



**A Comprehensive System for Modeling  
Variation in Mechanical Assemblies**

ADCATS Report No. 93-2

Angela Trego, graduate student  
Department of Mechanical Engineering  
Brigham Young University

MS Thesis sponsored by ADCATS

April 1993

**ABSTRACT**

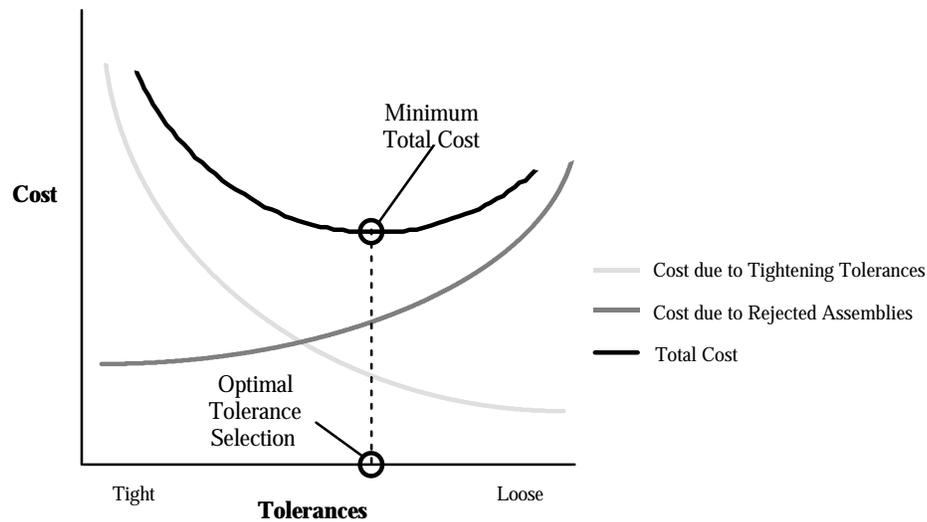
Tolerance analysis allows the designer to quantitatively estimate the affects of variation on design requirements in the early design phases. Tolerances play a significant role in the development and cost of manufactured products. By creating assemblies which perform properly, are cost efficient and readily manufacturable, engineers can assist in producing high quality, marketable products.

Research presented in this thesis focuses on developing a comprehensive 2-D system for modeling manufacturing variations. Previous studies included creating an engineering model based on part, datums, kinematic joints, vector loops, and form variations. Contributions from this thesis include: 1) A comprehensive system for defining assembly tolerance specifications. Just as feature controls are applied to individual parts, a similar system of assembly tolerance specifications may be applied to assemblies of parts. 2) A generalized model for manual process variations due to fastener clearances or other non-deterministic variations. 3) An algorithm for automatic open loop generation. Open vector loops are often necessary to represent design constraints on an assembly. 4) A method for detecting degree of freedom redundancy.

## CHAPTER 1: INTRODUCTION

---

Tolerances play a significant role in the development and cost of manufactured products. Tolerances are required on mechanical assemblies to ensure proper assembly and function, but at the same time tight tolerances drive up the cost of manufacturing (see figure 1.1). A product's manufacturability is important in the design process. If the product cannot be manufactured profitably, the designer has failed. By creating assemblies which are useful, cost efficient and readily manufacturable, engineers can assist in producing high quality, marketable products.

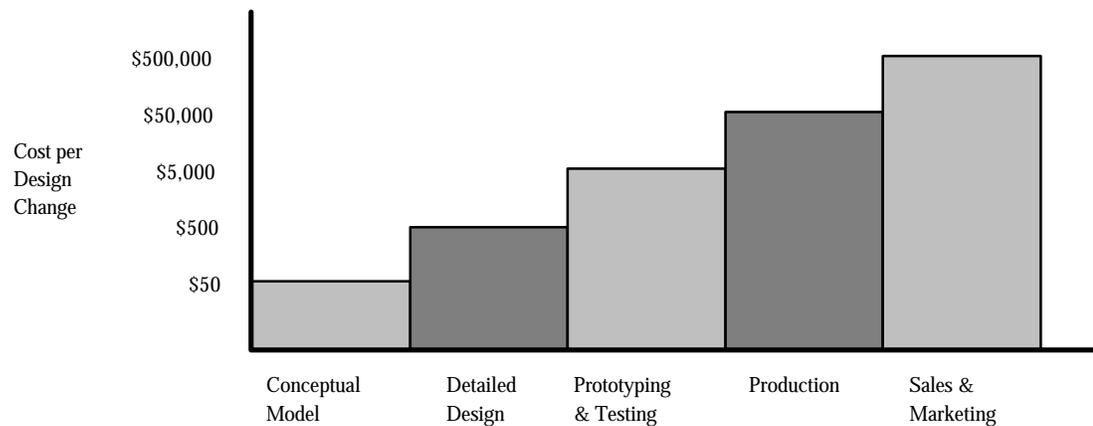


**Figure 1.1.** Cost versus Tolerances.

All manufactured parts exhibit variation which may accumulate in an assembly. As a result of the variation or tolerance stack-up the assembly may not function properly. Tolerance analysis allows the designer to quantitatively estimate the affects of variation on design requirements. It ties the component tolerance requirements, which manufacturing systems people must meet, to the assembly tolerance requirements, which result from engineering design requirements. Applying tolerance analysis in the early design phases, allows the designer to save valuable time and money by decreasing rework costs.

As shown in figure 1.2, rework cost increases dramatically as a design progresses from concept to sales. Profit loss due to design changes therefore increases at later phases of the product development processes. It is desirable to do a better job of design in the early stages to avoid costly changes later. New programs, such as concurrent or simultaneous

engineering, design for manufacture, and design for assembly are attempts to formalize the process of considering the manufacturing consequences of design decisions.

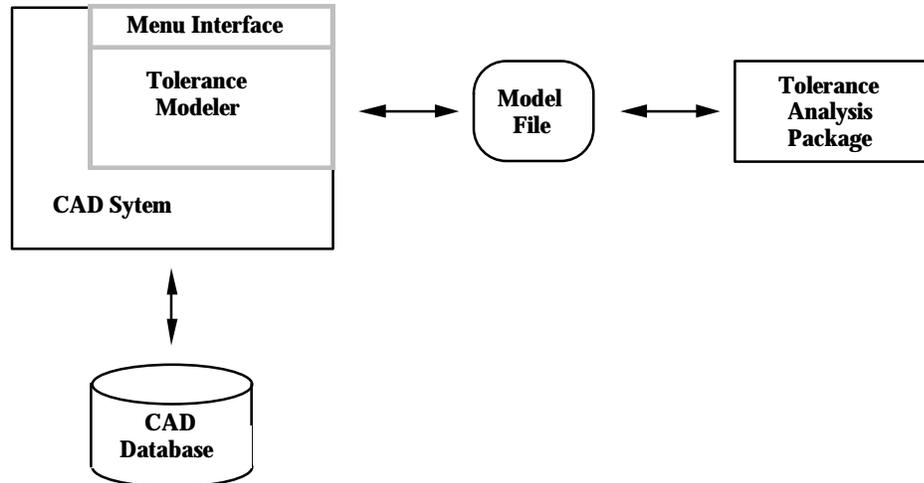


**Figure 1.2.** Cost of design changes.

The next section will describe a new quantitative tool for integrating manufacturing considerations into the engineering design process.

### **CAD Tolerance Modeling**

Engineers are beginning to utilize CAD/CAM for more than just drawings, but also for applying tolerance modeling and analysis in the early design stages. A CAD database stores the geometric information required for tolerance analysis. By creating an interface with the CAD system, information may be directly accessed from the drawings, instead of tediously determined by hand. Engineering models define a set of relationships which describe how different assembly parts interact due to manufacturing variations. These relationships are in the form of vector loops joined by kinematic joints placed at the contact points between pairs of mating parts. These engineering models of assemblies require complex equations. CAD modeling systems allow the equations to be created graphically and stored as part of the CAD model. CAD/CAM drawings then allow the designer to change tolerances with a stroke of a key.



**Figure 1.3.** CAD tolerance modeling system with analysis package.

Figure 1.3 illustrates a new tolerance modeling system which is integrated with a CAD system. A menu-driven, user-interface commands the modeler and appends the assembly tolerance model to the CAD database just as the CAD system itself accesses geometry from its database. Once the model has been created, a neutral file, or file containing essential information needed to perform tolerance analysis is generated. This model file may then be sent to the analysis package. The assembly tolerance analysis package predicts the magnitude of the variations which occur at assembly time. Results are then passed back to the CAD system along with tolerance modifications for incorporation into the drawings.

### Research Objectives

The objectives of this thesis are to 1) create a more complete assembly modeling system which is capable of representing a wider range of assembly problems, and 2) develop algorithms for automating more assembly modeling tasks in order to decrease the modeling expertise required by design engineers and to reduce modeling errors.

### Method

A comprehensive 2-D system will be developed for defining assembly tolerance specifications, modeled after the ANSI Y14.5 system for feature controls on single parts. This work is being carried out jointly with David Carr who is currently working on methods for solving open loop equations for each new design specification [Carr, 1993].

A new modeling element called manual process variations will also be defined. The data structure and neutral file will be updated to allow for the new modeling elements. In addition, an improved system for testing degree of freedom (DOF) redundancy will be developed.

Algorithms for automatic open loop generation will be studied, complimenting existing algorithms for closed loops. Design procedures will be examined and a more intuitive system for adding these entities to a tolerance model will be devised. Finally, the new modeling techniques will be implemented into AutoCAD using the programming language AutoLISP.

### **Overview**

The following chapter discusses research related to computer aided tolerance modeling and analysis. Chapter three introduces assembly tolerances or design specifications. Automation of kinematic chain or open vector loop generation is discussed in chapter four. Analysis and user enhancements are presented in chapter five. Chapter six presents example assemblies to be modeled and analyzed. Conclusions and recommendations for future research are in chapter seven.

## **CHAPTER 2: BACKGROUND**

---

Recently, research on computer-aided tolerance modeling using CAD systems has grown with many new applications and developments. This chapter will focus on first, general tolerance modeling requirements and secondly, the existing CAD tolerance modeling system.

### **Tolerance Modeling Requirements**

With most assemblies stored in CAD systems, precise geometric data is available for engineering applications. Although this provides a great deal of information, it is not sufficient for assembly tolerance analysis. Srikanth has pointed out that graphical representation of individual part interfaces may be provided by CAD databases, but CAD applications end there [Srikanth et al., 1990; Shah et al., 1990]. The CAD systems need to provide more than just graphical input of geometry, rather they need to provide kinematic relationships between mating parts and assembly constraints.

Ranyak and Fridshal [Ranyak, 1988] created a tolerance analysis method on a feature-based solid modeler. They created a hierarchical feature model for the specific purpose of defining tolerances on the feature. Since a feature is to a part as a part is to an assembly, this method was limited to tolerance specifications for single parts only. CAD systems needs to provide information on tolerance limits applied to the size, form, orientation and location of each feature of a component, as well as the assembly tolerance limits set by engineering design requirements.

Srikanth and Turner took the feature based method one step further. They discuss an assembly hierarchy which includes mating features between individual parts [Srikanth et al., 1990]. The kinematic relationships are then combined with geometric information creating an entire mechanical assembly model.

Larsen [Larsen 1991] noted that assembly tolerance analysis models need to accurately account for:

- Relative rigid body translation and rotation between components
- Propagation of size variations
- Propagation of form and feature variations
- Propagation of kinematic adjustments
- Tolerance accumulation

Rigid body motion is the resultant motion within an assembly due to a single part's variation. Every feature on a part deviates from the ideal design. Form and feature variations also must be modeled just as size variations need to be modeled. Finally, kinematic adjustments, or adjustments at assembly time due to varied part dimensions, are a crucial part in accurately modeling assembly variations.

According to Shah and Miller, a tolerance model, besides being geometrically associated with entities in the CAD database, must support geometric tolerances (ANSI Y14.5 specifications), datum reference frames and datum precedence, provide default tolerances and graphical feedback [Shah, Miller 1990; Larsen 1991]. Faux [Faux, 1981] also supports the need for an assembly data structure which includes mating features and the nature of fit between assembly components. A CAD tolerance modeler, such as figure 1.1, which includes these qualities could provide the essential graphical interface which allows both geometric and kinematic information to be analyzed. Many steps have been taken to model all assembly variations correctly using a CAD database system such as figure 1.1 [Larsen, 1991; Marler, 1988; Robison, 1989; Ward, 1992].

For an assembly tolerance modeler to be complete it must also have the capability of providing parameters to the governing equations of an assembly. It is desirable to automate this process as much as possible. Wang introduced a method of generating kinematic chains automatically with the application of network graph theory [Wang, 1990]. This theory has been developed for a limited type of assembly models [Robison, 1989; Simmons, 1990; Larsen, 1991].

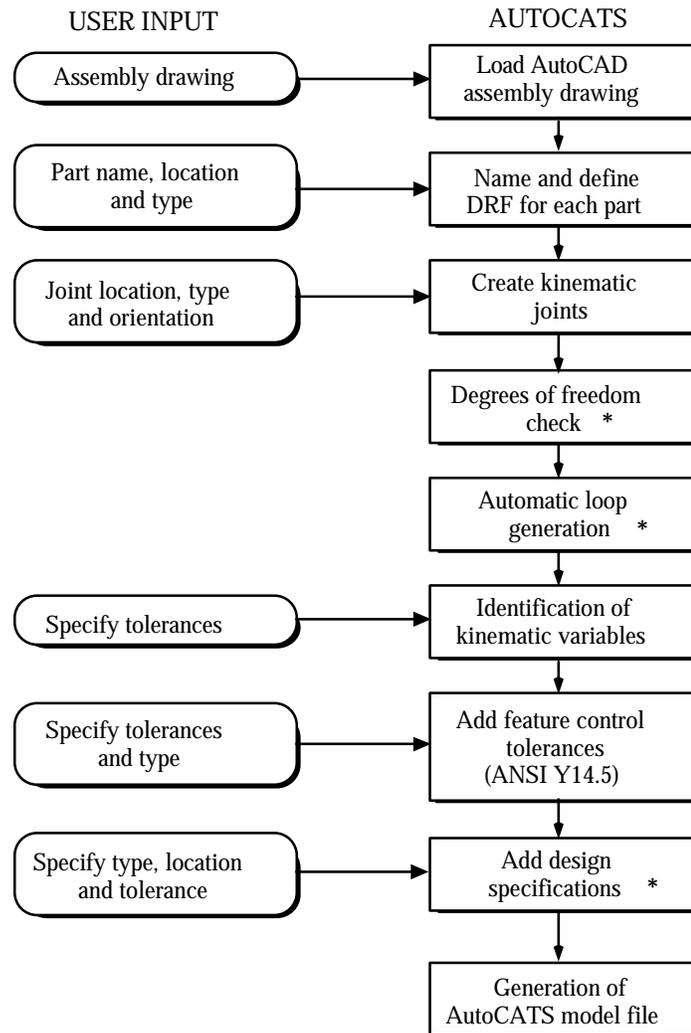
Now that the general requirements for an assembly tolerance modeler have been defined, an in-depth look at the existing CAD modeling system which is based on this approach will be discussed. This existing system forms the starting basis for the research of this thesis.

## **Modeling Assemblies**

The existing prototype modeling package, AutoCATS, uses a hierarchical structure as suggested by both Ranyak and Srikanth [Ranyak, 1988; Srikanth et al., 1990]. CATS.BYU, Computer-Aided Tolerance Specification, is a system for assembly tolerance analysis which has been developed through research efforts at Brigham Young University since 1984. The software allows the designer to create 2-D assembly models using a graphical preprocessor called AutoCATS, and use these models for predicting design consequences of manufacturing variations.

The assembly drawing is first created in a CAD system. Using a step-by-step procedure, a model is created using parts, feature datums, kinematic joints, form tolerances, and vector loops (figure 2.1). Datum reference frames distinguish each part of an assembly. Kinematic joints relate contact points between individual parts. They are located relative to each part by datum paths, or vector paths. Form tolerances may also be applied at each interface. Vector loops are then created which join relevant dimensions into chains describing resultant assembly dimensions and related features. A neutral file is then generated which may be sent to an analysis software package which generates the governing equations. The equations permit the analysis of tolerance accumulation and prediction of assembly reject rates due to failure to meet assembly tolerance limits.

### Procedure for Creating AutoCATS Models for Tolerance Analysis of Assemblies

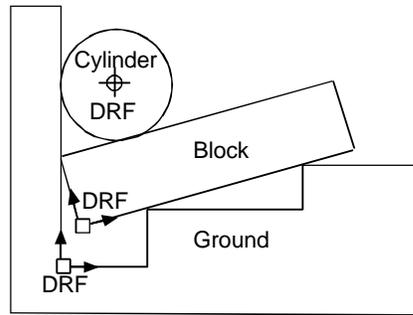


**Figure 2.1.** Modeling procedure for AutoCATS.

\* New modeling steps by author

## PARTS

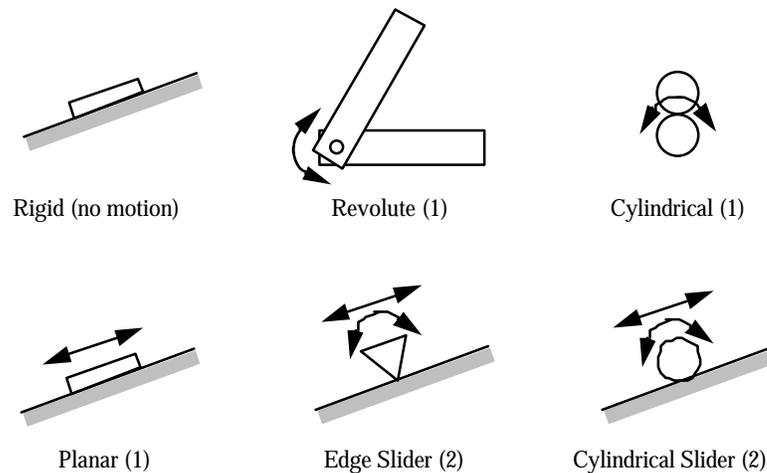
In order to distinguish between individual parts of an assembly and to permit datum-referenced dimensioning, a datum reference frame (DRF) is created on each part. A DRF is a local coordinate system for each part. During production, all part features are ultimately referenced to these specified datum planes used to fixture the part. Selection of the location for the DRF origin determines to a great extent which part dimensions will contribute to assembly variations. Figure 2.2 shows a three part assembly with a DRF defined on each part.



**Figure 2.2.** Assembly with Parts and DRFs defined.

## JOINTS

The points of contact between mating assembly parts are called joints. Kim and Lee first developed the system of deriving an assembly model only from mating conditions [Kim et al., 1989]. A joint defines a kinematic pair which constrains the relative motion between two mating parts. For example, a block on a plane is constrained to slide parallel to the plane. There are six joint types which model a wide variety of 2-D assembly conditions. They are shown below in figure 2.3 with their associated degrees of freedom. Each joint type reduces the degrees of freedom of an assembly by constraining motion.



**Figure 2.3.** 2-D joint types with associated degrees of freedom (DOF).

Each joint must be located relative to the DRF of both parts connected by the joint. The chain of vectors which locates the joint from a datum reference frame (DRF) is called a datum path. Each vector in a datum path must be either a controlled component dimension, for which the designer may specify a tolerance, or a kinematic assembly dimension, which adjusts at assembly time. Kinematic dimensions are determined by a

chain of component dimensions, hence the tolerances are the result of a tolerance stack-up of component tolerances.

## LOOPS

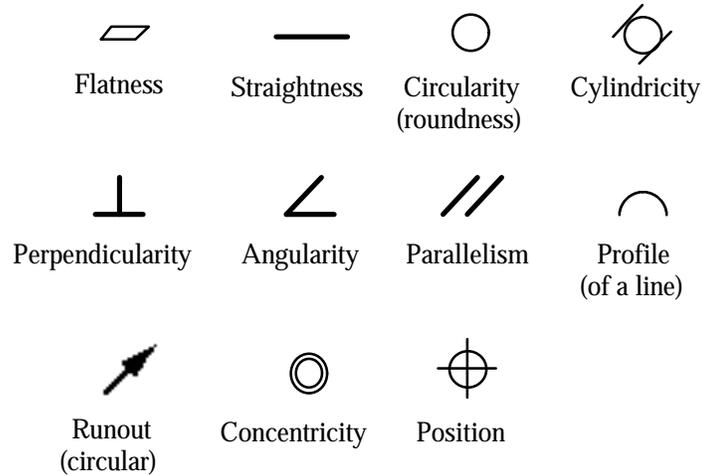
The relationships representing the assembly, both geometrically and kinematically, are obtained by creating a set of vector loops which connect contact joints in the assembly. The loops may be either open or closed:

- *Closed loops* start and end at the same location and represent kinematic constraints on the assembly. For example, a kinematic constraint may state that all parts in the assembly must maintain contact in order for the tolerance model to be valid [Larsen, 1991].
- *Open loops* are used to determine assembly resultants of interest such as a clearance, orientation or position. For example, a fan blade must maintain a certain clearance in an assembly to operate properly.

In the previous system, closed loops were generated either manually or automatically using a generalized modeling approach [Larsen, 1991]. Larsen's algorithm was limited to the generation of closed loops only and did not allow for zero length vectors in the loop. Once a complete loop is formed from joints and datum paths, tolerances are specified on those vector lengths which are independent of assembly adjustments. That is, tolerances must be specified on those vectors which correspond to manufactured feature dimensions, such as the radius of a corner. Tolerances may also be optionally applied to any angles between the vectors in an assembly which are independent (the angle between two adjacent vectors on the same part) as these represent machined or manufactured angles. Dimensions or angles which correspond to kinematic variables adjust during assembly. Their variations are determined by kinematic analysis [Chase, 1992].

## FEATURES

Feature variations are variations in shape or form of a part. For example, a machined cylinder will not be perfectly cylindrical, rather, it may be slightly elliptical. The cylinder may still be circular enough to perform its function properly. Feature control tolerances, which include form, location, position and orientation, constrain a variation to fall within a certain tolerance zone. The ANSI Y14.5 standard symbols for form and feature variations are shown in figure 2.4. Current study on feature variations include developing mathematical models for each type of feature variation [Goodrich, 1991; Ward, 1992].



**Figure 2.4.** ANSI Y14.5 Feature Control Symbols.

### MODEL FILE

Once the assembly tolerance model is complete, a neutral file is created containing the information needed for assembly tolerance analysis. This neutral file, used by the analysis software package CATS.BYU, generates a set of algebraic equations to determine the values of the dependent variables. The dependent variables include both the kinematic and assembly variables.

### Analysis

Once vector equations are derived from the geometric model the next step in the analysis process is to linearize the equations for small variations about the nominal by Taylor's series expansion, retaining first order derivatives. Each derivative is evaluated using the nominal dimensions of the dependent and independent variables. The component tolerances  $dx_i$  are substituted and the system of linearized equations are solved for the corresponding variation in the kinematic variables and resultant assembly dimensions.

