1989

A tolerance analysis and synthesis module for relational geometric modeling environment interface /

Patrick J. Treacy
Lehigh University

A Tolerance Analysis and Synthesis Module

For Relational Geometric Modeling

Environment Interface

by

Patrick J. Treacy

A Thesis

Presented to the Graduate Committee

of Lehigh University

in Candidacy fo the Degree of

Master of Science

in

Mechanical Engineering

Lehigh University
1988

This thesis is accepted and approved in partial fulfillment of the requirements for the degree of Master of Science.

12/12/88
(date)

John B Ochs
Professor in charge

F. Erdogan
Chairman of Department

Acknowledgements

The author wishes to sincerely thank all of the people who aided with my research effort and compiling this thesis. Those deserving of special thanks are first and foremost my parents Arthur and Elizabeth Treacy who gave me the opportunity to begin the study of mechanical engineering and were constantly supportive and understanding of any of my endeavors. I would also like to thank: Dr. Tulga Ozsoy for his input to the project's direction and for his computer graphics expertise; Ms. Nanxin whose work in geometric modelling databases provided the foundation for this project; Dr. Marylu Coviello for her aid in the structuring of the thesis. Finally, I would like to thank Professor John Ochs for without whose experience and encouragement this thesis would not exist.

Table of Contents

List of Figures

List of Tables

Abstract

The problem at issue in tolerance analysis is specifying tolerances at design time that will optimize both the overall cost and quality of the assembly while quantifying the risk of failure to function properly. In order to obtain a higher yield of assemblies performing to specification, a stronger link between design and manufacturing is needed. If process capabilities are readily available to a design engineer at design time, realistic tolerances can be specified that are satisfactory for the design's functional requirements and cost effective to produce. This study is concerned with developing design and tolerance analysis tools. The method of risk assessment developed accepts statistical data on dimensional variability that is inherent in manufacturing.

The computer module to implement this task was developed in conjunction with the a relational geometric model data structure being developed by Ms. Wang. The analysis procedures employed were developed by Oyvind Bjorke. Bjorke has shown how statistical methods can be extended to complex mechanical assemblies. Bjorke clearly demonstrates the use of a variety of vector types that bring statistical methods into mechanical engineering.

Two approaches to tolerance analysis are considered. First, the statistical parameters of an assembly's functional dimension are determined based on component parts tolerances and parameters to determine the percentage of assemblies that will fail to assemble with the specified component tolerances. Second, for a given set of characteristics of a functional dimension, tolerances are assigned to the individual components of the assembly while maintaining the desired functional characteristics. These approaches were implemented in an interactive, menu driven computer module that interfaces with a solid geometric modeling environment.

1. Introduction

1.1. Problem Statement

1.1.1. Variation of component dimensions

When an engineer designs an assembly, the first design is usually a nominal design consisting of idealized dimensions. In a population of manufactured components, variations exist between the actual sizes of the individual parts. It would be unrealistic and extremely costly to attempt to manufacture parts without any variation. The designer, therefore, specifies the limited variations in part dimensions by the use of tolerances such that the assembly of parts performs as intended. The cost of producing and assembling a mechanical system is often directly related to the tolerances assigned to its components. If the imposed tolerances are too tight, the cost of manufacturing increases. If on the other hand, the tolerances are too loose, the percentage of functionally unacceptable assemblies may be too high.

1.1.2. Tool for design stage

In many cases after the design engineer specifies tolerances that will adequately conform to the functional requirement, the problem is turned over to the manufacturing department. It is up to the manufacturing engineer to determine the process or processes that will conform to the specifications set by the design engineer. The design engineer has the responsibility to turn out a product that will meet specifications for cost, government regulation, reliability, etc. The manufacturing engineer on the other hand is concerned with producing the parts within the specified tolerances according to the cost and plant resource constraints placed on him. Very often these two groups disagree about what tolerances are actually needed on the parts, and consequently, specifications are often debated and sometimes even ignored. The industry has seemingly degraded to the design engineers deliberately tightening tolerances more than necessary anticipating that the manufacturing engineer is going to loosen them to meet machining cost and plant resources. The manufacturing engineer may automatically loosen constraints even further because he expects that the design engineer has over-tightened the tolerances. With the presence of a tool that had access to a facility's actual process capabilities at design time, the gap between design and manufacturing could be narrowed, if not closed altogether.

Given the growing use of geometric modeling data as a common data base among design, analysis, and manufacturing, there is no more obvious environment in which to perform tolerance analysis and tolerance synthesis of an assembly of parts. Tolerance analysis examines the tolerances of component parts to determine the variation in a specified functional dimension. Conversely, tolerance synthesis is concerned with obtaining component tolerances based on a specified functional dimension of an assembly.

Assessing the dimensional relationships between component dimensions and design parameters is tedious and often difficult to manually assign. It is this feature relationship between parts that can exist in a geometric modeling data base that we wish to exploit.

1.2. Background literature

The goal of this project was either to develop a method or utilize an existing method of tolerance analysis that would lend itself well to a geometric modeling interface. Several authors have published works in the field of tolerance analysis and optimal tolerance assignment. Some authors have focused on completely manual methods, while others employ some computer methods.

D.B. Parkinson [1,2,3] has published several papers on reliability methods in tolerancing and its implementation in computer-aided design. Parkinson presents the relationship between components in the form of "limit state equations". The most useful analysis scheme in terms of available information is his second moment analysis. M.J. Wozny, et. al. [4] have proposed a system based on a Monte Carlo approach that is based in computer aided design. J.N. Siddall [5,6] presents a mathematical model for optimally allocating tolerances. The most comprehensive theory is advanced by Bjorke [7] based on a relationship between components in an assembly which he calls a "fundamental equation".

Except for that of Wozny, all of the methods mentioned are either manual methods or manual methods partially extended to computer algorithms. The Wozny paper however is vague as to how the method is actually interfaced with the solid model. The most complete method is that developed by Bjorke. Although much of the analysis schemes set forth by Bjorke feature manual work, his relational concept between parts is well suited for implementation with a geometric modeling system.

Several commercial packages are available on the market such as "VSA", developed by Applied Computer Solutions, Inc. The available packages however require reams of input from the user in the form of dimensional data as well as relational data. A user creates the model of the assembly by inputting both the functional requirements and the relationships between parts.

1.3. Approach

1.3.1. Bjorke's method.

The approach taken in this research using Bjorke's method, was chosen for its completeness in relational aspects. Additionally, since the statistical information that is readily available on part variation is seldom more than means and variances, in the method presented by Bjorke, either the actual sample data can be used, or typical machining characteristics can be assumed. Therefore, Bjorke's model can be developed for a wide variety of possible geometries; this paper presents such a development with a number of examples analyzed.

1.3.2. Engineering Design and Analysis Package

The labor involved in generating and analyzing a mechanical or electro-mechanical design can be greatly reduced if the engineer has access to a complete Geometric and Informational Data Base that includes the data and relationships necessary for various analysis procedures. Such design and analysis procedures may feature finite element analysis, tolerance analysis, sheet metal flat pattern layout design, etc. The trend in design philosophy is increasingly in the direction of the total integrated design and manufacturing package with the geometric model as its basis.

Figure 1.1 below details the flow of information in the software methodology. The geometric description would be contained in the GEOMETRIC DATA BASE in which the topology would be further augmented by relational information. Once the geometrical and relational information is in place, any number of the analysis and design schemes mentioned above could be carried out via an APPLICATIONS module. The applications module would have the ability to communicate back to the data base and either fill in any missing information, such as an unknown tolerance, or suggest changes in the design. The segment

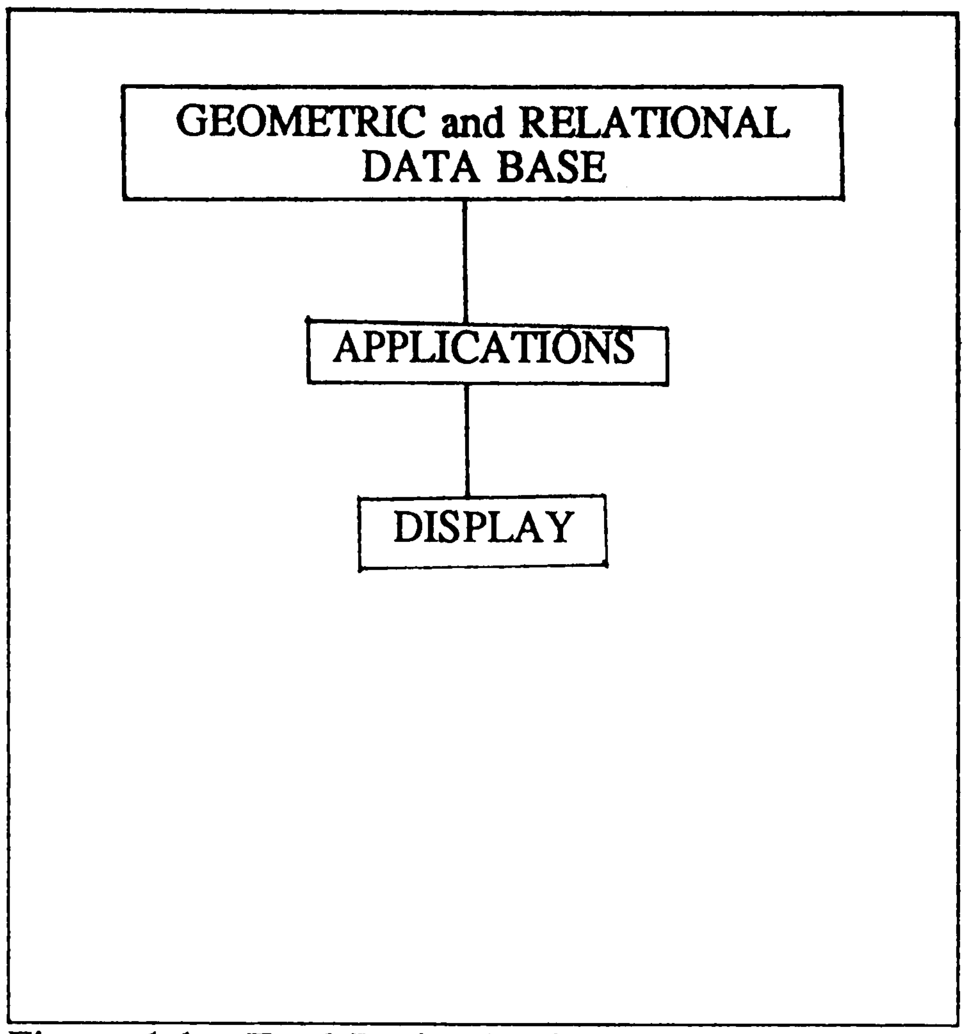

Figure 1.1 Total Design Package

labeled DISPLAY would provide the interface with the user. This segment represents both graphical results of any of the applications modules and the input of design geometry at the geometric model data base stage.

The geometric model data base segment of the proposed package is currently the doctoral work of Wang at Lehigh University. That research purports to "develop a feature oriented modeling strategy which will allow the user to design part components with features, specify datum references, tolerances and mating features, and automatically assemble components according to the mating features and mating conditions, with tolerances analyzed".[8] The development of a tolerance analysis and synthesis applications module that interfaces with Ms. Wang's work is the focus of this thesis.

1.3.3. Geometric model interface.

The manual work necessary to use Bjorke's method is almost totally eliminated by interfacing with the geometric modeling data base that is currently the doctoral work of Ms. Wang. The geometry of the assembly to be analyzed would be created and edited in the geometric modeling package, with the functional relationships between the parts generated automatically. The data present in the geometric modeling data base with the design constraints could be used to establish the functional relationships of the components, thus relieving the manual tedium present in existing methods. The interface created has the capability of transferring geometrical information between the data base and the analysis package. Any tolerances assigned by the analysis package would be used to update the data base.

1.3.4. Computer implementation

The package designed is a menu driven, interactive computer program. The software allows the user to interact with several data files and supports the human interface. The program was written in VAX FORTRAN, and can run on any of Digital Equipment Corporation's VAX computers. Results of an analysis are displayed graphically using the graphics package Graph3d.LU, which is a set of FORTRAN callable graphics subroutines developed at Lehigh University's Computer Aided Design laboratory by Dr. Tulga Ozsoy [9]. The graphics applications routines were written to run on Tektronix 41xx series of terminals.

1.4. Organization of thesis

The remaining chapters of the thesis will discuss the implementation of Bjorke's theory in the menu driven framework outlined above and some examples. Chapter 2 will review Bjorke's approach to analyzing tolerances on an assembly as well as the methods he sets forth for tolerance assignment. Chapter 3 will detail the implementation of the scheme into a computer algorithm as well as describe some of the concepts to situations that Bjorke does not address. In Chapter 4 example assemblies will be used to illustrate the package's capabilities and its operation; some of the examples involve case studies along the lines of a "what if" analysis. Since Tolerance analysis in CAD is genuinely an open field with considerable potential, conclusions and recommendations for future work will be reviewed in Chapter 5.

2. Bjorke's Method

2.1. Tolerance Chains

2.1.1. Fundamental equations and sum dimensions

The sum or functional dimension of a mechanical assembly is a particular dimension whose size is critical to the satisfactory operation of the assembly. The variation in the sum dimension is affected by the variation of other dimensions in the assembly. Consider the simple example of a bore and a shaft in a journal bearing assembly in Figure 2.1.

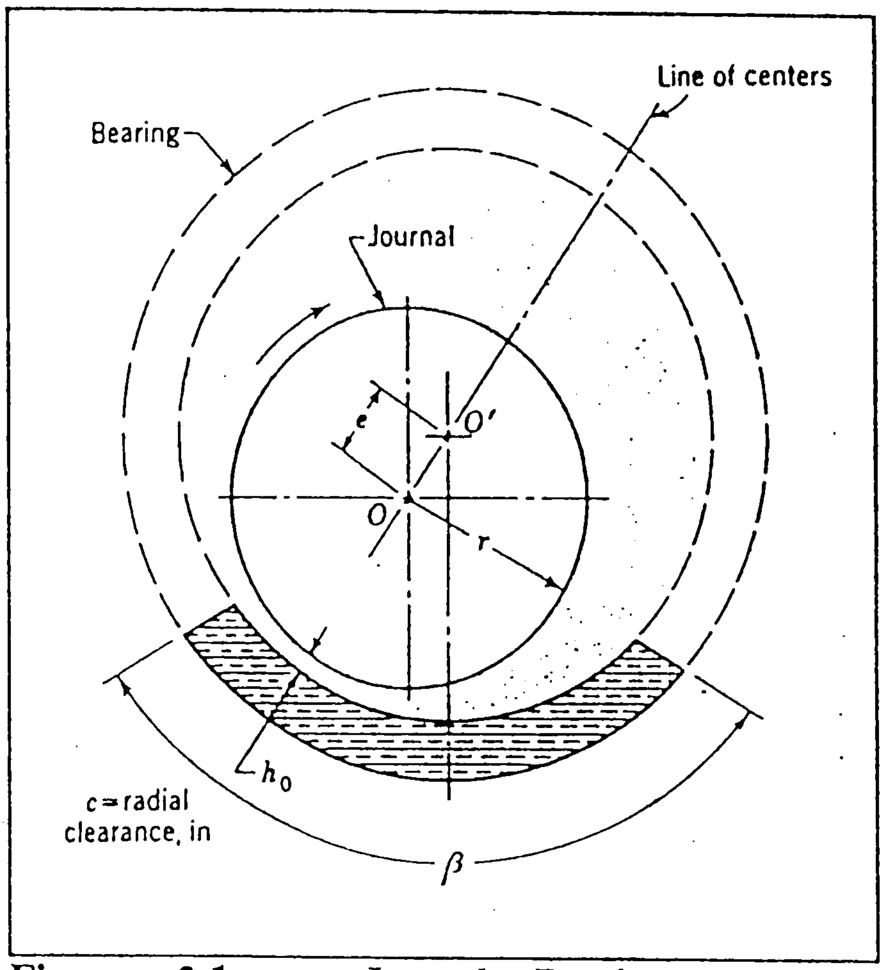

Figure 2.1 Journal Bearing Assembly (reproduced from Shigley [14]).

For this example, the size of this particular journal and bore can vary slightly without failing from the load placed on it. Therefore, it has been decided that the clearance between the bore and the shaft is of critical importance, not so much the actual sizes of the bore and the journal. The sizes of the bore and the journal can vary as long as the clearance between them, the sum dimension, remains within the specified range.

The sum dimension is not only a function of the size of the other dimensions in the assembly, but also their relative locations. Examples of this will be seen later.

As was seen above, the sum dimension, denoted $X\Sigma$, is a function of the other dimensions in the assembly.

$$X\Sigma = f(X_i)$$

The mathematical representation of the relationship between dimensions is called the *fundamental equation.* In the case of the journal bearing assembly, the fundamental equation would be,

$$X\Sigma = X_b - X_a$$

with X_a being the dimension of the shaft and X_b the dimension of the bore.

In many assemblies there is more than one functional requirement and therefore more than one fundamental equation is needed. In the example given above, the function of the fundamental equation is linear, but it must be noted that this is not always the case. Bjorke likens a linearized fundamental equation to the concept of a chain, which he calls a *tolerance chain*.

2.2. Model of individual dimensions

Before the model of the individual dimensions can be discussed, some terminology associated with Bjorke's approach must be presented. The variable names used in this thesis will be the same as those used by Bjorke to maintain consistency with the figures he uses in the development of the method. The basis of the study undertaken is to predict the effect of variation in parts' dimensions on the final assembly. This variation in a part's size is a continuous region, called its tolerance zone, ranging from the part's minimum size to the part's maximum size. Several notations exist to describe a part's tolerance zone, some of which are shown below in Figure 2.2. The basic or nominal size of the part is denoted as "BX".

For most of the tolerance calculations limits on the dimension will be know an "LX" for the minimum dimension limit and "UX" for the maximum dimension limit. The tolerance zone will be reffered to as "TX".

