]** will remain unknown until the parts are assembled analytically and the values in **[*KCn*]**$_{2,2}$ are obtained. Later in this chapter, it will be shown how to assemble the parts analytically and solve for the unknown KC-related values.

$$[KCn]_{1,1}{}^{T} = \begin{array}{cccc} KC_1 & KC_2 & KC_3 & KC_4 \\ \left[[KCn_1]_{1,1} \right. & [KCn_2]_{1,1} & [KCn_3]_{1,1} & \left. [KCn_4]_{1,1} \right] \end{array}$$

$$[KCn]_{2,2}{}^{T} = \begin{array}{ccccccc} KC_5 & KC_6 & KC_7 & KC_8 & KC_9 & KC_{10} & KC_{11} \\ \left[[KCn_1]_{2,2} \right. & [KCn_2]_{2,2} & [KCn_3]_{2,2} & [KCn_4]_{2,2} & [KCn_5]_{2,2} & [KCn_6]_{2,2} & \left. [KCn_7]_{2,2} \right] \end{array}$$

$$[KCd]_{1,1}{}^{T} = \begin{array}{cccc} KC_1 & KC_2 & KC_3 & KC_4 \\ \left[6 \right. & 4 & 5 & \left. 3 \right] \end{array}$$

$$[KCd]_{2,2}{}^{T} = \begin{array}{ccccccc} KC_5 & KC_6 & KC_7 & KC_8 & KC_9 & KC_{10} & KC_{11} \\ \left[4 \right. & 4 & 5 & 5 & 3 & 3 & \left. 6 \right] \end{array}$$

Note: While for this assembly, the nominal values and directions for the performance KC's equal the nominal values and directions for KC's in configuration ***B***$\boldsymbol{B_{2,2}}$ this is not generally the case

$$[KCL] = \begin{array}{c|c} \left[[KCn_1]_{2,2} + 0.20 \right. & KC_5 \\ {} [KCn_2]_{2,2} - 0.20 & KC_6 \\ {} [KCn_3]_{2,2} + 0.15 & KC_7 \\ {} [KCn_4]_{2,2} - 0.15 & KC_8 \\ {} [KCn_5]_{2,2} + .02 & KC_9 \\ {} [KCn_6]_{2,2} - .02 & KC_{10} \\ \left. 0 \right] & KC_{11} \end{array}$$

$$[KCt]^{T} = \begin{array}{ccccccc} KC_5 & KC_6 & KC_7 & KC_8 & KC_9 & KC_{10} & KC_{11} \\ \left[1 \right. & 1 & 1 & 1 & 1 & 1 & \left. 2 \right] \end{array}$$

$$[KCl]^{T} = \begin{array}{ccccccc} KC_5 & KC_6 & KC_7 & KC_8 & KC_9 & KC_{10} & KC_{11} \\ \left[.20 \right. & -.20 & .15 & -.15 & .02 & -.02 & \left. 0 \right] \end{array}$$

$$[KCn] = [KCn]_{2,2} \qquad [KCd] = [KCd]_{2,2}$$

Figure 3.10. The KC Nominal Values, Directions, Type, & Specification Limits

3.1.6 Identifying Intra-Part Relationships

An arbitrary configuration $\boldsymbol{B_{i,j}}$ in an assembly contains $\boldsymbol{r_{i,j}}$ intra-part relationships connecting features within parts. Intra-part relationships are each labeled using a triple subscripted ***R*** such that $\boldsymbol{R_{i,j,k}}$ is the ***k***th intra-part relationship in the ***j***th configuration in the ***i***th assembly state. $\boldsymbol{R_{2,3,1}}$ would therefore be the first intra-part relationships in the third configuration of the second assembly state. Analysis of fingernail clipper reveals that $\boldsymbol{B_{1,1}}$ has three intra-part relationships and $\boldsymbol{B_{2,2}}$ has seven intra-part relationships ($\boldsymbol{r_{1,1}}$ = 3 and $\boldsymbol{r_{2,2}}$ = 7). Table 3.3 lists the intra-part relationships, the features they join, and a brief description of each.

Intra-Part Relationship	Features It Joins	Description (directions of dimensions it contains)
$R_{1,1,1}$	$E_{1,1,1}$ & $E_{1,1,2}$	offset between upper and lower lips on fixture (x,y,θz)
$R_{1,1,2}$	$E_{1,1,3}$ & $E_{1,1,4}$	thickness of Lower Cutter (z)
$R_{1,1,3}$	$E_{1,1,5}$ & $E_{1,1,6}$	thickness of Upper Cutter Base (z)
$R_{2,2,1}$	$E_{2,2,1}$ & $E_{2,2,2}$	postion of top lip on Cutter Base (x,y,z,θz)
$R_{2,2,2}$	$E_{2,2,1}$ & $E_{2,2,3}$	position of lower blade on Cutter Base (x,y,θz)
$R_{2,2,3}$	$E_{2,2,9}$ & $E_{2,2,10}$	angle of bend in Upper Cutter Body (θx)
$R_{2,2,4}$	$E_{2,2,9}$ & $E_{2,2,11}$	position of upper blade on Upper Cutter Body (x,y,θz)
$R_{2,2,5}$	$E_{2,2,1}$ & $E_{2,2,5}$	distance to hole in Cutter Base (y)
$R_{2,2,6}$	$E_{2,2,6}$ & $E_{2,2,7}$	distance from head to slot on Pin (z)
$R_{2,2,7}$	$E_{2,2,8}$ & $E_{2,2,9}$	distance from rod to fulcrum tip on Handle (y)

Table 3.3. Description of Intra-Part Relationships in Fingernail Clipper

An arbitrary intra-part relationship, $\boldsymbol{R_{i,j,k}}$, is composed of a chain of $\boldsymbol{d_{i,j,k}}$ serially-linked dimensions, where $\boldsymbol{d_{i,j,k}}$ is any number between one and six (in assemblies with multiple assembly states, $\boldsymbol{R_{i,j,k}}$ can also be contain linking KCs from a previous assembly and these are also counted as dimensions when determining $\boldsymbol{d_{i,j,k}}$). For an arbitrary configuration, $\boldsymbol{B_{i,j}}$, of an assembly, the total number of dimensions in that configuration, $\boldsymbol{dt_{i,j}}$, is computed as follows:

$$dt_{i,j} = \sum_{k=1}^{r_{i,j}} d_{i,j,k_i}$$

$$\begin{aligned}&\mathit{for}\\ &i = 1,2,\ldots,a\\ &j = 1,2,\ldots,b_i\end{aligned}$$

Equation 3.6

The set of all nominal values for the dimensions of an arbitrary configuration, $\boldsymbol{B_{i,j}}$, can be contained in a ($\boldsymbol{dt_{i,j}}$ x ***1***) vector, **[*Dn*]**$_{\boldsymbol{i,j}}$ (angles should be expressed in radians not degrees) called the configuration-level dimension nominal value matrix. The set of all dimension directions ($\boldsymbol{\theta x}$ = 1, $\boldsymbol{\theta y}$ = 2 , $\boldsymbol{\theta z}$ = 3, ***x*** = 4, ***y*** = 5 , and ***z*** = 6) can be contained in a **(**$\boldsymbol{dt_{i,j}}$ x ***1*****)** vector **[*Dd*]**$_{\boldsymbol{i,j}}$ called the configuration level dimension direction matrix. Entry **[*Dn*$_{\boldsymbol{k}}$]**$_{\boldsymbol{i,j}}$ is therefore the ***k***th dimension of the ***j***th configuration in the ***i***th assembly state. For example, if **[*Dd*$_{\boldsymbol{3}}$]**$_{\boldsymbol{2,2}}$ equals one, then the third dimension in the second configuration of the second assembly state is an angle of rotation about the ***X***-axis in the ***Y-Z*** plane of the coordinate frame local to that dimension.

Since a single dimension may be included in multiple configurations, another counting variable, ***dtt***, is defined as the total number of <u>unique</u> dimensions (now excluding linking KCs) for an entire assembly and in general:

$$dtt \neq \sum_{i=1}^{a}\sum_{j=1}^{b_i} dt_{i,j}$$

Equation 3.7

Dimensions for an entire assembly are labeled $\boldsymbol{D_1}$, $\boldsymbol{D_2}$,…, $\boldsymbol{D_{dtt}}$. For example, $\boldsymbol{D_7}$ would be the seventh dimension of an assembly and it can be included in one or more configurations. For an entire assembly, the set of all nominal values for the dimensions in that assembly can be contained in a **(*dtt*** x ***1*****)** vector **[*Dn*]** called the nominal dimension matrix. The set of all dimension directions (***θx*** = 1, ***θy*** = 2 , ***θz*** = 3, ***x*** = 4, ***y*** = 5 , and ***z*** = 6) can be contained in a **(*dtt*** x ***1*)** vector **[*Dd*],** called the dimension direction matrix. The set of all upper and lower bounds on those dimensions (due to packaging constraints or fabrication limitations) can be contained respectively in two **(*dtt*** x ***1*)** vectors **[*Dul*]** and **[*Dll*],** called respectively the upper dimension bound and lower dimension bound matrices. **[Dn_i]** is therefore the nominal value of the ***i***th dimension in the assembly. If **[Dd_4]** equals two, then the fourth dimension in the assembly is an angle of rotation about the ***Y***-axis in the ***X-Z*** plane of the coordinate frame local to that dimension.

An arbitrary dimension, D_i, can be assigned a PDF, $f_{Di}(\delta_i)$, and a CDF, $F_{Di}(\delta_i)$ that are characterized by specifying distribution type and the corresponding distribution parameters (depending on the distribution anywhere from one to three parameters are needed). Each PDF and CDF also has an inverse (which may not be defined for all real numbers), denoted generically $f^{-1}_{Di}(\delta_i)$ and $F^{-1}_{Di}(\delta_i)$ respectively. These functions take probability densities or probabilities as inputs and output the corresponding values that have those probability densities or probabilities. The set of all distribution types can be contained in a **(*dtt*** **x** ***1*)** vector of character strings, **[*Dt*]** called the dimension distribution type matrix. Each row in **[*Dt*]** is a string of letters that equals the name of a distribution (e.g. 'Normal', 'Uniform', 'Poisson', etc), the set of all distribution parameters for these distributions can be contained in a **(*dtt*** **x** ***3*)** matrix, **[*Dp*]** called the dimension distribution parameter matrix, where each row contains the distribution parameters for the distribution of the corresponding dimension (if a distribution has less than three parameters, then the extra entries should be entered as “NaN” meaning “Not a Number”).

Analysis of the fingernail clipper reveals that $\boldsymbol{B_{1,1}}$ has five dimensions and $\boldsymbol{B_{2,2}}$ has fourteen ($\boldsymbol{dt_{1,1}}$ = 5, $\boldsymbol{dt_{2,2}}$ = 14). Of the fourteen dimensions in $\boldsymbol{B_{2,2}}$, four are linking KCs from $\boldsymbol{B_{1,1}}$ (therefore $\boldsymbol{dtt}$ = 5 + 14 - 4 = 15 instead of 19 as one might expect). Table 3.4 shows the chain of serially-linked dimensions that comprises each intra-part relationship for configurations $\boldsymbol{B_{1,1}}$ and $\boldsymbol{B_{2,2}}$ of the fingernail clipper. For each intra-part relationship the chain starts at the first feature listed in the second column of Table 3.3 and ends at the second feature listed in second column of Table 3.3. Figure 3.11, Figure 3.12, and Figure 3.13 show schematically how the dimensions for the fingernail clipper relate to the features and intra-part relationships. Figure 3.14 shows $\boldsymbol{[Dn]_{1,1}}$, $\boldsymbol{[Dn]_{2,2}}$, $\boldsymbol{[Dd]_{1,1}}$, $\boldsymbol{[Dd]_{2,2}}$, $\boldsymbol{[Dn]}$, $\boldsymbol{[Dd]}$, $\boldsymbol{[Dul]}$, and $\boldsymbol{[Dll]}$ for the fingernail clipper. Figure 3.15 shows $\boldsymbol{[Dt]}$ and $\boldsymbol{[Dp]}$ that contain the distribution types and characteristic parameters respectively for the PDFs and CDFs representing each dimension of the fingernail clipper (while this example only uses distributions that are either Uniform or Normal, this methodology is not limited to such distributions).

Intra-Part Relationship	Features It Joins	Chain of Dimensions or Linking KC's Comprising Each Intra-Part Relationship (in order from 1st feature to 2nd feature)
$R_{1,1,1}$	$E_{1,1,1}$ & $E_{1,1,2}$	D_1, D_2, D_3
$R_{1,1,2}$	$E_{1,1,3}$ & $E_{1,1,4}$	D_4
$R_{1,1,3}$	$E_{1,1,5}$ & $E_{1,1,6}$	D_5
$R_{2,2,1}$	$E_{2,2,1}$ & $E_{2,2,2}$	KC_1, KC_2, KC_3, KC_4
$R_{2,2,2}$	$E_{2,2,1}$ & $E_{2,2,3}$	D_6, D_7, D_8
$R_{2,2,3}$	$E_{2,2,10}$ & $E_{2,2,9}$	D_9
$R_{2,2,4}$	$E_{2,2,9}$ & $E_{2,2,11}$	D_{10}, D_{11}, D_{12}
$R_{2,2,5}$	$E_{2,2,1}$ & $E_{2,2,5}$	D_{13}
$R_{2,2,6}$	$E_{2,2,6}$ & $E_{2,2,7}$	D_{14}
$R_{2,2,7}$	$E_{2,2,8}$ & $E_{2,2,9}$	D_{15}

Table 3.4. Chain of Dimensions or KCs Comprising Each Intra-Part Relationship

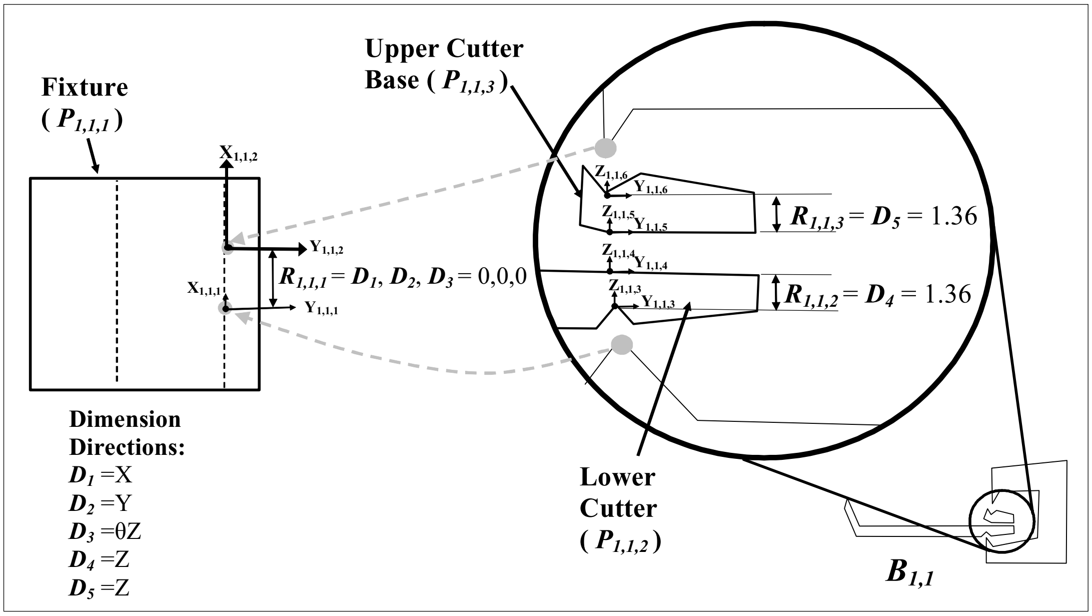


Figure 3.11. Dimensions for Configuration $B_{1,1}$ of the Fingernail Clipper

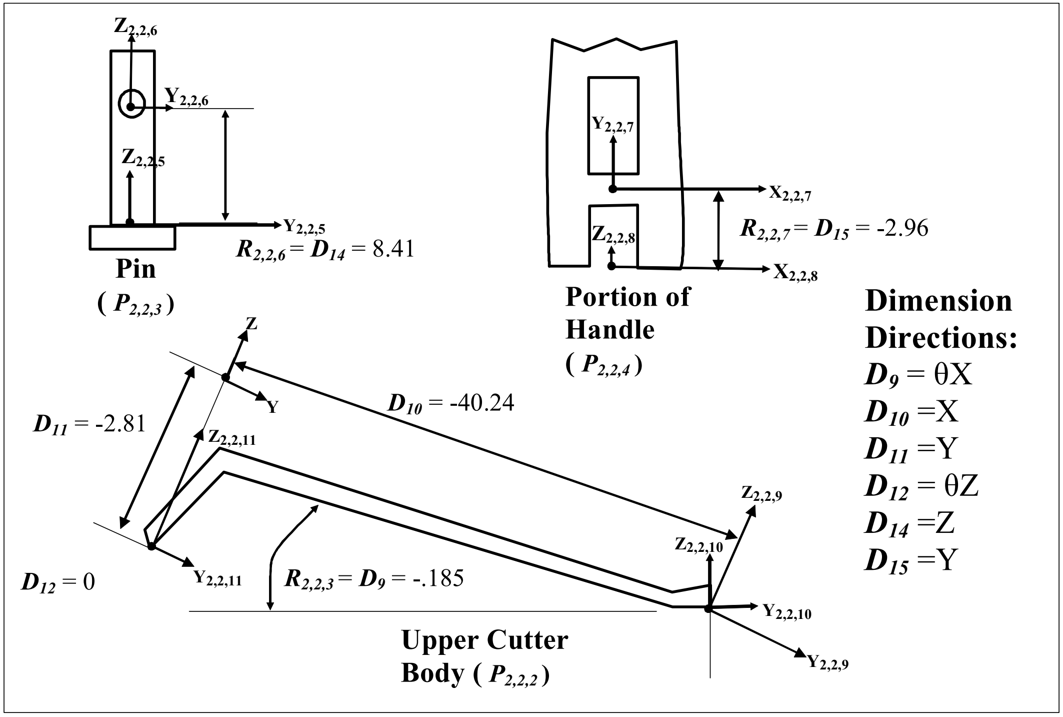


Figure 3.12. Dimensions for Some Parts in Configuration $B_{2,2}$

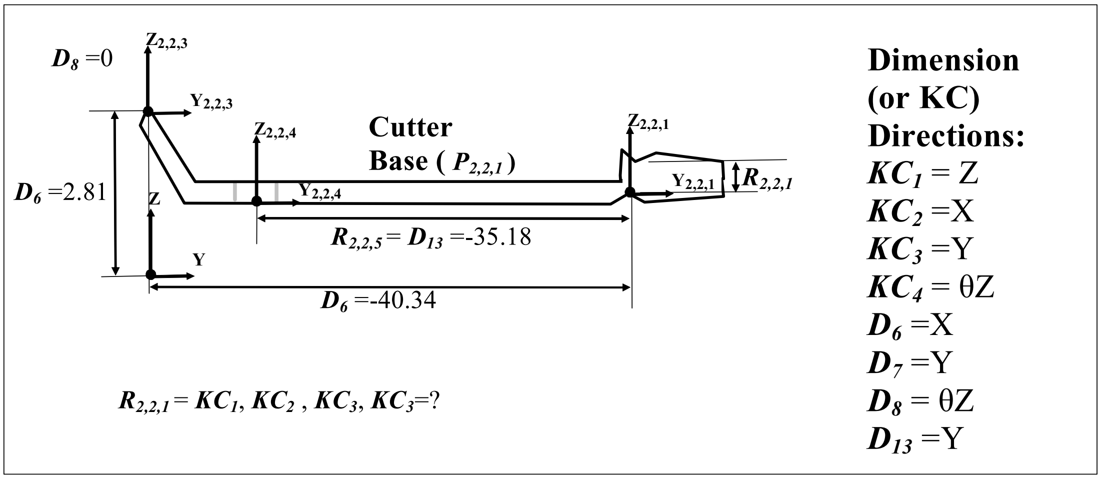


Figure 3.13. Dimensions for Cutter Base in Configuration $B_{2,2}$

$$[Dn]^T = \begin{array}{ccccccccccccccc} D_1 & D_2 & D_3 & D_4 & D_5 & D_6 & D_7 & D_8 & D_9 & D_{10} & D_{11} & D_{12} & D_{13} & D_{14} & D_{15} \\ \hline 0 & 0 & 0 & 1.36 & 1.36 & -40.34 & 2.81 & 0 & -0.185 & -40.24 & -2.81 & 0 & -35.18 & 8.41 & -2.96 \end{array}$$

$$[Dd]^T = \begin{array}{ccccccccccccccc} D_1 & D_2 & D_3 & D_4 & D_5 & D_6 & D_7 & D_8 & D_9 & D_{10} & D_{11} & D_{12} & D_{13} & D_{14} & D_{15} \\ \hline 4 & 5 & 3 & 6 & 6 & 5 & 6 & 3 & 1 & 5 & 6 & 3 & 5 & 6 & 5 \end{array}$$

$$[Dul]^T = \begin{array}{ccccccccccccccc} D_1 & D_2 & D_3 & D_4 & D_5 & D_6 & D_7 & D_8 & D_9 & D_{10} & D_{11} & D_{12} & D_{13} & D_{14} & D_{15} \\ \hline .05 & .05 & .001 & 2.5 & 2.5 & -40 & 3.5 & .1 & -.16 & -40 & -2.5 & .1 & -27 & 10 & -2 \end{array}$$

$$[Dll]^T = \begin{array}{ccccccccccccccc} D_1 & D_2 & D_3 & D_4 & D_5 & D_6 & D_7 & D_8 & D_9 & D_{10} & D_{11} & D_{12} & D_{13} & D_{14} & D_{15} \\ \hline -.05 & -.05 & -.001 & .5 & .5 & -50 & 2.5 & -.1 & -.2 & -50 & -3.5 & -.1 & -37 & 6 & -4 \end{array}$$

$$[Dn]_{1,1} = [Dn_{1:4}] \qquad [Dd]_{1,1} = [Dd_{1:4}]$$

$$[Dn]_{2,2}{}^T = \begin{array}{ccccccccccccccc} KC_1 & KC_2 & KC_3 & KC_4 & D_6 & D_7 & D_8 & D_9 & D_{10} & D_{11} & D_{12} & D_{13} & D_{14} & D_{15} \\ \hline [KCn_1]_{1,1} & [KCn_2]_{1,1} & [KCn_3]_{1,1} & [KCn_4]_{1,1} & -40.34 & 0 & 0 & -0.185 & -40.24 & 0 & 0 & -35.18 & 8.41 & -2.96 \end{array}$$

$$[Dd]_{2,2}{}^T = \begin{array}{ccccccccccccccc} KC_1 & KC_2 & KC_3 & KC_4 & D_6 & D_7 & D_8 & D_9 & D_{10} & D_{11} & D_{12} & D_{13} & D_{14} & D_{15} \\ \hline [KCd_1] & [KCd_2] & [KCd_3] & [KCd_4] & 5 & 6 & 3 & 1 & 5 & 6 & 3 & 5 & 6 & 5 \end{array}$$

Figure 3.14. Dimension Nominal Values, Directions, and Upper and Lower Limits

$$[Dt] = \begin{array}{c|c} \left[\begin{matrix} Normal \\ Uniform \\ Uniform \\ Normal \\ Normal \\ Normal \\ Uniform \\ Normal \\ Normal \\ Normal \\ Uniform \\ Normal \\ Normal \\ Normal \\ Normal \end{matrix}\right] & \begin{matrix} D_1 \\ D_2 \\ D_3 \\ D_4 \\ D_5 \\ D_6 \\ D_7 \\ D_8 \\ D_9 \\ D_{10} \\ D_{11} \\ D_{12} \\ D_{13} \\ D_{14} \\ D_{15} \end{matrix} \end{array} \qquad [Dp] = \begin{array}{cc} \left[\begin{matrix} 0 & .03 & NaN \\ -.028 & .028 & NaN \\ -.0015 & -.0023 & NaN \\ 1.36 & .01 & NaN \\ 1.36 & .006 & NaN \\ -40.34 & .02 & NaN \\ 2.75 & 2.91 & NaN \\ 0 & .0035 & NaN \\ -.185 & .001 & NaN \\ -40.24 & .028 & NaN \\ -2.89 & -2.73 & NaN \\ .003 & .0028 & NaN \\ -35.18 & .05 & NaN \\ 8.41 & .04 & NaN \\ -2.96 & .04 & NaN \end{matrix}\right] & \begin{matrix} D_1 \\ D_2 \\ D_3 \\ D_4 \\ D_5 \\ D_6 \\ D_7 \\ D_8 \\ D_9 \\ D_{10} \\ D_{11} \\ D_{12} \\ D_{13} \\ D_{14} \\ D_{15} \end{matrix} \end{array}$$

Figure 3.15. Distribution Types & Parameters for Dimensions

Notice that some of the entries in **[*Dn*]*2,2*** are unknown variables from **[*KCn*]*1,1***. In this way, two configurations of a Type-II assembly can be linked together. In general, if the set of dimensions for an arbitrary configuration, $\boldsymbol{B_{i,j}}$, includes one or more linking KCs from one or more configurations of a previous state, when forming $\boldsymbol{[Dn]_{i,j}}$ and $\boldsymbol{[Dd]_{i,j}}$, the nominal values (even if unknown) and directions of the linking KCs should be entered before the values and directions of the rest of the dimensions. Following this convention facilitates the process of relating configurations to each other that will be shown in later in this chapter and also in Chapter 4.

3.1.7 Identifying Real Mates

An arbitrary configuration $\boldsymbol{B_{i,j}}$ in an assembly contains $\boldsymbol{m_{i,j}}$ real mates that connect pairs of features. Mates are each labeled using a triple subscripted $\boldsymbol{M}$ such that $\boldsymbol{M_{i,j,k}}$ is the $\boldsymbol{k}$th mate in the $\boldsymbol{j}$th configuration in the $\boldsymbol{i}$th assembly state. $\boldsymbol{M_{3,3,1}}$ would therefore be the first mate in the third configuration of the third assembly state.

Analysis of fingernail clipper reveals that $\boldsymbol{B_{1,1}}$ has three mates and $\boldsymbol{B_{2,2}}$ has four mates ($\boldsymbol{m_{1,1}}$ = 3 and $\boldsymbol{m_{2,2}}$ = 4). Table 3.1 lists the mates, the features they join, and a brief description of each (more concise descriptions will be given later). Figure 3.16 and Figure 3.17 show the mates on the diagrams for $\boldsymbol{B_{1,1}}$ and $\boldsymbol{B_{2,2}}$.

Real Mate	Features It Joins	Description
$M_{1,1,1}$	$E_{1,1,1}$ & $E_{1,1,3}$	rigid clamp allowing no motion
$M_{1,1,2}$	$E_{1,1,4}$ & $E_{1,1,5}$	two planar surfaces in contact
$M_{1,1,3}$	$E_{1,1,1}$ & $E_{1,1,6}$	clamp allowing some motion
$M_{2,2,1}$	$E_{2,2,4}$ & $E_{2,2,5}$	compliant revolute joint
$M_{2,2,2}$	$E_{2,2,6}$ & $E_{2,2,7}$	revolute joint
$M_{2,2,3}$	$E_{2,2,8}$ & $E_{2,2,9}$	revolute joint
$M_{2,2,4}$	$E_{2,2,2}$ & $E_{2,2,10}$	cylinder on planar surface

Table 3.5. Description of Real Mates in Fingernail Clipper

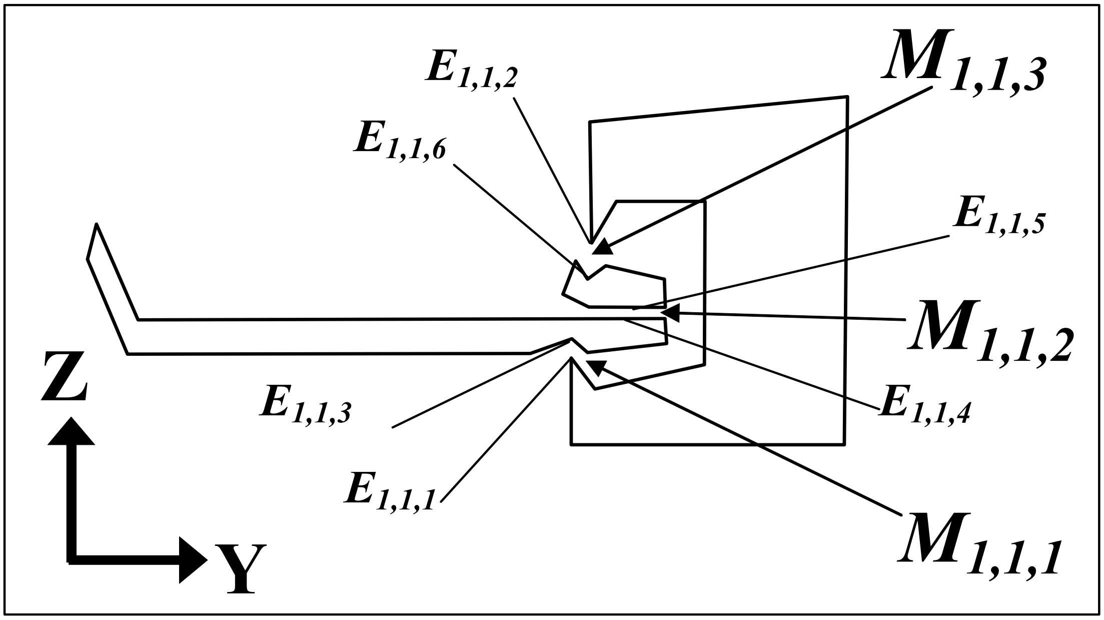


Figure 3.16. Enlarged Diagram of Configuration $B_{1,1}$ Showing Real Mates

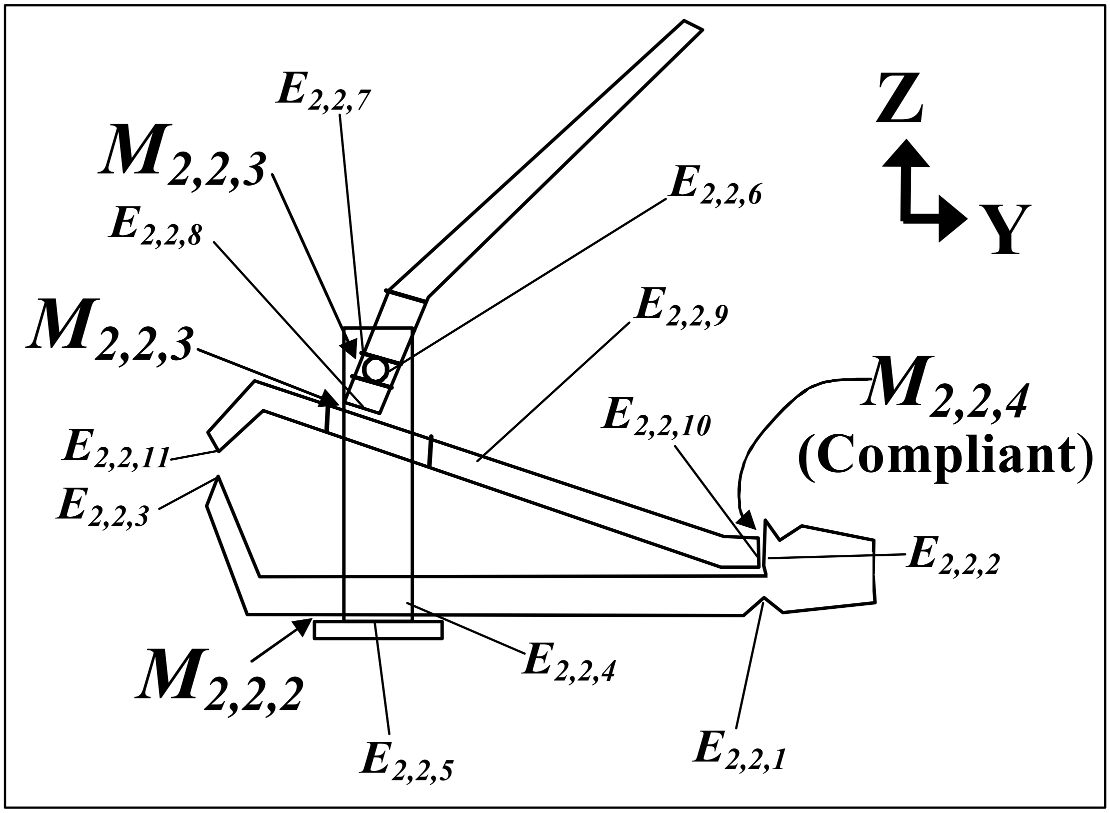


Figure 3.17. Enlarged Diagram of Configuration $B_{2,2}$ Showing Real Mates

An arbitrary real mate, $\boldsymbol{M_{i,j,k}}$, is composed of a six elements where each element is either a DOF, a DOS, or a DOC. In other words, $\boldsymbol{M_{i,j,k}}$ can be expressed as a continuous serial chain of $\boldsymbol{f_{i,j,k}}$ DOF, $\boldsymbol{c_{i,j,k}}$ DOC, and $\boldsymbol{s_{i,j,k}}$ DOS. $\boldsymbol{f_{i,j,k}}$ is the number of DOF in $\boldsymbol{M_{i,j,k}}$. Likewise, $\boldsymbol{c_{i,j,k}}$ is the number of DOC and $\boldsymbol{s_{i,j,k}}$ is the number of DOS in $\boldsymbol{M_{i,j,k}}$. In each chain, the DOC must always come first followed by the DOS and then the DOF. $\boldsymbol{f_{i,j,k}}$, $\boldsymbol{c_{i,j,k}}$, and $\boldsymbol{s_{i,j,k}}$ for any real mate must always be zero or positive and will always obey the following relationship:

$$
\begin{array}{l}
f_{i,j,k} + c_{i,j,k} + s_{i,j,k} = 6 \\
for \\
i = 1,2,...,a \\
j = 1,2,...,b_i \\
k = 1,2,...,m_{i,j}
\end{array}
$$

Equation 3.8

If $\boldsymbol{M_{i,j,k}}$ is a compliant mate, then it will have an associated **($\boldsymbol{s_{i,j,k}}$ x $\boldsymbol{s_{i,j,k}}$)** matrix $[\boldsymbol{Sk}]_{i,j,k}$, called the mate-level stiffness matrix. Entry $[\boldsymbol{Sk_{l,n}}]_{i,j,k}$ is equal to the magnitude of force or moment transmitted in the direction of the $\boldsymbol{l}^{th}$ DOS when a unit deflection occurs in the $\boldsymbol{n}^{th}$ DOS. If $\boldsymbol{M_{i,j,k}}$ does not have any DOS, then it is a non-compliant mate which means that $[\boldsymbol{Sk}]_{i,j,k}$, is a null matrix and Equation 3.8 simplifies as follows:

$$f_{i,j,k} + c_{i,j,k} = 6$$
$$for$$
$$i = 1,2,...,a$$
$$j = 1,2,...,b_i$$
$$k = 1,2,...,m_{i,j}$$

Equation 3.9

For an arbitrary configuration ***B*$_{i,j}$**, the total number of DOF, ***ft*$_{i,j}$**, the total number of DOC, ***ct*$_{i,j}$**, and the total number of DOS, ***st*$_{i,j}$**, are computed respectively as follows:

$$ft_{i,j} = \sum_{k=1}^{m_{i,j}} f_{i,j,k}$$
$$for$$
$$i = 1,2,...,a$$
$$j = 1,2,...,b_i$$

Equation 3.10

$$ct_{i,j} = \sum_{k=1}^{m_{i,j}} c_{i,j,k}$$
$$for$$
$$i = 1,2,...,a$$
$$j = 1,2,...,b_i$$

Equation 3.11

$$st_{i,j} = \sum_{k=1}^{m_{i,j}} s_{i,j,k}$$
$$for$$
$$i = 1,2,...,a$$
$$j = 1,2,...,b_i$$

Equation 3.12

$\boldsymbol{ft_{i,j}}$, $\boldsymbol{ct_{i,j}}$, and $\boldsymbol{st_{i,j}}$ for any configuration must always be zero or positive and must always obey the following relationship:

$$
\begin{aligned}
& ft_{i,j} + ct_{i,j} + st_{i,j} = 6m_{i,j} \\
& for \\
& i = 1,2,...,a \\
& j = 1,2,...,b_i
\end{aligned}
$$

Equation 3.13

For an arbitrary configuration, $\boldsymbol{B_{i,j}}$, the initial guesses for the nominal values of the DOF and DOS and exact nominal values for the DOC are contained in a **(*$ft_{i,j}$* x *1*)** vector, **[*Fn*]*$_{i,j}$***, a **(*$st_{i,j}$*** x ***1*)** vector, **[*Sn*]*$_{i,j}$***, and a **(*$ct_{i,j}$* x *1*)** vector, **[*Cn*]*$_{i,j}$*** respectively called the configuration level DOF, DOC, and DOS nominal value matrices. **[*Cn_k*]*$_{i,j}$*** is therefore the nominal value of the ***k***th DOC in the ***j***th configuration of the ***i***th assembly state (angles should be expressed in radians not degrees). Upper bounds on the nominal values of the DOF and DOS are contained in a **(*$ft_{i,j}$* x *1*)** vector, **[*Ful*]*$_{i,j}$***, and a **(*$st_{i,j}$* x *1*)** vector, **[*Sul*]*$_{i,j}$***, respectively called the configuration-level upper DOF and upper DOS bound matrices. Lower bounds on the nominal values of the DOF and DOS are contained in a **(*$ft_{i,j}$* x *1*)** vector, **[*Fll*]*$_{i,j}$***, and a **(*$st_{i,j}$* x *1*)** vector, **[*Sll*]*$_{i,j}$***, respectively called the configuration-level lower DOF and lower DOS bound matrices.

The directions and pitches for the DOF, DOC, and DOS are contained in a **(*$ft_{i,j}$* x *4*)** matrix, **[*Fd*]*$_{i,j}$***, a **(*$ct_{i,j}$* x *4*)** matrix, **[*Cd*]*$_{i,j}$***, and a **(*$st_{i,j}$* x *4*)** matrix, **[*Sd*]*$_{i,j}$*** respectively called the configuration level DOF, DOC, and DOS direction matrices. The first three entries in each row of $[\boldsymbol{Fd}]_{i,j}$ are the $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$ components of a unit vector that describes the direction of translational, rotational or helical motion allowed by a DOF, the first three entries in each row of $[\boldsymbol{Cd}]_{i,j}$ are the $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$ components of a unit vector that describes the direction in which forces, moments, or force/moment couples are transmitted by a DOC, and the first three entries in each row of $[\boldsymbol{Sd}]_{i,j}$ are the $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$ components of a unit vector that describes the energy-storing direction of a DOS. The vector is located in a coordinate frame local to the corresponding DOF, DOC, or DOS. The final entry in each row of $[\boldsymbol{Fd}]_{i,j}$, $[\boldsymbol{Cd}]_{i,j}$, and $[\boldsymbol{Sd}]_{i,j}$ is the pitch of the DOF, DOC, or DOS and can be any number between zero (for pure rotation) and infinity (for pure translation).

For example, to find the direction of the second DOS in the third configuration of the fourth assembly state, one would look at $[\boldsymbol{Sd}_{2,1:4}]_{4,3}$. Further, if $[\boldsymbol{Cd}_{2,1:4}]_{4,3}$ equals $[\boldsymbol{0.7071} , \boldsymbol{0.7071} , \boldsymbol{0} , \infty]$, this would mean the second DOC in third configuration of the fourth assembly state is capable of transmitting a pure translational force directed along an vector that is rotated forty-five degrees from the $\boldsymbol{X}$-axis in the $\boldsymbol{X}$-$\boldsymbol{Y}$ plane of the coordinate frame local to that DOC. If $[\boldsymbol{Fd}_{3,1:4}]_{1,1}$ equals $[\boldsymbol{0} , \boldsymbol{0} , \boldsymbol{1} , \boldsymbol{0}]$, this would mean the third DOF in first configuration of the first assembly state is a pure rotational motion directed along an axis aligned with the $\boldsymbol{Z}$-axis of the coordinate frame local to that DOF.

For an arbitrary configuration, $\boldsymbol{B}_{i,j}$, the stiffness matrices for all the compliant mates can be contained in an $(\boldsymbol{st}_{i,j}$ x $\boldsymbol{st}_{i,j})$ matrix, $[\boldsymbol{SK}]_{i,j}$, called the configuration-level stiffness matrix that is composed as follows:

$$[SK]_{i,j}=\begin{bmatrix} [Sk]_{i,j,1} & \left[0_{\left(s_{i,j,1}\times s_{i,j,2}\right)}\right] & \cdots & \left[0_{\left(s_{i,j,1}\times s_{i,j,m_{i,j}}\right)}\right] \\ \left[0_{\left(s_{i,j,2}\times s_{i,j,1}\right)}\right] & [Sk]_{i,j,2} & & \vdots \\ \vdots & & \ddots & \vdots \\ \left[0_{\left(s_{i,j,m_{i,j}}\times s_{i,j,1}\right)}\right] & \cdots & \cdots & [Sk]_{i,j,m_{i,j}} \end{bmatrix}$$

Equation 3.14

Entry $\boldsymbol{[SK_{k,l}]_{i,j}}$ is equal to the magnitude of force or moment transmitted in the direction of the $\boldsymbol{k}$th DOS when a unit deflection occurs in the $\boldsymbol{l}$th DOS in configuration $\boldsymbol{B_{i,j}}$.

Since individual DOF, DOC, and DOS are sometimes present in multiple configurations, three other numbers, ***ftt***, ***ctt***, and ***stt*** are defined as the total number of unique DOF, DOC, and DOS in an entire assembly respectively. Since DOF, DOC, and DOS may be present in multiple configurations, the sum of $\boldsymbol{ft_{i,j}}$, $\boldsymbol{ct_{i,j}}$, and $\boldsymbol{st_{i,j}}$ for all configurations in an assembly does not generally yield ***ftt***, ***ctt***, and ***stt*** respectively. This can be explicitly expressed as follows::

$$ftt \neq \sum_{i=1}^{a}\sum_{j=1}^{b_i} ft_{i,j}$$

$$ctt \neq \sum_{i=1}^{a}\sum_{j=1}^{b_i} ct_{i,j}$$

$$stt \neq \sum_{i=1}^{a}\sum_{j=1}^{b_i} st_{i,j}$$

Equation 3.15

The DOF, DOC, and DOS elements for an entire assembly are respectively labeled $\boldsymbol{F_1}$, $\boldsymbol{F_2}$,…, $\boldsymbol{F_{ftt}}$, $\boldsymbol{C_1}$, $\boldsymbol{C_2}$,…, $\boldsymbol{C_{ctt}}$, and $\boldsymbol{S_1}$, $\boldsymbol{S_2}$,…, $\boldsymbol{S_{stt}}$. For example, $\boldsymbol{S_9}$ would be the ninth DOS of an assembly and can be included in one or more configurations. The directions and pitches for the DOF, DOC, and DOS of an entire assembly are contained in a **(*ftt* x *4*)** matrix, **[*Fd*],** a **(*ctt* x *4*)** matrix, **[*Cd*],** and a **(*stt* x *4*)** matrix, **[*Sd*]** respectively called the DOF, DOC, and DOS direction matrices. The first three entries in each row of **[*Fd*]** are the ***x***, ***y***, and ***z*** components of a unit vector that describes the direction of translational, rotational or helical motion allowed by a DOF, the first three entries in each row of **[*Cd*]** are the ***x***, ***y***, and ***z*** components of a unit vector that describes the direction in which forces, moments, or force/moment couples are transmitted by a DOC, and the first three entries in each row of **[*Sd*]** are the ***x***, ***y***, and ***z*** components of a unit vector that describes the energy-storing direction of a DOS. The vector is located in a coordinate frame local to the DOF, DOC, or DOS. The final entry in each row of **[*Fd*],** **[*Cd*],** and **[*Sd*]** is the pitch of the DOF, DOC, or DOS and can be any number between zero (for pure rotation) and infinity (for pure translation).

For example to find the direction for $\boldsymbol{S_4}$ in the assembly one would look at **[**$\boldsymbol{Sd_{4,1:4}}$**]**. If **[**$\boldsymbol{Fd_{6,1:4}}$**]** equals **[*.866 , 0 , .5 , 2*]**, this would mean the sixth DOF is a helical motion along a vector that is rotated thirty degrees the ***X***-axis in the ***X-Z*** plane of the coordinate frame local to that DOF. One radian of rotation about that vector will cause two length units of translation along that vector.

For an entire assembly, the nominal stiffness matrices can be compiled into an **(*stt* x *stt*)** matrix, **[*SK*]** (without subscripts outside of the brackets), called the assembly-level stiffness matrix. Entry **[*$SK_{i,j}$*]** is equal to the magnitude of force or moment transmitted in the direction of the ***i***th DOS when a unit deflection occurs in the ***j***th DOS of the assembly.

Analysis of fingernail clipper reveals that configuration $\boldsymbol{B_{1,1}}$ has 6 DOF, 12 DOC, and 0 DOS and configuration $\boldsymbol{B_{2,2}}$ has 6 DOF, 17 DOC, and 1 DOS. There are no DOF, DOC, or DOS that are part of more than one configuration ($\boldsymbol{ft_{1,1}}$ = 6, $\boldsymbol{ct_{1,1}}$ = 12, $\boldsymbol{st_{1,1}}$ = 0, $\boldsymbol{ft_{2,2}}$ = 6, $\boldsymbol{ct_{2,2}}$ = 17, $\boldsymbol{st_{2,2}}$ = 1, ***ftt*** = 12, ***ctt*** = 29, and ***stt*** = 1). Table 3.6 shows the serial chain of continuous DOF, DOC, and DOS that comprise each real mate in configurations $\boldsymbol{B_{1,1}}$ and $\boldsymbol{B_{2,2}}$ of the fingernail clipper. Figure 3.18 shows $[\boldsymbol{Fn}]_{1,1}$, $[\boldsymbol{Ful}]_{1,1}$, $[\boldsymbol{Fll}]_{1,1}$, $[\boldsymbol{Sn}]_{1,1}$, $[\boldsymbol{Sul}]_{1,1}$, $[\boldsymbol{Sll}]_{1,1}$, $[\boldsymbol{Cn}]_{1,1}$, $[\boldsymbol{Fd}]_{1,1}$, $[\boldsymbol{Sd}]_{1,1}$, $[\boldsymbol{Cd}]_{1,1}$, $[\boldsymbol{SK}]_{1,1}$, $[\boldsymbol{Fn}]_{2,2}$, $[\boldsymbol{Ful}]_{2,2}$, $[\boldsymbol{Fll}]_{2,2}$, $[\boldsymbol{Sn}]_{2,2}$, $[\boldsymbol{Sul}]_{2,2}$, $[\boldsymbol{Sll}]_{2,2}$, $[\boldsymbol{Cn}]_{2,2}$, $[\boldsymbol{Fd}]_{2,2}$, $[\boldsymbol{Sd}]_{2,2}$, $[\boldsymbol{Cd}]_{2,2}$, $[\boldsymbol{SK}]_{2,2}$, **[*Fd*]**, **[*Sd*]**, **[*Cd*]**, and **[*SK*]** for the fingernail clipper. It is important to note that the values in $[\boldsymbol{Fn}]_{1,1}$, $[\boldsymbol{Sn}]_{1,1}$, $[\boldsymbol{Fn}]_{2,2}$, and $[\boldsymbol{Sn}]_{2,2}$ are only initial guesses for the nominal values. This is because the actual nominal values of the DOF and DOS can only be determined exactly after the parts are assembled analytically. Later how to solve for these values exactly (it is this solving process that requires the nominal values of the DOF and DOS to be bounded).

$$[Fn]_{1,1}{}^{T} = \begin{array}{cccccc} F_1 & F_2 & F_3 & F_4 & F_5 & F_6 \\ \hline 0 & 0 & 0 & 3 & 0 & 0 \end{array} \quad [Ful]_{1,1}{}^{T} = -[Fll]_{1,1}{}^{T} = \begin{array}{cccccc} F_1 & F_2 & F_3 & F_4 & F_5 & F_6 \\ \hline 5 & 5 & 1.57 & 10 & 1.57 & 1.57 \end{array}$$

$$[Fn]_{2,2}{}^{T} = \begin{array}{cccccc} F_7 & F_8 & F_9 & F_{10} & F_{11} & F_{12} \\ \hline 0 & 1.57 & 0 & -1.57 & 0 & 35 \end{array} \quad [Ful]_{2,2}{}^{T} = -[Fll]_{2,2}{}^{T} = \begin{array}{cccccc} F_7 & F_8 & F_9 & F_{10} & F_{11} & F_{12} \\ \hline 1.57 & 3.14 & 5 & 3.14 & 1.57 & 50 \end{array}$$

$$[Fd]^{T} = \begin{bmatrix} [Fd]_{1,1} \\ [Fd]_{2,2} \end{bmatrix}^{T} = \begin{array}{cccccc|cccccc} F_1 & F_2 & F_3 & F_4 & F_5 & F_6 & F_7 & F_8 & F_9 & F_{10} & F_{11} & F_{12} \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ \infty & \infty & 0 & \infty & 0 & 0 & 0 & 0 & \infty & 0 & 0 & \infty \end{array}$$

$$[Cn]_{1,1}{}^{T} = \begin{array}{cccccccccccc} C_1 & C_2 & C_3 & C_4 & C_5 & C_6 & C_7 & C_8 & C_9 & C_{10} & C_{11} & C_{12} \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

$$[Sn]_{1,1} = [Sul]_{1,1}{}^{T} = [Sll]_{1,1}{}^{T} = [\emptyset]$$

$$[Sn]_{2,2}{}^{T} = \begin{array}{c} S_1 \\ {[0]} \end{array} \qquad [Sul]_{2,2}{}^{T} = -[Sll]_{2,2}{}^{T} = \begin{array}{c} S_1 \\ {[1.57]} \end{array}$$

$$[Sd] = \begin{bmatrix} [Sd]_{1,1} \\ [Sd]_{2,2} \end{bmatrix} = \begin{bmatrix} [\emptyset] \\ \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \end{bmatrix} S_1$$

$$[SK] = \begin{bmatrix} [SK]_{1,1} & \left[0_{(st_{1,1} \times st_{2,2})}\right]_{1,1} \\ \left[0_{(st_{1,1} \times st_{2,2})}\right]_{1,1} & [SK]_{2,2} \end{bmatrix} = \begin{array}{cc|c} & S_1 & \\ {[\emptyset]} & [\emptyset] & \\ {[\emptyset]} & 25000 & S_1 \end{array}$$

$$[Cn]_{2,2}{}^{T} = \begin{array}{ccccccccccccccccc} C_{13} & C_{14} & C_{15} & C_{16} & C_{17} & C_{18} & C_{19} & C_{20} & C_{21} & C_{22} & C_{23} & C_{24} & C_{25} & C_{26} & C_{27} & C_{28} & C_{29} \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

$$[Cd] = \begin{bmatrix} [Cd]_{1,1} \\ [Cd]_{2,2} \end{bmatrix} = \begin{array}{cccc|c} 1 & 0 & 0 & \infty & C_1 \\ 0 & 1 & 0 & \infty & C_2 \\ 0 & 0 & 1 & \infty & C_3 \\ 1 & 0 & 0 & 0 & C_4 \\ 0 & 1 & 0 & 0 & C_5 \\ 0 & 0 & 1 & 0 & C_6 \\ 0 & 0 & 1 & \infty & C_7 \\ 1 & 0 & 0 & 0 & C_8 \\ 0 & 1 & 0 & 0 & C_9 \\ 1 & 0 & 0 & \infty & C_{10} \\ 0 & 1 & 0 & \infty & C_{11} \\ 0 & 0 & 1 & 0 & C_{12} \\ \hline 1 & 0 & 0 & \infty & C_{13} \\ 0 & 1 & 0 & \infty & C_{14} \\ 0 & 0 & 1 & \infty & C_{15} \\ 1 & 0 & 0 & 0 & C_{16} \\ 0 & 1 & 0 & 0 & C_{17} \\ 1 & 0 & 0 & 0 & C_1 \\ 0 & 1 & 0 & \infty & C_{19} \\ 0 & 0 & 1 & \infty & C_{20} \\ 0 & 1 & 0 & \infty & C_{21} \\ 0 & 0 & 1 & 0 & C_{22} \\ 0 & 0 & 1 & 0 & C_{23} \\ 0 & 1 & 0 & \infty & C_{24} \\ 1 & 0 & 0 & 0 & C_{25} \\ 0 & 1 & 0 & \infty & C_{26} \\ 0 & 0 & 1 & \infty & C_{27} \\ 0 & 1 & 0 & \infty & C_{28} \\ 0 & 0 & 1 & 0 & C_{29} \end{array}$$

Figure 3.18. DOF, DOC, and DOS Nominal Values, Directions, & Pitches

Real Mate	Features It Joins	DOF (direction)	DOC (direction)	DOS (direction)	Chain of DOF, DOS, & DOC Comprising Each Real Mate (in order from 1st Feature to 2nd Feature)
$M_{1,1,1}$	$E_{1,1,1}$ & $E_{1,1,3}$	None	6 (x,y,z,θx,θy,θz)	None	C_1,C_2,C_3,C_4,C_5,C_6
$M_{1,1,2}$	$E_{1,1,4}$ & $E_{1,1,5}$	3 (x,y,θz)	3 (z,θx,θy)	None	C_7,C_8,C_9,F_1,F_2,F_3
$M_{1,1,3}$	$E_{1,1,1}$ & $E_{1,1,6}$	3 (z,θx,θy)	3 (x,y,θz)	None	$C_{10},C_{11},C_{12},F_4,F_5,F_6$
$M_{2,2,1}$	$E_{2,2,4}$ & $E_{2,2,5}$	1 (θz)	3 (x,y,z,θx,θy)	None	$C_{13},C_{14},C_{15},C_{16},C_{17},F_7$
$M_{2,2,2}$	$E_{2,2,6}$ & $E_{2,2,7}$	1 (θx)	3 (x,y,z,θy,θz)	None	$C_{18},C_{19},C_{20},C_{21},C_{22},F_8$
$M_{2,2,3}$	$E_{2,2,8}$ & $E_{2,2,9}$	1 (x,y,θx,θz)	3 (z,θy)	None	$C_{23},C_{24},F_9,F_{10},F_{11},F_{12}$
$M_{2,2,4}$	$E_{2,2,2}$ & $E_{2,2,10}$	None	3 (x,y,z,θy,θz)	1 (θx)	$C_{25},C_{26},C_{27},C_{28},C_{29},S_1$

Table 3.6. Chain of DOF, DOC & DOS Comprising Each Real Mate

3.1.8 Creating Annotated Liaison Diagrams

Using the data above, each assembly can be represented with ***bt*** annotated liaison diagrams [Bourjault], one for each configuration. In this work, the annotated liaison diagram for an arbitrary configuration $\boldsymbol{B_{i,j}}$ consists of $\boldsymbol{p_{i,j}}$ circles representing the $\boldsymbol{p_{i,j}}$ parts (remember that compliant entities like the "cutter" are modeled as multiple parts related by compliant mates), $\boldsymbol{e_{i,j}}$ nodes representing the $\boldsymbol{e_{i,j}}$ features, $\boldsymbol{m_{i,j}}$ arcs representing the $\boldsymbol{m_{i,j}}$ real mates between parts, $\boldsymbol{r_{i,j}}$ arcs representing the $\boldsymbol{r_{i,j}}$ intra-part relationships inside of parts, and $\boldsymbol{h_{i,j}}$ directed double lines representing the $\boldsymbol{h_{i,j}}$ KC feature pairs. Each real mate, intra-part relationship, and KC feature pair, must also be assigned a direction. This direction always points away from the first feature in each relationship. In addition, each annotated liaison diagram must be assigned a base feature.

The annotated liaison diagrams for configurations $\boldsymbol{B_{1,1}}$ and $\boldsymbol{B_{2,2}}$, are shown in Figure 3.19. A key describing the six different symbols and the seven different labels used in the diagrams is shown in Figure 3.20. The base feature for $\boldsymbol{B_{1,1}}$ is $\boldsymbol{E_{1,1,1}}$ and the base feature for $\boldsymbol{B_{2,2}}$ is $\boldsymbol{E_{2,2,1}}$. The base part for $\boldsymbol{B_{1,1}}$ is $\boldsymbol{P_{1,1,1}}$ (the fixture) and the base part for $\boldsymbol{B_{2,2}}$ is $\boldsymbol{P_{2,2,1}}$ (the cutter base).