The linearized closed loop equations may be written as [Marler, 1988]:

$$dh_x(x_i, u_j, \alpha_k) = \sum \left( \frac{h_x}{x_i} dx_i \right) + \sum \left( \frac{h_x}{u_j} du_j \right) + \sum \left( \frac{h_x}{k} d\alpha_k \right) = 0$$

$$dh_y(x_i, u_j, \alpha_k) = \sum \left( \frac{h_y}{x_i} dx_i \right) + \sum \left( \frac{h_y}{u_j} du_j \right) + \sum \left( \frac{h_y}{k} d\alpha_k \right) = 0$$

$$dh_\theta(x_i, u_j, \alpha_k) = \sum \left( \frac{h_\theta}{x_i} dx_i \right) + \sum \left( \frac{h_\theta}{u_j} du_j \right) + \sum \left( \frac{h_\theta}{k} d\alpha_k \right) = 0$$

where  $dx_i$  are the specified tolerances on the independent dimensions,  $du_j$  are the resultant variations in the dependent assembly dimensions and  $d\alpha_k$  are the form and feature variations. In matrix notation these equations become:

$$[\mathbf{A}] \{\mathbf{dx}\} + [\mathbf{B}] \{\mathbf{du}\} + [\mathbf{F}] \{\mathbf{da}\} = \{\mathbf{0}\} \quad \text{Closed Loop Equations}$$

where  $[\mathbf{A}]$  is the matrix of sensitivities to variations in the independent variables  $x_i$   
 $\{\mathbf{dx}\}$  is the vector of specified variations of the independent variables  $dx_i$   
 $[\mathbf{B}]$  is the matrix of sensitivities to variations in the independent variables  $u_j$   
 $\{\mathbf{du}\}$  is the vector of unknown variations of the dependent variables  $du_j$   
 $[\mathbf{F}]$  is the matrix of sensitivities to variations in the form variables  $\alpha_k$   
 $\{\mathbf{da}\}$  is the vector of specified variations of the form variables  $d\alpha_k$

The linearized open loop equations in matrix form may be expressed [Carr, 1992]:

$$\{\mathbf{dg}\} = [\mathbf{C}] \{\mathbf{dx}\} + [\mathbf{E}] \{\mathbf{du}\} + [\mathbf{G}] \{\mathbf{da}\} \quad \text{Open Loop Equations}$$

where  $\mathbf{C}$ ,  $\mathbf{E}$  and  $\mathbf{G}$  represent sensitivities to variations in  $dx_i$ ,  $du_j$  and  $d\alpha_k$ , respectively.

$\mathbf{dg}$  is the non-zero vector of specified assembly tolerance limits for gaps, rotations or other feature variations.

From the above closed and open loop equations, expressions for the predicted accumulation of variations may be generated. Therefore, once the engineering model is created, the equations are derived, dependent variables separated, sensitivities calculated, and expressions for tolerance accumulation computed.

CATS.BYU allows the designer to use several tolerance analysis options including [Chase, et al. 1987]:

- Worst case and statistical tolerance accumulation
- Propagation of variations, including kinematic, dimensional and form variations
- Percent rejects prediction
- Accounting for process mean shifts using the Motorola Six Sigma statistical model

Tolerance information which has been created on a CAD modeling system may be sent to CATS.BYU for analysis or CATS/PC, the microcomputer implementation of CATS.BYU.

CATS estimates tolerance accumulation or stack-up from one of the expressions in Table 2.1. Which one you use depends upon customer requirements, process data available and desired accuracy.

**Table 2.1.** Assembly Tolerance Accumulation Formulas.

<p><b>Worst Case</b></p> $dU = \sum \left  \frac{U}{x_i} \right  dx_i \leq T_{ASM}$	<p>Assures 100% assembly acceptance if all parts are within specification.</p> <p>Costly design model. Requires excessively tight component tolerances</p>
<p><b>Root Sum Square</b></p> $dU = \sqrt{\sum \left( \frac{U}{x_i} \right)^2 dx_i^2} \leq T_{ASM}$	<p>Assumes normal distribution and <math>\pm 3\sigma</math> tolerances. Some fraction of assemblies will not meet specification.</p> <p>Less costly. Permits looser component tolerances.</p>
<p><b>General Root Sum Square</b></p> $dU = Z_{ASM} \sqrt{\sum \left( \frac{U}{x_i} \right)^2 \left( \frac{dx_i}{Z_i} \right)^2} \leq T_{ASM}$	<p>More versatile.</p> <p>May adjust <math>Z_{ASM}</math> to obtain desired quality.</p>
<p><b>Six Sigma</b></p> $dU = Z_{ASM} \sqrt{\sum \left( \frac{U}{x_i} \right)^2 \left( \frac{dx_i}{3Cpk_i} \right)^2} \leq T_{ASM}$	<p>Most realistic estimates.</p> <p>Accounts for process mean shifts and their long-term affects on assembly distribution.</p>

In the above table, dU is the predicted variation in the resultant assembly sum dimension, dx<sub>i</sub> is the component dimensional variation, U/ x<sub>i</sub> is the sensitivity that a variation in dx<sub>i</sub> has on U, T<sub>ASM</sub> is the design limit for variations in dU. Z<sub>ASM</sub> and Z<sub>i</sub> are

the number of standard deviations corresponding to the assembly and component tolerance limits, respectively,  $C_{pk_i}$  is the process capability index and is a measure of the shift in the process mean [Chase et al., 1992].

As presented above, previous modeling techniques included the use of kinematic joints, feature controls and vector loops. Modeling capabilities not fully integrated into the earlier modeling techniques, but developed and implemented through this research include:

- A comprehensive system for assembly or design specifications

Just as feature controls apply to individual parts, assembly tolerance specifications should be applied to assemblies of parts.

- Manual process variations or fastener clearances

These processes are not expressible as constrained kinematic models.

- Automatic open loop generation

Open vector loops are often necessary to represent design constraints on the assembly.

- Degree of Freedom analysis procedures

Only a limited number of redundancy checks are present. A method for detecting degree of freedom redundancy is essential to creating a correct engineering model.

## **CHAPTER 3: DESIGN SPECIFICATIONS**

---

Tolerances may be specified on dimensions or features of a production part to establish limits to expected manufacturing variations. Tolerances may also be specified to assemblies of parts. Both tolerances are important for the quality and functionality of an assembly, but differ greatly in meaning and application.

Component tolerances are applied to individual part dimensions. The precision specified influences the production processes and tooling selected to make the part. It also affects the choice of inspection methods used to accept or reject the parts as they are produced. Component tolerances are inspected before assembly.

Form tolerances are also applied to individual components of a part. Form tolerances are limits on the shape, orientation or location of the features on a part. Feature controls or form tolerances are applied to parts to constrain the variation due to different production methods to acceptable limits. For example, a circle may not be a true circle, but it may be considered circular enough for its purpose. Form tolerances are inspected before assembly to assure quality of the part.

Assembly tolerances are applied to the overall or resultant dimensions of an assembly. The resultant dimension of the assembly is generally a function of two or more parts in the assembly. Thus, component tolerances may accumulate or stack up, causing increased variation in the resultant assembly dimensions. Assembly tolerances are inspected after parts are drawn from stock and assembled.

Assembly tolerances are applied in order to meet engineering requirements and insure functionality of the final part. Component tolerances are only required when they contribute to assembly variation. Tolerance analysis relates the component tolerances to the assembly tolerances. By this means, the assembly limits determine the allowable variation in the components.

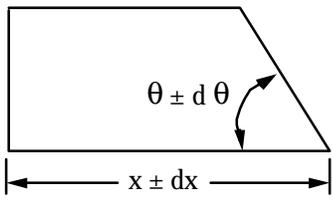
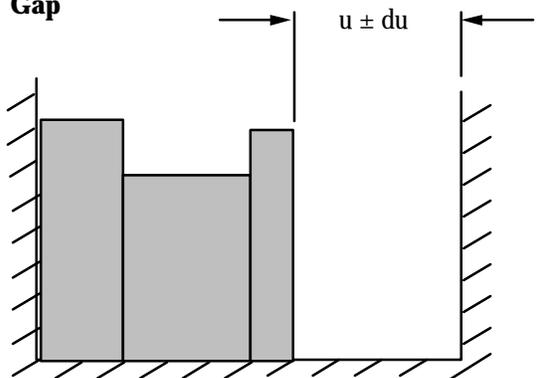
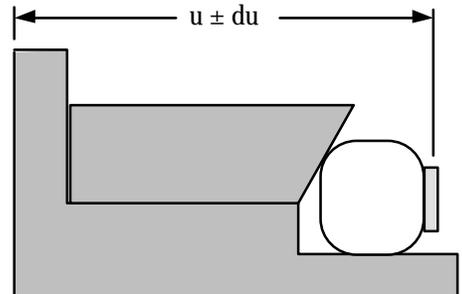
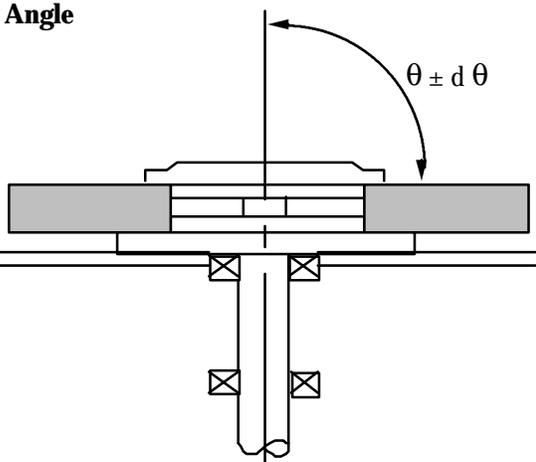
Assembly tolerances are an important design tool for engineers. A variety of tolerance specifications are needed in practice. However, there is no accepted standard which covers all design situations. The geometric tolerancing standard, ANSI Y14.5 is a comprehensive system for specifying tolerances, but it applies to individual components or to simple assemblies. This chapter will develop a similar system of design specifications

suitable for complex assemblies. The modeling procedures and a short example will also follow.

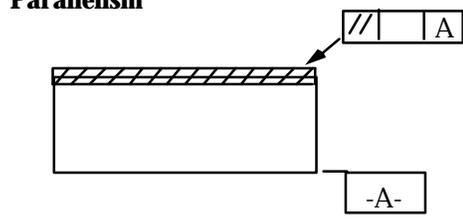
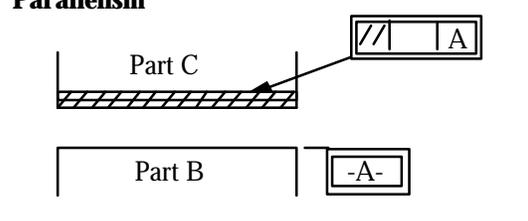
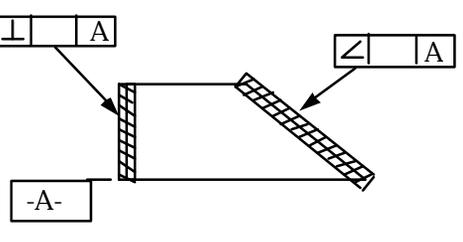
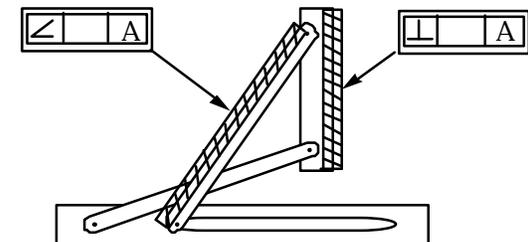
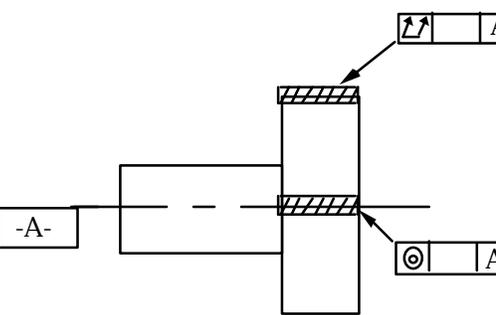
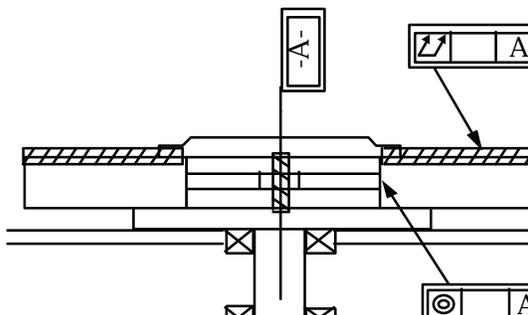
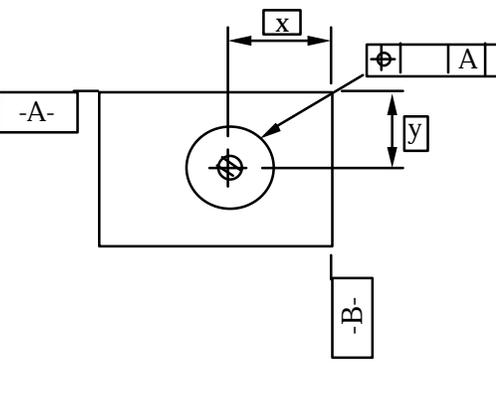
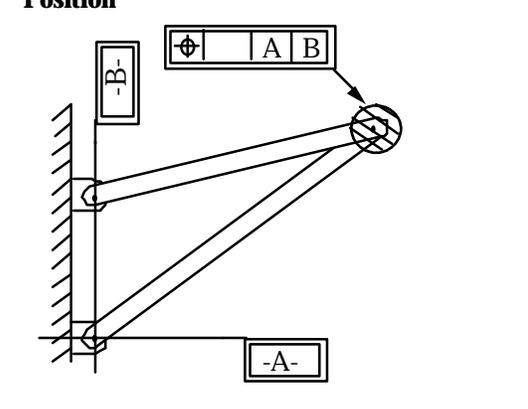
### **Design Specifications**

Examination of current engineering practice reveals that assembly tolerances or design specifications are analogous to form tolerances applied to individual components. For every ANSI Y14.5 form tolerance which specifies a datum, there is an assembly tolerance counterpart. Table 3.1 and 3.2 following relates current feature controls applied to components to similar feature controls applied to assembly tolerances. Together, they form a set of assembly tolerance specifications as comprehensive as ANSI Y14.5.

**Table 3.1.** Dimensional tolerance specifications.

Component Tolerances	Assembly Tolerances
<p><b>Length &amp; Angle</b></p>  <p>A trapezoidal component is shown with a horizontal dimension line below it labeled <math>x \pm dx</math>. An angle dimension line is shown at the right-hand slanted edge, labeled <math>\theta \pm d\theta</math>.</p>	<p><b>Gap</b></p>  <p>A cross-sectional view shows two grey blocks of different heights on a hatched base. A gap is formed between them. A dimension line above the gap is labeled <math>u \pm du</math>.</p> <p><b>Length</b></p>  <p>A cross-sectional view shows a grey component with a hole. A dimension line above the component is labeled <math>u \pm du</math>.</p> <p><b>Angle</b></p>  <p>A cross-sectional view of a bolted assembly. A vertical line indicates the intended axis. An angle dimension line is shown between this vertical line and the top surface of the assembly, labeled <math>\theta \pm d\theta</math>.</p>

**Table 3.2.** Form, orientation and location tolerances specifications.

Component Tolerances	Assembly Tolerances
<p><b>Parallelism</b></p> 	<p><b>Parallelism</b></p> 
<p><b>Perpendicularity &amp; Angularity</b></p> 	<p><b>Perpendicularity &amp; Angularity</b></p> 
<p><b>Concentricity &amp; Runout</b></p> 	<p><b>Concentricity &amp; Runout</b></p> 
<p><b>Position</b></p> 	<p><b>Position</b></p> 

From the previous tables there are two assembly tolerances or design specifications which are applied to kinematic variables in closed loops, the rest require open loops to be generated. The following table distinguishes the closed and open loop design specifications.

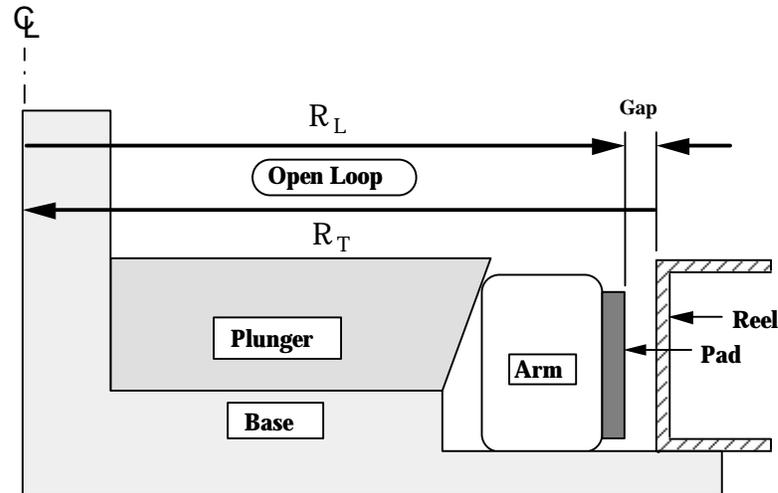
**Table 3.3.** Design specifications according to required loop type.

Closed Loop	Open Loop
Dependent Length Dependent Angle	Gap Global Orientation Parallelism and Perpendicularity Relative Angularity Position

A detailed description of each of the seven design specifications is presented below.

### **Clearance or Gap**

A gap specification is the allowed variation in the minimum distance between two parts. It is measured after all parts have been pushed together, leaving the gap as the only clearance remaining. Tolerance limits are set on the maximum and minimum values of the gap. The gap specification, which requires an open loop, is a representation of the ANSI Y14.5 length tolerance. Clearances or interference fits are also assigned tolerances in this manner. For example, in order for a fan to operate properly a certain clearance from the blades must be maintained. Another example is that of a locking computer tape hub (see figure 3.1). The plunger is rigidly set against the base, forcing the arm to slide radially outward. An interference fit between the pad and the reel must be assured (a clearance is shown for clarity). An interference gap specification is assigned between the pad and the reel.



**Figure 3.1.** Locking computer tape hub with gap specification and open loop.

A tolerance is placed on the interference distance between the two parts. An open loop is then created from the end points of the gap specification. The loop consists of two vectors,  $R_T$  and  $R_L$  forming a chain from one side of the gap to the other. The gap will then be the result of the overall dimension  $R_T$  and the 2-D resultant stack-up  $R_L$ :

$$\text{Gap} = R_T - R_L$$

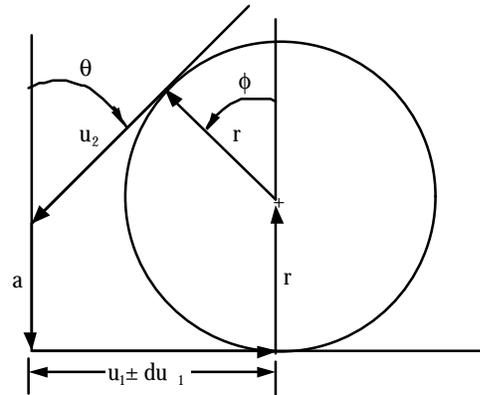
The analysis will then determine whether the gap variation will meet the desired specifications.

The next two design specifications are assembly tolerances on kinematic variables. No open loops are required since they are already included in a closed vector loop as kinematic variables. The two specifications are dependent length and dependent angle.

### Dependent Length

A dependent length specification constrains the variation of a point's location between two mating parts. It is similar to a length component tolerance. Almost anytime one has a contact with a sliding plane a length specification may be applied. For example, a planar, cylindrical slider or edge slider kinematic joint will contain a sliding plane. A dependent length is a tolerance applied to a kinematic variable in a closed loop, it describes a single kinematic degree of freedom. An example of a critical length would be the vector  $u_1$  in the assembly of figure 3.2. Vectors  $u_1$ ,  $u_2$ , and  $f$  are dependent variables. They are therefore, functions of  $r$ ,  $a$  and  $q$ . A length specification could be placed on any of the

dependent variables. A specification placed on vector  $u_1$  would constrain the vector to the length of  $u_1 \pm du_1$

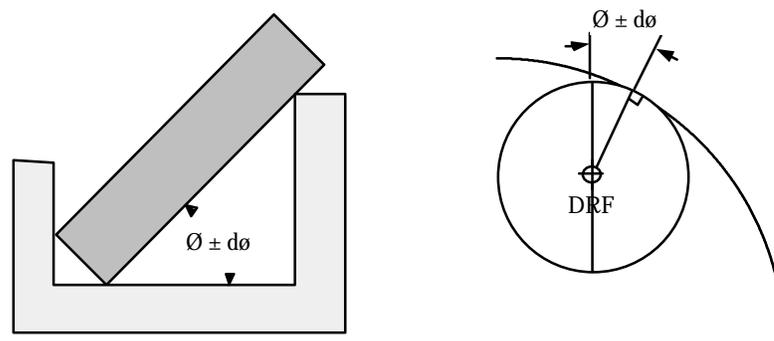


**Figure 3.2.** Dependent length specification applied to vector  $u_1$ .

Once analysis is complete, a check is run to see if the kinematic variable is within the desired tolerance range. If the variable is within specification the mechanism should function properly.

### Dependent Angle

A dependent angle specification, comparable to an angle component tolerance, is the angular variation between two mating parts, or it is the dependent angle measured from a cylindrical datum reference frame and a contact point. It describes a single kinematic degree of freedom (see figure 3.3).



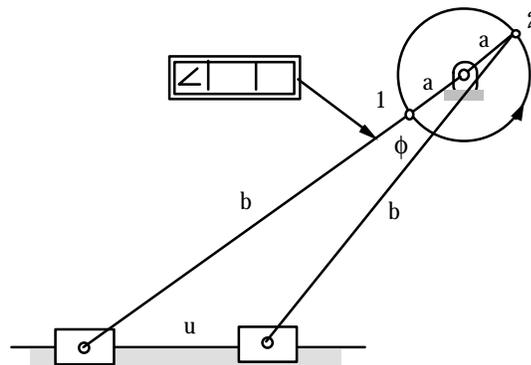
**Figure 3.3.** Mating part and cylindrical datum relative angle specification.

For example, an offset slider crank mechanism shown in figure 3.4. The crank  $a$  rotates at a constant speed. The connecting rod  $b$  pulls the slider back and forth on the track  $u$ . As

the crank rotates counterclockwise from position 1 to position 2 the connecting rod pulls the slider from 1 to 2 on the track. This is the forward or power stroke. The return stroke occurs as the crank continues to rotate past position 2 and back to 1. Such reciprocating mechanisms are often used in shavers, clippers, and feed mechanisms. A long forward stroke allows smooth transmission of power. The unloaded return stroke is shortened for efficiency. The timing ratio is calculated to be:

$$TR = \frac{180 + \phi}{180 - \phi}$$

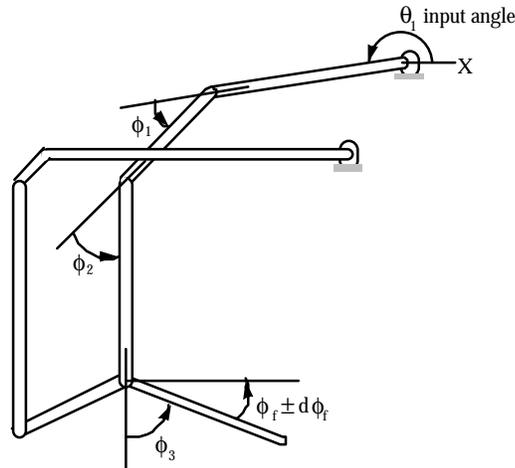
A dependent angle specification is applied to  $\phi$  in order to maintain a proper timing ratio.