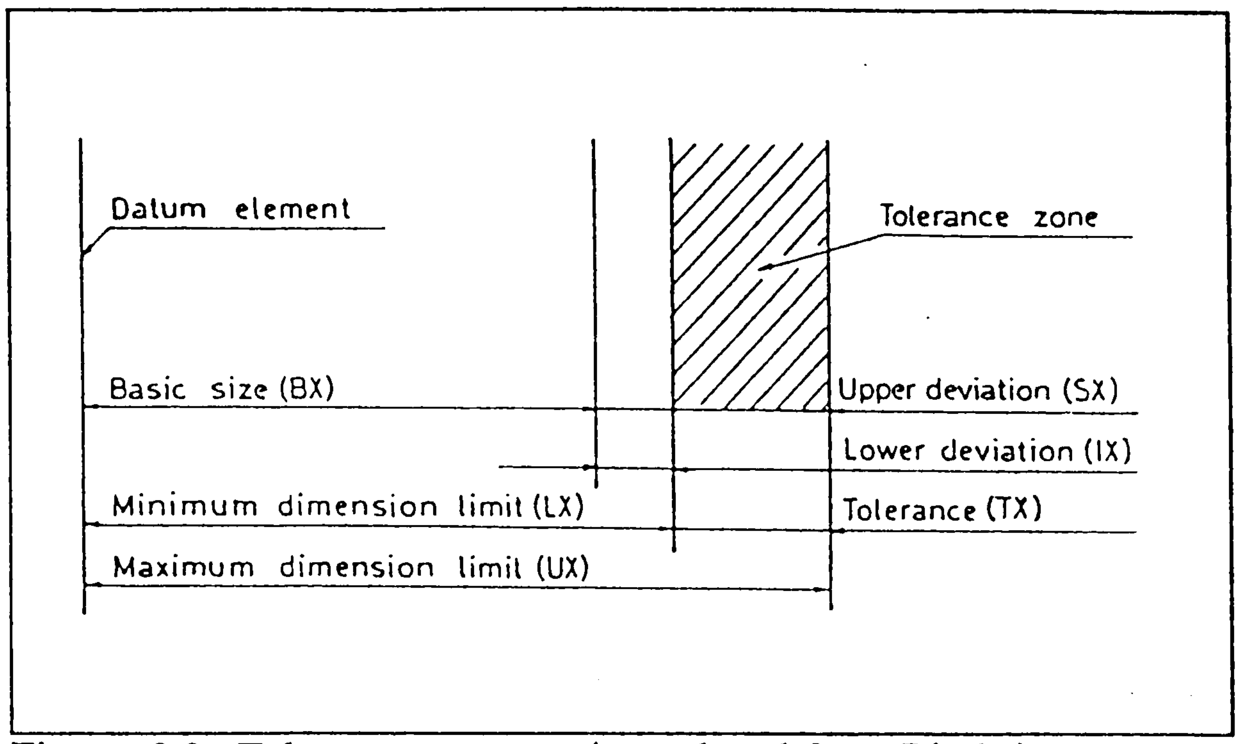

Figure 2.2 Tolerance concepts (reproduced from Bjorke).

A hole might be dimensioned as shown below in Figures 2.3 and 2.4.

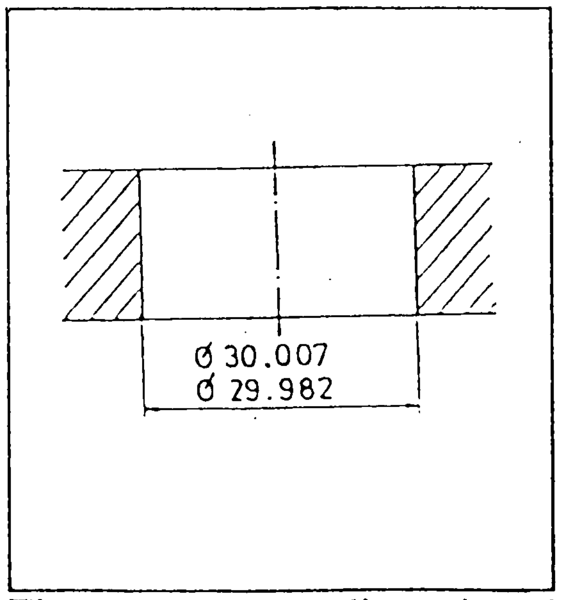

Figure 2.3 Hole dimensioned by limits (reproduced from Bjorke).

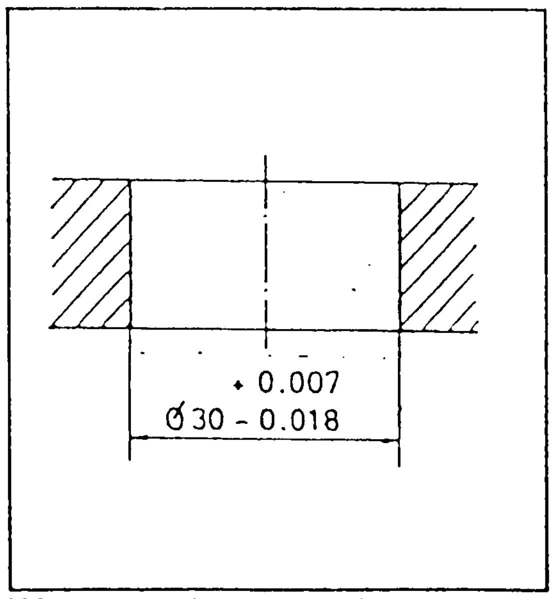

Figure 2.4 Hole dimensioned by basic size and upper and lower deviations (reproduced from Bjorke).

Figure 2.3 denotes the dimension by listing the holes upper and lower limits, while Figure 2.4 graphically presents the hole's basic size and its upper and lower deviations. ANSI and ISO standards also exist for the dimensioning of holes and shafts.

The range of variation in a part's size is generally small as compared to the part's basic or nominal size. The approach taken by Bjorke to represent the variation in a part's size is to break down the part into a constant element and a stochastic element. The convention dictates the use of the middle of the range of variation as the constant part of the dimension. A dimension X is represented as

$$X = MX + \Delta X \qquad \text{(eq. 2.1)}$$

MX : distance to the middle of the tolerance zone

ΔX : the stochastic part of the dimension

The two statistical parameters most important to the development of the model are the expectation and the variance.

$$E[X] = E[MX + \Delta X] \qquad \text{(eq. 2.2)}$$

$$EX = MX + E\Delta X$$

$$var[X] = var[MX + \Delta X]$$

$$varX = var\Delta X \qquad \text{(eq. 2.3)}$$

To determine these statistical parameters, the distribution of X must be known; alternatively, the means and variances themselves could be determined by estimation theory from the past experience of a particular process or machine. Machines being used in a particular factory could be monitored to determine the means and variances of the parts it is producing.

Bjorke lists a table of traditional machining processes and tolerance grades accompanied by their parameters (see Appendix I). It is known that most machining processes vary between a normal and rectangular distribution. Therefore, if the distribution is not known, and sample data is not available from previous part lots, a rectangular distribution could reasonably be used as a worst case example.

Because of quality control efforts, the range of the parts entering the assembly is not always indicative of the range being produced by the machining process (RX). The quality control inspector maintains specifications of the dimensions that can enter the assembly. Parts that agree with the specifications (TX) are allowed to enter an assembly; those that do not are returned to be re-machined or are scrapped. The distribution of the dimensions entering assembly phase are cropped at the dimension limits specified. Dimensions of the parts entering the assembly have a range equal to the tolerance of the parts.

$$RX = TX \qquad \text{(eq. 2.4)}$$

Bjorke's model of the individual dimensions is illustrated below in Figure 2.5. It is obvious that it is desired to use a machine with a process capability (PC) equal to or less than the specified tolerance.

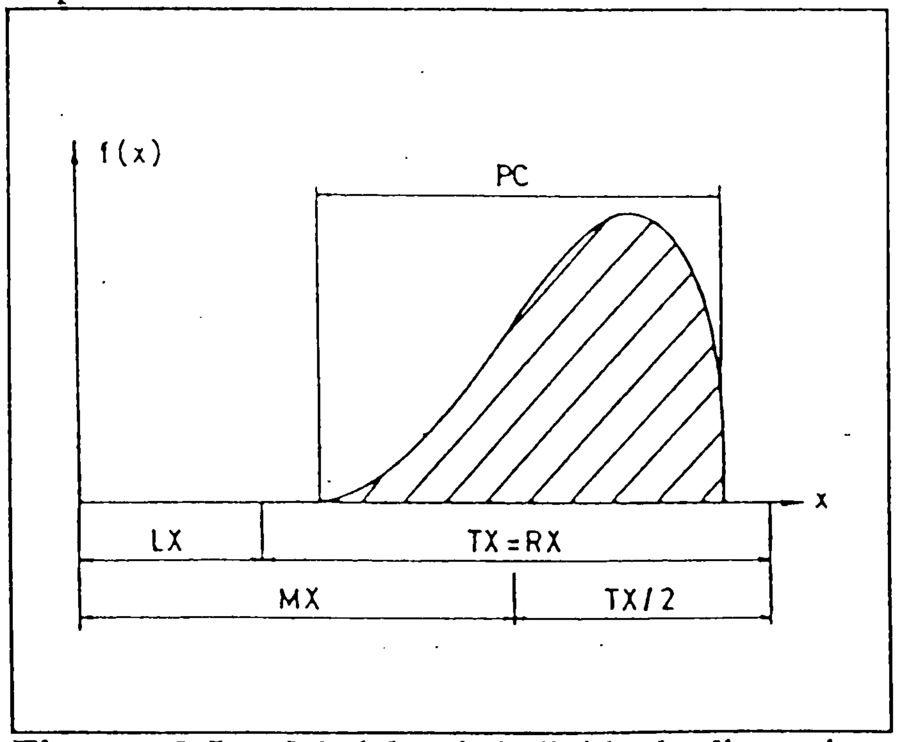

Figure 2.5 Model of individual dimensions (reproduced from Bjorke).

2.2.1. Determining parameters of the individual dimensions

Having an understanding of the model of the individual dimensions, it is time to determine the actual parameters' behavior of the sum dimension. It would require an infinite number of entries to tabulate the expectations and variances for every dimension in the various processes. Bjorke has developed a scheme by which to "unify" the tolerance capability of manufacturing processes. To do so a unit distribution is used. The distribution of the individual dimension is transformed such that its minimum dimension limit is at the origin, and its maximum dimension limit is at 1. The unit dimension variable is z which has a range from 0 to 1. The equations are listed below and an example follows to illustrate this concept.

$$\Delta X = TX(z-0.5) \qquad \text{(eq. 2.5)}$$

Example:

Length X has been assigned a dimension 10.0±0.5. TX in this case is 1.0, and the middle of the tolerance zone M_x is 10.0. From equation 2.2

$X = 10.0 + \Delta X$

From the tolerance zone it can be seen that ΔX has from -0.5 to 0.5. Table 2.1 below created from equation 2.5 illustrates the transformation from the z domain to the X.

Table 2.1 Domain transformations.

z	TX	ΔX	X
0.0	1.0	-0.5	9.5
0.5	1.0	0.0	10.0
1.0	1.0	0.5	10.5

Taking the expectation and variance of equation 2.5 the statistical parameters of the unit distribution are obtained.

$$E[\Delta X] = E[TX(z - 0.5)]$$

$$E[\Delta X] = TX(Ez - 0.5) \qquad \text{(eq. 2.6)}$$

$$var[\Delta X] = var[TX(z - 0.5)]$$

$$var[\Delta X] = var[TX(z)]$$

$$var[\Delta X] = TX^2 varz \qquad \text{(eq. 2.7)}$$

If the statistical parameters EX and varX for the individual dimensions are not known from experience, they may be calculated from the predetermined parameters of the unit dimensions.

The parameters of the unit distribution of widely used processes are taken from Bjorke and listed in Appendix I. They are listed according to form element and technological process. When the form element and process are determined, the tolerance needed is selected from the limits listed for the process. The expectation and variance can be linearly interpolated for the tolerance specification. These parameters, along with the transformation equations 2.6 and 2.7, yield the statistical parameters of the individual dimensions.

2.3. Sum dimensions

2.3.1. Statistical parameters

The statistical parameters of the sum dimension must be determined from the parameters of the individual dimensions. Because the individual dimensions are stochastic variables, the sum dimension will also be stochastic. The sum dimension is not dependent on a manufacturing process, but rather on the assembly of a random sampling of the components whose parameters are process dependent.

Statistical parameters of the individual dimensions have been calculated in the local coordinate system of the individual parts. When the parts are placed in the assembly, they are placed in the global coordinate system of the sum dimension. The effect of the individual part's dimensions on the sum dimension must be determined. Individual dimensions must be transformed according to direction and also according to location. An individual part's influence is transformed to the sum direction and also to the coordinate system of the sum direction. The influence of an individual dimensions is represented by the linear equation:

$$X\Sigma = \Sigma\, A_i\, X_i \qquad \text{(eq. 2.8)}$$

A_i: signed constant

With this relationship, the range that the sum dimension follows to be,

$$RX\Sigma = \Sigma\, |A_i|\, RX_i \qquad \text{(eq. 2.9)}$$

and the distance to the middle of the range of the sum dimension:

$$MX\Sigma_R = \Sigma\, A_i\, MX_i \qquad \text{(eq 2.10)}$$

By substituting equation 2.2 into 2.8, the sum dimension can be broken down into its constant part and its stochastic part.

$$MX\Sigma_R + \Delta X\Sigma = \Sigma\, A_i(MX_i + \Delta X_i) \qquad \text{(eq. 2.11)}$$

Equation (2.10) is subtracted from (2.11) to give the fundamental equation of the stochastic part of the sum dimension.

$$\Delta X\Sigma = \Sigma\, \Delta X_i$$

It follows that the expectation and variance of the stochastic part of the sum dimension is:

$$E\Delta X_\Sigma = \Sigma\, A_i\, E\Delta X_i \qquad \text{(eq. 2.12)}$$

$$\mathrm{Var}\Delta X_\Sigma = \Sigma\, A_i^2\, \mathrm{Var}\Delta X_i \qquad \text{(eq. 2.13)}$$

The statistical parameters of the sum dimension are thus given by equations (2.9), and substituting (2.1) into (2.10), and finally by substituting (2.6) into (2.12) and (2.7) into (2.13).

$$MX_{\Sigma R} = \Sigma\, A_i\, MX_i \qquad \text{(eq. 2.14)}$$

$$R\Delta X_\Sigma = \Sigma\, |A_i|\, TX_i \qquad \text{(eq. 2.15)}$$

$$E\Delta X_\Sigma = \Sigma\, A_i\, TX_i\, (Ez_i - 0.5) \qquad \text{(eq. 2.16)}$$

$$\mathrm{Var}\Delta X = \Sigma\, A_i^2\, TX_i^2\, \mathrm{Var}z_i \qquad \text{(eq. 2.17)}$$

2.3.2. Model of sum dimension

Equations 2.14 - 2.15 show that the parameters of the sum dimension can be computed without knowing the actual distributions of the individual dimensions. The distribution of the sum dimension however must be known in order to determine the confidence level to which the parts will assemble. Determining the distribution of the sum dimension from actual distributions is not practical for two reasons. Many times the distribution assigned to a process is only approximate, and if the distributions were known exactly, the calculations would become unwieldy. Since only means and variances are summed, and not the actual distributions of the individual parts, the distribution of the sum dimension must be satisfactorily approximated to yield confidence levels of assemblies.

Many techniques employ of the central limit theorem, which states that the sum distribution approaches the normal distribution as the number of individual dimensions increases, independent of distributions of the individual dimensions.

One drawback of this method is the requirement of a large number of dimensions for the above assumption to hold true. The normal distribution, therefore, was not used to model the sum dimension for the work covered in this study. For a more detailed discussion of how to approximate the sum dimension as a normal distribution, refer to Bjorke[7].

The only real advantages to approximating the sum dimension as a normal distribution is the simplicity of the calculations involved. The disadvantages far outweigh its benefits for it to be used as a viable model. The normal distribution is only a two parameter distribution, those parameters being the mean and variance. The use of the normal distribution does not allow for the sum dimension to be asymmetrical, nor does it allow for the movement of the tolerance zone within the range of the sum dimension. Last, as mentioned above, the model is not sufficiently accurate for a tolerance chain comprised of few dimensions.

The approach taken in this research work to be covered here is to employ the beta distribution. As a four parameter distribution, the beta distribution model is more flexible and does not encounter many of the problems of the normal distribution referred to above. Greater flexibility of the beta distribution permits it to better approximate the actual distribution of the sum dimension. Changes in the parameters in the beta distribution, can cover actual distributions ranging from the rectangular distribution to normal distribution which are shown in Figure 2.6.

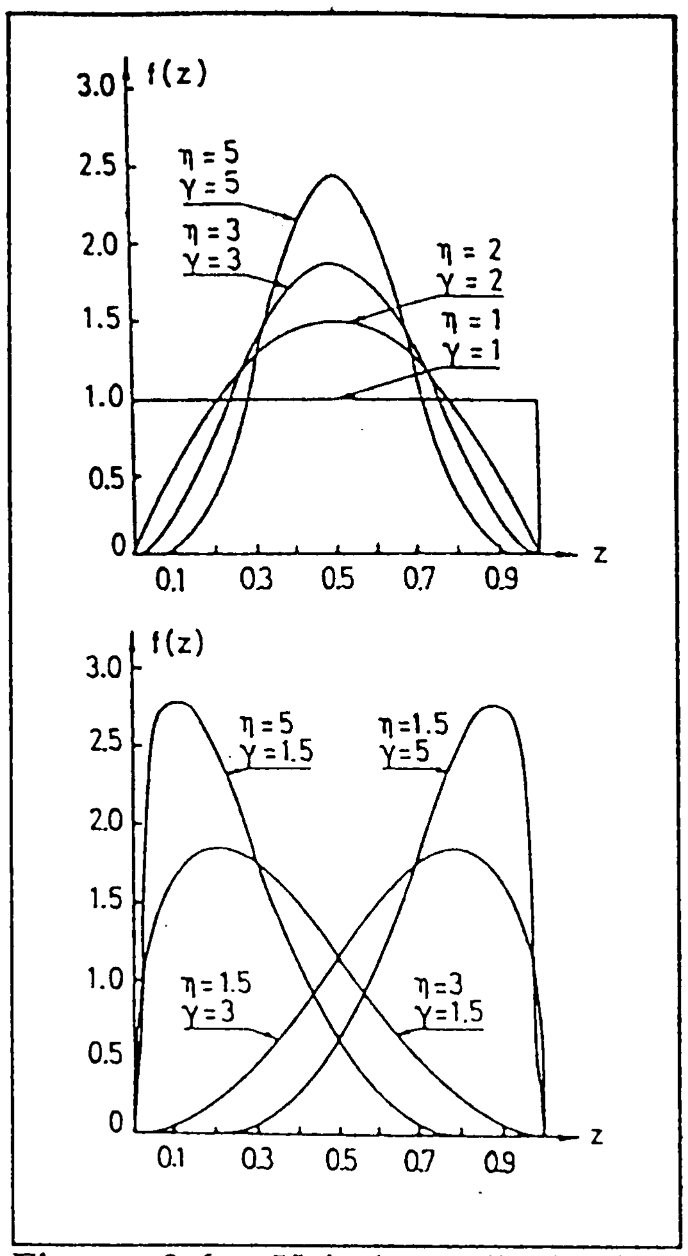

Figure 2.6 Unit beta distributions (reproduced from Bjorke).