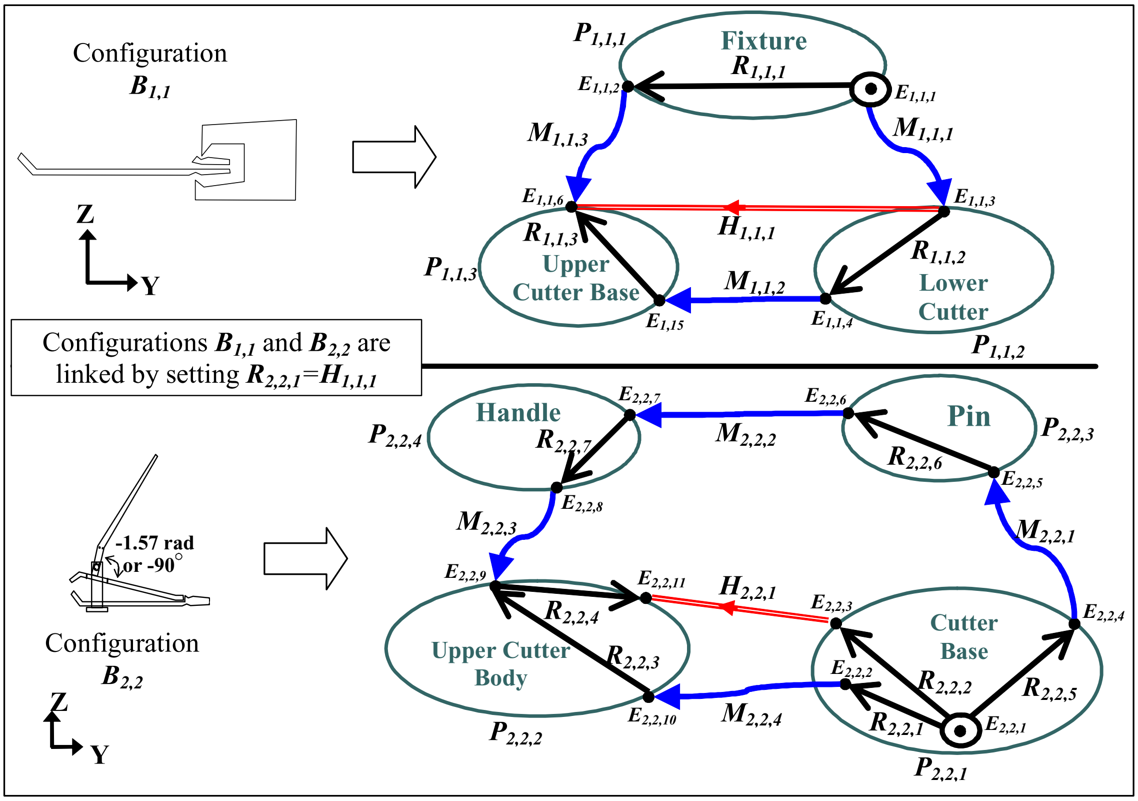


Figure 3.19. Annotated Liaison Diagrams for Fingernail Clipper

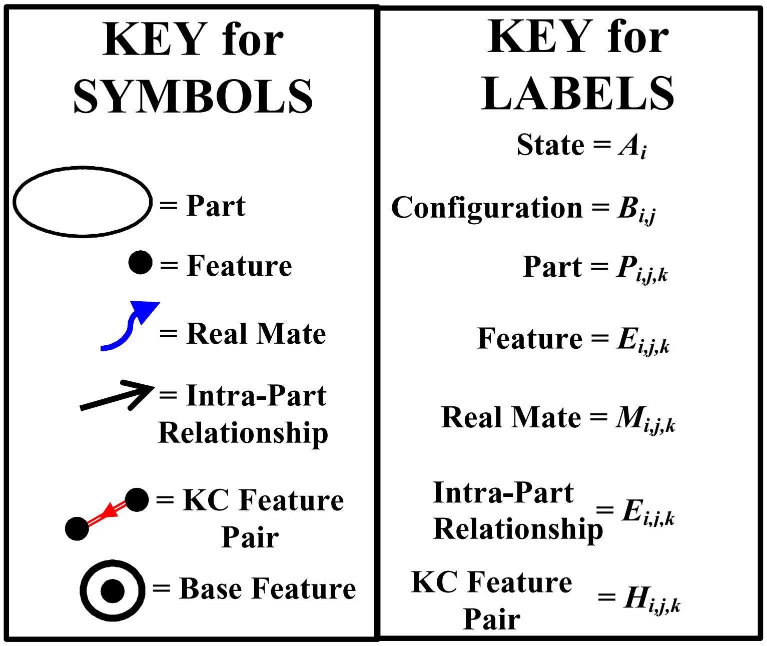


Figure 3.20. Key for Annotated Liaison Diagram

3.1.9 Finding Closed-Loops & Open-Loops

An arbitrary configuration $\boldsymbol{B}_{i,j}$ in an assembly contains $\boldsymbol{\lambda}_{i,j}$ independent closed-loops and $\boldsymbol{g}_{i,j}$ independent open-loops (it will be shown later how to find the set of independent loops for a configuration). Independent closed-loops and independent open-loops are each labeled using a triple subscripted $\boldsymbol{\Lambda}$ or $\boldsymbol{G}$ respectively such that $\boldsymbol{\Lambda}_{i,j,k}$ is the $\boldsymbol{k}$th independent closed-loop in the $\boldsymbol{j}$th configuration in the $\boldsymbol{i}$th assembly state and $\boldsymbol{G}_{i,j,k}$ is the $\boldsymbol{k}$th independent open-loop in the $\boldsymbol{j}$th configuration in the $\boldsymbol{i}$th assembly state. For an arbitrary configuration $\boldsymbol{B}_{i,j}$, the number of independent closed-loops, $\boldsymbol{\lambda}_{i,j}$ can be computed as follows (adapted from [Seshu and Reed]):

$$\begin{aligned}
&\lambda_{i,j} = m_{i,j} - p_{i,j} + 1 \\
&\mathit{for} \\
&i = 1,2,\ldots,a \\
&j = 1,2,\ldots b_i
\end{aligned}$$

Equation 3.16

A closed-loop exists in a configuration when one can start at a feature, traverse the arcs (either real mates or intra-part relationships) to other features, and return to the same feature without back tracking or crossing the path of travel. The base feature should always be selected so that it is part of at least one closed-loop, unless the configuration has no closed-loops, in which case any feature will suffice.

An open-loop in a configuration exists when one can start at the base feature and traverse the arcs to other features, finally arriving at a feature that only shares one arc with other features (note that the feature at the end of each open-loop should be part of a KC feature pair, otherwise there is no reason to include the open-loop in the model). Each open-loop, $\boldsymbol{G_{i,j,k}}$, is assigned a virtual mate, $\boldsymbol{V_{i,j,k}}$, that is composed of a chain of six serially-linked DOV that starts at the base feature and ends at the feature at the end of the open-loop the virtual mate is assigned to. Each open-loop must also be assigned to one of the $\boldsymbol{h_{i,j}}$ KC feature pairs. Each virtual mate contains one DOV for each of the coordinate directions local to a coordinate frame that is located at the first feature of the KC feature pair the open-loop is assigned to. The DOV of virtual mates are always put in the following order: $\boldsymbol{\theta x}$ first, $\boldsymbol{\theta y}$ second, $\boldsymbol{\theta z}$ third, $\boldsymbol{x}$ fourth, $\boldsymbol{y}$ fifth, and $\boldsymbol{z}$ sixth. The total number of DOV, for an arbitrary configuration is equal to $\boldsymbol{6g_{i,j}}$. Individual open-loops and hence individual DOV are sometimes present in multiple configurations so $\boldsymbol{vtt}$ is defined as the total number of unique DOV in an entire assembly. The DOV in an entire assembly are labeled $\boldsymbol{V_1}$, $\boldsymbol{V_2}$,…, $\boldsymbol{V_{vtt}}$ and in general:

$$vtt \neq \sum_{i=1}^{a}\sum_{j=1}^{b_i} 6g_{i,j}$$

Equation 3.17

For a given configuration, a complete set of independent loops, both open and closed, exists when each loop in the set contains at least one real mate or intra-part relationship that is not shared by any other loop (either open or closed) and all real mates and intra-part relationships are part of at least one loop (either open or closed). Figure 3.21 and Figure 3.22 indicate the locations of the independent loops in configuration $\boldsymbol{B_{2,2}}$ and $\boldsymbol{B_{1,1}}$ of the fingernail clipper respectively ($\lambda_{1,1}$ = 1, $g_{1,1}$ = 0, $\lambda_{2,2}$ = 1, and $g_{2,2}$ = 2). Table 3.7 shows the set of features contained in each loop and the set of serially-linked DOV connecting the base feature to the end of each open-loop.

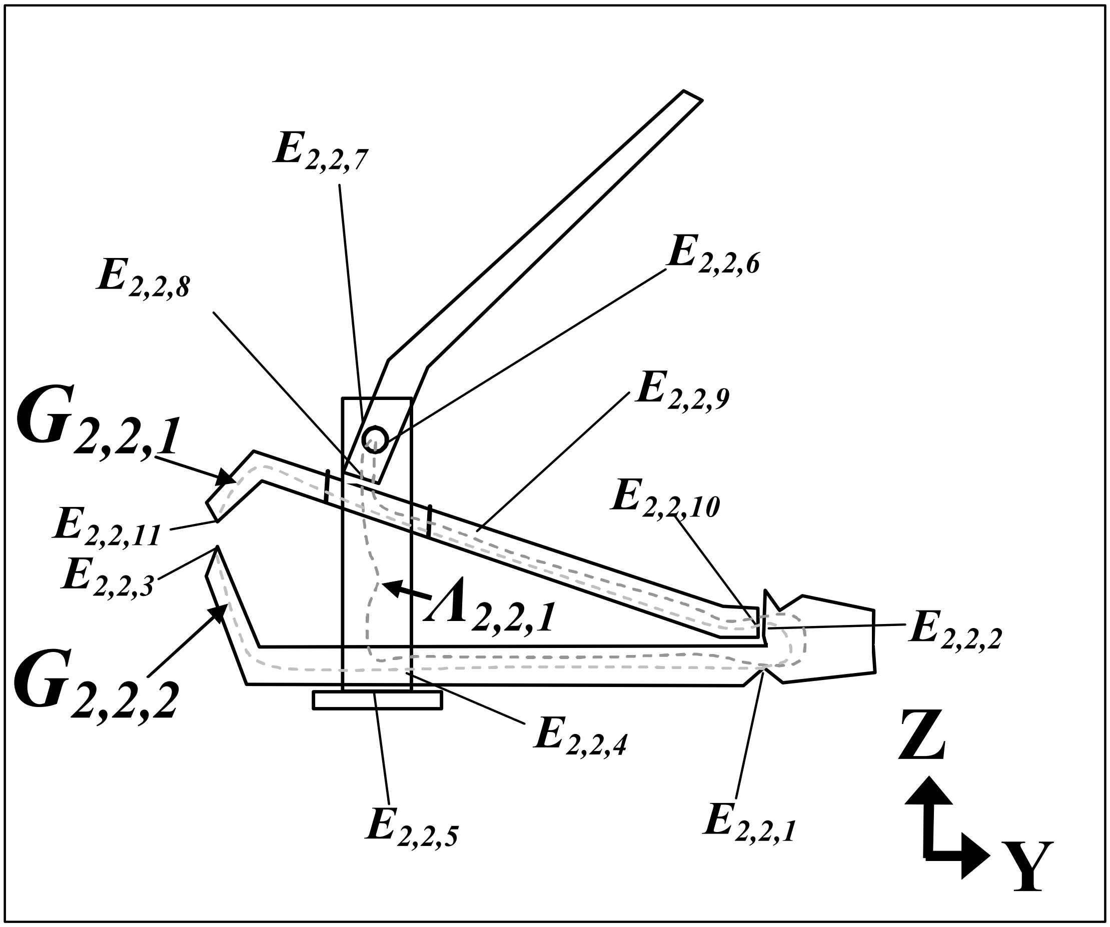


Figure 3.21. Enlarged Diagram of Configuration $B_{2,2}$ Showing Independent Loops

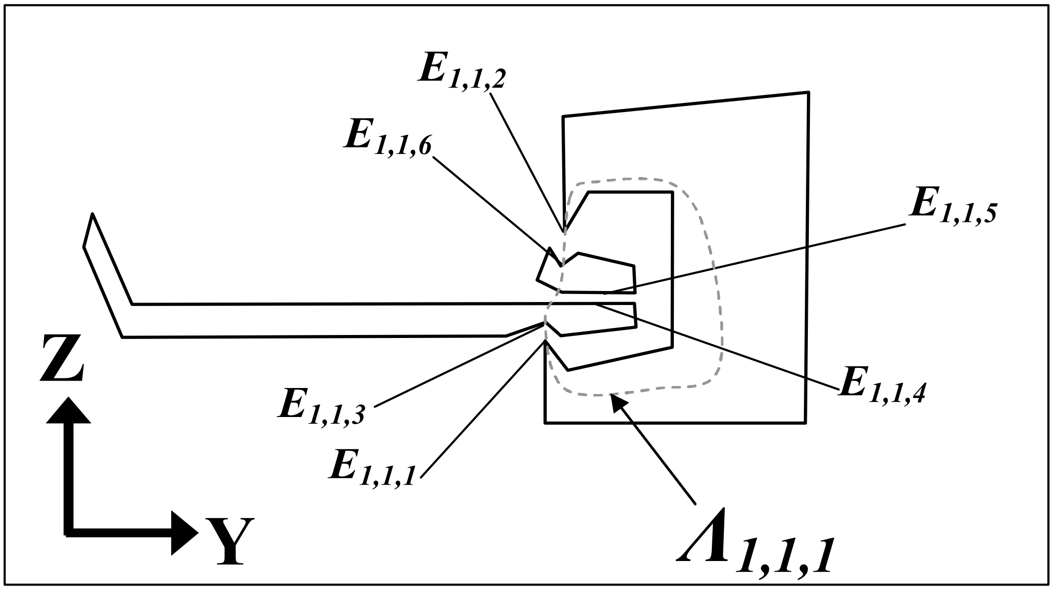


Figure 3.22. Enlarged Diagram of Configuration $B_{1,1}$ Showing Closed-Loop

Closed-Loop	Features it Contains (in order from beginning to end)
$\Lambda_{1,1,1}$	$E_{1,1,1}, E_{1,1,3}, E_{1,1,4}, E_{1,1,5}, E_{1,1,6}, E_{1,1,2}, E_{1,1,1},$
$\Lambda_{2,2,1}$	$E_{2,2,1}, E_{2,2,4}, E_{2,2,5}, E_{2,2,6}, E_{2,2,7}, E_{2,2,8}, E_{2,2,9}, E_{2,2,10}, E_{2,2,2}, E_{2,2,1}$

Open-Loop	Features it Contains (in order from beginning to end)	Chain of DOV Connecting Base and End Features (in order from Base to End)	KC Feature Pair it is assigned to	First Feature in this KC Feature Pair
$G_{2,2,1}$	$E_{2,2,1}, E_{2,2,2}, E_{2,2,10}, E_{2,2,9}, E_{2,2,11}$	$V_1, V_2, V_3, V_4, V_5, V_6$	$H_{2,2,1}$	$E_{2,2,3}$
$G_{2,2,2}$	$E_{2,2,1}, E_{2,2,3}$	$V_7, V_8, V_9, V_{10}, V_{11}, V_{12}$	$H_{2,2,1}$	$E_{2,2,3}$

Table 3.7. Description of Each Loop of the Fingernail Clipper

One way to identify the complete set of independent loops for a configuration uses the concept of a spanning tree [Seshu and Reed]. Here a spanning tree is seen as a simplified version of an annotated liaison diagram that includes only enough real mates and intra-part relationships to connect all the features but still contains no closed-loops. Once a spanning tree is identified, real mates and intra-part relationships are then added to the spanning tree one at a time. Each time a real mate or intra-part relationship is added, one or more closed-loops will be formed. Arbitrarily, one of these closed-loops is chosen and it becomes part of the complete set of independent closed-loops. Real mates and intra-part relationships are continually added to the spanning tree one at a time until the annotated liaison diagram is complete and the complete set of independent closed-loops has been identified.

Once the set of independent closed-loops has been identified, the set of independent open-loops can be added. The set of independent open-loops contains one open-loop for each feature in the annotated liaison diagram (except the base feature) that only shares one real mate or intra-part relationship with another feature. Both the open-loops and closed-loops in this complete independent set must be assigned directions. An open-loop is always assigned a direction that points away from the base feature and point towards the feature at the end of the open-loop. A closed-loop can be assigned either direction. Closed-loops are represented on an annotated liaison diagram with a directed circle and open loops are indicated by a double arrow pointing to the feature at the end of the open-loop.

Figure 3.23 shows how the spanning tree technique was used to find the loops for the fingernail clipper. The completed annotated liaison diagrams including loops are shown in Figure 3.24. A key for the labels and symbols is shown in Figure 3.25

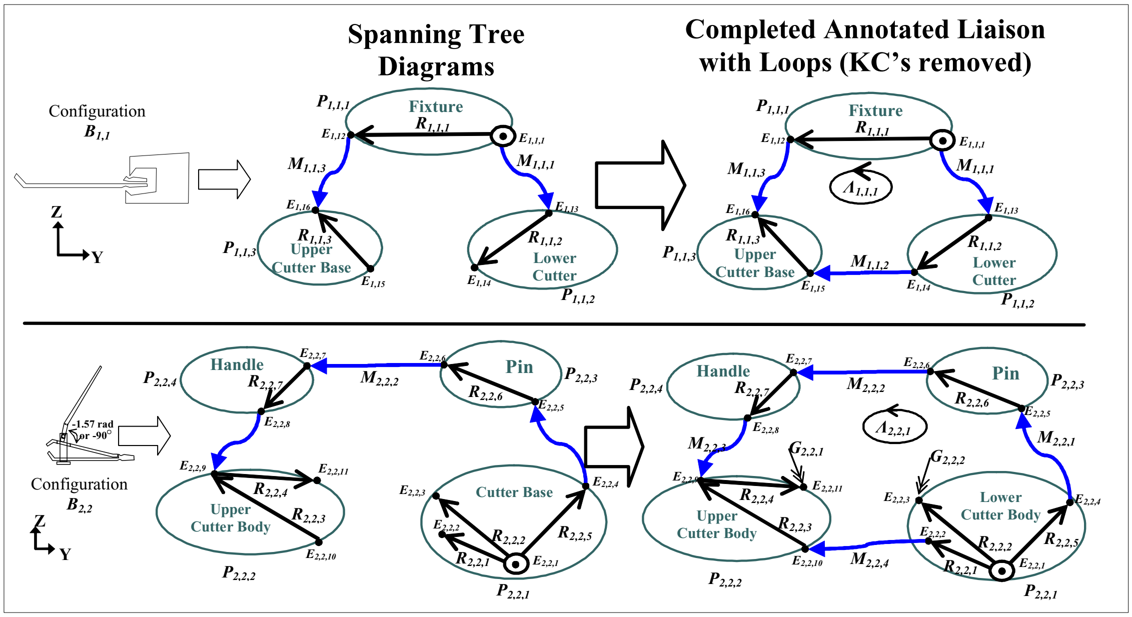


Figure 3.23. Finding the Closed-Loops of an Assembly Using Spanning Trees

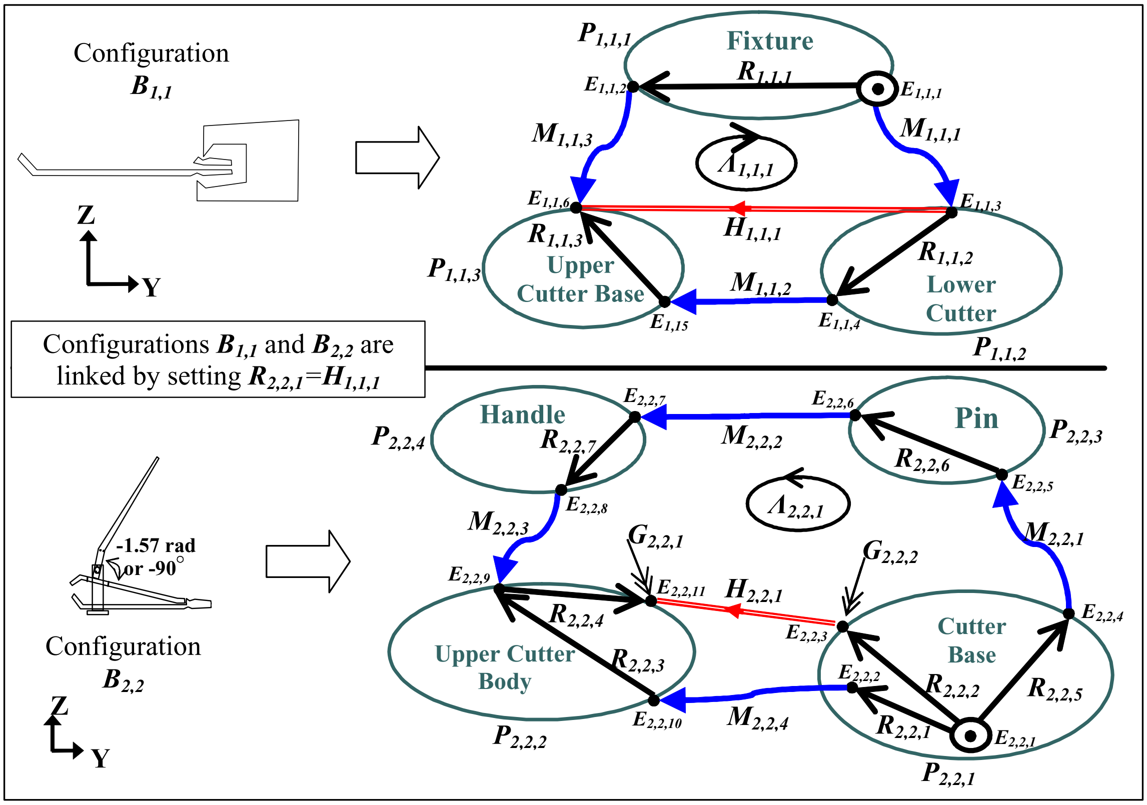


Figure 3.24. Completed Annotated Liaison Diagrams Including Loops

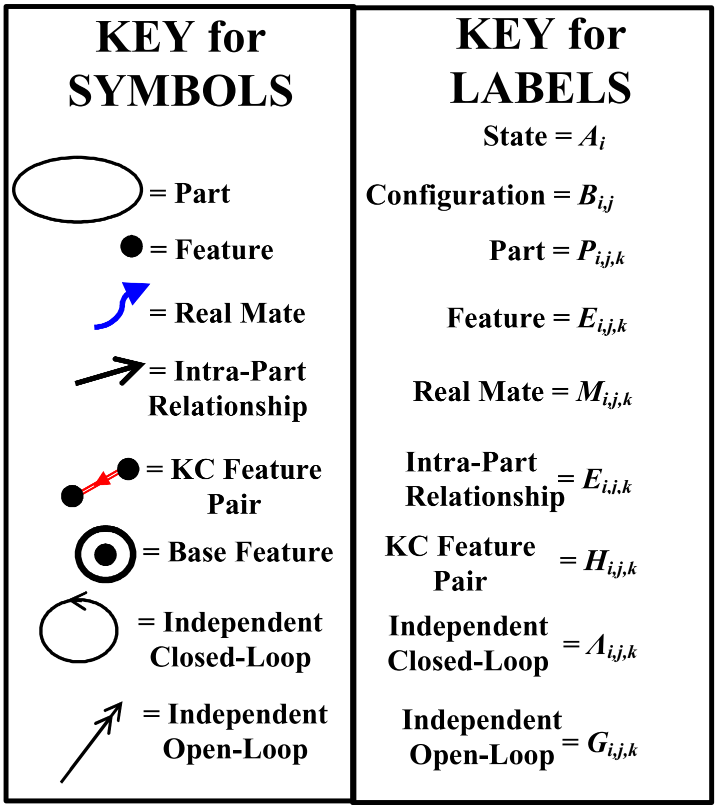

KEY for SYMBOLS
= Part
= Feature
= Real Mate
= Intra-Part Relationship
= KC Feature Pair
= Base Feature
= Independent Closed-Loop
= Independent Open-Loop
KEY for LABELS
State = A_i
Configuration = $B_{i,j}$
Part = $P_{i,j,k}$
Feature = $E_{i,j,k}$
Real Mate = $M_{i,j,k}$
Intra-Part Relationship = $E_{i,j,k}$
KC Feature Pair = $H_{i,j,k}$
Independent Closed-Loop = $\Lambda_{i,j,k}$
Independent Open-Loop = $G_{i,j,k}$

Figure 3.25. Key for Symbols and Labels in Annotated Liaison Diagram

3.1.10 Finding Chains for DOF, DOC, DOS, Dimensions, KCs, & Loops

For an arbitrary configuration, $\boldsymbol{B_{i,j}}$, each of the $\boldsymbol{\lambda_{i,j}}$ independent closed-loops can be expressed as a chain of serially-linked DOF, DOC, DOS, and dimensions that connects the features in that closed-loop. Each of the $\boldsymbol{g_{i,j}}$ independent open-loops can be expressed as a chain of serially-linked DOF, DOC, DOS, and dimensions that connects the base feature to the feature at the end of that open-loop. Each of the $\boldsymbol{kct_{i,j}}$ KCs can be expressed as a chain of serially-linked DOF, DOC, DOS, and dimensions that connects the first feature to the second feature related by the KC. Finally, the location of each of the $\boldsymbol{dt_{i,j}}$

dimensions, $\boldsymbol{ft_{i,j}}$ DOF, $\boldsymbol{ct_{i,j}}$ DOC, and $\boldsymbol{st_{i,j}}$ DOS can be expressed as a chain of serially-linked DOF, DOC, DOS, and dimensions that connects it to the base feature.

If the dimensions are numbered $\boldsymbol{1}$ through $\boldsymbol{dtt}$, the DOF are numbered $\boldsymbol{dtt+1}$ through $\boldsymbol{dtt+ftt}$, the DOS are numbered $\boldsymbol{dtt+ftt+1}$ through $\boldsymbol{dtt+ftt+stt}$, and DOC are numbered $\boldsymbol{dtt+ftt+stt+1}$ through $\boldsymbol{dtt+ftt+stt+ctt}$, then for an arbitrary configuration, $\boldsymbol{B_{i,j}}$, the set of all chains describing the global location of all the dimensions can be expressed as a matrix $\boldsymbol{[CHd]_{i,j}}$, called the dimension chain matrix. $\boldsymbol{[CHd]_{i,j}}$ has $\boldsymbol{dt_{i,j}}$ rows, one for each dimension in the configuration. In general, the number of columns in $\boldsymbol{[CHd]_{i,j}}$, will be different for each assembly.

The set of all these chains describing the global location of all the DOF, DOS, and DOC can be expressed as a matrix $\boldsymbol{[CHm]_{i,j}}$, called the mate chain matrix (the order of the rows is DOF first, then DOS, then DOC). $\boldsymbol{[CHm]_{i,j}}$ will always have $\boldsymbol{6m_{i,j}}$ rows but it can have any number of columns (just like $\boldsymbol{[CHd]_{i,j}}$).

The set of all chains describing the independent closed-loops can be expressed as a matrix $\boldsymbol{[CHc]_{i,j}}$, called the closed-loop chain matrix. $\boldsymbol{[CHc]_{i,j}}$, will always have $\boldsymbol{\lambda_{i,j}}$ rows, one for each independent closed-loop, but it can have any number of columns. The set of all chains describing the independent open-loops can be expressed as a matrix $\boldsymbol{[CHo]_{i,j}}$, called the open-loop chain matrix. $\boldsymbol{[CHo]_{i,j}}$ has $\boldsymbol{g_{i,j}}$ rows, one for each independent open-loop, and can have any number of columns. Finally, the set of chains describing the KCs can be expressed as a matrix $[\boldsymbol{CHk}]_{i,j}$, called the KC chain matrix. $[\boldsymbol{CHk}]_{i,j}$ has $\boldsymbol{kct}_{i,j}$ rows, one for each KC, and can have any number of columns.

In summary, each row in $[\boldsymbol{CHd}]_{i,j}$, $[\boldsymbol{CHm}]_{i,j}$, $[\boldsymbol{CHc}]_{i,j}$, $[\boldsymbol{CHo}]_{i,j}$, and $[\boldsymbol{CHk}]_{i,j}$, contains a set of integers that correspond to a chain of elements for the appropriate dimension, DOF, DOS, DOC, closed-loop, open-loop, or KC. The number of columns in $[\boldsymbol{CHd}]_{i,j}$, $[\boldsymbol{CHm}]_{i,j}$, $[\boldsymbol{CHc}]_{i,j}$, $[\boldsymbol{CHo}]_{i,j}$, or $[\boldsymbol{CHk}]_{i,j}$, is equal to the number of elements in the longest chain. For rows with fewer elements, the extra entries at the end should be entered as zeros. For each row, the sign of the $\boldsymbol{i}^{\text{th}}$ entry with value $\boldsymbol{j}$ is positive if the direction of the corresponding chain is the same as the direction assigned to the intra-part relationship or real mate containing the $\boldsymbol{j}^{\text{th}}$ dimension, DOF, DOS, or DOC. The sign of the $\boldsymbol{i}^{\text{th}}$ entry is negative if the direction of the corresponding chain is opposite the direction assigned to the intra-part relationship or real mate containing the $\boldsymbol{j}^{\text{th}}$ dimension, DOF, DOS, or DOC. Figure 3.26 shows in $[\boldsymbol{CHd}]_{1,1}$, $[\boldsymbol{CHm}]_{1,1}$, $[\boldsymbol{CHc}]_{1,1}$, $[\boldsymbol{CHo}]_{1,1}$, and $[\boldsymbol{CHk}]_{1,1}$, for configuration $\boldsymbol{B}_{1,1}$ of the fingernail clipper.

$$[CHd]_{1,1} = \left[\begin{array}{ccccccccccccc|c}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & D_1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & D_2 \\
1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & D_3 \\
12 & 13 & 14 & 15 & 16 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & D_4 \\
12 & 13 & 14 & 15 & 16 & 17 & 4 & 18 & 19 & 20 & 6 & 7 & 8 & D_5
\end{array}\right.$$

$$[CHm]_{1,1} = \left[\begin{array}{cccccccccccc|c}
12 & 13 & 14 & 15 & 16 & 17 & 4 & 18 & 19 & 20 & 0 & 0 & F_1 \\
12 & 13 & 14 & 15 & 16 & 17 & 4 & 18 & 19 & 20 & 6 & 0 & F_2 \\
12 & 13 & 14 & 15 & 16 & 17 & 4 & 18 & 19 & 20 & 6 & 7 & F_3 \\
1 & 2 & 3 & 21 & 22 & 23 & 0 & 0 & 0 & 0 & 0 & 0 & F_4 \\
1 & 2 & 3 & 21 & 22 & 23 & 9 & 0 & 0 & 0 & 0 & 0 & F_5 \\
1 & 2 & 3 & 21 & 22 & 23 & 9 & 10 & 0 & 0 & 0 & 0 & F_6 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & C_1 \\
12 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & C_2 \\
12 & 13 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & C_3 \\
12 & 13 & 14 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & C_4 \\
12 & 13 & 14 & 15 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & C_5 \\
12 & 13 & 14 & 15 & 16 & 0 & & 0 & 0 & 0 & 0 & 0 & C_6 \\
12 & 13 & 14 & 15 & 16 & 17 & 4 & 0 & 0 & 0 & 0 & 0 & C_7 \\
12 & 13 & 14 & 15 & 16 & 17 & 4 & 18 & 0 & 0 & 0 & 0 & C_8 \\
12 & 13 & 14 & 15 & 16 & 17 & 4 & 18 & 19 & 0 & 0 & 0 & C_9 \\
1 & 2 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & C_{10} \\
1 & 2 & 3 & 21 & 16 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & C_{11} \\
1 & 2 & 3 & 21 & 22 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & C_{12}
\end{array}\right.$$

$$[CHc]^{T}{}_{1,1} = \underset{\Lambda_{1,1,1}}{\begin{bmatrix} 12 \\ 13 \\ 14 \\ 15 \\ 16 \\ 17 \\ 4 \\ 18 \\ 19 \\ 20 \\ 6 \\ 7 \\ 8 \\ 5 \\ -11 \\ -10 \\ -9 \\ -23 \\ -22 \\ -21 \\ -3 \\ -2 \\ -1 \end{bmatrix}} \begin{array}{c} \textit{Element} \\ \textit{Numbers} \end{array} = \begin{array}{cc}
1 & D_1 \\
2 & D_2 \\
3 & D_3 \\
4 & D_4 \\
5 & D_5 \\
6 & F_1 \\
7 & F_2 \\
8 & F_3 \\
9 & F_4 \\
10 & F_5 \\
11 & F_1 \\
12 & C_1 \\
13 & C_2 \\
14 & C_3 \\
15 & C_4 \\
16 & C_5 \\
17 & C_6 \\
18 & C_7 \\
19 & C_8 \\
20 & C_9 \\
21 & C_{10} \\
22 & C_{11} \\
23 & C_{12}
\end{array}$$

Figure 3.26. Chain Matrices Configuration *B_{1,1}* of the Fingernail Clipper

Figure 3.27 shows **[*CHo*]$_{2,2}$** for configuration ***B*$_{2,2}$** of the fingernail clipper (while not shown, **[*CHd*]$_{2,2}$**, **[*CHm*]$_{2,2}$**, **[*CHc*]$_{2,2}$**, and **[*CHk*]$_{2,2}$**, for configuration ***B*$_{2,2}$** of the fingernail clipper are formed in the same way as **[*CHd*]$_{1,1}$**, **[*CHm*]$_{1,1}$**, **[*CHc*]$_{1,1}$**, and **[*CHk*]$_{1,1}$**). Notice that the last seven entries of the first row in **[*CHo*]$_{2,2}$** are not officially part of the chain of DOF, DOS, DOC, and dimensions that connect the base feature to the feature at the end of open-loop ***G*$_{2,2,1}$**. These extra entries are needed so that the DOV of virtual mate $\boldsymbol{V_{2,2,1}}$ assigned to $\boldsymbol{G_{2,2,1}}$ will be oriented in the same way as the DOV of virtual mate $\boldsymbol{V_{2,2,2}}$ assigned to $\boldsymbol{G_{2,2,2}}$. This insures that the KCs are captured properly.

$$[CHo]_{2,2}{}^{T} = \begin{array}{cc} \left[\begin{array}{cc} 1 & 5 \\ 2 & 6 \\ 3 & 7 \\ 4 & 0 \\ 34 & 0 \\ 35 & 0 \\ 36 & 0 \\ 37 & 0 \\ 38 & 0 \\ 21 & 0 \\ 8 & 0 \\ 9 & 0 \\ 10 & 0 \\ 11 & 0 \\ -11 & 0 \\ -8 & 0 \\ -21 & 0 \\ -38 & 0 \\ -37 & 0 \\ -4 & 0 \\ 7 & 0 \end{array}\right] \\ \begin{array}{cc} G_{2,2,1} & G_{2,2,2} \end{array} \end{array} \qquad [CHo_{1,1:21}]_{2,2}{}^{T} = \begin{array}{cc} \left[\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 34 \\ 35 \\ 36 \\ 37 \\ 38 \\ 21 \\ 8 \\ 9 \\ 10 \\ 11 \\ -11 \\ -8 \\ -21 \\ -38 \\ -37 \\ -4 \\ 7 \end{array}\right] & \begin{array}{c} KC_1 \\ KC_2 \\ KC_3 \\ KC_4 \\ C_{25} \\ C_{26} \\ C_{27} \\ C_{28} \\ C_{29} \\ S_1 \\ D_9 \\ D_{10} \\ D_{11} \\ D_{12} \\ D_{12} \\ D_9 \\ S_1 \\ C_{29} \\ C_{28} \\ KC_4 \\ D_8 \end{array} \\ G_{2,2,1} & \end{array} \qquad [CHo_{2,1:3}]_{2,2}{}^{T} = \begin{array}{cc} \left[\begin{array}{c} 5 \\ 6 \\ 7 \end{array}\right] & \begin{array}{c} D_6 \\ D_7 \\ D_8 \end{array} \\ G_{2,2,2} & \end{array}$$

These last entries must be added so that the directions of the DOV of the virtual mate $\boldsymbol{V_{2,2,1}}$ associated with open-loop $\boldsymbol{G_{2,2,1}}$ will be aligned with the first feature $\boldsymbol{E_{2,2,3}}$ in KC feature pair $\boldsymbol{H_{2,2,1}}$ as indicated in Table 3.7.

Figure 3.27. [*CHo*]$_{2,2}$ for Configuration *B*$_{2,2}$ of the Fingernail Clipper

3.1.11 Summary of Element Identification & Location Phase

This concludes the Element Identification & Location Phase. Table 3.8, Table 3.9, Table 3.10, Table 3.11, Table 3.12, and list a description of the variables and matrices that are generated by completing the tasks of this phase and list their values or sizes in the case of the fingernail clipper.

Variable	Description	Value
a	Total # of assembly states	**2**
bt	Total # of configurations	**5**
kctt	Total # of KC's	**11**
kcl	Total # of Linking KC's	**4**
kcp	Total # of Performance KC's	**7**
dtt	Total # of Dimensions	**15**
ftt	Total # of DOF	**12**
ctt	Total # of DOC	**29**
stt	Total # of DOS	**1**

Table 3.8. Assembly-Level Counting Variables for Fingernail Clipper

Vector or Matrix	Description	Size
[*Fd*]	DOF directions for entire assembly	**(*12* x *4*)**
[*Cd*]	DOC directions for entire assembly	**(*29* x *4*)**
[*Sd*]	DOS directions for entire assembly	**(*1* x *4*)**
[*SK*]	Nominal DOS stiffnesses for entire assembly	**(*1* x *1*)**
[*Dn*]	Nominal dimension values for entire assembly	**(*15* x *1*)**
[*Dul*]	upper limits for dimensions for entire assembly	**(*15* x *1*)**
[*Dll*]	lower limits for dimensions for entire assembly	**(*15* x *1*)**
[*Dd*]	Dimension directions for entire assembly	**(*15* x *1*)**
[*Dt*]	Distribution types for dimensions of entire assembly	**(*15* x *1*)**
[*Dp*]	Distribution parameters for dimensions of entire assembly	**(*15* x *3*)**
[*KCt*]	Performance KC types for entire assembly	**(*7* x *1*)**
[*KCd*]	Performance KC directions for entire assembly	**(*7* x *1*)**
[*KCn*]	Performance KC nominal values for entire assembly	**(*7* x *1*)**
[*KCL*]	Performance KC spec. limits for entire assembly	**(*7* x *1*)**

Table 3.9. Assembly-Level Matrices for Fingernail Clipper

Variable	Description	Value
b_{1}	Total # of configurations in state A_{1}	**1**
b_{2}	Total # of configurations in state A_{2}	**4**
$p_{1,1}$	Total # of parts in configuration $B_{1,1}$	**3**
$e_{1,1}$	Total # features in configuration $B_{1,1}$	**6**
$m_{1,1}$	Total # of mates in configuration $B_{1,1}$	**3**
$r_{1,1}$	Total # of intra-part relationships in configuration $B_{1,1}$	**3**
$h_{1,1}$	Total # of KC feature pairs in configuration $B_{1,1}$	**1**
$\lambda_{1,1}$	Total # indpendent closed-loops for configuration $B_{1,1}$	**1**
$g_{1,1}$	Total # indpendent open-loops for configuration $B_{1,1}$	**0**
$ft_{1,1}$	Total # of DOF for configuration $B_{1,1}$	**6**
$ct_{1,1}$	Total # of DOC for configuration $B_{1,1}$	**12**
$st_{1,1}$	Total # of DOS for configuration $B_{1,1}$	**0**
$dt_{1,1}$	Total # of Dimensions for configuration $B_{1,1}$	**5**
$kct_{1,1}$	Total # of KC's for configuration $B_{1,1}$	**4**
$p_{2,2}$	Total # of parts in configuration $B_{2,2}$	**4**
$e_{2,2}$	Total # features in configuration $B_{2,2}$	**11**
$m_{2,2}$	Total # of mates in configuration $B_{2,2}$	**4**
$r_{2,2}$	Total # of intra-part relationships in configuration $B_{2,2}$	**7**
$h_{2,2}$	Total # of KC feature pairs in configuration $B_{2,2}$	**1**
$\lambda_{2,2}$	Total # independent closed-loops for configuration $B_{2,2}$	**1**
$g_{2,2}$	Total # independent open-loops for configuration $B_{2,2}$	**2**
$ft_{2,2}$	Total # of DOF for configuration $B_{2,2}$	**6**
$ct_{2,2}$	Total # of DOC for configuration $B_{2,2}$	**17**
$st_{2,2}$	Total # of DOS for configuration $B_{2,2}$	**1**
$dt_{2,2}$	Total # of Dimensions for configuration $B_{2,2}$	**14**
$kct_{2,2}$	Total # of KC's for configuration $B_{2,2}$	**7**

Table 3.10. Configuration-Level Counting Variables for Fingernail Clipper

Vector or Matrix	Description	Size
$[\boldsymbol{Fn}]_{1,1}$	Guesses for Nominal DOF Values for configuration $\boldsymbol{B}_{1,1}$	(6×1)
$[\boldsymbol{Ful}]_{1,1}$	Upper Limits on DOF values for configuration $\boldsymbol{B}_{1,1}$	(6×1)
$[\boldsymbol{Fll}]_{1,1}$	Lower Limits on DOF values for configuration $\boldsymbol{B}_{1,1}$	(6×1)
$[\boldsymbol{Fd}]_{1,1}$	direcions for DOF in configuration $\boldsymbol{B}_{1,1}$	(6×4)
$[\boldsymbol{Sn}]_{1,1}$	Guesses for Nominal DOS Values for configuration $\boldsymbol{B}_{1,1}$	(0×0)
$[\boldsymbol{Sul}]_{1,1}$	Upper Limits on DOS values for configuration $\boldsymbol{B}_{1,1}$	(0×0)
$[\boldsymbol{Sll}]_{1,1}$	Lower Limits on DOS values for configuration $\boldsymbol{B}_{1,1}$	(0×0)
$[\boldsymbol{Sd}]_{1,1}$	direcions for DOS in configuration $\boldsymbol{B}_{1,1}$	(0×0)
$[\boldsymbol{SK}]_{1,1}$	Nominal DOS stiffness matrix for configuration $\boldsymbol{B}_{1,1}$	(0×0)
$[\boldsymbol{Cn}]_{1,1}$	Nominal DOC values for configuration $\boldsymbol{B}_{1,1}$	(12×1)
$[\boldsymbol{Cd}]_{1,1}$	direcions for DOC in configuration $\boldsymbol{B}_{1,1}$	(12×4)
$[\boldsymbol{Dn}]_{1,1}$	Nominal dimension values for configuration $\boldsymbol{B}_{1,1}$	(5×1)
$[\boldsymbol{Dd}]_{1,1}$	direcions for dimensions in configuration $\boldsymbol{B}_{1,1}$	(5×1)
$[\boldsymbol{KCn}]_{1,1}$	Nominal KC values for configuration $\boldsymbol{B}_{1,1}$	(4×1)
$[\boldsymbol{KCd}]_{1,1}$	KC directions for configuration $\boldsymbol{B}_{1,1}$	(4×1)
$[\boldsymbol{CHd}]_{1,1}$	Matrix of dimension chains for configuration $\boldsymbol{B}_{1,1}$	(5×13)
$[\boldsymbol{CHm}]_{1,1}$	Matrix of DOF, DOS,and DOC chains for configuration $\boldsymbol{B}_{1,1}$	(18×12)
$[\boldsymbol{CHc}]_{1,1}$	Matrix of closed-loop chains for configuration $\boldsymbol{B}_{1,1}$	(1×23)
$[\boldsymbol{CHo}]_{1,1}$	Matrix of open-loop chains for configuration $\boldsymbol{B}_{1,1}$	(0×0)
$[\boldsymbol{CHk}]_{1,1}$	Matrix of KC chains for configuration $\boldsymbol{B}_{1,1}$	(4×8)

Table 3.11. Matrices for Configuration $\boldsymbol{B}_{1,1}$ of the Fingernail Clipper

Vector or Matrix	Description	Size
$[\boldsymbol{Fn}]_{2,2}$	Guesses for Nominal DOF Values for configuration $\boldsymbol{B}_{2,2}$	$(6 \text{ x } 1)$
$[\boldsymbol{Ful}]_{2,2}$	Upper Limits on DOF values for configuration $\boldsymbol{B}_{2,2}$	$(6 \text{ x } 1)$
$[\boldsymbol{Fll}]_{2,2}$	Lower Limits on DOF values for configuration $\boldsymbol{B}_{2,2}$	$(6 \text{ x } 1)$
$[\boldsymbol{Fd}]_{2,2}$	direcions for DOF in configuration $\boldsymbol{B}_{2,2}$	$(6 \text{ x } 4)$
$[\boldsymbol{Sn}]_{2,2}$	Guesses for Nominal DOS Values for configuration $\boldsymbol{B}_{2,2}$	$(1 \text{ x } 1)$
$[\boldsymbol{Sul}]_{2,2}$	Upper Limits on DOS values for configuration $\boldsymbol{B}_{2,2}$	$(1 \text{ x } 1)$
$[\boldsymbol{Sll}]_{2,2}$	Lower Limits on DOS values for configuration $\boldsymbol{B}_{2,2}$	$(1 \text{ x } 1)$
$[\boldsymbol{Sd}]_{2,2}$	direcions for DOS in configuration $\boldsymbol{B}_{2,2}$	$(1 \text{ x } 4)$
$[\boldsymbol{SK}]_{2,2}$	Nominal DOS stiffness matrix for configuration $\boldsymbol{B}_{2,2}$	$(1 \text{ x } 1)$
$[\boldsymbol{Cn}]_{2,2}$	Nominal DOC values for configuration $\boldsymbol{B}_{2,2}$	$(17 \text{ x } 1)$
$[\boldsymbol{Cd}]_{2,2}$	direcions for DOC in configuration $\boldsymbol{B}_{2,2}$	$(17 \text{ x } 4)$
$[\boldsymbol{Dn}]_{2,2}$	Nominal dimension values for configuration $\boldsymbol{B}_{2,2}$	$(14 \text{ x } 1)$
$[\boldsymbol{Dd}]_{2,2}$	direcions for dimensions in configuration $\boldsymbol{B}_{2,2}$	$(14 \text{ x } 1)$
$[\boldsymbol{KCn}]_{2,2}$	Nominal KC values for configuration $\boldsymbol{B}_{2,2}$	$(7 \text{ x } 1)$
$[\boldsymbol{KCd}]_{2,2}$	KC directions for configuration $\boldsymbol{B}_{2,2}$	$(7 \text{ x } 1)$
$[\boldsymbol{CHd}]_{2,2}$	Matrix of dimension chains for configuration $\boldsymbol{B}_{2,2}$	$(14 \text{ x } 14)$
$[\boldsymbol{CHm}]_{2,2}$	Matrix of DOF, DOS,and DOC chains for configuration $\boldsymbol{B}_{2,2}$	$(18 \text{ x } 19)$
$[\boldsymbol{CHc}]_{2,2}$	Matrix of closed-loop chains for configuration $\boldsymbol{B}_{2,2}$	$(1 \text{ x } 32)$
$[\boldsymbol{CHo}]_{2,2}$	Matrix of open-loop chains for configuration $\boldsymbol{B}_{2,2}$	$(2 \text{ x } 21)$
$[\boldsymbol{CHk}]_{2,2}$	Matrix of KC chains for configuration $\boldsymbol{B}_{2,2}$	$(7 \text{ x } 17)$

Table 3.12. Matrices for Configuration $B_{2,2}$ of the Fingernail Clipper

3.2 Element Quantification & Integration Phase

The second phase of the modeling methodology is the Element Quantification & Integration Phase. The goal of this phase is to create analytical forms of all the elements of the assembly that are relevant to its manufacturability and combined them together into one unified model. Figure 3.28 shows a flow diagram for this phase. Each box corresponds to a task that must be completed and the arrows between the boxes indicate the order in which they should be completed. The numbers in parentheses indicate which subsection of Section 3.2 addresses the corresponding task.

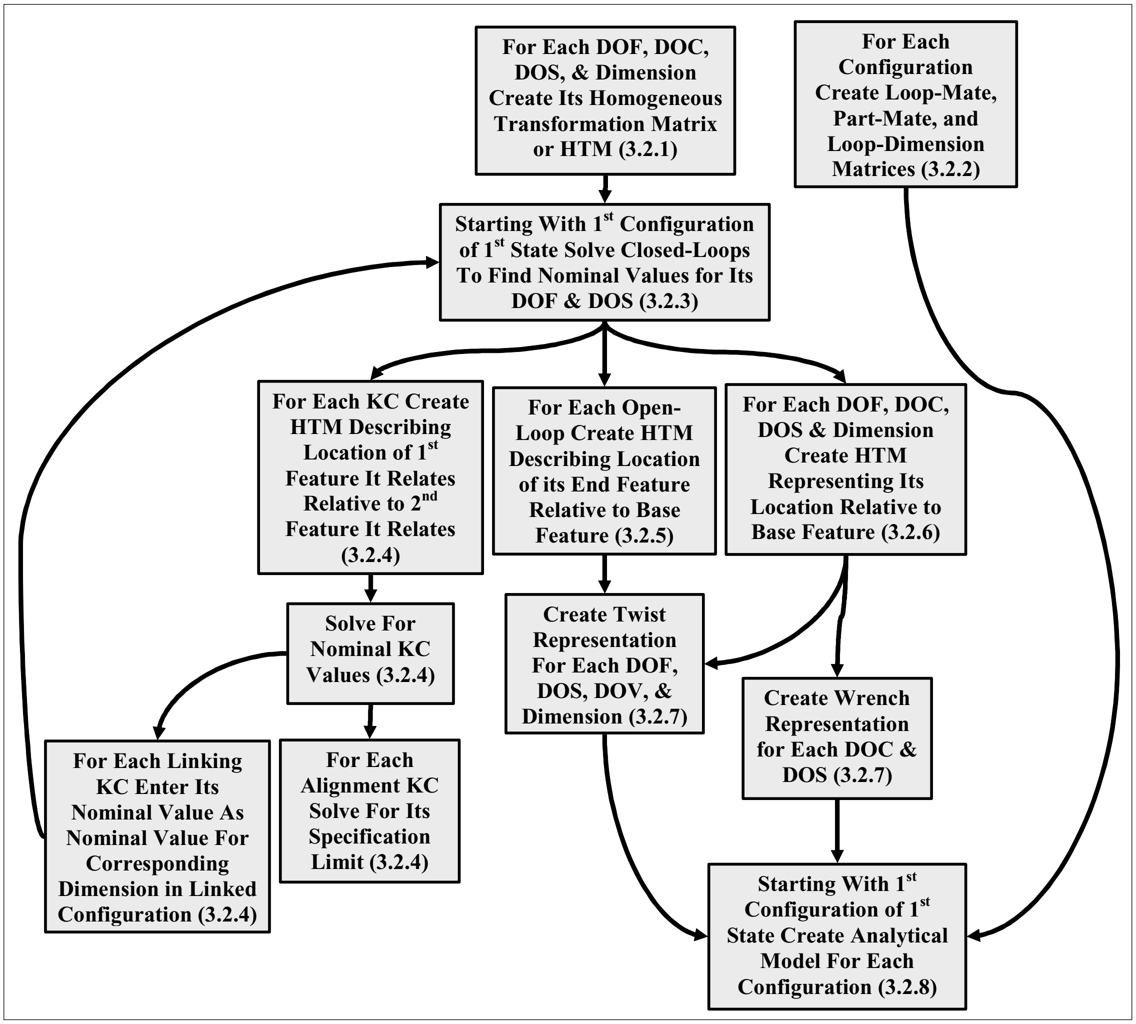

For Each DOF, DOC, DOS, & Dimension Create Its Homogeneous Transformation Matrix or HTM (3.2.1)
For Each Configuration Create Loop-Mate, Part-Mate, and Loop-Dimension Matrices (3.2.2)
Starting With 1st Configuration of 1st State Solve Closed-Loops To Find Nominal Values for Its DOF & DOS (3.2.3)
For Each KC Create HTM Describing Location of 1st Feature It Relates Relative to 2nd Feature It Relates (3.2.4)
For Each Open-Loop Create HTM Describing Location of its End Feature Relative to Base Feature (3.2.5)
For Each DOF, DOC, DOS & Dimension Create HTM Representing Its Location Relative to Base Feature (3.2.6)
Solve For Nominal KC Values (3.2.4)
Create Twist Representation For Each DOF, DOS, DOV, & Dimension (3.2.7)
Create Wrench Representation for Each DOC & DOS (3.2.7)
For Each Linking KC Enter Its Nominal Value As Nominal Value For Corresponding Dimension in Linked Configuration (3.2.4)
For Each Alignment KC Solve For Its Specification Limit (3.2.4)
Starting With 1st Configuration of 1st State Create Analytical Model For Each Configuration (3.2.8)

Figure 3.28. Element Quantification & Integration Phase

3.2.1 Creating Homogeneous Transformation Matrices

Homogeneous Transformation Matrices, or HTMs are (4×4) matrices that can be used to describe the nominal position and orientation (hereafter the combination of position and orientation will be called location) of one coordinate frame relative to another in all six coordinate directions. HTMs have been used by [Denavit and Hartenberg] to represent kinematic linkages, by [Ahuja and Coons] and others for displaying CAD geometry on a computer screen, by [Paul] and others for modeling robots, and by researchers such as [Lee and Gossard] and [Whitney] for modeling and connecting parts in assemblies.

HTMs have several useful properties. For two coordinate frames, denoted ***1*** and ***2***, the basic equation for the HTM, **[*Z12*],** that describes the location of coordinate frame ***2*** relative to coordinate frame ***1*** is as follows:

$$[Z12\ ] = \begin{bmatrix} [RR12] & [z12] \\ [0_{(1\times 3)}] & sf \end{bmatrix}$$

Equation 3.18

[*z12*] is a **(*3* x *1*)** vector that describes the position of the second coordinate frame relative to the first, **[*RR12*]** is a **(*3* x *3*)** matrix that describes the orientation of the second coordinate frame relative to the first, and ***sf*** is a scaling factor, which will always equal one in this work. Expanded, Equation 3.18 takes the following form:

$$[Z12] = \begin{bmatrix} [RR12_{1,1}] & [RR12_{1,2}] & [RR12_{1,3}] & [z12_1] \\ [RR12_{2,1}] & [RR12_{2,2}] & [RR12_{2,3}] & [z12_2] \\ [RR12_{3,1}] & [RR12_{3,2}] & [RR12_{3,3}] & [z12_3] \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Equation 3.19

[$z12_1$], **[$z12_2$]**, and **[$z12_3$]** are the positions of coordinate frame ***2*** along coordinate frame ***1***'s, ***X***- (or first), ***Y***- (or second), and ***Z-*** (or third) axes respectively. **[$RR12_{i,j}$]** is the direction cosine of the i^{th} coordinate axis in the frame ***1*** to the j^{th} coordinate axis in the frame ***2***.

HTMs have two useful properties. First, if **[*Z12*]** describes the location of frame ***2*** relative to frame ***1***, and **[*Z21*]** describes the location of frame ***1*** relative to frame ***2***, the following relationship holds:

$$[Z12] = [Z21]^{-1}$$

Equation 3.20

Second, if **[*Z12*]** describes the location of frame ***2*** relative to frame ***1***, **[*Z23*]** describes the location of frame ***3*** relative to frame ***2***, and **[*Z13*]** describes the location of frame ***3*** relative to frame ***1***, the following relationship is true:

$$[Z13] = [Z12][Z23]$$

Equation 3.21

Extending the property shown in Equation 3.21 to ***n*** coordinate frames yields the following equation:

$$[Z1n] = [Z12][Z23]\ldots[Z(n-1)n]$$

Equation 3.22

The HTM representation, denoted $\boldsymbol{[D^l]_{i,j,k}}$ (or for a linking KC $\boldsymbol{[KC^l]_{i,j,k}}$) for an arbitrary nominal dimension, $\boldsymbol{D_l}$ (or linking KC $\boldsymbol{KC_l}$) with nominal value, $\boldsymbol{[Dn_k]_{i,j}}$ (or $\boldsymbol{[KCn_k]_{i,j}}$) and direction, $\boldsymbol{[Dd_k]_{i,j}}$ (or $\boldsymbol{[KCd_k]_{i,j}}$), is formulated as follows (shown for dimensions but extends equally to linking KCs):

$$
\left[D^l\right]_{i,j,k} = \begin{cases}
\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & Q_1 & -Q_2 & 0 \\ 0 & Q_2 & Q_1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} if \left[Dd_k\right]_{i,j} = 1 &
\begin{bmatrix} 1 & 0 & 0 & \left[Dn_k\right]_{i,j} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} if \left[Dd_k\right]_{i,j} = 4 \\
\begin{bmatrix} Q_1 & 0 & Q_2 & 0 \\ 0 & 1 & 0 & 0 \\ -Q_2 & 0 & Q_1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} if \left[Dd_k\right]_{i,j} = 2 &
\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & \left[Dn_k\right]_{i,j} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} if \left[Dd_k\right]_{i,j} = 5 \\
\begin{bmatrix} Q_1 & -Q_2 & 0 & 0 \\ Q_2 & Q_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} if \left[Dd_k\right]_{i,j} = 3 &
\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \left[Dn_i\right]_{i,j} \\ 0 & 0 & 0 & 1 \end{bmatrix} if \left[Dd_k\right]_{i,j} = 6
\end{cases}
$$

$$
\begin{aligned}
& where \\
& Q_1 = \cos\left(\left[Dn_k\right]_{i,j}\right) \\
& Q_2 = \sin\left(\left[Dn_k\right]_{i,j}\right) \\
& for \\
& i = 1,2,\mathrm{K},a \\
& j = 1,2,\mathrm{K},b_a \\
& k = 1,2,\mathrm{K},dt_{i,j} \\
& l = 1,2,\mathrm{K},or\ dtt
\end{aligned}
$$

Equation 3.23

Figure 3.29 shows the HTMs for each dimension in the fingernail clipper that were obtained by entering the values from **[*Dn*]** and **[*Dd*]** (see Figure 3.14) into Equation 3.23.