**Figure 3.4.** Offset slider crank mechanism.

### Global Orientation

Global orientation is the angular variation between an axis or edge of a part and the global X-axis. This open loop specification is analogous to the ANSI Y14.5 orientation feature tolerance. For example, the remote positioner linkage system may contain a global orientation specification (see figure 3.5). This specification will require the angle of the final link to be within a certain tolerance from the global X-axis.



**Figure 3.5.** Remote positioner with global orientation and rotation loop.

$q_1$  represents a specified input angle. The open loop is created using the global coordinates origin as the starting node and the ending linkage as the end point.  $f_1$ ,  $f_2$ , and  $f_3$  represent relative angles between consecutive vectors.  $f_f$  is the final angle relative to the global X-axis. The rotation loop will be:

$$q_1 + f_1 + f_2 + f_3 + f_f = 0$$

Solving the rotational equation using the worst case and statistical methods result in:

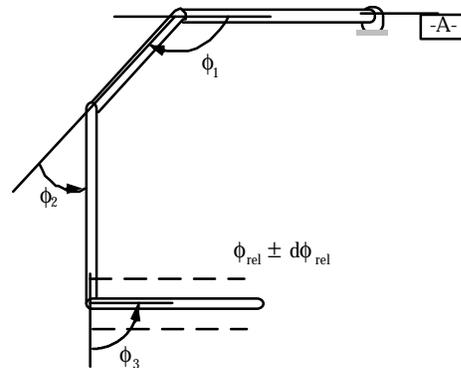
$$df_f = dq_1 + df_1 + df_2 + df_3 \quad (\text{WC})$$

$$d\phi_f = \sqrt{d\theta_1^2 + d\phi_1^2 + d\phi_2^2 + d\phi_3^2} \quad (\text{RSS})$$

$df_f$  is then applied to the nominal to see if the variation is within the specification limits.

### Parallelism and Perpendicularity

A parallelism or perpendicularity specification limits the deviation from parallel or perpendicular between two parts, either axes or edges. These specifications are analogous to the parallelism or perpendicularity ANSI Y14.5 for tolerance bands placed on features of parts. Once the two parts are established to be parallel or perpendicular nominally, a tolerance band is placed on the parallelism or perpendicularity of the assembly (see figure 3.6).



**Figure 3.6.** Remote positioner with parallelism specification and rotation loop.

Since the parts are considered parallel, an open loop will contain one less rotation than if the assembly were modeled as relative angularity. This is deduced from the equation:

$$-f_1 + f_2 + f_3 - f_{\text{rel}} = 0$$

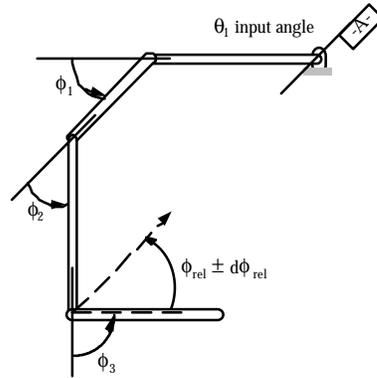
where  $f_{\text{rel}}$  is considered to nominally be  $0^\circ$  with no input angle. Solving the rotational equation using the worst case and statistical methods result in:

$$\begin{aligned} df_{\text{rel}} &= df_1 + df_2 + df_3 \quad (\text{WC}) \\ d\phi_{\text{rel}} &= \sqrt{d\phi_1^2 + d\phi_2^2 + d\phi_3^2} \quad (\text{RSS}) \end{aligned}$$

For the designer's convenience the open loop created will be identical to that of global orientation. Since the open loop is flagged as parallelism, the analysis side will account for the extra node in the loop. The variation is then compared to the parallelism design requirement to see if the linkage functions properly.

### Relative Angle

The relative angle specification is much like the ANSI Y14.5 angularity tolerance only a relative angle specification is the angular variation between two non-mating parts. This design specification allows the designer to correlate an input angle to an output angle of a mechanism. It is therefore a more general case of both global orientation and parallelism or perpendicularity where the nominal input angle is either relative to the ground,  $0^\circ$  or  $90^\circ$  respectively. For example, a linkage may require that the input to output angle ratio be one-to-one. Assigning a relative angle specification between the input linkage and the output linkage will assure that the engineering requirements will be met (see figure 3.7).



**Figure 3.7.** Remote positioner with angularity specification and rotation loop.

As shown above, all angles from the input to the final linkage are required for analysis. The resultant relative angle is determined by summing all the relative angles of the intermediate parts. The open loop will obtain the needed rotation equation:

$$q_1 + f_1 + f_2 + f_3 - f_{rel} = 0$$

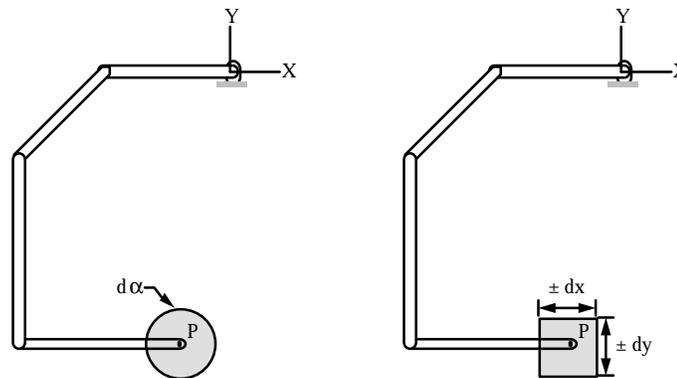
where  $f_{rel}$  is considered to nominally be  $0^\circ$  with no input angle. The worst case and statistical variation is then solved for and applied to the relative angle specification:

$$df_{rel} = dq_1 + df_1 + df_2 + df_3 \quad (\text{WC})$$

$$d\phi_{rel} = \sqrt{d\theta_1^2 + d\phi_1^2 + d\phi_2^2 + d\phi_3^2} \quad (\text{RSS})$$

**Position**

A position specification places limits on the location of a point in the assembly just as an ANSI Y14.5 position tolerance constrains a point on a part. The point location may vary horizontally and vertically (x, y). By constraining a point on a linkage, for example point P in figure 3.8, the path of the linkage system is guaranteed.



a) diameter tolerancing      b) rectangular tolerancing

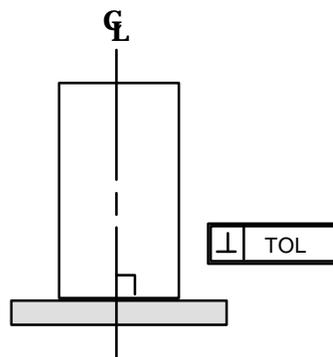
**Figure 3.8.** Linkage system with constrained point in assembly.

In order to analyze the position specification an open loop, starting on one part and ending on another, must be specified. The two end points are datums which are each associated with a single part, unlike joints which are associated with two parts. One end point need not be the global reference frame.

A tolerance is then placed on the desired point relative to the assembly origin as either a diameter tolerance or rectangular tolerance. In rectangular tolerancing, the x and y coordinates of a point are specified relative to a set of datum axes. Many object to rectangular tolerancing because the permissible position error varies with direction. It is equal to dx in the x direction, dy in the y direction, and  $\sqrt{dx^2+dy^2}$  in the diagonal direction of the rectangular tolerance zone. ANSI Y14.5 has established a non-directional tolerance represented by a circular tolerance zone. No tolerances are applied to the x and y dimensions in this case. The value of the tolerance specifies the diameter of the tolerance zone, da.

From the open loop, equations may be generated which will allow the designer to better predict if the assembled mechanism meets the desired specifications.

The design specifications modeled for two-dimensions do not include concentricity or runout assembly tolerances. Although runout is easier to measure, perpendicularity may also be used for cylinder faces. By creating a perpendicular specification between the axis of one part and an edge of another part which is perpendicular, the equivalent assembly tolerance is applied (see figure 3.9). A concentricity design specification may be modeled as a gap or clearance in 2-dimensions.



**Figure 3.9.** Perpendicularity between cylinder face and center line axis.

## **Summary**

Assembly tolerances are important design tools for engineers. By incorporating the six design specifications, or assembly tolerances developed into tolerance analysis, a more realistic solution may be provided. This allows for engineers to place constraints on assemblies to insure functionality.

## **CHAPTER 4: OPEN LOOP GENERATION**

---

The previous chapter presented a new system for specifying limits on assembly variations. By assembly tolerance analysis the designer may predict the affects of variation on design requirements. Open loops, discussed in this chapter, are required to generate the equations necessary to solve for many of these variations.

The end product of the assembly tolerance modeling process is a set of vector loops which describe the assembly relationships in terms of geometric and kinematic information (see Chp 2 sec Loops). These loops may be either open or closed loops.

*Closed loops* start and end at the same location and represent kinematic constraints on the assembly.

*Open loops* are used to determine assembly resultants of interest, such as a clearance, orientation or position. They are used to model the engineering constraints on an assembly

There must be enough closed loops to solve for all the dependent kinematic variables. There must also be enough open loops to solve for all other assembly resultants of interest.

Creating the vector loops is a major portion of the engineering modeling procedure. There are many possible loops in an assembly, but only a few are required for analysis. Not only that, but the loops, whether open or closed, are not unique since each loop may follow a variety of possible paths. These paths may be difficult to define since the choice of loop paths is up to the designer. Below is a brief overview of the existing closed loop theory.

### **Manual Closed Loop Creation**

When creating vector loops several rules must be followed. By following these rules, enough equations may be generated to solve for dependent kinematic variables and sensitivities. Three beginning modeling rules include:

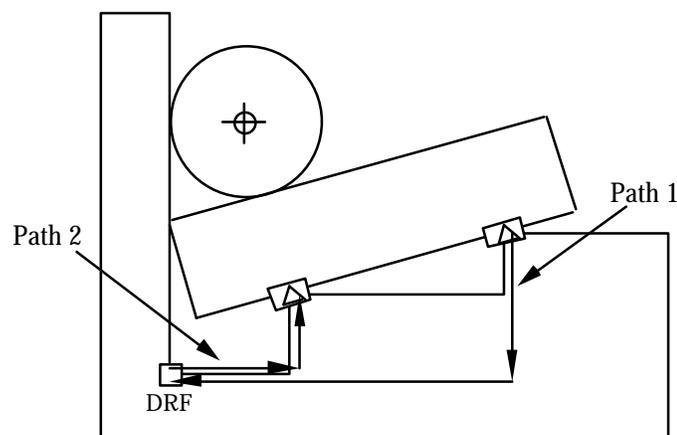
- The set of loops must pass through every part and every joint in the assembly.

- No single loop may pass through a part or joint more than once, but it may start and end at the same point.
- There must be enough loops to solve for all the kinematic variables -- one loop for every three variables.

The vector loop passes from mating part to mating part, always crossing through the joints that connect the parts. In a similar fashion, the vector loop enters a part through one joint (using an incoming vector) and leaves through another joint (using an outgoing vector), thus crossing the part. The vector loop always follows paths along associated datum paths created with each joint.

While creating vector paths the following three modeling rules apply:

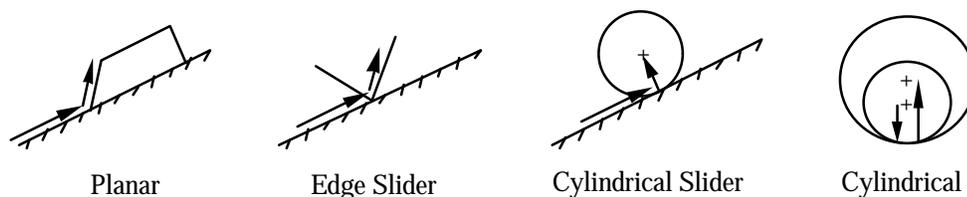
- A vector loop must pass from one part to the next mating part through a common joint.
- The path across a part must pass through a Datum Reference Frame (DRF), following the datum path from the incoming joint to the DRF and then following the datum path from the outgoing joint to the DRF, only in reverse.
- If the path across the part doubles back on itself with an equal and opposite vector, the two vectors cancel each other and will be omitted from the loop.



**Figure 4.1.** Datum paths define the path across a part.

Different joints along the vector paths place requirements on the vector loop. These requirements assure the angles and vectors corresponding to the kinematic variables will be present in the assembly, so their variations can be calculated.

- For a cylindrical slider joint, either the incoming or outgoing vector must be normal to the sliding plane and end at the center of the cylinder.
- For joints having a sliding plane (planar, cylindrical slider or edge slider), either the incoming or outgoing vector must lie in the sliding plane.
- For cylindrical joints (parallel cylinders in contact), the path through the joint must start at the center of one cylinder and end at the center of the mating cylinder, passing through the contact point.



**Figure 4.2.** Required vector paths through joints.

### Manual Open Loop Creation

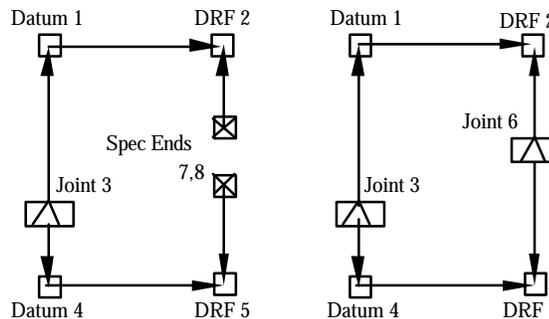
A similar theory for manual open loop creation will now be developed. Many of the same closed loop rules apply for open loop generation, but, the differences are important. Open loops describe engineering constraints on our assembly. It is therefore possible to model an entire system without any open loops. This is one major difference between open and closed vector loops.

Open loops must pass through at least two parts since design specifications are applied between parts. The parts, whether two or more, contact in some manner, therefore, open loops also require at least one joint. Since open loops are not closed, the starting and ending points must be different to create an open loop. Also, assembly tolerances determine the variation of the entire assembly which contribute to the design specifications. This requires there to be an open loop for each specification. The general rules for open loops are summarized as:

- The loop must pass through at least 2 parts and 1 joint.
- A single loop may not start and end at the same point.
- There must be a maximum of one open loop for every design specification.

It should be noted that the open loop does not have to pass through every part or every joint. Also, depending on the number of assembly constraints, or assembly tolerances, there may be any number of open loops.

Vector paths are created in the same manner as closed loops, the only difference is that the end points of specifications are rectangular datums rather than kinematic joints. Each endpoint is associated with one part and has one datum path. This allows the vector loops to remain open loops. For example, Joint 6 (figure 4.3) requires two datum paths closing the loop from DRF 2 to DRF 5. On the other hand, Specification endpoints 7 and 8 only have one datum path, there is therefore only one vector leaving each endpoint and the loop may not be closed.



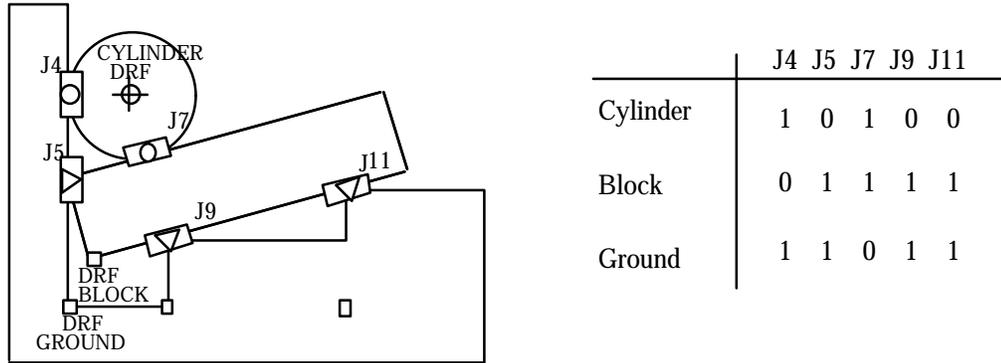
**Figure 4.3.** Open loop specification datum paths versus closed loop datum paths.

Open vector loops also have the same joint requirements as closed loops.

By automating these procedures, not only will the number of errors be reduced, but also the modeling time. Much of the existing theory for loop automation, the modified Automatic Vector Loop (AVL) algorithm, developed by Larsen and Simmons [Larsen, 1991; Simmons, 1990] has been extended for use in open loop generation. The modified AVL theory will be discussed next.

### **Modified AVL**

The AVL algorithm begins by creating a connectivity matrix. A connectivity or incidence matrix [Larsen, 1991] is a matrix used to map the relationships between parts and joints. By convention, matrix rows represent different parts in the assembly and columns represent individual joints. Each joint touches two parts, so two 1's are placed in each joint column for the two corresponding parts. 0's fill the rest of the column.



**Figure 4.4.** Connectivity matrix and corresponding assembly.

For example, in figure 4.4, the connectivity matrix indicates that joint 4 connects the Cylinder to the Ground and the Block contacts the Ground at joints 5, 9 and 11.

A set of loops is obtained by moving between the 1's in the matrix. A horizontal path between 1's represents crossing a part, while vertical paths between 1's represent transferring between parts through joints.

The number of loops required for tolerance analysis [Larsen, 1991] is:

$$\text{Loops} = \text{Joints} - \text{Parts} + 1$$

For the assembly in figure 4.4:

$$L = 5 - 3 + 1 = 3 \text{ loops}$$

#### VIRTUAL TREE REPRESENTATION

Once the connectivity matrix has been constructed, candidate paths are found and compared. An exhaustive search must be made for all possible loop candidates and a selection made of an optimum set, as determined by a set of criteria to be described later.

To obtain a candidate closed loop a starting joint and part are chosen from the matrix. A candidate loop is obtained when the tree search completes a loop by returning to its starting joint. A candidate closed loop is shown for the assembly of figure 4.4:

	J4	J5	J7	J9	J11
Cylinder	1	0	1	0	0
Block	0	1	1	1	1
Ground	1	1	0	1	1

**Figure 4.5.** Closed path representation in a connectivity matrix.

To find the next candidate loop, the path is traversed until reaching a place where an alternate path may be explored. The downward search is then continued until the path doubles back on itself, or arrives at the starting joint. This step is iterated until all possible loops have been found [Larsen, 1991]. To keep track of the loops found the 1 elements along the path are changed to -1's. Thus, the -1 paths are visible in the matrix.

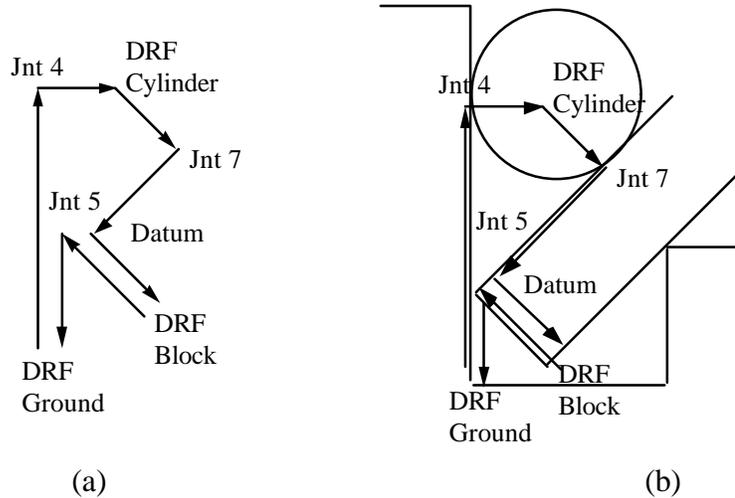
For example, in figure 4.6, the Block has four joints, indicated by the four ones in its row. The path enters the block through joint 7 (the incoming joint). There are three possible outgoing joints (joints 5, 9 and 11). The first loop exited through joint 9 in figure 4.5. the alternate loop in figure 4.6 exits through joint 5. An additional candidate loop is possible, using joint 11 as the outgoing joint from the Block.

	J4	J5	J7	J9	J11
Cylinder	1	0	1	0	0
Block	0	1	1	1	1
Ground	1	1	0	1	1

**Figure 4.6.** Alternate closed path in the connectivity matrix.

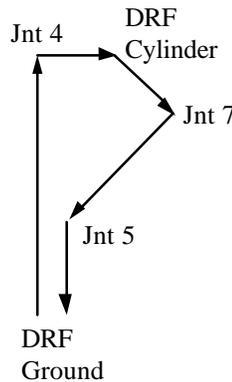
Candidate loops, once found, are then expanded to include not only the different joints and parts, but also the paths across the parts. This ensures that the vector loop consists of only controlled and kinematic dimensions. The datum paths, or paths from the joint to the DRF of each touching part, are now included in the vector loop and a complete candidate loop is defined [Larsen, 1991].

The loop found in the connectivity matrix of figure 4.6 can be expanded and constructed as a graphical overlay on the assembly drawings, as shown in figure 4.7. Overlaying this diagram on the assembly reveals the finished vector path through the assembly.



**Figure 4.7.** Expanded loop with joints, parts and datums (from figure 4.4)  
 (a) vector loop and (b) overlaid on assembly.

A check must be made for redundant dimensions. For example, if the two datum paths overlap, they share common dimensions on the paths back to the DRF. For example, the datum path in figure 4.7 from joint 5 to the Block DRF and back is redundant. These redundant dimensions are not needed for the tolerance analysis and may introduce unrealistic variation into the tolerance model. The redundant dimensions are removed from the candidate loop until a complete candidate loop is defined [Larsen, 1991].



**Figure 4.8.** Completed candidate loop (from figure 4.4).

SELECTION OF LOOPS

A criteria has been established for comparing candidate loops. The first criterion is the number of nodes in the loop. The fewer the nodes, the better the loop. Candidate loops are compared with the current best loop for that starting joint. Thus, if the candidate loop has fewer nodes than the current best loop, the candidate loop becomes the current best loop.

The next criterion is physical length. If both loops being compared have the same number of nodes, then the expanded length of the loops are compared. The loop which is shortest dimensionally becomes the current best loop. The current best loop is stored and the process of obtaining candidate loops continues until all possible loop candidates which begin at the selected starting point have been tested.

A new starting point for the next closed loop is chosen from a joint which was not used in the first loop. If all the joints have been used in a loop, then the loop will start at a joint which was not a starting joint for any other loop. This searching process continues until the required number of loops is determined using the formula previously stated as [Larsen, 1991].:

$$\text{Loops} = \text{Joints} - \text{Parts} + 1$$

This is an exhaustive search. The number of possible candidate loops is:

$$NT = \prod_{i=1}^n (N_i - 1)$$

where  $N_i$  is the number of joints on part  $i$   
 $n$  is the number of parts  
 $NT$  is the total number of loops possible.