The parameters of the beta distribution allow for the sum dimension to be skewed, thus more realistically modeling actual manufacturing experience. The distribution is also a finite distribution, which allows an integration to be performed along the entire range. Though the drawbacks of the normal distribution model are addressed by the beta distribution model, the four parameter beta distribution model requires greater computation.

Figure 2.7 illustrates of the beta distribution model of a sum dimension. It can be seen that the confidence area, which is represented by the cross hatched area, dictates the size of the tolerance zone TX_Σ. It can also be seen that

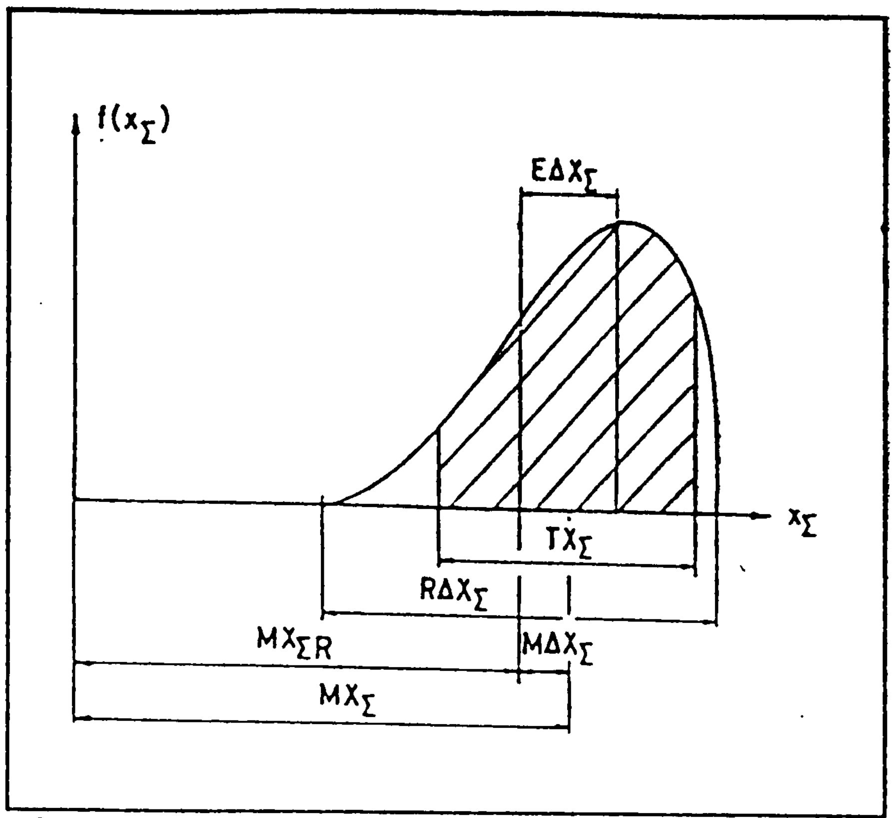

Figure 2.7 Beta distribution model (reproduced from Bjorke).

the tolerance zone is free to move within the range of the sum dimension. This movement, of the tolerance zone, $M\Delta X_\Sigma$, establishes a relationship between the confidence level and the middle point of the tolerance zone. Two different middle point and tolerance zone pairs could yield an identical confidence level.

Bjorke presents a method to determine the relationship between tolerances and confidence levels using normalized dimensions.

$$TX_\Sigma = TW_\Sigma \, (\mathrm{var}\Delta X_\Sigma)^{1/2} \qquad \text{(eq. 2.18)}$$

$$M\Delta X_\Sigma = MW_\Sigma \, (\mathrm{var}\Delta X_\Sigma)^{1/2} \qquad \text{(eq. 2.19)}$$

A normalized beta distribution has an expectation of zero and the variance is 1.0. Tabulated values of normalized beta distribution parameters are listed in Appendix II. The four parameters needed to model the sum dimension are expectation, variance, range, and asymmetry. By tabulating normalized middle point movements along with normalized tolerance zones, values of the actual tolerance zone and its location can be determined from equations (2.18) and (2.19). The method of using tables to determine the tolerance zone of a sum dimension based on a desired confidence level will not be discussed in further detail. If a more in depth explanation is desired, it is suggested that the reader refer to the Bjorke reference. Implementation of the beta distribution to model the sum dimension for a tolerance analysis does not make use of the tabulated values. Instead, the beta distribution is directly integrated using the known parameters to determine the confidence level of the assembly.

The above discussion features two methods of modeling the sum dimension. Choosing between either method is based on a compromise between computing precision and computing effort. Justification of using the beta distribution over the normal distribution lies in the beta distribution's greater precision. As Bjorke argues,

> A sensitive parameter in the models is the normalized tolerance. In Figure 2.8, this parameter is drawn as a function of range, where the full drawn lines belong to the beta distribution model, and the dotted lines belong to the normal distribution model. The curves have been drawn with different confidence levels, and the influence of asymmetry is given by the hatched areas. The conclusion drawn from this figure is that the difference between the models increases with decreasing ranges, and increases with increasing confidence levels, either one or both models produce large computing errors.
>
> The exact tolerances in Figure 2.8 can only be given for the real distributions of the sum dimensions, but these are not known. On the other hand, something can be said in general about the precision of normalized tolerances, and this will be done in the following discussion. As stated in the foregoing, the beta distribution model is more flexible than the normal distribution model. It is, therefore, generally true that a beta distribution is more able to approximate a given distribution than is a normal distribution. This is especially true in our cases since all individual dimensions have distributions with finite ranges.
>
> Another indication of the errors in Figure 2.8 is given by the 100 % confidence level line. This line represents the upper limit on the tolerance as a function of range. Consequently all estimates of tolerances above this line are definitely wrong.

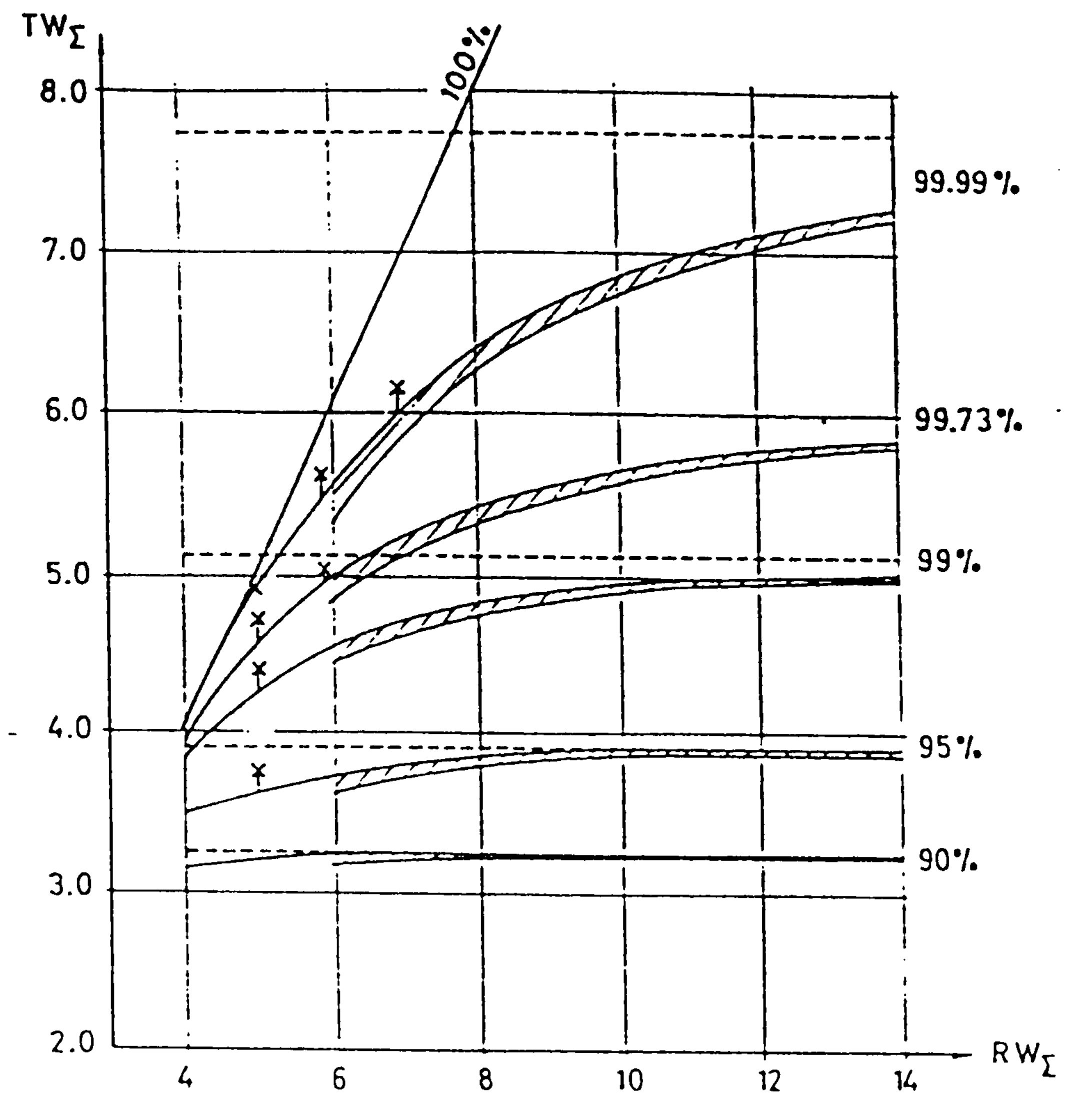

Fig. 2.7. Normalized tolerance as function of normalized range
------ Normal distribution model
——— Beta distribution model
x Exact values found by convolution of rectangular distribution

Figure 2.8 Normalized tolerance as a function of normalized range (reproduced from Bjorke).

Although the exact distributions of sum dimensions are not known in general, we may calculate the exact distributions in some special cases by the convolution integral . This has been done under the assumption of rectangular distributions. The exact tolerances when two, three or four equal rectangularly distributed dimensions have been summed up, are given by the crosses in Figure 2.8. It can be seen from the figure that the normal distribution model gives large errors in these special cases while the errors given by the beta distribution model are small. As an example, the errors on the 99.73% level are given in Table 2.2. On the 99.73 % level the normal distribution model gives errors greater than 5 % until more than eleven rectangular distributed dimensions have been summed up.

Table 2.2 Computing error by summing up equal rectangular dimensions (reproduced from Bjorke).

number of dimensions	normal distribution model	beta distribution model
2	29.7 %	3.0 %
3	18.5 %	2.0 %

It has been concluded, therefore, that the beta distribution is generally a more precise model than the normal distribution. The beta distribution model can be used in all cases, without the risk of losing accuracy when few dimensions are present in the analysis. In conclusion it can be noted that, although the beta distribution requires slightly more computing effort and book keeping because of its four parameters, its advantages exceed its disadvantages.

2.4. Chain Links

2.4.1. Link characteristics

The influence of the individual dimensions on the sum dimension is calculated by use of chain links. A chain link is a dimension whose size and location have an effect on the sum dimension and, which can be comprised of more than one "sub-dimension". Geometry of the chain links are divided into two categories. A span is a dimension between surfaces on a part. A gap is a dimension between the surfaces on mating parts such as a shaft and a bore. A chain link that is a gap is comprised of the two parts that create the gap.

Chain links must also be classified according to their direction in reference to the sum dimension. A chain link is called a line vector if its dimension and variation lie completely in or parallel to the direction of the sum dimension. A chain link is said to be a plane vector if it completely lies within the plane of the sum dimension or a plane parallel to it. If these criteria are not met, the link is referred to as a space vector. Space vectors are further broken down into components that are parallel and perpendicular to the plane of the sum dimension. The perpendicular component has no influence on the sum dimension, and the component in the parallel plane is treated as a plane vector.

All of the categories of vectors can be seen in Figure 2.9. In the left most column are the vector types. In the figure, plane vectors are broken down into two subcategories. Plane vector chain links are said to have either lumped or distributed direction. A link with lumped direction has a given but uncertain angle to the sum direction. A link with distributed direction has a direction that is completely unknown; i.e., it can have any angle with respect to the sum dimension.

Along the top of the figure are the classifications of spans and gaps. Gaps are divided into lumped and distributed magnitude. A gap with lumped magnitude has a given but uncertain value, whereas the magnitude of a gap with distributed

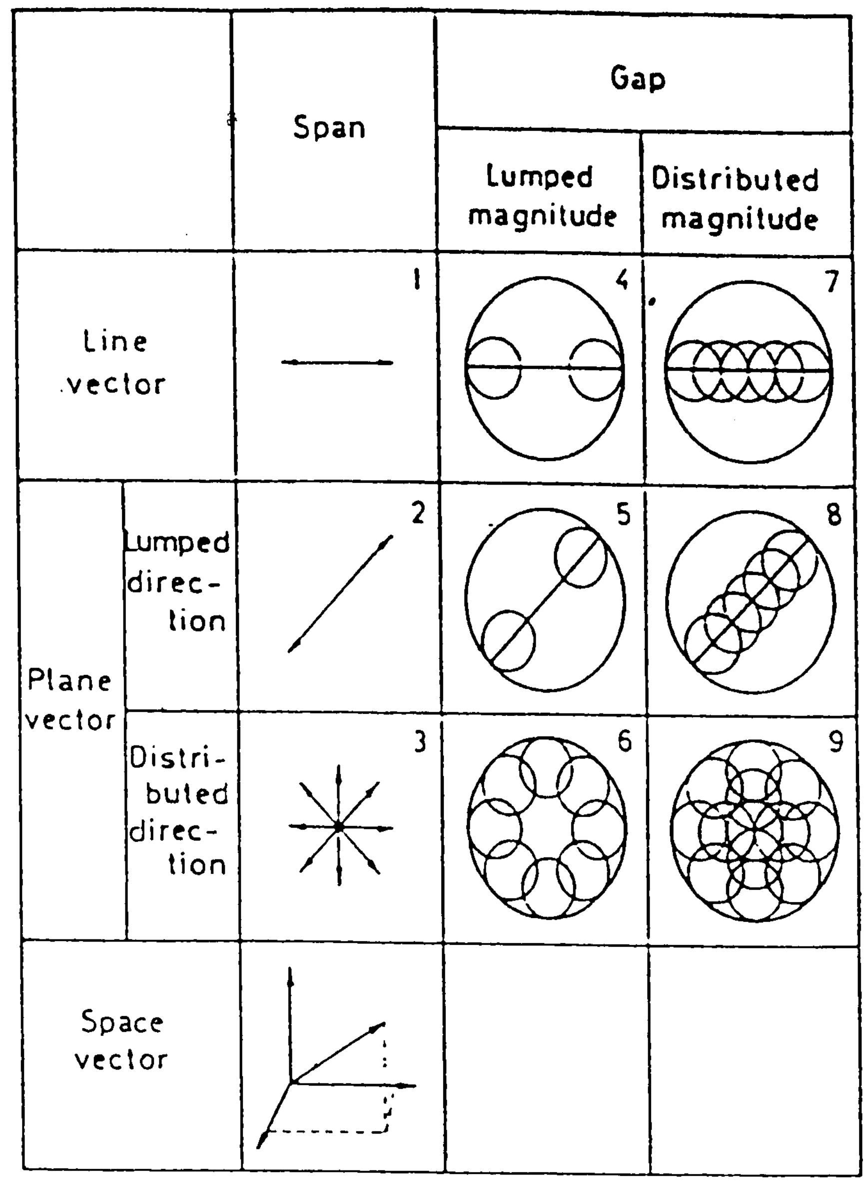

Figure 2.9 Classification of chain links (reproduced from Bjorke).

can not be determined. An example of a gap with lumped magnitude is the gap between a shaft and a bore when they are touching. If the shaft is touching one of the sides of the bore, then it is known that the size of the gap is the diameter of the bore minus the diameter of the shaft. Since the diameters themselves are stochastic, the dimension of the gap, which is given by the relationship between the shaft and the bore, is unknown. If the shaft were not against the bore, but located anywhere within, the size of the gap would be completely uncertain.

It is standard design practice to give a dimension a specified clearance. The gap types illustrated in Figure 2.8 do not account for fit and clearance specifications. Link equations will also be developed for clearances and transitions. Interferences, or fits, always form a "negative gap" and do not have any effect on the sum dimension.

The influence of location of a chain link on the sum dimension must also be taken into account. A signed constant "A_i", as in equation 2.8, is introduced to transform the chain link's influence from its local coordinate system to the coordinate system of the sum dimension. A chain link is positive when an increase in magnitude increases the sum dimension and therefore A_i is positive. For any chain link whose increase in magnitude decreases, the sum dimension is said to be negative, with A_i correspondingly negative. The location of a chain link may be such that its correspondence with the sum dimension may not be one-to-one. The magnitude of the constant A_i is therefore representative of the influence on the sum dimension due to a change in the chain link.