$[\boldsymbol{KC^1}]$, $[\boldsymbol{KC^2}]$, $[\boldsymbol{KC^3}]$, and $[\boldsymbol{KC^4}]$ are the values of the linking KCs from $\boldsymbol{B_{1,1}}$ that become dimensions in $\boldsymbol{B_{2,2}}$ and as such they are expressed in terms of the unknown values in $[\boldsymbol{KCn}]_{1,1}$ (it will be shown how to solve for these values later).

$$[D^1],[D^2],[D^3],[D^8],[D^{12}]=[I_{(4\times4)}]$$

$$[D^4],[D^5]=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&1.36\\0&0&0&1\end{bmatrix}\quad [D^9]=\begin{bmatrix}1&0&0&0\\0&.983&.184&0\\0&-.184&.983&0\\0&0&0&1\end{bmatrix}\quad [D^{13}]=\begin{bmatrix}1&0&0&0\\0&1&0&-35.18\\0&0&1&0\\0&0&0&1\end{bmatrix}$$

$$[D^6]=\begin{bmatrix}1&0&0&0\\0&1&0&-40.34\\0&0&1&0\\0&0&0&1\end{bmatrix}\quad [D^{10}]=\begin{bmatrix}1&0&0&0\\0&1&0&-40.24\\0&0&1&0\\0&0&0&1\end{bmatrix}\quad [D^{14}]=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&8.41\\0&0&0&1\end{bmatrix}$$

$$[D^{15}]=\begin{bmatrix}1&0&0&0\\0&1&0&-2.96\\0&0&1&0\\0&0&0&1\end{bmatrix}$$

$$[D^7]=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&2.81\\0&0&0&1\end{bmatrix}\quad [D^{11}]=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&-2.81\\0&0&0&1\end{bmatrix}$$

$$[KC^1]_{2,2}=\begin{bmatrix}1&0&0&[KCn_1]_{2,2}\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}\quad [KC^2]_{2,2}=\begin{bmatrix}1&0&0&0\\0&1&0&[KCn_2]_{2,2}\\0&0&1&0\\0&0&0&1\end{bmatrix}$$

$$[KC^3]_{2,2}=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&[KCn_3]_{2,2}\\0&0&0&1\end{bmatrix}\quad [KC^4]_{2,2}=\begin{bmatrix}1&0&0&0\\0&\cos([KCn_4]_{2,2})&-\sin([KCn_4]_{2,2})&0\\0&\sin([KCn_4]_{2,2})&\cos([KCn_4]_{2,2})&0\\0&0&0&1\end{bmatrix}$$

Figure 3.29. The HTM Representations for the Fingernail Clipper Dimensions

The HTM representation, $[\boldsymbol{F^l}]_{i,j,k}$, $[\boldsymbol{C^l}]_{i,j,k}$, or $[\boldsymbol{S^l}]_{i,j,k}$ for an arbitrary DOF, $\boldsymbol{F_k}$, DOC, $\boldsymbol{C_k}$, or DOS, $\boldsymbol{S_k}$, with nominal value (or initial guess), $[\boldsymbol{Fn_k}]_{i,j}$, $[\boldsymbol{Cn_k}]_{i,j}$, or $[\boldsymbol{Sn_k}]_{i,j}$ and direction,

[*Fd*_*k,1:4*]*_i,j_*, **[*Cd*_*k,1:4*]*_i,j_***, or **[*Sd*_*k,1:4*]*_i,j_*** respectively is formulated as follows (shown only for DOF but extends equally to DOC and DOS by swapping out the appropriate variables):

$$\left[F^{l}\right]_{i,j,k} = \begin{cases} \begin{bmatrix} 1 & 0 & 0 & x\theta \\ 0 & 1 & 0 & y\theta \\ 0 & 0 & 1 & z\theta \\ 0 & 0 & 0 & 1 \end{bmatrix} & if\ v = \infty \\ \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & v\theta \\ 0 & 0 & 0 & 1 \end{bmatrix} & if\ \|z\| = 1\ \&\ v \neq \infty \\ \begin{bmatrix} \left[F^{l}_{1,1}\right] & \left[F^{l}_{1,2}\right] & \left[F^{l}_{1,3}\right] & \left[F^{l}_{1,4}\right] \\ \left[F^{l}_{2,1}\right] & \left[F^{l}_{2,2}\right] & \left[F^{l}_{2,3}\right] & \left[F^{l}_{2,4}\right] \\ \left[F^{l}_{3,1}\right] & \left[F^{l}_{3,2}\right] & \left[F^{l}_{3,3}\right] & \left[F^{l}_{3,4}\right] \\ 0 & 0 & 0 & 1 \end{bmatrix} & if\ \|z\| \neq 1\ \&\ v \neq \infty \end{cases}$$

$$\begin{aligned}
&where \\
&v = \left[Fd_{k,4}\right]_{i,j}, w = \sqrt{1-\left[Fd_{k,3}\right]_{i,j}^{2}}, x = \left[Fd_{k,1}\right]_{i,j}, y = \left[Fd_{k,2}\right]_{i,j}, z = \left[Fd_{k,3}\right]_{i,j}, \theta = \left[Fn_{k}\right]_{i,j} \\
&\left[F^{l}_{1,1}\right]_{i,j,k} = x^{2} + \left(\left(x^{2}z^{2}+y^{2}\right)\cos(\theta)\right)/w^{2} \quad , \quad \left[F^{l}_{2,1}\right]_{i,j,k} = xy + z\sin(\theta) - xy\cos(\theta) \\
&\left[F^{l}_{3,1}\right]_{i,j,k} = xz(1-\cos(\theta)) - y\sin(\theta) \quad , \quad \left[F^{l}_{1,2}\right]_{i,j,k} = -xy - z\sin(\theta) - xy\cos(\theta) \\
&\left[F^{l}_{2,2}\right]_{i,j,k} = y^{2} + \left(\left(y^{2}z^{2}+x^{2}\right)\cos(\theta)\right)/w^{2} \quad , \quad \left[F^{l}_{3,2}\right]_{i,j,k} = yz(1-\cos(\theta)) + x\sin(\theta) \\
&\left[F^{l}_{1,3}\right]_{i,j,k} = xz(1-\cos(\theta)) + y\sin(\theta) \quad , \quad \left[F^{l}_{2,3}\right]_{i,j,k} = yz(1-\cos(\theta)) - x\sin \\
&\left[F^{l}_{3,3}\right]_{i,j,k} = z^{2} + w^{2}\cos(\theta) \quad , \quad \left[F^{l}_{1,4}\right]_{i,j,k} = \theta vx \quad , \quad \left[F^{l}_{2,4}\right]_{i,j} = \theta vy \quad , \quad \left[F^{l}_{3,4}\right]_{i,j} = \theta vz \\
&for \\
&i = 1,2,\mathrm{K},a \\
&j = 1,2,\mathrm{K},b_{a} \\
&k = 1,2,\mathrm{K},ft_{i,j} \\
&l = 1,2,\mathrm{K},or\ ftt
\end{aligned}$$

Equation 3.24

Figure 3.30 shows the HTM representation for each DOF, DOC, and DOS of the fingernail clipper obtained by substituting values from **[*Fn*]*_1,1_***, **[*Fn*]*_2,2_***, **[*Fd*]*_1,1_***, **[*Fd*]*_2,2_***,

[*Cn*]*1,1*, [*Cn*]*2,2*, [*Cd*]*1,1*, [*Cd*]*2,2*, [*Sn*]*1,1*, [*Sn*]*2,2*, [*Sd*]*1,1*, [*Sd*]*2,2*, (see Figure 3.18) into Equation 3.23. Notice that the HTMs for the DOF and DOS are expressed in terms of the values in **[*Fn*]*1,1*, [*Fn*]*2,2*, [*Sn*]*1,1***, and **[*Sn*]*2,2*** since they contain only guesses and not exact values (it will be shown below how to solve for these values).

$$\left[F^{1}\right]_{1,1,1}=\begin{bmatrix}1&0&0&[Fn_1]_{1,1}\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}\qquad \left[F^{7}\right]_{2,2,1}=\begin{bmatrix}\cos\left([Fn_1]_{2,2}\right)&-\sin\left([Fn_1]_{2,2}\right)&0&0\\\sin\left([Fn_1]_{2,2}\right)&\cos\left([Fn_1]_{2,2}\right)&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}$$

$$\left[F^{2}\right]_{1,1,2}=\begin{bmatrix}1&0&0&0\\0&1&0&[Fn_2]_{1,1}\\0&0&1&0\\0&0&0&1\end{bmatrix}\qquad \left[F^{8}\right]_{2,2,2}=\begin{bmatrix}1&0&0&0\\0&\cos\left([Fn_2]_{2,2}\right)&-\sin\left([Fn_2]_{2,2}\right)&0\\0&\sin\left([Fn_2]_{2,2}\right)&\cos\left([Fn_2]_{2,2}\right)&0\\0&0&0&1\end{bmatrix}$$

$$\left[F^{3}\right]_{1,1,3}=\begin{bmatrix}\cos\left([Fn_3]_{1,1}\right)&-\sin\left([Fn_3]_{1,1}\right)&0&0\\\sin\left([Fn_3]_{1,1}\right)&\cos\left([Fn_3]_{1,1}\right)&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}\qquad \left[F^{9}\right]_{2,2,3}=\begin{bmatrix}1&0&0&[Fn_3]_{2,2}\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}$$

$$\left[F^{4}\right]_{1,1,4}=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&[Fn_4]_{1,1}\\0&0&0&1\end{bmatrix}\qquad \left[F^{10}\right]_{2,2,4}=\begin{bmatrix}1&0&0&0\\0&\cos\left([Fn_4]_{2,2}\right)&-\sin\left([Fn_4]_{2,2}\right)&0\\0&\sin\left([Fn_4]_{2,2}\right)&\cos\left([Fn_4]_{2,2}\right)&0\\0&0&0&1\end{bmatrix}$$

$$\left[F^{5}\right]_{1,1,5}=\begin{bmatrix}1&0&0&0\\0&\cos\left([Fn_5]_{1,1}\right)&-\sin\left([Fn_5]_{1,1}\right)&0\\0&\sin\left([Fn_5]_{1,1}\right)&\cos\left([Fn_5]_{1,1}\right)&0\\0&0&0&1\end{bmatrix}\qquad \left[F^{11}\right]_{2,2,5}=\begin{bmatrix}\cos\left([Fn_5]_{2,2}\right)&-\sin\left([Fn_5]_{2,2}\right)&0&0\\\sin\left([Fn_5]_{2,2}\right)&\cos\left([Fn_5]_{2,2}\right)&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}$$

$$\left[F^{6}\right]_{1,1,6}=\begin{bmatrix}\cos\left([Fn_6]_{1,1}\right)&0&\sin\left([Fn_6]_{1,1}\right)&0\\0&1&0&0\\-\sin\left([Fn_6]_{1,1}\right)&0&\cos\left([Fn_6]_{1,1}\right)&0\\0&0&0&1\end{bmatrix}\qquad \left[F^{12}\right]_{2,2,6}=\begin{bmatrix}1&0&0&0\\0&1&0&[Fn_6]_{2,2}\\0&0&1&0\\0&0&0&1\end{bmatrix}$$

$$\left[S^{1}\right]_{2,2,1}=\begin{bmatrix}1&0&0&0\\0&\cos([Sn_1])&-\sin([Sn_1])&0\\0&\sin([Sn_1])&\cos([Sn_1])&0\\0&0&0&1\end{bmatrix}\qquad \left[C^{l}\right]_{i,j,k}=\left[I_{(4\times4)}\right]\begin{cases}for\ i,j=1,1\ or\ 2,2\\for\ k=1,2,\ldots,12\ or\ 1,2,\ldots,17\\for\ l=1,2,\ldots,or\ 29\end{cases}$$

Figure 3.30. The HTMs for DOF, DOC, & DOS of the Fingernail Clipper

3.2.2 Create Part-Mate, Loop-Mate, & Loop-Dimension Matrices

Network Theory (see [Seshu and Reed], [Strang], and [Davies, 1981]) can be used to analytically relate the various elements in an assembly to each other. In Network Theory, graphs containing arcs (in this case the arcs are the intra-part relationships, the mates, and the DOF, DOC, DOS, DOV and dimensions that comprise them) and nodes (in this case the nodes are the parts and features) like an annotated liaison diagram, can be converted into two-dimensional matrices with the columns representing arcs and the rows representing either the loops or the nodes. Specifically, there are three such matrices that will prove useful in this work: the part-mate matrix[5], $\boldsymbol{[PM]_{i,j}}$ the loop-mate matrix, $\boldsymbol{[LM]_{i,j}}$ and the loop-dimension matrix $\boldsymbol{[LD]_{i,j}}$.

For an arbitrary configuration, $\boldsymbol{B_{i,j}}$, a $\boldsymbol{(\{p_{i,j} -1\} \text{ x } \{ct_{i,j} + st_{i,j}\})}$ matrix, $\boldsymbol{[PM]_{i,j}}$, called the part-mate matrix, can be formed using the annotated liaison diagram. Entry $\boldsymbol{[PM_{k,l}]_{i,j}}$ is a "+1" if the head of the arc representing the mate containing the $\boldsymbol{l}^{\text{th}}$ DOC or DOS (DOS should always go after the DOC) for configuration $\boldsymbol{B_{i,j}}$ touches the $\boldsymbol{j}^{\text{th}}$ part (excluding the base part) and a "-1" if the tail touches the $\boldsymbol{j}^{\text{th}}$ part. Entry $\boldsymbol{[PM_{k,l}]_{i,j}}$ is a "0" if the arc representing the mate containing the $\boldsymbol{l}^{\text{th}}$ DOC or DOS for configuration $\boldsymbol{B_{i,j}}$ does not touch the $\boldsymbol{j}^{\text{th}}$ part. Figure 3.31 shows $\boldsymbol{[PM]_{1,1}}$ and $\boldsymbol{[PM]_{2,2}}$ for the fingernail clipper.

[5] This matrix is also called the incidence matrix in the literature [Seshu and Reed].

$$[PM]_{1,1} = \begin{array}{cccccccccccc|c}
 & & \overbrace{\qquad\qquad}^{M_{1,1}} & & & & & \overbrace{\qquad}^{M_{1,2}} & & & \overbrace{\qquad}^{M_{1,3}} & & \\
C_1 & C_2 & C_3 & C_4 & C_5 & C_6 & C_7 & C_8 & C_9 & C_{10} & C_{11} & C_{12} & \\
1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & 0 & 0 & 0 & P_{1,1,2} \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & P_{1,1,3}
\end{array}$$

$$[PM]_{2,2} = \begin{array}{cccccccccccccccccc|c}
 & & \overbrace{\qquad\qquad}^{M_{2,1}} & & & & & \overbrace{\qquad\qquad}^{M_{2,2}} & & & \overbrace{\qquad}^{M_{2,3}} & & & & \overbrace{\qquad\qquad}^{M_{2,4}} & & & & \\
C_{13} & C_{14} & C_{15} & C_{16} & C_{17} & C_{18} & C_{19} & C_{20} & C_{21} & C_{22} & C_{23} & C_{24} & C_{25} & C_{26} & C_{27} & C_{28} & C_{29} & S_1 & \\
1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & P_{2,2,3} \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & P_{2,2,4} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & P_{2,2,2}
\end{array}$$

Figure 3.31. The Part-Mate Matrices for the Fingernail Clipper Example

For an arbitrary configuration, $\boldsymbol{B_{i,j}}$, a **($\{\lambda_{i,j} + g_{i,j}\}$ x $\{ft_{i,j} + st_{i,j} + 6g_{i,j}\}$)** matrix **$[LM]_{i,j}$**, called the loop-mate matrix, can be formed using the annotated liaison diagram. Entry **$[LM_{k,l}]_{i,j}$** is a "+1" if the arc representing the mate containing the $\boldsymbol{l}^{\text{th}}$ DOF, DOS or DOV (the order should be DOF, then DOS, then DOV) for configuration $\boldsymbol{B_{i,j}}$ is in the $\boldsymbol{j}^{\text{th}}$ independent loop (open or closed) and points in the same direction as the sense of the loop and a "-1" if it is in the $\boldsymbol{j}^{\text{th}}$ independent loop but points in the opposite direction as the sense of the loop. Entry **$[LM_{k,l}]_{i,j}$** is a "0" if the mate containing the $\boldsymbol{l}^{\text{th}}$ DOF, DOS, or DOV is not part of the $\boldsymbol{j}^{\text{th}}$ independent loop. Remember that each open-loop has one virtual mate containing six DOV that connect the base feature to the feature at the end of an open-loop and point away from the base feature. DOV can never be included in a closed-loop and may only be part of a single open-loop. If included in an open-loop, the corresponding entry will always be a "-1" since DOV point toward the feature at the end of an open-loop and it is via the DOV that one returns from the end feature to the base feature.

For an arbitrary configuration, $\boldsymbol{B_{i,j}}$, a ($\{\boldsymbol{\lambda_{i,j}} + \boldsymbol{g}_{\boldsymbol{i,j}}\}$ **x** $\boldsymbol{dt_{i,j}}$) matrix $\boldsymbol{[LD]_{i,j}}$, called the loop-dimension matrix, can be formed using the annotated liaison diagram. Entry $\boldsymbol{[LD_{k,l}]_{i,j}}$ is a "+1" if the arc representing the intra-part relationship containing the $\boldsymbol{l}^{\text{th}}$ dimension for configuration $\boldsymbol{B_{i,j}}$ is in the $\boldsymbol{j}^{\text{th}}$ independent loop (open or closed) and points in the same direction as the sense of the loop and a "-1" if it is in the $\boldsymbol{j}^{\text{th}}$ independent loop but points in the opposite direction as the sense of the loop. Entry $\boldsymbol{[LD_{k,l}]_{i,j}}$ is a "0" if the intra-part relationship containing the $\boldsymbol{l}^{\text{th}}$ dimension is not part of the $\boldsymbol{j}^{\text{th}}$ independent loop. Figure 3.32 shows $\boldsymbol{[LM]_{1,1}}$, $\boldsymbol{[LM]_{2,2}}$, $\boldsymbol{[LD]_{1,1}}$, and $\boldsymbol{[LD]_{2,2}}$ for fingernail clipper configurations $\boldsymbol{B_{1,1}}$ and $\boldsymbol{B_{2,2}}$.

$$[LM]_{1,1} = \begin{array}{ccccccc} \multicolumn{3}{c}{\overbrace{\hphantom{F_1\quad F_2\quad F_3}}^{M_{1,2}}} & \multicolumn{3}{c}{\overbrace{\hphantom{F_4\quad F_5\quad F_6}}^{M_{1,3}}} & \\ F_1 & F_2 & F_3 & F_4 & F_5 & F_6 & \\ 1 & 1 & 1 & -1 & -1 & -1 & \Lambda_{1,1,1} \end{array}$$

$$[LD]_{1,1} = \begin{array}{cccccc} \multicolumn{3}{c}{\overbrace{\hphantom{D_1\quad D_2\quad D_3}}^{R_{1,1}}} & \overbrace{\hphantom{D_4}}^{R_{1,2}} & \overbrace{\hphantom{D_5}}^{R_{1,3}} & \\ D_1 & D_2 & D_3 & D_4 & D_5 & \\ -1 & -1 & -1 & 1 & 1 & \Lambda_{1,1,1} \end{array}$$

$$[LM]_{2,2} = \begin{array}{cccccccccccccccccccc} \overbrace{\hphantom{F_7}}^{M_{2,1}} & \overbrace{\hphantom{F_8}}^{M_{2,2}} & \multicolumn{4}{c}{\overbrace{\hphantom{F_9\quad F_{10}\quad F_{11}\quad F_{12}}}^{M_{2,3}}} & \overbrace{\hphantom{S_1}}^{M_{2,4}} & \multicolumn{6}{c}{\overbrace{\hphantom{V_1\quad V_2\quad V_3\quad V_4\quad V_5\quad V_6}}^{G_{2,1}}} & \multicolumn{6}{c}{\overbrace{\hphantom{V_7\quad V_8\quad V_9\quad V_{10}\quad V_{11}\quad V_{12}}}^{G_{2,1}}} & \\ F_7 & F_8 & F_9 & F_{10} & F_{11} & F_{12} & S_1 & V_1 & V_2 & V_3 & V_4 & V_5 & V_6 & V_7 & V_8 & V_9 & V_{10} & V_{11} & V_{12} & \\ 1 & 1 & 1 & 1 & 1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \Lambda_{2,2,1} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & G_{2,2,1} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 & -1 & -1 & G_{2,2,2} \end{array}$$

$$[LD]_{2,2} = \begin{array}{ccccccccccccccc} \multicolumn{4}{c}{\overbrace{\hphantom{KC_1\quad KC_2\quad KC_3\quad KC_4}}^{R_{2,1}}} & \multicolumn{3}{c}{\overbrace{\hphantom{D_6\quad D_7\quad D_8}}^{R_{2,2}}} & \multicolumn{3}{c}{\overbrace{\hphantom{D_{10}\quad D_{11}\quad D_{12}}}^{R_{2,4}}} & \overbrace{\hphantom{D_{13}}}^{R_{2,5}} & \overbrace{\hphantom{D_{14}}}^{R_{2,6}} & \overbrace{\hphantom{D_{15}}}^{R_{2,7}} & \overbrace{\hphantom{D_9}}^{R_{2,3}} & \\ KC_1 & KC_2 & KC_3 & KC_4 & D_6 & D_7 & D_8 & D_{10} & D_{11} & D_{12} & D_{13} & D_{14} & D_{15} & D_9 & \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & \Lambda_{2,2,1} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & G_{2,2,1} \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & G_{2,2,2} \end{array}$$

Figure 3.32. The Loop-Mate and Loop-Dimension Matrix the Fingernail Clipper

3.2.3 Solving the Closed-Loops

An arbitrary closed-loop, $\boldsymbol{\Lambda}_{i,j,k}$ can be expressed as an HTM, $[\boldsymbol{\Lambda}]_{i,j,k}$, that is formed by multiplying in series, using Equation 3.20 and Equation 3.22, all the HTMs for the DOF, DOC, DOS, and dimensions that comprise the loop. The order of multiplication for the HTMs of the DOF, DOC, DOS, and dimensions and can be obtained by looking at the appropriate row in $[\boldsymbol{CHc}]_{i,j}$. Figure 3.33 shows the chain of matrices that is multiplied to form $[\boldsymbol{\Lambda}]_{1,1,1}$ for the fingernail clipper (see Figure 3.26, Figure 3.29, and Figure 3.30 for $[\boldsymbol{CHc}]_{1,1}$, and the HTMs for the DOF, DOC, DOS, and dimensions). While not shown, $[\boldsymbol{\Lambda}]_{2,2,1}$ is obtained in the same manner. Notice that $[\boldsymbol{\Lambda}]_{1,1,1}$ cannot be computed because the exact nominal values of the DOF in the chain remain unknown.

$$\begin{aligned}
&[\Lambda]_{1,1,1} = \left[C^1\right]_{1,1,1}\left[C^2\right]_{1,1,2}\left[C^3\right]_{1,1,3}\left[C^4\right]_{1,1,4}\left[C^5\right]_{1,1,5}\left[C^6\right]_{1,1,6}\\
&\left[D^4\right]_{1,1,4}\left[F^1\right]_{1,1,1}\left[F^2\right]_{1,1,2}\left[F^3\right]_{1,1,3}\left[C^7\right]_{1,1,7}\left[C^8\right]_{1,1,8}\left[C^9\right]_{1,1,9}\\
&\left[D^5\right]_{1,1,5}\left[F^4\right]_{1,1,4}^{-1}\left[F^5\right]_{1,1,5}^{-1}\left[F^6\right]_{1,1,6}^{-1}\left[C^{10}\right]_{1,1,10}^{-1}\left[C^{11}\right]_{1,1,11}^{-1}\\
&\left[C^{12}\right]_{1,1,12}^{-1}\left[D^3\right]_{1,1,3}^{-1}\left[D^2\right]_{1,1,2}^{-1}\left[D^1\right]_{1,1,1}^{-1} = ?
\end{aligned}$$

Figure 3.33. Series of HTMs That One Multiplies To Get $[\boldsymbol{\Lambda}]_{1,1,1}$

For an arbitrary configuration, $\boldsymbol{B}_{i,j}$, $[\boldsymbol{Fn}]_{i,j}$ and $[\boldsymbol{Sn}]_{i,j}$ can be obtained exactly by solving the following optimization problem:

$$\begin{aligned}
&\textit{Find } [Sn]_{i,j} \textit{ and / or } [Fn]_{i,j} \textit{ that minimizes } \beta_{i,j}\\
&\textit{where}\\
&\beta_{i,j} = \frac{1}{2}\left([Sn]_{i,j}{}^{T} [SK]_{i,j} [Sn]_{i,j} \right)\\
&\textit{subject to}\\
&[\Lambda_{l,n}]_{i,j,k} = [I_{(4\times4)}]\\
&[Fll_l]_{i,j} \le [Fn_l]_{i,j} \le [Ful_l]_{i,j}\\
&[Sll_n]_{i,j} \le [Sn_n]_{i,j} \le [Sul_n]_{i,j}\\
&\textit{for}\\
&i = 1,2,\ldots,a\\
&j = 1,2,\ldots,b_i\\
&k = 1,2,\ldots,\lambda_{i,j}\\
&l = 1,2,\ldots,ft_{i,j}\\
&n = 1,2,\ldots,st_{i,j}
\end{aligned}$$

Equation 3.25

Equation 3.25 must be solved for the configurations of an earlier state before moving onto the configurations of the next state. This is because some of the dimensions in a configuration from a latter state will depend on the linking KCs from a configuration in a previous state. $\boldsymbol{\beta_{i,j}}$ in Equation 3.25 is the amount of potential stored energy in the compliant mates of configuration $\boldsymbol{B_{i,j}}$. $\boldsymbol{\beta_{i,j}}^*$ is defined as the minimum stored energy for that configuration, and this occurs when the configuration is in static equilibrium. The constraint (**[*Λ*]*i,j,k* =[*I(4x4)*]**) imposes the condition that the net distance traveled around the closed-loop in any direction is zero. The bounds on **[*Fn*]*i,j*** and **[*Sn*]*i,j*** insure that the nominal values of the DOF do not exceed their allowable limits (e.g. most hinges due to practical considerations do not allow 360 degrees of rotation). **[*Fn**]*i,j*** and **[*Sn**]*i,j*** are defined as the set of nominal values for the DOF and DOS respectively that minimize the stored energy in configuration $\boldsymbol{B_{i,j}}$.

If an assembly configuration is under-constrained, then there will be more than one $[\boldsymbol{Fn}^*]_{i,j}$ and/or $[\boldsymbol{Sn}^*]_{i,j}$ that result in a minimum $\boldsymbol{\beta_{i,j}}^*$. If this is the case, then one or more of the nominal values of the DOF and/or DOS should be fixed temporarily so that a unique $[\boldsymbol{Fn}^*]_{i,j}$ and/or $[\boldsymbol{Sn}^*]_{i,j}$ can be found for each configuration.

If the assembly is over-constrained, then it is possible that there may not exist a $[\boldsymbol{Fn}^*]_{i,j}$ and/or $[\boldsymbol{Sn}^*]_{i,j}$ such that the constraint in the optimization problem above is satisfied ($[\boldsymbol{\Lambda}]_{2,2,1} = [\boldsymbol{I}_{(4\mathrm{x}4)}]$. If this is true, then the dimensions and mates have been specified in the design such that the parts in their nominal states cannot be assembled so that all the closed-loops actually close. If the assembly is designed to have preload in its nominal state (i.e. locked in strain energy as in the case of some ball bearings due to over-constraint) then some of the entities should be modeled as multiple parts connected with compliant mates using a lumped-parameter approach [Slocum]. Figure 3.34 shows $\boldsymbol{\beta_{1,1}}^*$, $[\boldsymbol{Fn}^*]_{1,1}$, and $[\boldsymbol{Sn}^*]_{1,1}$ for configuration $\boldsymbol{B_{1,1}}$ of the fingernail clipper obtained by solving Equation 3.25. Since there were no compliant mates, the stored energy is automatically zero for $\boldsymbol{B_{1,1}}$.

$$\beta_{1,1}^{\ *} = 0$$

$$\left[Fn^{*}\right]_{1,1}^{\ T} = \begin{array}{cccccc} F_1 & F_2 & F_3 & F_4 & F_5 & F_6 \\ \left[0\right. & 0 & 0 & 2.72 & 0 & \left.0\right] \end{array}$$

$$\left[Sn^{*}\right]_{1,1} = [\emptyset]$$

Figure 3.34. Solved Nominal DOF & DOS Values for Configuration *B₁,₁*

3.2.4 Find HTMs and Nominal Values for KCs

An arbitrary KC, $\boldsymbol{KC_{i,j,k}}$, can be expressed as an HTM, $[\boldsymbol{kc}]_{i,j,k}$, that is formed by multiplying in series, using Equation 3.20 and Equation 3.22, all the HTMs for the DOF, DOC, DOS, and dimensions that connect the two features being related by the KC (starting with the first feature). The order of multiplication for the HTMs of the DOF, DOC, DOS, and dimensions and can be obtained by looking at the appropriate row in $[\boldsymbol{CHk}]_{i,j}$. Figure 3.35 shows the chain of matrices that is multiplied to form $[\boldsymbol{kc}]_{1,1,1}$ for the fingernail clipper (see Figure 3.26, Figure 3.29, and Figure 3.30 for $[\boldsymbol{CHk}]_{1,1}$, and the HTMs for the DOF, DOC, DOS, and dimensions). While not shown, $[\boldsymbol{kc}]_{1,1,2}$ through $[\boldsymbol{kc}]_{1,1,4}$ and $[\boldsymbol{kc}]_{2,2,1}$ through $[\boldsymbol{kc}]_{2,2,7}$ are obtained in the same manner.

$$[kc]_{1,1,1}=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&1.36\\0&0&0&1\end{bmatrix}\begin{bmatrix}1&0&0&[Fn^*{}_1]_{1,1}\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}\begin{bmatrix}1&0&0&0\\0&1&0&[Fn^*{}_2]_{1,1}\\0&0&1&0\\0&0&0&1\end{bmatrix}$$

$$\begin{bmatrix}\cos\left([Fn^*{}_3]_{1,1}\right)&-\sin\left([Fn^*{}_3]_{1,1}\right)&0&0\\\sin\left([Fn^*{}_3]_{1,1}\right)&\cos\left([Fn^*{}_3]_{1,1}\right)&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}\left[I_{(4\times4)}\right]\left[I_{(4\times4)}\right]\left[I_{(4\times4)}\right]\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&1.36\\0&0&0&1\end{bmatrix}=$$

$$[kc]_{1,1,1}=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&1.36\\0&0&0&1\end{bmatrix}\left[I_{(4\times4)}\right]\left[I_{(4\times4)}\right]\left[I_{(4\times4)}\right]\left[I_{(4\times4)}\right]\left[I_{(4\times4)}\right]\left[I_{(4\times4)}\right]\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&1.36\\0&0&0&1\end{bmatrix}=$$

$$[kc]_{1,1,1}=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&2.72\\0&0&0&1\end{bmatrix}$$

Figure 3.35. Series of HTMs That One Multiplies To Get [*H*]*1,1,1*

The nominal value, **[*KCn$_l$*]*$_{i,j}$***, of the ***l***$^{\text{th}}$ KC in configuration ***B$_{i,j}$***, can be found by entering its corresponding direction, **[*KCd$_l$*]*$_{i,j}$*** and the corresponding HTM **[*kc*]*$_{i,j,k}$*** into the following equation:

$$\left[KCn_l\right]_{i,j} = \begin{cases} \operatorname{atan} 2\left(\left[kc_{3,2}\right]_{i,j,k}, \operatorname{sign}\left(\left[kc_{1,1}\right]_{i,j,k}\right)\left[kc_{2,2}\right]_{i,j,k} \right) & \mathit{if}\ \left[KCd_l\right]_{i,j} = 1 \\ \operatorname{atan} 2\left(\left[kc_{1,3}\right]_{i,j,k}, \operatorname{sign}\left(\left[kc_{2,2}\right]_{i,j,k}\right)\left[kc_{3,3}\right]_{i,j,k} \right) & \mathit{if}\ \left[KCd_l\right]_{i,j} = 2 \\ \operatorname{atan} 2\left(\left[kc_{2,1}\right]_{i,j,k}, \operatorname{sign}\left(\left[kc_{3,3}\right]_{i,j,k}\right)\left[kc_{1,1}\right]_{i,j,k} \right) & \mathit{if}\ \left[KCd_l\right]_{i,j} = 3 \\ \left[kc_{1,4}\right]_{i,j,k} \quad \mathit{if}\ \left[KCd_l\right]_{i,j} = 4 \\ \left[kc_{2,4}\right]_{i,j,k} \quad \mathit{if}\ \left[KCd_l\right]_{i,j} = 5 \\ \left[kc_{3,4}\right]_{i,j,k} \quad \mathit{if}\ \left[KCd_l\right]_{i,j} = 6 \end{cases}$$

where

$$\mathit{sign}(\mathit{input1}) = \begin{cases} 1\ \mathit{for\ input1} > 0 \\ 0\ \mathit{for\ input1} = 0 \\ -1\ \mathit{for\ input1} < 0 \end{cases}$$

$$\operatorname{atan} 2\ (\mathit{input2}, \mathit{input3}) = \begin{cases} \tan^{-1}\left(\dfrac{\mathit{input2}}{\mathit{input3}}\right) \mathit{for\ input3} > 0 \\ \mathit{sign}(\mathit{input2})\pi + \tan^{-1}\left(\dfrac{\mathit{input2}}{\mathit{input3}}\right) \mathit{for\ input3} < 0 \\ \dfrac{\mathit{sign}(\mathit{input2})\pi}{2} \mathit{for\ input3} = 0 \end{cases}$$

for

$i = 1,2,\ldots,a$

$j = 1,2,\ldots,b_i$

$k = 1,2,\ldots,\mathit{or}\ h_{i,j}$

$l = 1,2,\ldots,kct_{i,j}$

Equation 3.26

Figure 3.36 shows $[\boldsymbol{KCn}]_{1,1}$, $\boldsymbol{\beta}_{2,2}{}^{*}$, $[\boldsymbol{Fn}^{*}]_{2,2}$, $[\boldsymbol{Sn}^{*}]_{2,2}$, $[\boldsymbol{kc}]_{2,2,1}$ through $[\boldsymbol{kc}]_{2,2,7}$, $[\boldsymbol{KCn}]_{2,2}$, $[\boldsymbol{KCn}]$**,** and $[\boldsymbol{KCL}]$, for the fingernail clipper. $[\boldsymbol{KCn}]_{1,1}$ was obtained by entering the appropriate values from $[\boldsymbol{KCd}]_{1,1}$ (see Figure 3.10), and $[\boldsymbol{kc}]_{1,1,1}$ through $[\boldsymbol{kc}]_{1,1,4}$ (see Figure 3.35) into Equation 3.26. The values in $[\boldsymbol{KCn}]_{1,1}$ (all linking KC values) were then used as dimensions in $\boldsymbol{B}_{2,2}$ so that $\boldsymbol{\beta}_{2,2}{}^{*}$, $[\boldsymbol{Fn}^{*}]_{2,2}$, $[\boldsymbol{Sn}^{*}]_{2,2}$, $[\boldsymbol{\Lambda}]_{2,2,1}$ (not shown), and $[\boldsymbol{kc}]_{2,2,1}$ through $[\boldsymbol{kc}]_{2,2,7}$ could be solved for. $[\boldsymbol{KCd}]_{1,1}$ (see Figure 3.10) and $[\boldsymbol{kc}]_{2,2,1}$ through $[\boldsymbol{kc}]_{2,2,7}$ were used to solve for $[\boldsymbol{KCn}]_{2,2}$, $[\boldsymbol{KCn}]$, and $[\boldsymbol{KCL}]$.

$$\beta_{2,2}{}^{*} = 145.26$$

$$\left[Fn^{*}\right]_{2,2}{}^{T} = \begin{array}{cccccc} F_7 & F_8 & F_9 & F_{10} & F_{11} & F_{12} \\ \hline 0 & 1.49 & 0 & -1.57 & 0 & 35.51 \end{array}$$

$$\left[Sn^{*}\right]_{2,2} = \begin{array}{c} S_1 \\ \left[0.1078\right] \end{array}$$

$$\left[KCn\right]_{1,1}{}^{T} = \begin{array}{cccc} KC_1 & KC_2 & KC_3 & KC_4 \\ \hline 2.72 & 0 & 0 & 0 \end{array}$$

$$\left[kc\right]_{2,2,1}\ \mathit{through}\ \left[kc\right]_{2,2,7} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & .997 & .0771 & 0 \\ 0 & -.0771 & -.997 & 0.214 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\left[KCn\right]_{2,2}{}^{T} = \left[KCn\right]^{T} = \begin{array}{ccccccc} KC_5 & KC_6 & KC_7 & KC_8 & KC_9 & KC_{10} & KC_{11} \\ \hline 0 & 0 & .003 & .003 & 0 & 0 & .214 \end{array}$$

$$\left[KCL\right]^{T} = \begin{array}{ccccccc} .1 & -.1 & .1 & -.1 & .01 & -.01 & 0 \\ \hline KC_5 & KC_6 & KC_7 & KC_8 & KC_9 & KC_{10} & KC_{11} \end{array}$$

Figure 3.36. Solved Nominal DOF & DOS Values for Configuration *B*$_{2,2}$

Because $\boldsymbol{B_{2,2}}$ is under-constrained, the nominal value of $\boldsymbol{F_{10}}$, the rotation angle between the handle fulcrum and the top surface of the upper clipper was set about the ***X***-axis of the handle, was set at -1.57 radians (this is the point at which the cutters come closest to touching and this what distinguishes $\boldsymbol{B_{2,2}}$ from the other configurations in state $\boldsymbol{A_{2}}$) so that $[\boldsymbol{Fn^{*}}]_{2,2}$ and $[\boldsymbol{Sn^{*}}]_{2,2}$ are unique. $\boldsymbol{\beta_{2,2}}^{*}$ is positive because the nominal value of the compliant mate, $\boldsymbol{M_{2,2,4}}$, is nonzero. This stored energy is what causes the handle to recoil after it is depressed in an effort to clip a fingernail. In other words, in the absence of an external force, $\boldsymbol{B_{2,4}}$ will always revert back to $\boldsymbol{B_{2,3}}$ (see Figure 3.4).

3.2.5 Finding HTMs for Open-Loops

An arbitrary open-loop, $\boldsymbol{G_{i,j,k}}$ can be expressed as an HTM, $[\boldsymbol{G}]_{i,j,k}$, that is formed by multiplying in series, using Equation 3.20 and Equation 3.22, all the HTMs for the DOF, DOC, DOS, and dimensions that connect the base feature to the feature at the end of the open-loop. The order of multiplication for the HTMs of the DOF, DOC, DOS, and dimensions and can be obtained by looking at the appropriate row in $[\boldsymbol{CHo}]_{i,j}$. Figure 3.37 shows the chain of matrices that is multiplied to form $[\boldsymbol{G}]_{2,2,1}$ and $[\boldsymbol{G}]_{2,2,2}$ for the fingernail clipper ($[\boldsymbol{CHo}]_{2,2}$ is not shown but Figure 3.29,and Figure 3.30 show the HTMs for the DOF, DOC, DOS, and dimensions in the chains).

$$[G]_{2,2,1} = \left[KC^{1}\right]_{2,2,1}\left[KC^{2}\right]_{2,2,2}\left[KC^{3}\right]_{2,2,3}\left[KC^{4}\right]_{2,2,4}\left[C^{25}\right]_{2,2,13}\left[C^{26}\right]_{2,2,14}\left[C^{27}\right]_{2,2,15}$$

$$\left[C^{28}\right]_{2,2,16}\left[C^{29}\right]_{2,2,17}\left[S^{1}\right]_{2,2,1}\left[D^{9}\right]_{2,2,8}\left[D^{10}\right]_{2,2,9}\left[D^{11}\right]_{2,2,10}\left[D^{12}\right]_{2,2,11}\left[D^{12}\right]_{2,2,11}^{-1}$$

$$\left[D^{9}\right]_{2,2,8}^{-1}\left[S^{1}\right]_{2,2,1}^{-1}\left[C^{28}\right]_{2,2,16}^{-1}\left[C^{29}\right]_{2,2,17}^{-1}\left[KC^{4}\right]_{2,2,4}^{-1}\left[D^{8}\right]_{2,2,7} =$$

$$[G]_{2,2,1} = \left[I_{(4\times4)}\right]\left[I_{(4\times4)}\right]\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&2.72\\0&0&0&1\end{bmatrix}\left[I_{(4\times4)}\right]\left[I_{(4\times4)}\right]\left[I_{(4\times4)}\right]\left[I_{(4\times4)}\right]\left[I_{(4\times4)}\right]\left[I_{(4\times4)}\right]\begin{bmatrix}1&0&0&0\\0&.994&-.108&0\\0&.108&.994&0\\0&0&0&1\end{bmatrix}$$

$$\begin{bmatrix}1&0&0&0\\0&.993&.184&0\\0&-.184&.993&0\\0&0&0&1\end{bmatrix}\begin{bmatrix}1&0&0&0\\0&1&0&-40.24\\0&0&1&0\\0&0&0&1\end{bmatrix}\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&-2.81\\0&0&0&1\end{bmatrix}\left[I_{(4\times4)}\right]\left[I_{(4\times4)}\right]\begin{bmatrix}1&0&0&0\\0&.993&-.184&0\\0&.184&.993&0\\0&0&0&1\end{bmatrix}$$

$$\begin{bmatrix}1&0&0&0\\0&.994&.108&0\\0&-.108&.994&0\\0&0&0&1\end{bmatrix}\left[I_{(4\times4)}\right]\left[I_{(4\times4)}\right]\left[I_{(4\times4)}\right]\left[I_{(4\times4)}\right]=$$

$$[G]_{2,2,1} = \begin{bmatrix}1&0&0&0\\0&1&0&-40.24\\0&0&1&3.03\\0&0&0&1\end{bmatrix}$$

$$[G]_{2,2,2} = \left[D^{6}\right]_{2,2,5}\left[D^{7}\right]_{2,2,6}\left[D^{8}\right]_{2,2,7} = \begin{bmatrix}1&0&0&0\\0&1&0&-40.34\\0&0&1&0\\0&0&0&1\end{bmatrix}\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&2.81\\0&0&0&1\end{bmatrix}\left[I_{(4\times4)}\right]=$$

$$[G]_{2,2,2} = \begin{bmatrix}1&0&0&0\\0&1&0&-40.34\\0&0&1&2.81\\0&0&0&1\end{bmatrix}$$

Figure 3.37. Obtaining [*G*]$_{2,2,1}$ and [*G*]$_{2,2,2}$ for the Fingernail Clipper

3.2.6 Finding HTMs for Location of DOS, DOC, DOS, and Dimensions

For an arbitrary DOF, DOC, or DOS in configuration $\boldsymbol{B_{i,j}}$, its location relative to the base feature can be expressed as an HTM, $[\boldsymbol{f}^{\boldsymbol{l}}]_{i,j,k}$, $[\boldsymbol{c}^{\,\boldsymbol{l}}]_{i,j,k}$, or $[\boldsymbol{s}^{\,\boldsymbol{l}}]_{i,j,k}$, respectively that is formed by multiplying in series, using Equation 3.20 and Equation 3.22, all the HTMs for the

DOF, DOC, DOS, and dimensions that connect the base feature to the DOF, DOC, or DOS. The order of multiplication for the HTMs of the DOF, DOC, DOS, and dimensions and can be obtained by looking at the appropriate row in $[\boldsymbol{CHm}]_{i,j}$. Figure 3.38 shows the chain of matrices that is multiplied to obtain $[\boldsymbol{f}^{8}]_{2,2,2}$, the location of the revolute joint between the handle and the pin in $\boldsymbol{B}_{2,2}$ ($[\boldsymbol{CHm}]_{2,2}$ is not shown but Figure 3.29, and Figure 3.30 show the HTMs for the DOF, DOC, DOS, and dimensions in the chains). The HTMs representing the global locations of the other DOF, DOC, and DOS are obtained in the same way but are not shown here.

$$\begin{aligned}\left[f^{8}\right]_{2,2,2} = {} & \left[D^{13}\right]_{2,2,12}\left[C^{13}\right]_{2,2,1}\left[C^{14}\right]_{2,2,2}\left[C^{15}\right]_{2,2,3}\left[C^{16}\right]_{2,2,4}\left[C^{17}\right]_{2,2,5} \\ & \left[F^{7}\right]_{2,2,1}\left[D^{14}\right]_{2,2,13}\left[C^{18}\right]_{2,2,6}\left[C^{19}\right]_{2,2,7}\left[C^{20}\right]_{2,2,8}\left[C^{21}\right]_{2,2,9}\left[C^{22}\right]_{2,2,10} =\end{aligned}$$

$$\left[f^{8}\right]_{2,2,2} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -35.18 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\left[I_{(4\times4)}\right]\left[I_{(4\times4)}\right]\left[I_{(4\times4)}\right]\left[I_{(4\times4)}\right]\left[I_{(4\times4)}\right]$$

$$\left[I_{(4\times4)}\right]\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 8.41 \\ 0 & 0 & 0 & 1 \end{bmatrix}\left[I_{(4\times4)}\right]\left[I_{(4\times4)}\right]\left[I_{(4\times4)}\right]\left[I_{(4\times4)}\right]\left[I_{(4\times4)}\right] =$$

$$\left[f^{8}\right]_{2,2,2} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -35.18 \\ 0 & 0 & 1 & 8.41 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Figure 3.38. Example of Obtaining the HTM for the Location of a DOF

For an arbitrary dimension (or linking KC) in configuration, $\boldsymbol{B_{i,j}}$, its location relative to the base feature can be expressed as an HTM, $\boldsymbol{[d^{l}]_{i,j,k}}$ (or $\boldsymbol{[kc^{l}]_{i,j,k}}$) that is formed by multiplying in series, using Equation 3.20 and Equation 3.22, all the HTMs for the DOF, DOC, DOS, and dimensions that connect the base feature to the dimension. The order of multiplication for the HTMs of the DOF, DOC, DOS, and dimensions and can be obtained by looking at the appropriate row in $\boldsymbol{[CHd]_{i,j}}$. Figure 3.39 shows the chain of matrices that is multiplied to obtain $\boldsymbol{[d^{11}]_{2,2,10}}$, the location of $\boldsymbol{D_{11}}$, the height of the upper cutter blade in the upper cutter body in the $\boldsymbol{Z}$-direction in $\boldsymbol{B_{2,2}}$ ($\boldsymbol{[CHd]_{2,2}}$ is not shown but Figure 3.29, and Figure 3.30 show the HTMs for the DOF, DOC, DOS, and dimensions in the chains). The HTMs representing the locations of the other dimensions relative to the base feature are obtained in the same way but are not shown here.

$$[d^{11}]_{2,2,10} = [KC^{1}]_{2,2,1}[KC^{2}]_{2,2,2}[KC^{3}]_{2,2,3}[KC^{4}]_{2,2,4}[C^{25}]_{2,2,13}$$

$$[C^{26}]_{2,2,14}[C^{27}]_{2,2,15}[C^{28}]_{2,2,16}[C^{29}]_{2,2,17}[S^{1}]_{2,2,1}[D^{9}]_{2,2,8}[D^{10}]_{2,2,9} =$$

$$[d^{11}]_{2,2,10} = [I_{(4\times4)}][I_{(4\times4)}]\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 2.72\\ 0 & 0 & 0 & 1\end{bmatrix}[I_{(4\times4)}][I_{(4\times4)}][I_{(4\times4)}][I_{(4\times4)}][I_{(4\times4)}][I_{(4\times4)}]$$

$$\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & .994 & -.108 & 0\\ 0 & .108 & .994 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & .993 & .184 & 0\\ 0 & -.184 & .993 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & -40.24\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix} =$$

$$[d^{11}]_{2,2,10} = \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & .997 & .0771 & -40.24\\ 0 & -.0771 & .997 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

Figure 3.39. Example of Obtaining the HTM for the Location of a Dimension

3.2.7 Finding Twists & Wrenches For DOF, DOC, DOS, DOV & Dimensions

Screw Theory can be used to analytically represent the direction and magnitude of the relative displacement allowed between two parts by a DOF, DOS, or DOV, the direction and magnitude of the force transmitted between two parts by a DOC or DOS, and the magnitude and direction of variability of a dimension. Screw Theory is a six-dimensional vector algebra that can be used for describing velocities and forces originally presented by [Ball]. Its many uses and the properties of its elements have since been explored in detail by [Baker], [Davies and Primrose], [Hunt], [Phillips, 1984 and 1990], [Roth], [Waldron] and others. The two primarily elements of Screw Theory are the **twist**, a (***1***x***6***) vector denoted generically in this work as **[*t_i*]**, and the **wrench**, a (***1*** x ***6***) vector denoted generically in this work as **[*w_i*]**. The base forms of a twist (in axis formation) and the base form of a wrench (in ray formation) are expressed as follows:

$$[t_i] = [\omega_x \quad \omega_y \quad \omega_z \quad v_x \quad v_y \quad v_z]$$

Equation 3.27

$$[w_i] = [\varsigma_x \quad \varsigma_y \quad \varsigma_z \quad \tau_x \quad \tau_y \quad \tau_z]$$

Equation 3.28

A twist describes the instantaneous velocity (linear or angular) of a rigid body and is reference frame dependent. A wrench describes the instantaneous force (linear or angular) acting on a rigid body and is likewise reference frame dependent. Each twist (or wrench) also has an associated line vector, called its instantaneous spin axis (ISA), and an associated pitch, which can be zero, any real positive number, or infinite. Twists (or wrenches) with zero pitch describe pure linear velocities (or linear forces) along their respective ISAs. Twists (or wrenches) with infinite pitch describe pure angular velocities (or angular forces, also called moments) along their ISAs. Twists (or wrenches) with finite positive pitch describe velocities (or forces) that are helical in the sense that they have both a linear component along and angular component about their ISAs. As a special case, the ISA of twist with infinite pitch (or a wrench with zero pitch) is not a single vector, but an infinite set of parallel vectors.

The first three entries of a twist, $\boldsymbol{\omega_x}$, $\boldsymbol{\omega_y}$, and $\boldsymbol{\omega_z}$ are the instantaneous components, about the reference frame's $\boldsymbol{X}$-,$\boldsymbol{Y}$-, and $\boldsymbol{Z}$-axes respectively, of a rigid body's angular velocity about the ISA of the twist. The last three entries of a twist, $\boldsymbol{v_x}$, $\boldsymbol{v_y}$, and $\boldsymbol{v_z}$ are the instantaneous components, along the reference frame's $\boldsymbol{X}$-,$\boldsymbol{Y}$-, and $\boldsymbol{Z}$-axes respectively, of the linear velocity of a point that is rotating about the ISA of the twist and is instantaneously situated at the origin of the reference frame. If the twist has infinite pitch, then the first three entries are all zeros and the last three are simply the components of the linear velocity along the reference frame's $\boldsymbol{X}$-,$\boldsymbol{Y}$-, and $\boldsymbol{Z}$-axes respectively.

The last three entries of a wrench, $\boldsymbol{\tau_x}$, $\boldsymbol{\tau_y}$, and $\boldsymbol{\tau_z}$ are the instantaneous components, along the reference frame's $\boldsymbol{X}$-,$\boldsymbol{Y}$-, and $\boldsymbol{Z}$-axes respectively, of the force acting on a rigid body along the ISA of the wrench. The last first three of a wrench, ς_x, ς_y, and ς_z are the instantaneous components, along the reference frame's $\boldsymbol{X}$-,$\boldsymbol{Y}$-, and $\boldsymbol{Z}$-axes respectively, of the angular force (also called a moment) that would be applied at a point instantaneously situated at the origin of the reference frame by the force acting along the ISA of the wrench. If the wrench has zero pitch, then the last three entries are all zeros and the first three are simply the components of the moment about the reference frame's $\boldsymbol{X}$-,$\boldsymbol{Y}$-, and $\boldsymbol{Z}$-axes respectively.

Each DOF of a real mate can be represented by a unit twist and each DOC of a mate by a unit wrench. The DOS of a mate must be represented by both a unit twist and a unit wrench. The unit twist representation, $[\boldsymbol{T}^{l}]_{i,j,k}$, for the $\boldsymbol{k}$th DOF or DOS in configuration $\boldsymbol{B}_{i,j}$ whose direction and pitch is contained in $[\boldsymbol{Fd}_{k,1:4}]$ or $[\boldsymbol{Sd}_{k,1:4}]$, and whose global location is described by the HTM, $[\boldsymbol{f}^{l}]_{i,j,k}$ or $[\boldsymbol{s}^{l}]_{i,j,k}$ respectively, is computed as follows (adapted from [Whitney]):

$$
\begin{aligned}
&[T^l]_{i,j,k} = [\,[\omega] \quad [v]\,] \\
&\textit{where} \\
&[q] = [f^l]_{i,j,k} \ \textit{or}\ [s^l]_{i,j,k}
\end{aligned}
$$

$$
\left.
\begin{aligned}
&[\omega] = [Fd_{k,1:3}]_{i,j}\,[q_{1:3,1:3}]^{\,T} \\
&[u] = [q_{1:3,4}]^T \times [\omega] \\
&[v] = \begin{bmatrix} \left([u_1] + [Fd_{k,4}]_{i,j}\,[\omega_1]\right) \\ \left([u_2] + [Fd_{k,4}]_{i,j}\,[\omega_2]\right) \\ \left([u_3] + [Fd_{k,4}]_{i,j}\,[\omega_3]\right) \end{bmatrix}^T
\end{aligned}
\right\} \textit{if}\ 0 \le [Fd_{k,4}]_{i,j} < \infty
$$

$$
\left.
\begin{aligned}
&[\omega] = [0 \quad 0 \quad 0] \\
&[v] = [Fd_{k,1:3}]_{i,j}\,[q_{1:3,1:3}]^{\,T}
\end{aligned}
\right\} \textit{if}\ [Fd_{l,4}]_{i,j} = \infty
$$

$$
\begin{aligned}
&\textit{for} \\
&i = 1,2,\ldots,a \\
&j = 1,2,\ldots,b_i \\
&k = 1,2,\ldots,ft_{i,j} + st_{i,j} \\
&l = 1,2,\ldots,\textit{or ftt or}\ l = 1,2,\ldots,\textit{or stt}
\end{aligned}
$$

Equation 3.29

The unit wrench representation, $[W^l]_{i,j,k}$, for the kth DOC or DOS in configuration $B_{i,j}$ whose direction and pitch is contained in $[Cd_{k,1:4}]$ or $[Sd_{k,1:4}]$, and whose global location is described by the HTM, $[c^l]_{i,j,k}$ or $[s^l]_{i,j,k}$ respectively, is computed as follows:

$$\left[W^{l}\right]_{i,j,k}=\left[\,[\varsigma]\quad[\tau]\,\right]$$

$$\begin{array}{l}
where\\
[q]=\left[c^{l}\right]_{i,j,k}\ or\ \left[s^{l}\right]_{i,j,k}
\end{array}$$

$$\left.\begin{array}{l}
[\tau]=\left[Cd_{l,1:3}\right]_{i,j}\left[q_{1:3,1:3}\right]^{T}\\[2ex]
[\upsilon]=\left[q_{1:3,4}\right]^{T}\times[\tau]\\[2ex]
[\varsigma]=\begin{bmatrix}\left([\upsilon_1]+[\tau_1]\Big/\left[Cd_{k,4}\right]_{i,j}\right)\\ \left([\upsilon_2]+[\tau_3]\Big/\left[Cd_{k,4}\right]_{i,j}\right)\\ \left([\upsilon_3]+[\tau_2]\Big/\left[Cd_{k,4}\right]_{i,j}\right)\end{bmatrix}^{T}
\end{array}\right\}if\ 0<\left[Cd_{k,4}\right]_{i,j}<\infty$$

$$and$$

$$\left.\begin{array}{l}
[\tau]=\left[Cd_{l,1:3}\right]_{i,j}\left[q_{1:3,1:3}\right]^{T}\\[2ex]
[\varsigma]=\left[q_{1:3,4}\right]^{T}\times[\tau]
\end{array}\right\}if\ \left[Cd_{k,4}\right]_{i,j}=\infty$$

$$and$$

$$\left.\begin{array}{l}
[\tau]=\begin{bmatrix}0&0&0\end{bmatrix}\\[2ex]
[\varsigma]=\left[Cd_{k,1:3}\right]_{i,j}\left[q_{1:3,1:3}\right]^{T}
\end{array}\right\}if\ \left[Cd_{k,4}\right]_{i,j}=0$$

$$\begin{array}{l}
for\\
i=1,2,\mathrm{K},a\\
j=1,2,\mathrm{K},b_i\\
k=1,2,\mathrm{K},ct_{i,j}+st_{i,j}\\
l=1,2,\mathrm{K},or\ ctt\ or\ l=1,2,\mathrm{K},or\ stt
\end{array}$$

Equation 3.30

The **(*6* x *6*)** unit twist representation, **[*V*]**$_{i,j,k}$, for the virtual mate assigned to open-loop, ***G***$_{i,j,k}$, whose global location is described by the HTM, **[*G*]**$_{i,j,k}$ is computed as follows:

$$[V]_{i,j,k} = \begin{bmatrix} [G_{1:3,1:3}]^{T}_{i,j,k} & [Q]^{T} \\ [0_{(3x3)}] & [G_{1:3,1:3}]^{T}_{i,j,k} \end{bmatrix}$$

where

$$[Q] = \begin{bmatrix} [G_{1:3,1}]^{T}_{i,j,k} \times [G_{1:3,4}]^{T}_{i,j,k} \\ [G_{1:3,2}]^{T}_{i,j,k} \times [G_{1:3,4}]^{T}_{i,j,k} \\ [G_{1:3,3}]^{T}_{i,j,k} \times [G_{1:3,4}]^{T}_{i,j,k} \end{bmatrix}$$

for

$i = 1,2,\ldots,a$

$j = 1,2,\ldots,b_i$

$k = 1,2,\ldots,g_{i,j}$

Equation 3.31

The set of all unit twists representing the nominal DOF, DOS, and DOV for an arbitrary configuration, ***B***$_{i,j}$, can be contained in a ({ ***ft***$_{i,j}$ + ***st***$_{i,j}$ + ***6g***$_{i,j}$ } x ***6***) matrix, **[*TM*]**$_{i,j}$, and the set of all unit wrenches representing the nominal DOC and DOS for that configuration can be contained in a ({ ***ct***$_{i,j}$ + ***st***$_{i,j}$ } x ***6***) matrix, **[*WM*]**$_{i,j}$. **[*TM*]**$_{i,j}$ and **[*WM*]**$_{i,j}$ are created by simply stacking the unit twists and unit wrenches on top of each other using the same order as the columns in **[*LM*]**$_{i,j}$ and **[*PM*]**$_{i,j}$ respectively (see Figure 3.32 and Figure 3.31) [Davies, 2000].