For the assembly in figure 4.4 the equation becomes:

$$NT = (2-1)(4-1)(4-1) = 9 \text{ loops possible}$$

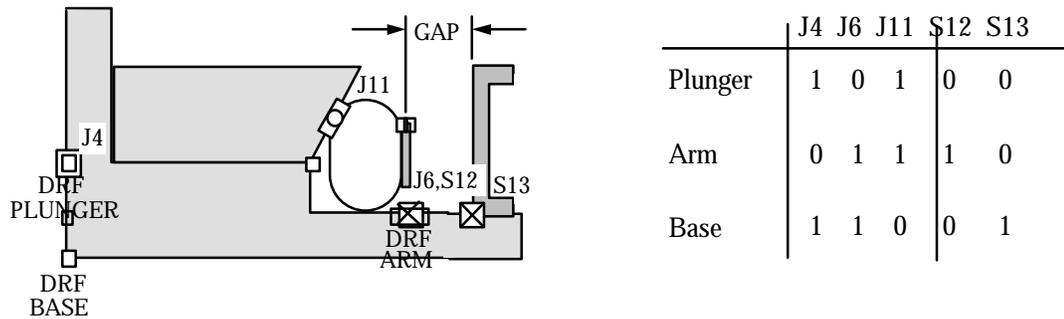
Now that the closed loop AVL theory has been reviewed, a similar development for the AVL open loop theory will be presented.

### **Modified AVL for Open Loops**

Candidate loops are found and compared much the same as in the modified AVL algorithm. On the other hand, starting and ending points are restricted for each open loop. Open loops must start at one endpoint of a specification and end at the other

corresponding endpoint. It is also true that open loops do not have to pass through every part, nor every joint of an assembly from the development of the manual loop creation.

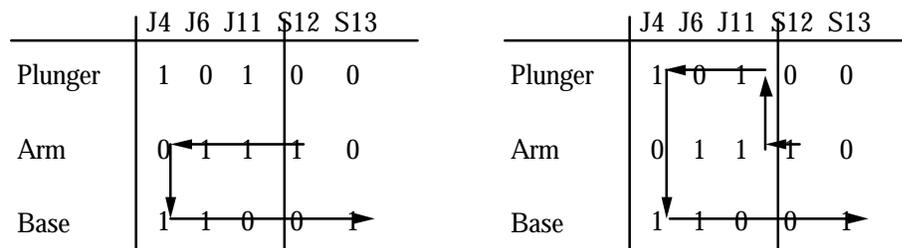
Defining open loops is more restrictive than closed loops, but many possibilities for open loop paths still exist. Using the modified connectivity matrix the same algorithm may be performed with modification. To obtain the modified connectivity matrix the matrix is altered to allow design specification endpoints to be incorporated into the matrix. Specifications (the end points) are placed in columns appended to the original connectivity matrix.



**Figure 4.9.** Assembly with modified connectivity matrix.

Each specification end point is associated with a single part. A 1 is placed in the respective part's row. So, for the assembly in figure 4.9, endpoint 12 is associated with the arm and endpoint 13 is associated with the base. 0's fill the rest of the column.

Only one set of specification endpoints are appended to the connectivity matrix at a time. This ensures that only one design specification will be involved in any one loop at a time. This is important in solving for the engineering constraints placed on the assembly's function. Therefore, for every design specification, a new connectivity matrix will be created. Once the loop is found for the specification, the specification columns may be removed from the connectivity matrix and a new set may be added for another specification.



**Figure 4.10.** Alternative open loop candidates.

The open loops are tested for node number and physical length to find the best loop (figure 4.10). The loops are also expanded and checked for redundant dimensions just as closed loops. Once a loop has been found for the specification the connectivity matrix is modified for the next specification and so on, until there is a corresponding open loop for each design specification. The number of required loops for both open and closed loops is defined as:

$$\text{Loops} = \text{Joints} - \text{Parts} + 1 + (\text{Spec end points}) / 2$$

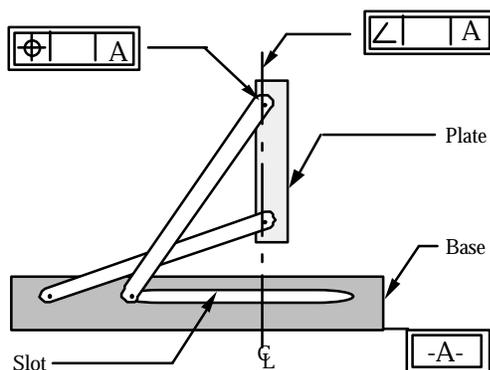
Where

Loops =	number of loops required to perform tolerance analysis
Joints =	number of non-redundant contact points in the assembly
Parts =	number of parts or DRFs in the assembly
Spec end points =	number of assembly tolerance endpoints.

Additional tests are required to remove redundant joints. Redundant joints are joints which create an over-determined system. These will be discussed in greater detail in chapter 6.

### Multiple Assembly Constraints

When modeling open loops it may be possible to have two assembly tolerance specifications with the same endpoints. A common application occurs when we want to know the position and orientation of a point on an assembly (see figure 4.11).



**Figure 4.11.** Position and relative angle specifications with two common endpoints.

There arise several instances where two specifications may be determined from one open loop. Any valid vector loop will have associated with it one vector loop equation with

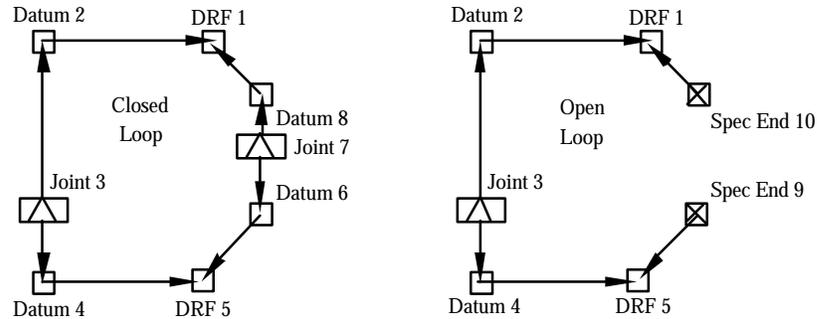
three scalar equations corresponding to the X, Y, and rotational components. A position assembly tolerance constrains the X and Y position of a point in an assembly relative to the global origin. The equations needed to solve for the sensitivities are the X and Y scalar components. Likewise, a relative angle specification constrains the final angular position of a link on an assembly relative to another part. To solve for the relative angle sensitivity the rotational equation is used. One may then use the same vector loop with the vector equation of X, Y, and rotation to solve for the two specifications, providing that the endpoints are identical. This is true for the combinations in table 4.1.

**Table 4.1.** Open Loop Combinations.

Design Specification	Combined with:
Gap	Global Orientation, Relative Angle, Parallelism, Position
Position	Global Orientation, Relative Angle, Parallelism, Gap

It should be noted that since a gap or clearance specification only involves constraints in one direction and a position tolerance may be transformed into a diameter tolerance, they too may be determined from the same equations.

Sometimes, the desired open loop is just part of an existing closed loop. If the two endpoints of a specification are elements of a closed loop then the open loop may be taken as part of the open loop and a new open loop does not need to be found. For analysis, the part of the closed loop associated with the open loop may be extracted (with associated tolerances) to solve for the open loop constraints. For example, the open loop of nodes [10, 1, 2, 3, 4, 9] is the same as part of the closed loop, specifically nodes [8, 1, 2, 3, 4, 5, 6]. The open loop may take the information already stored with the closed loop to create the associated equations.



**Figure 4.12.** Closed and open loops using same vector paths.

### Summary of New AVL Algorithm

The closed loop AVL algorithm has been modified and extended in order to automatically generate both open and closed loops. The steps for the New Automatic Vector Loop Algorithm are shown below (those steps with asterisks have been modified or added):

1. Set up connectivity matrix with parts and joints from actual assembly \*
2. Determine number of required loops \*
3. Select starting joint and part for first loop
4. Determine current joint's other mating part
5. Use virtual tree procedures until candidate loop is found
6. Expand contact joint loop using datum paths and form complete loop
7. Eliminate adjacent geometrically equivalent dimensions \*
8. Compare candidate loop with current best loop
9. Store current best loop from current starting joint
10. Select new starting joint
11. Repeat steps 4-10 until required number of closed loops determined
12. Append specification to connectivity matrix\*
13. Select starting node and part as first specification end point \*
14. Repeat steps 4-9 \*
15. Repeat 12-14 until required number of open loops determined. \*

## **CHAPTER 5: AutoCATS ENHANCEMENTS**

---

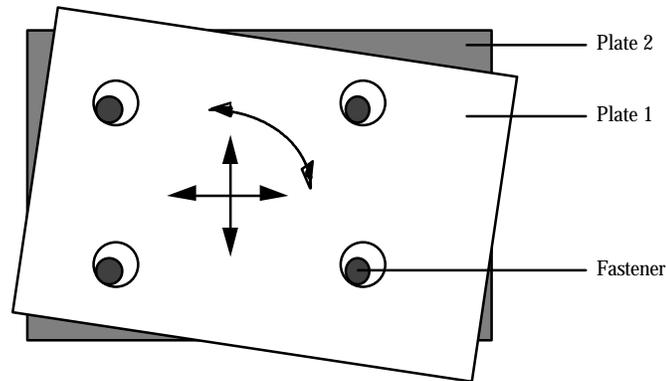
Several new analysis and user interface enhancements have been implemented in AutoCATS in order to model mechanical assemblies more accurately and create a robust environment. A new modeling element allows for the representation of manual processes. The DOF analysis has been extended to cover more than just instantaneous centers. Feature controls now include characteristic lengths. Many user interface modifications have also been made. These new features will be discussed below.

### **Manual Assembly Processes**

In most assemblies, fasteners are used to connect two parts. Depending on how the fasteners are inserted and tightened variation may occur in the assembly. These variations were not associated with the previous prototype kinematic analysis model system, yet they still introduce variations which propagate through the assembly. By adding a new model for manual process variations to the assembly, a more complete tolerance analysis can be performed.

Consider an assembly where two members are fastened together with a mating pattern of four holes, as shown in Figure 5.1. This is an assembly process which introduces variations which may not be determined by kinematic analysis. Deterministic kinematic variation requires constrained movement between parts. The variation associated with the fasteners is non-kinematic since it is not permitted to adjust to other variation sources. For most analysis, once parts are fastened together they are presumed to act as a single part.

Before tightening the fasteners, there are three kinematic degrees of freedom due to the clearance between the fasteners and the holes. The part is able to translate in two directions and rotate relative to plate 1. The magnitude of relative shift between them will depend upon the amount of clearance and the maximum distance between the holes.



**Figure 5.1.** Kinematic degrees of freedom for mating hole patterns.

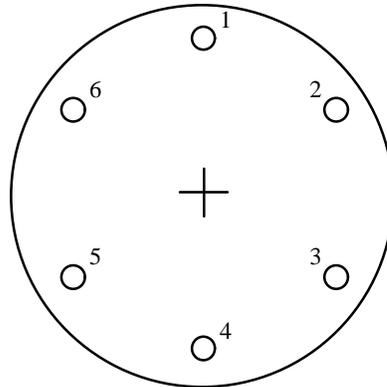
However, if this is a manual assembly operation, the operator controls the final location of the parts when the bolts are tightened. It can be anywhere within the range of the clearance. Thus, there is an additional source of variation due to the manual assembly process. This type of variation, called clearance variation, may accumulate and propagate in an assembly just the same as dimensional variations.

Clearance variation may be associated not only with fasteners, but any welding, bonding, or joining process. Depending upon the characteristics of the process, different variations will occur. The variation may be introduced by either human or robotic methods. For example, before welding together two parts the parts are aligned. Depending on how the two parts are aligned, in a holding, centering or other device, variation may occur between the two parts before they are welded.

If the operator always pushed and twisted a bolted assembly to an extreme position, with the bolts bearing against the sides of the holes, kinematic constraints could be applied. But it is more likely that the operator will locate the parts somewhere in-between the extremes, with more assemblies clustered at midrange. Thus, as a first approximation, we might assume a normal distribution for this assembly variation and define the extreme nominal values to be  $\pm 3\sigma$  limits. A more conservative estimate would be to assume a uniform distribution, that is, the final location is equally likely to be anywhere in the range. The nominal extremes would then correspond to  $\pm \sqrt{3}\sigma$  limits.

Manual process variations may thus be modeled by introducing independent kinematic variations with as many as three degrees of freedom, in the plane of contact between the two mating parts. Clearance variation permits relative motion between two parts similar to joints. Unlike joints, however, the location of the clearance variation is at the centroid

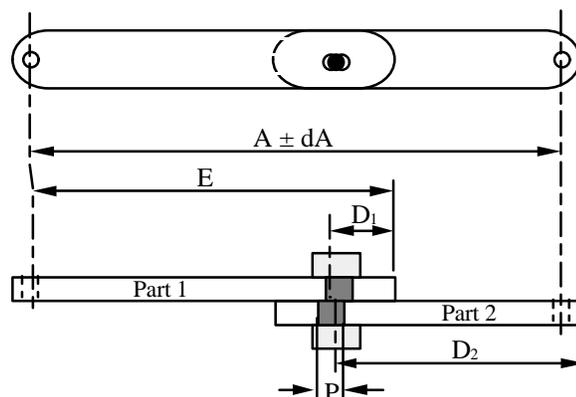
of the bolt hole pattern. Although the translational variation of the bolts may be applied anywhere in the vector loop, the rotational variation is best applied at the centroid of the hole pattern. This ensures the rotational variation will account for the extreme clearance of all the bolts simultaneously. For example, if the hole pattern contained six holes in a circular pattern (see figure 5.2) the clearance variation would be applied at the center of the circle. The loop would then pass through the centroid of the bolt hole pattern and not be required to pass through any bolt holes.



**Figure 5.2.** Top view of bolted members.

### SLIP JOINT

Another example of bolted members may better illustrate the mathematical model for the non-kinematic variation due to process variation. Suppose that two members are bolted together (figure 5.3). This assembly is called a slip joint. The clearance of the two mating holes around the bolt allows the length  $A$  of the assembly to vary at assembly time.



**Figure 5.3.** Top and side view of bolted members.

The total length of variation  $\pm dA$ , is the result of the accumulation of several component tolerances, including the basic locating dimensions, D1 and D2 for the holes, the two hole size tolerances and the bolt diameter. The assembly specifications are as follows:

D1:	1.33
D2:	3.740
Bonus Tolerances (B1, B2):	0.0005
True Position Tolerance for Part 1 (t1):	0.0015
True Position Tolerance for Part 2 (t2):	0.0020

	Maximum Dimension	Minimum Dimension
Clearance hole in Part 1 (c1):	0.3753	0.3748
Clearance hole in Part 2 (c2):	0.3816	0.3811
Pin size (P):	0.3748	0.3738

The equivalent symmetric tolerance for Part 1:

$$T1 = (t1 + B1 + c1 - P) / 2$$

$$= (0.0015 + 0.0005 + 0.3753 - 0.3738) / 2 = 0.00175$$

and for Part 2:

$$T2 = (t2 + B2 + c2 - P) / 2$$

$$= (0.0020 + 0.0005 + 0.3816 - 0.3738) / 2 = 0.00515$$

The average assembly dimension may be calculated from:

$$A = E - D1 + D2$$

$$= 2.280 - 1.330 + 3.740 = 4.690$$

The variation in A is:

$$dA = \sqrt{T_e^2 + T1^2 + T2^2}$$

$$= \sqrt{0.005^2 + 0.00175^2 + 0.0015^2}$$

$$= 0.00739$$

The resultant assembly dimension then becomes  $4.690 \pm 0.00739$ .

When these two parts are applied as a sub-component of an assembly the variation would be expressed as:

$$dH = \sqrt{\left(\frac{\partial H}{\partial A} dA\right)^2 + \dots} \quad \text{where H is the assembly function.}$$

which becomes:

$$dH = \sqrt{\left(\frac{\partial H}{\partial A} \left[ \left(\frac{\partial A}{\partial T_1} dT_1\right)^2 + \left(\frac{\partial A}{\partial T_2} dT_2\right)^2 + \left(\frac{\partial A}{\partial T_e} dT_e\right)^2 \right]\right)^2 + \dots}$$

when expanded to include all terms.

Substituting values, the variation found is the same:

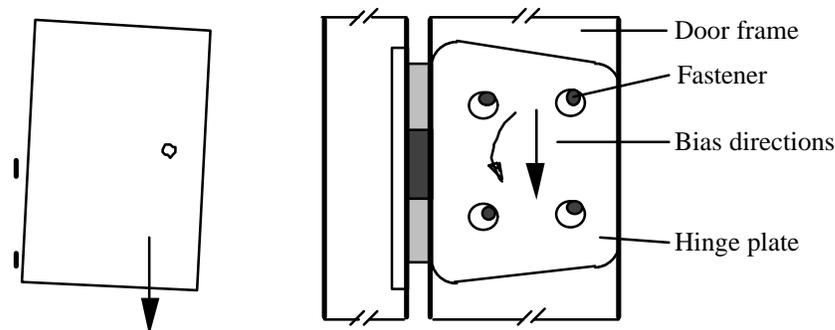
$$\begin{aligned} dH &= \sqrt{\left(\frac{\check{Z}H}{\check{Z}T_1}dT_1\right)^2 + \dots} \\ &= \sqrt{T_1^2 + T_2^2 + T_e^2} \\ &= \sqrt{0.005^2 + 0.00175^2 + 0.0015^2} \\ &= 0.00739 \end{aligned}$$

Reducing this expression results in the finding that the mathematical model cannot tell the difference between applying the clearance variation before or after assembly of the two parts. The variation found is the same, namely  $4.690 \pm 0.00739$ .

The tolerance associated with manual process variables is *prescribed* rather than calculated, similar to form tolerances. Information on tolerances for each allowable degree of freedom (DOF) and the type of distribution which will model the axis is needed. The prescribed tolerance allows the designer to predict the amount of variation due to the characteristics of the joining process, whether it be manual or automatic position of the two parts. Distribution types include both normal and uniform distribution types corresponding to the extreme nominal values of  $\pm 3\sigma$  or  $\pm \sqrt{3}\sigma$  limits respectively.

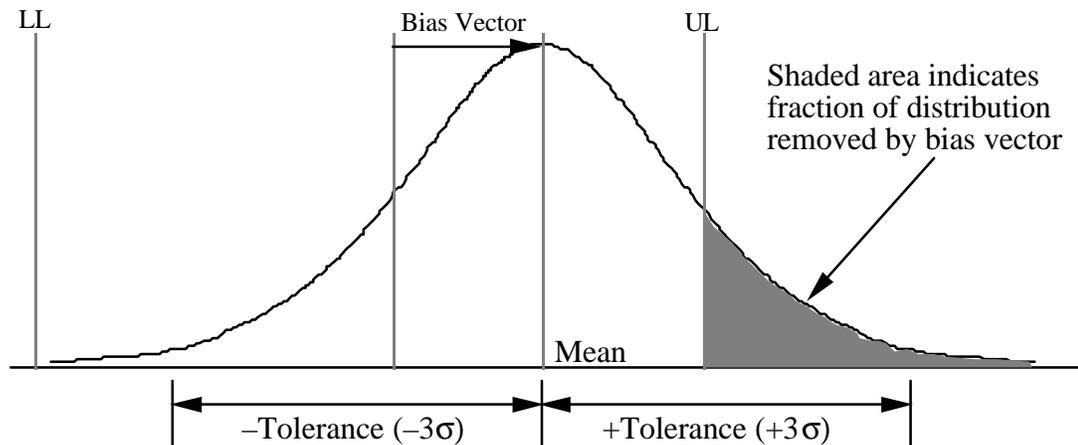
### BIAS VECTOR

A bias vector may be applied to manual process variations to allow for non-central bias in the process. An example is illustrated in figure 5.4, a door mounted with hinge plates.



**Figure 5.4.** Bias factor variation.

The door would tend to shift vertically downward and rotate about a point due to gravity. By applying a bias vector in the direction of the shift a more accurate model will be created. A bias may also be applied to the rotational degree of freedom, to represent similar biasing of rotational assembly processes. The bias vector has the effect of shifting the mean of the assembly distribution and possibly increasing the percent rejects (see figure 5.5).



**Figure 5.5.** Bias factor effect on distribution of sensitivities.

The model for manual process variation is similar to position variation and other feature controls, where a zero length vector is inserted into the vector loop of the assembly and given a tolerance variation. Only, the bias vector is not zero length..

Sensitivities are then obtained for the propagation of these variations through the kinematic vector loop equations. Expressions for tolerance accumulation are formed. The variances of the manual processes are added to the dimensional and form variances as worst case or root sum square sums. Their percent contribution to overall assembly variation may then be predicted quantitatively.

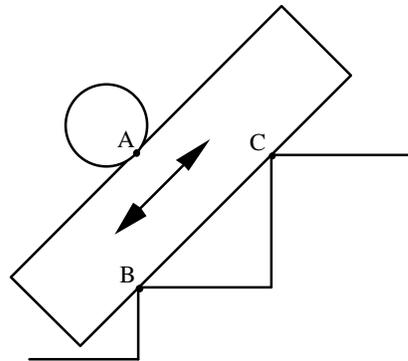
### Degrees of Freedom Analysis

The previous AutoCATS modeler did not contain a complete Degrees of Freedom (DOF) analysis check. Not having a complete DOF analysis required the user to perform more procedures by hand. If the user did not know how complete the analysis system was or was unsure how to do DOF analysis the initial system's advantages were lost. As previously mentioned in Chapter 2, kinematic joints remove degrees of freedom from an

assembly part. When the joints on the part are collectively considered they restrict a part's motion. An assembly is not properly constrained when too few joints are associated with the part. Now the DOF check has been generalized further to permit the detection of unconstrained assemblies.

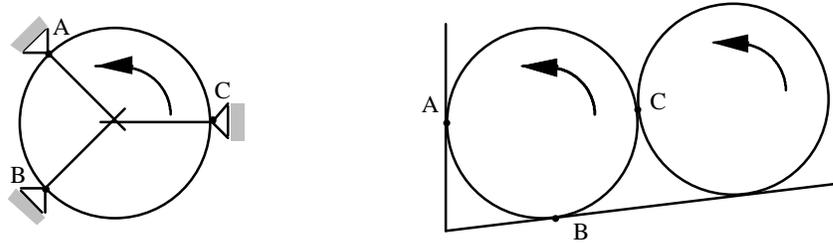
The previous prototype DOF analysis accounted for both the number of degrees of freedom and the associated directions. This allowed the program to analyze both parallel translational degrees of freedom and co-linear rotational degrees of freedom. Yet, it was not fully tested and did not account for non-coincident instantaneous centers. These issues are addressed and resolved below.

Several test models were run to evaluate the degree of freedom analysis implemented by Larsen [Larsen, 1991]. The Degree of Freedom (DOF) analysis program successfully checked for parallel translational degrees of freedom. When using an assembly such as figure 5.6 with several joints (A, B and C) restricting rotation and creating parallel translational degrees of freedom on a part, the correct translational degree of freedom remained after analysis.