2.4.2. Link types

Link routines have been developed for fourteen different chain links. These routines transform the statistical parameters from the chain links to the sum dimension as specified in the previous sections. A sum dimension could be made up of any number and type of chain link. Equations are listed as if the fundamental equation were made up solely of the type of link being developed.

2.4.2.1. Spans

The simplest type of link with respect to equation development is the line vector span. A line vector span has all of its influence on the sum dimension lying in the axis of the sum direction or an axis parallel to it. The fundamental equation for a line vector span is:

$$X\Sigma = A\ X$$

The equations of a line vector span are:

$$MX\Sigma_R = A\ X$$

$$R\Delta X\Sigma = |A|\ TX \qquad \text{(eq.s 2.20)}$$

$$E\Delta X\Sigma = A\ TX\ (Ez - 0.5)$$

$$\mathrm{var}\Delta X\Sigma = A^2\ TX^2\ \mathrm{varz}$$

Plane vector spans can exist as either lumped direction or distributed direction. A plane vector span with lumped direction forms a given but uncertain angle with the sum direction (Figure 2.10).

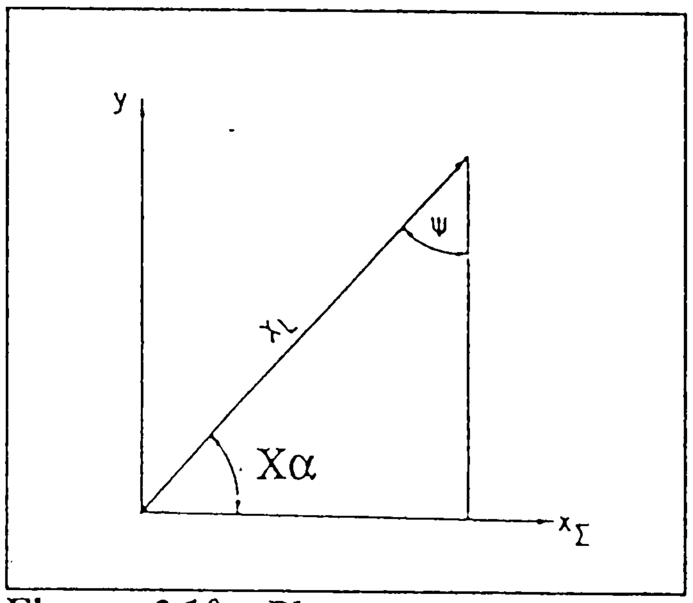

Figure 2.10 Plane vector span with lumped direction (reproduced from Bjorke).

The fundamental equation of a plane vector span with lumped direction is:

$$X\Sigma = A\ X_L \cos(X\alpha) \qquad \text{(eq. 2.21)}$$

X_L: length of span

$X\alpha$: angle of span with sum direction

Equation (2.21) is not a linear equation because the length X_L and the angle $X\alpha$ are both random variables. To linearize the equation, it will be assumed that the angular variation is small as compared to the variation in the length. The influence of a plane vector span on the sum dimension can be calculated by using the following equations.

$$MX\Sigma_R = A\cos(\alpha)\, MX_L$$

$$R\Delta X\Sigma = |A\cos(\alpha)|\, TX_L \qquad \text{(eq.s 2.22)}$$

$$E\Delta X\Sigma = A\cos(\alpha)\, TX_L\, (Ez_L - 0.5)$$

$$var\Delta X\Sigma = A^2 \cos^2(\alpha)\, TX_L^2\, varz_L$$

The last type of span to be presented is the plane vector span with distributed direction. This type of link is referred to as an eccentricity. The direction of an eccentricity has no preference. The development of the eccentricity link type is fairly elaborate, and some of the concepts are imperative in the development of other link types. The development is taken directly from Bjorke [7] and can be found in Appendix III. The equations for a plane vector span with distributed direction are listed below, with some of the important concepts and results summarized. An illustration of a typical eccentricity is shown here in Figure 2.11 to aid the visualization of the concept.

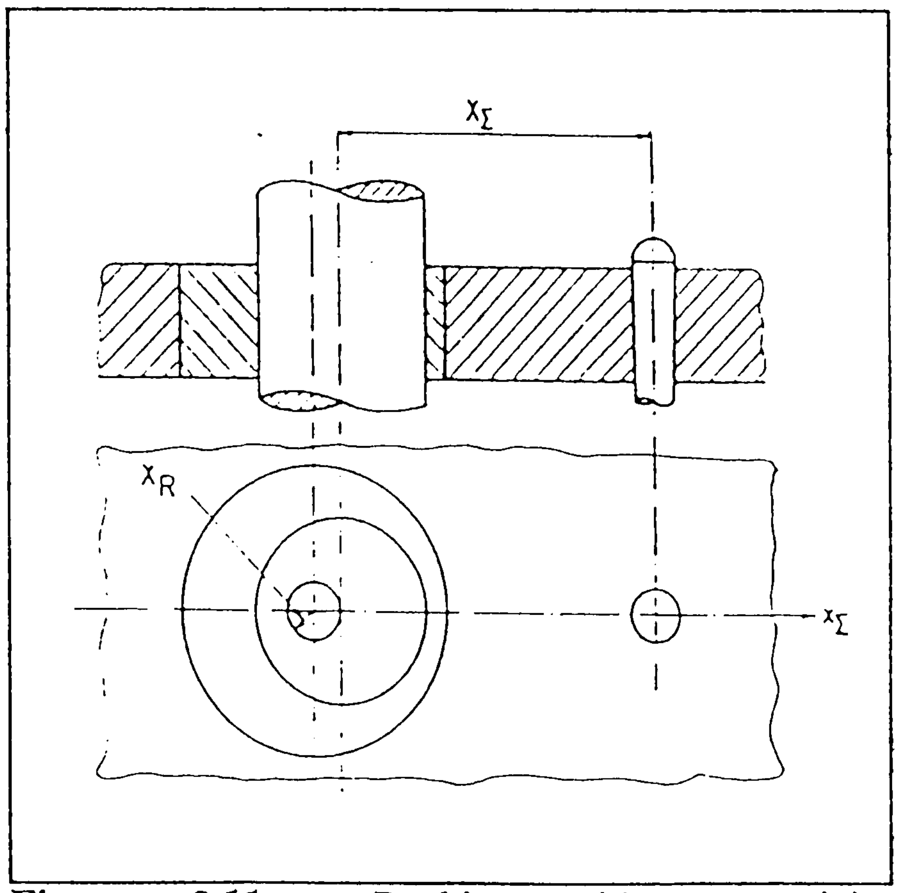

Figure 2.11 Bushing with eccentricity (reproduced from Bjorke).

$$MX\Sigma_R = 0.0$$

$$R\Delta X\Sigma = |A|\ 2TX_R \qquad \text{(eq.s 2.23)}$$

$$E\Delta X\Sigma = 0.0$$

$$\mathrm{var}\Delta X\Sigma = A^2\ 0.5(TX_R{}^2\ \mathrm{var}_o z_R)$$

Because of the symmetry that exists in the bushing, $MX\Sigma_R$ and $E\Delta X\Sigma$ are both zero. The variable X_R ranges from zero to X_R, which also happens to be its tolerance. Because of symmetry the middle of the range is zero, X exists only in the stochastic part of the dimension as illustrated in equation 2.1. Since the model of individual dimensions states that the range of the individual dimensions equals the tolerance of the individual dimensions, it can be said that:

$$TX_R = X_R$$

Further, Figure 2.11, shows that the range of $R\Delta X\Sigma$ must be equal to twice X_R.

2.4.2.2. Gaps

The analysis of chain links that are gaps will be discussed in two categories line vector gaps and plane vector gaps.

Line Vector Gaps

A line vector gap can have either lumped or distributed magnitude. A line vector gap with lumped magnitude is a link with its two components touching at either of the extremes along an axis parallel to the sum dimension (Figure 2.12). The point of contact is dependent on the forces applied to the part. The important factor for a gap with lumped magnitude is that its components are, by definition, touching.

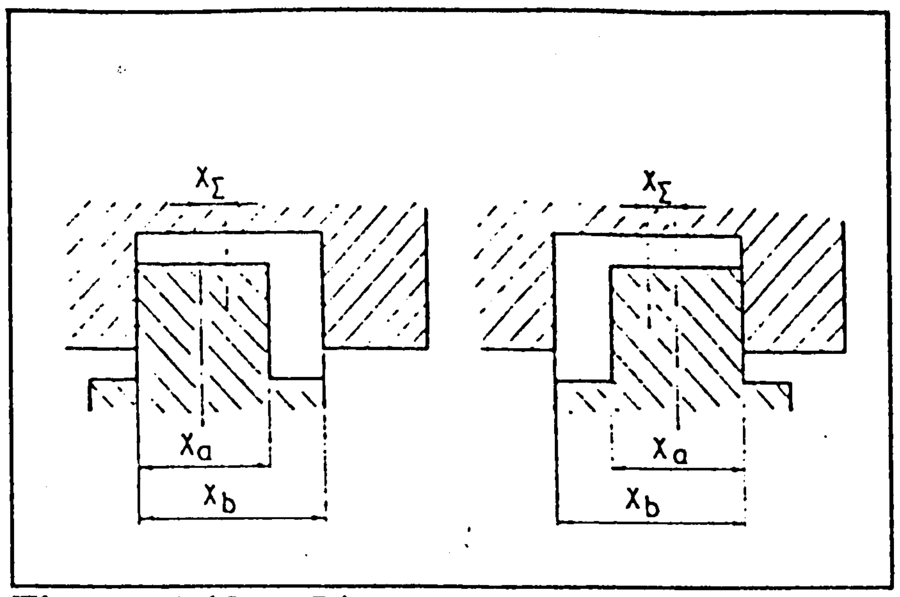

Figure 2.12 Line vector gap with lumped magnitude (reproduced from Bjorke).

Among the line vector gaps with lumped magnitude, three distinctions may be made. The gap may be a clearance, an interference, or a transition. In the first case of a clearance, the gap is always positive requiring that the minimum dimension of X_b always be greater than the maximum dimension of X_a. In the second case the gap is always negative (fit condition). The maximum dimension of X_a is always greater than X_b. The gap in the case of a transitional fit is intermittently positive and negative. Figures 2.13 -2.15 illustrate graphically the three different types of fits.

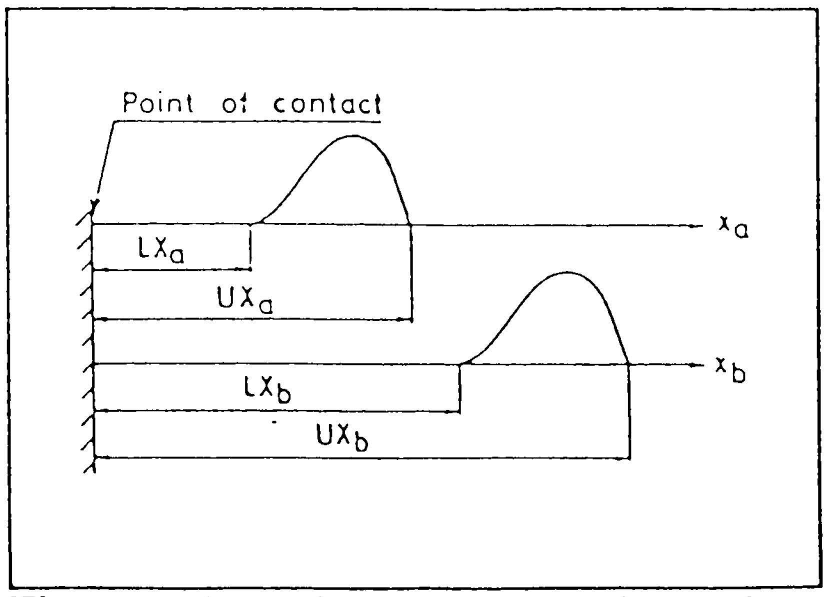

Figure 2.13 Clearance (reproduced from Bjorke).

clearance: lower limit of $X_b \geq$ upper limit of X_a

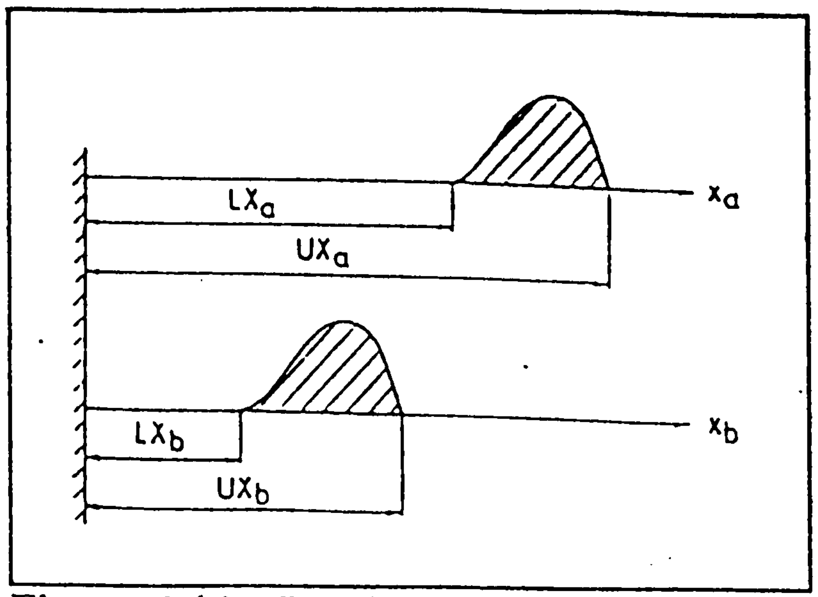

Figure 2.14 Interference (reproduced from Bjorke).

interference: upper limit of $X_b \leq$ lower limit of X_a

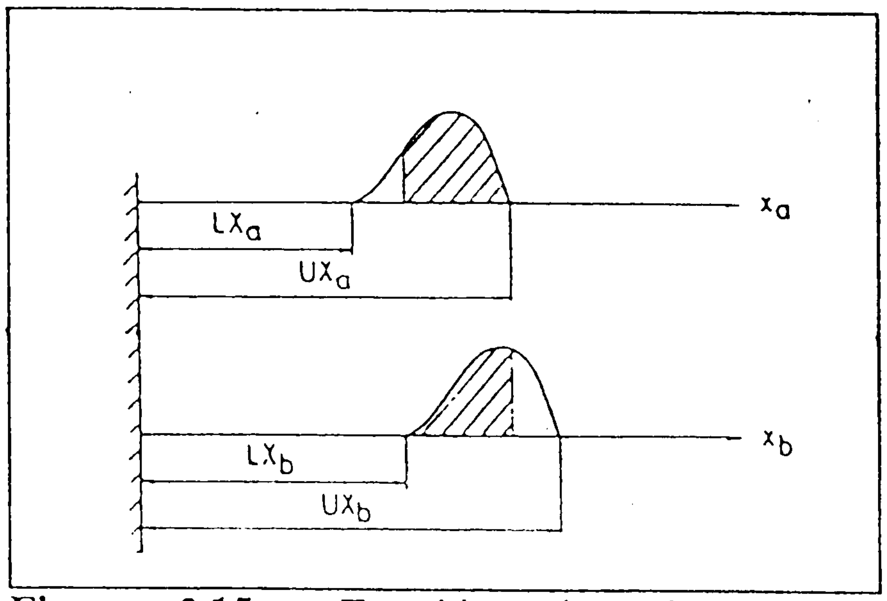

Figure 2.15 Transition (reproduced from Bjorke).

transition: lower limit of $X_b \leq$ upper limit of X_a

and

upper limit of $X_b \geq$ lower limit of X_a

The fundamental equation of a clearance is:

$$X\Sigma = A\ (X_b - X_a)\ /\ 2$$

It can be seen that this equation is comprised of the two line vector spans X_b and X_a. The equations for a line vector span with lumped magnitude that is classified as a clearance are thus derived from equations (2.20).

$$MX\Sigma_R = A/2\ (MX_b - MX_a)$$

$$R\Delta X\Sigma = |A|/2\ (TX_b + TX_a) \qquad \text{(eq.s 2.24)}$$

$$E\Delta X\Sigma = A/2\ (TX_b\ (Ez_b - 0.5) - TX_a\ (Ez_a - 0.5))$$

$$\text{var}\Delta X\Sigma = A^2/4\ (TX_b^2\ \text{varz}_b + TX_a^2\ \text{varz}_a)$$

An interference type gap, because of its fit condition, does not allow any relative movement between the two parts. Although the functionality of the fit may be affected by the variability of the parts, the sum dimension is not affected as long as the parts remain in a fit condition. Any links in a tolerance chain that are determined interferences can therefore be neglected in the calculation of the sum dimension's parameters.