Figure 3.40 shows **[*TM*]**$_{1,1}$ and **[*TM*]**$_{2,2}$ and Figure 3.41 shows **[*WM*]**$_{1,1}$ and **[*WM*]**$_{2,2}$ for the fingernail clipper example. The individual unit wrenches and unit twists were obtained by using Equation 3.29, Equation 3.30, and Equation 3.31. The HTM representations for the global locations of the DOF, DOC, and DOS were obtained as described above (see Figure 3.38 for an example) and **[*G*]*2,2,1*** and **[*G*]*2,2,2***, were taken Figure 3.37 above. The directions and pitches of the DOF, DOC, and DOS were obtained from Figure 3.18.

$$[TM]_{1,1} = \begin{array}{c} \left[\begin{array}{cccccc} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 2.72 & 0 \\ 0 & 1 & 0 & -2.72 & 0 & 0 \end{array}\right] \\ \begin{array}{cccccc} \theta X & \theta Y & \theta Z & X & Y & Z \end{array} \end{array} \begin{array}{l} \left.\begin{array}{l} F_1 \\ F_2 \\ F_3 \end{array}\right\} M_{1,1,2} \\ \left.\begin{array}{l} F_4 \\ F_5 \\ F_6 \end{array}\right\} M_{1,1,3} \\ \end{array}$$

$$[TM]_{2,2} = \begin{array}{c} \left[\begin{array}{cccccc} 0 & 0 & 1 & -35.18 & 0 & -0 \\ 1 & 0 & 0 & 0 & 8.41 & 35.18 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 5.46 & 35.41 \\ 0 & .0771 & .997 & -35.72 & 0 & 0 \\ 0 & 0 & 0 & 0 & .997 & -.0771 \\ 1 & 0 & 0 & 0 & 2.72 & 0 \\ 1 & 0 & 0 & 0 & 3.02 & 40.34 \\ 0 & 1 & 0 & -3.02 & 0 & 0 \\ 0 & 0 & 1 & -40.34 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 2.81 & 40.34 \\ 0 & 1 & 0 & -2.81 & 0 & 0 \\ 0 & 0 & 1 & -40.34 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right] \\ \begin{array}{cccccc} \theta X & \theta Y & \theta Z & X & Y & Z \end{array} \end{array} \begin{array}{l} F_7 \} M_{2,2,1} \\ F_8 \} M_{2,2,2} \\ \left.\begin{array}{l} F_9 \\ F_{10} \\ F_{11} \\ F_{12} \end{array}\right\} M_{2,2,3} \\ S_1 \} M_{2,2,4} \\ \left.\begin{array}{l} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \end{array}\right\} G_{2,2,1} \\ \left.\begin{array}{l} V_7 \\ V_8 \\ V_9 \\ V_{10} \\ V_{11} \\ V_{12} \end{array}\right\} G_{2,2,2} \\ \end{array}$$

Figure 3.40. [*TM*]*1,1* and [*TM*]*2,2* for the Fingernail Clipper Assembly

$$[WM]_{1,1} = \begin{array}{l} \left[\begin{array}{cccccc} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 2.72 & 0 & 1 & 0 & 0 \\ -2.72 & 0 & 0 & 0 & 1 & 0 \end{array}\right] \begin{array}{l} \left.\begin{array}{l} C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ C_6 \end{array}\right\} M_{1,1,1} \\ \left.\begin{array}{l} C_7 \\ C_8 \\ C_9 \end{array}\right\} M_{1,1,2} \\ \left.\begin{array}{l} C_{10} \\ C_{11} \\ C_{12} \end{array}\right\} M_{1,1,3} \end{array} \\ \begin{array}{cccccc} \theta X & \theta Y & \theta Z & X & Y & Z \end{array} \end{array}$$

$$[WM]_{2,2} = \begin{array}{l} \left[\begin{array}{cccccc} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 35.18 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ -35.18 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 8.41 & 35.18 & 1 & 0 & 0 \\ -8.41 & 0 & 0 & 0 & 1 & 0 \\ -35.18 & 0 & 0 & 0 & 0 & 1 \\ 0 & .997 & -.0771 & 0 & 0 & 0 \\ -35.51 & 0 & 0 & 0 & .0771 & .997 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 2.72 & 0 & 1 & 0 & 0 \\ -2.72 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \begin{array}{l} \left.\begin{array}{l} C_{13} \\ C_{14} \\ C_{15} \\ C_{16} \\ C_{17} \end{array}\right\} M_{2,2,1} \\ \left.\begin{array}{l} C_{18} \\ C_{19} \\ C_{20} \\ C_{21} \\ C_{22} \end{array}\right\} M_{2,2,2} \\ \left.\begin{array}{l} C_{23} \\ C_{24} \end{array}\right\} M_{2,2,3} \\ \left.\begin{array}{l} C_{25} \\ C_{26} \\ C_{27} \\ C_{28} \\ C_{29} \\ S_1 \end{array}\right\} M_{2,2,4} \end{array} \\ \begin{array}{cccccc} \theta X & \theta Y & \theta Z & X & Y & Z \end{array} \end{array}$$

Figure 3.41. [*WM*]$_{1,1}$ and [*WM*]$_{2,2}$ for the Fingernail Clipper Assembly

The unit twist representation, **[*U*^{l}]$_{i,j,k}$**, for the variability in the ***k***$^{\text{th}}$ dimension (or linking KC) in configuration ***B*$_{i,j}$** whose direction and pitch is contained in contained in **[*Dd*$_{k,1:4}$]$_{i,j}$** and whose global location is described by the HTM, **[*d*^{l}]$_{i,j,k}$** is computed as follows (shown for dimensions bust extends equally to linking KC's by substituting the appropriate variable):

$$
\left[U^{l}\right]_{i,j,k} = \begin{cases}
\left[\left[d^{l}{}_{1:3,1}\right]_{i,j,k}^{T} \quad Q_1 \right] \; if \left[Dd_k\right]_{i,j} = 1 \\
\left[\left[d^{l}{}_{1:3,1}\right]_{i,j,k}^{T} \quad Q_2 \right] \; if \left[Dd_k\right]_{i,j} = 2 \\
\left[\left[d^{l}{}_{1:3,1}\right]_{i,j,k}^{T} \quad Q_3 \right] \; if \left[Dd_k\right]_{i,j} = 3 \\
\left[\left[0_{(1\times3)}\right] \quad \left[d^{l}{}_{1:3,1}\right]_{i,j,k}^{T} \right] \; if \left[Dd_k\right]_{i,j} = 4 \\
\left[\left[0_{(1\times3)}\right] \quad \left[d^{l}{}_{1:3,2}\right]_{i,j,k}^{T} \right] \; if \left[Dd_k\right]_{i,j} = 5 \\
\left[\left[0_{(1\times3)}\right] \quad \left[d^{l}{}_{1:3,3}\right]_{i,j,k}^{T} \right] \; if \left[Dd_k\right]_{i,j} = 6
\end{cases}
$$

$$
\begin{aligned}
&where \\
&Q_1 = \left[d^{l}{}_{1:3,4}\right]_{i,j,k}^{T} \times \left[d^{l}{}_{1:3,1}\right]_{i,j,k}^{T} \\
&Q_2 = \left[d^{l}{}_{1:3,4}\right]_{i,j,k}^{T} \times \left[d^{l}{}_{1:3,2}\right]_{i,j,k}^{T} \\
&Q_3 = \left[d^{l}{}_{1:3,4}\right]_{i,j,k}^{T} \times \left[d^{l}{}_{1:3,3}\right]_{i,j,k}^{T} \\
&for \\
&i = 1,2,\ldots,a \\
&j = 1,2,\ldots,b_i \\
&k = 1,2,\ldots,dt_{i,j} \\
&l = 1,2,\ldots,dtt
\end{aligned}
$$

Equation 3.32

The set of all unit twists representing the variability in the dimensions (or linking KCs) for an arbitrary configuration, ***B***$\boldsymbol{B_{i,j}}$, can be contained in a ($\boldsymbol{dt_{i,j}}$ x ***6***) matrix, $\boldsymbol{[UD]_{i,j}}$, called the dimensional uncertainty matrix. $\boldsymbol{[UD]_{i,j}}$ is created by simply stacking the unit twists on top of each other using the same order as the columns in $\boldsymbol{[LD]_{i,j}}$ (see Figure 3.32) [Gao, Chase, and Magleby].

Figure 3.42 shows **[*UD*]*1,1*** and **[*UD*]*2,2*** for the fingernail clipper example. The individual unit twists were obtained using Equation 3.32. The HTM representations for the global locations of the dimensions were obtained as described above (see Figure 3.39 for an example) and the directions of the dimensions were obtained from Figure 3.10 and Figure 3.14. Notice that some of the rows in **[*UD*]*2,2*** correspond to the linking KCs from configuration ***B1,1*** (these rows should always come first). In this way the two assembly states are linked together. In general this insures that dimensional variability of a former state propagates into the later state.

$$[UD]_{1,1} = \begin{array}{c} \left[\begin{array}{cccccc} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 2.72 & 1 \end{array}\right] \\ \begin{array}{cccccc} \theta X & \theta Y & \theta Z & X & Y & Z \end{array} \end{array} \begin{array}{l} \left.\begin{array}{l} D_1 \\ D_2 \\ D_3 \end{array}\right\} R_{1,1,1} \\ D_4 \} R_{1,1,2} \\ D_5 \} R_{1,1,3} \\ \end{array}$$

$$[UD]_{2,2} = \begin{array}{c} \left[\begin{array}{cccccc} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & -40.34 & 0 & 0 \\ 1 & 0 & 0 & 0 & 2.72 & 0 \\ 0 & 0 & 0 & 0 & .997 & -.0771 \\ 0 & 0 & 0 & 0 & .0771 & .997 \\ 0 & .0771 & .997 & -40.34 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & .0771 & .997 \end{array}\right] \\ \begin{array}{cccccc} \theta X & \theta Y & \theta Z & X & Y & Z \end{array} \end{array} \begin{array}{l} \left.\begin{array}{l} KC_1 \\ KC_2 \\ KC_3 \\ KC_4 \end{array}\right\} R_{2,2,1} \\ \left.\begin{array}{l} D_6 \\ D_7 \\ D_8 \end{array}\right\} R_{2,2,2} \\ D_9 \} R_{2,2,3} \\ \left.\begin{array}{l} D_{10} \\ D_{11} \\ D_{12} \end{array}\right\} R_{2,2,4} \\ D_{13} \} R_{2,2,5} \\ D_{14} \} R_{2,2,6} \\ D_{15} \} R_{2,2,7} \\ \end{array}$$

Figure 3.42. [*UD*]*1,1* and [*UD*]*2,2* for the Fingernail Clipper Assembly

3.2.8 Creating the Complete Analytical Model

Now that the mates and dimensions are analytically represented they can be integrated into one unified model. For an arbitrary configuration $\boldsymbol{B_{i,j}}$, $\mathbf{[}\boldsymbol{TM}\mathbf{]}_{\boldsymbol{i,j}}$ can be combined with $\mathbf{[}\boldsymbol{LM}\mathbf{]}_{\boldsymbol{i,j}}$ to create a $\mathbf{(\{}\boldsymbol{6\lambda_{i,j} + 6g_{i,j}}\mathbf{\}\ x\ \{}\boldsymbol{ft_{i,j} + st_{i,j} + 6g_{i,j}}\mathbf{\})}$ matrix, $\mathbf{[}\boldsymbol{TL}\mathbf{]}_{\boldsymbol{i,j}}$, called the twist-loop matrix, and $\mathbf{[}\boldsymbol{WM}\mathbf{]}_{\boldsymbol{i,j}}$ can be combined with $\mathbf{[}\boldsymbol{PM}\mathbf{]}_{\boldsymbol{i,j}}$ to create a $\mathbf{(\{}\boldsymbol{6p_{i,j}\ -6}\mathbf{\}\ x\ \{}\boldsymbol{ct_{i,j} + st_{i,j}}\mathbf{\})}$ matrix, $\mathbf{[}\boldsymbol{WP}\mathbf{]}_{\boldsymbol{i,j}}$, called the wrench-part matrix as follows (adapted from [Davies, 2000]):

$$
\begin{aligned}
&\left[TL_{k,l}\right]_{i,j} = \left[TM_{l,n}\right]_{i,j}\left[LM_{q,l}\right]_{i,j} \\
&for \\
&i = 1,2,\mathrm{K},a \\
&j = 1,2,\mathrm{K},b_i \\
&k = 1,2,...,(6\lambda_{i,j} + 6g_{i,g}) \\
&l = 1,2,...,\left(ft_{i,j} + st_{i,j} + 6g_{i,j}\right) \\
&n = \mathrm{mod}_6\left(k-1\right)+1 \\
&q = \mathrm{int}\left(\left(k-1\right)/6\right)+1
\end{aligned}
$$

Equation 3.33

$$
\begin{aligned}
&\left[WP_{k,l}\right]_{i,j} = \left[WM_{l,n}\right]_{i,j}\left[PM_{q,l}\right]_{i,j} \\
&for \\
&i = 1,2,\mathrm{K},a \\
&j = 1,2,\mathrm{K},b_i \\
&k = 1,2,...,(6p_{i,j} - 6) \\
&l = 1,2,...,\left(ct_{i,j} + st_{i,j}\right) \\
&n = \mathrm{mod}_6\left(k-1\right)+1 \\
&q = \mathrm{int}\left(\left(k-1\right)/6\right)+1
\end{aligned}
$$

Equation 3.34

The "**int**" function accepts any real number input and outputs the next lowest integer from the input. The "**mod$_6$**" function accepts any integer input and outputs the integer remainder one gets when dividing the input by six. Figure 3.43 shows **[*TL*]*$_{1,1}$*** and **[*WP*]*$_{1,1}$*** and Figure 3.44 shows **[*TL*]*$_{2,2}$*** and **[*WP*]*$_{2,2}$*** for the fingernail clipper example that were obtained using Equation 3.33 and Equation 3.34. **[*TM*]*$_{1,1}$*** and **[*TM*]*$_{2,2}$*** were obtained from Figure 3.40, **[*WM*]*$_{1,1}$*** and **[*WM*]*$_{2,2}$*** were obtained from Figure 3.41, **[*PM*]*$_{1,1}$*** and **[*PM*]*$_{2,2}$*** were obtained from Figure 3.31, and **[*LM*]*$_{1,1}$*** and **[*LM*]*$_{2,2}$*** were obtained from Figure 3.32.

$$[TL]_{1,1} = \begin{array}{c} \left[\begin{array}{cccccc} 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 2.72 \\ 0 & 0 & 1 & 0 & -2.72 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \end{array}\right] \\ \begin{array}{cccccc} F_1 & F_2 & F_3 & F_4 & F_5 & F_6 \end{array} \end{array} \left.\begin{array}{c} \theta X \\ \theta Y \\ \theta Z \\ X \\ Y \\ Z \end{array}\right\}\Lambda_{1,1,1}$$

$$[WP]_{1,1} = \begin{array}{c} \left[\begin{array}{cccccccccccc} 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -2.72 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 2.72 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{array}\right] \\ \begin{array}{cccccccccccc} C_1 & C_2 & C_3 & C_4 & C_5 & C_6 & C_7 & C_8 & C_9 & C_{10} & C_{11} & C_{12} \end{array} \end{array} \begin{array}{l} \left.\begin{array}{c} \theta X \\ \theta Y \\ \theta Z \\ X \\ Y \\ Z \end{array}\right\}P_{1,1,2} \\ \left.\begin{array}{c} \theta X \\ \theta Y \\ \theta Z \\ X \\ Y \\ Z \end{array}\right\}P_{1,1,3} \end{array}$$

Figure 3.43. [*TL*]*$_{1,1}$* and [*WP*]*$_{1,1}$* for Configuration *$B_{1,1}$* of the Fingernail Clipper

$$[TL]_{2,2} = \begin{array}{ccccccccccccccccccccc}
0&1&0&1&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&\theta X&\\
0&0&0&0&.0771&0&0&0&0&0&0&0&0&0&0&0&0&0&0&\theta Y&\\
1&0&0&0&.997&0&0&0&0&0&0&0&0&0&0&0&0&0&0&\theta Z&\\
-35.18&0&1&0&-35.51&0&0&0&0&0&0&0&0&0&0&0&0&0&0&X&\Lambda_{2,2,1}\\
0&8.41&0&5.46&0&.997&-2.72&0&0&0&0&0&0&0&0&0&0&0&0&Y&\\
0&35.18&0&35.41&0&-.0771&0&0&0&0&0&0&0&0&0&0&0&0&0&Z&\\
0&0&0&0&0&0&1&-1&0&0&0&0&0&0&0&0&0&0&0&\theta X&\\
0&0&0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&\theta Y&\\
0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&\theta Z&\\
0&0&0&0&0&0&0&0&3.02&40.34&-1&0&0&0&0&0&0&0&0&X&G_{2,2,1}\\
0&0&0&0&0&0&2.72&-3.02&0&0&0&-1&0&0&0&0&0&0&0&Y&\\
0&0&0&0&0&0&0&-40.34&0&0&0&0&-1&0&0&0&0&0&0&z&\\
0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0&\theta X&\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&\theta Y&\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&\theta Z&\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&2.81&40.34&-1&0&0&X&G_{2,2,2}\\
0&0&0&0&0&0&0&0&0&0&0&0&0&-2.81&0&0&0&-1&0&Y&\\
0&0&0&0&0&0&0&0&0&0&0&0&0&-40.34&0&0&0&0&-1&Z&\\
F_7&F_8&F_9&F_{10}&F_{11}&F_{12}&S_1&V_1&V_2&V_3&V_4&V_5&V_6&V_7&V_8&V_9&V_{10}&V_{11}&V_{12}&&
\end{array}$$

$$[WP]_{2,2} = \begin{array}{cccccccccccccccccccc}
1&0&0&0&-35.18&0&0&0&8.41&35.18&0&0&0&0&0&0&0&0&\theta X&\\
0&1&0&0&0&-1&0&-8.41&0&0&0&0&0&0&0&0&0&0&\theta Y&\\
0&0&35.18&0&0&0&-1&-35.18&0&0&0&0&0&0&0&0&0&0&\theta Z&\\
0&0&1&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&X&P_{2,2,3}\\
0&0&0&1&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&Y&\\
0&0&0&0&1&0&0&0&0&-1&0&0&0&0&0&0&0&0&Z&\\
0&0&0&0&0&0&0&0&-8.41&-35.18&0&35.72&0&0&0&0&0&0&\theta X&\\
0&0&0&0&0&1&0&8.41&0&0&-.997&0&0&0&0&0&0&0&\theta Y&\\
0&0&0&0&0&0&1&35.18&0&0&.0771&0&0&0&0&0&0&0&\theta Z&\\
0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&X&P_{2,2,4}\\
0&0&0&0&0&0&0&0&1&0&0&-.0771&0&0&0&0&0&0&Y&\\
0&0&0&0&0&0&0&0&0&1&0&-.997&0&0&0&0&0&0&Z&\\
0&0&0&0&0&0&0&0&0&0&0&-35.72&0&0&0&-2.72&0&1&\theta X&\\
0&0&0&0&0&0&0&0&0&0&.997&0&1&0&2.72&0&0&0&\theta Y&\\
0&0&0&0&0&0&0&0&0&0&-.0771&0&0&1&0&0&0&0&\theta Z&\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&X&P_{2,2,2}\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&Y&\\
0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&1&0&Z&\\
C_{13}&C_{14}&C_{15}&C_{16}&C_{17}&C_{18}&C_{19}&C_{20}&C_{21}&C_{22}&C_{23}&C_{24}&C_{25}&C_{26}&C_{27}&C_{28}&C_{29}&S_1&&
\end{array}$$

Figure 3.44. [*TL*]*2,2* and [*WP*]*2,2* for Configuration *B2,2* of the Fingernail Clipper

In general, **[*TL*]*i,j*** has one column for each DOF, DOS, and DOV and one row for each of the six coordinate directions in each closed-loop and open-loop. **[*WP*]*i,j*** has one column for each DOC or DOS and one row for each of the six coordinate direction of each part (excluding the base part). If the columns of **[*TL*]*i,j*** are assigned variables compiled into a **(**$\{ft_{i,j}+st_{i,j}+6g_{i,j}\}$ x 1**)** vector, **[*Ω*]*i,j***, with each variable representing the displacement allowed by the corresponding DOF, DOS, or DOV, and the columns of **[*WP*]*i,j*** are assigned variables compiled into a **(**$\{ct_{i,j}+st_{i,j}\}$ x 1**)** vector, $[\varphi]_{i,j}$, with each variable representing the force (or moment) transmitted by each DOC and DOS, then by setting each row in $[TL]_{i,j}$ and $[WP]_{i,j}$ equal to zero, $(6\lambda_{i,j}+6g_{i,j}+6p_{i,j}-6)$ conditions are imposed on these variables that collectively require the net change in the velocity around any closed-loop or open-loop in any of the six coordinate directions to be zero and the sum of the forces acting on each part (excluding the base part) in any of the six coordinate directions to be zero. This can be expressed by the following two equations (adapted from [Davies, 2000]):

$$\left[TL_{((6\lambda_{i,j}+6g_{i,g})\times(ft_{i,j}+st_{i,j}+6g_{i,j}))}\right]_{i,j}\left[\Omega_{((ft_{i,j}+st_{i,j}+6g_{i,j})\times 1)}\right]_{i,j}=\left[0_{((6\lambda_{i,j}+6g_{i,g})\times 1)}\right]$$

$$\begin{aligned}&\mathit{for}\\&i=1,2,\ldots,a\\&j=1,2,\ldots,b_i\end{aligned}$$

Equation 3.35

$$\left[WP_{((6p_{i,j}-6)\times(ct_{i,j}+st_{i,j}))}\right]_{i,j}\left[\varphi_{((ct_{i,j}+st_{i,j})\times 1)}\right]_{i,j}=\left[0_{((6p_{i,j}-6)\times 1)}\right]$$

$$\begin{aligned}&\mathit{for}\\&i=1,2,\ldots,a\\&j=1,2,\ldots,b_i\end{aligned}$$

Equation 3.36

In addition, if any of the real mates have compliance, then $st_{i,j}$ of the variables in $[\Omega]_{i,j}$ (those associated with the columns representing the DOS) are related to the last $st_{i,j}$ of the variables in $[\boldsymbol{\varphi}]_{i,j}$, (those associated with the columns representing the DOS) as follows (adapted from [Fasse]:

$$
\begin{aligned}
&\left[\varphi_{q_1:q_2,1}\right]_{i,j} = \left[\, SK \,\right]_{i,j} \left[\Omega_{q_3:q_4,1}\right]_{i,j} \\
&\mathit{where} \\
&q_1 = ct_{i,j} + 1 \\
&q_2 = ct_{i,j} + st_{i,j} \\
&q_3 = ft_{i,j} + 1 \\
&q_4 = ft_{i,j} + st_{i,j} \\
&\mathit{for} \\
&i = 1,2,\mathrm{K}\,,a \\
&j = 1,2,\mathrm{K}\,,b_i
\end{aligned}
$$

Equation 3.37

Equation 3.35 is the mechanical equivalent of Kirchoff's Voltage Law [Strang] requiring the net change in voltage (or displacement) around any loop (open-loop or closed loop) in a circuit (or configuration) to be zero. Equation 3.36 is the mechanical equivalent of Kirchoff's Current Law [Strang] requiring the net current (or force) entering a leaving a node (or part) in a circuit (or configuration) to be zero. Equation 3.37 is the mechanical equivalent of Ohm's Law [Strang] stating that the voltage drops across (or displacements allowed by) the resistors (or compliant mates) is equal to the resistances (stiffness values in the stiffness matrix) times the currents traveling through (or forces transmitted by) the resistors (compliant mates).

For an arbitrary configuration ***B***$\boldsymbol{B_{i,j}}$, Equation 3.35, Equation 3.36, and Equation 3.37 can be combined into one system of equations as follows:

$$[B]_{i,j}\,[\Psi]_{i,j}=\left[0_{\left(\left(6m_{i,j}+6g_{i,j}+st_{i,j}\right)\times 1\right)}\right]$$

$$\textit{where}$$

$$[B]_{i,j}=\left[\begin{array}{ccccc} \multicolumn{3}{c}{\left[TL_{\left(\left(6\lambda_{i,j}+6g_{i,j}\right)\times\left(ft_{i,j}+6g_{i,j}+st_{i,j}\right)\right)}\right]_{i,j}} & \multicolumn{2}{c}{\left[0_{\left(\left(6\lambda_{i,j}+6g_{i,j}\right)\times\left(ct_{i,j}+st_{i,j}\right)\right)}\right]} \\ \left[0_{\left(st_{i,j}\times ft_{i,j}\right)}\right] & \left[SK_{\left(st_{i,j}\times st_{i,j}\right)}\right]_{i,j} & \left[0_{\left(st_{i,j}\times 6g_{i,j}\right)}\right] & \left[0_{\left(st_{i,j}\times ct_{i,j}\right)}\right] & \left[-I_{\left(st_{i,j}\times st_{i,j}\right)}\right] \\ \multicolumn{3}{c}{\left[0_{\left(\left(6p_{i,j}-6\right)\times\left(ft_{i,j}+6g_{i,j}+st_{i,j}\right)\right)}\right]} & \multicolumn{2}{c}{\left[WP_{\left(\left(6p_{i,j}-6\right)\times\left(ct_{i,j}+st_{i,j}\right)\right)}\right]_{i,j}} \end{array}\right]$$

$$[\Psi]_{i,j}=\left[\begin{array}{cc}\left[\Omega_{\left(\left(ft_{i,j}+6g_{i,j}+st_{i,j}\right)\times 1\right)}\right]_{i,j} & \left[\varphi_{\left(\left(ct_{i,j}+st_{i,j}\right)\times 1\right)}\right]_{i,j}\end{array}\right]$$

$$\textit{for}$$
$$i=1,2,\ldots,a$$
$$j=1,2,\ldots,b_i$$

Equation 3.38

[*B*]**$_{i,j}$, is a **(**$\boldsymbol{\{6m_{i,j}+6g_{i,j}+st_{i,j}\}}$ **x** $\boldsymbol{\{6m_{i,j}+6g_{i,j}+st_{i,j}\}}$**)** matrix, called the nominal configuration matrix, and represents the assembly configuration in its nominal form including its mates, loops, and parts. **[*Ψ*]**$_{i,j}$, is a **(**$\boldsymbol{\{6m_{i,j}+6g_{i,j}+st_{i,j}\}}$ **x *1*)** vector that contains the unknown displacements allowed or forces transmitted by the DOF, DOC, DOS, and DOV of the mates.

For an arbitrary configuration $\boldsymbol{B_{i,j}}$, $\boldsymbol{[UD]_{i,j}}$ can be combined with $\boldsymbol{[LD]_{i,j}}$ to create a $\boldsymbol{(\{6\lambda_{i,j}+6g_{i,j}\}\ \mathrm{x}\ dt_{i,j})}$ matrix, $\boldsymbol{[UL]_{i,j}}$, called the uncertainty-loop matrix and this is done as follows:

$$
\begin{aligned}
&\left[UL_{k,l}\right]_{i,j} = \left[UD_{n,l}\right]_{i,j}\left[LD_{l,q}\right]_{i,j} \\
&for \\
&i = 1,2,\mathrm{K}\,,a \\
&j = 1,2,\mathrm{K}\,,b_i \\
&k = 1,2,...,6(\lambda_{i,j} + g_{i,g}) \\
&l = 1,2,...,dt_{i,j} \\
&n = \mathrm{int}((k-1)/6)+1 \\
&q = \mathrm{mod}_6(k-1)+1
\end{aligned}
$$

Equation 3.39

Figure 3.45 shows $\boldsymbol{[UL]_{1,1}}$ and $\boldsymbol{[UL]_{2,2}}$ for the fingernail clipper example obtained using Equation 3.39 and the results shown in Figure 3.32 and Figure 3.42 .

$$[UL]_{1,1} = \begin{array}{c}\left[\begin{array}{ccccc} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{array}\right] \\ \begin{array}{ccccc} D_1 & D_2 & D_3 & D_4 & D_5 \end{array}\end{array} \begin{array}{l}\left.\begin{array}{c} \theta X \\ \theta Y \\ \theta Z \\ X \\ Y \\ Z \end{array}\right\}\Lambda_{1,1,1} \\ \\ \end{array}$$

$$[UL]_{2,2} = \begin{array}{c}\left[\begin{array}{cccccccccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & -2.72 & 0 & 0 & 0 & 1 & 0 & .0771 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & .997 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & .0771 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & .997 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -40.45 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 2.72 & .997 & .0771 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -.0771 & .997 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -40.34 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \\ \begin{array}{cccccccccccccc} KC_1 & KC_2 & KC_3 & KC_4 & D_6 & D_7 & D_8 & D_9 & D_{10} & D_{11} & D_{12} & D_{13} & D_{14} & D_{15} \end{array}\end{array} \begin{array}{l}\left.\begin{array}{c} \theta X \\ \theta Y \\ \theta Z \\ X \\ Y \\ Z \end{array}\right\}\Lambda_{2,2,1} \\ \left.\begin{array}{c} \theta X \\ \theta Y \\ \theta Z \\ X \\ Y \\ Z \end{array}\right\}G_{2,2,1} \\ \left.\begin{array}{c} \theta X \\ \theta Y \\ \theta Z \\ X \\ Y \\ Z \end{array}\right\}G_{2,2,2} \\ \\ \end{array}$$

Figure 3.45. $[UL]_{1,1}$ and $[UL]_{2,2}$ for the Fingernail Clipper Assembly

In general, $[UL]_{i,j}$ has one row for each coordinate direction of each closed-loop and open-loop and one column for each dimension. If the columns of $[UL]_{i,j}$ are assigned variables compiled into a $(dt_{i,j}$ x $1)$ vector, $[\Delta]_{i,j}$, with each variable representing the difference between the nominal value of the dimension and the actual value of the dimension, by setting each row in $[UL]_{i,j}$ equal to zero, $(6\lambda_{i,j} + 6g_{i,j})$ conditions are imposed on these variables that collectively require the net change in the nominal values of the DOF, DOS, and DOV of around any loop (open or closed) in any coordinate direction due to difference between the nominal and actual dimension values to be zero. This can be expressed as follows (adapted from [Gao, Chase, and Magleby]):

$$\left[UL_{((6\lambda_{i,j}+6g_{i,g})\times dt_{i,j})}\right]_{i,j}\left[\Delta_{(dt_{i,j}\times 1)}\right]_{i,j}=\left[0_{((6\lambda_{i,j}+6g_{i,g})\times 1)}\right]$$

$$\begin{aligned}&\mathit{for}\\&i,=1,2,\ldots,a\\&j,=1,2,\ldots,b_i\end{aligned}$$

Equation 3.40

To complete the variability model, a **(\{6*m**$_{i,j}$**+6g**$_{i,j}$**+st**$_{i,j}$**\}** **x** ***dt**$_{i,j}$**)** matrix **[*U*]**$_{i,j}$, called the uncertainty configuration matrix, can be formed and set equal to zero as follows (adapted from [Gao, Chase, and Magleby]):

$$[U]_{i,j}[\Delta]_{i,j}=\left[0_{((6m_{i,j}+6g_{i,g}+st_{i,j})\times 1)}\right]$$

$$\mathit{where}$$

$$[U]_{i,j}=\begin{bmatrix}\left[UL_{((6\lambda_{i,j}+6g_{i,j})\times dt_{i,j})}\right]_{i,j}\\ \\ \left[0_{((6p_{i,j}-6+st_{i,j})\times dt_{i,j})}\right]_{i,j}\end{bmatrix}$$

$$\begin{aligned}&\mathit{for}\\&i=1,2,\ldots,a\\&j=1,2,\ldots,b_i\end{aligned}$$

Equation 3.41

A matrix of zeros is added so that **[*U*]**$_{i,j}$ has the same number of rows as **[*B*]**$_{i,j}$, namely one row for each coordinate direction of each closed-loop, open-loop, and part (excluding the base part) plus one row for each DOS. If external forces exhibiting variability were applied to the assembly, then **$[U]_{i,j}$** would require a different formulation and could have additional columns, one for each applied force. The first **$(6\lambda_{i,j}+6g_{i,j}+st_{i,j})$** entries in each of those rows would all be zeros and the last **$(6p_{i,j} - 6)$** rows would correspond to the components of the forces applied in each of the six coordinate directions to each of the parts (excluding the base part).

With the nominal and variable properties of the assembly modeled, a unified model of the assembly can be created. For an arbitrary configuration, ***$B_{i,j}$***, Equation 3.38 and Equation 3.41 and can be combined as follows:

$$\begin{aligned}&[B]_{i,j}[\Psi]_{i,j}+[U]_{i,j}\,[\Delta]_{i,j}=[0]_{i,j}\\&for\\&i,=1,2,\ldots,a\\&j,=1,2,\ldots,b_i\end{aligned}$$

Equation 3.42

If nominal static applied forces were present, then the last **$(6p_{i,j}$-6)** entries of the zero vector on the right side of Equation 3.42 would be replaced with the resultant applied forces on each part excluding the base part in each of the six coordinate directions. When there is no variability or uncertainty in the dimensions, the Equation 3.42 reduces to Equation 3.38. If all the parts can be assumed to be rigid, then Equation 3.42 simplifies as follows (adapted from [Gao, Chase, and Magleby]):

$$[TL]_{i,j}[\Omega]_{i,j}+[UL]_{i,j}[\Delta]_{i,j}=[0]_{i,j}$$
$$for$$
$$i,=1,2,\ldots,a$$
$$j,=1,2,\ldots,b_i$$

Equation 3.43

Equation 3.42 resembles, with some differences, the familiar equation used to conduct static load analysis of structures (like trusses) in equilibrium (adapted from [Paz and Leigh]):

$$\hat{K}\overline{x}+\hat{L}\overline{F}=\overline{0}$$

Equation 3.44

$\hat{K}$ is the global stiffness matrix and is analogous to the configuration matrix **[*B*]*_{i,j}***. $\hat{L}$ is a matrix containing all the unit loads and/or displacements and is analogous to the matrix containing all the unit changes in dimensions, $[\boldsymbol{U}]_{i,j}$. $\overline{F}$ is a vector of scalars that combines the unit loads and deflections into a single loading case and is analogous to the vector containing the differences between the average values of the dimensions and the actual values of the dimensions, $[\boldsymbol{\Delta}]_{i,j}$. Finally, $\overline{x}$ contains the changes in length/shape of the structure's members (and/or the changes in internal forces acting through the members) that result from the loading case in $\overline{x}$ and is like the vector containing the changes in velocity and force at the mates, $[\boldsymbol{\Psi}]_{i,j}$.

The main difference between the system of equations in Equation 3.43 and the system of equation in Equation 3.42 are the unknown variables represented by $\bar{x}$ and $[\boldsymbol{\Psi}]_{i,j}$. While $\bar{x}$ contains the changes in length/shape (and/or the change in forces acting through the members) of the structure members, $[\boldsymbol{\Psi}]_{i,j}$ contains the displacements allowed and forces transmitted by the DOF, DOC, DOS, and DOV of the mates. These variables are desired because in this work, manufacturability is defined as the variability in the KCs, which are by definition geometric relationships between features on different parts. Thus to find the variability of the KCs, a model is needed whose solvable variables are the changes in location of the parts and the features they contain relative to each other rather than the internal forces and deflections of the parts.

3.3 Summary

In this chapter is has been shown how to create a model for the nominal configurations of an assembly. While some of the equations and procedures presented in this chapter are complex, a major advantage regarding them is that all are completely amendable to computer implementation. Using Matlab® programming language, an executable, entitled modeler.m, was developed that automatically creates the analytical assembly model ($[\boldsymbol{B}]_{i,j}$, $[\boldsymbol{U}]_{i,j}$, $[\boldsymbol{TL}]_{i,j}$, and $[\boldsymbol{WP}]_{i,j}$) and solves for the nominal KC values ($[\boldsymbol{KCn}]_{i,j}$) for a single configuration ($\boldsymbol{B}_{i,j}$) of an assembly. As inputs, modeler.m takes the vectors and matrices shown in Table 3.14 (see Table 3.13 for values for counting variable descriptions) and provides as outputs, the matrices shown in Table 3.15. A copy of modeler.m is included in Appendix A. It can be run on Release 13 of Matlab® and requires the Optimization Toolbox to be installed. In addition to using standard Matlab$^{®}$ functions and some specialized functions from the Optimization Toolbox, modeler.m calls two other subroutines called energyloop.m and combiner.m that are also included in Appendix A.

To model an entire assembly, one must enter the appropriate inputs for the first configuration of the first assembly state and solve for the nominal values of the linking KC in $[\boldsymbol{KCn}]_{i,j}$. Then these values can be entered as input dimensions for the linked configuration(s). Repeating this procedure for all assembly states and configurations, one can obtain a model of the entire assembly.

Variable	Description
$p_{i,j}$	Total # of parts in configuration $B_{i,j}$
$e_{i,j}$	Total # features in configuration $B_{i,j}$
$m_{i,j}$	Total # of mates in configuration $B_{i,j}$
$r_{i,j}$	Total # of intra-part relationships in configuration $B_{i,j}$
$h_{i,j}$	Total # of KC feature pairs in configuration $B_{i,j}$
$\lambda_{i,j}$	Total # indpendent closed-loops for configuration $B_{i,j}$
$g_{i,j}$	Total # indpendent open-loops for configuration $B_{i,j}$
$ft_{i,j}$	Total # of DOF for configuration $B_{i,j}$
$st_{i,j}$	Total # of DOS for configuration $B_{i,j}$
$ct_{i,j}$	Total # of DOC for configuration $B_{i,j}$
$dt_{i,j}$	Total # of Dimensions for configuration $B_{i,j}$
$kct_{i,j}$	Total # of KC's for configuration $B_{i,j}$

Table 3.13. Counting Variables for Configuration $B_{i,j}$ of an Assembly

Vector or Matrix	Description	Size
$[Fn]_{i,j}$	Guesses for Nominal DOF Values	$(ft_{i,j}\ \mathrm{x}\ 1)$
$[Ful]_{i,j}$	Upper Limits on DOF values	$(ft_{i,j}\ \mathrm{x}\ 1)$
$[Fll]_{i,j}$	Lower Limits on DOF values	$(ft_{i,j}\ \mathrm{x}\ 1)$
$[Fd]_{i,j}$	Direcions for DOF	$(ft_{i,j}\ \mathrm{x}\ 4)$
$[Sn]_{i,j}$	Guesses for Nominal DOS Values	$(st_{i,j}\ \mathrm{x}\ 1)$
$[Sul]_{i,j}$	Upper Limits on DOS values	$(st_{i,j}\ \mathrm{x}\ 1)$
$[Sll]_{i,j}$	Lower Limits on DOS values	$(st_{i,j}\ \mathrm{x}\ 1)$
$[Sd]_{i,j}$	Direcions for DOS	$(st_{i,j}\ \mathrm{x}\ 1)$
$[SK]_{i,j}$	Nominal DOS stiffness matrix	$(st_{i,j}\ \mathrm{x}\ st_{i,j})$
$[Cn]_{i,j}$	Nominal DOC values	$(ct_{i,j}\ \mathrm{x}\ 1)$
$[Cd]_{i,j}$	Direcions for DOC	$(ct_{i,j}\ \mathrm{x}\ 4)$
$[Dn]_{i,j}$	Nominal dimension values	$(dt_{i,j}\ \mathrm{x}\ 1)$
$[Dd]_{1,1}$	Direcions for dimensions	$(dt_{i,j}\ \mathrm{x}\ 1)$
$[KCd]_{i,j}$	KC directions	$(kct_{i,j}\ \mathrm{x}\ 1)$
$[KCt]_{i,j}$	KC type	$(kct_{i,j}\ \mathrm{x}\ 1)$
$[KCl]_{i,j}$	KC specification offsets	$(kct_{i,j}\ \mathrm{x}\ 1)$
$[CHd]_{i,j}$	Matrix of dimension chains	$(dt_{i,j}\ \mathrm{x}\ ?)$
$[CHm]_{i,j}$	Matrix of DOF, DOS,and DOC chains	$(6m_{i,j}\ \mathrm{x}\ ?)$
$[CHc]_{i,j}$	Matrix of closed-loop chains	$(\lambda_{i,j}\ \mathrm{x}\ ?)$
$[CHo]_{i,j}$	Matrix of open-loop chains	$(g_{i,j}\ \mathrm{x}\ ?)$
$[CHk]_{i,j}$	Matrix of KC chains	$(kct_{i,j}\ \mathrm{x}\ ?)$

Table 3.14. Modeler.m Inputs for Configuration $B_{i,j}$ of an Assembly

Vector or Matrix	Description	Size
$[B]_{i,j}$	Nominal Configuration Matrix for configuration $B_{i,j}$	$(\{6m_{i,j}+6g_{i,j+}st_{i,j}\}\ \mathrm{x}\ \{6m_{i,j}+6g_{i,j}+st_{i,j}\})$
$[U]_{i,j}$	Uncertainty Matrix for configuration $B_{i,j}$	$(\{6m_{i,j}+6g_{i,j}+st_{i,j}\}\ \mathrm{x}\ dt_{i,j})$
$[TL]_{i,j}$	Twist Loop Matrix for configuration $B_{i,j}$	$(\{6\lambda_{i,j}+6g_{i,j}\}\ \mathrm{x}\ \{ft_{i,j}+st_{i,j}+6g_{i,j}\})$
$[WP]_{i,j}$	Wrench Part Matrix for configuration $B_{i,j}$	$(\{6p_{i,j}-6\}\ \mathrm{x}\ \{ct_{i,j}+st_{i,j}\})$
g	Number of open-loops for configuration $B_{i,j}$	$(1\ \mathrm{x}\ 1)$
$\beta_{i,j}$	Stored energy in $B_{i,j}$	$(1\ \mathrm{x}\ 1)$
$[Fn^{*}]_{i,j}$	Solved Nominal DOF values for configuration $B_{i,j}$	$(ft_{i,j}\ \mathrm{x}\ 1)$
$[Sn^{*}]_{i,j}$	Solved Nominal DOS values for configuration $B_{i,j}$	$(st_{i,j}\ \mathrm{x}\ 1)$
$[KCL]_{i,j}$	KC Specifciation Limits for configuration $B_{i,j}$	$(kct_{i,j}\ \mathrm{x}\ 1)$
$[KCn]_{i,j}$	Nominal KC Values for configuration $B_{i,j}$	$(kct_{i,j}\ \mathrm{x}\ 1)$

Table 3.15. Modeler.m Outputs for Configuration $B_{i,j}$ of an Assembly

4. VERIFYING MANUFACTURABILITY

In Chapter 3, a generalized quantitative model of a mechanical assembly design was developed (see Equation 3.42). In this chapter, it is shown how this model can be used to verify, among other things, the manufacturability of an assembly. This is accomplished via five types of analysis, namely:

- ***Motion/Constraint***
- ***Sensitivity***
- ***Performance***
- ***Contribution***
- ***Coupling***

Each of these will now be discussed in more detail. The discussion that follows is by no means all inclusive and is intended to merely illustrate some of the uses of the assembly model developed in Chapter 3.

4.1 Motion/Constraint Analysis

If an assembly has excess under-constraints, then for Equation 3.42, there may exist more than one vector $[\psi]_{i,j}$,for every input $[\boldsymbol{\Delta}]_{i,j}$, and as such, $[\boldsymbol{B}]_{i,j}$ will not be full rank which means it will not be invertible. If the assembly has excess under-constraints, then there may not exist any vector $[\psi]_{i,j}$, for a given input $[\boldsymbol{\Delta}]_{i,j}$, and this will also result in $[\boldsymbol{B}]_{i,j}$ not being full rank. If either or both of these situations exist, then the usefulness of the model developed in Chapter 3 is severely limited. Therefore, there is a need to determine whether excess under-constraints and/or over-constraints exist and eliminate them if possible.

Motion analysis and constraint analysis are the names traditionally given to the processes of determining whether and assembly has any excess under-constraints and over-constraints respectively. There are many examples of such methods including those using Screw Theory by [Adams and Whitney], [Baker], [Davies, 1983b, 1983c, and 2000], [Shukla and Whitney], [Smith], [Waldron] and those using CAD systems by [Anantha, Kramer, and Crawford] and [Mullins and Anderson]. Here a modified formulation from [Davies, 2000] is used.

$[\boldsymbol{B}]_{i,j}$ is invertible when there are no excess under-constraints or over-constraints and this is true, for the model developed in this work, when one of the following equivalent expressions is valid:

$$
\begin{array}{l}
r1_{i,j} + r2_{i,j} = 6\left(\lambda_{i,j} + g_{i,j} + p_{i,j} - 1\right) \\
or \\
r1_{i,j} + r2_{i,j} = 6m_{i,j} + 6g_{i,j} \\
where \\
r1_{i,j} = \mathrm{rank}\left([TL]_{i,j} \right) \\
r2_{i,j} = \mathrm{rank}\left([WP]_{i,j} \right) \\
for \\
i = 1,2,\ldots,a \\
j = 1,2,\ldots,b_i
\end{array}
$$

Equation 4.1

The "**rank**" function takes a matrix input and returns the number of independent columns in the matrix. The number of excess under-constraints, ***fe***$\boldsymbol{fe_{i,j}}$ (excluding under-constraints eliminated by DOS), and the number of excess over-constraints, $\boldsymbol{ce_{i,j}}$ (excluding over-constraints eliminated by DOS), for an arbitrary configuration, $\boldsymbol{B_{i,j}}$, are defined as follows (modified from [Davies, 2000]):

$$
\begin{aligned}
&fe_{i,j} = \left(6p_{i,j} - 6\right) - r2_{i,j} \\
&for \\
&i = 1,2,\ldots,a \\
&j = 1,2,\ldots,b_i
\end{aligned}
$$

Equation 4.2

$$
\begin{aligned}
&ce_{i,j} = \left(6\lambda_{i,j} + 6g_{i,j}\right) - r1_{i,j} \\
&for \\
&i = 1,2,\ldots,a \\
&j = 1,2,\ldots,b_i
\end{aligned}
$$

Equation 4.3

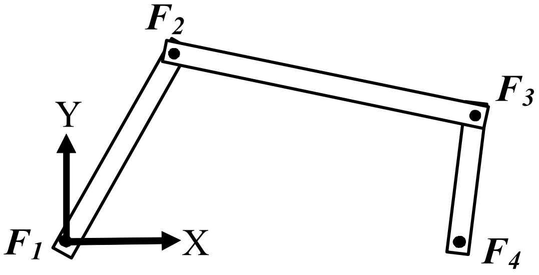
If the parts can be assumed to be rigid, then there is not need to create $\boldsymbol{[WP]_{i,j}}$ because $\boldsymbol{[B]_{i,j}}$ reduces to $\boldsymbol{[TL]_{i,j}}$ (see Equation 3.43). In this case, $\boldsymbol{fe_{i,j}}$ is calculated using ***r1***, the number of independent columns in $\boldsymbol{[TL]_{i,j}}$, as follows (modified from [Davies, 2000]):

$$
\begin{aligned}
&fe_{i,j} = ft_{i,j} + 6g_{i,j} - r1_{i,j} \\
&for \\
&i = 1,2,\ldots,a \\
&j = 1,2,\ldots,b_i
\end{aligned}
$$

Equation 4.4

It is important to note that both $\boldsymbol{fe_{i,j}}$ and $\boldsymbol{ce_{i,j}}$ have been defined such that each will always be either zero or a positive integer. Also, they are defined in terms of the rank of two matrices, $\boldsymbol{[TL]_{i,j}}$ and $\boldsymbol{[WP]_{i,j}}$. This means that the values of $\boldsymbol{fe_{i,j}}$ and $\boldsymbol{ce_{i,j}}$ are subject to all the problems associated with numerical-based methods such as truncation error, precision error, and round-off error. Also, some assemblies that have states that are under-constrained may have different values of $\boldsymbol{fe_{i,j}}$ and $\boldsymbol{ce_{i,j}}$ for two different configurations in those states [Davies, 1983a]. Figure 4.1 shows a simple rigid 4-bar linkage assembly that undergoes a change of constraint-status when the four revolute joints become aligned along the $\boldsymbol{X}$-axis.

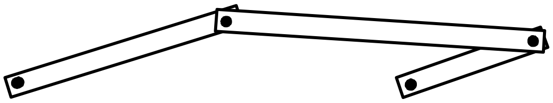
$$[TL]_{1,1} = \begin{array}{c} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 3.46 & 2.84 & 0 \\ 0 & -2 & -6.96 & -6 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\ \begin{array}{cccc} F_1 & F_2 & F_3 & F_4 \end{array} \end{array}$$



$$fe_{1,1} = ft_{1,1} + 6g_{1,1} - r1_{1,1} = 4 + 0 - 3 = 1$$
$$ce_{1,1} = 6(\lambda_{1,1} + g_{1,1}) - r1_{1,1} = 6(1+0) - 3 = 3$$

Under-Constraints: θz
Over-Constraints: z, θx, θy

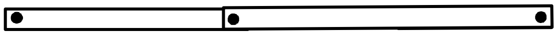
$$[TL]_{1,2} = \begin{array}{c} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1.03 & .899 & 0 \\ 0 & -3.86 & -8.86 & -6 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\ \begin{array}{cccc} F_1 & F_2 & F_3 & F_4 \end{array} \end{array}$$



$$fe_{1,1} = ft_{1,1} + 6g_{1,1} - r1_{1,1} = 4 + 0 - 3 = 1$$
$$ce_{1,1} = 6(\lambda_{1,1} + g_{1,1}) - r1_{1,1} = 6(1+0) - 3 = 3$$

Under-Constraints: θz
Over-Constraints: z, θx, θy

$$[TL]_{1,3} = \begin{array}{c} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & -4 & -9 & -6 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\ \begin{array}{cccc} F_1 & F_2 & F_3 & F_4 \end{array} \end{array}$$



$$fe_{1,1} = ft_{1,1} + 6g_{1,1} - r1_{1,1} = 4 + 0 - 2 = 2$$
$$ce_{1,1} = 6(\lambda_{1,1} + g_{1,1}) - r1_{1,1} = 6(1+0) - 2 = 4$$

Under-Constraints: θz, y
Over-Constraints: x, z, θx, θy

Figure 4.1. Example of an Assembly That Undergoes a Change in Constraint-Status

Keeping the above mentioned issues in mind, Equation 4.2 and Equation 4.3 can be substituted back into Equation 4.1 to yield the following result:

$$
\begin{aligned}
&fe_{i,j} + ce_{i,j} = 0 \\
&\mathit{for} \\
&i = 1,2,\ldots,a \\
&j = 1,2,\ldots,b_i
\end{aligned}
$$

Equation 4.5

If Equation 4.5 is true, then **[*B*]*ᵢ,ⱼ*** is invertible and consequently, the assembly model is amendable to analysis. Equation 4.5 is not true, there are several explanations. The first is that the model was made incorrectly. In this case the model must be corrected to simulate the true behavior of the assembly. If the model is correct, or after correcting it, Equation 4.5 is still not true, then a second explanation is that one or more design mistakes were made (unintentional under-constraints or over-constraints remained in the design) and these mistakes resulted in an undesirable constraint-status. If this is the case, then the design must be modified to eliminate these mistakes by eliminating the unintentional over-constraints and under-constraints. If no design mistakes were made, or after fixing them, Equation 4.5 is still not true, it means that the remaining over-constraints and under-constraints are intentional and critical to the design. Nonetheless, one would still like to analyze such assemblies and so several strategies are suggested for "eliminating" the excess under-constraints and over-constraints.

If intentional under-constraints exist, there are two potential remedies. The first is to reduce ***fe$_{i,j}$*** by choosing a discrete set of configuration(s) for the under-constrained assembly state(s). For each configuration, one or more DOF should be temporarily converted into DOC and the modeling procedure in Chapter 3 repeated. If done properly, the resulting models for all configurations will not have any excess under-constraints. Care must be taken to pick the proper DOF for conversion so that the number of under-constraints is reduced but the number of over-constraints remains unchanged.

Finding a **(*ft$_{i,j}$*x *fe$_{i,j}$*)** matrix, **[*TLr*]*$_{i,j}$***, also called the reduced twist-loop matrix, is very helpful for identifying which mates to remove or which DOF to convert into DOC. **[*TLr*]*$_{i,j}$*** can be obtained from the following equation (adapted from [Davies, 2000]):

$$
\begin{aligned}
&\left[\, TLr \,\right]_{i,j} = \mathrm{null}_r\!\left(\left[\, TL_{\,1:6\lambda_{i,j_1},1:ft_{i,j}} \right]_{i,j} \right)\\
&where\\
&q_1 = 6\lambda_{i,j}\\
&for\\
&i = 1,2,\ldots,a\\
&j = 1,2,\ldots,b_i
\end{aligned}
$$

Equation 4.6

The "**null$_{\mathbf{r}}$**" function takes a matrix input and returns a matrix whose columns form a "rational" basis that spans the null-space of the input matrix. There exist many methods, such as that presented by [Dai and Jones], for finding the null-space of a matrix. Many computer programs, such as Matlab®, also posses such capabilities. Requiring that

$[\boldsymbol{TLr}]_{i,j}$ is a rational basis insures that each of its $\boldsymbol{fe_{i,j}}$ column vectors is aligned with one of the DOF of the mates in the assembly and that each column in $[\boldsymbol{TLr}]_{i,j}$ represents a set of interdependent DOF that together are responsible for one of the excess under-constraints. If entry $[\boldsymbol{TLr_{k,l}}]_{i,j}$ is nonzero, then it means that the $\boldsymbol{k}^{\text{th}}$ DOF is partly responsible for the $\boldsymbol{l}^{\text{th}}$ excess under-constraint and by temporarily fixing the $\boldsymbol{k}^{\text{th}}$ DOF, the $\boldsymbol{l}^{\text{th}}$ under-constraint will be temporarily eliminated.

A second option to deal with, but no remove, the intentional under-constraint(s) is to convert one or more of the DOF into DOS. Care must be taken when doing this so that the number of excess over-constraints is not increased. $[\boldsymbol{TLr}]_{i,j}$ can help identify which DOF should be converted into DOS.

If excess over-constraints exist, they can be dealt with in one of two ways. The first is by converting one or more of the DOC into DOF or adding additional mates such that the excess over-constraints are eliminated but the number of under-constraints remains unchanged.

Finding a $(\boldsymbol{ct_{i,j}}\mathbf{x}\ \boldsymbol{ce_{i,j}})$ matrix, $[\boldsymbol{WPr}]_{i,j}$, also called the reduced wrench-part matrix, is very helpful for identifying which DOF to add. $[\boldsymbol{WPr}]_{i,j}$ can be obtained from the following equation:

$$
\begin{aligned}
&\left[\, WPr \right]_{i,j} = \mathrm{null}_r\left(\left[\, WP_{1:6\left(\lambda_{i,j}+g_{i,g}\right)_1 ,\, 1:ct_{i,j}} \right]_{i,j} \right) \\
&\textit{for} \\
&i = 1,2,\ldots,a \\
&j = 1,2,\ldots,b_i
\end{aligned}
$$

Equation 4.7

Each of the $\boldsymbol{ce_{i,j}}$ columns of $\mathbf{[}\boldsymbol{WPr}\mathbf{]}_{\boldsymbol{i,j}}$ is aligned with one of the DOC of the mates in the assembly and each column in $\mathbf{[}\boldsymbol{WPr}\mathbf{]}_{\boldsymbol{i,j}}$ represents a set of interdependent DOC that together are responsible for one of the excess over-constraints. If entry $\mathbf{[}\boldsymbol{WPr_{k,l}}\mathbf{]}_{\boldsymbol{i,j}}$ is nonzero, then it means that the $\boldsymbol{k}$th DOC is partly responsible for the $\boldsymbol{l}$th intentional over-constraint and by temporarily converting the $\boldsymbol{k}$th DOC into a DOF, the $\boldsymbol{l}$th over-constraint will be temporarily eliminated.

A second option to deal with, but not remove, the over-constraints is to convert one or more of the DOC into DOS and care should be taken to insure that no excess under-constraints result. $\mathbf{[}\boldsymbol{WPr}\mathbf{]}_{\boldsymbol{i,j}}$ can help identify which DOC should be converted into DOS.

Figure 4.2 shows the constraint analysis results for $\boldsymbol{B_{1,1}}$ and $\boldsymbol{B_{2,2}}$ of the fingernail clipper assembly. It can be seen that $\boldsymbol{B_{1,1}}$ has no excess over-constraints or under-constraints and $\boldsymbol{B_{2,2}}$ has no excess over-constraints but has one excess under-constraint.

Configuration *B*$_{1,1}$

$$ft_{1,1}=6,\ ct_{1,1}=12,\ st_{1,1}=0,$$
$$\lambda_{1,1}=1,\ g_{1,1}=0,\ p_{1,1}=3,$$
$$r1_{1,1}=6,\ r2_{1,1}=12$$

$$fe_{1,1}=6(p_{1,1}-1)-r2_{1,1}=6(3-1)-12=0$$
or
$$fe_{1,1}=ft_{1,1}+6g_{1,1}-r1_{1,1}=6+6\times0-6=0$$

and
$$ce_{1,1}=6(\lambda_{1,1}+g_{1,1})-r1_{1,1}=6(1+0)-6=0$$

so
$$fe_{1,1}+ce_{1,1}=$$
is
$$\boxed{0+0=0}$$

0 Excess Under-Constraints & 0 Excess Over-Constraints

Configuration *B*$_{2,2}$

$$ft_{2,2}=6,\ ct_{2,2}=17,\ st_{2,2}=1,$$
$$\lambda_{2,2}=1,\ g_{2,2}=2,\ p_{2,2}=4,$$
$$r1_{2,2}=18,\ r2_{2,2}=17$$

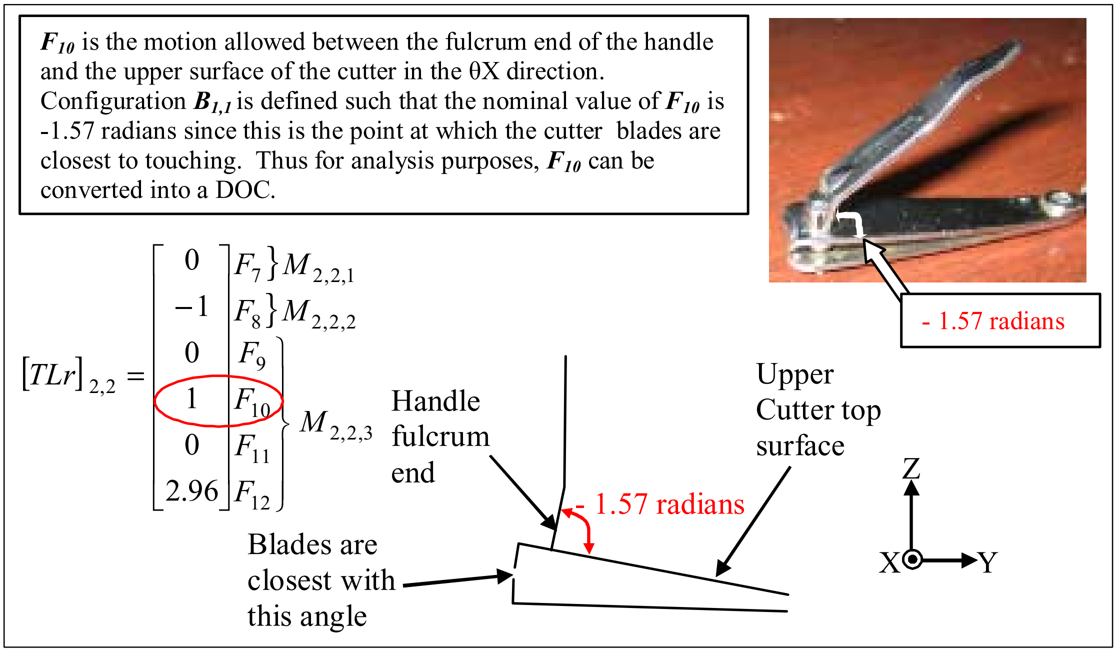
$$fe_{2,2}=6(p_{2,2}-1)-r2_{2,2}=6(4-1)-17=1$$
and
$$ce_{2,2}=6(\lambda_{2,2}+g_{2,2})-r1_{2,2}=6(1+2)-18=0$$
or

so
$$fe_{2,2}+ce_{2,2}=$$
is
$$\boxed{1+0\neq0}$$

1 Excess Under-Constraint & 0 Excess Over-Constraints

Figure 4.2. Constraint Analysis of the Fingernail Clipper

The results in Figure 4.2 reveal that ***B*$_{1,1}$** is analyzable but ***B*$_{2,2}$** is not. Since all the entities except the clipper are essentially rigid, and this compliance has already been modeled, it does not make sense to add more compliance to try and satisfy Equation 4.5. Therefore, one of the DOF should be converted into a DOC so that the excess under-constraint is eliminated. As a guide, **[*TLr*]$_{2,2}$** is shown in Figure 4.3 (**[*WPr*]$_{2,2}$** is an empty matrix since ***ce*$_{i,j}$** equals zero).



F $_{10}$ is the motion allowed between the fulcrum end of the handle and the upper surface of the cutter in the θX direction. Configuration ***B*** $_{1,1}$ is defined such that the nominal value of ***F*** $_{10}$ is -1.57 radians since this is the point at which the cutter blades are closest to touching. Thus for analysis purposes, ***F*** $_{10}$ can be converted into a DOC.

$$[TLr]_{2,2} = \begin{bmatrix}0\\-1\\0\\1\\0\\2.96\end{bmatrix}\begin{matrix}F_7\}M_{2,2,1}\\F_8\}M_{2,2,2}\\\left.\begin{matrix}F_9\\F_{10}\\F_{11}\\F_{12}\end{matrix}\right\}M_{2,2,3}\end{matrix}$$

Figure 4.3. Identifying and Eliminating the Under-Constraint in Configuration *B*$_{2,2}$

From **[*TLr*]**$_{2,2}$, it is clear that the excess under-constraint is a correlated combination of several DOF. However, looking at the fourth entry in **[*TLr*]**$_{2,2}$, it is seen that the under-constraint could be eliminated by removing ***F*** $_{10}$, the motion allowed between the fulcrum end of the handle and the upper surface of the cutter in the ***θx*** direction. Since configuration ***B*** $_{2,2}$ is defined such that the nominal value of ***F*** $_{10}$ is -1.57 radians (this is the point at which the cutter blades are closest to touching during the opening process), this seems like a logical choice. Thus for analysis purposes, ***F*** $_{10}$ can be converted into a DOC with a direction ***θx*** and nominal value -1.57. Making this change yields the constraint analysis results shown in Figure 4.4 and it is seen that now that ***B*** $_{2,2}$ is analyzable.