**Figure 5.6.** Translational Degree of Freedom Analysis.

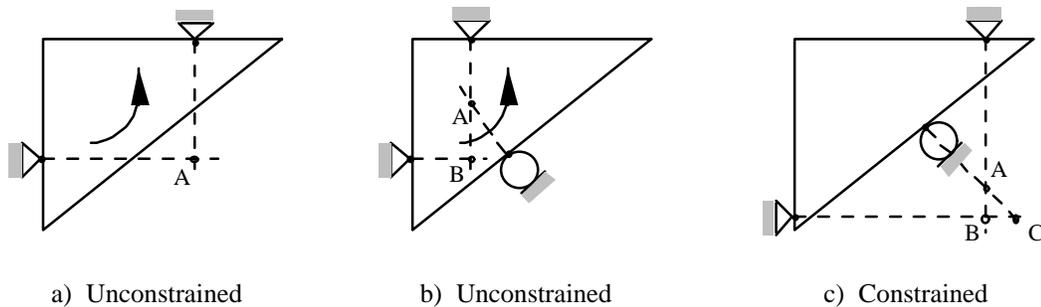
Some representative rotational test cases which were successfully completed included instantaneous centers and co-linear rotational degrees of freedom with both edge and cylindrical sliders (see figure 5.7). An edge joint removes one degree of freedom. Theoretically, the 3 joints should remove 3 degrees of freedom. However, the three joints positioned as shown, permit rotations about the axis formed by the intersection of surface normals. In all cases tested, the correct rotational degree of freedom remained.



**Figure 5.7.** Degree of Freedom Analysis with coincidental centers.

Another case involving instant centers has now been studied in which a degree of freedom still exists even though the centers are not coincidental. Figure 5.8a is unconstrained. The two edge joints still allow the triangle to have a rotational degree of freedom about the instantaneous center. When a cylindrical slider joint is added to the assembly as in figure 5.8b the part may still rotate, even though the centers are not coincident.

Through kinematic principles, it was demonstrated that a part will only be constrained when the number of the centers which intersect outside of the part of concern equal the number of contact points, or joints. This is defined by assuming that separation does not occur and only small motions occur. For example, when the intersection of points A, B or C are inside the triangle (figure 5.8b) the part may rotate and is said to be unconstrained. When all points of intersection A, B, and C are outside the geometric restraints of the triangle, as in figure 5.8c, then the part is fully constrained and may not move in any direction.



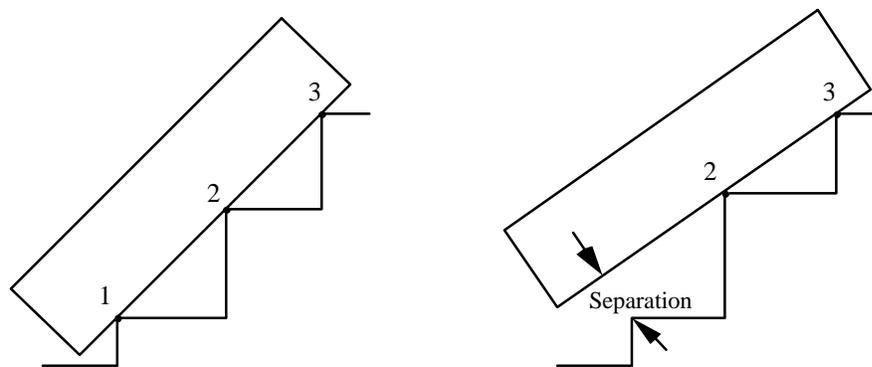
**Figure 5.8.** Rotational Degree of Freedom Analysis.

**REDUNDANT JOINTS**

While modeling assemblies, a system may become over-constrained by modeling too many joints. Autoloop generation cannot function properly when redundant joints are present. It will find extra loops according to the number of closed loops calculated from the equation

$$L = J - P + 1.$$

For example, consider a block with three steps (see figure 5.9). There is a tolerance zone on the block and a tolerance zone on the steps. Due to manufacturing variation, it is highly unlikely that all three corners will be in contact simultaneously. If the tolerance zone on step 2 is out of the tolerance zone on the block, step 2 will most likely be the highest step. Steps 2 and 3 will support the block and separation will occur at step 1 as in figure 5.9. Step 1 is then redundant and an extra loop will be created if kept. By running a DOF check on each part, parts which are over-constrained may be found before running the automatic loop generation.



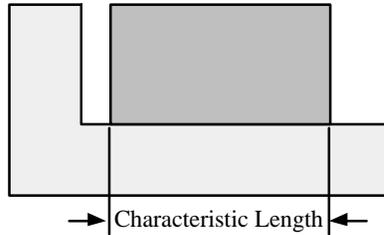
**Figure 5.9.** Redundant joint with tolerance bands.

Modeling of the assembly assumes that no separation occurs between parts. In order to predict separation due to the redundant joint, one may model the assembly two ways. The first is to create an open loop assuming separation at one joint and create closed loops with the other joints. Another, more accurate modeling technique would be to analyze the system using first joints 1 and 2, then 2 and 3, and then again with joints 1 and 3. By finding the greatest slope variation of the block using two joints at a time the joint pair which creates the greatest variation may be identified.

### **Feature Controls Requiring a Characteristic Length**

Feature controls account for surface variations on a part. When two parts are in contact along a plane (in two dimensions), then a characteristic length is required. A characteristic length is the length of contact between two parts. The characteristic length allows the analysis to be performed on the entire length of contact. For example, when two planes

are in contact (figure 5.10) and a feature control is applied to one or both of the surfaces, the surface variation only affects the assembly over the length of contact.



**Figure 5.10.** Characteristic length with a planar joint.

In tolerance analysis, the characteristic length is used to approximate the rotational variation introduced into an assembly at a planar or rigid joint. The designer may now define a characteristic length for each planar or rigid joint as part of the engineering model data structure.

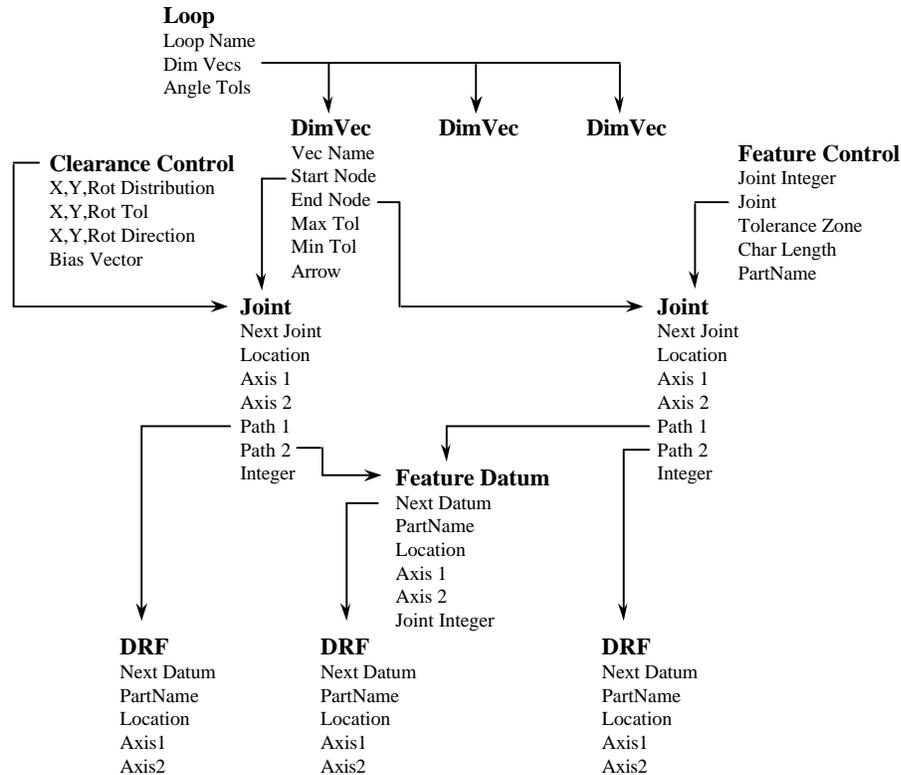
A characteristic length is not required for joint types which are point contacts, since the length of contact is considered to be zero. Point contact joints include: edge sliders, cylindrical sliders, parallel cylinders and revolute joints.

### Data Structure

The addition of manual process variations, design specifications and open loops to the assembly tolerance modeler, AutoCATS required major modifications to the data structure. This information also had to be added to the neutral file which transmits the geometric and kinematic information to the analysis package CATS/PC.

The engineering modeling elements are combined into a hierarchical data structure for robustness and ease of use. The data structure is based on the AutoCAD database and blocking features. Kinematic information like numeric data, pointers and modeling elements are stored with geometric entities in AutoCAD by assigning attributes to block entities such as lines and symbols. Pointers have been created between related elements to generate a continuous flow of information linking the elements information in the database. As a new entity or modeling element is created or deleted the database is automatically updated to show this modification. Figure 5.11 shows the AutoCATS

hierarchical data structure schematically. All model data is stored as part of an AutoCAD assembly model.

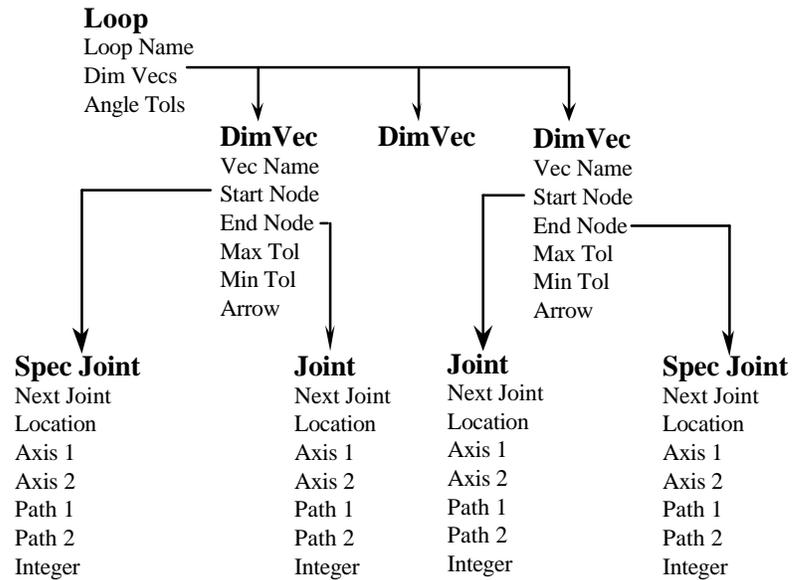


**Figure 5.11.** AutoCATS Data Structure.

The loop is at the top of the data structure. The loop block points to dimension vectors which form the loop. Each dimension vector contains nominal and tolerance information about the specific vector. Each vector also points to the two specified endpoints, either joints or datums. Feature controls also point to the associated joint entities. The joints contain such information as the sliding plane axis and pointers to two datum paths. A chain of pointers traces each associated datum path to a datum reference frame (DRF). The DRFs are the final step in the hierarchy. In this way all kinematic and geometric information is linked into a continuous engineering model which may be accessed from any point in the data structure.

While figure 5.11 shows an example of a closed loop figure 5.12 illustrates the differences in an open loop structure. Open loops differ from closed loops in that the starting and ending nodes of the dimension vector will be specification datums rather than a single joint. The first dimension vector will contain the starting point of a specification.

Likewise, the final dimension vector will contain the ending specification end point. This creates a loop which does not close on itself. In essence, one vector is removed from the loop to create an open loop. In this way, the first and last dimension vector define the start and ending nodes of an open loop respectively.



**Figure 5.12.** AutoCATS Open Loop Data Structure.

Several user interface modifications from the previous prototype allow AutoCATS to be a more intuitive system. Editing capabilities have been added as has a new menu structure. These enhancements which have been added in order to save time and reduce rework for the end user will be discussed next.

### Editing Capabilities

Most mistakes in AutoCATS occur while creating datum paths for joints or specification endpoints. Several new editing features have been implemented in order to allow the user to backup a step in the process of creating a joint, rather than creating the joint from the beginning. If a datum path is created incorrectly, the user has the choice to not only delete the entire datum path, but also to delete the feature datums which have been created for that datum path. Once a joint is created, another editing feature has been added to allow the datum paths with the associated DRFs to be displayed. This menu item can be helpful when predicting vector loops.

In the previous prototype, once a vector loop was created and tolerances assigned to the dimensions, the tolerances could not be changed without erasing the vector loops and recreating the loops again. For longer vector loops this process was very time consuming. Now, tolerances may be changed by simply selecting the vector to be modified, provided it is not a dependent variable, and change the tolerance. The old tolerance is shown as the default tolerance. Once a new tolerance is entered the database is updated.

Besides editing capabilities, all modeling elements may now be deleted. Previously, only feature datums, parts and joints could be deleted. In the new system, all modeling elements including feature controls, manual process variations, design specifications and closed or open vector loops may be deleted.

### **Menu Structure**

A new menu structure was created to parallel the AutoCAD menu structure both in form and function. Each menu contains a list of all other menus on the same level so that menu changes may be made without returning to the AutoCATS main menu. The menu structure was also modified in order to create sub levels, such as Parts and Joints, which were parallel in appearance and function and allowed the new modeling and editing enhancements to be implemented.

All AutoCAD functions are available within AutoCATS. Several AutoCAD functions which were used most often have been enhanced and are now displayed on AutoCATS menus. These include zoom, pan, redraw and cancel. Figure 5.13 describes the AutoCATS new menu structure.

AutoCATS Main Menu						
SETUP RESET PARTS JOINTS LOOPS FEATURES SPECS DISPLAY CATSFILE						
AutoCAD						
AutoCATS <b>PARTS</b> * * * * *	AutoCATS <b>JOINTS</b> * * * * *	AutoCATS <b>LOOPS</b> * * * * *	AutoCATS <b>FEATURES</b> * * * * *	AutoCATS <b>SPECS</b> * * * * *	AutoCATS <b>DISPLAY</b> * * * * *	AutoCATS <b>CATSFILE</b> * * * * *
NEW_P DELETE_P ----- JOINTS LOOPS FEATURES SPECS DISPLAY CATSFILE ----- Zoom In Zoom Out Pan Redraw Cancel AutoCAD	NEW_J EDIT J DELETE_J DOFcheck ----- PARTS LOOPS FEATURES SPECS DISPLAY CATSFILE ----- Zoom In Zoom Out Pan Redraw Cancel AutoCAD	NEW_L EDIT_TOL DELETE_L AUTOLOOP ----- PARTS JOINTS FEATURES SPECS DISPLAY CATSFILE ----- +/- TOL + TOL - TOL ----- Zoom In Zoom Out Pan Redraw Cancel AutoCAD	NEW_F DELETE_F ----- PARTS JOINTS LOOPS SPECS DISPLAY CATSFILE ----- +/- TOL + TOL - TOL ----- Zoom In Zoom Out Pan Redraw Cancel AutoCAD	CATSON NEW_S DELETES ----- PARTS JOINTS FEATURES LOOPS DISPLAY CATSFILE ----- +/- TOL + TOL - TOL ----- Zoom In Zoom Out Pan Redraw Cancel AutoCAD	CATSOFF Drfs Joints FeatDtms Loops Features Specs ----- PARTS JOINTS LOOPS FEATURES SPECS CATSFILE ----- Zoom In Zoom Out Pan Redraw Cancel AutoCAD	WRITE_NF READ_NF ----- PARTS JOINTS LOOPS FEATURES SPECS DISPLAY ----- Zoom In Zoom Out Pan Redraw Cancel AutoCAD

Figure 5.13. AutoCATS Side Menu Structure.

### Icon Menus

With the computer industry moving towards graphical input and selection devices, icon menus have also been implemented. Users may either use pop-up menus or icon menus depending on their preference. By creating an environment which the user feels they may control and change, AutoCATS becomes a more marketable product, and a product which designers will want to work with.

### Manual and On-Line Help

The previous prototype did not include a manual or any help commands. A manual containing step-by-step examples, a command section and glossary have allowed AutoCATS to be complete enough for beta software testing. The manual has greatly increased the ability for AutoCATS to be a complete modeling package.

Just as AutoCAD contains an on-line help command, AutoCATS now contains an on-line help command. By appending information to the AutoCAD help database AutoCATS' on-line help is transparent to the user. This allows the user to obtain a list of both AutoCAD and AutoCATS commands in case a command name or format is forgotten.

### Comparison to Previous Prototype

The new AutoCATS modeling system has grown from the generalized approach previously implemented into a more comprehensive engineering modeling system which may model a wide variety of problems using relatively few modeling elements and techniques. Below is a comparison to the previous AutoCATS version (table 6.1).

**Table 5.1.** Comparison of AutoCATS versions.

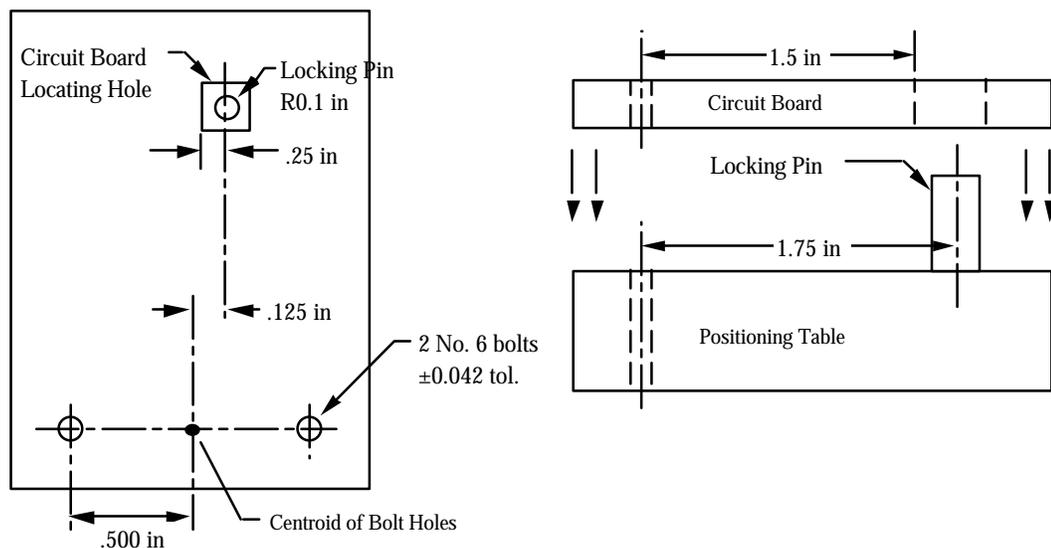
	Previous Version	New Version
<b>Modeling Features</b>		
Design Specifications	no	yes
Open Loops	no	yes
Manual Process Variations	no	yes
Automatic loop generation	closed loop	closed and open loops
<b>Model Enhancements</b>		
DOF analysis	directions only	non-coincidental centers
Characteristic Lengths	no	yes
Zero Length Vectors	no	yes
<b>Interface Enhancements</b>		
Editing Capabilities	partial	full
Deleting Elements	partial	full
Icon menus	no	yes
On-line Help	no	yes
Users Manual	no	yes
Compatibility with Rev 11	no	yes

## CHAPTER 6: CASE STUDIES

In this chapter, three case studies representing the new modeling capabilities: manual process variables, design specifications, and open loops will be presented. These test cases will depict a wide variety of realistic assemblies which could not be modeled previous to this thesis. First, a circuit board with a locating mechanism will be modeled showing how manual process variations and design specifications may be used. Next, a positioner assembly will portray how two design specifications may be modeled with one open loop. Finally, a clutch will be modeled using kinematic design specifications.

### Circuit Board Assembly

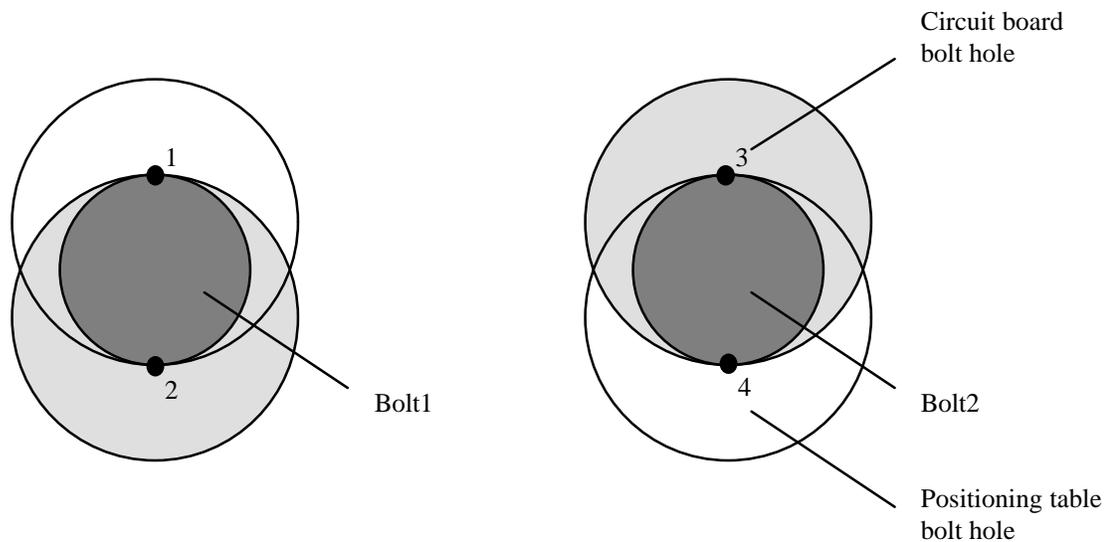
This assembly will illustrate the difference between modeling with and without manual process variations as well as the use of a gap specification and an open loop. Suppose that a circuit board is bolted to a positioning table and must align with a locating pin.



**Figure 6.1.** Top and side view of circuit board assembly.

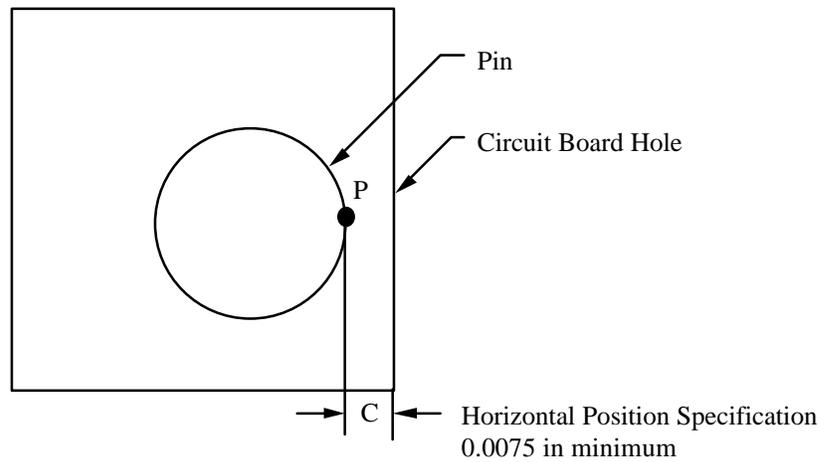
If the locating pin is unable to fit through the square receiver hole of the circuit board, the board is not correctly placed and as a result the assembly will not function. There are two ways to model this assembly, either using an extreme case study, or using manual process variations.

This extreme case requires that the circuit board be rotated clockwise until contact is established between the edge of the holes and the bolts (points 1 and 4 in figure 6.2). This sub-assembly is then rotated until similar contact is achieved between the bolts and the edge of the holes in the positioning table (points 2 and 3 in figure 6.2). A close view of the bolt holes would appear as in figure 6.2. The engineering model requires four parallel cylinder joints to model the contacts between the four parts.