The transition line vector gap is somewhat of a combination of the two gaps just described. The characteristic of this gap is distributed between an interference and a clearance. In order to describe the behavior of a transition, the variable X_R is introduced. If X_R is the size of the gap,

$$X_R = (X_a - X_b)\ /\ 2$$

then the fundamental equation of a transition gap can be written:

$$X\Sigma = A\ X_R \qquad \text{for} \qquad X_R \geq 0$$

$$X\Sigma = 0 \qquad \text{for} \qquad X_R < 0$$

In order to determine the influence of a transition link on the sum dimension, the distribution of X_R must be determined. In other words what part of the time the link is a clearance and when it is an interference must be determined. If the variable X_R has a probability density function $f_1(X_R)$, then the probability density function of $X\Sigma$ is:

$$f(X\Sigma) = f_1(X\Sigma) \text{ for } \quad X_R > 0$$
$$f(X\Sigma) = \infty \quad \text{for} \quad X_R = 0$$
$$f(X\Sigma) = 0 \quad \text{for} \quad X_R < 0$$

From the definitions of mean and variance, $EX\Sigma$ and $varX\Sigma$ can be calculated as:

$$EX\Sigma = X\Sigma\ f(X\Sigma)\ \Delta X\Sigma$$
$$varX = X\Sigma^2\ f(X\Sigma)\ \Delta X\Sigma - (EX\Sigma)^2$$

The general probability density function $f(X\Sigma)$ above has been integrated by Haugrud [10] for the normal and rectangular distributions. The results are shown in Figures 2.16 and 2.17 below. The abscissas of the diagrams are values of the normalized expectation of X_R, i.e. EX_R/σ_{XR}. The expectation of X_R, EX_R, is calculated from the equations 2.24 and equation 2.1.

$$EX_R = MX_R + E\Delta X_R$$
$$varX_R = var\Delta X_R$$

From (2.23):

$$E\Delta X_R = 1/2\ (TX_b\ (Ez_b - 0.5) - TX_a\ (Ez_a - 0.5))$$
$$var\Delta X_R = 1/4\ (TX_b^2\ varz_b + TX_a^2\ varz_a)$$
$$MX_R = 1/2\ (MX_b - MX_a)$$

The expectation and variance of the variable X_R can be written:

(eq.s 2.25)

$$EX_R = 1/2\ (MX_b - MX_a + TX_b\ (Ez_b - 0.5) - TX_a\ (Ez_a - 0.5))$$

$$varX_R = 1/4\ (TX_b^2\ varz_b + TX_a^2\ varz_a)$$

Note the scaling factor "A" is left out above. This omission results from the fact that the influence on the sum dimension is not being calculated here.

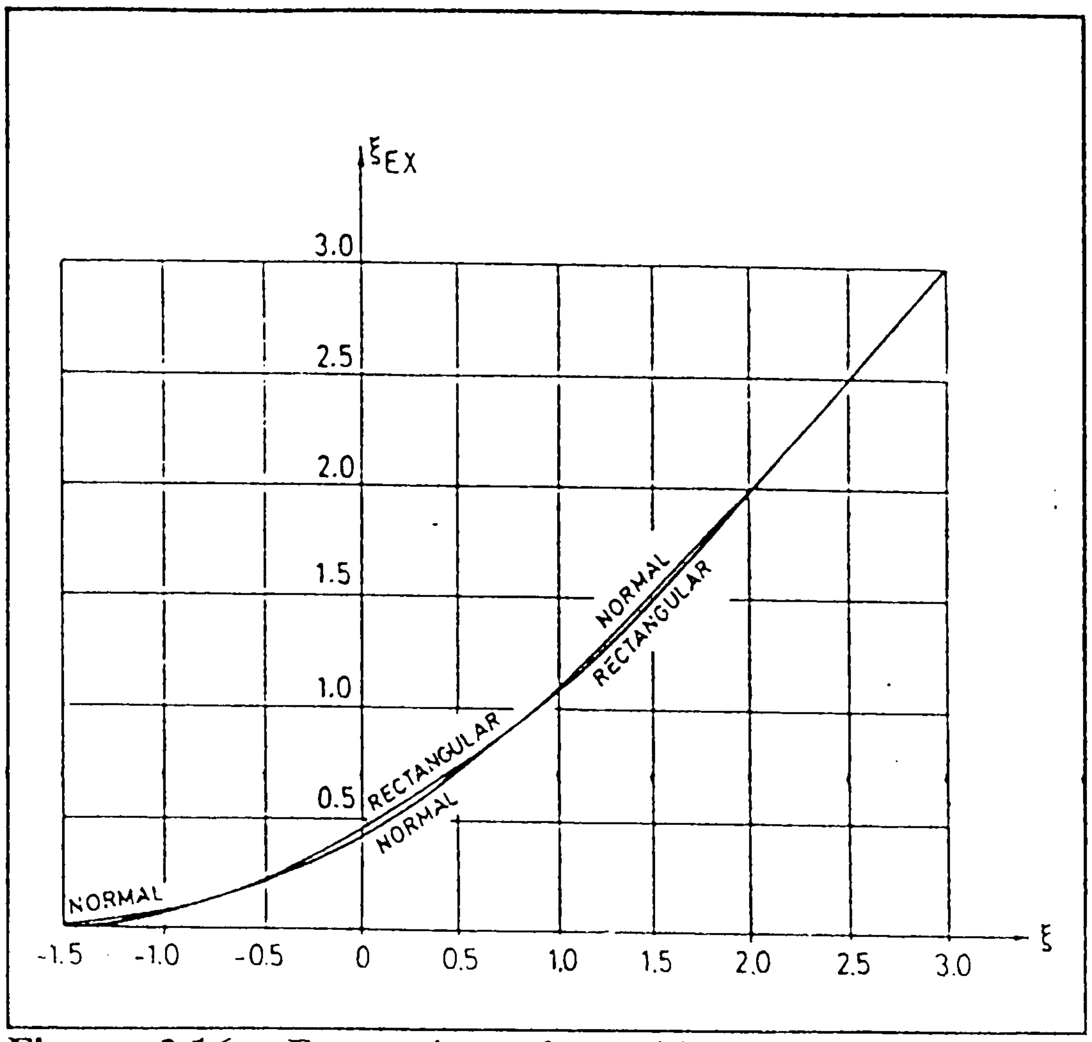

Figure 2.16 Expectation of transitions (reproduced from Bjorke).

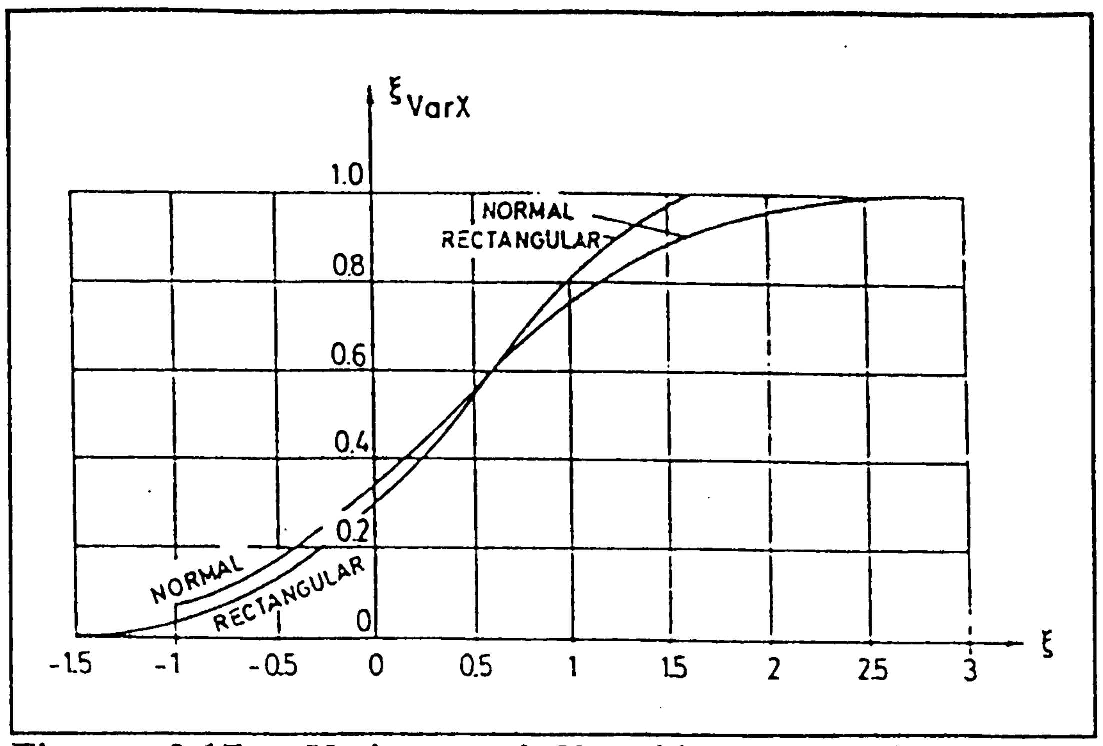

Figure 2.17 Variance of Transitions (reproduced from Bjorke).

As stated above, the abscissas, δ, are values of the normalized expectation of X_R.

$$\delta = EX_R/\sigma_{XR}$$

The ordinates on the two diagrams are:

$$\delta_{EX} = EX\Sigma/\sigma_{XR} \qquad \text{and} \qquad \delta_{varx} = varX\Sigma/varx_R$$

Therefore,

$$EX\Sigma = \delta_{EX}\ \sigma_{XR} \qquad \text{and} \qquad varX\Sigma = \delta_{varx}\ varx_R.$$

Finally the equations for the effect on the sum dimension of a line vector gap classified as a transition can be found using the Haugrud plots and the following equations:

$$MX\Sigma_R = A/4\ (UX_b - LX_a)$$
$$R\Delta X\Sigma = |A|/2\ (UX_b - LX_a) \qquad \text{(eq.s 2.26)}$$
$$E\Delta X\Sigma = A\ \delta_{EX}\ \sigma_{XR} - MX\Sigma_R$$
$$var\Delta X\Sigma = A^2\ \delta_{varx}\ varx_R$$

The final type of line vector gap is the line vector gap with distributed magnitude. In the lumped magnitude line vector gap, the mating parts were touching so the gap was known as a relation between the two parts. The dimension of the gap in this type of link however is completely unknown and can be located anywhere along a axis coincident or parallel to the sum dimension. An example of this possible geometry is shown below in Figure 2.18.

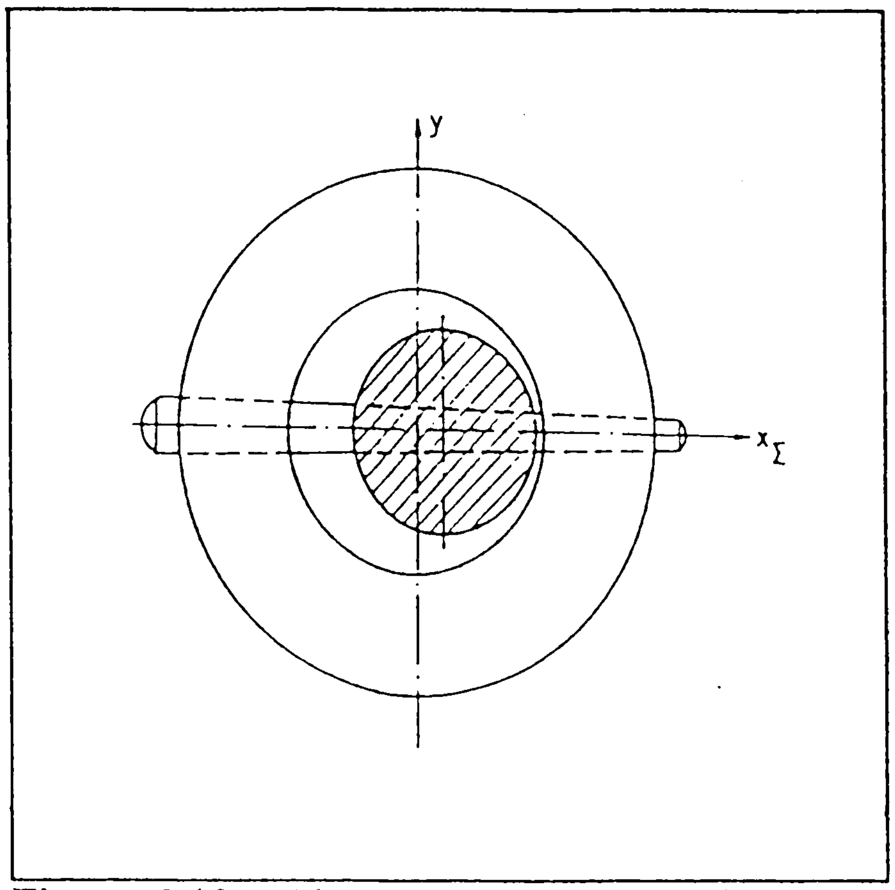

Figure 2.18 Line vector gap with distributed magnitude (reproduced from Bjorke).

For this link the variable X_R is be defined as:

$$X_R = (X_b - X_a) / 2 \qquad \text{(eq. 2.27)}$$

The location of X_a can be given by a probability distribution function and ranges from $[-X_R, X_R]$, as shown in Figure 2.19.

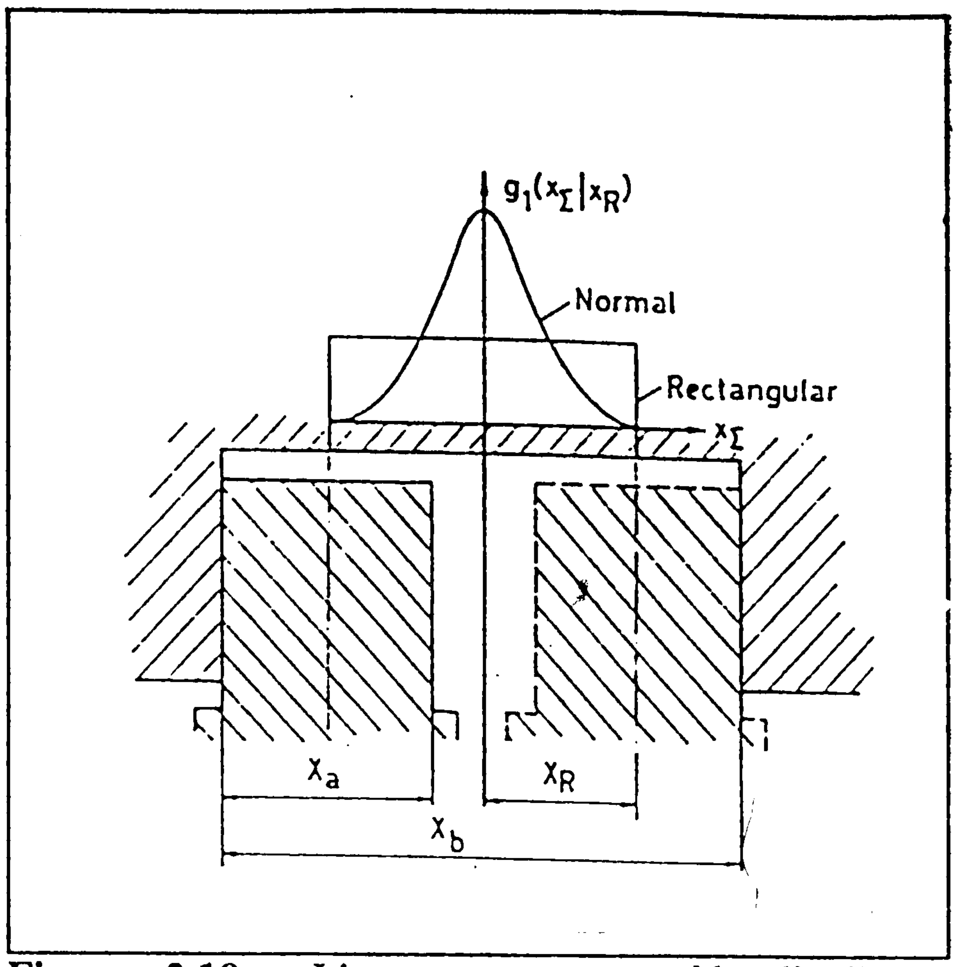

Figure 2.19 Line vector gap with distributed magnitude (reproduced from Bjorke).

In the figure above, the function of the link's effect on the sum dimension is shown as a normal and rectangular distribution. Regardless of its distribution, the function g_1 is dependent on X_R which is a variable itself. The function g_1 is therefore the conditional probability density of $X\Sigma$, given X_R.

$$g_1(X\Sigma|X_R) = \frac{f(X\Sigma,\ X_R)}{f_1(X_R)} \qquad \text{(eq. 2.28)}$$

The probability density function $f_1(X_R)$ is a function of the manufacturing of the two mating parts, while the function $g_1(X\Sigma|X_R)$ results from the assembly. For sum dimension analysis, it is desired to determine an unconditional density function in the sum direction $g(X\Sigma)$. The density $g(X\Sigma)$ is given by:

$$g(X\Sigma) = \int_{-\infty}^{\infty} f(X\Sigma,\ X_R)\, dX_R$$

and from equation (2.28)

$$g(X\Sigma)=\int_{-\infty}^{\infty} g_1(X\Sigma|X_R)\ f_1(X_R)\ dX_R$$

Both integrals on the right hand side are usually known, but the derivation of the function of the sum dimension requires only that the g_1 be known, and the distribution of $X\Sigma$ be left as general. The derivation of the effect of this link on the sum dimension is almost entirely calculations and is repeated verbatim from Bjorke[7] in Appendix III. The equations are derived for the conditional probability function's being normal. The steps for the rectangular distribution are similar to those for the normal and they are omitted from this discussion. The results however for the rectangularly distributed center location are included. The resulting steps in the derivations are very interesting and it is recommended that the reader review them.

Results for line vector gaps with distributed center location are listed below first for normally distributed center location and then for the rectangularly distributed center location.

$$MX\Sigma_R = 0$$

$$R\Delta X\Sigma = |A|\ 2TX_R \qquad \text{(eq. 2.29)}$$

$$E\Delta X\Sigma = 0$$

$$\mathrm{var}\Delta X\Sigma = A^2/9\ TX_R^{\ 2}\ \mathrm{var}_o z_R$$

$MX\Sigma_R$ and $E\Delta X\Sigma$ are both zero because of symmetry of the gap (as in the eccentricity). The range of the X_R is also similar to the range of eccentricities. The range must be two times X_R, and like the eccentricity, the model of the individual link is such that the range equals the tolerance. The range of X_R equals the tolerance of X_R and can be represented as:

$$TX_R = 1/2\ (UX_b - UX_a). \qquad \text{(eq. 2.30)}$$

The variance $var_o z_R$ is the variance of the unit distribution as measured relative to the origin instead of the mean. The details of this calculation are included in Appendix III. The value of the variance in relation to the origin, $var_o z_R$, is not needed to be known by the user of the computer module, the module calculates the value invisibly to the user.

The equations for the rectangular distribution are:

$$MX\Sigma_R = 0$$
$$R\Delta X\Sigma = |A|\ 2TX_R \qquad \text{(eq.s 2.31)}$$
$$E\Delta X\Sigma = 0$$
$$var\Delta X\Sigma = A^2/3\ TX_R^2\ var_o z_R$$

where TX_R is calculated the same as for the normally distributed center location.