Configuration $B_{2,2}$

$$ft_{2,2} = 5, ct_{2,2} = 18, st_{2,2} = 1,$$
$$\lambda_{2,2} = 1, g_{2,2} = 2, p_{2,2} = 4,$$
$$r1_{2,2} = 18, r2_{2,2} = 18$$

$$fe_{2,2} = 6\left(p_{2,2} - 1\right) - r2_{2,2} = 6\left(4 - 1\right) - 18 = 0$$

and

$$ce_{2,2} = 6\left(\lambda_{2,2} + g_{2,2}\right) - r1_{2,2} = 6\left(1 + 2\right) - 18 = 0$$

or

so

$$fe_{2,2} + ce_{2,2} =$$

is

$$\boxed{0 + 0 = 0}$$

0 Excess Under-Constraints & 0 Excess Over-Constraints

Figure 4.4. Constraint Analysis After Eliminating the Under-Constraint in *B*$_{2,2}$

Using Matlab® programming language, an executable called constraintdetector.m was created that automatically calculates the number of excess under-constraints (***fe***$_{i,j}$) and excess over-constraints (***ce***$_{i,j}$) and provides the reduces twist mate ([***TLr***]$_{i,j}$) and reduced wrench part ([***WPr***]$_{i,j}$) matrices for a single configuration (***B***$_{i,j}$). Constraintdetector.m takes as inputs the vectors and matrices shown in Table 4.1 and provides as outputs, the matrices shown in Table 4.2. A copy of constraintdetector.m is included in Appendix A. It can be run on Release 13 of Matlab®.

Matrix or Variable	**Description**	**Size**
$[TL]_{i,j}$	Twist Loop Matrix for $B_{i,j}$	$(\{6\lambda_{i,j}+6g_{i,j}\}$ x $\{ft_{i,j}+st_{i,j}+6g_{i,j}\})$
$[WP]_{i,j}$	Wrench Part Matrix for $B_{i,j}$	$(\{6p_{i,j}-6\}$ x $\{ct_{i,j}+st_{i,j}\})$
g	# of Open-Loops for $B_{i,j}$	$(1$ x $1)$
$[SK]_{i,j}$	Nominal Stiffness Matrix for $B_{i,j}$	$(st_{i,j}$ x $st_{i,j})$

Table 4.1. Inputs for Matlab® Executable constraintdetector.m

Matrix or Variable	**Description**	**Size**
$fe_{i,j}$	# of Excess Under-Constraints for $B_{i,j}$	$(1$ x $1)$
$ce_{i,j}$	# of Excess Over-Constraints for $B_{i,j}$	$(1$ x $1)$
$[TLr]_{i,j}$	Reduced Twist Loop Matrix for $B_{i,j}$	$(ft_{i,j}$ x $fe_{i,j})$
$[WPr]_{i,j}$	Reduced Wrench Part Matrix for $B_{i,j}$	$(ct_{i,j}$ x $ce_{i,j})$

Table 4.2. Outputs for Matlab® Executable constraintdetector.m

4.2 Sensitivity Analysis

Assuming Equation 4.5 is true, Equation 3.42 can be solved for $[\Psi]_{i,j}$ in terms of $[\Delta]_{i,j}$ as follows (adapted from [Gao, Chase, and Magleby]):

$$
\begin{aligned}
&[\psi]_{i,j} = [Sm]_{i,j}\,[\Delta]_{i,j} \\
&\mathit{where} \\
&\left[Sm_{((6m_{i,j}+st_{i,j}+6g_{i,j})\times dt_{i,j})}\right]_{i,j} = -[B]_{i,j}^{-1}[U]_{i,j} \\
&\mathit{for} \\
&i = 1,2,\ldots,a \\
&j = 1,2,\ldots,b_i
\end{aligned}
$$

Equation 4.8

$[Sm]_{i,j}$ is the mate-sensitivity matrix. Entry $[Sm_{k,l}]_{i,j}$ is the normalized sensitivity of k^{th} DOF, DOV, DOC or DOS of the mates (each DOS has two rows in $[Sm_{k,l}]_{i,j}$, one for the motion allowed and one for the force transmitted) to a unit increase in the nominal value of the $\boldsymbol{l}^{\text{th}}$ dimension. To determine the sensitivity of the KCs, one must express the changes in the KCs as linear combinations of the changes in displacement or force allowed by the DOF, DOV, DOS, or DOC. If the existing mates do not suffice, then appropriate virtual mates should be added and the DOV of those virtual mates used. Using this approach, the changes in the KCs for an arbitrary configuration, $\boldsymbol{B}_{i,j}$, can be expressed as a $(\boldsymbol{kct}_{i,j}$ **x** $\{6\boldsymbol{m}_{i,j}+\boldsymbol{st}_{i,j}+\boldsymbol{6g}_{i,j}\})$ matrix, $[\boldsymbol{KC}]_{i,j}$, called the KC combination matrix. Entry $[\boldsymbol{KC}_{k,l}]_{i,j}$ is "+1" if the difference between the specification limit and the nominal value of the $\boldsymbol{k}^{\text{th}}$ KC is decreased by an positive change in the $\boldsymbol{l}^{\text{th}}$ DOF, DOV, DOC, or DOS, is "-1" if the difference between the specification limit and the nominal value of the $\boldsymbol{k}^{\text{th}}$ KC is increased by an positive change, and is "0" if the nominal value of the $\boldsymbol{k}^{\text{th}}$ KC is unaffected by a change in the $\boldsymbol{l}^{\text{th}}$ DOF, DOV, DOC, or DOS. Now, a $(\boldsymbol{kct}_{i,j}$ **x** $\boldsymbol{dt}_{i,j})$ matrix $[\boldsymbol{KCS}]_{i,j}$, called the configuration-level KC sensitivity matrix, can be obtained for an arbitrary configuration, $\boldsymbol{B}_{i,j}$, from the following equation:

$$\left[KCS_{\left(kct_{i,j}\times dt_{i,j}\right)}\right]_{i,j}=\left[KC_{\left(kct_{i,j}\times\left(6m_{i,j}+st_{i,j}+6g_{i,j}\right)\right)}\right]_{i,j}\left[Sm_{\left(\left(6m_{i,j}+st_{i,j}+6g_{i,j}\right)\times dt_{i,j}\right)}\right]_{i,j}$$
$$for$$
$$i=1,2,\mathrm{K},a$$
$$j=1,2,\mathrm{K},b_i$$

Equation 4.9

Entry $[\boldsymbol{KCS}_{k,l}]_{i,j}$ is the first-order sensitivity of the $\boldsymbol{k}^{\text{th}}$ KC in a configuration to a positive increase in the nominal value of the $\boldsymbol{l}^{\text{th}}$ dimension in that same configuration. Figure 4.5 shows $[\boldsymbol{KC}]_{1,1}$, $[\boldsymbol{KC}]_{2,2}$, $[\boldsymbol{KCS}]_{1,1}$, and $[\boldsymbol{KCS}]_{2,2}$, for the fingernail clipper example.

$$[KC]_{1,1} = \begin{array}{c|c} \left[\begin{array}{cccccccccccccccccc} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] & \begin{array}{c} KC_1 \\ KC_2 \\ KC_3 \\ KC_4 \end{array} \\ \begin{array}{cccccccccccccccccc} F_1 & F_2 & F_3 & F_4 & F_5 & F_6 & C_1 & C_2 & C_3 & C_4 & C_5 & C_6 & C_7 & C_8 & C_9 & C_{10} & C_{11} & C_{12} \end{array} & \end{array}$$

$$[KCS]_{1,1} = \begin{array}{c|c} \left[\begin{array}{ccccc} 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{array}\right] & \begin{array}{c} KC_1 \\ KC_2 \\ KC_3 \\ KC_4 \end{array} \\ \begin{array}{ccccc} D_1 & D_2 & D_3 & D_4 & D_5 \end{array} & \end{array}$$

KC₁, the distance in the ***z*** direction between the notch in the lower cutter and the notch in the upper cutter base is sensitive to ***D₄*** and ***D₅***, the thicknesses of the lower cutter and the upper cutter base

$$[KC]_{2,2} = \left[\begin{array}{c|cccccccccccc|c|c} & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & & KC_5 \\ & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & & KC_6 \\ & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & & KC_7 \\ [0_{(7\times6)}] & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & [0_{(7\times19)}] & KC_8 \\ & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & & KC_9 \\ & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & & KC_{10} \\ & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & & KC_{11} \\ F_7, F_8, F_9, F_{11}, F_{12}, S_1 & V_1 & V_2 & V_3 & V_4 & V_5 & V_6 & V_7 & V_8 & V_9 & V_{10} & V_{11} & V_{12} & C_{13} - C_{29}, F_{10}, S_1 & \end{array}\right]$$

KC₅ and ***KC₆***, the alignment in the ***x*** direction between the two cutter blades is sensitive to linking ***KC₄*** which is in turn sensitive ***D₂***, the alignment of the upper cutter base and lower cutter in the ***θz*** direction

$$[KCS]_{2,2} = \begin{array}{c|c} \left[\begin{array}{cccccccccccccc} 0 & 1 & 0 & 40.34 & 0 & 0 & 0 & 0 & 0 & 0 & .105 & 0 & 0 & 0 \\ 0 & -1 & 0 & -40.34 & 0 & 0 & 0 & 0 & 0 & 0 & -.105 & 0 & 0 & 0 \\ -.009 & 0 & .999 & 0 & -1 & 0 & 0 & 0 & .997 & .077 & 0 & .001 & .009 & .009 \\ .009 & 0 & -.999 & 0 & 1 & 0 & 0 & 0 & -.997 & -.077 & 0 & -.001 & -.009 & -.009 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & .997 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & -.997 & 0 & 0 & 0 \\ .132 & 0 & .088 & 0 & 0 & 1 & 0 & 0 & 0 & -.997 & 0 & -.088 & -1.132 & -1.136 \end{array}\right] & \begin{array}{c} KC_5 \\ KC_6 \\ KC_7 \\ KC_8 \\ KC_9 \\ KC_{10} \\ KC_{11} \end{array} \\ \begin{array}{cccccccccccccc} KC_1 & KC_2 & KC_3 & KC_4 & D_6 & D_7 & D_8 & D_9 & D_{10} & D_{11} & D_{12} & D_{13} & D_{14} & D_{15} \end{array} & \end{array}$$

Figure 4.5. The KC Combination and KC Sensitivity Matrices for *B₁,₁* & *B₂,₂*

It can be seen that ***KC₁***, the distance in the ***z*** direction between the notch in the lower cutter and the notch in the upper cutter base, is sensitive to ***D₄*** and ***D₅***, the thicknesses of the lower cutter and the upper cutter base. ***KC₁₁***, the clearance in the ***z*** direction between the two cutter blades, is sensitive to $\boldsymbol{D_{14}}$ and $\boldsymbol{D_{15}}$, the height of the pin and the length of the fulcrum arm of the handle respectively. $\boldsymbol{KC_5}$ and $\boldsymbol{KC_6}$, the alignment in the $\boldsymbol{x}$ direction between the two cutter blades, is sensitive to linking $\boldsymbol{KC_4}$ which is in turn sensitive to $\boldsymbol{D_2}$, the alignment of the upper cutter base and lower cutter in the $\boldsymbol{\theta z}$ direction.

The sensitivities of the KCs of configuration $\boldsymbol{B_{i,j}}$, to the dimensions of a configuration $\boldsymbol{B_{k,l}}$ (and $\boldsymbol{k}$ must be less than $\boldsymbol{i}$), where $\boldsymbol{B_{i,j}}$ and $\boldsymbol{B_{l,k}}$ are related by $\boldsymbol{kcl_{i,j\rightarrow k,l}}$ linking KCs, can be contained in a **(**$\boldsymbol{kct_{i,j}}$ **x** $\boldsymbol{dt_{k,l}}$**)** matrix, **[**$\boldsymbol{KCS}$**]**$_{\boldsymbol{i,j\rightarrow k,l}}$, called the configuration linking matrix for those two configurations, that is computed as follows:

$$\left[KCS_{\left(kct_{i,j}\times dt_{k,l}\right)}\right]_{i,j\rightarrow k,l}=\left[KCS_{1-kct_{i,j},1-kcl_{ij\rightarrow k,l}}\right]_{i,j}\left[KCS_{1-kcl_{ij\rightarrow k,l},1-dt_{k,l}}\right]_{k,l}$$

$$\begin{aligned}&where\\&i,k=1,2,\ldots,a-1\\&j,l=1,2,\ldots,b_i,b_k\\&k<i\end{aligned}$$

Equation 4.10

The set of matrices obtained by repetitious use of Equation 4.10 can be compiled into a single **(*kcp* x *dtt*)** matrix, **[*KCS*]**, called the performance KC sensitivity matrix. Entry **[**$\boldsymbol{KCS_{i,j}}$**]** is the first-order sensitivity of the ***i***th performance KC of the assembly to a positive increase in the value of the ***j***th dimension of the assembly.

There are three possibilities for the units of entry **[**$\boldsymbol{KCS_{i,j}}$**]**. If the ***i***th dimension and the ***j***th KC are both linear distances or they are both angles, then **[**$\boldsymbol{KCS_{i,j}}$**]** is unitless. If the ***i***th dimension is an angle expressed in radians and the $\boldsymbol{j}^{\text{th}}$ KC is a linear distance, then **[*KCS$_{i,j}$*]** has units of length to the minus one (e.g. 1/mm). If the $\boldsymbol{i}^{\text{th}}$ dimension is a linear distance and the $\boldsymbol{j}^{\text{th}}$ KC is an angle expressed in radians, then **[*KCS$_{i,j}$*]** has units of length (e.g. mm).

Figure 4.6 shows **[*KCS*]*$_{2,2\rightarrow 1,1}$*** and **[*KCS*]** for the fingernail clipper example obtained using Equation 4.10. ***KC$_7$*** and ***KC$_8$***, the alignment in the ***y*** direction (positive and negative sense respectively) between the two cutter blades, are sensitive to ***D$_6$,*** the distance in the ***y*** direction between the lower notch in the cutter base and the lower blade on the cutter base. The sign are opposite for the two sensitivities because ***KC$_7$*** has an upper specification limit and ***KC$_8$*** has a lower specification limit.

$$[KSC]_{2,2\rightarrow 1,1} = \begin{array}{cccccc} \left[\begin{array}{ccccc} 1 & 0 & 40.34 & 0 & 0 \\ -1 & 0 & -40.34 & 0 & 0 \\ 0 & .999 & 0 & -.009 & -.009 \\ 0 & -.999 & 0 & .009 & .009 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & .088 & 0 & .132 & .132 \end{array}\right] & \begin{array}{c} KC_5 \\ KC_6 \\ KC_7 \\ KC_8 \\ KC_9 \\ KC_{10} \\ KC_{11} \end{array} \\ \begin{array}{ccccc} D_1 & D_2 & D_3 & D_4 & D_5 \end{array} & \end{array}$$

KC$_7$ and ***KC$_8$***, the alignment in the ***y*** direction between the two cutter blades are sensitive to ***D$_6$,*** the distance in the ***y*** direction between the lower notch in the cutter base and and the lower blade on the cutter base. The sign are opposite for the two sensitivities because ***KC$_7$*** has an upper specification limit and ***KC$_8$*** has a lower specification limit.

$$[KCS] = \begin{array}{cc} \left[\begin{array}{ccccccccccccccc} 1 & 0 & 40.34 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & .105 & 0 & 0 & 0 \\ -1 & 0 & -40.34 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -.105 & 0 & 0 & 0 \\ 0 & .999 & 0 & -.009 & -.009 & -1 & 0 & 0 & 0 & .997 & .077 & 0 & .001 & .009 & .009 \\ 0 & -.999 & 0 & .009 & .009 & 1 & 0 & 0 & 0 & -.997 & -.077 & 0 & -.001 & -.009 & -.009 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & .997 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -.997 & 0 & 0 & 0 \\ 0 & 0 & 0 & .132 & .132 & 0 & 1 & 0 & 0 & .077 & -.997 & 0 & -.088 & -1.13 & -1.13 \end{array}\right] & \begin{array}{c} KC_5 \\ KC_6 \\ KC_7 \\ KC_8 \\ KC_9 \\ KC_{10} \\ KC_{11} \end{array} \\ \begin{array}{ccccccccccccccc} D_1 & D_2 & D_3 & D_4 & D_5 & D_6 & D_7 & D_8 & D_9 & D_{10} & D_{11} & D_{12} & D_{13} & D_{14} & D_{15} \end{array} & \end{array}$$

Figure 4.6. The Configuration Linking and Performance KC Sensitivity Matrices

4.3 Performance Analysis

Any dimension, $\boldsymbol{D_i}$, in an assembly has a PDF function $\boldsymbol{f_{Di}(\delta_i)}$ that gives the probability density of $\boldsymbol{\delta_i}$. Using these PDF functions, the set of all average values for the dimensions in an assembly can be contained in a **(*dtt* x *1*)** vector, **[*Da*]**. Entry **[$\boldsymbol{Da_i}$]** of this vector is computed as follows [Drake]:

$$
\begin{aligned}
&[Da_i] = \int_{-\infty}^{\infty} \delta_i \, f_{D_i}(\delta_i)\, d\delta_i \\
&for \\
&i = 1,2,\ldots,dtt
\end{aligned}
$$

Equation 4.11

Any two dimensions in an assembly, $\boldsymbol{D_i}$ and $\boldsymbol{D_j}$ also have a joint PDF, or JPDF, denoted generically as $\boldsymbol{f_{Di,Dj}(\delta_i ,\delta_j)}$. $\boldsymbol{f_{Di,Dj}(\delta_i ,\delta_j)}$ gives the joint probability density at $\boldsymbol{\delta_i}$ and $\boldsymbol{\delta_j}$. Using these functions, the variances and covariances of the dimensions in an assembly can be calculated and placed in a **(*dtt* x *dtt*)** matrix, **[*Dv*]**, called the dimensional covariance matrix. Entry **[$\boldsymbol{Dv_{i,j}}$]** of this matrix is computed as follows [Drake]:

$$
\begin{aligned}
&[Dv_{i,j}] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (\delta_i - [Da_i])(\delta_j - [Da_j])\, f_{D_i,D_j}(\delta_i,\delta_j)\, d\delta_i d\delta_j \\
&for \\
&i,j = 1,2,\ldots,dtt
\end{aligned}
$$

Equation 4.12

If $\boldsymbol{i}$ equals $\boldsymbol{j}$, then **[$\boldsymbol{Dv_{i,j}}$]** is the variance of the $\boldsymbol{i}^{\text{th}}$ dimension. If $\boldsymbol{i}$ does not equal $\boldsymbol{j}$, then **[$\boldsymbol{Dv_{i,j}}$]** is the covariance between the $\boldsymbol{i}^{\text{th}}$ dimension and $\boldsymbol{j}^{\text{th}}$ dimension. If all the dimensions are independent from each other, as is assumed in this work, then there are no covariances and Equation 4.12 simplifies as follows [Drake]:

$$[Dv_{i,j}] = \begin{cases} \int_{-\infty}^{\infty} (\delta_i - [Da_i])^2 f_{D_i}(\delta_i)\, d\delta_i \; if\ i = j \\ 0 \quad if\ i \neq j \end{cases}$$

for

$i, j = 1,2,\mathrm{K}, dtt$

Equation 4.13

Figure 4.7 shows **[*Da*]** and **[*Dv*]** for the fingernail clipper obtained using Equation 4.11 and Equation 4.13. Comparing **[*Dn*]** in Figure 3.14 to **[*Da*]** in Figure 4.7 reveals that ***D*$_3$**, ***D*$_7$**, and ***D*$_{12}$**, have mean shifts of +0.0004, +.02, and +.003 respectively. These mean shifts indicate that the average KC values will be different than the nominal values and this could adversely affect the manufacturability of the assembly.

$$[Da]^T = \begin{array}{ccccccccccccccc} 0 & 0 & .0004 & 1.36 & 1.36 & -40.34 & 2.83 & 0 & -.185 & -40.24 & -2.81 & .003 & -35.18 & 8.41 & -2.96 \\ D_1 & D_2 & D_3 & D_4 & D_5 & D_6 & D_7 & D_8 & D_9 & D_{10} & D_{11} & D_{12} & D_{13} & D_{14} & D_{15} \end{array}$$

$$[Dv] = \begin{array}{cccccccccccccccc} 9.0E-4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & D_1 \\ 0 & 2.6E-4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & D_2 \\ 0 & 0 & 1.2E-6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & D_3 \\ 0 & 0 & 0 & 1.0E-4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & D_4 \\ 0 & 0 & 0 & 0 & 3.6E-5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & D_5 \\ 0 & 0 & 0 & 0 & 0 & 4.0E-4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & D_6 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2.13E-3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & D_7 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.23E-5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & D_8 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0E-6 & 0 & 0 & 0 & 0 & 0 & 0 & D_9 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 7.8E-4 & 0 & 0 & 0 & 0 & 0 & D_{10} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.13E-3 & 0 & 0 & 0 & 0 & D_{11} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 7.84E-6 & 0 & 0 & 0 & D_{12} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.5E-3 & 0 & 0 & D_{13} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.6E-3 & 0 & D_{14} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.6E-3 & D_{15} \\ D_1 & D_2 & D_3 & D_4 & D_5 & D_6 & D_7 & D_8 & D_9 & D_{10} & D_{11} & D_{12} & D_{13} & D_{14} & D_{15} & \end{array}$$

Figure 4.7. [*Da*] and [*Dv*] for the Fingernail Clipper Assembly

The average values of the performance KCs, contained in a vector, **[*KCa*]**, can be obtained exactly by replacing **[*Dn*]** with **[*Da*]** and repeating the modeling process presented in Chapter 3. Alternatively, using **[*KCn*]**, **[*KCS*]**, **[*Da*]**, and **[*Dn*]**, the average values of the performance KCs can be approximated to first-order as follows:

$$\left[KCa_{(kcp\times 1)}\right]\approx\left[KCn_{(kcp\times 1)}\right]+\left[KCS_{(kcp\times dtt)}\right]\left(\left[Da_{(dtt\times 1)}\right]-\left[Dn_{(dtt\times 1)}\right]\right)$$

Equation 4.14

Figure 4.8 compares **[*KCa*]** for the fingernail clipper that was obtained exactly by replacing **[*Dn*]** with **[*Da*]** and repeating the procedure in Chapter 3 with **[*KCa*]** that was obtained approximately using Equation 4.14. The maximum error in this case is about three percent and that is for $\boldsymbol{KC_5}$ and $\boldsymbol{KC_6}$. In general, if any of the values in **[*KCa*]** are outside the specification limits, then this represents a serious problem since on average, the design is not delivering one of the KCs. Comparing **[*KCa*]** to **[*KCL*]** (obtained from Figure 3.36) reveals that for the fingernail clipper, this is not the case.

Exact Average Values $$[KCa]^T = \begin{array}{ccccccc} \lfloor .01607 & .01607 & .00314 & .00314 & .00339 & .00339 & .1916 \rfloor \\ KC_5 & KC_6 & KC_7 & KC_8 & KC_9 & KC_{10} & KC_{11} \end{array}$$

Approximate Avg. Values $$[KCa]^T \approx \begin{array}{ccccccc} \lfloor .01660 & .01660 & .00313 & .00313 & .00341 & .00341 & .1916 \rfloor \\ KC_5 & KC_6 & KC_7 & KC_8 & KC_9 & KC_{10} & KC_{11} \end{array}$$

% Error $$\begin{array}{ccccccc} \lfloor 3.19 & 3.19 & .319 & .319 & .590 & .590 & < .1 \rfloor \\ KC_5 & KC_6 & KC_7 & KC_8 & KC_9 & KC_{10} & KC_{11} \end{array}$$

Specification Limits $$[KCL]^T = \begin{array}{ccccccc} \lfloor .20 & -.20 & .15 & -.15 & .02 & -.02 & 0 \rfloor \\ KC_5 & KC_6 & KC_7 & KC_8 & KC_9 & KC_{10} & KC_{11} \end{array}$$

Nominal Values $$[KCn]^T = \begin{array}{ccccccc} KC_5 & KC_6 & KC_7 & KC_8 & KC_9 & KC_{10} & KC_{11} \\ \lfloor 0 & 0 & .003 & .003 & 0 & 0 & .214 \rfloor \end{array}$$

Figure 4.8. The Approximate and Exact Values for the Average KC Values

For a single assembly that is fabricated and assembled, the difference between the actual values for the dimensions of the parts in that assembly and the nominal set of values for those same dimension can be contained in a **(*dtt* x *1*)** vector, **[Δ]**, and actual values for the performance KCs can be contained in a **(*kcp* x *1*)** vector, **[*KCo*]**. Entry **[Δ_j]** is the actual value minus the nominal value of the ***j***th dimension. Entry **[*KCo*$_i$]** is the actual value of the ***i***th performance KC. One approach to obtain **[*KCo*]**, is to replace **[*Dn*]** with (**[*Dn*]**+**[Δ^i])** and reapply the methods presented in Chapter 3 using the new vector of dimensions. However, using **[*KCS*]**, **[*KCo*]** can also be approximated as follows:

$$\left[KCo_{(kcp\times 1)}\right] \approx \left[KCn_{(kcp\times 1)}\right] + sign\left(\left[KCL_{(kcp\times 1)}\right] - \left[KCn_{(kcp\times 1)}\right] \right) \left[KCS_{(kcp\times dtt)}\right]\left[\Delta_{(dtt\times 1)}\right]$$

Equation 4.15

One way to predict how much variability the KCs might exhibit in an entire population of assemblies is to use Monte Carlo simulation to generate a large set of ***n*** random vectors **[Δ^1]**, **[Δ^2]**, …,**[Δ^n]**, (they can be generated via computer methods using the PDFs for each dimension) and use the modeling process in Chapter 3 after substituting **[*Dn*]** with (**[*Dn*]**+ **[Δ]**) for each vector. Then the resulting KC values, **[*KCo^1*]**, **[*KCo^2*]**, …,**[*KCo^n*]** could plotted on ***kcp*** histograms, one for each performance KC. From these histograms could be extracted the Cpk, or process capability index, and reject rate for each KC. One such approach is presented by [Whitney]. While effective, this method can be very computationally intensive and doesn't directly provide any information regarding the sensitivities of the KCs to the dimensions.

Another approach is to linearize the behavior of the assembly using Taylor series expansion (in this case, [***KCS***] would be the linearized behavior of the assembly). Then the PDFs and CDFs of the dimensions could be approximated with a set of first moments (or means) contained in **[Δ*1*]**, and second moments (or standard deviations in this case) **[Δ*2*]**. Then **[Δ]** can be replaced by (**[Δ*1*]**+ **[Δ*2*]**) in Equation 3.15. The resulting [***KCo***] would be what is called a "worst-case" set of offsets of the KCs from their nominal values and could be used to estimate the Cpks and reject rates for the KCs (a root-sum-squares, or RSS, version of this method also exists). [Parkinson, Sorenson, and Pourhoussan] present one such approach. Such approaches are efficient but produce inaccurate results when the standard deviations are large or the dependence of the KCs on the dimensions is highly nonlinear.

In this work a different approach is presented similar to that used in [Parameshaven and Kjerengtroen]. To understand this approach, the concept of a "**Most Probable Point**, or MPP, originally developed in the area of structural reliability analysis and presented by [Melchers] and others, must be explained. Each assembly has ***kcp*** MPPs, one for each performance KC. The location of the $\boldsymbol{i}^{th}$ MPP is described by a **(*dtt* x *1*)** vector, **[Δ*]*$_i$***, which is found by solving the following optimization problem:

$$
\begin{aligned}
&\textit{Find } \left[\Delta^*\right]_i \textit{ that maximizes } \beta_i\\
&\textit{subject to}\\
&\left[KCo_i^{\;*}\right] = \left[KCL_{\;i}\right]\\
&\textit{where}\\
&\beta_i = \text{Probability}\left(\left[\Delta\;\right] = \left[\Delta^*\right]_i\right)\\
&\left[KCo_i^{\;*}\right] \approx \left[KCn_{\;i}\right] + sign\left(\left[KCL_{\;i}\right] - \left[KCn_{\;i}\right]\right)\left[KCS_{\;i,1:dtt}\right]\left[\Delta^*\right]_i\\
&\textit{for}\\
&i = 1,2,\ldots,kcp
\end{aligned}
$$

Equation 4.16

[*KCo*] can either be obtained using the approximation in shown in Equation 4.15 or it can be obtained by replacing **[*Dn*]** with (**[*Dn*]**+**[Δ]**) and reapplying the methods presented in Chapter 3. Conceptually, **[Δ*]*$_i$*** describes the set of actual dimension offsets from nominal that has both the highest probability density and results in an assembly that is on the failure limit for the ***i***th KC. For any assembly there exists ***kcp*** vectors, $[\mathbf{\Delta}^{*}]_{1}$, $[\mathbf{\Delta}^{*}]_{2}$,…, $[\mathbf{\Delta}^{*}]_{kcp}$ that describe the locations of the ***kcp*** MPPs.

In practice, the optimization problem presented in Equation 4.16 can be difficult to solve. However, one approach is presented here using what is called a standard normal space transformation. To overcome the challenge of working with many different distribution types, the CDF, $\boldsymbol{F_{D\,i}(\delta_i)}$, for an arbitrary dimension, $\boldsymbol{D_i}$, can be mapped point by point onto the standard normal CDF, $\boldsymbol{\Phi(\gamma_i)}$, (a standard normal distribution is a normal distribution with a mean equal to zero and a standard deviation equal to one), by using the following "**Quantile-Quantile**" relationship [Rosenblatt]:

$$F_{D_i}\left(\,[Dn_i]+[\Delta_i]\,\right)=\Phi\left(\,[\gamma_i]\,\right)$$
$$\textit{for any } [\Delta_i]$$
$$\textit{where } \text{Probability}\left(\,D_i=[Dn_i]+[\Delta_i]\,\right)>0$$

Equation 4.17

Based on Equation 4.17, and assuming the variables are independent, the following two relationships exist between an arbitrary value, ($[\boldsymbol{Dn_i}]$+$[\boldsymbol{\Delta_i}]$), in the original random variable space and its corresponding value, $[\boldsymbol{\gamma_i}]$, in the standard normal space [Melchers]:

$$[\gamma_i]=\Phi^{-1}\left(F_{D_i}\left(\,[Dn_i]+[\Delta_i]\,\right)\right)$$
$$\textit{and}$$
$$[\Delta_i]=F_{D_i}^{-1}(\Phi\left(\,[\gamma_i]\,\right))-[Dn_i]$$

Equation 4.18

Now **[Δ *]**$_1$, **[Δ *]**$_2$, …,**[Δ *]**$_{kcp}$, can be found in a much simpler way by working in the standard normal space where all variables have equal weight. Once in the variables are in the standard normal space, the optimization problem presented in Equation 4.16 can be reformulated as follows (adapted from [Ditlevsen and Madsen]):

$$
\begin{aligned}
&Find\ \left[\Delta^*\right]_i\ that\ maximizes\ \ \beta_i \\
&subject\ \ to \\
&\left[KCo\ _i{}^*\right] = \left[KCL\ _i\right] \\
&where \\
&\ \beta_i = \left(\ [\gamma]_i^T [\gamma]_i\right)^{\frac{1}{2}} \\
&\left[KCo_i{}^*\right] \approx \left[KCn\ _i\right] + sign\left(\left[KCL\ _i\right] - \left[KCn\ _i\right]\right)\left[KCS_{\,i,1:dtt}\right]\left[\Delta^*\right]_i \\
&\left[\gamma_j\right]_i = \Phi^{-1}\left(F_{D_j}\left(\left[\Delta_j\right]_i\right)\right) \\
&\left[\Delta_j\right]_i = F_{D_j}^{-1}\left(\Phi\left(\left[\gamma_j\right]\right)\right) - \left[Dn_j\right]_i \\
&for \\
&i = 1,2,...,\ kcp \\
&j = 1,2,...,\ dtt
\end{aligned}
$$

Equation 4.19

The following iterative algorithm can be used to solve Equation 4.19 (adapted from [Ditlevsen and Madsen]):

$$
\begin{aligned}
&[\gamma]_i^{(n+1)} = \left([\gamma]_i^{(n)} \cdot [\alpha]_i^{(n)}\right)[\alpha]_i^{(n)} + \frac{[KCo_i]^{(n)} - [KCL_i]^{(1)}}{\left\| [\nabla Q]^{(n)} \right\|}[\alpha]_{i_n}^{(n)} \\
&\textit{where} \\
&[\gamma_j]_i^{(n)} = \Phi^{-1}\left(F_{D_j}\left([\Delta_j]_i^{(n)}\right)\right) \\
&[\nabla Q_j]^{(n)} = \frac{[KCS_{i,j}]^{(n)}[\mu_j]_i^{(n)}}{[\nu_j]_i^{(n)}} \\
&[\alpha]_i^{(n)} = -\frac{[\nabla Q]^{(n)}}{\left\| [\nabla Q]^{(n)} \right\|} sign\left([KCL_i]^{(1)} - [KCn_i]^{(1)}\right) \\
&[u_j]_i^{(n)} = f_\Phi\left([\gamma_j]_i^{(n)}\right) \\
&[v_j]_i^{(n)} = f_{D_j}\left([Dn_j] + [\Delta_j]_i^{(n)}\right) \\
&\textit{for} \\
&i = 1,2,\ldots, kcp \\
&j = 1,2,\ldots, dtt
\end{aligned}
$$

Equation 4.20

If the iterative algorithm presented in Equation 4.20 is used to solve Equation 4.19, then **[*KCo*]**$^{(n)}$ and **[*KCS*]**$^{(n)}$ and should be updated and replaced by **[*KCo*]**$^{(n+1)}$ and **[*KCS*]**$^{(n+1)}$ before beginning the next iteration, thus capturing any non-linearities in the dependencies of the KCs and their sensitivities on the dimensions. This is done by replacing $[\Delta_j]_i^{(n)}$ with $[\Delta_j]_i^{(n+1)}$ and following the steps in Chapter 3 and the beginning of this chapter to find **[*KCo*]**$^{(n+1)}$ and **[*KCS*]**$^{(n+1)}$ (see Equation 4.9, Equation 4.10, and Equation 4.15). $[\Delta_j]_i^{(n+1)}$ can be obtained as follows:

$$
\left[\Delta_j\right]_i^{(n+1)} = F_{D_j}^{-1}\left(\Phi\left(\left[\gamma_j\right]_i^{(n+1)}\right)\right) - \left[Dn_j\right]_i
$$

$$
\begin{aligned}
&for\\
&i = 1,2,\ldots,\ kcp\\
&j = 1,2,\ldots,\ dtt
\end{aligned}
$$

Equation 4.21

For the $\boldsymbol{i}^{\text{th}}$ performance KC, the minimum value of $\boldsymbol{\beta_i}$ Equation 4.19, called $\boldsymbol{\beta_i}^*$, is the distance in the standard normal space from the nominal assembly state to the point on the $\boldsymbol{i}^{\text{th}}$ failure limit state (**[*KCo*$_i$] = [*KCL*$_i$]**) that has the highest probability density and is a measure of the manufacturability of the assembly with respect to the $\boldsymbol{i}^{\text{th}}$ KC. The larger $\boldsymbol{\beta_i}^*$ is, the higher the Cpk and the lower the reject rate of the $\boldsymbol{i}^{\text{th}}$ KC will be. $[\boldsymbol{\gamma}^*]_i$ is a vector with length $\boldsymbol{\beta}^*_i$ that points in the standard normal space to the MPP and its tail is located at the origin in the standard normal space. $[\boldsymbol{\gamma}^*]_i$ is the projection of the vector with length $\boldsymbol{\beta}^*_i$ onto the axis of the $\boldsymbol{j}^{\text{th}}$ dimension in the standard normal space. Assuming the failure limit state (**[*KCo*$_i$] = [*KCL*$_i$]**) is linear (according to the First-Order Reliability Method, or "**FORM**" [Melchers]), the set of Cpks for the performance KCs can be contained in a **(*kcp* x *1*)** vector **[*Cpk*]**, called the process capability matrix, and the reject rates with respect to the performance KCs can contained in a **(*kcp* x *1*)** vector **[*Rej*]**, called reject rate matrix, where entry **[*Cpk*$_i$]** and entry **[*Rej*$_i$]** are obtained respectively from the following equations (adapted from [Ditlevsen and Madsen]):

$$
[Cpk_i] = \begin{cases} 0 \;\; if \; q_1 + q_2 > q_3 \;\&\; q_1 > q_2 \\ otherwise \approx \dfrac{\beta_i^*}{3} \end{cases}
$$

$$
\begin{aligned}
&where \\
&q_1 = Abs\left([KCa_i] - [KCn_i] \right) \\
&q_2 = Abs\left([KCa_i] - [KCL_i] \right) \\
&q_3 = Abs\left([KCL_i] - [KCn_i] \right) \\
&for \\
&i = 1,2,..., kcp
\end{aligned}
$$

Equation 4.22

$$
[Rej_i] \approx \begin{cases} \approx 100\left(1 - \left(\Phi\left(-\beta_i^* \right) \right) \right) \;\; if \; q_1 + q_2 > q_3 \;\&\; q_1 > q_2 \\ otherwise \approx 100\left(\Phi\left(-\beta_i^* \right) \right) \end{cases}
$$

$$
\begin{aligned}
&where \\
&q_1 = Abs\left([KCa_i] - [KCn_i] \right) \\
&q_2 = Abs\left([KCa_i] - [KCL_i] \right) \\
&q_3 = Abs\left([KCL_i] - [KCn_i] \right) \\
&for \\
&i = 1,2,..., kcp
\end{aligned}
$$

Equation 4.23

The “**if**” statements are needed in case the average values of any of the KCs are outside their corresponding specification limits. If that is the case, those Cpks are automatically zero and the reject rate will now be one minus the result obtained from solving Equation 4.19. In general, a manufacturable design is one whose values in **[*Cpk*]** are all above a target value, generically noted here as ***KCT*** (which is often 1.33 in practice).

Figure 4.9 shows **[*Cpk*]** and **[*Rej*]** for the fingernail clipper that was obtained by solving Equation 4.19 using the iterative algorithm shown in Equation 4.20 and entering the results in Equation 4.22 and Equation 4.23. ***KCT*** is set at 1.33 for the fingernail clipper. Relative to this target, it can be seen that only $\boldsymbol{KC_6}$ and $\boldsymbol{KC_{10}}$ exhibit an acceptable level of variability. The next chapter will present some remedies for improving this situation. To facilitate that process, it is helpful to obtain a few more results as shown in the next two sections.

$$[Rej]^T = \begin{bmatrix} .0073 & .0002 & .0092 & .0092 & .0190 & 2.9E-5 & 2.87 \\ KC_5 & KC_6 & KC_7 & KC_8 & KC_9 & KC_{10} & KC_{11} \\ (+x) & (-x) & (+y) & (-y) & (+\theta z) & (-\theta z) & (-z) \end{bmatrix}$$

$$[Cpk]^T = \begin{bmatrix} 1.27 & 1.58 & 1.25 & 1.25 & 1.18 & 1.67 & 0.633 \\ KC_5 & KC_6 & KC_7 & KC_8 & KC_9 & KC_{10} & KC_{11} \end{bmatrix}$$

$$[KCn]^T = \begin{bmatrix} 0 & 0 & .003 & .003 & 0 & 0 & .212 \\ KC_5 & KC_6 & KC_7 & KC_8 & KC_9 & KC_{10} & KC_{11} \end{bmatrix}$$

$$[KCL]^T = \begin{bmatrix} .20 & -.20 & .15 & -.15 & .02 & -.02 & 0 \\ KC_5 & KC_6 & KC_7 & KC_8 & KC_9 & KC_{10} & KC_{11} \end{bmatrix}$$

Figure 4.9. [*Rej*], [*Cpk*], [*Kcn*] and [*KCL*] for the Fingernail Clipper Assembly

4.4 Contribution Analysis

The contribution of the variability of each dimension to the variability of each performance KC of an assembly can complied into a matrix, **[*Co*]**, called the contribution matrix, where entry **[*Co*** $_{i,j}$**]** is computed as follows [Melchers]:

$$\left[Co_{i,j} \right] = 100 \left(\frac{\left(\left[\gamma_j^{*} \right]_i \right)^2}{\left(\beta_i^{*} \right)^2} \right)$$

$$\begin{aligned} &for \\ &i = 1,2,\ldots,kcp \\ &j = 1,2,\ldots,dtt \end{aligned}$$

Equation 4.24

Entry $[\boldsymbol{Co_{i,j}}]$ is the percentage contribution of variability in the $\boldsymbol{j}^{\text{th}}$ dimension in the assembly to variability in the $\boldsymbol{i}^{\text{th}}$ performance KC. $[\boldsymbol{Co_{i,j}}]$ does not represent the contribution of the $\boldsymbol{j}^{\text{th}}$ dimension to the reject rate of the assembly with respect to the $\boldsymbol{i}^{\text{th}}$ performance KC unless there is no mean shift associated with any of the dimensions. (mean shift can have a dramatic effect on the reject rates). Figure 4.10 shows the contribution matrix, **[*Co*],** for the fingernail clipper example.

$$[Co] = \begin{array}{ccccccccccccccc|c}
99.30 & 0 & .6921 & 0 & 0 & 0 & 0 & 0 & 0 & .0004 & 0 & .0095 & 0 & 0 & 0 & KC_5 \\
99.88 & 0 & .1078 & 0 & 0 & 0 & 0 & 0 & 0 & .0002 & 0 & .0096 & 0 & 0 & 0 & KC_6 \\
0 & 1.045 & .0001 & .0007 & .0002 & 32.98 & 0 & 0 & 0 & 64.25 & 1.70 & 0 & .0001 & .0108 & .0108 & KC_7 \\
0 & 1.044 & 0 & .0005 & .0002 & 32.98 & 0 & 0 & 0 & 64.26 & 1.70 & 0 & .0001 & .0083 & .0083 & KC_8 \\
0 & 0 & 4.427 & 0 & 0 & 0 & 0 & 58.41 & 0 & 0 & 0 & 37.16 & 0 & 0 & 0 & KC_9 \\
0 & 0 & 1.404 & 0 & 0 & 0 & 0 & 60.41 & 0 & 0 & 0 & 38.34 & 0 & 0 & 0 & KC_{10} \\
0 & .0003 & 0 & .023 & .008 & 0 & 24.03 & 0 & 0 & .0551 & 23.89 & 0 & .227 & 25.79 & 25.94 & KC_{11} \\
\hline
D_1 & D_2 & D_3 & D_4 & D_5 & D_6 & D_7 & D_8 & D_9 & D_{10} & D_{11} & D_{12} & D_{13} & D_{14} & D_{15} &
\end{array}$$

Figure 4.10. The Contribution Matrix for the Fingernail Clipper Assembly

Looking at the contribution results in Figure 4.10, it can be seen that the variability in $\boldsymbol{D_1}$ **(**the alignment of the upper cutter base and lower cutter in $\boldsymbol{x}$ direction) is most responsible for the variability in $\boldsymbol{KC_5}$ and $\boldsymbol{KC_6}$ (the alignment of the cutter blades in the $\boldsymbol{x}$

direction). $\boldsymbol{D_6}$ and $\boldsymbol{D_{10}}$ (the distance in the ***y*** direction to the cutter blades from the notch in the cutter base) are the biggest contributors to the unacceptable levels of variability in $\boldsymbol{KC_7}$ and $\boldsymbol{KC_8}$ (the alignment of the cutter blades in the ***y*** direction), $\boldsymbol{D_8}$ and $\boldsymbol{D_{12}}$ **(**the rotation of the cutter blades in the $\boldsymbol{\theta z}$ direction) are the biggest contributors to the levels of variability in $\boldsymbol{KC_9}$ and $\boldsymbol{KC_{10}}$ (the alignment of the cutter blades in the $\boldsymbol{\theta z}$ direction), and $\boldsymbol{D_7}$ and $\boldsymbol{D_{11}}$ **(**the length of the cutter blades in the ***z*** direction), together with $\boldsymbol{D_{14}}$ and $\boldsymbol{D_{15}}$ (the length of the pin and the fulcrum arm of the handle respectively) are the biggest contributors to the unacceptable levels of variability in $\boldsymbol{KC_{11}}$ (the clearance of the cutter blades in the ***z*** direction).

4.5 Coupling Analysis

When there are multiple performance KCs, one is often interested to know whether there is any correlation between them. [Arora], [Suh], and [Whitney, Mantripragada, Adams, and Cunningham] all present methods for both detecting such correlation and eliminating it (or at least minimize its effect). Such correlation can be contained in two **(*kcp* x *kcp*)** matrices called the KC covariance matrix and the KC correlation matrix. The KC covariance matrix, **[*KCv*]**, can be obtained as follows [Melchers]:

$$\left[KCv_{(kcp\times kcp)}\right] = \left[KCS_{(kcp\times dtt)}\right]\left[Dv_{(dtt\times dtt)}\right]\left[KCS_{(kcp\times dtt)}\right]^{T}$$

Equation 4.25

Entry **[$KCv_{i,j}$]** is the covariance between the i^{th} and j^{th} KCs and entry **[$KCv_{i,i}$]** is the variance of the i^{th} KC. Using **[*KCv*]**, the KC correlation matrix **[*KCc*]**, can be also be obtained where entry **[$KCc_{i,j}$]** is computed as follows [Melchers]:

$$
\left[KCc_{i,j}\right] = \frac{\left[KCv_{i,j}\right]}{\left(\left(\left[KCv_{i,i}\right]\right)^{\frac{1}{2}}\left(\left[KCv_{j,j}\right]\right)^{\frac{1}{2}}\right)}
$$

$$
\begin{aligned}
&for \\
&i, j = 1,2,\ldots,kcp
\end{aligned}
$$

Equation 4.26

If entry **[$KCc_{i,j}$]** is positive, dimensional errors (such as mean shift) that cause the nominal value of the i^{th} KC to approach its specification limit will more than likely cause the nominal value of the j^{th} KC to approach its specification limit. If entry **[$KCc_{i,j}$]** is negative, dimensional errors that cause the nominal value of the i^{th} KC to approach its specification limit will more than likely cause the nominal value of the j^{th} KC to move away from its specification limit. If entry **[$KCc_{i,j}$]** is zero, then the two KCs are independent of each other with respect to all the dimensions.

Figure 4.11 shows **[*KCv*]** and **[*KCc*]** for the fingernail clipper assembly. In addition to the obvious coupling of KCs that are the upper and lower limit of the same one-dimensional geometric relationship (i.e. KC_5 and KC_6, KC_7 and KC_8, and KC_9 and KC_{10}), KC_5 and KC_6 are both correlated with KC_9 and KC_{10}. KC_7 and KC_8 are also both correlated with KC_{11}.

$$[KCv] = \left(10^{-4}\right) \begin{array}{cc} \left[\begin{array}{ccccccc} 28.53 & -28.53 & 0 & 0 & .493 & -.493 & 0 \\ -28.53 & 28.53 & 0 & 0 & -.493 & .493 & 0 \\ 0 & 0 & 14.56 & -14.56 & 0 & 0 & -1.12 \\ 0 & 0 & -14.56 & 14.56 & 0 & 0 & 1.12 \\ .493 & -.493 & 0 & 0 & .212 & -.212 & 0 \\ -.493 & .493 & 0 & 0 & -.212 & .212 & 0 \\ 0 & .0 & -1.12 & 1.12 & 0 & 0 & 84.14 \end{array}\right] & \begin{array}{c} KC_5 \\ KC_6 \\ KC_7 \\ KC_8 \\ KC_9 \\ KC_{10} \\ KC_{11} \end{array} \\ \begin{array}{ccccccc} KC_5 & KC_6 & KC_7 & KC_8 & KC_9 & KC_{10} & KC_{11} \end{array} & \end{array}$$

$$[KCc] = \begin{array}{cc} \left[\begin{array}{ccccccc} 1 & -1 & 0 & 0 & .20 & -.20 & 0 \\ -1 & 1 & 0 & 0 & -.20 & .20 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & -.0319 \\ 0 & 0 & -1 & 1 & 0 & 0 & .0319 \\ .20 & -.20 & 0 & 0 & 1 & -1 & 0 \\ -.20 & .20 & 0 & 0 & -1 & 1 & 0 \\ 0 & .0 & -.0319 & .0319 & 0 & 0 & 1 \end{array}\right] & \begin{array}{c} KC_5 \\ KC_6 \\ KC_7 \\ KC_8 \\ KC_9 \\ KC_{10} \\ KC_{11} \end{array} \\ \begin{array}{ccccccc} KC_5 & KC_6 & KC_7 & KC_8 & KC_9 & KC_{10} & KC_{11} \end{array} & \end{array}$$

Figure 4.11. The KC Covariance and KC Correlation Matrices

To find out which dimensions are causing the correlation, **[*Co*]** can be used. If any column of **[*Co*]** has more than one nonzero entry, then its means all KCs corresponding to the nonzero entries in that column are correlated with each other. For example, looking at Figure 4.10, it can be seen that the positive correlation between ***KC₅*** or ***KC₆*** (blade alignment in the ***x*** direction) and ***KC₉*** or ***KC₁₀*** (blade alignment in the ***θz*** direction) is due to ***D₃*** (alignment in the ***θz*** direction between the upper cutter base and the lower cutter) and ***D₁₂*** (alignment in the ***θz*** direction of the upper blade on the upper cutter body)

4.6 Summary

In this chapter five types of analysis were discussed. In may ways Motion/Constraint Analysis is the most important because if $[\boldsymbol{B}]_{i,j}$ is not invertible due to excess over-constraints and under-constraints, none of the other techniques are possible. Sensitivity, Performance, and Contribution Analyses are three methods that work together to help assess the manufacturability of the assembly design. Sensitivity Analysis determines how sensitive the KCs are to variability in the dimensions. Performance Analysis assesses the Cpks and reject rates for the assembly with respect to the KCs. Contribution Analysis determines what dimensions are contributing most to the variability in the KCs. Finally, Coupling Analysis allows a designer to understand the interrelationships between the variability of the KCs.

While some of the equations and procedures for Sensitivity, Performance, Contribution, and Coupling Analysis are complex, a major advantage regarding them is that all are completely amendable to computer implementation. Using Matlab® programming language an executable, called manufacturability.m, was created that automatically carries out these four analysis techniques as shown above for a Type-I assembly[6] with a single configuration (Motion/Constraint analysis can be carried out using constraintdetector.m that was described earlier). Manufacturability.m takes as inputs the vectors and matrices shown in Table 4.3 and provides as outputs the matrices shown in Table 4.4 (see Table 4.5 for counting variables). A copy of manufacturability.m is

[6] A program for analyzing Type-II assemblies was also created by the author but is not presented in this thesis.

included in Appendix A. It can be run on Release 13 of Matlab® and requires both the Statistics and Optimization Toolboxes to be installed. In addition to using standard Matlab® functions and some functions from the Optimization and Statistics Toolboxes, manufacturability.m calls two other specialized subroutines called mppfinder.m and modeler.m that are also included in Appendix A.

Vector or Matrix	Description	Size
$[Fn]_{1,1}$	Guesses for Nominal DOF Values	(ftt x 1)
$[Ful]_{1,1}$	Upper Limits on DOF values	(ftt x 1)
$[Fll]_{1,1}$	Lower Limits on DOF values	(ftt x 1)
$[Fd]_{1,1}$	Direcions for DOF	(ftt x 4)
$[Sn]_{1,1}$	Guesses for Nominal DOS Values	(stt x 1)
$[Sul]_{1,1}$	Upper Limits on DOS values	(stt x 1)
$[Sll]_{1,1}$	Lower Limits on DOS values	(stt x 1)
$[Sd]_{1,1}$	Direcions for DOS	(stt x 4)
$[SK]_{1,1}$	Nominal DOS stiffness matrix	(stt x stt)
$[Cn]_{1,1}$	Nominal DOC values	(ctt x 1)
$[Cd]_{1,1}$	Direcions for DOC	(ctt x 4)
$[Dn]_{1,1}$	Nominal dimension values	(dtt x 1)
$[Dd]_{1,1}$	Direcions for dimensions	(dtt x 1)
$[Dt]$	distribution types for dimensions	(dtt x 1) **(Character Strings)**
$[Dp]$	distribution parameters for dimensions	(dtt x 3)
$[KCd]_{1,1}$	KC directions	(kcp x 1)
$[KCl]$	performance KC offsets	(kcp x 1)
$[KCt]$	performance KC types	(kcp x 2)
$[KC]$	KC combination matrix	(kcp x $\{6m_{1,1}+stt+6g_{1,1}\}$)
$[CHd]_{1,1}$	Matrix of dimension chains	(dtt x $?$)
$[CHm]_{1,1}$	Matrix of DOF, DOS,and DOC chains	($6m_{1,1}$ x $?$)
$[CHc]_{1,1}$	Matrix of closed-loop chains	($\lambda_{1,1}$ x $?$)
$[CHo]_{1,1}$	Matrix of open-loop chains	($g_{1,1}$ x $?$)
$[CHk]_{1,1}$	Matrix of KC chains	(kcp x $?$)
KCT	Target Cpk	(1 x 1)

Table 4.3. Manufacturability.m Inputs for Type-I Single Configuration Assembly

Vector or Matrix	Description	Size
[Da]	Average Dimension Values	(dtt x 1)
[Dv]	Dimension Covariance Matrix	(dtt x dtt)
[KCS]	Performance KC Senstitivity Matrix	(kcp x dtt)
[Cpk]	Performance KC Cpks	(kcp x 1)
[Rej]	Performance KC Reject Rates	(kcp x 1)
[Co]	KC Contribution Matrix	(kcp x dtt)
[KCv]	KC Covariance Matrix	(kcp x kcp)
[KCc]	KC Correlation Matrix	(kcp x kcp)
[KCa]	Average Peformance KC Values	(kcp x 1)
[KCn]	Nominal Peformance KC Values	(kcp x 1)
[KCL]	Performance KC Specification Limits	(kcp x 1)
[KCL]	Performance KC Specification Limits	(1 x 1)

Table 4.4. Manufacturability.m Outputs for Type-I Single Configuration Assembly

Variable	Description	Value
a	Total # of assembly states	1
bt	Total # of configurations	1
$kctt$	Total # of KC's	kcp
kcl	Total # of Linking KC's	0
kcp	Total # of Performance KC's	$kct_{1,1}$
dtt	Total # of Dimensions	$dt_{1,1}$
ftt	Total # of DOF	$ft_{1,1}$
ctt	Total # of DOC	$ct_{1,1}$
stt	Total # of DOS	$st_{1,1}$
$p_{1,1}$	Total # of Parts	$p_{1,1}$
$m_{1,1}$	Total # of Real Mates	$m_{1,1}$
$\lambda_{1,3}$	Total # of Closed-Loops	$\lambda_{1,1}$
$g_{1,4}$	Total # of Open-Loops	$g_{1,1}$

Table 4.5. Counting Variables for Type-I Single Configuration Assembly

5. IMPROVING MANUFACTURABILITY

If the analysis techniques in Chapter 4 reveal that a design has an unacceptable level of manufacturability, there exist several options for remedying the situation. These remedies can be thought of as pertaining to one of the following three categories according to [Taguchi, 1993]:

- **System-Level Strategies**
- **Parameter-Level Strategies**
- **Tolerance-Level Strategies**

The discussion that follows concerning these three categories of strategies is not meant to be all inclusive but is instead intended to provide a basic framework that illustrates how the model developed in Chapter 3 and the results from Chapter 4 can be used to support manufacturability improvement efforts.

5.1 System-Level Strategies

Making changes at the system-level can have far reaching effects on assembly behavior and manufacturability. In fact, many system-level changes require that the KCs be redefined entirely. Generally speaking, the modeling and verification techniques presented in this work do not provide much insight into what system-level design changes one might make to improve manufacturability. Nonetheless, due to the importance of system-level issues, a brief discussion on this topic is still appropriate. Specifically, changes in the following system-level characteristics will be discussed here:

- **Part-Status**
- **Loop-Status**
- **Constraint-Status**
- **Type-Status**

Consider, for example, if the three-legged table shown in the left side of Figure 5.1 exhibited too much variability in the levelness of its top with respect to the floor. Firstly, one might consider changing its part-status, or in other words the number of parts of the way in which they are interconnected. One way to accomplish this is by reducing the number of parts in each leg from three to one as shown in the top-right of Figure 5.1. Secondly, one might consider changing the loop-status, or in other words, the number of open-loops and closed-loops and the parts and features they contain. This could be achieved by adopting the design shown in the center-right of Figure 5.1. Thirdly, one can change the constraint-status of an assembly, or in other words, the number and nature of over-constraints and under-constraints in the assembly. This might be accomplished by making the legs adjustable as in the bottom-right of Figure 5.1.

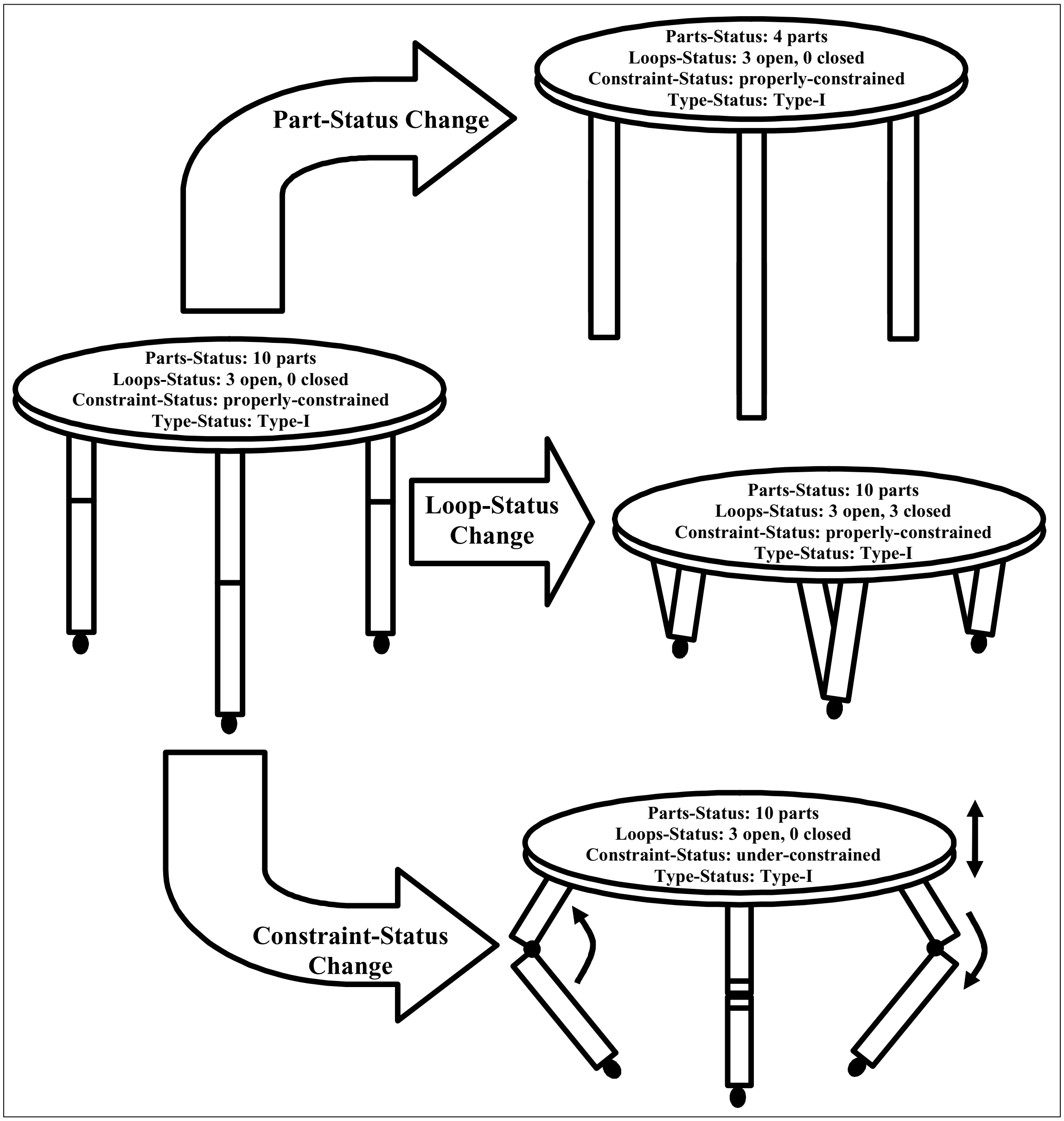

Parts-Status: 4 parts
Loops-Status: 3 open, 0 closed
Constraint-Status: properly-constrained
Type-Status: Type-I
Part-Status Change
Parts-Status: 10 parts
Loops-Status: 3 open, 0 closed
Constraint-Status: properly-constrained
Type-Status: Type-I
Loop-Status Change
Parts-Status: 10 parts
Loops-Status: 3 open, 3 closed
Constraint-Status: properly-constrained
Type-Status: Type-I
Constraint-Status Change
Parts-Status: 10 parts
Loops-Status: 3 open, 0 closed
Constraint-Status: under-constrained
Type-Status: Type-I

Figure 5.1. Examples of System-Level Changes to a Table

One final system-level change discussed here involves a change in the type-status [Whitney], or the way in which the parts are assembled. This could be accomplished in the case of the three-legged table by changing the way the legs are assembled as shown in Figure 5.2.