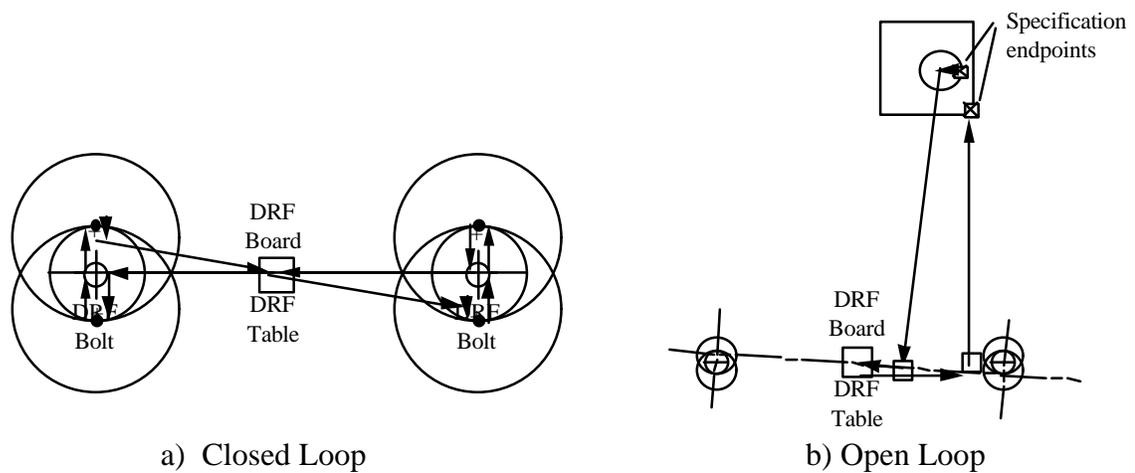
**Figure 6.2.** Bolt holes with parts in extreme contact positions.

An assembly tolerance requirements is needed between the pin and the locating hole. A position tolerance specification will be applied from the corner of the hole to the outer radius of the pin (see figure 6.3). A position tolerance is used here so the vertical location of the minimum gap need not be specified. However, the nominal location of point P on the edge of the point must be specified precisely. The minimum horizontal clearance is specified at 0.0075 inches for proper operation of the assembly.



**Figure 6.3.** Position Specification on circuit board assembly.

The number of loops required for this assembly is one open loop and one closed loop ( $L = 4 \text{ joints} - 4 \text{ parts} + 1 = 1$ ). The automatic loop generator created both the open and closed loops automatically as shown in figure 6.4.



a) Closed Loop

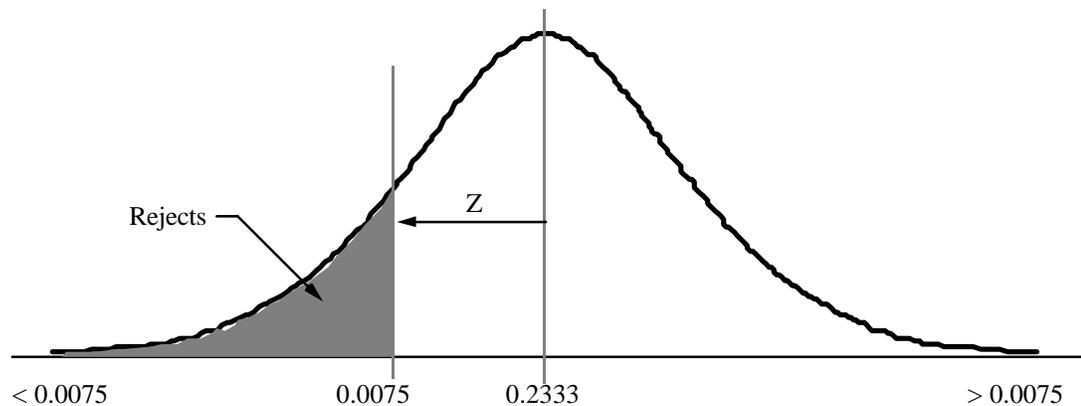
b) Open Loop

**Figure 6.4.** Loops for extreme position assembly.

The resulting neutral file may be found in the Appendix. The results from analyzing the model in the extreme position are shown in table 6.1, while figure 6.5 shows graphically the results.

**Table 6.1.** Analysis Results for Extreme Position.

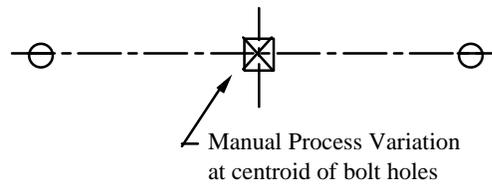
Average Dim	Worst Case	Tolerances RSS Case	STD Deviation	Upper Tail Z/Rejects	Lower Tail Z / Rejects
0.23330	0.03353	0.01401	0.004671	N/A	1.61 / 54135.4

**Figure 6.5.** Graphical interpretation of analysis.

Under normal assembly conditions, it is not reasonable to assume that the assembly will be located in the extreme position. It is highly unlikely that the operator will deliberately assemble the circuit board at the extreme position. Rather, it is more likely that the assembly will be somewhere in the clearance zone. This may be accounted for by assuming a normal distribution for the assembly process. Extreme cases are associated with the tails of the distribution. This means that the results from table 6.1 are higher than what is expected. A similar analysis could be performed for the other extreme position, for which the circuit board is rotated in the counter-clockwise direction.

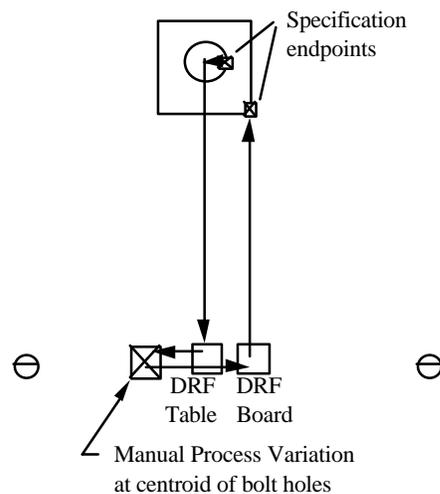
An alternate method, using new theory, will now be used to model this assembly. A manual process variation will be applied at the nominal positions rather than using extreme positions of the assembly. The manual process will assume that the circuit board is located somewhere in the clearance zone, with the probable location described by a normal distribution. The parts are first oriented in their nominal positions. At this point the manual process variation is applied between the circuit board and the positioning table at the centroid of the bolt holes (see figure 6.6). By applying the variation at the hole pattern centroid, the rotational variation is accounted for at each of the bolt holes. A set

of tolerances are then specified for the X, Y, and rotational variation as determined from the clearance about the bolts.



**Figure 6.6.** Manual process variation applied at centroid of fastener pattern.

The tolerance specification is again applied to the clearance between the pin and hole. This engineering model only requires one open loop and no closed loops. The open loop generated appears as below in figure 6.7. The neutral file may be found in the Appendix.



**Figure 6.7.** Open loop using a manual process variation.

The results of the open loop analysis (table 6.2) show that the percent rejects have been significantly reduced, approximately by a factor 85. All other values are approximately a third of the extreme position results. This shows that one will predict an excessive amount of rejects modeling the assembly at extreme contact positions. A more realistic estimate, using the predicted variation of a worker or machine, results from using the manual process variation.

**Table 6.2.** Analysis Results using Manual Process Variation.

Basic Dim	Worst Case	Tolerances RSS Case	STD Deviation	Upper Tail Z/ Rejects	Lower Tail Z / Rejects
0.29155	0.01441	0.00694	0.002314	3.22 / 638.6	3.26 / 552.7

**Table 6.3.** Analysis Results using Extreme Position.

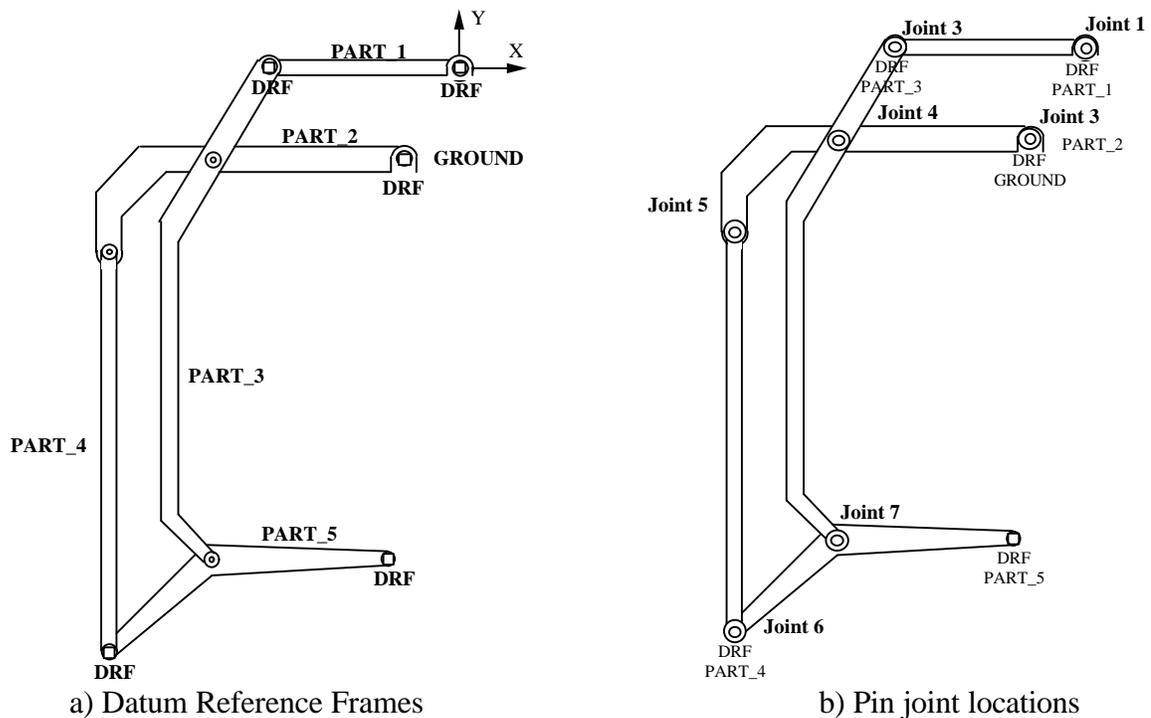
Basic Dim	Worst Case	Tolerances RSS Case	STD Deviation	Upper Tail Z/ Rejects	Lower Tail Z / Rejects
0.23330	0.03353	0.01401	0.004671	1.61 / 54201.6	1.61 / 54135.4

From this it can be seen that in some cases using manual process variations not only reduces the number of required loops and modeling steps, but also better represents the variation associated with clearance of bolt holes or other joining processes. The estimated number of rejects was reduced from 54,136 to only 553 using the entire clearance zone rather than an extreme position.

### Remote Positioner

In the next example, the remote positioner presents a practical application of tolerance modeling using design specifications defined by an open loop. The remote positioner (see figure 6.8) is a linkage system whose function is to rotate link 5 about a fixed point P, while maintaining a specified angle relative to the input link or part 1. For example, an input angle of 20° at link 1 should correspond to an angle of 20° for link 5. Applying a global orientation design specification requires the angle of rotation of link 5 to be within a specified tolerance to the rotation of the input link (link 1). Applying a position assembly tolerance on the resulting position of point P establishes its allowable range. Before this research, the remote positioner could not have the proper constraints applied and therefore was not modeled accurately.

Parts are defined on the remote positioner by locating the local datum reference frame (DRF) on each part. There are six parts, including the ground. Notice that the ground and link 2 have identical DRF locations (see figure 6.8a).

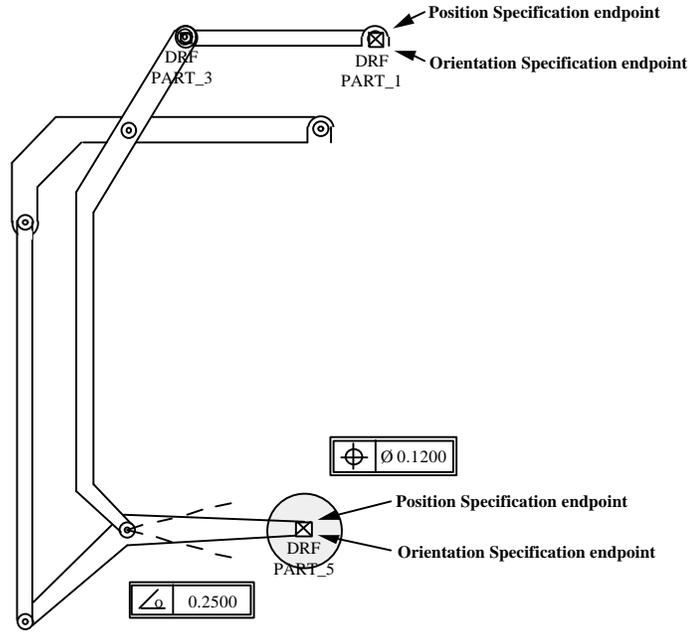


**Figure 6.8.** Model elements defined for remote positioner.

The are assembled by seven revolute or pin joints, as shown in figure 6.8b. Each joint introduces a kinematic or rotational degree of freedom into the assembly. Any manufacturing variations in the link lengths will produce small rotational adjustments at the joints, resulting in errors in the position of point P and orientation of link or part 5.

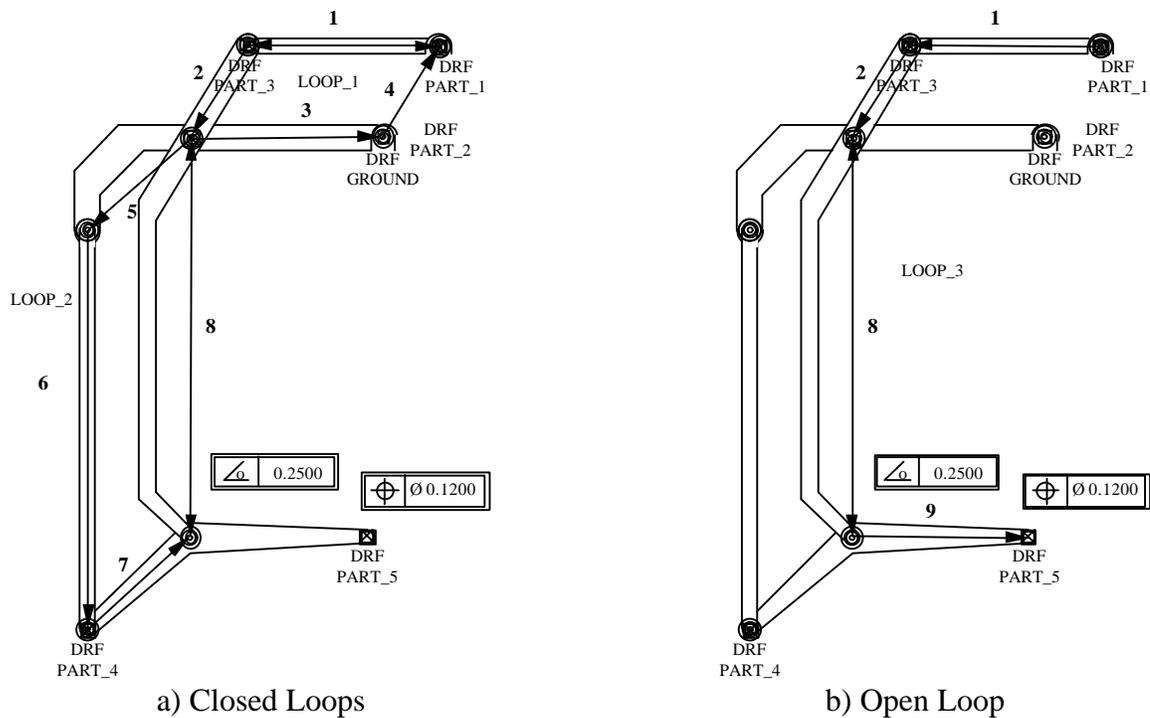
Assembly tolerances are must be applied to specify the engineering requirements. As stated before, the remote positioner requires link 5 to be within a certain tolerance zone relative to the global input angle and origin in order to function properly. The global orientation tolerance specification requires link 5 to be within an angular tolerance zone of  $0 \pm 0.25^\circ$ . The design specification requires link 5 to be in the tolerance zone even with the accumulated tolerance stack-up of the assembly.

Point P at the tip of link 5 must also maintain a position tolerance relative to the global origin of  $\pm 0.12$  inches or the mechanism will be too inaccurate. A position tolerance limit is applied to point P to meet this constraint. Figure 6.9 shows the application of these design specifications.



**Figure 6.9.** Remote positioner with design specifications.

As stated earlier, the position and global orientation specification may be analyzed using one common open loop. The other loops required to analyze this assembly include two closed loops, which are used to determine kinematic variables (see figure 6.10).



**Figure 6.10.** Loops for remote positioner.

The open loop begins at the global origin, which is also the DRF for link 1, and follows vectors 1, 2, 8 and 9, ending at point P. Notice that both the global orientation and the position assembly tolerance have both beginning and ending datums at identical locations (see figure 6.9). The dimensioned variables with associated manufacturing tolerances are shown below in table 6.4 for each loop:

**Table 6.4.** Dimensions and Tolerances for Remote Positioner.

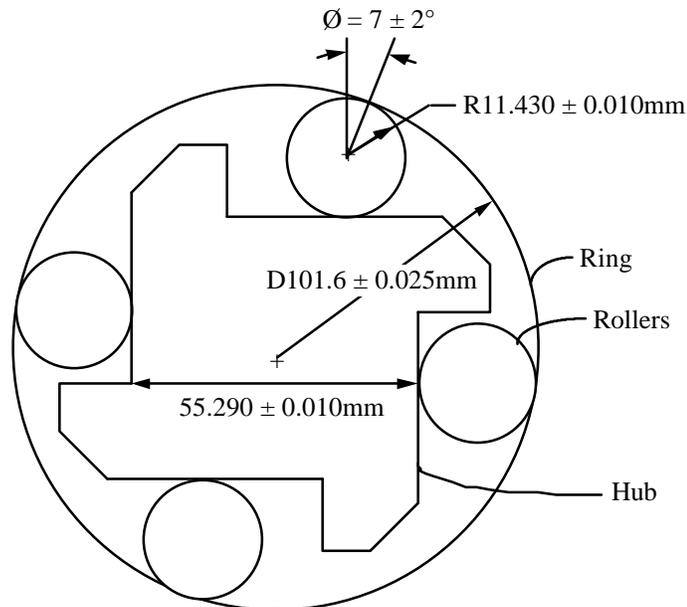
	VECTOR	DIMENSION	TOLERANCE
LOOP 1	1	22.0000	$\pm 0.005$
	2	10.4932	$\pm 0.005$
	3	22.0000	$\pm 0.005$
	4	10.4932	$\pm 0.005$
LOOP 2	5	12.915	$\pm 0.005$
	6	49.312	$\pm 0.005$
	7	12.915	$\pm 0.005$
	8	49.312	$\pm 0.005$
LOOP 3	1	22.0000	$\pm 0.005$
	2	10.4932	$\pm 0.005$
	8	49.312	$\pm 0.005$
	9	22.0000	$\pm 0.005$

Once both the open and closed loops have been defined, the assembly model is complete and contains all the manufacturing and engineering requirements necessary for tolerance analysis. The open loop includes tolerance accumulation effects from the entire assembly through interaction with the closed loops. By solving for the tolerance sensitivities and assembly tolerances, the designer will be able to make sure that the design specifications are met and the linkage will function correctly. The neutral file for this problem may be found in the Appendix.

### One-Way Clutch

The one-way clutch problem (see figure 6.11) illustrates the use of kinematic design specifications to constrain the resultant variation of an important assembly variable. This common device, used to start gas lawn mowers and automobile engines, is designed to allow rotation in only one direction. The hub is attached to a shaft, which in turn is attached to a driving mechanism. When the hub rotates clockwise relative to the ring, the roller slips on the inside of the ring. If the hub rotates counter-clockwise the spring causes

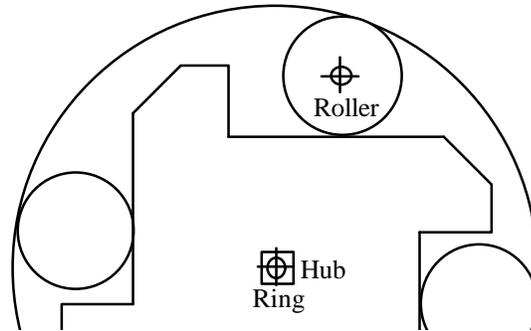
the roller to wedge between the hub and the ring, causing the two to lock and rotate together.



**Figure 6.11.** Dimensioned One-Way Clutch.

The contact angle  $\emptyset$  between the roller and the ring controls the performance of the clutch. If the contact angle is too large, then the clutch will never lock leaving the rollers slipping. Conversely, if the contact angle is too small, then the clutch will not unlock. In order to assure the clutch functions properly the angle must be within a specified limits of  $7^\circ \pm 2^\circ$ , as shown in figure 6.11. This tolerance will be defined using a dependent angle design specification.

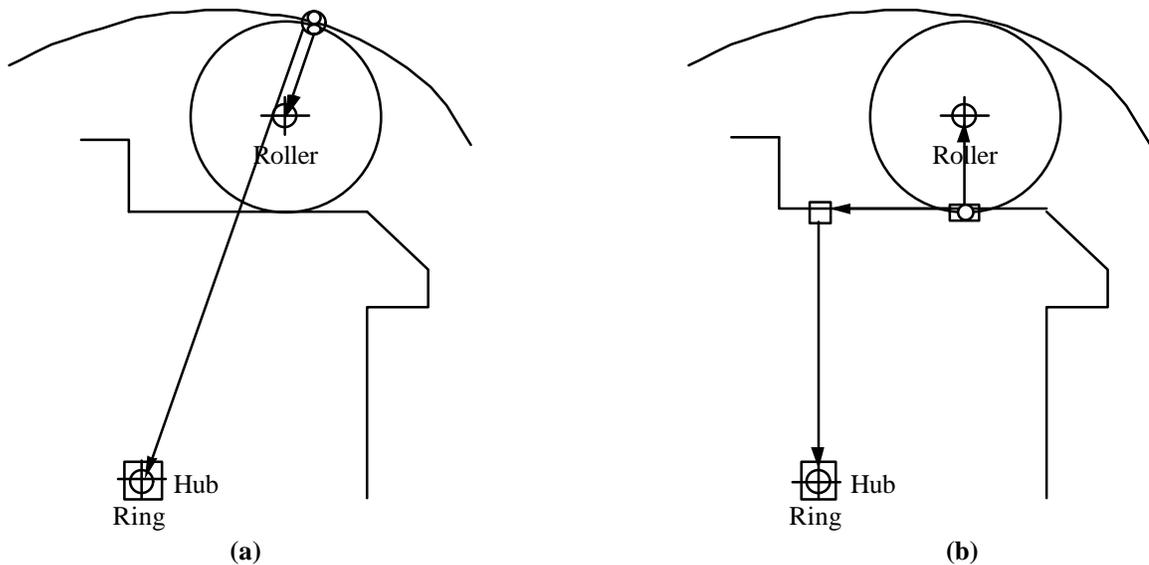
Three parts are identified in the assembly, the Ring, Hub and Roller. The Datum Reference Frames (DRFs) for each part of the assembly are shown below (see figure 6.12). Notice that the ring and hub DRFs have identical locations.