Plane vector gaps.

The effect on the sum dimension of plane vector gaps with lumped magnitude is determined much the same way that line vector gaps were determined. The plane vector gap with lumped direction has a given but uncertain angle $X\alpha$ in reference to the sum direction. The fundamental equation for a lumped direction plane vector gap is identical to the fundamental equation for line vector gaps, except for the last term.

$$X\Sigma = A\ ((X_b - X_a)\ /\ 2\ )\ \cos(X\alpha) \qquad \text{(eq. 2.32)}$$

It is the last term that transforms the influence of the link to the direction of the sum dimension.

Equation 2.32 is not a linear equation. Both the magnitude and the angle are stochastic variables. It is generally the case that the variation in the direction is negligible as compared to the variation in the magnitude. Equation 2.32 is linearized in the same manner as is equation 2.21, by considering small variations from the mean of angle α. Equation 2.32 is therefore written:

$$X\Sigma = A\ ((X_b - X_a) / 2)\cos(\alpha) \qquad \text{(eq. 2.33)}$$

and the equations for plane vector gaps with lumped direction are similar to the line vector gaps with A replaced by A $\cos(\alpha)$. Equations for lumped direction plane vector clearances, transitions, and distributed center location, both rectangularly and normally distributed, are obtained by replacing A with A $\cos(\alpha)$.

The last category of links are plane vector gaps with distributed direction. A distributed direction plane vector gap can form any angle with the sum direction. These gaps can be either lumped magnitude, as shown if Figure (2.20), or distributed magnitude as shown is Figure (2.21). The lumped magnitude link requires that the parts always be touching; that characteristic establishes the relative position of the parts. In the distributed magnitude, the internal part can be located anywhere within the circle of radius X_R shown in Figure (2.20).

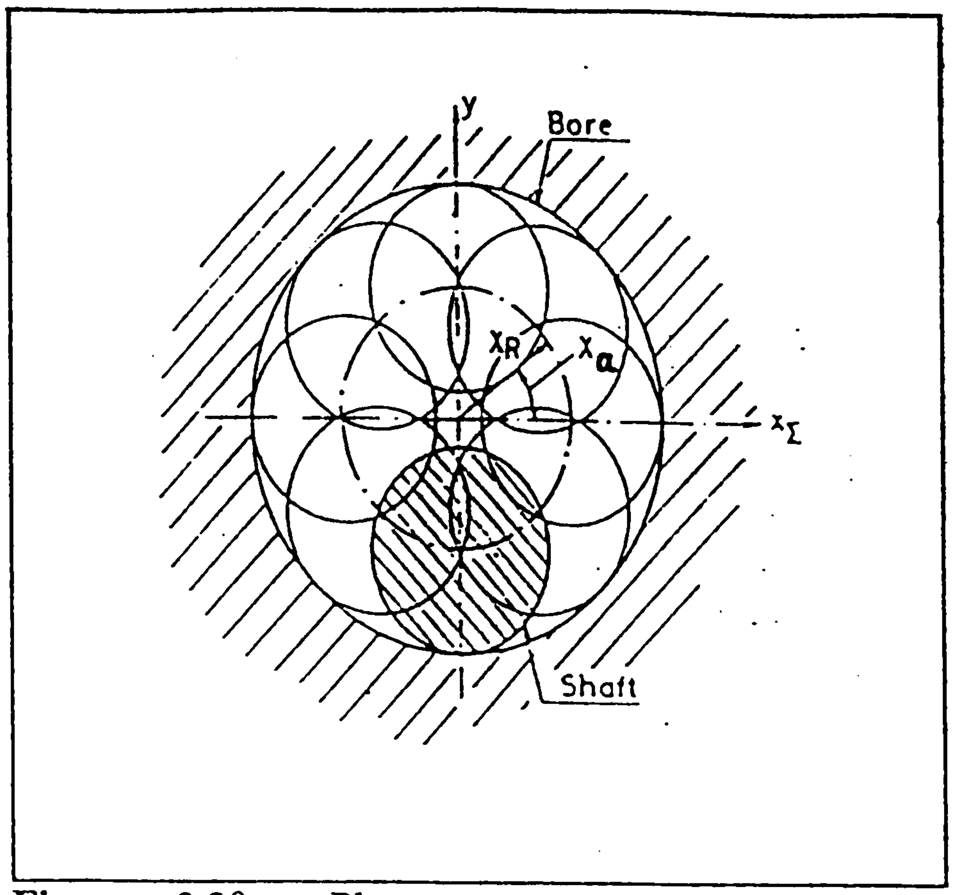

Figure 2.20 Plane vector gap with lumped magnitude and distributed direction (reproduced from Bjorke).

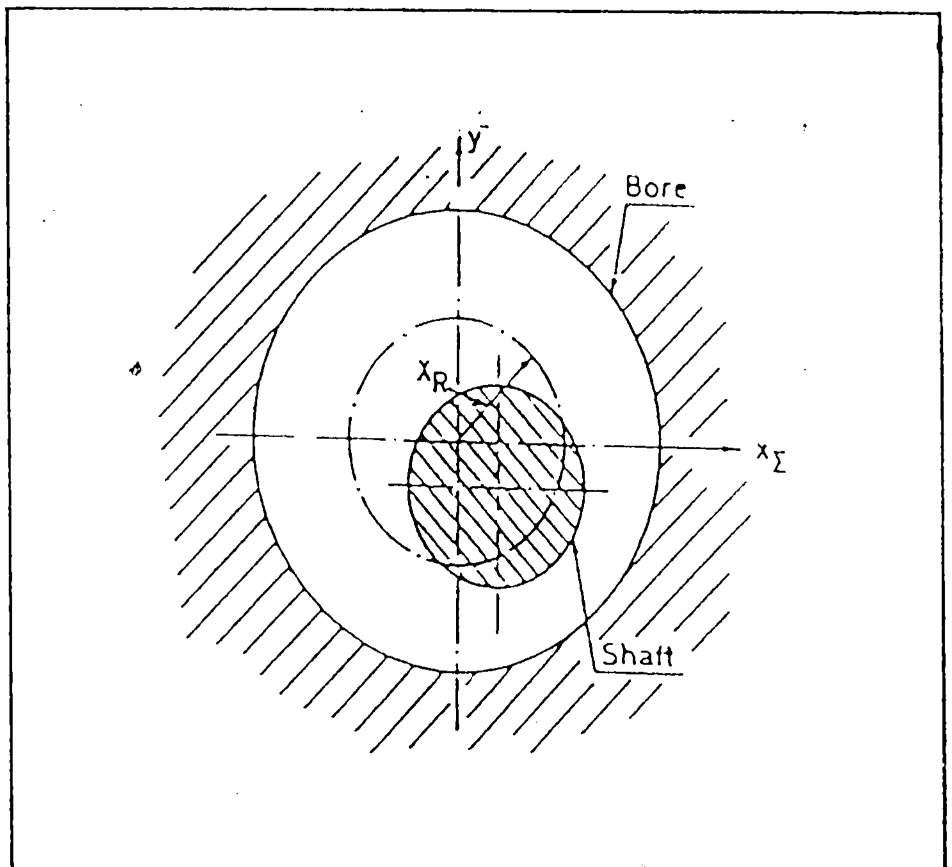

Figure 2.21 Plane vector gap with distributed magnitude and distributed direction (reproduced from Bjorke).

Lumped magnitude

The fundamental equation of a plane vector gap with lumped magnitude and distributed direction is the same as that for the plane vector gap with lumped magnitude and lumped direction, with the exception that the variate, X_α, is distributed.

$$X_R = (X_b - X_a) / 2$$

$$X\Sigma = A\ XR \cos(X_\alpha) \qquad \text{(eq. 2.34)}$$

The distribution of the angle X_α is a result of the assembly and not manufacture. As in eccentricity, it is assumed that the location takes no preference, and thus is characterized by a rectangular distribution.

$$f_2(X_\alpha) = 1/(2\pi)$$

The equations for the plane vector gap with distributed direction and lumped magnitude are similar to those of the eccentricity. Therefore, the equation 2.34 is algebraically similar as the equation for the eccentricity. Although the variable X_R has a different meaning, the solution of the equation is the same. The equations are again presented below, where TX_R is found from equation (2.30).

$$MX\Sigma_R = 0$$

$$R\Delta X\Sigma = |A|\ 2TX_R \qquad \text{(eq.s 2.35)}$$

$$E\Delta X\Sigma = 0$$

$$\mathrm{var}\Delta X\Sigma = A^2\ 1/2\ TX_R^2\ \mathrm{var}_o z_R$$

Distributed magnitude

The plane vector gap with distributed magnitude can also have its direction distributed. In this case the internal part can be located within the circle in Figure 2.19. The density of the sum dimension is again conditional, as was the line vector gap with lumped magnitude because X_R is a variate. In this link, however, the position of the internal part is unknown, therefore making the location a bivariate distribution of $X\Sigma$ and now also Y, given X_R. This distribution is given by $g_2(X\Sigma,Y|X_R)$. Similar to the previous link, the probability density function of X_R, $f_1(X_R)$, is a result of the machining of the parts, and the conditional distribution, $g_2(X\Sigma, Y|X_R)$, is dependent on the assembly. It is still the intention to determine the statistical parameters of the sum dimension as a function of the univariate distribution $g(X\Sigma)$. The conditional density can be calculated as:

$$g_1(X\Sigma|X_R) = \int_{-\infty}^{\infty} g_2(X\Sigma, Y|X_R)\,dY \qquad \text{(eq. 2.36)}$$

and the final univariate density function $g(X\Sigma)$,

$$g(X\Sigma) = \int_{-\infty}^{\infty} g_1(X\Sigma|X_R)\, f_1(X_R)\, dX_R \qquad \text{(eq. 2.37)}$$

as was derived in the previous link development. As before, once g_2 is known, g_1 can be calculated. The results for a distribution with g_2 normal and rectangular are listed. The development for the two is similar and is carried out only for the bivariate normally distributed center location.

The location of the internal part is said to be normally distributed in both the $X\Sigma$ and the Y direction. The definition of a bivariate normal distribution is:

$$g_2(X_\Sigma,Y) = \frac{\exp\left[-\frac{1}{2(1-\rho^2)}\left(\left(\frac{X_\Sigma - \zeta_{X_\Sigma}}{\sigma_{X_\Sigma}}\right)^2 - 2\rho\left(\frac{X_\Sigma - \zeta_{X_\Sigma}}{\sigma_{X_\Sigma}}\right)\left(\frac{Y - \zeta_Y}{\sigma_Y}\right) + \left(\frac{Y - \zeta_Y}{\sigma_Y}\right)^2\right)\right]}{2\pi\sigma_{X_\Sigma}\sigma_Y\sqrt{1-\rho^2}} \qquad \text{(eq. 2.38)}$$

Since the variables are uncorrelated, p=0, the function becomes:

$$g_2(X_\Sigma,Y) = \frac{1}{2\pi\sigma_{X_\Sigma}\sigma_Y}\exp\left[-\frac{1}{2}\left(\left(\frac{X_\Sigma - \zeta_{X_\Sigma}}{\sigma_{X_\Sigma}}\right)^2 + \left(\frac{Y - \zeta_Y}{\sigma_Y}\right)^2\right)\right] \qquad \text{(eq. 2.39)}$$

It is also valid to assume to say that the expectation and the variance of the two dimensions are equal. To do so reduces the above equation to:

$$g_2(X_\Sigma,Y) = \frac{1}{2\pi\sigma^2}\exp\left[-\frac{1}{2}\left(\left(\frac{X_\Sigma - \zeta}{\sigma}\right)^2 + \left(\frac{Y - \zeta}{\sigma}\right)^2\right)\right] \quad \text{(eq. 2.40)}$$

If it is assumed that the above equation is the distribution of the center location, the expectation is set to zero. It is further assumed that the range X_R complies with the 3σ limits.

Substituting $\mu = 0$, and $\sigma = X_R/3$ into the above equation yields the bivariate conditional density for $X\Sigma$ and Y given X_R:

$$g_2(X_\Sigma,Y|X_R) = \frac{9}{2\pi X_R^2}\exp\left[-\frac{1}{2}\left(\left(\frac{3X_\Sigma}{X_R}\right)^2 + \left(\frac{3Y}{X_R}\right)^2\right)\right] \quad \text{(eq. 2.41)}$$

By integrating out Y, as above in equation (2.36), the conditional univariate probability density function of $X\Sigma$ given X_R is:

$$g_2(X_\Sigma,Y|X_R) = \int_{-\infty}^{\infty} \frac{9}{2\pi X_R^2}\exp\left[-\frac{1}{2}\left(\left(\frac{3X_\Sigma}{X_R}\right)^2 + \left(\frac{3Y}{X_R}\right)^2\right)\right] dY$$

$$= \frac{3}{\sqrt{2\pi}\, X_R}\exp\left[-\frac{9}{2}\left(\frac{X_\Sigma}{X_R}\right)^2\right] \quad \text{(eq. 2.42)}$$

It can be seen that equation (2.42) is identical to the equation for normally distributed center location discussed in Appendix III; the development of the sum dimension equations is therefore identical. The effect of a plane vector gap with a bivariate normal distribution on a sum dimension is from equations (2.29):

$MX\Sigma_R = 0$

$R\Delta X\Sigma = |A|\ 2TX_R$ (eq.s 2.43)

$E\Delta X\Sigma = 0$

$\mathrm{var}\Delta X\Sigma = A^2/9\ TX_R{}^2\ \mathrm{var}_o z_R$

The equations for the effect on the sum dimension from a link that is a plane vector gap with a bivariate rectangular are given as the following:

$MX\Sigma_R = 0$

$R\Delta X\Sigma = |A|\ 2TX_R$ (eq.s 2.44)

$E\Delta X\Sigma = 0$

$\mathrm{var}\Delta X\Sigma = A^2/4\ TX_R{}^2\ \mathrm{var}_o z_R$

Significantly the only difference between the equations for the bivariate normal distribution and the bivariate rectangular distribution is the factor in front of the variance equation. Bjorke states that it can therefore be determined that any distribution between rectangular and normal will have a constant between 1/9 and 1/4. This is important because the equations can be determined for any distribution between the normal and rectangular without going through the calculations above.

2.4.3. Summation of chain links

The summation of the chain links is performed to determine the statistical parameters of the sum dimension based on the parameters of the individual links. The manner in which the individual chain links are summed depends on the sum dimension being analyzed. There are three basic types of sum dimensions that Bjorke differentiates, two of which will be considered in this thesis. The first case occurs when the sum dimension is on a stationary part or between two parts that don't experience any relative motion between them. The second is identified when the sum dimension exists between a rotating part and a stationary part. And the third case which is not developed here is reserved for a sum dimension between two rotating parts, such as a gear set.

2.4.3.1. Stationary parts.

The parameters for a sum dimension existing between two stationary parts are determined from the equations developed in section 2.3.1, equations 2.14 - 2.17; the results are simply summed up. The unit parameters for these equations can be found in the table of unit parameters listed in Appendix I.

2.4.3.2. Stator - Rotor case.

When the sum dimension exists between two rotating parts such as shaft and a bore, it is no longer sufficient to sum up the parameters in the X direction only. What is of concern is actually is the maximum deviation of the stator and the rotor in the radial direction as shown in Figure 2.22.

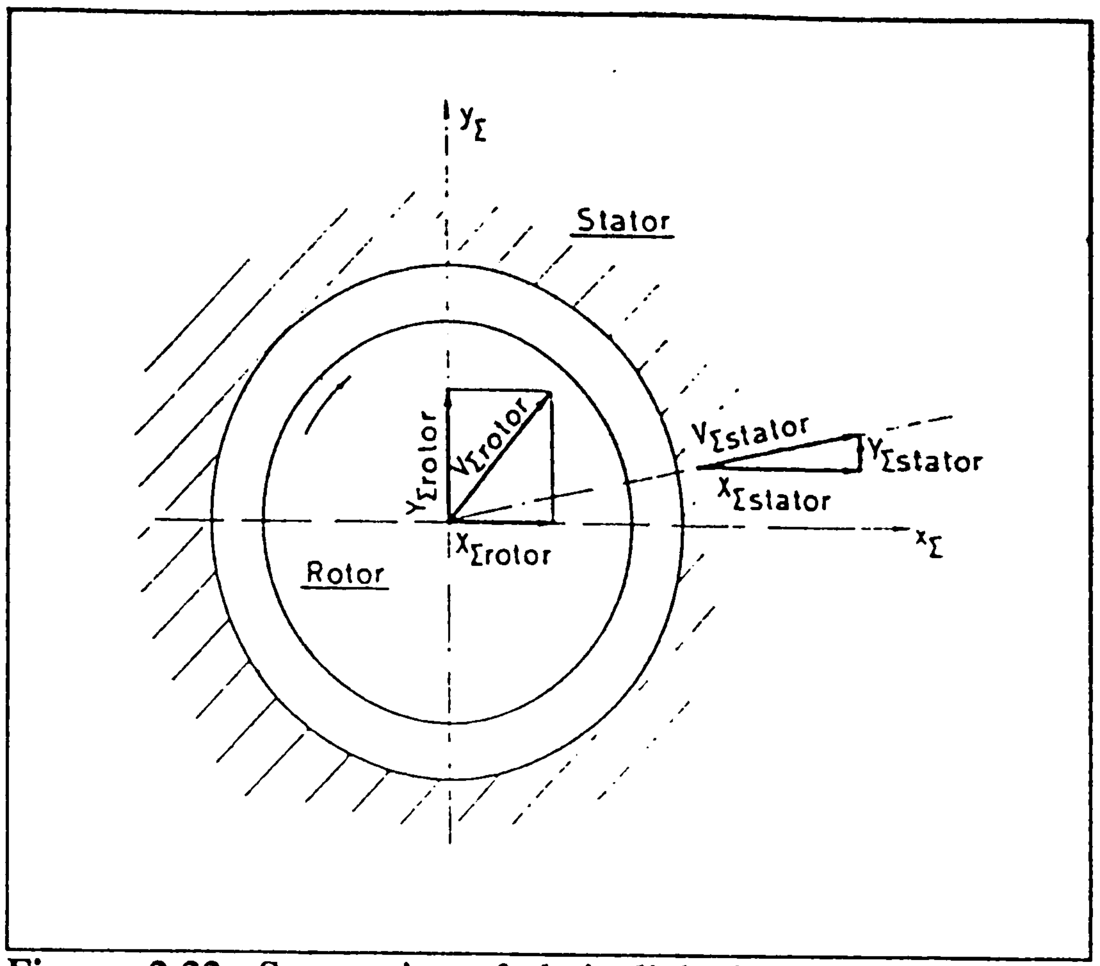

Figure 2.22 Summation of chain links between a stationary and a rotating part (reproduced from Bjorke).