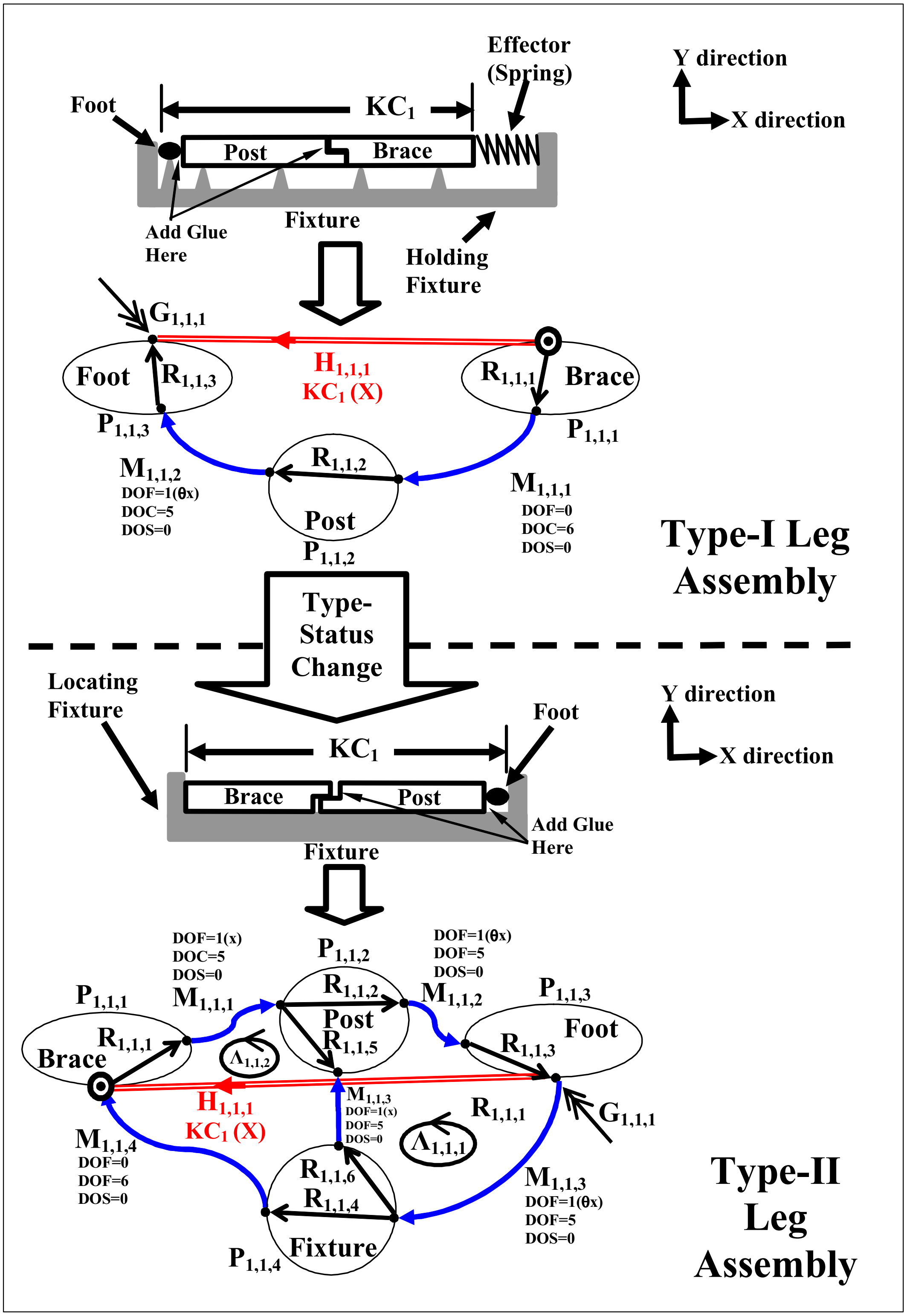


Figure 5.2. Example of a Type-Status Change for the Assembly of Table Legs

Using the Type-II assembly process shown in the bottom of Figure 5.2 instead of the Type-I assembly process shown in the top of Figure 5.2 could be beneficial because the length of the leg and variability in its length (in the ***x*** direction) is independent of the lengths of the brace, foot and post and is only dependent on the fixture. Thus if a single fixture could be fabricated with high accuracy to the desired length, all legs assembled using that fixture would also have high accuracy.

For Type-II assemblies, a possible system-level change is to find an assembly sequence, ***s***, and a set of nominal dimension values, **[*Dn*]**, that result in uncorrelated KCs. Then, measurement equipment together with adjustable fixtures can be used to set each KC sequentially so that their actual values equal or are very close to the nominal target values. This approach can be expressed loosely as an optimization problem as follows:

$$
\begin{aligned}
&\textit{Choose } [Da] \textit{ and } "s" \textit{ that minimzes } \beta \\
&\textit{where} \\
&\beta = \left(\sum_{i=1}^{4} \sum_{j=1}^{4} \left([KCc_{i,j}] \right)^2 \right) \\
&\textit{Subject to} \\
&[KCa_k] = [KCn_k] \, if \, [KCt_k] = 1 \\
&Abs\left([KCa_k] - [KCL_k] \right) \geq 0 \, if \, [KCt_k] = 2 \\
&[Da_l] = [Dn_l] \\
&[Dll_l] \leq [Dn_l] \leq [Dul_l] \\
&\textit{for} \\
&i, j, k = 1, 2, \ldots, kcp \quad (i \neq j) \\
&l = 1, 2, \ldots, dtt
\end{aligned}
$$

Equation 5.1

The minimum value of $\boldsymbol{\beta}$ in Equation 5.1, called $\boldsymbol{\beta}^*$, captures the amount of KC coupling and corresponds to the optimal assembly sequence, $\boldsymbol{s}^*$, and the optimal set of average dimensions values, $[\boldsymbol{Dn}^*]$, that minimize the KC coupling. Since this methodology presents no analytical form for $\boldsymbol{s}$, no mathematical method for solving Equation 5.1 is presented here. At best, one could use trial-and-error to try and arrive at a better assembly sequence with less KC coupling, checking $[\boldsymbol{KCc}]$ after each trial. Several more analytic and complete approaches to enumerating assembly sequences and/or identifying the best one(s) are presented by [Baldwin et al], [Klein], and [Milner, Graves, and Whitney].

Careful review of Equation 5.1 reveals that the solution is subject to four constraints. The first constraint insures that the nominal values of the alignment KCs remain constant. The second constraint insures that the nominal values the clearance KCs remain above their respective specification limits. Together, these two constraints insure that the nominal performance of the design does not change during optimization. The third constraint requires that the average dimension values equals the nominal dimension values. This constraint effectively eliminates any mean shift. The final constraint insures that the nominal values of dimensions do not exceed the upper and lower bounds as specified in $[\boldsymbol{Dul}]$ and $[\boldsymbol{Dll}]$ respectively (see Figure 3.14 for examples of these bounds in the case of the fingernail clipper assembly). This insures that the packing constraints on the design are satisfied. In an industry setting, these bounds might be set to reflect some sort of size envelope that the assembly has to fit in (e.g. an engine fitting under the hood of a car) or represent some limitations in the capabilities of the fabrication or assembly processes. All four constraints discussed here are applied to all the optimization techniques presented in this chapter.

In the case of the fingernail clipper, another possibility for a system-level change exists. This option involves waiting until the fingernail clipper is completely assembled before sharpening the blades simultaneously[7]. By doing this, the variability in the cutters as dictated by the fabrication and assembly processes does not affect the alignment of the blades since they are both sharpened at the same time after final assembly. Figure 5.3 illustrates how this is done. The schematic shows how the blades can be designed so that the grinding always leaves two sharp matched edges on the blades. In general, it is not always the best choice to conduct post assembly machining operations to insure manufacturability. In the case of the fingernail clipper it appears to be a very attractive option given the low achievable Cpks of the KCs. Most likely, real fingernail clippers are made this way because at high production volumes post-assembly grinding is probably much more economical than carefully controlling the manufacturing processes.

[7] Dan Whitney is credited for pointing out this additional option to the author.

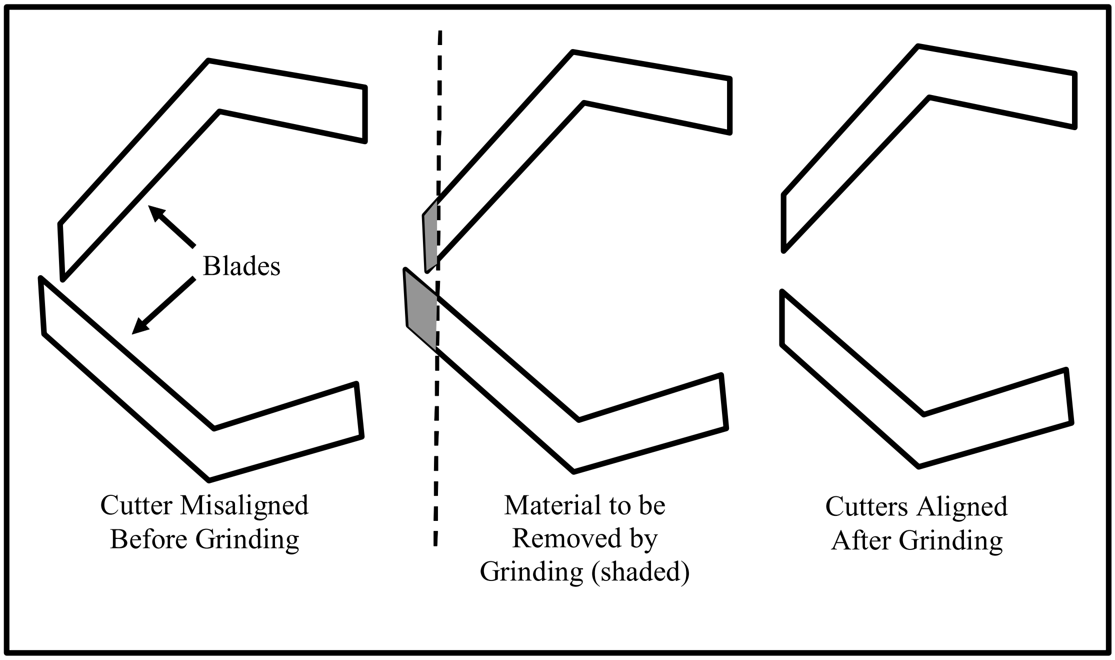

Blades
Cutter Misaligned Before Grinding
Material to be Removed by Grinding (shaded)
Cutters Aligned After Grinding

Figure 5.3. Post-Assembly Machining to Align Clipper Blades

In general, while many options exist for possible changes at the system-level, the model presented in this work does not support an analytical comparative search of such options. The only way to gain insight using the approach presented is to construct a detailed model of the original assembly design and each of the proposed new assembly designs that incorporate the potential system-level changes and compare their manufacturability using the method presented in Chapter 4. Admittedly, this is a very tedious and cumbersome approach as it requires extensive knowledge of the details of the design. In light of this, the author's suggestion is to apply this methodology only after engineering is at least somewhat confident that the optimal system-level design has been identified.

5.2 Parameter-Level Strategies

If making changes at the system-level is not an option or such changes do not improve manufacturability, another alternative is to make changes at the parameter-level. In this work, the only changeable parameters are the nominal values of the part dimensions contained in **[*Dn*]** (while in reality assemblies have other changeable design parameters relating to mass, stiffness, and other part properties, adjusting such parameters are not discussed in this work). Related to the dimensions, there are two types of changes that can be made:

- **Eliminating Mean Shifts**
- **Adjusting the Nominal Values**

Each of these will now be discussed briefly below.

5.2.1 Eliminating Mean Shift

Eliminating mean shift is often a simple way to improve the manufacturability of an assembly at the parameter level. This is accomplished by centering the manufacturing processes so that the averages or means of their distributions equal the nominal values as specified in the assembly design. Eliminating mean shift can be represented analytically as follows:

$$
\begin{array}{l}
\textit{Define } f_{D_i}(\delta_i) \\
\textit{such that} \\
[Da_i] = [Dn_i] \\
\textit{for} \\
i = 1,2,\mathrm{K}, dtt
\end{array}
$$

Equation 5.2

Figure 5.4 shows the effect of eliminating the mean shift in the dimensions, ***D*₃**, ***D*₇** and ***D*₁₂** on the Cpks and reject rates of the performance KCs for the fingernail clipper. While the Cpks of ***KC*₅**, ***KC*₉**, and ***KC*₁₁** went up, the Cpks of ***KC*₆**, and ***KC*₁₀** went down. This is because these KCs are from the same KC feature pairs and have the same directions as ***KC*₅** and ***KC*₉** except they represent lower specification limits instead of upper specification limits. Thus eliminating means shift will lower the Cpks of these KCs. Nonetheless, the Cpks of ***KC*₆** and ***KC*₁₀** are still above the target, ***KCT*** (which equals 1.33 in this case).

Before Eliminating Mean Shift

$$
[Rej]^T = \begin{bmatrix} .0073 & .0002 & .0092 & .0092 & .0190 & 2.9E-5 & 2.87 \\ KC_5 & KC_6 & KC_7 & KC_8 & KC_9 & KC_{10} & KC_{11} \\ (+x) & (-x) & (+y) & (-y) & (+\theta z) & (-\theta z) & (-z) \end{bmatrix}
$$

$$
[Cpk]^T = \begin{bmatrix} 1.27 & 1.58 & 1.25 & 1.25 & 1.18 & 1.67 & 0.633 \\ KC_5 & KC_6 & KC_7 & KC_8 & KC_9 & KC_{10} & KC_{11} \end{bmatrix}
$$

$$
[Rej]^T = \begin{bmatrix} .0010 & .0010 & .0092 & .0092 & .0009 & 0009 & 1.65 \\ KC_5 & KC_6 & KC_7 & KC_8 & KC_9 & KC_{10} & KC_{11} \\ (+x) & (-x) & (+y) & (-y) & (+\theta z) & (-\theta z) & (-z) \end{bmatrix}
$$

$$
[Cpk]^T = \begin{bmatrix} 1.42 & 1.42 & 1.25 & 1.25 & 1.43 & 1.43 & 0.710 \\ KC_5 & KC_6 & KC_7 & KC_8 & KC_9 & KC_{10} & KC_{11} \end{bmatrix}
$$

After Eliminating Mean Shift

Figure 5.4. The Effect of Eliminating Mean Shift on the Cpks and Reject Rates

5.2.2 Changing Nominal Dimension Values

If eliminating mean shift is not sufficient to improve the manufacturability of the design or is not possible, one can seek for a new set of nominal dimension values that will result in a design with improved manufacturability. In general, an improvement will be seen using this approach if there exists a set of nominal dimensions values (within the packaging constraints) that lowers the overall sensitivity of the performance KCs to the variability in the dimensions. In the specific case of clearance KCs, an improvement may also be seen if the nominal dimensions can be set such that the nominal values of the clearance KCs are increased, even if the sensitivity of those KCs to the variability is unchanged. Increasing the nominal values of clearance KCs will automatically increase their Cpks because the distance between the specification limit and the nominal KC value has increased [Terry]. In general choosing an optimal set of dimensions is commonly called parameter design [Taguchi, 1992] or parameter optimization and two such techniques for carrying this out will now be discussed.

[*Cpk*] contains the measures of variability that one can expect any of the performance KCs to exhibit. If one or more of the values in **[*Cpk*]** is lower than the target Cpk, ***KCT***, the performance of the entire assembly is in jeopardy. The lowest value of **[*Cpk*]** represents the KC that is need of the most improvement. The following optimization problem can be solved in an attempt to improve this minimum value:

$$
\begin{aligned}
&\textit{Find } [Dn] \textit{ that maximizes } \beta \\
&\textit{where} \\
&\beta = \textit{Maximize}\,(\,\textit{Min}\,(\,[Cpk]\,)\,) \\
&\textit{Subject to} \\
&[KCa_k] = [KCn_k] \textit{ if } [KCt_k] = 1 \\
&Abs(\,[KCa_k] - [KCL_k]\,) \geq 0 \textit{ if } [KCt_k] = 2 \\
&[Da_l] = [Dn_l] \\
&[Dll_l] \leq [Dn_l] \leq [Dul_l] \\
&\textit{for} \\
&i = 1,2,\mathrm{K}\,,kcp \\
&j = 1,2,\mathrm{K}\,,dtt
\end{aligned}
$$

Equation 5.3

The “**min**” function takes a vector as input and outputs the lowest value in the vector. If during the optimization search procedure, the KC with the minimum Cpk changes, the **min** function will pick the new lowest Cpk. This way, the optimization procedure will always be working to improve the lowest Cpk, even if that changes throughout the optimization search.

The value of $\boldsymbol{\beta}$ in Equation 5.3 is the minimum Cpk of the performance KCs. $\boldsymbol{\beta}^*$ is the maximum possible minimum Cpk of the performance KCs and this occurs when the nominal values have been optimized and set at **[*Dn**]**. Figure 5.5 shows $\boldsymbol{\beta}^*$ and **[*Dn**]** and the results of solving the optimization problem in Equation 5.3 for the fingernail clipper example. The main change is seen in $\boldsymbol{D}_7$.

$$[Dn]^T = \begin{array}{ccccccccccccccc} [0 & 0 & 0 & 1.36 & 1.36 & -40.34 & \boxed{2.81} & 0 & -1.85 & -40.34 & -2.81 & 0 & -35.18 & 8.41 & -2.96] \\ D_1 & D_2 & D_3 & D_4 & D_5 & D_6 & D_7 & D_8 & D_9 & D_{10} & D_{11} & D_{12} & D_{13} & D_{14} & D_{15} \end{array}$$

$$[Rej]^T = \begin{array}{ccccccc} [.0030 & .0032 & .0081 & .0081 & .0007 & 0008 & 1.66] \\ KC_5 & KC_6 & KC_7 & KC_8 & KC_9 & KC_{10} & KC_{11} \\ (+x) & (-x) & (+y) & (-y) & (+\theta z) & (-\theta z) & (-z) \end{array}$$

$$[Cpk]^T = \begin{array}{ccccccc} [1.34 & 1.33 & 1.26 & 1.26 & 1.44 & 1.44 & 0.710] \\ KC_5 & KC_6 & KC_7 & KC_8 & KC_9 & KC_{10} & KC_{11} \end{array}$$

Before Optimization

$\beta = 0.710$

$$[Dn^*]^T = \begin{array}{ccccccccccccccc} [0 & 0 & 0 & 1.36 & 1.36 & -40.34 & \boxed{2.5} & 0 & -1.85 & -40.34 & -2.81 & 0 & -35.18 & 8.41 & -2.96] \\ D_1 & D_2 & D_3 & D_4 & D_5 & D_6 & D_7 & D_8 & D_9 & D_{10} & D_{11} & D_{12} & D_{13} & D_{14} & D_{15} \end{array}$$

$$[Rej]^T = \begin{array}{ccccccc} [.0032 & .0030 & .0111 & .0059 & .0009 & 0008 & .0001] \\ KC_5 & KC_6 & KC_7 & KC_8 & KC_9 & KC_{10} & KC_{11} \\ (+x) & (-x) & (+y) & (-y) & (+\theta z) & (-\theta z) & (-z) \end{array}$$

$$[Cpk]^T = \begin{array}{ccccccc} [1.33 & 1.34 & 1.23 & 1.28 & 1.43 & 1.44 & 1.57] \\ KC_5 & KC_6 & KC_7 & KC_8 & KC_9 & KC_{10} & KC_{11} \end{array}$$

After Optimization

$\beta^* = 1.23$

Figure 5.5. The Effect Optimizing Dimensions on the Cpks and Reject Rates

A second option is to maximize a weighted sum of the Cpks of all the performance KCs (this is similar to the approached presented by [Parkinson] except here it is not required that one posses an explicit representation of the objective function). This approach is desirable when there are several KCs that share importance. By using a weighted sum instead of just a single Cpk, it allows one to prioritize the KCs by placing the appropriate weight on each. These weights ideally sum to one and can be contained in a **(*kcp* x 1)** vector, **[*Wa*]** called the dimension weighting matrix. The weighted Cpk optimization problem takes the following form:

$$
\begin{aligned}
&\textit{Find } [Dn] \textit{ that maximizes } \beta \\
&\textit{where} \\
&\beta = \textit{Maximize}\left([Wa]^T [Cpk]\right) \\
&\textit{Subject to} \\
&[KCa_k] = [KCn_k] \textit{ if } [KCt_k] = 1 \\
&Abs\left([KCa_k] - [KCL_k]\right) \geq 0 \textit{ if } [KCt_k] = 2 \\
&[Da_l] = [Dn_l] \\
&[Dll_l] \leq [Dn_l] \leq [Dul_l] \\
&\textit{for} \\
&i = 1,2,\ldots,kcp \\
&j = 1,2,\ldots,dtt
\end{aligned}
$$

Equation 5.4

The value of $\boldsymbol{\beta}$ in Equation 5.4 does not have any physical significance unless the weights in **[*Wa*]** sum to one in which case it is a weighted average of the performance KC Cpks. $\boldsymbol{\beta}^*$ for this optimization is the maximum weighted Cpk and corresponds to a set of optimal nominal dimensions, **[*Dn**]**. Figure 5.6 shows **[*Wa*]**, **[*Dn**],** and the results of solving the optimization problem in Equation 5.4 for the fingernail clipper example (Each different **[*Wa*]** will result in a different set of optimal dimensions **[*Dn**]**). Due to the bounds on the dimensions (see Figure 3.14 for the bound vectors) and the fact that this optimization started where the previous parameter optimization terminated (see Figure 5.5), no improvement is seen using this approach. As such, the Cpks of $\boldsymbol{KC_7}$ and $\boldsymbol{KC_8}$ are still below the target and the only remaining option is to make changes at the tolerance-level. Such changes are the topic of the next section.

$$[Dn]^T = \begin{array}{ccccccccccccccc} [0 & 0 & 0 & 1.36 & 1.36 & -40.34 & 2.5 & 0 & -1.85 & -40.34 & -2.81 & 0 & -35.18 & 8.41 & -2.96] \\ D_1 & D_2 & D_3 & D_4 & D_5 & D_6 & D_7 & D_8 & D_9 & D_{10} & D_{11} & D_{12} & D_{13} & D_{14} & D_{15} \end{array}$$

$$[Rej]^T = \begin{array}{ccccccc} [.0010 & .0010 & .0092 & .0092 & .0009 & 0009 & \approx 0] \\ KC_5 & KC_6 & KC_7 & KC_8 & KC_9 & KC_{10} & KC_{11} \\ (+x) & (-x) & (+y) & (-y) & (+\theta z) & (-\theta z) & (-z) \end{array}$$

Before Optimization

$\beta = 1.42$

$$[Cpk]^T = \begin{array}{ccccccc} [1.42 & 1.42 & 1.25 & 1.25 & 1.43 & 1.43 & 1.92] \\ KC_5 & KC_6 & KC_7 & KC_8 & KC_9 & KC_{10} & KC_{11} \end{array} \qquad [Wa]^T = \begin{array}{ccccccc} \left[\frac{1}{7}\right. & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \left.\frac{1}{7}\right] \\ KC_5 & KC_6 & KC_7 & KC_8 & KC_9 & KC_{10} & KC_{11} \end{array}$$

$$[Rej]^T = \begin{array}{ccccccc} [.0010 & .0010 & .0092 & .0092 & .0009 & 0009 & \approx 0\ ] \\ KC_5 & KC_6 & KC_7 & KC_8 & KC_9 & KC_{10} & KC_{11} \\ (+x) & (-x) & (+y) & (-y) & (+\theta z) & (-\theta z) & (-z) \end{array}$$

After Optimization

$\beta^* = 1.42$

$$[Cpk]^T = \begin{array}{ccccccc} [1.42 & 1.42 & 1.25 & 1.25 & 1.43 & 1.43 & 1.92\ ] \\ KC_5 & KC_6 & KC_7 & KC_8 & KC_9 & KC_{10} & KC_{11} \end{array}$$

$$[Dn^*]^T = \begin{array}{ccccccccccccccc} [0 & 0 & 0 & 1.36 & 1.36 & -40.34 & 2.5001 & 0 & -1.85 & -40.34 & -2.81 & 0 & -35.18 & 8.41 & -2.9604] \\ D_1 & D_2 & D_3 & D_4 & D_5 & D_6 & D_7 & D_8 & D_9 & D_{10} & D_{11} & D_{12} & D_{13} & D_{14} & D_{15} \end{array}$$

Figure 5.6. The Effect of Weighted Optimization on the Cpks and Reject Rates

Using Matlab® programming language an executable, called dimoptimizer.m, was created that automatically carries out the weighted parameter optimization technique as shown above for a Type-I assembly[8] with a single configuration. The executable dimoptimizer.m takes as inputs the vectors and matrices shown in Table 5.1 and provides as outputs the matrices shown in Table 5.2 (see Table 5.3 for counting variables). A copy of dimoptimizer.m is included in Appendix A. In addition to requiring both the Statistics and Optimization Toolboxes to be installed, it also relies on two specialized functions called dimobjective.m and dimconstraint.m that are also included in Appendix A.

[8] A program for analyzing Type-II assemblies was also created by the author but is not presented in this thesis.

Vector or Matrix	Description	Size
$[Fn]_{1,1}$	Guesses for Nominal DOF Values	(*ftt* x *1*)
$[Ful]_{1,1}$	Upper Limits on DOF values	(*ftt* x *1*)
$[Fll]_{1,1}$	Lower Limits on DOF values	(*ftt* x *1*)
$[Fd]_{1,1}$	Direcions for DOF	(*ftt* x *4*)
$[Sn]_{1,1}$	Guesses for Nominal DOS Values	(*stt* x *1*)
$[Sul]_{1,1}$	Upper Limits on DOS values	(*stt* x *1*)
$[Sll]_{1,1}$	Lower Limits on DOS values	(*stt* x *1*)
$[Sd]_{1,1}$	Direcions for DOS	(*stt* x *4*)
$[SK]_{1,1}$	Nominal DOS stiffness matrix	(*stt* x *stt*)
$[Cn]_{1,1}$	Nominal DOC values	(*ctt* x *1*)
$[Cd]_{1,1}$	Direcions for DOC	(*ctt* x *4*)
$[Dn]_{1,1}$	Nominal dimension values	(*dtt* x *1*)
[*Dul*]	Nominal dimension values	(*dtt* x *1*)
[*Dll*]	Nominal dimension values	(*dtt* x *1*)
$[Dd]_{1,1}$	Direcions for dimensions	(*dtt* x *1*)
[*Dt*]	distribution types for dimensions	(*dtt* x *1*) **(Character Strings)**
[*Dp*]	distribution parameters for dimensions	(*dtt* x *3*)
$[KCd]_{1,1}$	KC directions	(*kcp* x *1*)
[*KCl*]	performance KC offsets	(*kcp* x *1*)
[*KCt*]	performance KC types	(*kcp* x *2*)
[*KC*]	KC combination matrix	$(kcp \text{ x } \{6m_{1,1}+stt+6g_{1,1}\})$
$[CHd]_{1,1}$	Matrix of dimension chains	(*dtt* x *?*)
$[CHm]_{1,1}$	Matrix of DOF, DOS,and DOC chains	$(6m_{1,1} \text{ x } ?)$
$[CHc]_{1,1}$	Matrix of closed-loop chains	$(\lambda_{1,1} \text{ x } ?)$
$[CHo]_{1,1}$	Matrix of open-loop chains	$(g_{1,1} \text{ x } ?)$
$[CHk]_{1,1}$	Matrix of KC chains	(*kcp* x *?*)
[*Wa*]	KC Cpk Weights	(*kcp* x *1*)
KCT	Target Cpk	(*1* x *1*)

Table 5.1. Dimoptimizer.m Inputs for Type-I Single Configuration Assembly

Vector or Value	Description	Size
$[Dn^*]$	Optimum Dimension Values	(*dtt* x *1*)
β^*	Maximum Weighted Cpk	(*1* x *1*)

Table 5.2. Dimoptimizer.m Outputs for Type-I Single Configuration Assembly

Variable	Description	Value
a	Total # of assembly states	1
bt	Total # of configurations	1
$kctt$	Total # of KC's	kcp
kcl	Total # of Linking KC's	0
kcp	Total # of Performance KC's	$kct_{1,1}$
dtt	Total # of Dimensions	$dt_{1,1}$
ftt	Total # of DOF	$ft_{1,1}$
ctt	Total # of DOC	$ct_{1,1}$
stt	Total # of DOS	$st_{1,1}$
$p_{1,1}$	Total # of Parts	$p_{1,1}$
$m_{1,1}$	Total # of Real Mates	$m_{1,1}$
$\lambda_{1,3}$	Total # of Closed-Loops	$\lambda_{1,1}$
$g_{1,4}$	Total # of Open-Loops	$g_{1,1}$

Table 5.3. Counting Variables for Type-I Single Configuration Assembly

5.3 Tolerance-Level Strategies

If a design has low manufacturability, and efforts at the system and parameter level have been exhausted, a final option is to increase the capability of the fabrication and assembly processes that cause there to be variability in the dimensions. In the literature this is often called tolerance allocation or tolerance design [Hong and Chang]. This is often the least desirable and most expensive option because it requires actual equipment and/or process changes instead of just design changes.

One method of approaching such problems is to look at **[*Co*]**, find the dimensions whose variability most contributes to the variability of the KCs with the lowest Cpks, weigh this information against the cost of improving the capability of the processes that delivers each of those dimensions, and allocate resources accordingly. This can be expressed loosely as the following optimization problem:

$$
\begin{array}{l}
Use\ [Co]\, to\ define\ f_{D_i}(\delta_i)\, that\ maximizes\ \beta \\
and\ insures\ that\ [Cpk_i] \geq KCT \\
where \\
\ \beta = Minimize\ (cost\ of\ changing\ [Dv]) \\
\ Subject\ to \\
[KCa_k] = [KCn_k]\, if\ [KCt_k] = 1 \\
Abs([KCa_k] - [KCL_k]) \geq 0\ if\ [KCt_k] = 2 \\
[Da_l] = [Dn_l] \\
[Dll_l] \leq [Dn_l] \leq [Dul_l] \\
for \\
i = 1,2,\ldots,kcp \\
j = 1,2,\ldots,dtt
\end{array}
$$

Equation 5.5

In practice it is difficult to obtain accurate cost-performance functions for the various fabrication processes and no such functions are presented in analytical form here. However, several approaches to the general problem in Equation 5.5 have been presented in the literature by [Bjørke], [Chase, Greenwood, Loosli, and Hauglund], [Ngoi and Min], and others.

Even if an analytical approach doesn't exist for solving Equation 5.5, a trail-and-error approach can be used that captures the essence of the optimization problem. In the case of the fingernail clipper, this is accomplished by using **[*Co*]** as a guide for reducing the variability in one or more dimensions so that the Cpks of $\boldsymbol{KC_7}$ and $\boldsymbol{KC_8}$ are improved. Figure 5.7 shows how this is done. Since the variability in $\boldsymbol{D_{10}}$ contributes about 64% of the variability in $\boldsymbol{KC_7}$ and $\boldsymbol{KC_8}$, it seems like the most logical place to start. As it turns out, reducing the standard deviation of ***D₁₀*** from 0.028 to 0.024 results in a sufficient improvement in the Cpks of ***KC₇*** and ***KC₈***. With this final change, all the Cpks for the fingernail clipper KCs are at acceptable levels. This means that the manufacturability of the design is such that all KCs are delivered and the design is considered a success.

$$[Co]=\begin{array}{cc}
\left[\begin{array}{cccccccc}
99.69 & 0 & L & 0 & L & 0 & L & 0 \\
99.69 & 0 & & 0 & & 0 & & 0 \\
0 & 1.045 & & 32.98 & & 64.25 & & .0108 \\
0 & 1.045 & L & 32.98 & L & 64.26 & L & .0083 \\
0 & 0 & & 0 & & 0 & & 0 \\
0 & 0 & & 0 & & 0 & & 0 \\
0 & .0815 & & 0 & & 0 & & 49.75
\end{array}\right] &
\begin{array}{c} KC_5 \\ KC_6 \\ KC_7 \\ KC_8 \\ KC_9 \\ KC_{10} \\ KC_{11} \end{array} \\
\begin{array}{cccccccc} D_1 & D_2 & L & D_6 & L & D_{10} & L & D_{15} \end{array} &
\end{array}$$

Before adjusting capability of process that sets *D₁₀* so that σ₁₀ is reduced from 0.032 to 0.028

$$[Rej]^T=\begin{array}{ccccccc}
[.0010 & .0010 & .0092 & .0092 & .0009 & 0009 & \approx 0] \\
KC_5 & KC_6 & KC_7 & KC_8 & KC_9 & KC_{10} & KC_{11} \\
(+x) & (-x) & (+y) & (-y) & (+\theta z) & (-\theta z) & (-z)
\end{array}$$

$$[Cpk]^T=\begin{array}{ccccccc}
[1.42 & 1.42 & 1.25 & 1.25 & 1.43 & 1.43 & 1.92] \\
KC_5 & KC_6 & KC_7 & KC_8 & KC_9 & KC_{10} & KC_{11}
\end{array}$$

$$[Rej]^T=\begin{array}{ccccccc}
[.0010 & .0010 & .0029 & .0029 & .0009 & 0009 & \approx 0] \\
KC_5 & KC_6 & KC_7 & KC_8 & KC_9 & KC_{10} & KC_{11} \\
(+x) & (-x) & (+y) & (-y) & (+\theta z) & (-\theta z) & (-z)
\end{array}$$

$$[Cpk]^T=\begin{array}{ccccccc}
[1.42 & 1.42 & 1.34 & 1.34 & 1.43 & 1.43 & 1.92] \\
KC_5 & KC_6 & KC_7 & KC_8 & KC_9 & KC_{10} & KC_{11}
\end{array}$$

After improving capability of process that sets *D₁₀*

Figure 5.7. The Effect of Reducing Dimensional Variability on Manufacturability

5.4 Summary

In this chapter, several techniques for improving the manufacturability of assemblies by making changes at the system, parameter, and tolerance levels have been discussed. Together, Chapter 3, Chapter 4, and Chapter 5 represent all the theory for the methodology presented in this work. The fingernail clipper, while not part of the theory, served as a running example to illustrate how to carry out each phase of the methodology. In the next chapter, the methodology is applied to a valvetrain of a gasoline combustion engine and the results of this case study are presented.

6. TYPE-I CASE STUDY: RFF VALVETRAIN

With the aid of Matlab® executables modeler.m, constraintdetector.m, and manufacturability.m, the methodology presented in Chapter 3 through Chapter 5 was applied to a valvetrain of a Ford® V-8 2-Valve gasoline combustion engine. This chapter presents the results of this effort and is divided into background, modeling, verification, improvement, and summary sections.

6.1 Background

A Roller Finger Follower, or RFF, valvetrain (see Figure 6.1) is a set of parts on an engine that works together in a coordinated fashion to open and close an intake or exhaust valve of a combustion engine that serve respectively to intake the air-fuel mixture and expel the combusted gases. Assembly of an RFF valvetrain does not require any locating fixtures and as such it is considered a Type-I assembly. Further, none of the mates are compliant and all the parts can be assumed to be rigid[9]. The primary entities in a RFF valvetrain are the cylinder head, cam rod, cam lobe, follower, lash adjuster, valve seat, valve, valve guide, and sprocket (not shown in Figure 6.1). Together the cam rod and the eight cam lobes are called the camshaft. The sprocket is mounted to the end of the camshaft and has a key that mates with a keyway in the camshaft as shown in Figure 6.2. The sprocket is attached to the crankshaft of the engine via a chain and this insures that movement of the pistons and the valves are coordinated.

[9] The only compliant entity is a spring whose only purpose it to enforce force closure of the mates in the RFF valvetrain and as such does it not qualify as a part as that term is defined here.

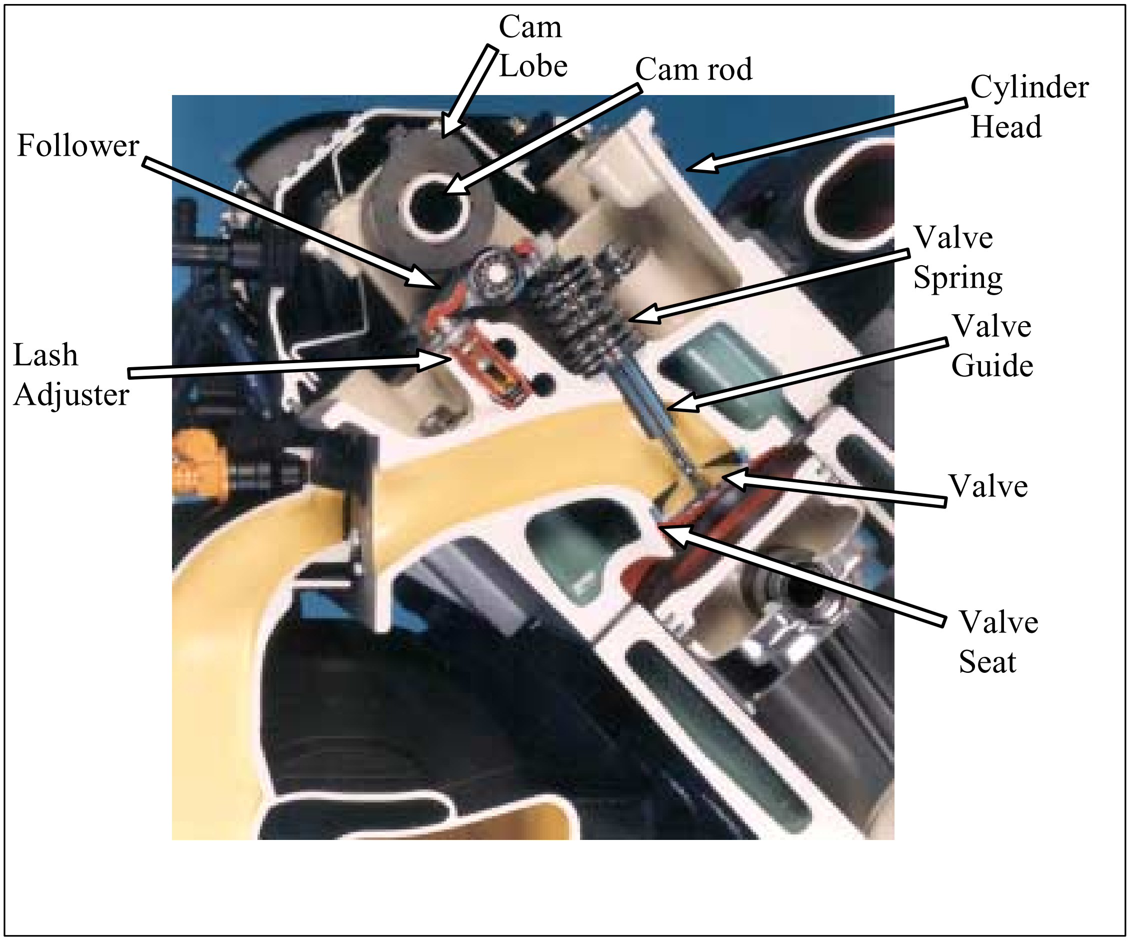


Figure 6.1. A Roller Finger Follower Valvetrain

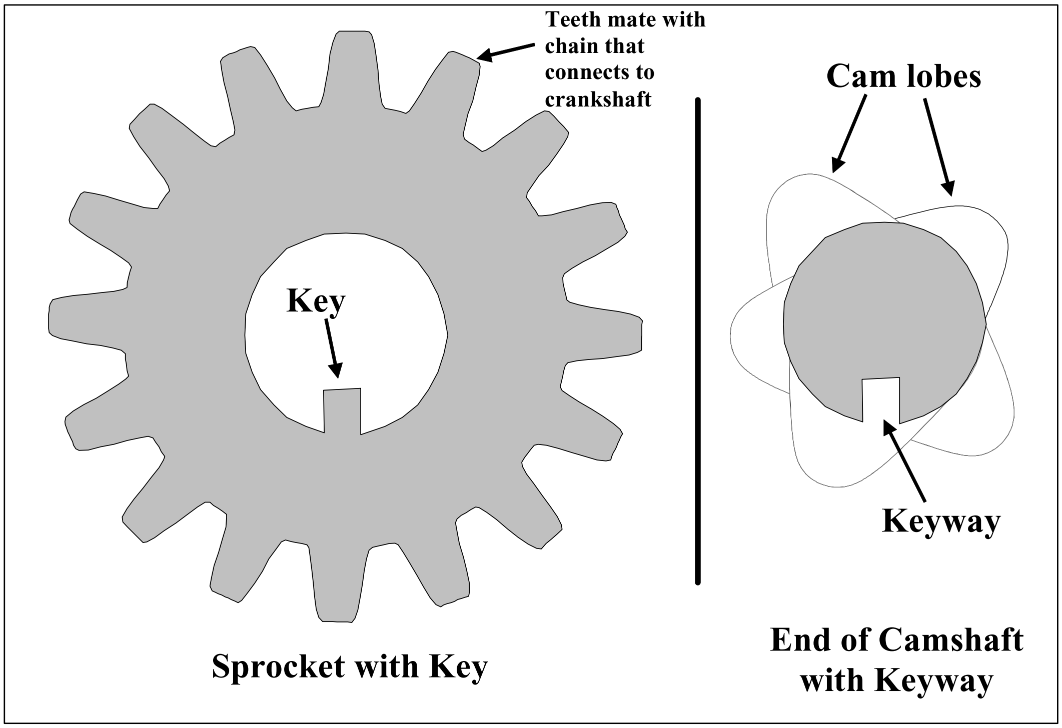


Figure 6.2. Sprocket with Key and Camshaft with Keyway

A V-8 2-valve engine has sixteen valvetrains like the one shown in Figure 6.1; four exhaust and four intake valvetrains for the four cylinders in the left bank, and four exhaust and four intake valvetrains for the four cylinders in the right bank (however each engine will only have two camshafts and two sprockets). The right and left sets of eight valvetrains are orthogonal to each other. The timing of the spark plugs, pistons, and fuel injectors are coordinated with the timing of the valvetrains using a set of chain driven sprockets (see Figure 6.2) such that the combustion events occur in pairs, with each pair including one cylinder (with its spark plug, fuel injector, and intake and exhaust valves) from the right bank and one cylinder from the left bank.

6.2 Modeling

While there are many customer expectations that depend on the proper function of the valvetrain, this case study only focuses on one: the level of noticeable vibration it causes. Due to the nature of V-engines, if the opening and closing of the valves for a cylinder on the right are not synchronized with the opening and closing of the valves for the matching cylinder on the left, the combustion events will not be balanced and the engine will vibrate. This vibration causes the frame and subsequently the seats of the passengers to vibrate as well and is most pronounced when the engine is idling or turning at a low speed. Roughly speaking, the amount of vibration is proportional to the level of asynchrony between sprockets of the right and left matching intake valvetrains. One way to measure this mismatch is to compare the absolute rotation angle of the left camshaft sprocket to the absolute rotation angle of the right camshaft sprocket when the intake valve is at its most open position, called “max-lift”. This measurement is sometimes called the bank-to-bank camshaft timing error. Analysis has shown that if the absolute value of the bank-to-bank camshaft timing error is greater than two degrees, the vibration is noticeable by the driver of the vehicle, a condition that is not acceptable[10].

Two KCs are needed to represent this vibration-related customer expectation since a positive or negative timing error creates unwanted vibration. Careful analysis revealed that the bank-to-bank camshaft timing error was equal to four times the timing error of a

[10] Joseph Stout and other employees from Ford Motor Company conducted a detailed analysis of valvetrain dynamics to arrive at this conclusion.

single camshaft sprocket relative to its corresponding cylinder head[11] (a V-engine has two cylinder heads). This 4:1 amplification is a result of the mirror image nature of the assembly operations used to assemble the right and left banks of valvetrains and the 2:1 gearing ratio of the sprockets to the crankshaft. Taking advantage of this and the fact that the assembly does not require any locating fixtures, the assembly model need only consist of a single configuration that includes a single valvetrain for an intake value that is set at max-lift. The bank-to-bank camshaft timing KCs can be replaced with two KCs that measure the angle of rotation of the sprocket mounted on the camshaft relative to the corresponding cylinder head. Figure 6.3 shows a schematic of a typical RFF valvetrain (spring omitted). The specification limit offsets for these new performance KCs are +.00873 and -.00873 radians (+.5 and -.5 degrees) and the target Cpk is 1.33. A simplified annotated liaison diagram showing only the real mates, intra-part relationships (unlabeled), parts, features (unlabeled) and the KC feature pair is shown in Figure 6.4.

[11] Chris Hocking and other employees from Ford Motor Company are credited for discovering this.

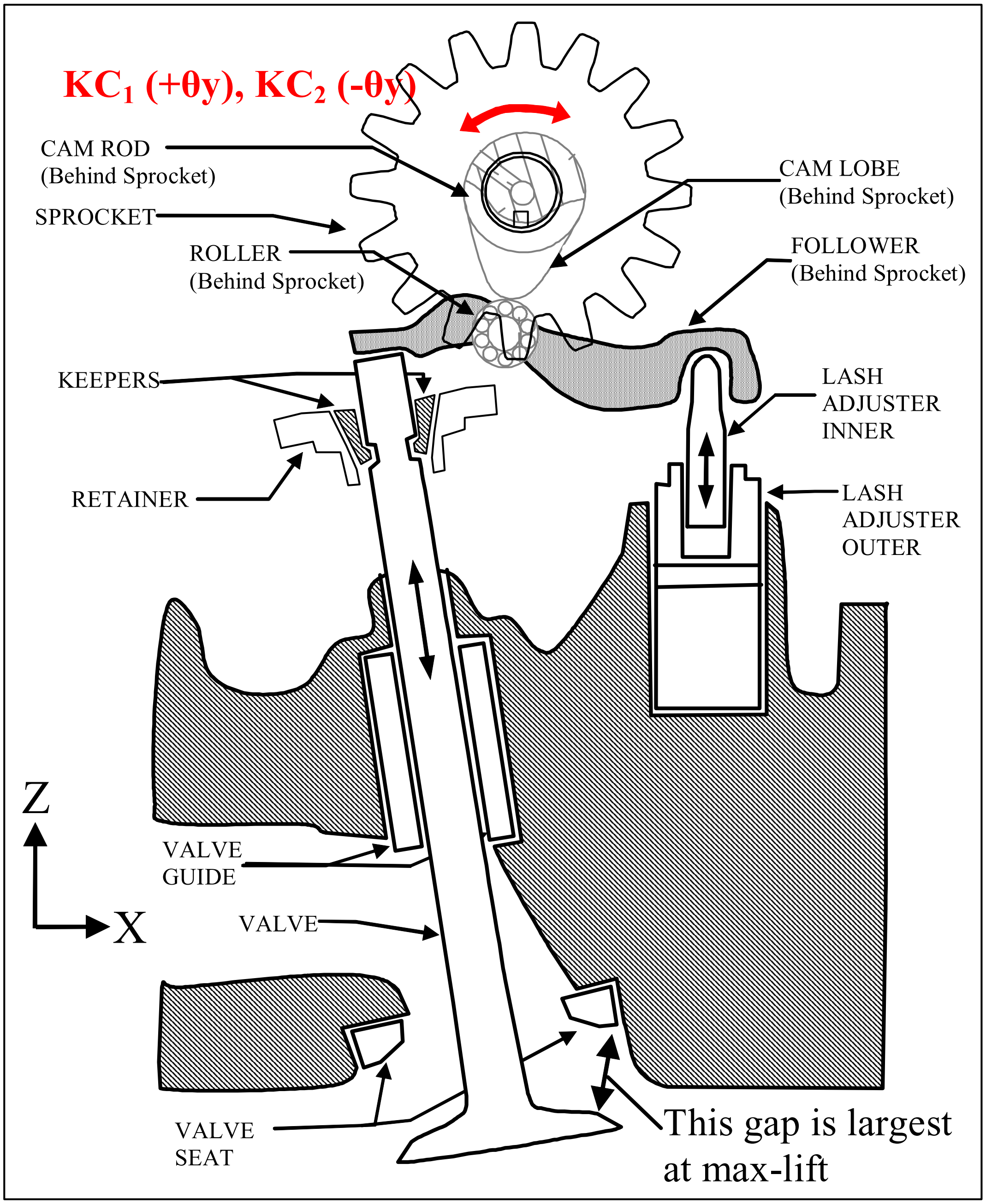


Figure 6.3. Vibration KCs for Valvetrain Shown at Max-Lift

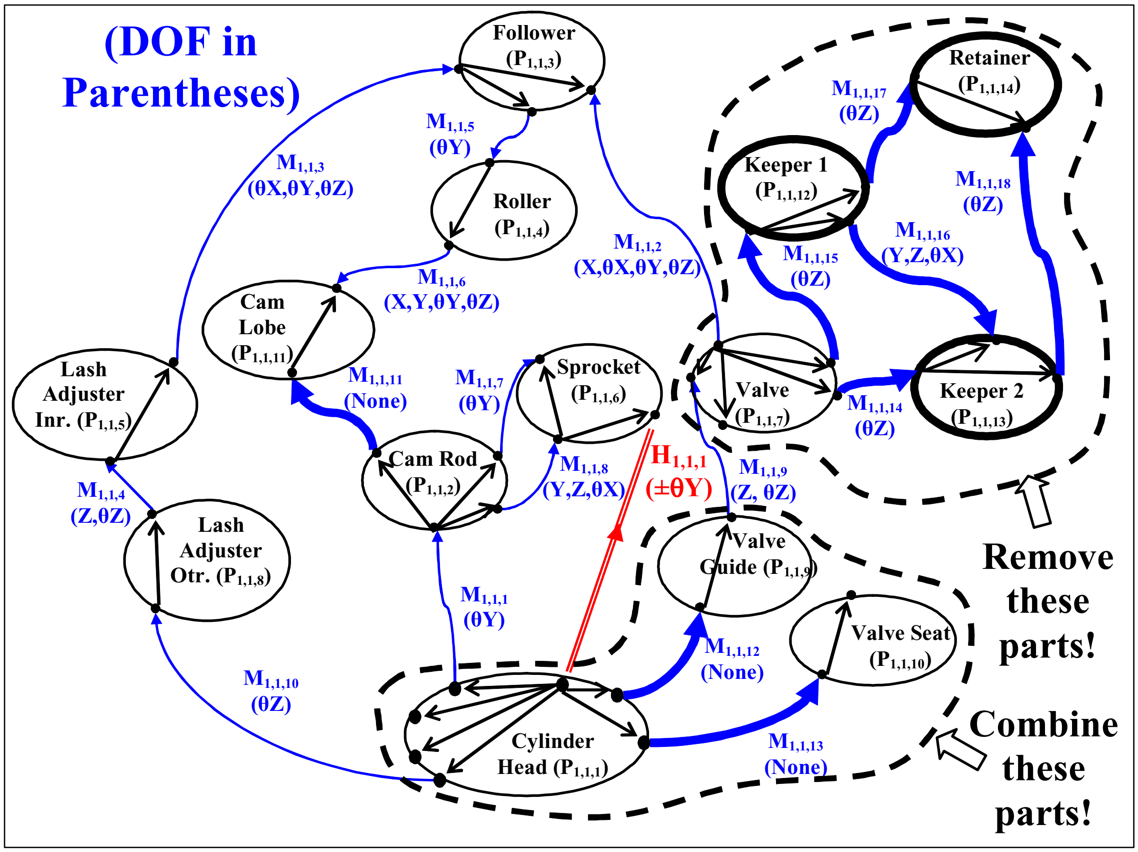


Figure 6.4. Annotated Liaison Diagram for RFF Valvetrain at Max Lift

Study of the annotated liaison diagram in Figure 6.4 reveals several things. First, the cam rod and the cam lobe can be considered one part, hereafter called the camshaft, since the mate between them is rigid and has no DOF. Second, the valve guide, valve seat, and cylinder head can be considered one part for the same reason, hereafter called the cylinder head (see bolded arrows in Figure 6.4 that are encircled with a dotted line). Third, the KCs of interest in this situation are unaffected by the keeper1, keeper2, and the retainer since those parts form closed-loops that do not share any dimensions or intra-part relationships with the closed-loops that contain the KC feature pair (see bolded arrows and parts that are encircled with a dotted line).

Based on the argument above, the annotated liaison diagram can be greatly simplified as shown in Figure 6.5. This simplified but analytically equivalent valvetrain contains 8 parts, 22 features, 14 intra-part relationships (containing 25 dimensions), 10 real mates (containing 23 DOF, 37 DOC, and 0 DOS), 1 open-loop, 3 closed-loops, and 1 KC feature pair (containing 2 KCs). Table 6.1 contains a description of the features for the configuration shown in Figure 6.5. Using Figure 6.5, dimensioned drawings of the parts obtained from Ford® Motor Company, and the Matlab® executable modeler.m discussed at the end of Chapter 3, an analytical valvetrain model was created[12].

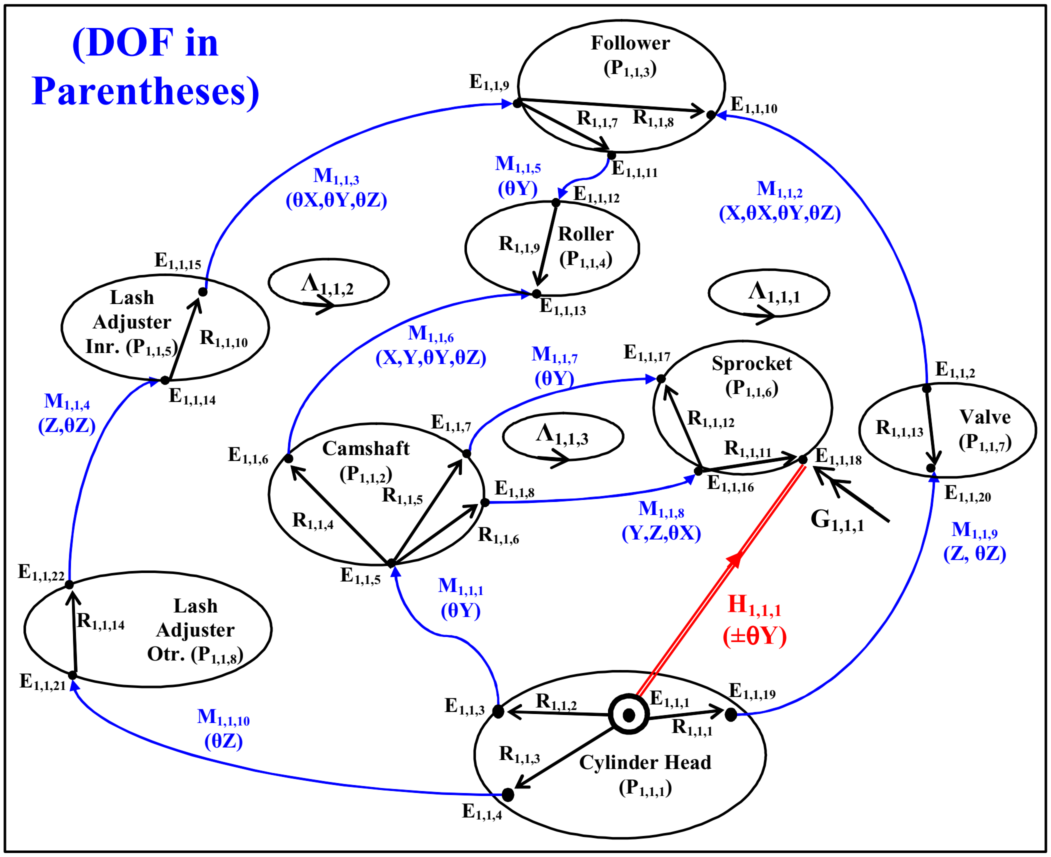


Figure 6.5. Revised Annotated Liaison Diagram for RFF Valvetrain at Max Lift

[12] For proprietary reasons, the actual values of the dimensions have been omitted from this case study.

Feature	Description
$\mathbf{E_{1,1,1}}$	Global origin on cylinder head (base feature and 1st feature in KC feature pair)
$\mathbf{E_{1,1,2}}$	valve tip (mates with follower tip)
$\mathbf{E_{1,1,3}}$	center axis of bore for camshaft (mates with camshaft)
$\mathbf{E_{1,1,4}}$	center axis of bore for lash adjuster (mates with lash adjuster outer)
$\mathbf{E_{1,1,5}}$	center axis of camshaft (mates with cylinder head)
$\mathbf{E_{1,1,6}}$	tip of camlobe (mates with roller)
$\mathbf{E_{1,1,7}}$	center axis of sprocket mounting plate (mates with sprocket back surface)
$\mathbf{E_{1,1,8}}$	center line of keyway (mates with sprocket key)
$\mathbf{E_{1,1,9}}$	center point of follwer pivot (mates with lash adjuster)
$\mathbf{E_{1,1,10}}$	follower tip (mates with valve)
$\mathbf{E_{1,1,11}}$	center axis of roller bearing (mates with roller)
$\mathbf{E_{1,1,12}}$	center axis of roller (mates with follower)
$\mathbf{E_{1,1,13}}$	surface of roller (mates with came lobe tip)
$\mathbf{E_{1,1,14}}$	center axis of lash adjuster inner (mates with lash adjuster outer)
$\mathbf{E_{1,1,15}}$	tip of lash adjuster inner (mates with follower pivot)
$\mathbf{E_{1,1,16}}$	sprocket key (mates with keyway in camshaft)
$\mathbf{E_{1,1,17}}$	sprocket back plate (mates with mounting plate on camshaft)
$\mathbf{E_{1,1,18}}$	center of timing tooth (2nd feature in KC feature pair)
$\mathbf{E_{1,1,19}}$	center axis of valve guide on cylinder head (mates with valve)
$\mathbf{E_{1,1,20}}$	valve center axis (mates with valve guide on cylinder head)
$\mathbf{E_{1,1,21}}$	axis of lash adjuster outer housing (mates with lash adjuster bore in cylinder head
$\mathbf{E_{1,1,22}}$	axis of lash adjuster outer (mates with lash adjuster inner)

Table 6.1. Description of Features for Valvetrain

6.3 Verification

6.3.1 Motion/Constraint Analysis

With the modeling complete, constraintdetector.m (discussed in Chapter 4) was used to determine whether any excess under-constraints or over-constraints exist. A summary of the results of this analysis are shown in Figure 6.6. It can be seen that the assembly has two over-constraints and six under-constraints. Using **[*TLr*]*1,1*** and **[*WPr*]*1,1*** (not shown here), the excess under-constraints and over-constraints were identified and they are also shown in Figure 6.6.