**Figure 6.12.** One-way clutch with DRFs defined.

The second step in modeling the clutch is to locate the contact joints between each part. The first joint is the revolute or pin joint between the ring and the hub, which is coincident with the DRFs of both the ring and the hub. The datum paths are therefore zero length.

A cylindrical joint is required between the roller and the ring. The datum paths from the joint to both the roller and the ring go directly to the DRF along the respective radii. Both datum paths, following controlled dimensions, are shown in figure 6.13a.

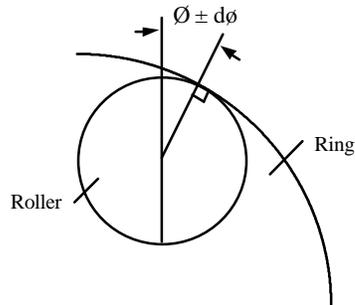


**Figure 6.13.** Cylindrical and Cylindrical slider joint datum paths.

The final joint is a cylindrical slider between the roller and the hub. The datum path for the roller goes directly back to the reference frame at the roller axis. The datum path for the Hub requires a feature datum on the hub surface. The feature datum splits the datum

path into one kinematic dimension, along the sliding plane of the roller, and a geometric dimension, the height of the hub (see figure 6.13b).

Once the contact points have been identified, design specifications may be modeled. The design specification for the clutch assembly is the angle measured between the cylindrical datum reference frame and a contact point. This is a dependent relative angle specification. Two vectors are identified which define the relative angle. For the clutch, the vectors are shown in figure 6.14 below:



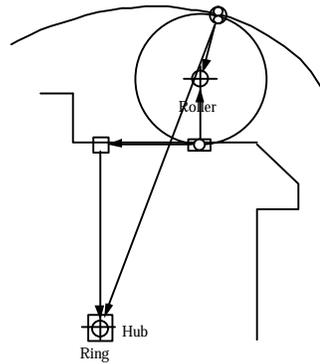
**Figure 6.14.** Relative angle specification.

A tolerance is then applied to the design specification, in this case  $\pm 2^\circ$ . Tolerance stack-up of the various component dimensions contribute to the overall variation of the angle  $\varnothing$ . With the assembly tolerance applied, the model has included all of the manufacturing variations which contribute to the desired angular dimension  $\varnothing$ . By meeting this assembly tolerance, the clutch's performance will be optimized.

Since the clutch is symmetric, the assembly only requires one closed loop as determined by the equation:

$$\begin{aligned} L &= J - P + 1 \\ &= 3 - 3 + 1 \end{aligned}$$

The kinematic loop relates all modeling elements to the resultant assembly dimensions. The model contains both kinematic and manufacturing variations which were selected while forming the datum paths. Once the loop has been generated, using the automatic loop generator (see figure 6.15), the modeler produces a neutral file which contains all the essential information for tolerance analysis by CATS. This neutral file is found in the Appendix. The engineering requirements may then be translated into manufacturing requirements.



**Figure 6.15.** Loop for one-way clutch.

### Limitations

The three test cases described above were all modeled correctly using new methodology developed in this thesis. While these test cases show the capability of the AutoCATS program with the new algorithms implemented, it should be remembered that there are inherent limitations to the software.

First, just as with any geometric application, AutoCATS is limited to the accuracy of the CAD system. AutoCAD only carries dimensions to the fourth decimal place accurately, although up to seven digits after the decimal place are displayed. Tolerance analysis is based on computation of differences in dimensions to the third or fourth decimal digit. With this limited accuracy it is imperative that one use construction tools to develop assembly geometry in order to assure accurate node locations and part orientation within the assembly and prevent geometric error. Although geometry may appear correct to the user, estimation of points, although a seemingly insignificant error, will result in inexact calculations and assignment of tolerance variables.

A second limitation is that the final outcome of the assembly model is dependent upon the correct connectivity matrix. If a system is over-constrained, then there is no guarantee of a correct solution. To prevent over-constrained systems, the Degree of Freedom (DOF) analysis should be run before generating vector loops. It is also assumed that the datum paths are correctly modeled using only geometric and kinematic dimensions. It should be remembered that it is the designer's responsibility to set up the problem correctly.

Once these precautions and limitations have been considered AutoCATS may be used to its full potential, providing valuable information to both the designer and manufacturer.

## **CHAPTER 7: CONCLUSIONS & RECOMMENDATIONS**

This research has produced a modeling system which better predicts tolerances. It has created a more comprehensive systematic step by step approach to tolerance modeling using familiar engineering elements.

### **Conclusions**

The author's contributions have resulted in several new capabilities and enhancements to the existing system. They are as follows:

- Automatic open loop generation has been developed and implemented. By expanding the Automatic Vector Loop (AVL) algorithm, a more robust algorithm has been designed. Redundant dimensions are eliminated and zero length vectors are now possible. Manual open loop generation was also developed in the process.
- Design specifications, a new modeling element, allow assembly tolerance limits to be applied to critical features of mechanical assemblies. Design specifications parallel feature controls, only tolerances are between two parts. Seven models have been developed which allow a wide variety of assemblies to be analyzed. The specification endpoints were associated with datums rather than joints since specification endpoints only have one datum path rather than two datum paths like joints.
- The modeling of unconstrained kinematic variables was introduced. Unconstrained kinematic variables exhibit variation associated with manual processes just as size and feature dimensions exhibit process variation. A bias vector was also created to account for non-central distributions.
- The Degree Of Freedom (DOF) analysis was fully tested and expanded. New analysis checks for coincident and non-coincident centers of rotation for multiple slider joints have been developed.

### **Recommendations for Future Work**

Although a fully functioning modeling software package has been developed using the contributions of this thesis there are several areas for future research including:

- Associating elements on an assembly with part drawings for automatic updating of both assembly and part drawings when the neutral file is returned from CATS/PC. It is recommended that the possibility of scanning a drawing list or using a hierarchical tree for updating be explored.
- Investigation and implementation of geometric associativity. New database structures and features of AutoCAD Rev12 should be examined for this purpose. Through geometric associativity when a drawing is revised the engineering model is automatically updated as well.
- Feature recognition creating the ability to automatically determine joint information. Create a system where the CAD system realize for example that an intersection between a line and a curved surface is a cylindrical slider, thus reducing further user input.
- Ability to discern between multiple position assembly drawings. Investigate the use of Robline, a robotics program, for different positions of a linkage system or draw a classical kinematic model for representing large kinematic motions.
- Design capability to provide feedback from CATS into the drawing environment, including a graphical interface showing pareto charts, interference between parts and highlighting suggested changes on the drawing. By creating a smoother link between analysis and modeling the analysis may be performed quicker. Using concurrently open windows will allow the user to see both the geometry and analysis.
- Ability to select an element from two identical elements which are coincident, such as two revolute joints on top of one another. Look into the possibility of creating a more specific filter, or selecting elements by node number. Also consider the possibility of automatically identifying the existence of stacked elements and showing the user a table with information for selection.

## **REFERENCES**

---

- Carr, C. David (1993), A Comprehensive Method for Specifying Tolerance Requirements for Assemblies, M.S. Thesis, Brigham Young University, Provo, UT, February.
- Chase K. W. and Greenwood W. H. (1987) "Computer-aided Tolerance Selection, CATS.BYU User Guide, Version 1.1 VAX/VMS," ADCATS Report No. 87-3, Mechanical Engineering Department, Brigham Young University, Provo, UT, September.
- Chase K. W. and Trego A. (1992) "AutoCATS User Manual," ADCATS Report No. 87-3, Mechanical Engineering Department, Brigham Young University, Provo, UT, August.
- Faux, I.D. (1981) "Preliminary Study of the Requirements for the Incorporation of Dimensions and Tolerances in a Geometric Model," Report #R-81-GM-03, CAM-I, Arlington, TX, June.
- Goodrich, Callis G. (1991), Representation and Modeling of Geometric Form Variations for 3-D Tolerance Analysis, M.S. Thesis, Brigham Young University, Provo, UT, June.
- Kim, S. H. and K. Lee (1989), "An Assembly Modeling System for Dynamic and Kinematic Analysis," Computer-Aided Design, Volume 21, Number 1, pp 2-12.
- Larsen, Glen C. (1991), A Generalized Approach to Kinematic Modeling for Tolerance Analysis of Mechanical Assemblies, M.S. Thesis, Brigham Young University, Provo, UT, July.
- Marler, Jaren D. (1988), Nonlinear Tolerance Analysis Using the Direct Linearization Method, M.S. Thesis, Brigham Young University, Provo, UT, December.
- Ranyak, Paul S. and Richard Fridshal (1988), "Features for Tolerancing a Solid Model," Computers in Engineering, Volume 1, pp. 275-280.
- Robison, Richard H. (1989), A Practical Method for Three-Dimensional Tolerance Analysis Using a Solid Modeler, M.S. Thesis, Brigham Young University, Provo, UT, April.
- Shah, Jami J. and David W. Miller (1990), "A Structure for Supporting Geometric tolerances in Product Definition Systems for CIM," ASME Manufacturing Review, Volume 3, Number 1, March, pp 23-31.
- Simmons, Andre P. (1990), Automated Vector Loop Generation for Kinematic Models of Mechanical Assemblies, M.S. Thesis, Brigham Young University, Provo, UT, August.

- Srikanth, S. (1990), A Unified Framework for Assembly Modeling, M.S. Thesis, Rensselaer Polytechnic Institute, Troy, NY, August.
- Wang, Nanxin and Tulga M. Ozsoy (1990), "Representation of Assemblies for Automatic Tolerance Chain Generation," Engineering With Computers, Volume 6, pp 121-126.
- Ward, Kerry R. (1992), A Practical Method for Three-Dimensional Tolerance Analysis Using a Solid Modeler, M.S. Thesis, Brigham Young University, Provo, UT, August.

# APPENDIX

## Circuit Board Extreme Postion Neutral File

```
FILE>> D:\CIRCUIT3.DAT 3-23-93 :
15:38
DESCRIPTION OF DATA FILE:
===== ASSEMBLY: CIRCUIT3 ( ) 2=====
DESCRIPTION:
TEX -----
-
ANM NAME:
ANO ASSM NO.:
DRA DRAW NO.:
MOD LOOP_2 ( ) LOOP_1 ( )
===== SPECIFICATIONS:
=====
DESCRIPTION:
TEX TYPE REF-NOD/DIM 2ND-NOD/DIM BASC DIM MAX DIM MIN DIM
SPE GAP( ) DATUM14 ( ) DATUM17 ( ) 0.23330( ) 0.24080( ) 0.22580(
)
===== PART: ___PIN (
)=====
DESCRIPTION:
TEX -----
--
PNM NAME:
PNU PART NO.:
DRA DRAW NO.:
===== PART: BOX (
)=====
DESCRIPTION:
TEX -----
--
PNM NAME:
PNU PART NO.:
DRA DRAW NO.:
===== PART: BOLT2 (
)=====
DESCRIPTION:
TEX -----
--
PNM NAME:
PNU PART NO.:
DRA DRAW NO.:
===== PART: BOLT1 (
)=====
DESCRIPTION:
TEX -----
--
PNM NAME:
PNU PART NO.:
DRA DRAW NO.:
===== DATUM LIST: (
)=====
DESCRIPTION:
TEX DATUM NAME TYPE X Y Z
DRF DATUM3 ( ) CYLCEN( ) .00035 .02099 .00000( )
AXI ( ) AXIAL ( ) .00000 .00000 1.00000( )
DRF DATUM9 ( ) CYLCEN( ) .00141 .04195 .00000( )
AXI ( ) AXIAL ( ) .00000 .00000 -1.00000( )
DRF DATUM2 ( ) RECTAN( ) .62500 .00000 .00000( )
DRF DATUM5 ( ) CYLCEN( ) 1.24859 -.04195 .00000( )
AXI ( ) AXIAL ( ) .00000 .00000 1.00000( )
DRF DATUM4 ( ) CYLCEN( ) 1.24965 -.02099 .00000( )
```

```

AXI          ( ) AXIAL ( )          .00000    .00000    1.00000( )
DRF DATUM7   ( ) CYLCEN( )         1.25000    .00000    .00000( )
AXI          ( ) AXIAL ( )          .00000    .00000    1.00000( )
DRF DATUM1   ( ) RECTAN( )         .62500    .00000    .00000( )
DRF DATUM11  ( ) CYLCEN( )         .00000    .00000    .00000( )
AXI          ( ) AXIAL ( )          .00000    .00000   -1.00000( )
DRF DATUM17  ( ) RECTAN( )         .96696    1.73095    .00000( )
DRF DATUM15  ( ) CYLCEN( )         .86718    1.73766    .00000( )
AXI          ( ) AXIAL ( )          .00000    .00000    1.00000( )
DRF DATUM16  ( ) RECTAN( )         .74972   -0.00839    .00000( )
DRF DATUM13  ( ) RECTAN( )         1.00000    .00000    .00000( )
DRF DATUM14  ( ) RECTAN( )         1.00000    1.50000    .00000( )

```

```

===== JOINT LIST: (
)=====

```

```

DESCRIPTION:
TEX JOINT NAME      TYPE          X          Y          Z
JOI JOINT12        ( ) PARCYL( )   .00524     .08985     .00000( )
AXI                ( ) AXIAL ( )   .00000     .00000     .00000( )
JOI JOINT10        ( ) PARCYL( )  -0.00081   -0.04802    .00000( )
AXI                ( ) AXIAL ( )   .00000     .00000     .00000( )
JOI JOINT6         ( ) PARCYL( )   1.25312     .04792     .00000( )
AXI                ( ) AXIAL ( )   .00000     .00000     .00000( )
JOI JOINT8         ( ) PARCYL( )   1.24849   -0.08999    .00000( )
AXI                ( ) AXIAL ( )   .00000     .00000     .00000( )
JOI JOINT12        ( ) PARCYL( )   .00524     .08985     .00000( )
AXI                ( ) AXIAL ( )   .00000     .00000     1.00000( )

```

```

===== LOOP: LOOP_2 (
)=====

```

```

DESCRIPTION:
TEX -----
--

```

```

TABLE:
TEX -----
--

```

```

TEX          VECTOR LOOP
TEX          X0          Y0          Z0          XANGO          YANG0          ZANGO
INI          .9670( )    1.7309( )    .0000( )    .000( )        .000( )        .000( )
TEX PART/J1/J2-JOINT  BAS DIM          MAX TOL          MIN TOL          DEP
NOD DATUM17
ZRO          ( ) 176.1512( ) .000000( ) .000000( ) ( )
DIM ___PIN/15-17 ( ) .100000( ) .003000( ) -.003000( ) ( )
NOD DATUM15
ZRO          ( ) 90.0000( ) .000000( ) .000000( ) DEP( )
DIM ___PIN/15-16 ( ) 1.750000( ) .005000( ) -.005000( ) ( )
NOD DATUM16
ZRO          ( ) -90.0000( ) .000000( ) .000000( ) ( )
DIM ___PIN/2-16 ( ) .125000( ) .005000( ) -.005000( ) ( )
NOD DATUM2
ZRO          ( ) 180.0000( ) .000000( ) .000000( ) ( )
DIM ___PIN/2-5 ( ) .625000( ) .005000( ) -.005000( ) ( )
NOD DATUM5
ZRO          ( ) 90.9622( ) .000000( ) .000000( ) DEP( )
DIM ___PIN/5-6 ( ) .089991( ) .005000( ) -.005000( ) ( )
NOD JOINT6
ZRO          ( ) 180.0000( ) .000000( ) .000000( ) ( )
DIM BOLT2/4-6 ( ) .069000( ) .005000( ) -.005000( ) ( )
NOD DATUM4
ZRO          ( ) 1.9244( ) .000000( ) .000000( ) DEP( )
DIM BOLT2/4-8 ( ) .069008( ) .005000( ) -.005000( ) ( )
NOD JOINT8
ZRO          ( ) 180.0000( ) .000000( ) .000000( ) ( )
DIM BOX/7-8 ( ) .090000( ) .005000( ) -.005000( ) ( )
NOD DATUM7
ZRO          ( ) 90.9622( ) .000000( ) .000000( ) DEP( )
DIM BOX/1-7 ( ) .625000( ) .005000( ) -.005000( ) ( )
NOD DATUM1
ZRO          ( ) -180.0000( ) .000000( ) .000000( ) ( )

```

DIM BOX/1-13	( )	.375000( )	.005000( )	-.005000( )	( )
NOD DATUM13					
ZRO	( )	90.0000( )	.000000( )	.000000( )	( )
DIM BOX/13-14	( )	1.500000( )	.005000( )	-.005000( )	( )
NOD DATUM14					

```

===== LOOP: LOOP_1      (
)=====
DESCRIPTION:
TEX -----
--
TABLE:
TEX -----
--
TEX              VECTOR LOOP
TEX      X0      Y0      Z0      XANGO      YANGO      ZANGO
INI      .0052( )  .0898( )  .0000( )  .000( )   .000( )   .000( )
TEX PART/J1/J2-JOINT  BAS DIM      MAX TOL      MIN TOL      DEP
NOD JOINT12
ZRO              ( )  -94.0568( )  .000000( )  .000000( )  ( )
DIM BOLT1/3-12   ( )  .069032( )  .005000( )  -.005000( )  ( )
NOD DATUM3
ZRO              ( )   3.0946( )  .000000( )  .000000( )  DEP( )
DIM BOLT1/3-10   ( )  .069017( )  .005000( )  -.005000( )  ( )
NOD JOINT10
ZRO              ( )  179.5512( )  .000000( )  .000000( )  ( )
DIM ___PIN/9-10  ( )  .090000( )  .005000( )  -.005000( )  ( )
NOD DATUM9
ZRO              ( )  -92.4379( )  .000000( )  .000000( )  DEP( )
DIM ___PIN/2-9   ( )  .625000( )  .005000( )  -.005000( )  ( )
NOD DATUM2
ZRO              ( )   .0000( )  .000000( )  .000000( )  ( )
DIM ___PIN/2-5   ( )  .625000( )  .005000( )  -.005000( )  ( )
NOD DATUM5
ZRO              ( )   90.9622( )  .000000( )  .000000( )  DEP( )
DIM ___PIN/5-6   ( )  .089991( )  .005000( )  -.005000( )  ( )
NOD JOINT6
ZRO              ( )  180.0000( )  .000000( )  .000000( )  ( )
DIM BOLT2/4-6    ( )  .069000( )  .005000( )  -.005000( )  ( )
NOD DATUM4
ZRO              ( )   1.9244( )  .000000( )  .000000( )  DEP( )
DIM BOLT2/4-8    ( )  .069008( )  .005000( )  -.005000( )  ( )
NOD JOINT8
ZRO              ( )  180.0000( )  .000000( )  .000000( )  ( )
DIM BOX/7-8      ( )  .090000( )  .005000( )  -.005000( )  ( )
NOD DATUM7
ZRO              ( )   90.9622( )  .000000( )  .000000( )  DEP( )
DIM BOX/1-7      ( )  .625000( )  .005000( )  -.005000( )  ( )
NOD DATUM1
ZRO              ( )   .0000( )  .000000( )  .000000( )  ( )
DIM BOX/1-11     ( )  .625000( )  .005000( )  -.005000( )  ( )
NOD DATUM11
ZRO              ( )  -93.3354( )  .000000( )  .000000( )  DEP( )
DIM BOX/11-12    ( )  .090000( )  .005000( )  -.005000( )  ( )
NOD JOINT12
===== END:
=====

```

## Circuit Board using Manual Process Variations Neutral File

```

FILE>> D:\CIRCUIT1.DAT                      3-19-93   :
9:27
DESCRIPTION OF DATA FILE:
===== ASSEMBLY: CIRCUIT1      ( )  2=====
DESCRIPTION:
TEX -----
-
-
ANM      NAME:
ANO ASSM NO.:
DRA DRAW NO.:
MOD      LOOP_1      ( )
===== SPECIFICATIONS:
=====
DESCRIPTION:
TEX A TYPE      REF-JOI/DIM      2ND-JOI/DIM      BAS DIM      MAX DIM      MIN DIM
SPE GAP( ) DATUM5      ( ) DATUM6      ( ) 0.29150( ) 0.29900( ) 0.28400(
)
===== PART: PART_1      (
)=====
DESCRIPTION:
TEX -----
--
PNM      NAME:
PNU PART NO.:
DRA DRAW NO.:
===== PART: PART_2      (
)=====
DESCRIPTION:
TEX -----
--
PNM      NAME:
PNU PART NO.:
DRA DRAW NO.:
===== DATUM LIST:      (
)=====
DESCRIPTION:
TEX DATUM NAME      TYPE      X      Y      Z
DRF DATUM5      ( ) RECTAN( )      .85000      1.75000      .00000( )
DRF DATUM4      ( ) CYLCEN( )      .75000      1.75000      .00000( )
AXI      ( ) AXIAL( )      .00000      .00000      1.00000( )
DRF DATUM2      ( ) RECTAN( )      .75000      .00000      .00000( )
DRF JOINT3      ( ) CONPLA( )      .62500      .00000      .00000( )
DRF DATUM1      ( ) RECTAN( )      1.00000      .00000      .00000( )
===== JOINT LIST:      (
)=====
DESCRIPTION:
TEX JOINT NAME      TYPE      X      Y      Z
== FEATURE TOL LIST:
=====
DESCRIPTION:
TEX JOINT NAME      PARTNAME      TYPE      TOLERANCE      CHAR.LENGTH      CHAR.LENGTH
TEX -----
-
FEA JOINT3      PART_1      ANG( ) 0.0010000( ) 0.1250000( ) 0.0000000( )
FEA JOINT3      PART_1      TRU( ) 0.0010000( ) 0.1250000( ) 0.0000000( )

```

```

===== LOOP: LOOP_1      (
)=====
DESCRIPTION:
TEX -----
--
TABLE:
TEX -----
--
TEX          VECTOR LOOP
TEX          X0          Y0          Z0          XANGO          YANGO          ZANGO
INI          .8500( )    1.7500( )    .0000( )    .000( )    .000( )    .000( )
TEX PART/J1/J2-JOINT    BAS DIM          MAX TOL          MIN TOL          DEP
NOD DATUM5
ZRO          ( )    180.0000( )    .000000( )    .000000( )    ( )
DIM PART_2/4-5    ( )    .100000( )    .003000( )    -.003000( )    ( )
NOD DATUM4
ZRO          ( )    90.0000( )    .000000( )    .000000( )    DEP( )
DIM PART_2/2-4    ( )    1.750000( )    .005000( )    -.005000( )    ( )
NOD DATUM2
ZRO          ( )    -90.0000( )    .000000( )    .000000( )    ( )
DIM PART_2/2-3    ( )    .125000( )    .003000( )    -.003000( )    ( )
NOD JOINT3
ZRO          ( )    -180.0000( )    .000000( )    .000000( )    ( )
DIM PART_1/1-3    ( )    .375000( )    .005000( )    -.005000( )    ( )
NOD DATUM1
ZRO          ( )    90.0000( )    .000000( )    .000000( )    ( )
DIM PART_1/1-6    ( )    1.500000( )    .005000( )    -.005000( )    ( )
NOD DATUM6
===== END:
=====