The technique requires summing the parameters on the stator and rotor separately, as was done in the stationary case, and then determining the interaction between the parameters and tolerances in the radial direction. Bjorke separates the link types into two categories to make the summation more convenient. These categories are sum slants and sum eccentricities. Sum slants are link types that have non-zero expectations.

The development that Bjorke presents assumes that only line vector gaps make up the chain. The scheme will be developed for all link types in Chapter 3 of the paper.

The equations for the stochastic part of a line vector span in the radial direction are listed below.

$$\Delta X_\Sigma = \Sigma\ A_i \cos\alpha_i\ \Delta X_i \qquad \text{(eq.s 2.45)}$$

$$\Delta Y_\Sigma = \Sigma\ A_i \sin\alpha_i\ \Delta X_i$$

The expectation and variance of the components are therefore:

$$E\Delta X\Sigma = \Sigma\ A_i \cos(\alpha_i)\ TX_{Li}\ (Ez_{Li} - 0.5)$$
$$E\Delta Y\Sigma = \Sigma\ A_i \sin(\alpha_i)\ TX_{Li}\ (Ez_{Li} - 0.5) \qquad \text{(eq.s 2.46)}$$
$$\mathrm{var}\Delta X\Sigma = \Sigma\ A_i^2 \cos^2(\alpha_i)\ TX_{Li}^2\ \mathrm{varz}_{Li}$$
$$\mathrm{var}\Delta Y\Sigma = \Sigma\ A_i^2 \sin^2(\alpha_i)\ TX_{Li}^2\ \mathrm{varz}_{Li}$$

It can be seen that the components are correlated. The covariance is shown to be:

$$\mathrm{cov}\Delta X\Sigma\Delta Y\Sigma = \Sigma\ A_i^2 \sin(\alpha_i) \cos(\alpha_i)\ TX_{Li}^2\ \mathrm{varz}_{Li} \qquad \text{(eq. 2.47)}$$

Using equations (2.46) and (2.47)

$$\mathrm{var}X\theta = \mathrm{var}\Delta X\Sigma \cos^2(\theta) + \mathrm{var}\Delta Y\Sigma \sin^2(\theta) + 2\mathrm{cov}\Delta X\Sigma\Delta Y\Sigma \sin(\theta) \cos(\theta)$$

By way of analogy to the principle stress formulas, the principle axis of variance can be stated as:

$$\tan 2\theta_p = 2\ \mathrm{cov}\Delta X\Sigma\Delta Y\Sigma\ /\ (\mathrm{var}\Delta X\Sigma - \mathrm{var}\Delta Y\Sigma) \qquad \text{(eq. 2.48)}$$

and the principle variances are a combination of equation (2.47) and (2.48).

(eq.s 2.49)

$$\mathrm{var}\Delta a\Sigma = \mathrm{var}\Delta X\Sigma \cos^2(\theta_p) + \mathrm{var}\Delta Y\Sigma \sin^2(\theta_p) + \mathrm{cov}\Delta X\Sigma\Delta Y\Sigma \sin(2\theta_p)$$

$$\mathrm{var}\Delta b\Sigma = \mathrm{var}\Delta X\Sigma \sin^2(\theta_p) + \mathrm{var}\Delta Y\Sigma \cos^2(\theta_p) - \mathrm{cov}\Delta X\Sigma\Delta Y\Sigma \sin(2\theta_p)$$

A diagram of the calculations being performed is taken from Bjorke and shown below in Figure 2.23.

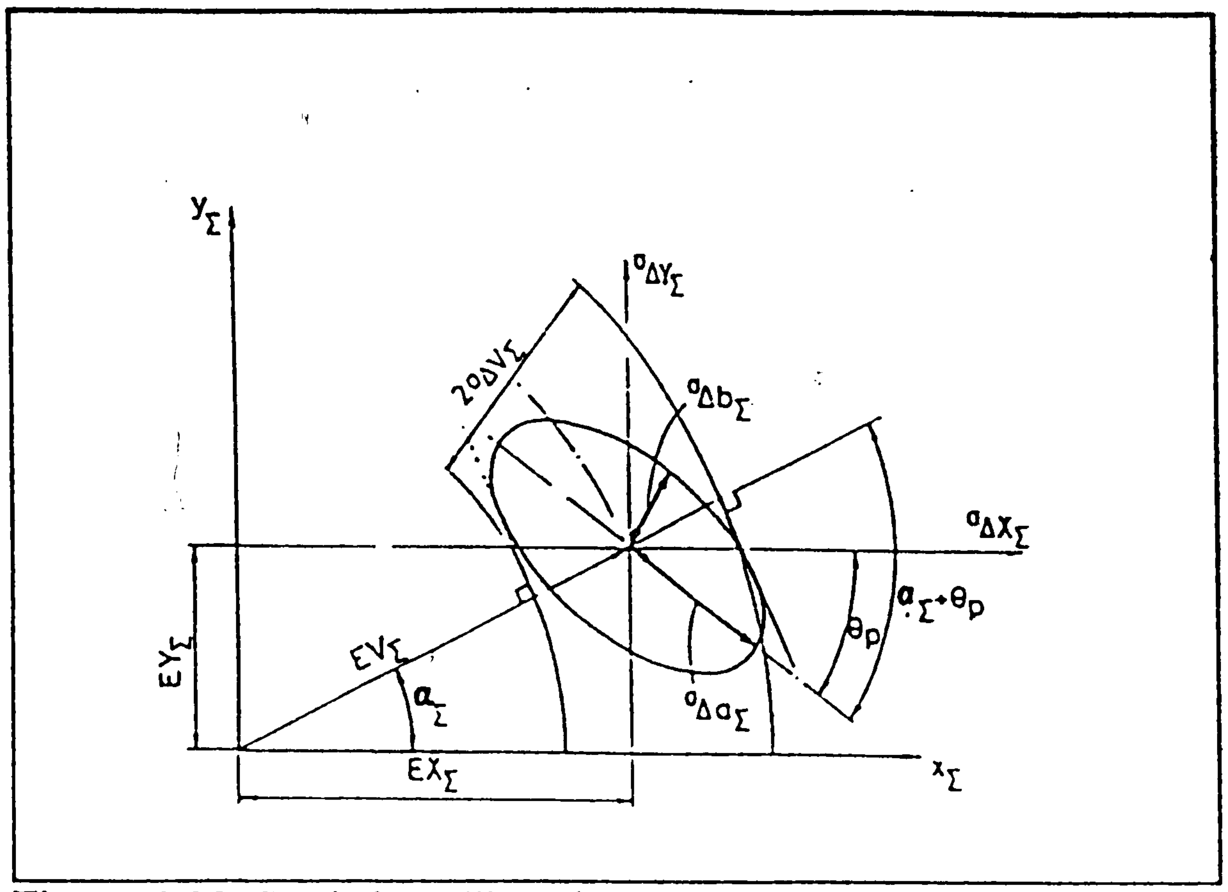

Figure 2.23 Deviation ellipse (reproduced from Bjorke).

The goal of these calculations is to determine the uncertainty in the radial direction. Two concentric circles are shown in the figure that are representative of the physical geometry. The point at which the circles and the ellipse touch is estimated as the point where the ellipse is touching a normal line to the radius vector. The variance is then calculated to be

(eq. 2.50)

$$\mathrm{var}\Delta V_\Sigma = \mathrm{var}\Delta a_\Sigma \ \cos^2(\alpha_\Sigma + \theta_p) + \mathrm{var}\Delta b_\Sigma \ \sin^2(\alpha_\Sigma + \theta_p)$$

α_Σ: angle between the radius vector and the X axis

θ_p: angle between the principle axis and the X axis.

From Figure 2.23 it can be seen that the expectation in the radial direction is given by:

$$EV_\Sigma = (EX_\Sigma^2 + EY_\Sigma^2)^{1/2} \qquad \text{(eq. 2.51)}$$

The parameters in the radial direction may be determined for both the stator and the rotor and then algebraically summed.

Like the sum slants, the parameters of the sum eccentricities must be calculated in the radial direction.

The parameters of the sum eccentricity can be calculated by procedures similar to the eccentricity type link. Because of symmetry, the parameters in the X and Y direction are equal. The development shows that the length of an eccentricity approaches the Rayleigh distribution. Assuming the sum eccentricity to be Rayleigh distributed, the expectation and the variance of the variates are:

$$E\Delta V = (\pi/2 \ \mathrm{var}\Delta X)^{1/2} \qquad \text{(eq.s 2.52)}$$
$$\mathrm{var}\Delta V = (4-\pi)/2 \ \mathrm{var}\Delta X$$

The expectation and variance of the sum eccentricity are therefore:

$$E\Delta V_{\Sigma} = (\pi/2 \ \Sigma \mathrm{var}\Delta X_i)^{1/2} \qquad \text{(eq.s 2.53)}$$
$$\mathrm{var}\Delta V_{\Sigma} = (4-\pi)/2 \ \Sigma \mathrm{var}\Delta X_i$$

These parameters are both in the radial direction so the parameters may be added algebraically between the stator and rotor.

2.5. Tolerance Control

A tolerance analysis or tolerance control is performed when the parts exist and the tolerances of the components are known. A tolerance control, determines the tolerance of the sum dimension at a desired confidence level based on the component parts. A tolerance control may be performed when the tolerances on parts have been approximated and the influence on the assembly needs to be determined. The distribution of the sum dimension is determined from the component parts as they were in previous sections.

This section will list the steps involved in performing a tolerance control, with some modification of Bjorke's approach. Some of the steps are eliminated by the interface with the geometric model, and some are treated differently in the approach taken in this thesis.

1. Determine the dimensions that influence the sum dimension and identify the variables.

   $X_1, X_2, X_3, \ldots, X_n$

2. Determine the fundamental equation of the sum dimension.

   $X\Sigma = f(X_1, X_2, \ldots, X_n)$

3. Compute the scaling factors.

   A_i

4. Classify the chain links

5. Determine the tolerance of the individual dimensions.

   TX_i

6. Determine the unit expectation and unit variance for the individual dimensions.

Ez_i, $varz_i$

7. Compute the middle point in the tolerance zone, the range, and the expectation and variance for the individual dimensions.

MX_i, $R\Delta X_i$, $E\Delta X_i$, $var\Delta X_i$

8. Compute the middle point in the tolerance zone, range, expectation and variance for the sum dimension by summing up the contribution for the individual dimension to the chain.

The approach taken in this paper diverges from Bjorke's approach at this point and the steps listed are different from those outlined by Bjorke. Bjorke makes use of normalized tables for the tolerance analysis, while the approach employed here is to directly integrate the beta distribution of the sum dimension in order to determine the confidence level of the assembly.

9. Determine the desired tolerance zone and mid point of the tolerance zone of the sum dimension.

10. Compute the confidence level of the assembly

The above steps 1 - 5 are performed in the geometric model segment of the program and the data, needed by the tolerance analysis model are transferred via a data file. The interface between the two segments of the package will be discussed in detail later in the thesis.

It is possible to change the desired tolerance on the sum dimension and move the tolerance zone within the range of the sum dimension to observe the changes in the confidence level calculated by the assembly. For instance if the designer realizes that the tolerance on the sum dimension could be larger, he can use the new value to determine what the increased confidence would be.

2.6. Tolerance distribution

The purpose of a tolerance distribution is to determine the tolerances on the components of an assembly based on the assembly's functional requirement. Parts within the assembly have either predetermined or determinable tolerances. Parts acquired from vendors, such as screws, bolts, o-rings, etc., are usually standard and have predetermined tolerances. On the other hand, parts that are being designed specifically for an application or ordered specially have determinable tolerances. The goal for tolerance distribution is opposite of the goal in tolerance control but the mathematical basis remains the same.

The total number of links in a chain is the sum of the links with determinable tolerances and the links with predetermined tolerances.

$n = n_s + n_t$

n_s: number of links with predetermined tolerances

n_t: number of links with determinable tolerances

The variance of the sum dimension is separated into the contribution from parts with determinable tolerances and parts with predetermined tolerances:

$$\mathrm{var}\Delta X_\Sigma = \sum_{i=1}^{n} A_i^2\, TX_i^2\, \mathrm{varz}_i$$

$$\mathrm{var}\Delta X_\Sigma = \sum_{i=1}^{n_s} A_i^2\, TX_i^2\, \mathrm{varz}_i + \sum_{i=1}^{n_t} A_i^2\, TX_i^2\, \mathrm{varz}_i \quad \text{(eq. 2.54)}$$

The procedure for carrying out a tolerance distribution begins the same way as a tolerance control. Steps 1 - 6 in a tolerance distribution are identical to those of a tolerance control. In a tolerance control it was stated that Bjorke used a table of normalized parameters for the beta distribution. In a tolerance distribution, the approach taken in this thesis uses the same table. Steps 7 -13 in a tolerance distribution are:

7. Determine the desired confidence level.

8. Estimate the normalized range

$$RW\Sigma \approx \left(\frac{n}{varz_{mean}} \right)^{1/2} \qquad \text{(eq. 2.55)}$$

n : total number of links in the chain

$varz_{mean}$: mean unit variance $1/n \sum_{i=1}^{n} varz_i$

The skewness of the distribution will usually be selected as $FW\Sigma = 0$

9. Determine the normalized tolerance from the tables in Appendix II.

10. Determine the links that have predetermined tolerances

11. Substitute the above determined values into eq.(2.54)

$$\sum_{i=1}^{n_t} A_i^2\, TX_i^2\, varz_i = var\Delta X\Sigma - \sum_{i=1}^{n_s} A_i^2\, TX_i^2\, varz_i$$

eliminate $var\Delta X\Sigma$ by using the normalization formula:

$$TX\Sigma = TW\Sigma\, (var\Delta X\Sigma)^{1/2}$$

$$\sum_{i=1}^{n_t} A_i^2\, TX_i^2\, varz_i = (TX\Sigma/TW\Sigma)^2 - \sum_{i=1}^{n_s} A_i^2\, TX_i^2\, varz_i = RHS \qquad \text{(eq. 2.56)}$$

where RHS is the right hand side of the equation.

12. Compute the n_t unknown tolerances TX_i on the left hand side in (2.56) in such a way that (2.56) is satisfied and the manufacturing cost is minimized. The method chosen to do this will be discussed in the next section.

13. Determine the expectation of the n_t individual dimensions in such a way that the following equation is satisfied.

$$EX_\Sigma = \sum_{i=1}^{n} A_i(MX_i + TX_i(Ez_i - 0.5)) \qquad \text{(eq. 2.57)}$$

2.6.1. Complexity Factor Method

Bjorke describes several methods such as the linear programming method, Peter's method, and about half a dozen other approximate methods for assigning tolerances. The method used in this thesis chosen has been coined the "complexity factor method" by Bjorke. This method, been chosen for its generic applicability, assigns tolerances based on a complexity factor assigned to the part. The complexity factors can be assigned based on the criteria that best suits the application or manufacturing facility. The factors could represent cost of manufacturing, difficulty of manufacturing, time to manufacture, etc. Any criteria, which are almost always somehow related to cost, that is important to assigning tolerances to the assembly, can be represented by the complexity factor scheme. The procedure outlined by Bjorke is presented below. The factors assigned in this method should be normalized by the smallest value in the set of complexity factors; i.e., all of the complexity factors are divided by the smallest one, making the smallest factor equal to one. In the examples it will be shown how varying the complexity factors adjusts the tolerances assigned to a part.

1. Estimate the complexity factor (Ko_i) of the dimensions in the chain (and normalize them).

Ko_i

2. Compute the weight of each dimension.

$$Ve_i = \frac{Ko_i}{\sum_{j=1}^{n} Ko_j} \qquad \text{(eq. 2.58)}$$

3. compute the tolerances from the formula:

$$TX_i = 1/|A_i| \, (Ve_i \, RHS \, / \, varz_i) \qquad \text{(eq. 2.59)}$$

The validity of the equation (2.59) can be shown by squaring and summing both sides of (2.59).

$$A_i^2 \, TX_i^2 \, varz_i = Ve_i \, RHS = \frac{Ko_i \, RHS}{\sum_{j=1}^{n} Ko_j}$$

$$\sum_{i=1}^{n} A_i^2 \, TX_i^2 \, varz_i = \frac{\sum_{i=1}^{n} Ko_i \, RHS}{\sum_{i=1}^{n} Ko_j} = RHS$$

The result is identical to (2.56).

2.7. Summary of theory

Much of the theory presented is difficult to digest in the absence of concrete examples, which will be presented in Chapter 4. The theory presented relies heavily on Bjorke's calculations. In this thesis some of the theory has been modified and some has been expanded. The examples given in Chapter 4 as well as the discussion in Chapter 3 should serve to illustrate the theory presented thus far and any modifications and expansions.

3 Implementation of Theory

3.1 Program Structure

3.1.1 Modular design

The theory laid out in Chapter 2 has been implemented as an interactive, menu driven computer program. This program is a tolerance analysis module that receives input from other modules and data files, e.g., the geometric modeling module, as well as from a user in the form of keyboard strokes. Control of the program is provided by a menu processor.

The menu processor is the skeleton of the module that gives it its structure and sustains a manageable environment for programming. The menu processor is a flexible tool that can be adapted easily to accommodate changes in the structure of the program. How this is performed will be discussed in this chapter. Use of the menu processor allows for a very simple and concise main calling program and provides the organization for modular programming.