Constraint Analysis for Valvetrain

$ft_{1,1} = 23, ct_{1,1} = 37, st_{1,1} = 0,$

$\lambda_{1,1} = 3, g_{1,1} = 1, p_{1,1} = 8,$

$r1_{1,1} = 22, r2_{1,1} = 36$

$fe_{1,1} = 6(p_{1,1} - 1) - r2_{1,1} = 6(8-1) - 36 = 6$

or

$fe_{1,1} = ft_{1,1} + 6g_{1,1} - r1_{1,1} = 22 + 6\times 1 - 22 = 6$

and

$ce_{1,1} = 6(\lambda_{1,1} + g_{1,1}) - r1_{1,1} = 6(3+1) - 22 = 2$

$$fe_{1,1} + ce_{1,1} = 6 + 2 \neq 0$$

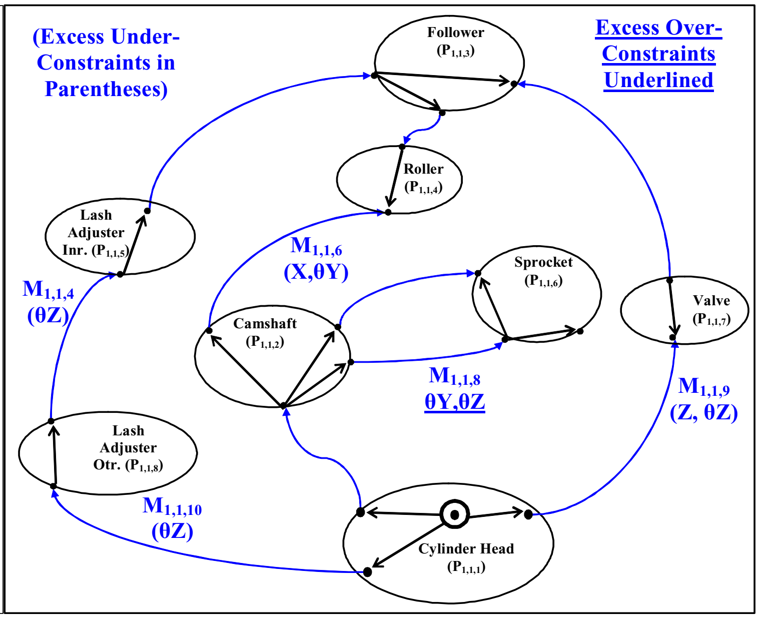


Figure 6.6. Motion/Constraint Analysis of Valvetrain

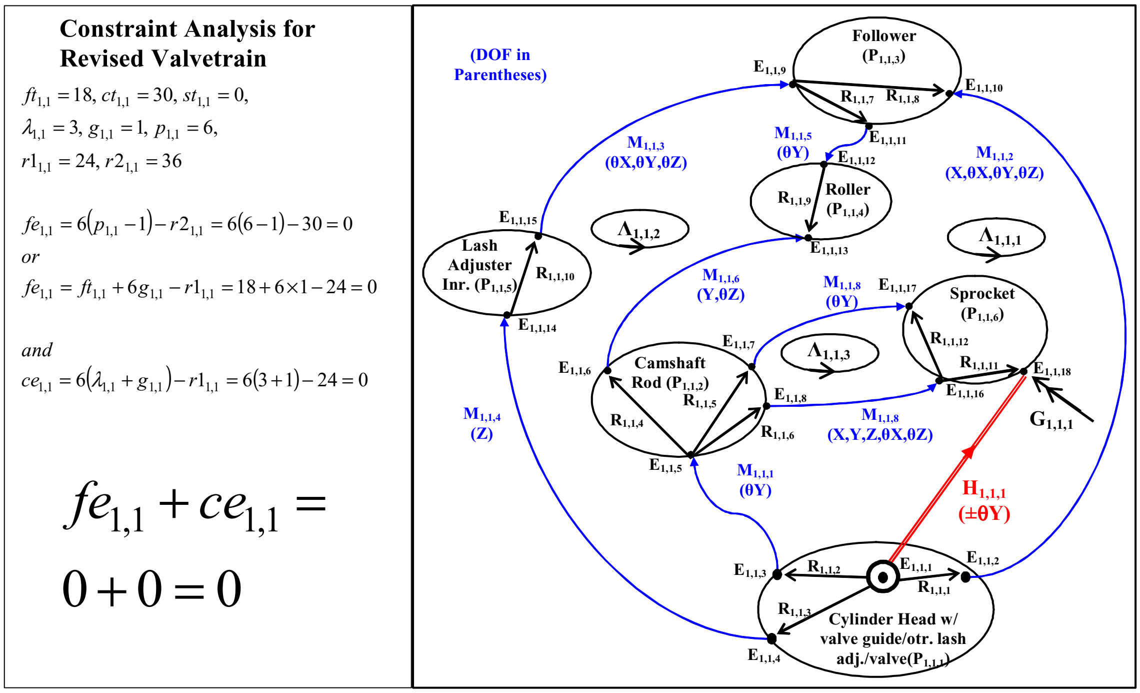
The two over-constraints are in the ***θy*** and ***θz*** directions in closed-loop $\Lambda_{1,1,1}$ that is shared by the camshaft and the sprocket. These over-constraints can be eliminated by changing the mate between the key and the keyway so that it has five DOF instead of three. Due to the clearance between the keyway and the key in the real assembly, this is actually the more accurate representation of the mating condition.

Of the six under-constraints, three are simply the rotations in the ***θz*** direction allowed by the mates between the outer lash adjuster and the cylinder head, the inner and out lash adjusters, and the valve guide and the valve. While such under-constraints are necessary in a real valvetrain so that the parts wear evenly they are not needed in this assembly model since to first-order they do not affect the KCs of interest. For this reason these three DOF can be temporarily removed and turned into DOC.

A fourth under-constraint is the motion in the z direction allowed between the valve and the valve guide. Since this configuration is defined to be at max-lift, this value can be fixed[13]. The last two under-constraints are present in the mate between the camshaft and the roller. By changing the number of DOF of this mate from four to two, these under-constraints can also be eliminated. This in effect imposes a no-slip condition on the mate between the cam lobe and roller and in this case, such an assumption is reasonable.

Figure 6.7 shows both the annotated liaison diagram for a valvetrain including the changes discussed above. Figure 6.7 also shows the results from the motion/constraint analysis. It can be seen that this new assembly contains no excess over-constraints or under-constraints.

[13] For proprietary reasons that actual value of this dimension is not shown here.



Constraint Analysis for Revised Valvetrain

$ft_{1,1}=18,\ ct_{1,1}=30,\ st_{1,1}=0,$

$\lambda_{1,1}=3,\ g_{1,1}=1,\ p_{1,1}=6,$

$r1_{1,1}=24,\ r2_{1,1}=36$

$fe_{1,1}=6(p_{1,1}-1)-r2_{1,1}=6(6-1)-30=0$

or

$fe_{1,1}=ft_{1,1}+6g_{1,1}-r1_{1,1}=18+6\times1-24=0$

and

$ce_{1,1}=6(\lambda_{1,1}+g_{1,1})-r1_{1,1}=6(3+1)-24=0$

$$fe_{1,1}+ce_{1,1}= 0+0=0$$

Figure 6.7. Motion/Constraint Analysis of Revised Valvetrain

6.3.2 Sensitivity Analysis

Using the new assembly model, the KC sensitivities, KC Cpks, and KC contributions were obtained using the Matlab® program manufacturability.m. Since there are only two KCs and the correlation between them is obvious it was not necessary to study the KC coupling in this case. Table 6.2 shows the sensitivity of the KCs the dimensions (placed in order of sensitivity magnitude). It can bee seen that the timing of the camshaft sprocket is very sensitive to several dimensions on the cylinder head, namely the alignment of the camshaft bore in the cylinder head (***D_1***, ***D_3***, ***D_{10}***, and ***D_{17}***) and the alignment of the lash adjuster bore in the cylinder head (***D_5*** and ***D_{15}***). Sprocket timing is also sensitive to the straightness of the camshaft, lash adjuster, and valve (***D_2***, ***D_4***, ***D_{13}***, ***D_{18}***, ***D_6***, ***D_{16}***, ***D_{23}***, and ***D_{25}***), the sprocket mounting dimensions (***D_7***, ***D_8***, ***D_9***, and ***D_{10}***), the roller, cam lobe, and follower tip radii ($\boldsymbol{D_{12}}$, $\boldsymbol{D_{14}}$, and $\boldsymbol{D_{21}}$), the valve guide and roller alignment ($\boldsymbol{D_{11}}$, $\boldsymbol{D_{19}}$, $\boldsymbol{D_{22}}$, and $\boldsymbol{D_{24}}$), and the valve length ($\boldsymbol{D_{20}}$). Some of these dimensions are shown schematically in Figure 6.8.

#	Sensitivity of KC_1 (+θY of Sprocket)	Sensitivity of KC_2 (+θY of Sprocket)	Units	Dimension Description
D_1	-4.0132	4.0132	(rad/rad)	alignment (θz) of camshaft bore in cylinder head
D_2	-4.0132	4.0132	(rad/rad)	straightness (θz) of camshaft
D_3	1.8408	-1.8408	(rad/rad)	alignment (θx) of camshaft bore in cylinder head
D_4	1.8408	-1.8408	(rad/rad)	straightness (θx) of camshaft
D_5	-0.6183	0.6183	(rad/rad)	alignment (θy) of lash adjuster bore in cylinder head
D_6	-0.6183	0.6183	(rad/rad)	straightness (θy) of lash adjuster
D_7	-0.0548	0.0548	(rad/mm)	alignment (x) of keyway in camshaft
D_8	-0.0548	0.0548	(rad/mm)	alignment (x) of key in sprocket
D_9	0.0547	-0.0547	(rad/mm)	alignment (x) of sprocket-mounting shaft in camshaft
D_{10}	0.0235	-0.0235	(rad/mm)	alignment (x) of camshaft bore in cylinder head
D_{11}	-0.0235	0.0235	(rad/mm)	alignment (x) of roller axis in follower
D_{12}	-0.0235	0.0235	(rad/mm)	radius (z) of roller
D_{13}	0.0235	-0.0235	(rad/mm)	straightness (x) of camshaft
D_{14}	0.0235	-0.0235	(rad/mm)	radius (z) of cam lobe
D_{15}	-0.0221	0.0221	(rad/mm)	alignment (x) of lash adjuster bore in cylinder head
D_{16}	-0.0221	0.0221	(rad/mm)	straightness (x) of lash adjuster
D_{17}	0.0108	-0.0108	(rad/mm)	alignment (z) of camshaft bore in cylinder head
D_{18}	0.0108	-0.0108	(rad/mm)	straightness (z) of camshaft
D_{19}	-0.0108	0.0108	(rad/mm)	alignment (z) of roller axis in follower
D_{20}	-0.0095	0.0095	(rad/mm)	valve length (z)
D_{21}	-0.0095	0.0095	(rad/mm)	radius of follower tip that mates to valve (z)
D_{22}	-0.0094	0.0094	(rad/mm)	alignment (x) of valve guide in cylinder head
D_{23}	-0.0094	0.0094	(rad/mm)	straightness (x) of valve
D_{24}	0.0012	-0.0012	(rad/rad)	alignment (θy) of valve guide in cylinder head
D_{25}	0.0012	-0.0012	(rad/rad)	straightness (θy) of valve

Table 6.2. Results of Sensitivity Analysis for RFF Valvetrain

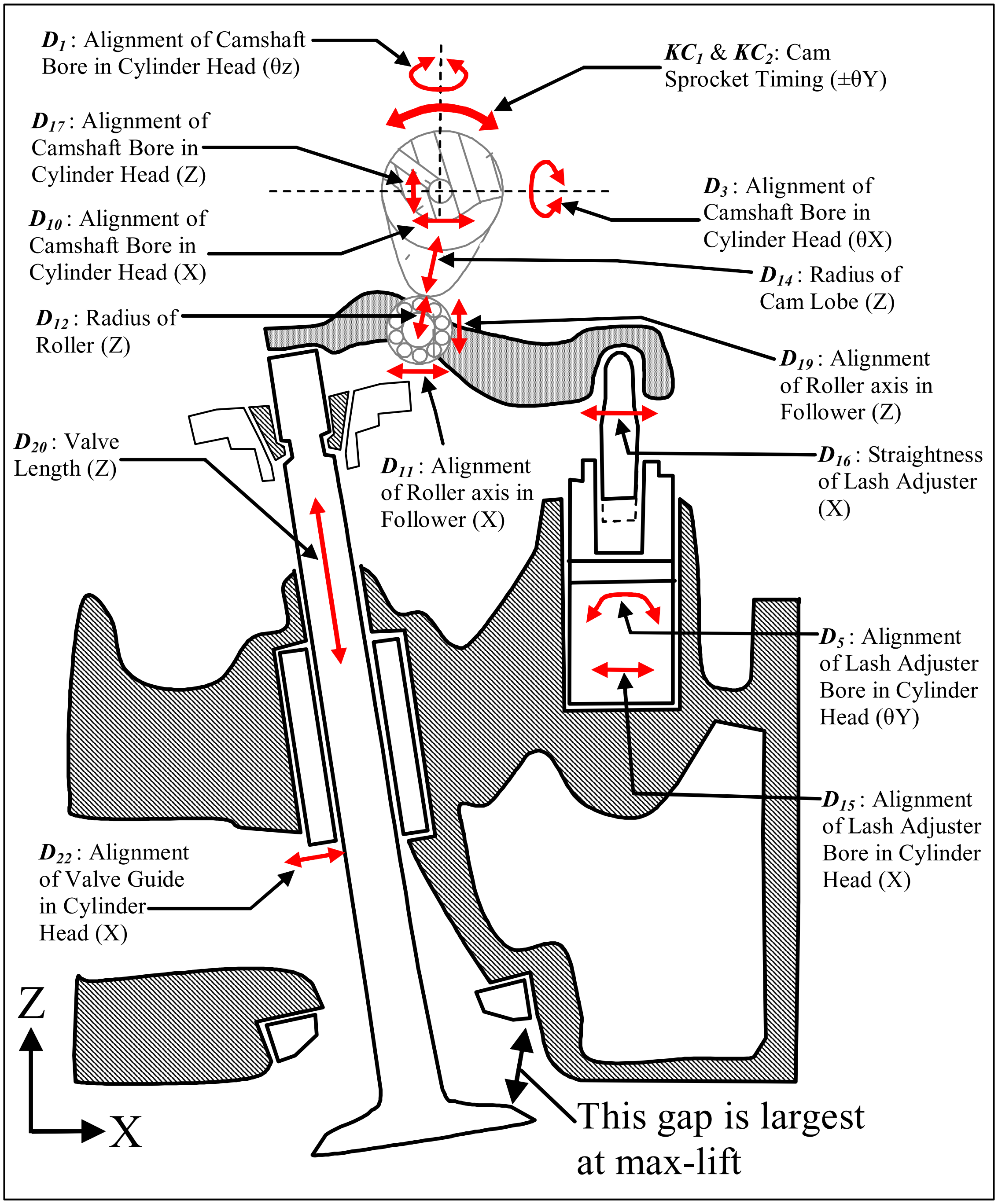

D₁ : Alignment of Camshaft Bore in Cylinder Head (θz)
KC₁ & *KC₂*: Cam Sprocket Timing (±θY)
D₁₇ : Alignment of Camshaft Bore in Cylinder Head (Z)
D₃ : Alignment of Camshaft Bore in Cylinder Head (θX)
D₁₀ : Alignment of Camshaft Bore in Cylinder Head (X)
D₁₄ : Radius of Cam Lobe (Z)
D₁₂ : Radius of Roller (Z)
D₁₉ : Alignment of Roller axis in Follower (Z)
D₂₀ : Valve Length (Z)
D₁₁ : Alignment of Roller axis in Follower (X)
D₁₆ : Straightness of Lash Adjuster (X)
D₅ : Alignment of Lash Adjuster Bore in Cylinder Head (θY)
D₁₅ : Alignment of Lash Adjuster Bore in Cylinder Head (X)
D₂₂ : Alignment of Valve Guide in Cylinder Head (X)
Z
X
This gap is largest at max-lift

Figure 6.8. Schematic of Valvetrain at Max Lift Showing Some Critical Dimensions

For comparison, another assembly model was created (using modeler.m) for the same valvetrain but in a different configuration; the position right before the valve starts to open (see Figure 6.9). This was done so that is could be compared directly to an Excel®-based valvetrain sensitivity model of the identical configuration[14]. Specifically, the sensitivities of the camshaft rotation and lash adjuster height to a select set of dimensions obtained using manufacturability.m were compared to the identical sensitivities derived from the Excel®-based model. It can be seen in Table 6.3 that the results from the two methods are in very close agreement (less than 1% difference).

Sensitivity of:	**Units**	**Excel®-Based Model**	**Manufacturability.m**	**Percent Difference (%)**
Camshaft Rotation (θY) to:				
valve length (z)	(deg/mm)	0.2389	0.2404	**0.640**
camshaft position (x)	(deg/mm)	1.6167	1.6163	**0.024**
roller postion (x)	(deg/mm)	1.5826	1.5825	**0.009**
lash adjuster postion (x)	(deg/mm)	0.2071	0.2071	**0.022**
Lash Adjuster Height (Z) to:				
valve length (z)	(mm/mm)	1.2787	1.2794	**0.052**
camshaft position (x)	(mm/mm)	0.0196	0.0195	**0.270**
roller postion (x)	(mm/mm)	0.2029	0.2026	**0.130**
lash adjuster postion (x)	(mm/mm)	1.5704	1.5707	**0.018**

Table 6.3. Comparison of Sensitivities from Excel® Model and Manufacturability.m

[14] This model was also created by the author with help from employees at Ford Motor Company.

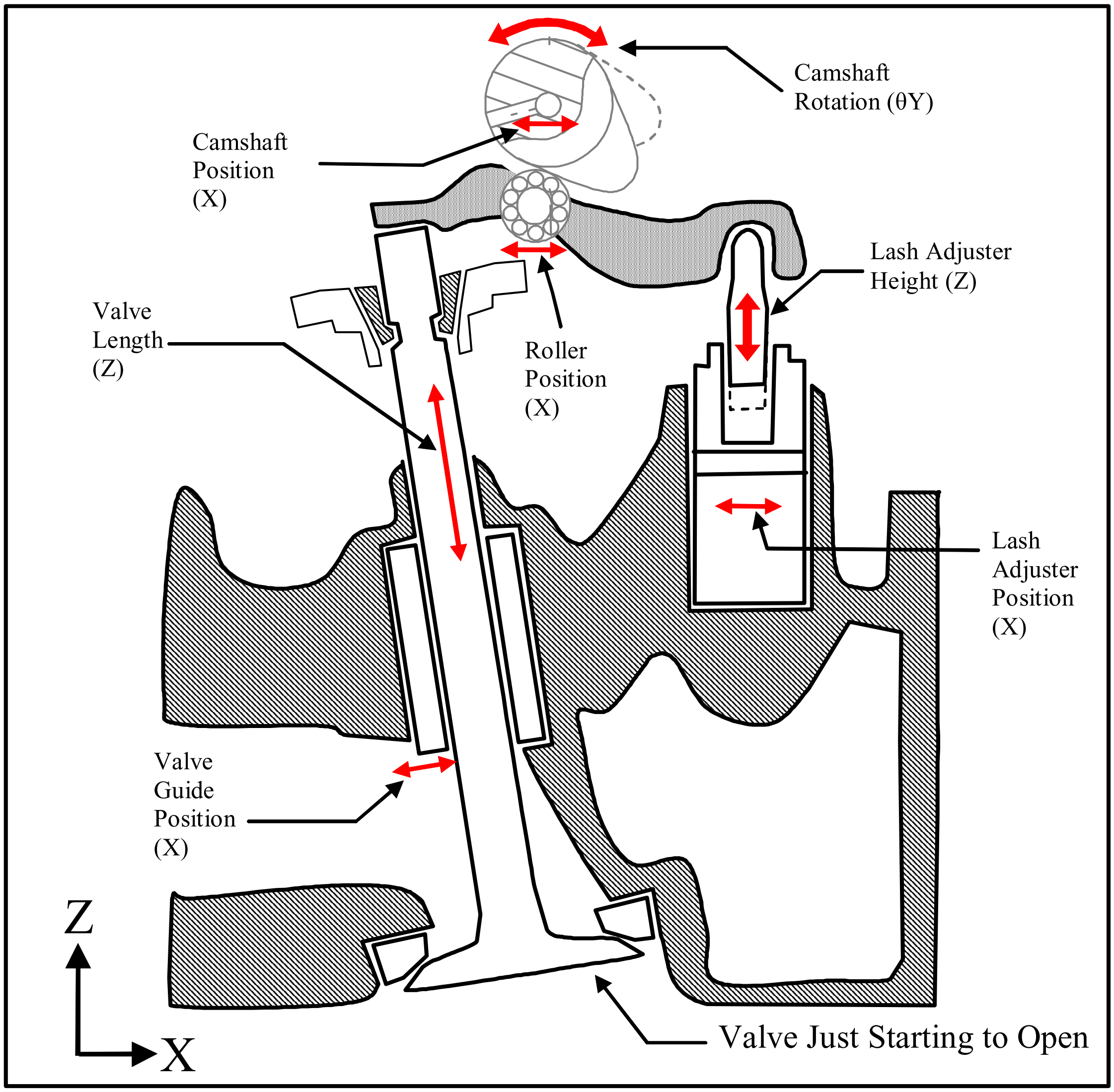


Figure 6.9. Valvetrain Configuration Just Before Valve Starts to Open

6.3.3 Performance Analysis

Using manufacturability.m and the data regarding the variability in the fabrication processes[15], the Cpks and reject rates of the KCs were obtained for the valvetrain at max-lift and these are shown in Table 6.4. Assuming the target Cpk is 1.33, the results in

[15] The variability in the fabrication processes for each dimension was estimated by assuming that each was normally distributed, mean centered, and had a Cpk of 1.33 with respect to the tolerance band assigned to the dimension on the part drawings. Due to the proprietary nature of these values, they are not shown here.

Table 6.4 reveals that the manufacturability of the valvetrain is unacceptably low since the Cpks of both KCs are predicted to be 1.01.

	Estimates Using Manufactuability.m		Estimates Using Modified VSA® Model	
KC	**KC Cpk**	**Reject Rate (%)**	**KC Cpk**	**Reject Rate (%)**
KC$_1$ (+θY of Sprocket)	**1.01**	**0.128**	**0.97**	**0.181**
KC$_2$ (+θY of Sprocket)	**1.01**	**0.128**	**0.97**	**0.181**

Table 6.4. Comparison of Cpk Estimates from VSA® and Manufacturability.m

For comparison, Table 6.4 also includes the results from a Monte Carlo simulation-based approach using a model developed with VSA® software [VSA]. Since the version of VSA® used was not capable of modeling assemblies with closed-loops, it had to be modified[16]. As can be seen, the results from manufacturability.m and the VSA®-based model are in very close agreement. The most noticeable difference between the methods is not in the results but in the computational efficiency. Whereas typical models in VSA® take several hours to produce results[17], manufacturability.m can usually obtain results in several minutes[18].

[16] Tim Bohr, an employee of Ford Motor Company, modified the basic VSA-3D software to create this model and modify the program so that the closed-loop nature of the valvetrain was captured.

[17] Dan Jakobcic, an employee of Ford Motor Company, provided this estimate during a personal communication with the author.

[18] This runtime estimate is for an assembly with similar complexity to the RFF valvetrain that is run on PC computer with an Intel® Pentium® 4 processor.

6.3.4 Contribution Analysis

Using manufacturability.m, Contribution Analysis was conducted for the valvetrain and the results are shown in Table 6.5. The rows in Table 6.5 corresponding to the dimensions in the valvetrain are ordered from highest to lowest contribution.

#	Contribution to Variability of KC_1 (%)	Contribution to Variability of KC_2 (%)	Dimension Description
D_7	20.92	20.92	alignment (x) of keyway in camshaft
D_5	13.09	13.09	alignment (θy) of lash adjuster bore in cylinder head
D_{11}	12.64	12.64	alignment (x) of roller axis in follower
D_8	12.26	12.26	alignment (x) of key in sprocket
D_{10}	6.45	6.45	alignment (x) of camshaft bore in cylinder head
D_9	6.43	6.43	alignment (x) of sprocket-mounting shaft in camshaft
D_{15}	5.77	5.77	alignment (x) of lash adjuster bore in cylinder head
D_{21}	4.86	4.86	radius of follower tip that mates to valve (z)
D_1	3.78	3.78	alignment (θz) of camshaft bore in cylinder head
D_{16}	3.49	3.49	straightness (x) of lash adjuster
D_{24}	2.69	2.69	alignment (θy) of valve guide in cylinder head
D_{19}	2.66	2.66	alignment (z) of roller axis in follower
D_{17}	1.36	1.36	alignment (z) of camshaft bore in cylinder head
D_{22}	1.07	1.07	alignment (x) of valve guide in cylinder head
D_3	0.80	0.80	alignment (θx) of camshaft bore in cylinder head
D_{13}	0.41	0.41	straightness (x) of camshaft
D_6	0.40	0.40	straightness (θy) of lash adjuster
D_{20}	0.35	0.35	valve length (z)
D_2	0.23	0.23	straightness (θz) of camshaft
D_{14}	0.11	0.11	radius (z) of cam lobe
D_{12}	0.10	0.10	radius (z) of roller
D_{18}	0.09	0.09	straightness (z) of camshaft
D_4	0.05	0.05	straightness (θx) of camshaft
D_{23}	0.00	0.00	straightness (x) of valve
D_{25}	0.00	0.00	straightness (θy) of valve

Table 6.5. Contribution Analysis Results Obtained Using Manufacturability.m

It can be seen in Table 6.5 that the variability in the key and keyway are most responsible for the variability in the sprocket rotation. The alignment of the bores for the camshaft and lash adjuster in the cylinder head are also big contributors.

Table 6.6 shows a comparison between the results obtained using manufacturability.m with the results obtained using the modified VSA$^{®}$-based valvetrain model for a few of the top contributors. The main differences are in the alignment of the roller axis and the alignment of the keyway. Further analysis revealed that in the case of the roller axis alignment, the VSA$^{®}$-based model included a lower variability for that dimension and this resulted in a lower contribution. This assumption was based on actual data received from the part supplier that indicated that the roller axis location on the follower was aligned more precisely than the drawing-specified tolerance. In the case of the key alignment, the VSA$^{®}$ model treated the clearance between the key and keyway and the alignment of the keyway as two separate dimensions whereas the model presented here combined them into a single dimension. This is most likely the reason the prediction using manufacturability.m is about twice the prediction using the VSA$^{®}$-based model.

#	Contribution Calculated Using Manufactuability.m	Contribution Calculated Using Modified VSA® Model	Dimension Description
D_7	20.92	25.63 (Includes D_9)	alignment (x) of keyway in camshaft
D_5	13.09	Not Included	alignment (θy) of lash adjuster bore in cylinder head
D_{11}	12.64	5.75	alignment (x) of roller axis in follower
D_8	12.26	5.76	alignment (x) of key in sprocket
D_{10}	6.45	6.48	alignment (x) of camshaft bore in cylinder head
D_9	6.43	Included in D_7	alignment (x) of sprocket-mounting shaft in camshaft
D_{15}	5.77	7.17	alignment (x) of lash adjuster bore in cylinder head

Table 6.6. Comparison of Contributions from VSA® and Manufacturability.m

6.4 Improvements

Since the Cpks of the KCs are below the target of 1.33, it is worthwhile to look at what changes might lead to improved values for these Cpks. At the system-level, there are several possibilities. One option is to change the type of valvetrain. For example, instead of a roller finger follower, or RFF, valvetrain, a direct acting mechanical bucket, or DAMB, valvetrain could be used (see Figure 6.10). DAMB valvetrains do not use a follower or lash adjuster but use a flat bucket instead. Switching to a DAMB valvetrain is advantageous because it completely eliminates the contributions of $\boldsymbol{D_{10}}$, $\boldsymbol{D_{11}}$, and $\boldsymbol{D_{12}}$ to variability in cam sprocket timing as shown in Figure 6.10. From Table 6.5, it can be seem that these three dimension account for 25% of the cam sprocket timing error! This represents a substantial improvement. However, since the camshaft in a DAMB valvetrain is directly over the valve, each bank usually requires two camshafts for each back of pistons. For this and other reasons, switching from a RFF valvetrain to a DAMB valvetrain represents a significant system-level change that is usually not possible unless the design is in the early concept phase.

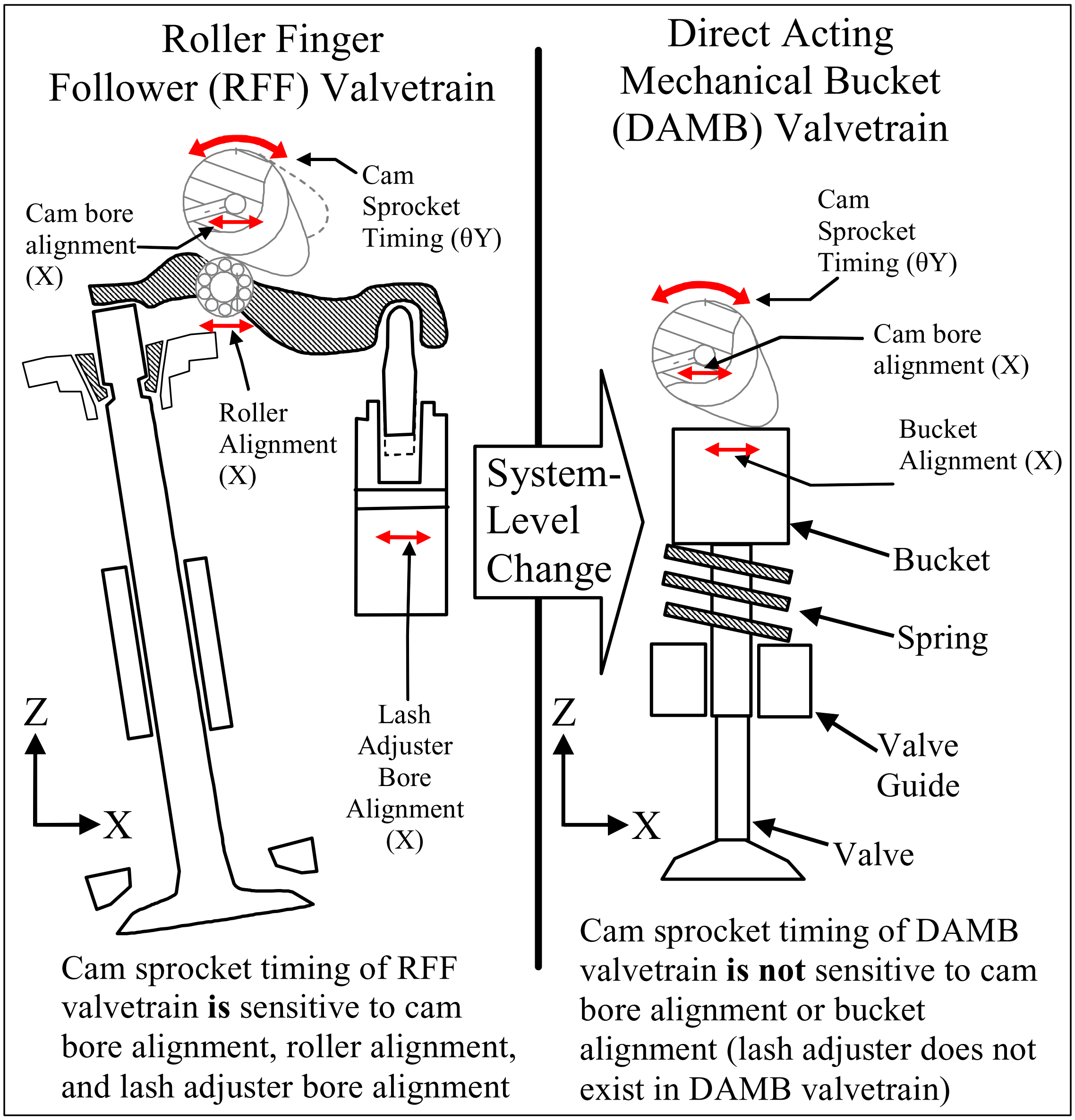

Roller Finger
Follower (RFF) Valvetrain
Cam bore
alignment
(X)
Cam
Sprocket
Timing (θY)
Roller
Alignment
(X)
Lash
Adjuster
Bore
Alignment
(X)
Z
X
Cam sprocket timing of RFF valvetrain **is** sensitive to cam bore alignment, roller alignment, and lash adjuster bore alignment
System-
Level
Change
Direct Acting
Mechanical Bucket
(DAMB) Valvetrain
Cam
Sprocket
Timing (θY)
Cam bore
alignment (X)
Bucket
Alignment (X)
Bucket
Spring
Valve
Guide
Valve
Z
X
Cam sprocket timing of DAMB valvetrain **is not** sensitive to cam bore alignment or bucket alignment (lash adjuster does not exist in DAMB valvetrain)

Figure 6.10. One Possible System-Level Change

Another system-level option is to change the assembly process. For example, the sprocket could be mounted onto the camshaft using a precision press fit operation[19]. This

[19] This is in fact what Ford Motor Company has done to reduce cam sprocket timing variability with some of their engines that have RFF valvetrains.

would reduce the effect of $\boldsymbol{D_7}$, $\boldsymbol{D_8}$, and $\boldsymbol{D_9}$, which together account for 40% of the variability in cam sprocket timing.

At the parameter level, one might decrease some of the clearances between the parts, especially between the key in the sprocket and the keyway in the camshaft. However, since there are many other valvetrain-related KCs in addition to the vibration caused by timing error, one must insure than any changes in the nominal dimension values do not have adverse effects on these other KCs.

At the tolerance level, the most logical place to start is with the dimensions that are the highest contributors to the variability in the KCs. Working with the supplier of the sprockets and followers, one could try to reduce the variability of the dimensions in those parts. Since the cylinder head is manufactured in-house, it may also be advantageous to see if the variability in the boring operations for the lash adjuster bore and camshaft bore can be reduced.

6.5 Summary

This chapter presented a case study involving the application of the methodology presented in this work an RFF valvetrain of an engine. Using the Matlab® programs included in Appendix A, a model was created and used to verify the manufacturability of the valvetrain. Since the predicted Cpks of the KCs were below the target Cpk, the results from the verification effort were used to suggest several ways to improve the manufacturability of the design. On the whole, comparison of the sensitivity, performance, and contribution results obtained using the Matlab® programs to results from other industry-standard tools showed close agreement. The main advantage of the method presented in this thesis as compared to some other methods is its computational efficiency. Whereas other methods might take hours to produce results, this method can produce results in minutes.

7. CONCLUSIONS

7.1 Summary

This thesis has presented a unified approach to modeling, verifying, and improving the manufacturability of mechanical assemblies. Chapter 1 introduced the goals of this thesis and presented some of the additional assumptions that the methodology is based on. Chapter 1 also reviewed the relevant prior research and provided a summary of the approach that was used in this work.

Chapter 2 laid out a framework for discussion by defining a set of terminology and how it was interpreted in this work. Chapter 2 also took a closer look at the various characteristics of mechanical assemblies that relate to assembly behavior. Chapter 2 concluded with a brief discussion of what aspects of assemblies were and were not included in the methodology presented.

Chapter 3 discussed in detail the modeling procedure that can be used to create a quantitative representation of a mechanical assembly design that accurately models its nominal and variable forms. This procedure is divided into two phases: the Element Identification & Location Phase and the Element Quantification & Integration Phase.

Chapter 4 revealed how the model developed in Chapter 3 can be used to verify the manufacturability of an assembly. The verification process is composed of a suite of analysis techniques, namely Motion/Constraint, Sensitivity, Performance, Contribution, and Coupling Analyses.

Chapter 5 discussed how the results from Chapter 3 and Chapter 4 can be used to support efforts to improve the manufacturability of the design. These efforts are divided into three categories: System-Level, Parameter-Level, and Tolerance-Level Strategies.

Chapter 6 showed how this methodology was applied to the design of a Roller Finger Follower, or RFF, Valvetrain on a V-8 Ford® Engine. The scope was limited by looking at a single customer expectation: the level of unwanted vibration caused by the engine. This expectation was mapped geometrically to the rotation angle of the cam sprocket, since any error that angle causes undesirable and noticeable vibration. Application of the methodology revealed that the combination of high variability in the manufacturing processes and high sensitivity of the design to that variability resulted in an assembly with low manufacturability. As validation of the methodology, these results were compared to predictions using other industry-standard tools. Chapter 6 concluded by using the assembly model and the results it provided to suggest several ways to improve the manufacturability of the valvetrain.

The strengths of the methodology developed in this thesis are several. First, within the assumptions and constraints listed at the beginning of the work, the method is general and context-free. This means that it can be applied equally to something very simple, like a fingernail clipper, or something very complex, like a valvetrain. Second, all the mathematics are amendable to matrix-based computation. This has made it possible to automate much of the methodology using Matlab® (see Appendix A). Third, by using probabilistic techniques such as the MPP concept, the need to use Monte Carlo simulation to calculate the sensitivities of the KCs to variability in the dimensions, the Cpks of the KCs, and contributions of the variability in the dimensions to the variability in the KCs was eliminated. This greatly reduced the complexity of the computation and opened the door for some of the optimization techniques presented in Chapter 5. Also, whereas current industry standard-tool for assessing manufacturability such as VSA® can take hours to produce results, this methodology can usually produce results in minutes.

Also brought to the surface were a several weaknesses of the methodology. First, the most obvious weakness is the need to convert all problem parameters into geometric equivalents. Needless to say, this can be very difficult in some cases. Second, related to this limitation is the inability to deal with important part properties such as mass. While stiffness is included, it is only covered in a lumped-parameter sense. Third, while handling dimensional variability in a full distribution sense, the methodology is currently limited to the case of independent variables. This limitation means that there are some situations when the methodology is incapable of providing meaningful results. A fourth weakness relates to the evaluative nature of the methodology. In order to produce results, a large amount of information must be known about the design. Often one would like to make judgments much earlier in the process when the final architecture of the design is not finalized or all the properties of the parts and the mates have not been specified. It is also possible that the manufacturing processes are not that well understood making geometric characterization difficult. Adapting this methodology so that the amount of detailed input required is reduced would open the door for using this methodology as a driver of design and not just as an evaluative tool.

7.2 Suggestions for Future Work

Based on the finding in the discussion above, much of the future work falls into three categories:

- **Improving Existing Capabilities**
- **Extending Methodology to Support Design**
- **Improving Computer Implementation**

Each of these categories will now be discussed.

7.2.1 Improving Existing Capabilities

The methodology in its current state is limited in the discussion of the nominal design to bidirectional mates that are frictionless, and parts that are massless. However, the mathematical underpinnings of the presented modeling techniques allow for the expansion of the model to accurately include such effects. For example [Roth] and [Ohwovoriole and Roth] discuss how Screw Theory can be extended to include unidirectional mates and friction. In addition, it may be possible to add another matrix to

7.2.2 Extending Methodology to Support Design

Another area for growth of this methodology is the applicability of it to the different phases of the design process. Currently, this methodology is extremely useful, as has been shown, to evaluate and improve a product once all or most of the details of the design have been identified. Ideally, this methodology could also be used earlier in the design process when making decisions about system architecture, competing viable concepts, or competing KCs. To make this possible, it would most likely mean that the mathematical model would have to be simplified so that the amount of input data required is reduced. While such an effort would invariably lead to a model whose outputs were less accurate than the one presented here, it may still provide insight that would help engineers make key design decisions early in the design process.

7.2.3 Improving Computer Implementation

Many of the modeling techniques presented in Chapter 3, Chapter 4, and Chapter 5 would be extremely difficult to implement manually. It is therefore advantageous to automate these techniques. Appendix A shows how many of these techniques can be automated using Matlab® software. Ideally, this methodology would be incorporated into a 3-D CAD modeling software such as Pro/Engineer®, Catia®, I-DEAS®, or Solidworks®. Then most of the input data could be taken directly from the CAD model. In addition, a graphical query-based user interface could facilitate the input of the remaining data such as KCs, assembly topology, and dimensional variability distributions and parameters.

REFERENCES

- Adams, J. D., “Feature-Based Analysis of Selective Limited Motion in Assemblies,” S.M. Thesis, Massachusetts Institute of Technology, Mechanical Engineering Department, Cambridge, MA, 1998.

- Adams J. D., and Whitney D. E., “Application of Screw Theory to Motion and Constraint Analysis of Mechanical Assemblies,” Proceedings of the IEEE Intl. Symposium on Assembly and Task Modeling, Porto, Portugal, July 1999.

- Ahuja D. V., and Coons S. A., “Geometry for Construction and Display,” IBM Systems Journal, vol. 7, no. 3 and 4, pp. 188-217, 1968.

- Anantha, R., Kramer, G. A., and Crawford, R. H., “Assembly Modeling By Geometric Constraint Satisfaction,” Computer-Aided Design, vol. 28, no. 9, pp. 707-722, 1995.

- Arora, J. S., “Key Characteristic Coupling and Resolving Key Characteristic Conflict,” S.M. Thesis, Massachusetts Institute of Technology, Department of Mechanical Engineering, Cambridge, MA, 2003.

- Ashiagbor A., Liu H. C., and Nnaji, B. O., “Tolerance Control and Propagation for the Product Assembly Modeler,” International Journal of Production Research, vol. 36, no. 1, pp. 75-93, 1998.

- Baker J. E., “On Relative Freedom between Links in Kinematic Chains with Cross-Jointing,” Mechanism and Machine Theory, vol. 15, pp. 397-413, 1980.

- Ball R. S. Sir, A Treatise on the Theory of Screws, Cambridge University Press, Cambridge, England, 1900.

- Baldwin, D. F., Abell, T. E., Lui, M.-C., De Fazio, T. L., and Whitney, D. E., "An Integrated Computer Aid for Generating and Evaluating Assembly Sequences for Mechanical Products,” IEEE Transactions on Automation and Robotics, vol. 7, no. 1, pp 78-94, 1991.

- Bihlmaier, B. F., “Tolerance Analysis of Flexible Assemblies Using Finite Element and Spectral Analysis,” S.M. Thesis, Brigham Young University, Provo, UT, 1999.

- Bjørke O., Computer-aided Tolerancing, 2nd edition, ASME Press, New York, NY, 1989.

- Blanding, D., Exact Constraint Design, ASME Press, New York, NY, 1999.

- Bourjault, A., “Contribution á une Approche Méthodologique de l’ Assemblage Automatisé: Evaluation automatique des séquences Operatoires,” Thesis to obtain Grade de Docteur des Sciences Physiques at l’ Université de France-Comté, November, 1984.

- Bralla, J., Design For Manufacturability Handbook 2nd edition, McGraw-Hill, New York, NY, 1998.

- Camelio, J. A., Hu, S. J., and Ceglarek, D., "Modeling Variation Propagation of Multi-Station Assembly Systems with Compliant Parts,” Proceedings of 2001 ASME Design Engineering Technical Conference, Pittsburgh, PA, 2001.

- Campbell, P., Basic Fixture Design, Industrial Press, New York, NY, 1994.

- Ceglarek, D., and Shi, J., “Dimensional Variation Reduction for Automotive Body Assembly,” Manufacturing Review, vol. 8, pp. 139-154, 1995.

- Chase, K. W., J. Gao, S. P. Magleby and C. D. Sorenson, "Including Geometric Feature Variations in Tolerance Analysis of Mechanical Assemblies", IIE Transactions, vol. 28, pp. 795-807, 1996.

- Chase, K. W., Greenwood, W. H., Loosli, B. G., and Hauglund, L. F., “Least Cost Tolerance Allocation for Mechanical Assemblies with Automated Process Selection,” Manufacturing Review, ASME, vol. 3, no. 1, pp. 49-59, March 1990.

- Chase, K. W., and Parkinson, A. R., “A Survey of Research in the Application of Tolerance Analysis to the Design of Mechanical Assemblies,” Research in Engineering Design, vol. 3, pp. 23-37, 1991.

- Chang, C. F., and Perng, D. B., “Assembly-Part Automatic Positioning Using High-Level Entities of Mating Features,” Computer Integrated Manufacturing Systems, vol. 10, no. 3, pp. 205-215, 1997.

- Chang, M., and Gossard, D. C., “Modeling the Assembly of Compliant, non-Ideal Parts,” Computer-Aided Design, vol. 29, no. 10, pp. 701-708, 1997.

- Clausing, D., Total Quality Deployment, ASME Press, New York, NY, 1994.

- Clement, A., et al, “Theory and Practice of 3-D Tolerancing for Assembly,” 1st CIRP Seminar on Computer Aided Tolerancing, Israel, 1991.

- Clement, A., and Riviere, A., "Tolerancing Versus Nominal Modeling in Next Generation CAD/CAM System," 3rd CIRP Seminar on Computer Aided Tolerancing, Cachan, France, pp. 97-114, 1993.

- Dai, J. S., and Jones, J. R., "Null-Space Construction Using Cofactors From A Screw-Algebra Context," Proceedings of The Royal Society Mathematical, Physical and Engineering Sciences, vol. 458, no. 2024, pp. 1783-2038, 2002.

- Davidson, J. K., and Shah, J. J., "Geometric Tolerances: A New Application For Line Geometry and Screws," Proceedings of A Symposium Commemorating the Legacy, Works, and Life of *Sir Robert Stawell Ball* Upon the 100th Anniversary of *A Treatise on the Theory of Screws*, University of Cambridge, Trinity College, 2000.

- Davidson, J. K., Mujezinović, A., and Shah, J. J., "A New Mathematical Model For Geometric Tolerances as Applied to Round Faces," Journal of Mechanical Design, vol. 124, pp 609-621, 2002.

- Davies, T. H., "Kirchoff's Circulation Law Applied to Multi-Loop Kinematic Chains," Mechanism and Machine Theory, vol. 16, pp. 171-183, 1981.

- Davies, T. H., "Mechanical Networks– I: Passivity and Redundancy," Mechanism and Machine Theory, vol. 18, no. 2, pp. 95-101, 1983a.

- Davies, T. H., "Mechanical Networks-II: Formulae for the Degrees of Mobility and Redundancy," Mechanism and Machine Theory, vol. 18, no. 2, pp. 103-106, 1983b.

- Davies, T. H., "Mechanical Networks-III: Wrenches On Circuit Screws," Mechanism and Machine Theory, vol. 18, no. 2, pp.107-112, 1983c.

- Davies, T. H., "Couplings, Coupling Networks and Their Graphs," Mechanism and Machine Theory, vol. 30, no. 7, pp. 991-1000, 1995.

- Davies, T. H., "The 1887 Committee Meets Again. Subject: Freedom and Constraint.," Proceedings of a Symposium Commemorating the Legacy, Works, and Life of *Sir Robert Stawell Ball* Upon the 100th Anniversary of *A Treatise on the Theory of Screws*, University of Cambridge, Trinity College, England, 2000.

- Davies, T. H., and Primrose E. J. F., "An Algebra for the Screw Systems of Pairs of Bodies in a Kinematic Chain," Proc. 3rd World Cong. for the Theory of Machines and Mechanisms, Kupari, Yugoslavia, Paper D-14, pp.199-212, 1971.

- De Fazio, T.L. et al, "A Prototype for Feature-Based Design for Assembly," ASME Journal of Mechanical Design, vol. 115, pp. 723-734, 1993.

- Denavit, J., and Hartenberg, R. S. "A Kinematic Notation for Lower Pair Mechanisms Based on Matrices," Journal of Applied Mechanics, vol. 22, pp. 215-221, 1965.

- Desrochers, A., "Modeling Three Dimensional Tolerance Zones Using Screw Parameters," Proceedings of 25th ASME Design Automation Conference, 1999.

- Ditlevsen, O., and Madsen, H. O., Structural Reliability Methods, John Wiley & Sons, Baffins Lane, England, 1996.

- Ding, Y., Ceglarek, D., Jin, J., Shi, J., "Process-Oriented Tolerance Synthesis For Multistage Manufacturing Systems," Manufacturing Science and Engineering, ASME 2000, vol. 11, pp. 15-22, 2000.

- Donald, B. R., and Pai, D. K., "On the motion of Compliantly-Connected Rigid Bodies in Contact, Part II: A System for Analyzing Designs for Assembly," Proceedings of the 1990 IEEE International Conference on Robotics and Automation, Cincinnati, OH, May 1990.

- Drake, A.W., Fundamentals of Applied Probability Theory, McGraw-Hill, New York, NY, 1967.

- Fasse, E., "Some Applications of Screw Theory to Lumped-Parameter Modeling of Visco-Elastically Coupled Rigid Bodies," Proceedings of A Symposium Commemorating the Legacy, Works, and Life of *Sir Robert Stawell Ball* Upon the 100th Anniversary of *A Treatise on the Theory of Screws*, University of Cambridge, Trinity College, England, 2000.

- Fasse, E., and Zhang, S., "Lumped-Parameter Modeling of Spatial Compliance Using a Twist-Based Potential Function," Proceedings of the ASME Dynamic Systems and Control Division, vol. 15, pp. 779-786, 1999.

- Foster, L. W., Geo-Metrics II, Addison-Wesley, Reading, MA, 1994.

- Gao, J., Chase, K. W., and Magleby, S. P., "Generalized 3-D Tolerance Analysis of Mechanical Assemblies with Small Kinematic Adjustments," IIE Transactions, vol. 30, no. 4, pp. 367-377, 1998.

- Goldenshteyn, G., “Design of Assemblies with Compliant Parts: Application to Datum Flow Chains,” S.M. Thesis, Massachusetts Institute of Technology, Department of Mechanical Engineering, Cambridge, MA, 2002.

- Green, W. G., Theory of Machines, Blackie and Son Ltd., London, England, 1961.

- Gui, J., and Mäntylä, M., “Functional Understanding of Assembly Modeling,” Computer-Aided Design, vol. 26, no. 6, pp. 435-451, 1994.

- Hart-Smith, D. J., “Interface Control – The Secret to Making DFMA Succeed,” Society of Automotive Engineers, 1997.

- Hu, J., “Stream-of-Variation Theory for Automotive Body Assembly,” Annals of CIRP, vol. 46 no. 1, 1997.

- Huang, T., Whitehouse, D. J., and Chetwynd, D. G., “A Unified Error Model for Tolerance Design, Assembly and Error Compensation of 3-DOF Parallel Kinematic Machines with Parallelogram Struts,” Annals of the CIRP, vol. 52, no. 1, pp. 297-301, 2002.

- Huebner, K. H., Dewhirst, D., Smith, D,. and Byrom, T., The Finite Element Method for Engineers 4th edition, John Wiley and Sons Inc., New York, 2001.

- Hong, Y. S., and Change, T. C., A Comprehensive Review of Tolerancing Research, International Journal of Product Research, vol. 40, no. 11, pp. 2425-2459, 2002.

- Howell, L. Compliant Mechanisms, John Wiley and Sons Inc., New York, NY, 2001.

- Hunt, K. H., Kinematic Geometry of Mechanisms, Oxford University Press, New York, NY, 1990.

- Johannesson H., and Söderberg R., “Structure and Matrix Models for Tolerance Analysis from Configuration to Detail Design,” Research in Engineering Design, vol. 12, no. 2, pp. 112-125, 2000.

- Kern, D. and Thornton, A., “Modeling Variation Propagation through a Manufacturing Process,” Proceedings of the 2003 ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference, vol. 3, pp. 157-165, 2003.

- Klein, C. J., "Generation and Evaluation of Assembly Sequence Alternatives," S.M. Thesis, Massachusetts Institute of Technology, Department of Mechanical Engineering, Cambridge, MA, 1986.

- Konkar R., and Cutkosky M., "Incremental Kinematic Analysis of Mechanisms," ASME Journal of Mechanical Design, vol. 117, pp. 589-596, December 1995.

- Konkar, R., "Incremental Kinematic Analysis and Symbolic Synthesis of Mechanisms," Ph.D. Dissertation, Stanford University, Stanford, CA, 1993.

- Kriegel, J. M., "Exact Constraint Design", Mechanical Engineering, pp. 88-90, May 1995.

- Laperriére, L., Ghie, W., and Desrochers, A., "Statistical and Deterministic Tolerance Analysis and Synthesis Using a Unified Jacobian-Torsor Model," Annals of the CIRP, vol. 51, no. 1, pp. 417-420, 2002.

- Laperriére, L., and Lafond, P., "Modeling Tolerances and Dispersions of Mechanical Assemblies Using Virtual Joints," Proceedings of the 25th ASME Design Automation Conferences, 1999.

- Lee, K., and Gossard, D., "A Hierarchical Data Structure for Representing Assemblies: part 1," Computer-Aided Design, vol. 17, no. 1, pp. 15-19, 1985.

- Lee, J., Long, Y., and Hu, S. J., "Robustness Evaluation for Compliant Assembly Systems," ASME Design for Manufacturing Conference, Baltimore, MD, 2000.

- Liu, S. C., and Hu, S. J., "Variation Simulation for Deformable Sheet Metal Assemblies Using Finite Element Methods," ASME Journal of Engineering for Industry, vol. 129, pp. 368-374, 1997.

- Mantripragada, R., and Whitney, D. E., "The Datum Flow Chain: A Systematic Approach to Assembly Design and Modeling," Research in Engineering Design, vol. 10, pp. 150-165, 1998.

- Mantripragada, R., and Whitney, D. E., "Modeling and Controlling Variation Propagation in Mechanical Assemblies Using State Transition Models," IEEE Transactions On Robotics and Automation, vol. 15, no. 1, 1999.

- Marguet, B., "Tolerancing Model by TTRS: A Survey," Internal Report Submitted to Center for Technology, Policy, and Industrial Development at MIT, Cambridge, MA, 1998.

- Maxwell, L. M., and Reed, M., The Theory of Graphs: A Basis for Network Theory, Pergamon Press, New York, NY, 1971.

- Melchers, R.E., Structural Reliability Analysis and Prediction, John Wiley & Sons, Chichester, England, 1999.

- Milner, J., Graves, S., and Whitney, D. E., "Using Simulated Annealing to Select Least-Cost Assembly Sequences," Proceeding of the IEEE Robotics and Automation Conference, 1994.

- Mullins, S. H., and Anderson, D. C., "Automatic Identification of Geometric Constraints in Mechanical Assemblies," Computer-Aided Design, vol. 30, no. 9, pp. 715-726, 1998.

- Ngoi, B. K. A., and Min, O. J., "Optimum Tolerance Allocation in Assembly," International Journal of Advanced Manufacturing Technology, vol. 15, pp. 660-665, 1999.

- Ohwovoriole, M. S., and Roth, B., "An Extension of Screw Theory," ASME Journal of Mechanical Design, vol. 103, pp. 725-735, October 1981.

- Pahl, G., and Beitz, W., Engineering Design: A Systematic Approach, Springer, New York, 1996.

- Parkinson, A., Sorensen, C., and Pourhassan, N., "A General Approach for Robust Optimal Design," ASME Journal of Mechanical Design, 115(1), pp.74-80, 1993.

- Parameshaven, S., and Kjerengtroen, L., "Determination of Failure Probabilities and Sensitivity Factors Based on First Order Reliability Method," 1994 Advances in Design Automation, vol. 1., pp. 495-501, 1994.

- Parkinson, D. B., "Application of a Robust Design Method to Tolerancing," Journal of Mechanical Design, vol. 122, no, 2, pp. 149-154, 2000

- Patterson, T., and Lipkin, H., "Structure of Robot Compliance," Journal of Mechanical Design. vol. 115, pp. 576-588, 1993a.

- Patterson, T., and Lipkin, H., "A Classification of Robot Compliance," Journal of Mechanical Design. vol. 115, pp. 581-584, 1993b.

- Paul, R. P., Robot Manipulators, MIT Press, Cambridge, 1981.

- Paz, M., and Leigh, W., Integrated Matrix Analysis of Structures, Kluwer Academic Publishers, Boston, MA, 2001.

- Phillips, J., Freedom in Machinery Volume 1: Introducing Screw Theory, Cambridge University Press, Cambridge, England, 1984.

- Phillips, J., Freedom in Machinery Volume 2: Screw Theory Exemplified, Cambridge University Press, Cambridge, England, 1990.

- Ramana, K., and Rao, P., "Automated Manufacturability Evaluation of Sheet Metal Components: A Review," Proceedings of the 2002 ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference, vol. 3, pp. 41-56, 2002.

- Requicha, A. A. G., "Toward a Theory of Geometric Tolerance," International Journal of Robotic Research, vol. 2, no. 4, pp. 45-60, 1983.

- Rivest, L., Fortin, C. and Morel, C. "Tolerancing a Solid Model with a Kinematic Formulation," Computer-Aided Design, vol. 26, no, 6, pp. 465-476, 1994.

- Rosenblatt, M., "Remarks on a Multivariate Transformation," Annals of Mathematical Statistics, vol. 23, pp. 470-472, 1952.

- Roth, B., "Screws, Motors, and Wrenches That Cannot Be Bought in a Hardware Store," In: Robotics Research, The First International Symposium, pp. 679-693, 1983.

- Sacks, E., and Joskowicz, L., "Parametric Kinematic Tolerance Analysis of General Planar Systems," Computer-Aided Design, vol. 30, no. 9, pp. 707-714, 1998.

- Seshu, S., and Reed, M., Linear Graphs and Electrical Networks, Addison-Wesley, Reading, MA, 1961.

- Shiu, B., Ceglarek, D., and Shi, J., "Flexible Beam-Based Modeling of Sheet Metal assembly for Dimension Control," Transactions of NAMRI, vol. 25, pp. 49-54, 1997.

- Shukla, G., "Augmenting Datum Flow Chain Method to Support the Top-Down Design Process for Mechanical Assemblies", S.M. Thesis, Massachusetts Institute of Technology, Department of Mechanical Engineering, Cambridge, MA, 2001.

- Shukla, G., and Whitney, D. E., "Systematic Evaluation of Constraint Properties of Datum Flow Chain," IEEE ISATP, Fukuoka, Japan, 2001.

- Slocum, A. H., Precision Machine Design, Prentice-Hall, New York, NY, 1991.

- Smith, D., "Constraint Analysis of Assemblies Using Screw Theory and Tolerance Sensitivities," S.M. Thesis, Brigham Young University, Provo, UT, 2001.

- Strang, G., Introduction to Applied Mathematics, Wellesley-Cambridge Press, Cambridge, MA, 1986.