```

**Remote Positioner Neutral File**

```

FILE>> REMOTE POSITIONER.DAT                2- 5-93   :
13:38
DESCRIPTION OF DATA FILE:
===== ASSEMBLY: REMOTELA      ( ) 2=====
DESCRIPTION:
TEX -----
-
ANM      NAME:
ANO ASSM NO.:
DRA DRAW NO.:
MOD      LOOP_3      ( ) LOOP_2      ( ) LOOP_1      ( )
===== SPECIFICATIONS:
=====
DESCRIPTION:
TEX A TYPE      REF-JOI/DIM      2ND-JOI/DIM      BAS DIM      MAX DIM      MIN DIM
SPE ORI      ( ) 5/14-18 ( )      ( ) 0.00000( )      2.00000( )      -2.0000( )
===== PART: 5      (
)=====
DESCRIPTION:
TEX -----
--
PNM      NAME:
PNU PART NO.:
DRA DRAW NO.:
===== PART: 4      (
)=====
DESCRIPTION:
TEX -----
--
PNM      NAME:
PNU PART NO.:
DRA DRAW NO.:
===== PART: 3      (
)=====
DESCRIPTION:
TEX -----
--
PNM      NAME:
PNU PART NO.:
DRA DRAW NO.:
===== PART: 2      (
)=====
DESCRIPTION:
TEX -----
--
PNM      NAME:
PNU PART NO.:
DRA DRAW NO.:
===== PART: 1      (
)=====
DESCRIPTION:
TEX -----
--
PNM      NAME:
PNU PART NO.:
DRA DRAW NO.:
===== PART: GROUND      (
)=====
DESCRIPTION:
TEX -----
--
PNM      NAME:
PNU PART NO.:
DRA DRAW NO.:

```

=====  
 ===== DATUM LIST:  
 =====

DESCRIPTION:

TEX DATUM NAME	TYPE	X	Y	Z
DRF DATUM6	( ) RECTAN( )	51.575978	7.0054327	0.0000000( )
DRF DATUM5	( ) RECTAN( )	20.077680	-1.745542	0.0000000( )
DRF DATUM4	( ) RECTAN( )	34.825962	65.317403	0.0000000( )
DRF DATUM3	( ) RECTAN( )	56.825962	65.317403	0.0000000( )
DRF DATUM2	( ) RECTAN( )	51.575978	56.317432	0.0000000( )
DRF DATUM1	( ) RECTAN( )	51.575978	56.317432	0.0000000( )
DAT DATUM15	( ) RECTAN( )	29.575978	7.0054327	0.0000000( )
DAT DATUM13	( ) RECTAN( )	29.575978	56.317432	0.0000000( )
DAT DATUM11	( ) RECTAN( )	29.575978	56.317432	0.0000000( )
DAT DATUM24	( ) RECTAN( )	20.077680	-1.745542	0.0000000( )
DAT DATUM23	( ) RECTAN( )	29.575978	56.317432	0.0000000( )
DAT DATUM22	( ) RECTAN( )	34.825962	65.317403	0.0000000( )
DAT DATUM21	( ) RECTAN( )	56.825962	65.317403	0.0000000( )
DAT DATUM20	( ) RECTAN( )	51.575978	7.0054327	0.0000000( )
DAT DATUM19	( ) RECTAN( )	56.825962	65.317403	0.0000000( )
DAT DATUM18	( ) RECTAN( )	51.575978	7.0054327	0.0000000( )
DAT DATUM17	( ) RECTAN( )	56.825962	65.317403	0.0000000( )

=====  
 ===== JOINT LIST:  
 =====

DESCRIPTION:

TEX JOINT NAME	TYPE	X	Y	Z
JOI JOINT16	( ) REVOLU( )	20.077680	-1.745542	0.0000000( )
AXI	AXIAL ( )	0.0000000	0.0000000	1.0000000( )
JOI JOINT14	( ) REVOLU( )	29.575978	7.0054327	0.0000000( )
AXI	AXIAL ( )	0.0000000	0.0000000	1.0000000( )
JOI JOINT12	( ) REVOLU( )	20.077680	47.566457	0.0000000( )
AXI	AXIAL ( )	0.0000000	0.0000000	1.0000000( )
JOI JOINT10	( ) REVOLU( )	29.575978	56.317432	0.0000000( )
AXI	AXIAL ( )	0.0000000	0.0000000	1.0000000( )
JOI JOINT9	( ) REVOLU( )	34.825962	65.317403	0.0000000( )
AXI	AXIAL ( )	0.0000000	0.0000000	1.0000000( )
JOI JOINT8	( ) REVOLU( )	51.575978	56.317432	0.0000000( )
AXI	AXIAL ( )	0.0000000	0.0000000	1.0000000( )
JOI JOINT7	( ) REVOLU( )	56.825962	65.317403	0.0000000( )
AXI	AXIAL ( )	0.0000000	0.0000000	1.0000000( )

== FEATURE TOL LIST:  
 =====

DESCRIPTION:

TEX JOINT NAME	PARTNAME	TYPE	TOLERANCE	CHAR. LENGTH	CHAR. LENGTH
FEA JOINT10	P1	CLR( )	0.0015000( )	0.0000000( )	0.0000000( )
===== LOOP: LOOP_3 ( ) =====					

DESCRIPTION:

TEX	VECTOR LOOP
ZRO	180.00000( ) 0.000000( ) 0.000000( ) ( )
DIM 2/9-17	( ) 22.000000( ) 0.005000( ) -0.00500( ) ( )
ZRO	59.743557( ) 0.000000( ) 0.000000( ) DEP( )
DIM 3/9-13	( ) 10.419300( ) 0.005000( ) -0.00500( ) ( )
ZRO	30.256443( ) 0.000000( ) 0.000000( ) ( )
DIM 3/13-14	( ) 49.312000( ) 0.005000( ) -0.00500( ) ( )

TABLE:

TEX	VECTOR LOOP	XANG0	YANG0	ZANG0
INI 56.825962( ) 65.317403( ) 0.0000000( ) 0.000000( ) 0.000000( ) 0.000000( )				
TEX NODE   PART/#1-#2 BASIC DIM MAX TOL MIN TOL DEP				
NOD DATUM17				
ZRO				
DIM 2/9-17				
NOD JOINT9				
ZRO				
DIM 3/9-13				
NOD DATUM13				
ZRO				
DIM 3/13-14				

```

NOD JOINT14
ZRO          90.000000( ) 0.000000( ) 0.000000( ) DEP( )
DIM 5/14-18  ( ) 22.000000( ) 0.005000( ) -0.00500( ) ( )
NOD DATUM18
===== LOOP: LOOP_2      (
)=====
DESCRIPTION:
TEX -----
-
TABLE:
TEX -----
-
TEX          VECTOR LOOP
TEX X0          Y0          Z0          XANG0          YANG0          ZANG0
INI 29.575978( ) 56.317432( ) 0.0000000( ) 0.000000( ) 0.000000( ) 0.000000( )
TEX NODE | PART/#1-#2 BASIC DIM      MAX TOL      MIN TOL      DEP
NOD JOINT10
ZRO          59.743557( ) 0.000000( ) 0.000000( ) DEP( )
DIM 3/9-10   ( ) 10.419300( ) 0.005000( ) -0.00500( ) ( )
NOD JOINT9
ZRO          -59.74355( ) 0.000000( ) 0.000000( ) DEP( )
DIM 2/7-9    ( ) 22.000000( ) 0.005000( ) -0.00500( ) ( )
NOD JOINT7
ZRO          -120.2564( ) 0.000000( ) 0.000000( ) DEP( )
DIM GROUND/7-8 ( ) 10.419300( ) 0.005000( ) -0.00500( ) ( )
NOD JOINT8
ZRO          -59.74355( ) 0.000000( ) 0.000000( ) DEP( )
DIM 1/8-10   ( ) 22.000000( ) 0.005000( ) -0.00500( ) ( )
NOD JOINT10
===== LOOP: LOOP_1      (
)=====
DESCRIPTION:
TEX -----
-
TABLE:
TEX -----
-
TEX          VECTOR LOOP
TEX X0          Y0          Z0          XANG0          YANG0          ZANG0
INI 20.077680( ) -1.745542( ) 0.0000000( ) 0.000000( ) 0.000000( ) 0.000000( )
TEX NODE | PART/#1-#2 BASIC DIM      MAX TOL      MIN TOL      DEP
NOD JOINT16
ZRO          42.655000( ) 0.000000( ) 0.000000( ) DEP( )
DIM 5/14-16  ( ) 12.915000( ) 0.005000( ) -0.00500( ) ( )
NOD JOINT14
ZRO          47.345000( ) 0.000000( ) 0.000000( ) DEP( )
DIM 3/10-14  ( ) 49.312000( ) 0.005000( ) -0.00500( ) ( )
NOD JOINT10
ZRO          132.65500( ) 0.000000( ) 0.000000( ) DEP( )
DIM 1/10-12  ( ) 12.915000( ) 0.005000( ) -0.00500( ) ( )
NOD JOINT12
ZRO          47.345000( ) 0.000000( ) 0.000000( ) DEP( )
DIM 4/12-16  ( ) 49.312000( ) 0.005000( ) -0.00500( ) ( )
NOD JOINT16
===== END:
=====

```

**Clutch Neutral File**

```

FILE>> CLUTCH.DAT                               Last revised:   3-27-93   :
11:34:
DESCRIPTION OF DATA FILE:
===== ASSEMBLY: A1
2=====
DESCRIPTION:
TEX -----
--
ANM      NAME:
ANU ASSM NO.:
DRA DRAW NO.:
===== SPECIFICATIONS:
=====
DESCRIPTION:
TEX A  TYPE      REF-JOI/DIM      2ND-JOI/DIM      BAS DIM      MAX DIM      MIN DIM
SPE ANG( )  A/1-2          ( ) E/5-1          ( ) 7.0        ( ) 7.5        ( ) 6.5      (
)
===== PART: E          (
)=====
DESCRIPTION:
TEX -----
--
PNM      NAME:
PNU PART NO.:
DRA DRAW NO.:
===== PART: C          (
)=====
DESCRIPTION:
TEX -----
--
PNM      NAME:
PNU PART NO.:
DRA DRAW NO.:
===== PART: B          (
)=====
DESCRIPTION:
TEX -----
--
PNM      NAME:
PNU PART NO.:
DRA DRAW NO.:
===== PART: A          (
)=====
DESCRIPTION:
TEX -----
--
PNM      NAME:
PNU PART NO.:
DRA DRAW NO.:
===== DATUM LIST:  (
)=====
DESCRIPTION:
TEX DATUM NAME      TYPE          X          Y          Z
DRF J2              ( ) RECTAN( )      .00000      27.64500      .00000( )
DRF J4              ( ) CYLCEN( )      4.79799      39.07500      .00000( )
AXI                 ( ) AXIAL ( )      .00000      .00000      -1.00000( )
DRF J6              ( ) RECTAN( )      .00000      -.00154      .00000( )
===== JOINT LIST:  (
)=====
DESCRIPTION:
TEX JOINT NAME      TYPE          X          Y          Z
JOI J1              ( ) REVOLU( )      .00000      .00000      .00000( )
AXI                 ( ) AXIAL ( )      .00000      .00000      1.00000( )
JOI J3              ( ) CYLSLI( )      4.79799      27.64500      .00000( )

```

AXI	( ) PLANAR( )	1.00000	.00000	.00000( )
AXI	( ) NORMAL( )	.00000	1.00000	.00000( )
JOI J5	( ) PARCYL( )	6.19096	50.41980	.00000( )
AXI	( ) AXIAL ( )	.00000	.00000	.00000( )

```

== FEATURE TOL LIST: (
)=====
DESCRIPTION:
TEX JOINT NAME  PART NAME  FC TYPE  FC TOL  CHAR.LENGTH
TEX -----
--
TEX
TEX JOINT NAME  PART NAME  FEATURE CONTROL LIST
FC TYPE  FC TOL  CHAR.LENGTH
FEA J3  C  FLA( )  .02500( )  .0000( )
FEA J3  C  ROU( )  .00300( )  .0000( )
FEA J5  C  ROU( )  .00300( )  .0000( )
FEA J5  E  ROU( )  .01000( )  .0000( )
===== LOOP: L1  (
)=====
DESCRIPTION:  LOOP1
TEX -----
--
TABLE:
TEX
TEX          X0          Y0          Z0          XANGO          YANGO          ZANGO
INI          .0000( )          .0000( )          .0000( )          .000( )          .000( )          90.000( )
TEX PART/J1/J2-JOINT  BAS DIM          MAX TOL          MIN TOL          DEP
NOD J1
ZRO          ( )          90.0000( )          .000000( )          .000000( )          DEP( )
DIM A/1-2          ( )          27.645000( )          .050000( )          -.050000( )          ( )
NOD J2
ZRO          ( )          -90.0000( )          .000000( )          .000000( )          ( )
DIM B/2-3          ( )          4.797996( )          .000000( )          .000000( )          DEP( )
NOD J3
ZRO          ( )          90.0000( )          .000000( )          .000000( )          ( )
DIM C/3-4          ( )          11.430000( )          .010000(*)          -.010000( )          ( )
NOD J4
ZRO          ( )          -7.0000( )          .000000( )          .000000( )          DEP( )
DIM C/4-5          ( )          11.430000( )          .010000(*)          -.010000( )          ( )
NOD J5
ZRO          ( )          180.0000( )          .000000( )          .000000( )          ( )
DIM E/5-1          ( )          50.800000( )          .012500( )          -.012500( )          ( )
NOD J1
===== END:
=====

```

## MAIN FUNCTIONS WITH SUB-FUNCTIONS -not including Funcs and VecFuncs

### Parts

New Part

    MakePart

        InsertFeatureDatum (PartName, DatumOrigin)

        GetDefaultName

        CheckDupName (PartName)

    CalcAxes (Axis1)

DeletePart

    NewFeatureDatum

### Joints

NewJoint

    MakeJoint

        GetDatumPath

            LinkDatumPath (DatumIDList)

EditJoint

    TraceCompletePath (PrevPos DatumID)

    ShowPath (JntID PathNum)

DeleteJoint

### DOF

DOFcheck

    JntsOnPart (JointsSS PartID PartName)

    CalcDOF (PartName JntIDsOnPart)

        SetUpJntDOF (JntID)

        RemovePartDOF (JntDOF PartDOF JntID Partname)

        DisplayPartDOF (PartDOF)

        DrawDOFarrow

### Features

NewFeatCtrl

    MakeFeatCtrl

        FeatNameFromTag (FeatTag)

        FeatSymFromTag (FeatTag)

EditFeatCtrl

DeleteFeatCtrl

### Specs

NewSpec

MakeSpecs  
    Endpoints (SpecType PartName PrevDRF)  
    DelSpecJnt (JntID1 JntID2)  
DeleteSpec

**Loops**

NewLoop

MakeLoop

GetLoop (ClosedLoop)

CommonPartPaths (Joint1ID Joint2ID CurrentPart ClosedLoop

FirstTime)

RemoveDupPaths (HandleList)

LoopInfo (HandleList PartNameList ClosedLoop)

CheckDep (HandleList ClsoedLoop)

GetDepInOut (PrevJnt Jnt NextJnt)

EdgeorCylSliderDep (PrevJnt JntID NextJntID)

LabelFromEntType (EntType)

HighlightVec (PrevJntPos JntPos HiLite)

DependAngle (EntIDList Entcount)

GetTolerance (TolType Nominal DefaultTol)

HighlightVec (PrevJntPos JntPos HiLite)

StoreLoop (HandleList Tollist PartNameList)

DeleteLoop

EditVecTol

**AutoLoop**

AutoLoops

AddJntToPart (PartName PartsUsed)

PartIndex (PartName PartsUsed)

FindIndex (Handle HandleList)

ExecuteAutoLoop

DrawLoop (HandleList)

LoopInfo (HandleList PartNameList)

**Nuetral File**

CatsFile

Date

Time

WriteSpecs (SpecsSS Chan)

WriteParts (PartsSS Chan)

WriteDatums (DatumsSS Chan)

WriteAxes (EntID EntType)

WriteJoints (JointsSS Chan)

WriteAxes (EntID EntType)

WriteFeatures (FeatCtrlsSS Chan)

WriteClrCntrl (ClrCntrlsSS Chan)

WriteLoops (LoopsSS Chan)

LabelFromEntType (EntID)

ThetaZYCalc (LoaclAxes Vec)

RotateAxes (LocalAxes Vec)  
FindAngDep (JntType)

**Colors**

GetLayerColor

**Main Sub-Functions Alphabetized***-not including Funcs and VecFuncs***AddJntToPart (PartName PartsUsed)**

Tracks number of joints associated with each part in assembly.

**AutoLoops**

Main function called which manually creates open loops. Generates and writes information needed for C program.

**CalcAxes (Axis1)**

Creates XYZ local axis for each datum inserted.

**CalcDOF (PartName JntIDsOnPart)**

Calculates the degrees of freedom removed from each part in assembly.

**CatsFile**

Main function called which obtains database information and generates the model file or compares and reads CATS analysis suggestions.

**CheckDep (HandleList ClosedLoop)**

Obtains information on dependency of each vector in vector loop.

**CheckDupName (PartName)**

Makes sure that each part created has a unique name.

**CommonPartPaths (Joint1ID Joint2ID CurrentPart ClosedLoop FirstTime)**

Determines common part name associated with the first two joints in a vector loop.

**Date**

Obtains the correct date from computer system

**DeleteFeatCtrl**

Deletes all information associated with feature controls.

**DeleteJoint**

Deletes all information associated with joints.

**DeleteLoop**

Deletes all information associated with loops.

**DeletePart**

Deletes all information associated with parts.

**DeleteSpec**

Deletes all information associated with specs.

**DelSpecJnt (JntID1 JntID2)**

Deletes specification endpoints if endpoints are not on two unique parts.

**DependAngle (EntIDList Entcount)**

Calculates if angle between two vectors is dependent.

**DisplayPartDOF (PartDOF)**

Informs user of results from degrees of freedom check.

**DOFcheck**

Main function called which obtains part and joint information needed for degrees of freedom analysis.

**DrawDOFarrow**

Draws direction of unconstrained motion in degrees of freedom analysis.

**DrawLoop**

Draws vector loop

**EdgeorCylSliderDep (PrevJnt JntID NextJntID)**

Checks to see if incoming or outgoing vectors are dependent for edge slider, cylindrical slider or planar joints.

**EditJoint**

Shows datum paths and parts associated with a joint.

**EditVecTol**

Allows user to edit tolerance placed on vector in vector loop.

**Endpoints (SpecType PartName PrevDRF)**

Obtains endpoint information needed for each different specification.

**ExecuteAutoLoop**

Calls C program to automatically generate vector loops.

**FeatNameFromTag (FeatTag)**

Obtains feature control type from menu selection.

**FeatSymFromTag (FeatTag)**

Obtains correct block name to insert for corresponding feature control.

**FindAngDep (JntType)**

Calculate if an angle at a joint between two vectors is dependent.

**FindIndex (Handle HandleList)**

For automatic loop generation obtains unique part number assigned from PartIndex.

**GetDatumPath**

Creates datum path from joint to a mating part.

**GetDefaultName**

Obtains unique default name for each part created.

**GetDepInOut (PrevJnt Jnt NextJnt)**

Obtains dependency of incoming or outgoing vectors for all joints in vector loop.

**GetLayerColor**

Obtains color information necessary for each modeling element's layer.

**GetLoop (ClosedLoop)**

Finds path of vector loop for manual generation.

**GetTolerance (TolType Nominal DefaultTol)**

Obtains tolerances on independent vectors and angles in open and closed loops.

**HighlightVec (PrevJntPos JntPos HiLite)**

Highlights, or creates red line between two points.

**InsertFeatureDatum (PartName, DatumOrigin)**

Inserts feature datums or datum reference frames with required information.

**JntsOnPart (JointsSS PartID PartName)**

Obtains number and types of joints associated with each part in assembly.

**LabelFromEntType (EntType)**

Obtains type of modeling element from entity identity.

**LinkDatumPath (DatumIDList)**

Creates list of datums which show endpoints of vectors on datum path.

**LoopInfo (HandleList PartNameList ClosedLoop)**

Calls subroutines necessary to determine dependency and tolerances of vectors as well as storing the loop information.

**MakeFeatCtrl**

Main function called which creates new feature controls.

**MakePart**

Main function called which creates new parts or datum reference frames.

**MakeJoint**

Main function called which creates new joints.

**MakeLoops**

Main function called which generates manual open and closed vector loops.

**MakeSpecs**

Main function called which creates new design specifications.

**NewFeatCtrl**

Obtains information needed to create feature control.

**NewFeatureDatum**

Obtains information needed to create a feature datum.

**New Part**

Obtains information needed to create datum reference frames or parts.

**NewJoint**

Obtains information needed to create joints.

**NewLoop**

Obtains information needed to generate open or closed vector loops.

**NewSpec**

Obtains information needed to create design specifications.

**PartIndex (PartName PartsUsed)**

For automatic loop generation assigns a unique number to each part

**RemoveDupPaths (HandleList)**

Removes duplicate paths from vector loops.

**RemovePartDOF (JntDOF PartDOF JntID Partname)**

Calculates degrees of freedom for each part.

**RotateAxes (LocalAxes Vec)**

Rotates global axis to correspond to local coordinate system.

**SetUpJntDOF (JntID)**

Obtains actual amount of degrees constrained for each joint type.

**ShowPath (JntID PathNum)**

Shows user a datum path for a joint.

**StoreLoop (HandleList Tollist PartNameList)**

Inserts loop information into database.

**ThetaZYCalc (LocalAxes Vec)**

Returns local axis coordinate system of a joint

**Time**

Calculates the current time from computer system.

**TraceCompletePath (PrevPos DatumID)**

Shows user path for specified joint and part.

**WriteAxes (EntID EntType)**

Calculates axis types and directions corresponding to different joint types.

**WriteClrCntrl (ClrCntrlsSS Chan)**

Obtains database information needed for analysis of manual process variations and writes to neutral file.

**WriteDatums (DatumsSS Chan)**

Obtains database information needed for all datums in assembly and writes to neutral file.

**WriteFeatures (FeatCtrlsSS Chan)**

Obtains database information needed for analysis of feature controls and writes to neutral file.

**WriteLoops (LoopsSS Chan)**

Obtains database information needed for analysis of both open and closed loops and writes information to neutral file.

**WriteJoints (JointsSS Chan)**

Obtains database information needed for analysis of joints and writes to neutral file.

**WriteParts (PartsSS Chan)**

Obtains database information of all datum reference frames and writes to neutral file.

**WriteSpecs (SpecsSS Chan)**

Obtains database information of all specifications and writes to neutral file.