- Söderberg, R., and Johannesson, H., "Tolerance Chain Detection by Geometrical Constraint Based Coupling Analysis," Journal of Engineering Design, vol. 10, November, pp. 5-24, 1999.

- Sodhi, R., and Turner, J. U., "Relative Positioning of Variational Part Models for Design Analysis," Computer-Aided Design, vol. 26, no. 5, pp. 366-378, 1994.

- Srikanth, S., and Turner, J., "Towards a Unified Representation of Mechanical Assemblies," Engineering with Computers, vol. 6, pp. 103-112, 1990.

- Suh, N. P., The Principles of Design, Oxford University Press, New York, NY, 1990.

- Sweder, T. A., and Pollack, J., "Full Vehicle Variability Modeling," SAE Paper Reprint #942334, SAE Inc., 1994.

- Taguchi, G., Taguchi Methods: Research and Development, ASI Press, Dearborn, MI, 1992.

- Taguchi, G., Taguchi On Robust Technology Development: Bringing Quality Engineering Upstream, ASME Press, New York, NY, 1993.

- Terry, A. M., "Improving Product Manufacturability Through the Integrated Use of Statistics," L.F.M. Thesis, Massachusetts Institute of Technology, Department of Mechanical Engineering and Sloan School of Management, Cambridge, MA, 2000.

- Thornton, A. C., "A Mathematical Framework for the Key Characteristic Process," Research in Engineering Design, vol. 11, pp. 145-157, 1999.

- Tischler, C. R., and Samuel, A. E., "Prediction of the Slop in General Spatial Linkages," The International Journal of Robotics Research, vol. 18, no. 8, pp. 845-858, 1999.

- Ulrich, K. T., and Eppinger, S. D., Product Design and Development, McGraw-Hill Inc., New York, NY, 1995.

- Varghese, P., Braswell, R. N., Wang, B., and Zhang, C., "Statistical Tolerance Analysis Using FRPDF and Numerical Convolution," Computer-Aided Design, vol. 28, no. 9, pp. 723-731, 1996.

- VSA, *VSA-3D Release 12.5 User Manual,* Variation System Analysis, Inc., 300 Maple Park Boulevard, St. Clair Shores, MI 48081, 1998.

- Villarreal, A., and Asada, H., "A Geometric Representation of Distributed Compliance for the Assembly of Flexible Parts," Proceedings of the 1991 IEEE International Conference on Robotics and Automation, Sacramento, CA, April 1991.

- Waldron, K. J., "The Constraint Analysis of Mechanisms," Journal of Mechanisms, vol. 1, pp. 101-114, 1966.

- Whitney, D. E., Mechanical Assemblies: Their Design, Manufacture, and Role in Product Development, Oxford University Press, New York, NY, 2004.

- Whitney, D. E., Gilbert, O., and Jastrzebski, M., "Representation of Geometric Variations Using Matrix Transforms for Statistical Tolerance Analysis in Assemblies," Research in Engineering Design, vol. 6, pp. 191-210, 1994.

- Whitney, D. E., Mantripragada, R., Adams, J. D., and Cunningham, T., "Use of Screw Theory to Detect Multiple Conflicting Key Characteristics," ASME Design Engineering Technical Conferences, Las Vegas, Sep. 1999.

- Whitney, D. E., Mantripragada, R., Adams, J. D., and Rhee, S. J., "Designing Assemblies," Research in Engineering Design, vol. 11, pp. 229-253, 1999.

- Wirtz, A., "Vectorial Tolerancing: a Basic Element for Quality Control," 2nd CIRP Seminar on Computer Aided Tolerancing, Penn State, PA, pp.77-84, 1991.

- Yau, H., "Generalization and Evaluation of Vectorial Tolerances," International Journal of Production Research, vol. 35, no. 6, pp. 1763-1783, 1997.

APPENDIX A: SUPPORTING MATLAB® CODE

This appendix contains source code for the following Matlab® executables:

- **input_file.m**
- **modeler.m**
  - **energyloop.m**
  - **combiner.m**
- **constraintdetector.m**
- **manufacturability.m**
  - **mppfinder.m**
- **dimoptimizer.m**
  - **dimobjective.m**
  - **dimconstraint.m**

Matlab® Release 13, including the Statistics and Optimization Toolboxes, is required to run these programs. The files listed above should be placed in a single directory that is linked to Matlab® using the "**Set Path…**" command in the "**File**" drop down menu. The easiest way to execute the programs is to create a separate input file for each assembly or configuration (see **input_file.m** for an example). After all the matrices have been entered into the input file and the file has been saved in the linked directory, the inputs can be stored in memory by typing the name of the file on the command like (without the **.m** extension). The programs can then be executed by typing the name of the program on the command line followed by the list of required input matrices in parentheses.

input_file.m

```
%Filename: input_file.m
%Capability: A sample input file containing the inputs for the fingernail clipper
%Author: J. Michael Gray, MIT Graduate Student, Mech. Eng. Dept.
%Date completed: May 1st, 2004
%Revisions:

%Dimension-related data:
%nominal values
Dn=[0;0;0;1.36;1.36;-40.34;2.81;0;-.185;-40.24;-2.81;0;-35.18;8.41;-2.96];
%directions
Dd=[4;5;3;6;6;5;6;3;1;5;6;3;5;6;5];
%upper bounds
Dul=[.05;.05;.001;2.5;2.5;-40;3.5;.1;-.1;-40;-2.5;.1;-27;10;-2];
%lower bounds
Dll=[-.05;-.05;-.001;.5;.5;-50;2.5;-.1;-.2;-50;-3.5;-.1;-37;6;-4];
%distribution types
Dt=strvcat('Normal', 'Uniform', 'Uniform', 'Normal', 'Normal','Normal', 'Uniform');
Dt=strvcat(Dt,'Normal', 'Normal', 'Normal', 'Uniform', 'Normal', 'Normal', 'Normal', 'Normal');

%distribution parameters
%These distribution parameters include mean shift
Dp=[0 .03 NaN; -.028 .028 NaN; -.0015 .0023 NaN;1.36 .01 NaN;1.36 .006 NaN];
Dp=[-40.34 .020 NaN; Dp;2.75 2.91 NaN ;0 .0035 NaN;-.185 .001 NaN;-40.24 .028 NaN];
Dp=[ -2.89 -2.73 NaN; Dp;.003 .0028 NaN;-35.18 .05 NaN;8.41 .04 NaN;-2.96 .04 NaN];

%These distribution parameters do not include mean shift are optimized
%Dp=[0 .03 NaN; -.028 .028 NaN; -.0019 .0019 NaN;1.36 .01 NaN;1.36 .006 NaN];
%Dp=[-40.34 .020 NaN;2.42 2.58 NaN ;0 .0035 NaN;-.185 .001 NaN;-40.24 .024 NaN];
%Dp=[ -2.89 -2.73 NaN;Dp;0 .0028 NaN;-35.18 .05 NaN;8.41 .04 NaN;-2.96 .04 NaN];

%dimension chain matrix
CHd=[-33 -32 -31 -30 -29 -28 zeros(1,11);-33 -32 -31 -30 -29 -28 1 zeros(1,10)];
CHd=[CHd;-33 -32 -31 -30 -29 -28 1 2 zeros(1,9);zeros(1,17);4 34 35 36 16 17 18 zeros(1,10)];
CHd=[CHd;zeros(1,17);6 zeros(1,16);6 7 zeros(1,15)];
CHd=[CHd; 4 34 35 36 16 17 18 5 53 54 55 56 57 27 zeros(1,3)];
CHd=[CHd; 4 34 35 36 16 17 18 5 53 54 55 56 57 27 9 zeros(1,2)];
CHd=[CHd;34 35 36 16 17 18 5 53 54 55 56 57 27 9 10 zeros(1,2)];
CHd=[CHd; 34 35 36 16 17 18 5 53 54 55 56 57 27 9 10 11 zeros(1,1)];
CHd=[CHd;zeros(1,17);13 40 41 42 43 44 22 zeros(1,10)];
CHd=[CHd;13 40 41 42 43 44 22 14 45 46 47 48 49 23 zeros(1,3)];

%DOF-related data:
%nominal values
Fn=[0;0;0;2.72;0;0;0;1.4936;0;0;35.514];
%directions
Fd=[1 0 0 Inf;0 1 0 Inf;0 0 1 0;0 0 1 Inf;1 0 0 0;0 1 0 0;0 0 1 0;1 0 0 0;0 0 1 0;1 0 0 Inf;0 1 0 Inf];

%upper bounds
Ful=[5;5;dtr(90);10;dtr(90);dtr(90);dtr(90);dtr(180);dtr(90);10;50];
%lower bounds
Fll=[-5;-5;-dtr(90);-10;-dtr(90);-dtr(90);dtr(-90);dtr(-150);-dtr(90);-10;-50];
```

```
%DOS-related data
%nominal values
Sn=[.1078];
%directions
Sd=[1 0 0 0];
%upper bounds
Sul=[dtr(90)];
%lower bounds
Sll=[-dtr(90)];

%nominal configuration-level stiffness matrix
SK=[25000];

%DOC-related data
%nominal values
Cn=[zeros(24,1); -pi/2; zeros(5,1)];
%directions
Cd=[1 0 0 Inf;0 1 0 Inf;0 0 1 Inf;1 0 0 0;0 1 0 0;0 0 1 0;0 0 1 Inf;1 0 0 0;0 1 0 0;1 0 0 Inf];
Cd=[Cd;0 1 0 Inf;0 0 1 0;1 0 0 Inf;0 1 0 Inf;0 0 1 Inf;1 0 0 0;0 1 0 0;1 0 0 Inf;0 1 0 Inf];
Cd=[Cd;0 0 1 Inf;0 1 0 0;0 0 1 0;0 0 1 Inf;0 1 0 0;1 0 0 0 ;1 0 0 Inf;0 1 0 Inf];
Cd=[Cd;0 0 1 Inf;0 1 0 0;0 0 1 0];

%mate chain matrix
CHm=[4 zeros(1,19);4 16 zeros(1,18); 4 16 17 zeros(1,17)];
CHm=[CHm; 4 16 17 18 34 35 36 5 -21 -20 -19 zeros(1,9)];
CHm=[CHm;4 16 17 18 34 35 36 5 -21 -20 zeros(1,10); 4 16 17 18 34 35 36 5 -21 zeros(1,11)];
CHm=[CHm;13 40 41 42 43 44 zeros(1,14);13 40 41 42 43 44 22 14 45 46 47 48 49 zeros(1,7)];
CHm=[CHm;13 40 41 42 43 44 22 14 45 46 47 48 49 23 15 50 51 52 zeros(1,2)];
CHm=[CHm;13 40 41 42 43 44 22 14 45 46 47 48 49 23 15 50 51 52 24 zeros(1,1)];
CHm=[CHm;13 40 41 42 43 44 22 14 45 46 47 48 49 23 15 50 51 52 24 25];
CHm=[CHm;4 34 35 37 16 17 18 5 53 54 55 56 57 zeros(1,7)];
CHm=[CHm;-33 -32 -31 -30 -29 -28 zeros(1,14)];
CHm=[CHm;-33 -32 -31 -30 -29 zeros(1,15);-33 -32 -31 -30 zeros(1,16)];
CHm=[CHm;-33 -32 -31 zeros(1,17);-33 -32 zeros(1,18);-33 zeros(1,19);4 zeros(1,19)];
CHm=[CHm;4 34 zeros(1,18)];
CHm=[CHm; 4 34 35 zeros(1,17);-33 -32 -31 -30 -29 -28 1 2 3 zeros(1,11)];
CHm=[CHm;-33 -32 -31 -30 -29 -28 1 2 3 37 zeros(1,10)];
CHm=[CHm; );-33 -32 -31 -30 -29 -28 1 2 37 38 zeros(1,10)];
CHm=[CHm;13 zeros(1,19); 13 40 zeros(1,18); 13 40 41 zeros(1,17); 13 40 41 42 zeros(1,16)];
CHm=[CHm;13 40 41 42 43 zeros(1,15);  13 40 41 42 43 44 22 14 zeros(1,12)];
CHm=[CHm;13 40 41 42 43 44 22 14 45 zeros(1,11)];
CHm=[CHm;13 40 41 42 43 44 22 14 45 46 zeros(1,10)];
CHm=[CHm;13 40 41 42 43 44 22 14 45 46 47 zeros(1,9)];
CHm=[CHm;13 40 41 42 43 44 22 14 45 46 47 48 zeros(1,8)];
CHm=[CHm;13 40 41 42 43 44 22 14 45 46 47 48 49 23 15 zeros(1,5)];
CHm=[CHm;13 40 41 42 43 44 22 14 45 46 47 48 49 23 15 50 zeros(1,4)];
CHm=[CHm;13 40 41 42 43 44 22 14 45 46 47 48 49 23 15 50 51 zeros(1,3)];
CHm=[CHm;4 34 35 36 16 17 18 5 zeros(1,12); 4 34 35 36 16 17 18 5 53 zeros(1,11)];
CHm=[CHm;4 34 35 36 16 17 18 5 53 54 zeros(1,10)];
CHm=[CHm;4 34 35 36 16 17 18 5 53 54 55 zeros(1,9)];
CHm=[CHm;4 34 35 36 16 17 18 5 53 54 55 56 zeros(1,8)];
```

```
%open-loop chain matrix
CHo=[4 34 35 36 16 17 18 5 53 54 55 56 57 27 9 10 11 12 -12 -9 -27 -57 -56 -18 -36 -35];
CHo=[CHo; 6 7 8 zeros(1,23)];

%closed-loop chain matrix
CHc=[28 29 30 31 32 33 4 34 35 36 16 17 18 5 -39 -38 -37 -21 -20 -19 -3 -2 -1 zeros(1,13)];
CHca=[13 40 41 42 43 44 22 14 45 46 47 48 49 23 15 50 51 52 24 25 26];
CHcb=[-9 -27 -57 -56 -55 -54 -53 -5 -18 -17 -16 -36 -35 -34 -4];
CHc=[CHc;Chca Chcb];

%KC related matrices:
%KC chain matrix
CHk=[-8 -7 -6 4 34 35 36 16 17 18 5 53 54 55 56 57 27 9 10 11 12];
CHk=[CHk;CHk;CHk;CHk;CHk;CHk;CHk];
%directions
KCd=[4;4;5;5;3;3;6];
%specification offsets
KCl=[.20;-.20;.15;-.15;.02;-.02;0];
%KC types
KCt=[1;1;1;1;1;1;2];

%Loop mate matrix
LM=[1 1 1 -1 -1 -1 zeros(1,18);-1 -1 -1 0 0 0 1 1 1 1 1 -1 zeros(1,12)];
LM=[LM;;1 1 1 zeros(1,8) 1 -ones(1,6) zeros(1,6);zeros(1,18) -ones(1,6)];

%Loop dimension matrix
LD=[-1 -1 -1 1 1 zeros(1,10);0 0 0 -1 -1 0 0 0 -1 0 0 0 1 1 1;0 0 0 1 1 0 0 0 1 1 1 1 0 0 0;0 0 0 0 0
1 1 1 0 0 0 0 0 0 0];

%Part mate matrix
PM=[-1 -1 -1 -1 -1 -1 0 0 0 -1 -1 -1 zeros(1,19)];
PM=[PM;0 0 0 0 0 0 1 1 1 1 1 1 zeros(1,13) -1 -1 -1 -1 -1 -1];
PM=[PM;zeros(1,22) 1 1 1 1 1 1 1 1 1;zeros(1,12) 1 1 1 1 1 -1 -1 -1 -1 -1 0 0 0 0 0 0 0 0 0];
PM=[PM;zeros(1,17) 1 1 1 1 1 -1 -1 -1 0 0 0 0 0 0];

%KC permutation matrix
KC=[zeros(1,12) 0 0 0 1 0 0 0 0 0 -1 0 0;zeros(1,12) 0 0 0 -1 0 0 0 0 0 1 0 0];
KC=[KC; zeros(1,12) 0 0 0 0 1 0 0 0 0 0 -1 0];
KC=[KC;zeros(1,12) 0 0 0 0 -1 0 0 0 0 0 1 0; zeros(1,12) 0 0 1 0 0 0 0 0 -1 0 0 0];
KC=[KC;zeros(1,12) 0 0 -1 0 0 0 0 0 1 0 0 0; zeros(1,12) 0 0 0 0 0 -1 0 0 0 0 0 1];
KC=[KC zeros(7,31)];

%Target Cpk
KCT=[1.33];
%KC Optimization Weighting Factors
Wa=[1/7;1/7;1/7;1/7;1/7;1/7;1/7];
```

modeler.m

```
%Filename: modeler.m
%Capability: Creates matrices representing the nominal and varied states of
%a Type-I assembly with a single configuration.  Also solves for the
%nominal values of the KCs and their specification limits.
%Author: J. Michael Gray, MIT Graduate Student, Mech. Eng. Dept.
%Date completed: May 1st, 2004
%Revisions:
function [B,U,TL,WP,g,beta_energy,Fnstar,Snstar,KCn,KCL]=modeler(Dn, Dd, CHd, Fn, Fd, Ful,
Fll,Sn, Sd, Sul, Sll, SK, Cn, Cd, CHm, CHc, CHo, KCd, KCt, KCl,CHk, LM, PM, LD)

%Getting the sizes of the matrices
[dn1,dn2]=size(Dn); [dd1,dd2]=size(Dd); [chd1,chd2]=size(CHd); [fn1,fn2]=size(Fn);
[fd1,fd2]=size(Fd); [ful1,ful2]=size(Ful); [fll1,fll2]=size(Fll); [sn1,sn2]=size(Sn);
[sd1,sd2]=size(Sd); [sul1,sul2]=size(Sul); [sll1,sll2]=size(Sll); [sk1,sk2]=size(SK);
[cn1,cn2]=size(Cn); [cd1,cd2]=size(Cd); [chm1,chm2]=size(CHm); [chc1,chc2]=size(CHc);
[cho1,cho2]=size(CHo); [kcd1,kcd2]=size(KCd); [kct1,kct2]=size(KCt); [kcl1,kcl2]=size(KCl);
[chk1,chk2]=size(CHk); [lm1,lm2]=size(LM); [pm1,pm2]=size(PM); [ld1,ld2]=size(LD);

%Checking to make sure the matrices are the right size
if dn1~=dd1 | dn1~=chd1 | dn2~=1 | dd2~=1
   disp('');
   disp('Dimension related inputs Dn, Dd, and/or CHd is/are the wrong size.');
   disp('End of program.');
   disp('');
   return
end
if fn1~=fd1 | fn1~=ful1 | fn1~=fll1 | fn2~=1 | fd2~=4 | ful2~=1 | fll2~=1
   disp('');
   disp('DOF related inputs Fn, Fd, Ful, and/or Fll is/are the wrong size.');
   disp('End of program.');
   disp('');
   return
end
if sn1~=sd1 | sn1~=sul1 | sn1~=sll1 | sn1~=sk1 | sn1~=sk2 | sn2~=1 | sd2~=4 | sul2~=1 | sll2~=1
   disp('');
   disp('DOS related inputs Sn, Sd, Sul, Sul, and/or SK is/are the wrong size.');
   disp('End of program.');
   disp('');
   return
end
if cn1~=cd1 | cn2~=1 | cd2~=4
   disp('');
   disp('DOC related inputs Cn and/or Cd is/are the wrong size.');
   disp('End of program.');
   disp('');
   return
end
if kcd1~=chk1 | kcd1~=kct1 | kcd1~=kcl1 | kcd2~=1 | kct2~=1 | kcl2~=1
   disp('');
   disp('KC related inputs KCd, KCt, KC1 and/or CHk is/are the wrong size.');
   disp('End of program.');
```

```
    disp('');
    return
end

%storing counting variables: dt=dimensions, ft=DOF, st=DOS, ct=DOC, kct=KCs
dt=dn1;
ft=fn1;
st=sn1;
ct=cn1;
kct=kcd1;

%More checking to make sure the matrices are the right size
if chm1~=(ft+st+ct)
    disp('');
    disp('Mate related input CHm has the wrong number of rows.');
    disp('End of program.');
    disp('');
    return
end
if st==0 & ct==0
    p=1;
    m=1;
    if lm1~=(chc1+cho1) | lm2~=(ft+st+6*cho1) | pm2~=0 | pm1~=0
        disp('');
        disp('Loop and part related inputs CHc, Cho, LM, and/or PM is/are the wrong size.');
        disp('End of program.');
        disp('');
        return
    end
else
    m=chm1/6;
    p=pm1+1;
    if chc1~=(m-pm1) | lm1~=(chc1+cho1) | lm2~=(ft+st+6*cho1) | pm2~=(ct+st)
        disp('');
        disp('Loop and part related inputs CHc, Cho, LM, and/or PM is/are the wrong size.');
        disp('End of program.');
        disp('');
        return
    end
end

%storing counting variables: lamba=closed-loops, g=open-loops
lambda=chc1;
g=cho1;

%More checking to make sure the matrices are the right size
if ld1~=(lambda+g)  | ld2~=dt
    disp('');
    disp('Dimension related input LD is the wrong size.');
    disp('End of program.');
    disp('');
    return
end
```

```
%Solving for the closed-loops and energy minimization
if lambda>0
   FSn=[Fn;Sn];
   FSul=[Ful;Sul];
   FSll=[Fll;Sll];
   FUN=@energyloop;
   OPTIONS1 =
Optimset('LargeScale','off','MaxFunEvals',125*dt,'MaxIter',125*dt,'Display','off','TolFun',1e-
9*max(abs(Dn)),'TolCon',1e-9*max(abs(Dn)));

[FSn,FVAL,EXITFLAG]=fmincon(FUN,FSn,[],[],[],[],FSll,FSul,[],OPTIONS1,Dn,Dd,Fd,Sd,SK,Cn,C
d,CHc);
   if EXITFLAG==0
      disp('');
      disp('Maximum number of iterations reached');
      disp('');
   end
   if EXITFLAG<0
      disp('');
      disp('Could not solve for one or more closed loops.');
      disp('End of program.');
      disp('');
      return
   end
   %Storing new nominal values for DOF and DOS
   Fnstar=FSn(1:ft,1);
   Snstar=FSn((ft+1):(ft+st),1);
else
   %Storing new nominal values for DOF and DOS
   Fnstar=Fn;
   Snstar=Sn;
end

%Storing stored energy (beta_energy)
beta_energy=.5*Snstar'*SK*Snstar;

%Solving for HTMS representation of dimensions, DOF, DOS, and DOC
HTM=zeros(4*(dt+ft+st+ct),4);
HTMn=[Dn;Fnstar;Snstar;Cn];
HTMd=[Dd zeros(dt,3);Fd;Sd;Cd];
for i=1:(dt+ft+st+ct)
   nominal=HTMn(i,1);
   if i<=dt
      direction=HTMd(i,1);
      if direction==1
         htm=rotx(nominal);
      elseif direction==2
         htm=roty(nominal);
      elseif direction==3
         htm=rotz(nominal);
      elseif direction==4
         htm=trans(nominal,0,0);
      elseif direction==5
         htm=trans(0,nominal,0);
      elseif direction==6
         htm=trans(0,0,nominal);
      else
         disp('Dimension direction must be integer between 1 and 6');
         return
      end
      HTM((4*i-3):4*i,1:4)=htm;
   elseif i>dt & i<=(dt+ft+st+ct)
      pitch=HTMd(i,4);
      xdir=HTMd(i,1);
      ydir=HTMd(i,2);
      zdir=HTMd(i,3);
      if pitch==Inf
         htm=eye(4);
         htm(1,4)=xdir*nominal;
         htm(2,4)=ydir*nominal;
         htm(3,4)=zdir*nominal;
      elseif abs(zdir)==1
         htm=rotz(nominal);
      elseif (xdir^2+ydir^2+zdir^2)^.5>.99 & (xdir^2+ydir^2+zdir^2)^.5<1.01
         htm=eye(4);
         w=(1-zdir^2)^.5;
         htm(1,1)=xdir^2+(xdir^2*zdir^2+ydir^2)*cos(nominal)/w^2;
         htm(2,1)=xdir*ydir+zdir*sin(nominal)-xdir*ydir*cos(nominal);
         htm(3,1)=xdir*zdir*(1-cos(nominal))-ydir*sin(nominal);
         htm(1,2)=xdir*ydir-zdir*sin(nominal)-xdir*ydir*cos(nominal);
         htm(2,2)=ydir^2+(ydir^2*zdir^2+xdir^2)*cos(nominal)/w^2;
         htm(3,2)=ydir*zdir*(1-cos(nominal))+xdir*sin(nominal);
         htm(1,3)=xdir*zdir*(1-cos(nominal))+ydir*sin(nominal);
         htm(2,3)=ydir*zdir*(1-cos(nominal))-xdir*sin(nominal);
         htm(3,3)=zdir^2+cos(nominal)*w^2;
         htm(1,4)=nominal*pitch*xdir;
         htm(2,4)=nominal*pitch*ydir;
         htm(3,4)=nominal*pitch*zdir;
      else
         disp('direction vector for DOF,DOS, or DOC must be unit length');
         return
      end
      HTM((4*i-3):4*i,1:4)=htm;
   else
      disp('Error in creating HTM matrices for Dimensions, DOF, DOS, and DOC.');
      return
   end
end

%Solving for twist representations of DOF & DOS
TM=zeros(ft+st+6*g,6);
for i=1:ft
   f=eye(4);
   for j=1:chm2
      if CHm(i,j)==0
```

```
         break
      else
         if sign(CHm(i,j))==-1
            f=f*inv(HTM((4*abs(CHm(i,j))-3):4*abs(CHm(i,j)),1:4));
         else
            f=f*HTM((4*abs(CHm(i,j))-3):4*abs(CHm(i,j)),1:4);
         end
      end
   end
   w=Fd(i,1:3)*f(1:3,1:3)';
   if Fd(i,4)==Inf
      TM(i,1:6)=[zeros(1,3) w];
   else
      u=cross(f(1:3,4)',w);
      v1=u(1,1)+Fd(i,4)*w(1,1);
      v2=u(1,2)+Fd(i,4)*w(1,2);
      v3=u(1,3)+Fd(i,4)*w(1,3);
      TM(i,1:6)=[w v1 v2 v3];
   end
end
for i=1:st
   s=eye(4);
   for j=1:chm2
      if CHm(i+ft,j)==0
         break
      else
         if sign(CHm(i+ft,j))==-1
            s=s*inv(HTM((4*abs(CHm(i+ft,j))-3):4*abs(CHm(i+ft,j)),1:4));
         else
            s=s*HTM((4*abs(CHm(i+ft,j))-3):4*abs(CHm(i+ft,j)),1:4);
         end
      end
   end
   w=Sd(i,1:3)*s(1:3,1:3)';
   if Sd(i,4)==Inf
      TM(i+ft,1:6)=[zeros(1,3) w];
   else
      u=cross(s(1:3,4)',w);
      v1=u(1,1)+Sd(i,4)*w(1,1);
      v2=u(1,2)+Sd(i,4)*w(1,2);
      v3=u(1,3)+Sd(i,4)*w(1,3);
      TM(i+ft,1:6)=[w v1 v2 v3];
   end
end

%Solving for twist representations of DOV for open-loops
count=0;
for i=1:g
   G=eye(4);
   for j=1:cho2
      if CHo(i,j)==0
         break
      else
```

```
         if sign(CHo(i,j))==-1
            G=G*inv(HTM((4*abs(CHo(i,j))-3):4*abs(CHo(i,j)),1:4));
         else
            G=G*HTM((4*abs(CHo(i,j))-3):4*abs(CHo(i,j)),1:4);
         end
      end
   end
   Q1=cross(G(1:3,1)',G(1:3,4)');
   Q2=cross(G(1:3,2)',G(1:3,4)');
   Q3=cross(G(1:3,3)',G(1:3,4)');
   V=[G(1:3,1:3)' [Q1;Q2;Q3]';zeros(3,3) G(1:3,1:3)'];
   TM((i+ft+st+count):(i+ft+st+5+count),1:6)=V;
   count=count+5;
end

%Solving for wrench representations of DOC & DOS
WM=zeros(ct+st,6);
for i=1:ct
   c=eye(4);
   for j=1:chm2
      if CHm(i+ft+st,j)==0
         break
      else
         if sign(CHm(i+ft+st,j))==-1
            c=c*inv(HTM((4*abs(CHm(i+ft+st,j))-3):4*abs(CHm(i+ft+st,j)),1:4));
         else
            c=c*HTM((4*abs(CHm(i+ft+st,j))-3):4*abs(CHm(i+ft+st,j)),1:4);
         end
      end
   end
   v=Cd(i,1:3)*c(1:3,1:3)';
   if Cd(i,4)==0
      WM(i,1:6)=[v zeros(1,3) ];
   elseif Cd(i,4)==Inf
      w=cross(c(1:3,4)',v);
      WM(i,1:6)=[w v];
   else
      u=cross(c(1:3,4)',v);
      w1=u(1,1)+v(1,1)/Cd(i,4);
      w2=u(1,2)+v(1,2)/Cd(i,4);
      w3=u(1,3)+v(1,3)/Cd(i,4);
      WM(i,1:6)=[w1 w2 w3 v];
   end
end
for i=1:st
   s=eye(4);
   for j=1:chm2
      if CHm(i+ft,j)==0
         break
      else
         if sign(CHm(i+ft,j))==-1
            s=s*inv(HTM((4*abs(CHm(i+ft,j))-3):4*abs(CHm(i+ft,j)),1:4));
         else
```

```
            s=s*HTM((4*abs(CHm(i+ft,j))-3):4*abs(CHm(i+ft,j)),1:4);
         end
      end
   end
   v=Sd(i,1:3)*s(1:3,1:3)';
   if Sd(i,4)==0
      WM(i+ct,1:6)=[v zeros(1,3) ];
   elseif Sd(i,4)==Inf
      w=cross(s(1:3,4)',v);
      WM(i+ct,1:6)=[w v];
   else
      u=cross(s(1:3,4)',v);
      w1=u(1,1)+v(1,1)/Sd(i,4);
      w2=u(1,2)+v(1,2)/Sd(i,4);
      w3=u(1,3)+v(1,3)/Sd(i,4);
      WM(i+ct,1:6)=[w1 w2 w3 v];
   end
end

%Solving for twist representations of Dimensions
UD=zeros(dt,6);
for i=1:dt
   d=eye(4);
   for j=1:chd2
      if CHd(i,j)==0
         break
      else
         if sign(CHd(i,j))==-1
            d=d*inv(HTM((4*abs(CHd(i,j))-3):4*abs(CHd(i,j)),1:4));
         else
            d=d*HTM((4*abs(CHd(i,j))-3):4*abs(CHd(i,j)),1:4);
         end
      end
   end
   if Dd(i,1)<=3
      q=cross(d(1:3,4)',d(1:3,Dd(i,1))');
      UD(i,1:6)=[d(1:3,Dd(i,1))' q];
   else
      UD(i,1:6)=[zeros(1,3) d(1:3,Dd(i,1)-3)'];
   end
end

%Combining all the matrices into nominal and varied model
TL=combiner(TM,LM);
WP=combiner(WM,PM);
B1=[TL zeros(6*(lambda+g),ct+st)];
B2=[zeros(st,ft) SK zeros(st,6*g+ct) -eye(st)];
B3=[zeros(6*(p-1),ft+st+6*g) WP];

%nominal model
B=[B1;B2;B3];
UL=combiner(UD,LD);
```

```
%varied model
U=[UL;zeros(6*(p-1)+st,dt)];

%Solving for KC nominal values and specification limits
KCn=zeros(kct,1);
KCL=zeros(kct,1);
for i=1:kct
    H=eye(4);
    for j=1:chk2
        if CHk(i,j)==0
            break
        else
            if sign(CHk(i,j))==-1
                H=H*inv(HTM((4*abs(CHk(i,j))-3):4*abs(CHk(i,j)),1:4));
            else
                H=H*HTM((4*abs(CHk(i,j))-3):4*abs(CHk(i,j)),1:4);
            end
        end
    end
    if KCd(i,1)==1
        KCn(i,1)=atan2(H(3,2),sign(H(1,1))*H(2,2));
    elseif KCd(i,1)==2
        KCn(i,1)=atan2(H(1,3),sign(H(2,2))*H(3,3));
    elseif KCd(i,1)==3
        KCn(i,1)=atan2(H(2,1),sign(H(3,3))*H(1,1));
    elseif KCd(i,1)==4
        KCn(i,1)=H(1,4);
    elseif KCd(i,1)==5
        KCn(i,1)=H(2,4);
    elseif KCd(i,1)==6
        KCn(i,1)=H(3,4);
    else
        disp('KC direction must be integer between 1 and 6');
        return
    end
    if KCt(i,1)==1
        KCL(i,1)=KCl(i,1)+KCn(i,1);
    elseif KCt(i,1)==2
        KCL(i,1)=KCL(i,1);
    else
        disp('KC type must be 1 (alignment) or 2 (clearance)');
        return
    end
end
```

energyloop.m

```
%Filename: energylooop.m
%Capability: Calculates stored energy plus loop error of a configuration.
%Author: J. Michael Gray, MIT Graduate Student, Mech. Eng. Dept.
%Date completed: May 1st, 2004
%Revisions:
function energyplusloop=energyloop(FSn,Dn,Dd,Fd,Sd,SK,Cn,Cd,CHc);
[st,z]=size(Sd);
[ft,z]=size(Fd);
if st==0
   %Stored energy is zero if there is no compliance
   energy=0;
else
   %Solving for stored energy
   energy=.5*FSn(ft+1:ft+st,1)'*SK*FSn(ft+1:ft+st,1);
end

%Solving for loop closure constraints
[ct,z]=size(Cd);
[dt,z]=size(Dd);
[lambda,chc2]=size(CHc);
Loop=zeros(4*lambda,4);
HTMn=[Dn;FSn;Cn];
HTMd=[Dd zeros(dt,3);Fd;Sd;Cd];
for i=1:lambda
   Loop((4*i-3):4*i,1:4)=eye(4);
   for j=1:chc2
      if CHc(i,j)==0
         break
      else
         %Finding HTMS for dimension in the loops
         nominal=HTMn(abs(CHc(i,j)),1);
         if abs(CHc(i,j))<=dt
            direction=HTMd(abs(CHc(i,j)),1);
            if direction==1
               htm=rotx(nominal);
            elseif direction==2
               htm=roty(nominal);
            elseif direction==3
               htm=rotz(nominal);
            elseif direction==4
               htm=trans(nominal,0,0);
            elseif direction==5
               htm=trans(0,nominal,0);
            elseif direction==6
               htm=trans(0,0,nominal);
            else
               disp('Dimension direction must be integer between 1 and 6');
               return
            end
            if sign(CHc(i,j))==-1 & direction<=3
               htm=htm';
            elseif sign(CHc(i,j))==-1
               htm(direction-3,4)=-htm(direction-3,4);
            else
            end

         %Finding HTMS for DOF, DOS, and DOC in the loops
         elseif abs(CHc(i,j))>dt & abs(CHc(i,j))<=(dt+ft+ct+st)
            pitch=HTMd(abs(CHc(i,j)),4);
            xdir=HTMd(abs(CHc(i,j)),1);
            ydir=HTMd(abs(CHc(i,j)),2);
            zdir=HTMd(abs(CHc(i,j)),3);
            if pitch==Inf
               htm=eye(4);
               htm(1,4)=xdir*nominal;
               htm(2,4)=ydir*nominal;
               htm(3,4)=zdir*nominal;
            elseif abs(zdir)==1
               htm=rotz(nominal);
            elseif (xdir^2+ydir^2+zdir^2)^.5>.99 & (xdir^2+ydir^2+zdir^2)^.5<1.01
               htm=eye(4);
               w=(1-zdir^2)^.5;
               htm(1,1)=xdir^2+(xdir^2*zdir^2+ydir^2)*cos(nominal)/w^2;
               htm(2,1)=xdir*ydir+zdir*sin(nominal)-xdir*ydir*cos(nominal);
               htm(3,1)=xdir*zdir*(1-cos(nominal))-ydir*sin(nominal);
               htm(1,2)=xdir*ydir-zdir*sin(nominal)-xdir*ydir*cos(nominal);
               htm(2,2)=ydir^2+(ydir^2*zdir^2+xdir^2)*cos(nominal)/w^2;
               htm(3,2)=ydir*zdir*(1-cos(nominal))+xdir*sin(nominal);
               htm(1,3)=xdir*zdir*(1-cos(nominal))+ydir*sin(nominal);
               htm(2,3)=ydir*zdir*(1-cos(nominal))-xdir*sin(nominal);
               htm(3,3)=zdir^2+cos(nominal)*w^2;
               htm(1,4)=nominal*pitch*xdir;
               htm(2,4)=nominal*pitch*ydir;
               htm(3,4)=nominal*pitch*zdir;
            else
               disp('direction vector for DOF,DOS, or DOC must be unit length');
               return
            end
            if sign(CHc(i,j))==-1
               htm=inv(htm);
            end
         else
            disp('Dimension direction must be integer between 1 and 6');
            return
         end
      end
      Loop((4*i-3):4*i,1:4)=Loop((4*i-3):4*i,1:4)*htm;
   end
end

%Calculating the loop closure error
count=0;
loop=0;
for i=1:(4*lambda)
```

```
   count=count+1;
   for j=1:4
      if count>4
         count=1;
      else
      end
      if count==j
      else
         loop=loop+Loop(i,j)^2;
      end
   end
end

%Calculating the combined loop closure/energy min. objective function
energyplusloop=energy+loop*10^8;
```

combiner.m

```
%Filename: combiner.m
%Capability: Relates twist or wrenches to each other using loop-mate,
%part-mate, or loop-dimension matrix.
%Author: J. Michael Gray, MIT Graduate Student, Mech. Eng. Dept.
%Date completed: May 1st, 2004
%Revisions:
function Network=combiner(Elements,Connections)

%Getting sizes of matrices
[e1,e2]=size(Elements);
[c1,c2]=size(Connections);
Network=zeros(6*c1,e1);

%Combining "Elements" according to topology data in "Connections"
if e2==6 & c2==e1
   for i=1:6*c1
      for j=1:e1
         Network(i,j)=Elements(j,mod(i-1,6)+1)*Connections(fix((i-1)/6)+1,j);
      end
   end
elseif e1==0 & e2==0 & c1==0 & c2==0
   %Null matrix returned if there are no elements and connections
   Network=zeros(0,0);
else
   disp('Input matrices are the wrong size.');
   return
end
```

constraintdetector.m

```
%Filename: constraintdetector.m
%Capability: Finds excess over-constraints and under-constraints in an assembly.
%Author: J. Michael Gray, MIT Graduate Student, Mech. Eng. Dept.
%Date completed: May 1st, 2004
%Revisions:

function [fe,ce,TLr,WPr]=constraintdetector(TL,WP,SK,g)
%Getting the sizes of the matrices and determining counting variables:
%st=DOS, ct=DOC, p=parts, lambda=closed-loops, g=open-loops, ft=DOF
[st,st2]=size(SK);
[tl1,tl2]=size(TL);
[wp1,wp2]=size(WP);
ct=wp2-st;
p=(wp1+6)/6;
lambda=(tl1-6*g)/6;
m=lambda+p-1;
ft=6*m-st-ct;

%r1=rank of twist loop matrix, r2=rank of wrench part matrix
r1=rank(TL);
r2=rank(WP);

%ce=excess over-constraints
ce=6*(lambda+g)-r1;
if st==0
  %fe=excess under-constraints if no compliance
  fe=ft+6*g-r1;
else
  %fe=excess under-constraints if compliance
  fe=6*(p-1)-r2;
end
if fe~=0
  %reduced twist loop matrix that is useful for identifying excess DOF
  TLr=null(TL(1:6*lambda,1:ft),'r');
else
  %Returns null matrix if no excess under-constraints
  TLr=[];
end
if ce~=0
  %reduced wrench part matrix that is useful for identifying excess DOC
  WPr=null(WP(1:6*(p-1),1:ct),'r');
else
  %Returns null matrix if no excess over-constraints
  WPr=[];
end
```

manufacturability.m

```
%Filename: manufacturability.m
%Capability: Determines the CPk of the KCs, their sensitivities, coupling,
%and the dimensions that contribute to their variability.
%Author: J. Michael Gray, MIT Graduate Student, Mech. Eng. Dept.
%Date completed: May 1st, 2004
%Revisions:
function [Da,Dv,KCS,Cpk,Rej,Co,KCv,KCc,KCa,KCn,KCL]=manufacturability(Dn, Dd, Dul, Dll,
CHd, Dt, Dp, Fn, Fd, Ful, Fll,Sn, Sd, Sul, Sll, SK, Cn, Cd, CHm, CHc, CHo, KCd, KCt, KCl,CHk,
LM, PM, LD, KC, KCT)

%Getting the sizes of the input matrices and vectors
[dn1,dn2]=size(Dn); [dd1,dd2]=size(Dd); [dul1,dul2]=size(Dul); [dll1,dll2]=size(Dll);
[dt1,dt2]=size(Dt); [dp1,dp2]=size(Dp); [chd1,chd2]=size(CHd); [fn1,fn2]=size(Fn);
[fd1,fd2]=size(Fd); [ful1,ful2]=size(Ful); [fll1,fll2]=size(Fll); [sn1,sn2]=size(Sn);
[sd1,sd2]=size(Sd); [sul1,sul2]=size(Sul); [sll1,sll2]=size(Sll); [sk1,sk2]=size(SK);
[cn1,cn2]=size(Cn); [cd1,cd2]=size(Cd); [chm1,chm2]=size(CHm); [chc1,chc2]=size(CHc);
[cho1,cho2]=size(CHo); [kcd1,kcd2]=size(KCd); [kct1,kct2]=size(KCt); [kcl1,kcl2]=size(KCl);
[chk1,chk2]=size(CHk); [lm1,lm2]=size(LM); [pm1,pm2]=size(PM); [ld1,ld2]=size(LD);
[kc1,kc2]=size(KC); [KCT1,KCT2]=size(KCT);

%Checking to make sure the matrices are the right sizes
if dn1~=dd1 | dn1~=chd1 | dn1~=dul1 | dn1~=dll1 | dn1~=dt1 | dn1~=dp1 | dn2~=1 | dd2~=1 |
dul2~=1 | dll2~=1 | dp2~=3
   disp('');
   disp('Dimension related inputs Dn, Dd, Dul, Dll, Dt, Dp, and/or CHd is/are the wrong size.');
   disp('End of program.');
   disp('');
   return
end
if fn1~=fd1 | fn1~=ful1 | fn1~=fll1 | fn2~=1 | fd2~=4 | ful2~=1 | fll2~=1
   disp('');
   disp('DOF related inputs Fn, Fd, Ful, and/or Fll is/are the wrong size.');
   disp('End of program.');
   disp('');
   return
end
if sn1~=sd1 | sn1~=sul1 | sn1~=sll1 | sn1~=sk1 | sn1~=sk2 | sn2~=1 | sd2~=4 | sul2~=1 | sll2~=1
   disp('');
   disp('DOS related inputs Sn, Sd, Sul, Sul, and/or SK is/are the wrong size.');
   disp('End of program.');
   disp('');
   return
end
if cn1~=cd1 | cn2~=1 | cd2~=4
   disp('');
   disp('DOC related inputs Cn and/or Cd is/are the wrong size.');
   disp('End of program.');
   disp('');
   return
end
if kcd1~=chk1 | kcd1~=kct1 | kcd1~=kcl1 | kcd1~=kc1 | kcd2~=1 | kct2~=1 kcl2~=1
```

```
   disp('');
   disp('KC related inputs KCd, KCl, KC, KCT, KCt, and/or CHk is/are the wrong size.');
   disp('End of program.');
   disp('');
   return
end

if kc2~=(fn1+2*sn1+6*cho1+cn1) | KCT1~=1 | KCT2~=1
   disp('');
   disp('KC related inputs KCd, KCl, KC, KCT, KCt, and/or CHk is/are the wrong size.');
   disp('End of program.');
   disp('');
   return
end

%storing counting variables: dt=dimensions, ft=DOF, st=DOS, ct=DOC, kct=KCs
dt=dn1;
ft=fn1;
st=sn1;
ct=cn1;
kct=kcd1;

%More checking to make sure the matrices are the right size
if chm1~=(ft+st+ct)
   disp('');
   disp('Mate related input CHm has the wrong number of rows.');
   disp('End of program.');
   disp('');
   return
end
if st==0 & ct==0
   p=1;
   m=1;
   if lm1~=(chc1+cho1) | lm2~=(ft+st+6*cho1) | pm2~=0 | pm1~=0
      disp('');
      disp('loop and part related inputs CHc, Cho, LM, and/or PM is/are the wrong size.');
      disp('End of program.');
      disp('');
      return
   end
else
   m=chm1/6;
   p=pm1+1;
   if chc1~=(m-pm1) | lm1~=(chc1+cho1) | lm2~=(ft+st+6*cho1) | pm2~=(ct+st)
      disp('');
      disp('Loop and part related inputs CHc, Cho, LM, and/or PM is/are the wrong size.');
      disp('End of program.');
      disp('');
      return
   end
end

%storing counting variables: lamba=closed-loops, g=open-loops
```

```
lambda=chc1;
g=cho1;

%More checking to make sure the matrices are the right size
if ld1~=(lambda+g)  | ld2~=dt
   disp('');
   disp('Dimension related input LD is the wrong size.');
   disp('End of program.');
   disp('');
   return
end

%Finding the average dimension values (Da) and dimension variances (Dv)
Da=zeros(dt,1);
Dv=zeros(dt,dt);
for i=1:dt
   Da(i,1)=icdf(strcat(Dt(i,1:dt2)),.5,Dp(i,1),Dp(i,2),Dp(i,3));
   Dv(i,i)=var(icdf(strcat(Dt(i,1:dt2)),normcdf(randn(1,100000),0,1),Dp(i,1),Dp(i,2),Dp(i,3)));
end

%Finding the average KC values (KCa)
[B,U,TL,WP,g,beta_energy,Fnstar,Snstar,KCn,KCL]=modeler(Dn, Dd, CHd, Fn, Fd, Ful, Fll,Sn,
Sd, Sul, Sll, SK, Cn, Cd, CHm, CHc, CHo, KCd, KCt, KCl,CHk, LM, PM, LD);
[B_a,U_a,TL_a,WP_a,g_a,beta_energy_a,Fnstar_a,Snstar_a,KCa,KCL_a]=modeler(Da, Dd,
CHd, Fn, Fd, Ful, Fll,Sn, Sd, Sul, Sll, SK, Cn, Cd, CHm, CHc, CHo, KCd, KCt, KCl,CHk, LM, PM,
LD);

%Checking to make sure the assembly is properly constrained (fe=0 ce=0)
[fe,ce,TLr,WPr]=constraintdetector(TL,WP,SK,g);
if fe~=0
   if ce==0
      disp('');
      disp('This assembly cannot be analyzed because it is under-constrained.');
      Number_of_Excess_Underconstraints=fe
      TLr
      disp('End of program.');
      disp('');
      return
   else
      disp('');
      disp('This assembly cannot be analyzed because it is both over-constrained & under-
constrained.');
      Number_of_Excess_Underconstraints=fe
      TLr
      Number_of_Excess_Overconstraints=fe
      WPr
      disp('End of program.');
      disp('');
      return
   end
end
if ce~=0
   disp('');
   disp('This assembly cannot be analyzed because it is over-constrained.');
   Number_of_Excess_Overconstraints=fe
   WPr
   disp('End of program.');
   disp('');
   return
end

%Solving for the KC sensitivity matrix at nominal (KCS)
KCS=KC*(-inv(B)*U);

%Solving for the KC sensitivity matrix at average (KCS)
KCS_a=KC*(-inv(B_a)*U_a);

%Solving for the KC covariance matrix (KCv)
KCv=KCS*Dv*KCS';

%Solving for the KC correlation matrix (KCc)
KCc=zeros(kct,kct);
for i=1:kct
   for j=1:kct
      KCc(i,j)=KCv(i,j)/(KCv(i,i)^.5*KCv(j,j)^.5);
   end
end

%Initializing the Cpk, Reject, and Contribution matrices
Cpk=zeros(kct,1);
Rej=zeros(kct,1);
Co=zeros(kct,dt);

%Solving for the MPP point
for i=1:kct
   Dn_m=Da;
   KCn_m=KCa;
   Fnstar_m=Fnstar_a;
   Snstar_m=Snstar_a;
   KCS_m=KCS_a;
   for j=1:6
      if abs(KCn_m(i,1)-KCL(i,1))<.001*abs(KCn(i,1)-KCL(i,1))
         break
      else
         %Iterative search algorithm looking for the MPP point
         [gamma,Dn_m]=mppfinder(Dn_m,KCn_m(i,1),KCL(i,1),Dt,Dp,KCS_m(i,1:dt),KCn(i,1));

         %Solving for the updated KC nominal values  (KCn_m(n+1))

[B_m,U_m,TL_m,WP_m,g_m,beta_energy_m,Fnstar_m,Snstar_m,KCn_m,KCL_m]=modeler(Dn
_m, Dd, CHd, Fnstar_m, Fd, Ful, Fll,Snstar_m, Sd, Sul, Sll, SK, Cn, Cd, CHm, CHc, CHo, KCd,
KCt, KCl,CHk, LM, PM, LD);

         %Checking to make sure the assembly is properly constrained (fe=0 ce=0)
         [fe_m,ce_m,TLr_m,WPr_m]=constraintdetector(TL_m,WP_m,SK,g_m);
         if fe~=0
```

```
            if ce==0
               disp('');
               disp('This assembly cannot be analyzed because it is under-constrained.');
               Number_of_Excess_Underconstraints=fe
               TLr_m
               disp('End of program.');
               disp('');
               return
            else
               disp('');
               disp('This assembly cannot be analyzed because it is both over-constrained & under-constrained.');
               Number_of_Excess_Underconstraints=fe
               TLr_m
               Number_of_Excess_Overconstraints=fe
               WPr_m
               disp('End of program.');
               disp('');
               return
            end
         end
         if ce~=0
            disp('');
            disp('This assembly cannot be analyzed because it is over-constrained.');
            Number_of_Excess_Overconstraints=fe
            WPr_m
            disp('End of program.');
            disp('');
            return
         end

         %Solving for the updated KC sensitivity matrix (KCS_m)
         KCS_m=KC*(-inv(B_m)*U_m);
      end
   end

   %Solving for the Contribution, CPk, and Reject Matrices
   Co(i,1:dt)=(100*(gamma.*gamma)/norm(gamma)^2)';
   if (abs(KCa(i,1)-KCn(i,1))+abs(KCa(i,1)-KCL(i,1)))>abs(KCL(i,1)-KCn(i,1)) & abs(KCa(i,1)-KCn(i,1))>abs(KCa(i,1)-KCL(i,1))
      Cpk(i,1)=0;
      Rej(i,1)=100*(1-normcdf(-norm(gamma),0,1));
   else
      Cpk(i,1)=norm(gamma)/3;
      Rej(i,1)=100*normcdf(-norm(gamma),0,1);
   end
end
```

mppfinder.m

```
%Filename: mppfinder.m
%Capability: Estimates the location of the MPP point
%Author: J. Michael Gray, MIT Graduate Student, Mech. Eng. Dept.
%Date completed: May 1st, 2004
%Revisions:
function [gamma,Dn_m]=mppfinder(Dn_m,KCn_m,kcl,Dt,Dp,KCS_m,kcn);
%Getting the sizes of the matrices
[dt,z]=size(Dn_m);
[dt1,dt2]=size(Dt);

%Initializing the variables
gradQ=zeros(dt,1);
alphan=0;
alpha=zeros(dt,1);
u1=zeros(dt,1);
u2=zeros(dt,1);
u3=zeros(dt,1);
gamma=zeros(dt,1);

%Transforming the dimensions into the standard normal space
for i=1:dt
   if strcmpi(strcat(Dt(i,1:dt2)),'Normal')==1 | strcmpi(strcat(Dt(i,1:dt2)),'norm')==1
      cumulative=cdf(strcat(Dt(i,1:dt2)),Dp(i,1)-abs(Dp(i,1)-Dn_m(i,1)),Dp(i,1),Dp(i,2),Dp(i,3));
      if cumulative==0
         cumulative=1e-300;
      end
      u1(i,1)=norminv(cumulative)*sign(Dp(i,1)-Dn_m(i,1));
   else
      cumulative=cdf(strcat(Dt(i,1:dt2)),Dn_m(i,1),Dp(i,1),Dp(i,2),Dp(i,3));
      if cumulative==0
         cumulative=1e-300;
      end
      if cumulative==1
         cumulative=0.9999999999999999;
      end
      u1(i,1)=norminv(cumulative);
   end
   %Finding the gradient of steepest decent towards the MPP
   gradQ(i,1)=sign(kcl-
kcn)*KCS_m(1,i)*(normpdf(u1(i,1),0,1))/pdf(strcat(Dt(i,1:dt2)),Dn_m(i,1),Dp(i,1),Dp(i,2),Dp(i,3));
end
%Finding the norm of the gradient
alphan=norm(gradQ);

%Finding the unit vector for the new direction
alpha(1:dt,1)=-gradQ(1:dt,1)*(1/alphan);
u2(1:dt,1)=alpha(1:dt,1)*(u1(1:dt,1)'*alpha(1:dt,1));

%Estimating distance between current point and MPP
u3(1:dt,1)=alpha(1:dt,1)*((KCn_m-kcl)/alphan);
```

```
%Estimating the gamma vector that points to the MPP in the std. norm. space
gamma(1:dt,1)=u2(1:dt,1)+u3(1:dt,1);

%Transforming the dimensions back into their original spaces
for i=1:dt
   if strcmpi(strcat(Dt(i,1:dt2)),'Normal')==1 | strcmpi(strcat(Dt(i,1:dt2)),'norm')==1
      n_cumulative=normcdf(-abs(gamma(i,1)));
      if n_cumulative==0
         n_cumulative=1e-300;
      end
      Dn_m(i,1)=Dp(i,1)+sign(gamma(i,1))*abs(Dp(i,1)-
icdf(strcat(Dt(i,1:dt2)),n_cumulative,Dp(i,1),Dp(i,2),Dp(i,3)));
   else
      n_cumulative=normcdf(gamma(i,1));
      if n_cumulative==0
         n_cumulative=1e-300;
      end
      if n_cumulative==1
         n_cumulative=0.9999999999999999;
      end
      Dn_m(i,1)=icdf(strcat(Dt(i,1:dt2)),n_cumulative,Dp(i,1),Dp(i,2),Dp(i,3));
   end
end
```

dimoptimizer.m

```
%Filename: dimoptimizer.m
%Capability: Tries to improve the CPks of the KCs by changing the nominal
%dimension values.
%Author: J. Michael Gray, MIT Graduate Student, Mech. Eng. Dept.
%Date completed: May 1st, 2004
%Revisions:
function [Dnstar,weighted_Cpk]=dimoptimizer(Dn, Dd, Dul, Dll, CHd, Dt, Dp, Fn, Fd, Ful, Fll,Sn,
Sd, Sul, Sll, SK, Cn, Cd, CHm, CHc, CHo, KCd, KCt, KCl,CHk, LM, PM, LD, KC, KCT, Wa)

%Finding Nominal KC values
[B,U,TL,WP,g,beta_energy,Fnstar,Snstar,KCn_target,KCL]=modeler(Dn, Dd, CHd, Fn, Fd, Ful,
Fll,Sn, Sd, Sul, Sll, SK, Cn, Cd, CHm, CHc, CHo, KCd, KCt, KCl,CHk, LM, PM, LD);

%Getting the number of dimensions to be optimized
[dt,z]=size(Dn);

%Starting procedure to find optimal dimensions
NONLCON=@dimconstraint;
FUN=@dimobjective;
OPTIONS1 =
Optimset('LargeScale','off','MaxFunEvals',20*dt,'MaxIter',20*dt,'Display','off','TolFun',1e-
4*max(abs(Dn)),'TolCon',1e-9*max(abs(Dn)));
[Dnstar,FVAL,EXITFLAG]=fmincon(FUN,Dn,[],[],[],[],Dll,Dul,NONLCON,OPTIONS1, Dd, Dul, Dll,
CHd, Dt, Dp, Fn, Fd, Ful, Fll,Sn, Sd, Sul, Sll, SK, Cn, Cd, CHm, CHc, CHo, KCd, KCt, KCl,CHk,
LM, PM, LD, KC, KCT, Wa,KCn_target);

weighted_Cpk=-FVAL;

%Error messages if fmincon terminated early or used all the iterations
if EXITFLAG==0
   disp('');
   disp('Maximum number of iterations reached');
   disp('');
end
if EXITFLAG<0
   disp('');
   disp('Could not optimize dimensions');
   disp('End of program.');
   disp('');
   return
end
```

dimobjective.m

```
%Filename: dimobjective.m
%Capability: Finds the weighted Cpk
%Author: J. Michael Gray, MIT Graduate Student, Mech. Eng. Dept.
%Date completed: May 1st, 2004
%Revisions:
function [objective]=dimobjective(Dn, Dd, Dul, Dll, CHd, Dt, Dp, Fn, Fd, Ful, Fll,Sn, Sd, Sul, Sll,
SK, Cn, Cd, CHm, CHc, CHo, KCd, KCt, KCl,CHk, LM, PM, LD, KC, KCT, Wa,KCn_target)

%Executes manufacturability.m to find Cpks
[Da,Dv,KCS,Cpk,Rej,Co,KCv,KCc,KCa,KCn,KCL]=manufacturability(Dn, Dd, Dul, Dll, CHd, Dt,
Dp, Fn, Fd, Ful, Fll,Sn, Sd, Sul, Sll, SK, Cn, Cd, CHm, CHc, CHo, KCd, KCt, KCl,CHk, LM, PM,
LD, KC, KCT);

%Weights Cpks according to weights in Wa
objective=-Wa'*Cpk;
```

dimconstraint.m

```
%Filename: dimconstraint.m
%Capability: Insures that nominal performance stays the same and that
%packaging constraints are met
%Author: J. Michael Gray, MIT Graduate Student, Mech. Eng. Dept.
%Date completed: May 1st, 2004
%Revisions:
function [C,Ceq]=dimconstraint(Dn, Dd, Dul, Dll, CHd, Dt, Dp, Fn, Fd, Ful, Fll,Sn, Sd, Sul, Sll, SK,
Cn, Cd, CHm, CHc, CHo, KCd, KCt, KCl,CHk, LM, PM, LD, KC, KCT, Wa,KCn_target)

%Finding the Nominal KC values
[B,U,TL,WP,g,beta_energy,Fnstar,Snstar,KCn,KCL]=modeler(Dn, Dd, CHd, Fn, Fd, Ful, Fll,Sn,
Sd, Sul, Sll, SK, Cn, Cd, CHm, CHc, CHo, KCd, KCt, KCl,CHk, LM, PM, LD);

%finding out how many KCs there are
[kct,z]=size(KCn);

%Initializing constraint variables
count=0;
for i=1:kct
   if KCt(i,1)==1
   else
      count=count+1;
   end
end
if count==0
   C=0;
else
   C=zeros(count,1);
end
if count==kct
   Ceq=0;
else
   Ceq=zeros(kct-count,1);
end
count=0;
%Calculating constraints that insure KC nominal values stay the same
for i=1:kct
   if KCt(i,1)==1
      Ceq(i-count,1)=abs(KCn_target(i,1)-KCn(i,1));
   else
      C(count+1,1)=KCn(i,1);
      count=count+1;
   end
end
```