Modular programming refers to a style of programming that places related ideas and tasks in manageable sub-programs. In this program, for instance, the main calling program is only 190 lines as compared to the sum of all of the routines, which is over 3000 lines. A modular approach allows for easier testing of the code, and future additions can be implemented with minimal effort as branches to the main program or an existing sub-program. Negatively, the modular technique does create more lines of code than a non-modular approach. Its advantages, however, in debugging and maintainability far outweigh the disadvantage of a somewhat longer code. Errors can more easily be traced when the code exists in modular form.

All of the routines written for the program are located in three object module libraries. The VAX/VMS operating system allows the user to create libraries of compiled routines to which other programs that use the routines can be linked. This capability allows for a highly organized directory for programming as well as installation of the system. Three object module libraries have been created to house all of the routines in the program except the main calling programming and several data files. These modules are named menu.olb, support.olb, and l_class.olb. They have been named in such a way that someone acquainting himself or herself with the operation of the program would have little difficulty finding a particular routine. Library menu.olb houses all of the routines necessary to drive the menu processor. L_class.olb contains all of the routines that calculate the individual link parameters. And similarly, the support.olb library contains all of the routines that provide the necessary support for the operation of the program. A list of the routines entered in these libraries is provided in Appendix IV.

All of the routines used in the module will be discussed in this chapter. Any reference to a subroutine in this chapter will be printed in italic print.

3.1.2 Menu processor

As stated above, the menu processor is the fundamental skeleton of the program. The menu processor contributes more than only the framework of the program. More importantly, it provides the interface between the user and the analysis. Through the menu system, the user is guided through the program to the functions that he desires. It is the menu processor that controls the direction the program takes as directed by input from the user. A diagram of the menu structure is illustrated in Figure 3.1.

When the program is run, execution begins by invoking the menu processor routine from the main program *TASM*. The third line in the main program is an include statement that tells the computer to include the file "MENU_INIT" at compile time. This file was created to keep from cluttering the main program. The file initializes the menu processor permitting the menu structure to be used in the

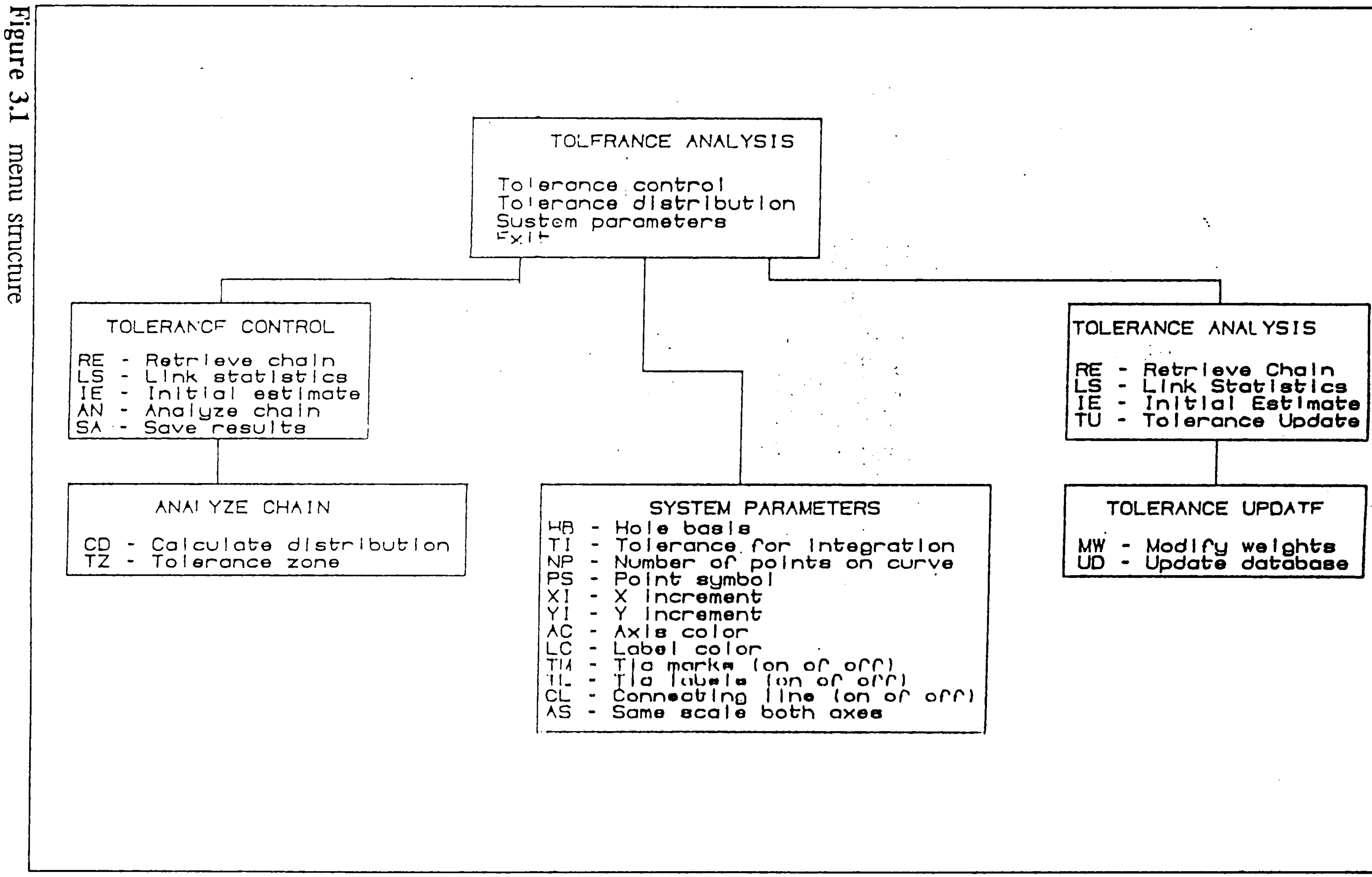

Figure 3.1 menu structure

program. It contains common blocks and data type declarations used in the menu processor routines.

It was stated earlier that the menu processor is flexible and can be modified if it was desired to alter the menu structure. This customization is performed in the "MENU_INIT" file that reads the menu data file to be used with the program. There exists one executable subroutine in the "MENU_INIT" file that is named "*read_menus*". This file reads in the data file created for the desired menu structure. When the structure of the menu is to be altered, i.e. selections added to a particular menu or other menus added to the menu tree, the changes are made to the menu data file, and the *read_menus* routine reads in the new menu structure.

The *read_menus* routine reads through the data file until it comes across a sentinel in the form of a "-1". When it comes across the first "-1", it begins to read a new menu. The first menu it reads is assigned a menu number one. When the routine encounters another "-1", it determines that it has reached the end of the contents for the current menu.

The information for a menu exists in the menu data file as follows in Figure 3.2:

```
-1
"n" entries in menu
menu title
entry 1
entry 2
.
.
.
entry "n"
-1
```

Figure 3.2 menu in menu.dat data file

The data file for the menu structure used in this program is listed in Figure 3.3. The last line in the data file is the text "end of file". When the *read_menus* routine encounters this statement, it stops reading menus and returns control to the main program, and the menu structure has been read in.

```
-1
4
TOLERANCE ANALYSIS
TC - TOLERANCE CONTROL
TD - TOLERANCE DISTRIBUTION
S- SYSTEM PARAMETERS
EX - EXIT
-1
-1
4
TOLERANCE CONTROL
RE - RETRIEVE CHAIN
LS - LINK STATISTICS
AN - ANALYZE CHAIN
SA - SAVE ANALYSIS RESULTS
-1
-1
4
TOLERANCE DISTRIBUTION
RE - RETRIEVE CHAIN
LS - LINK STATISTICS
IE - INITIAL STATISTICS
TU - TOLERANCE UPDATE
-1
-1
2
ANALYZE CHAIN
CD - CALCULATE DISTRIBUTION
TZ - TOLERANCE ZONE
-1
-1
2
TOLERANCE UPDATE
MW - MODIFY WEIGHTS
UD - UPDATE DATABASE
-1
END OF BUFFER
```

Figure 3.3 program menu data file

After the menus are read in, the main calling program *TASM* displays the main menu of the program. It does so by first calling a routine named *clear* that clears the screen and then calls the routine *get_option*. *Get_option* is a two parameter subroutine *get_option*(menu,iopt) that receives a menu number,"menu", calls display menu to display that menu number, and waits for the user to make a selection from the displayed menu. *Get_option* reads the selection made by the user, converts it to upper case, and compares it to the selections available on the active menu as well as several "global" menu selections. If it finds a match in either the current menu or the global menu, it returns the opcode for the choice through the variable "iopt" which directs the program's course.

The menu processor has three global menu selections that allow for easier movement about the menus. The selections that are available from any menu are "/", "!", and "M". These available choices are not displayed on any menu but are always active. The slash command "/" will bring the user up one menu level. Selecting the exclamation point command "!", the user is brought up to the main menu, and entering the "M", the user views the re-displayed active menu.

If the selection made by the user was from the global menu, the value assigned by *get_option* to "iopt" is "-1" for "!" and "-2" for "/". For an "M" selection, a value is not returned to the main program through iopt. Instead, the *get_option* routine is instructed to re-display the active menu and wait for another input from the user. If the selection was found in the active menu, the opcode corresponding to that selection is returned as a positive integer value. The opcode value returned through "iopt" is first tested to see if it is a global menu selection. If it is, the displays the appropriate menu. If the selection was a positive value, however, the program is directed to the appropriate location by way of a "computed go to" FORTRAN statement operating on the opcode.

If however the routine can not find a match on either the active menu or the global menu, an error message is prompted to alert the user to an improper choice, and the module re-displays the active menu.

The *get_option* routine allows the user to input choices into the menu processor with multiple command input. If, for example, a user is familiar with the menu structure and wants to access location in the program that is selected by a menu not on the current menu, but on one several menus deep, he or she could then make the appropriate successive menu selections from the current menu, and the processor will take the user through the selections, provided all choices are valid. For example suppose the user wants to select a choice "TC" from the main menu and knows on the next menu he will select "RE". Instead of selecting "TC" and waiting for the next menu to be displayed before he makes his selection of "RE", he can instead input "TC RE" at the first menu. The successive selections must be delimitated by a space, all valid on the menu that would normally appear next, and the entire command line may not be more that 30 characters long. If more than 30 characters are input in the multiple command set, the program notifies the user of the error and also displays the last command executed.

3.2 Data Input

3.2.1 Data Base Structure

When performing either a tolerance analysis or a tolerance distribution, the analysis routines must have access to the data file created by the geometric model. The data file includes all of the necessary information that constitutes a tolerance chain. This data file is read by selecting "retrieve chain", "RE", from either the tolerance control or the tolerance distribution menu. The format of the data file can be seen in the sample data file in Figure 3.4. The entire file shown in Figure 3.4 is set off at the beginning and end by a value of "-1". These serve as sentinels to delimitate where a chain link begins and ends. This feature will allow for a future capability of analyzing interrelated chains, i.e. chains with common links or chains with common probability. Both of these cases have equations that must be simultaneously satisfied. The files for now have only one chain in them, but an example will be given in Chapter 4 that illustrates links with common probability.

The second line in the tolerance chain data file is the assembly type number. As was described in the previous chapter, the summing procedure employed relies on the type of sum dimension being considered. An assembly type number "1" , "stationary", indicates that the sum dimension occurs between two relatively stationary parts. An assembly type number "2", "stator-rotor", is used for a sum dimension between a rotating and a stationary part.

After the two initial entries, there remains only two different records in the data file. Each link in the chain is described by one set of two records. The first record is comprised of integer values only and is called the "type_params" record for the "type of parameters". The second record is called "lparams" record for the "link parameters record".

The first of these two records is always made up of six integer values. The first entry in the type_params record denotes the type of chain link for which the data follows. The tolerance chain can be made up of a total of 14 types of

```
-1
1
1,0,0,0,1,1
-1.00000,11.00000,0.00000,-0.05000
1,0,1,2,3,1
-1.00000,9.500000,0.00000,-0.03000
1,0,2,4,5,1
-1.00000,0.500000,0.00000,-0.10000
1,0,3,7,8,1
-1.00000,133.000000,0.00000,-0.10000
5,0,4,11,9,1
1.00000,43.000000,43.00000,0.023000,0.01600,0.016000,0.00900
1,0,5,12,13,1
-1.00000,380.00000,0.05000,-0.050000
5,0,6,15,16,1
1.00000,100.000000,100.00000,0.035000,0.00000,-0.04000,-0.07500
1,0,7,16,17,1
-1.00000,65.00000,0.05000,-0.050000
6,0,8,19,17,1
1.00000,90.000000,90.00000,0.035000,0.00000,0.02500,0.00300
3,0,9,19,18,1
1.00000,0.00000,0.012000,0.000000
6,0,10,21,18,1
1.00000,190.000000,190.00000,0.046000,0.00000,0.00200,-0.03800
3,0,11,21,20,1
1.00000,0.00000,0.025000,0.000000
15,0,12,22,20,1
1.00000,255.000000,255.00000,0.046000,0.00000,0.00000,-0.03000
1,0,13,22,23,1
1.00000,580.00000,0.00000,-0.140000
1,0,14,25,26,1
1.00000,19.00000,0.00000,-0.100000
1,0,15,29,30,1
1.00000,0.80000,0.10000,0.000000
-1,0,0,0,0,0
```

Figure 3.4 Sample tolerance chain data file

individual chain links. The type of chain link that is read in is used to determine which link equations to use and therefor which subroutine is called to calculate the parameters for that link. A list of the link types and their link type numbers is listed in Table 3.1.

The second entry as well as the fourth entry in the type_params record is used in the geometric modeling program but not in the tolerance analysis module. It is read here and is written back to any updated data files sent back to the geometric modeling program. The third entry is a link number pointer, counting the links beginning at link number zero.

The fifth entry in the type_params record indicates the location of the chain link. As discussed in the section on summing between a stator and a rotor, the summation on each component is done separately. The fifth entry in the type_params designates whether the part is on the stator, or the rotor, or is a member of a stationary summation. The values denoting component location are:

0 - stationary
1 - stator
2 - rotor

The sixth and last entry in the type_params record signifies the links' determinability for a tolerance distribution. The link types for a tolerance distribution are categorized into two classes: links with determinable tolerances and links with predetermined tolerances. The sixth entry in the type_params record is:

0 - for links with undetermined tolerances.

1 - for links with predetermined tolerances.

Table 3.1 Link types and numbers

1 - line vector span

2 - plane vector span with lumped direction

3 - eccentricity

4 - space vector span

5 - clearance
(line vector gap with lumped magnitude)

6 - transition
(line vector gap with lumped magnitude)

7 - Normally distributed center location
(line vector gap with distributed magnitude)

8 - rectangularly distributed center location
(line vector gap with distributed magnitude)

9 - clearance
(plane vector gap with lumped direction and lumped magnitude)

10 - transition
(plane vector gap with lumped direction and lumped magnitude)

11 - normally distributed center location
(plane vector gap with lumped direction and distributed magnitude)

12 - rectangularly distributed center location
(plane vector gap with lumped direction and distributed magnitude)

13 - plane vector gap with distributed direction and lumped magnitude

14 - bivariate normal distributed center location
(plane vector gap with distributed direction and distributed magnitude)

15 - bivariate rectangular distributed center location
(plane vector gap with distributed direction and distributed magnitude)

The size and contents of the lparams record are contingent on the individual link. In all there are three variations in the size of the record and seven variations in the contents of the record.

In all variations of the record, the first entry is the geometry transformation constant "A" discussed in section 2.4.1. For link type 1, a line vector span, there are four entries in the lparams record. The entries in order from the second to the fourth are:

link type 1: X, UX, LX

2) X - the parts nominal dimension

3) UX - upper deviation

4) LX - lower deviation

There are seven entries for link type 2, plane vector span with lumped direction. They are from the second to the seventh:

link type 2: X_L, UX_L, LX_L, ANG, UANG, LANG

2) X_L - spans nominal dimension

3) UX_L - upper deviation of span

4) LX_L - lower deviation of span

5) ANG - angle of span with sum dimension

6) UANG - upper deviation of angle ANG

7) LANG - lower deviation of angle ANG

For the plane vector span with lumped direction, the values of the upper deviation of the angle (UANG) and lower deviation of the angle (LANG) are not used in the link parameter calculations in this program. The variation resulting from uncertainty in the angle is assumed small as compared to the span, and the fundamental equation link type two is linearized by the above assumption. They are included, however, if the situation arises in which angular variation is not negligible. A routine could easily be amended to account for the situation.

Link type 3, an eccentricity, has four parameters which are from the second to the fourth:

link type 3: X_R, UX_R, LX_R

2) X_R - nominal size of eccentricity

3) UX_R - upper deviation of eccentricity

4) LX_R - lower deviation of eccentricity

Link type 4 is retained for the space vector span. In the actual program the space vector span is never used; it is broken down into its constituent components as described in Chapter 2. The type number is however left on reserve in the event of future utilization.

Link types 5 - 8 all have the same seven parameters. The link names are:

link 5: line vector clearance.

link 6: line vector transition.

link 7: line vector gap, rectangularly distributed center location.

link 8: line vector gap, normally distributed center location.

link types 5 - 8: X_B, X_A, UX_B, LX_B, UX_A, LX_A,

X_B - nominal size of bore.

X_A - nominal size of shaft.

UX_B - upper deviation of size of bore.

LX_B - lower deviation of size of bore.

UX_A - upper deviation of size of shaft.

LX_A - lower deviation of size of shaft.

Link types 9 - 12 all have eight parameters and are the plane vector gap relatives of link types 5 - 8. Entries 2 through 8 in the record lparams for links of types 9 - 12 are listed in order as:

link types 9 - 12: ANG, X_B, X_A, UX_B, LX_B, UX_A, LX_A,

ANG - the angle of the gap with the sum dimension

X_B - nominal size of bore

X_A - nominal size of shaft

UX_B - upper deviation of size of bore

LX_B - lower deviation of size of bore

UX_A - upper deviation of size of shaft

LX_A - lower deviation of size of shaft

Link types 13 - 15 are distributed direction links, and all have the same seven parameters. The names of the links are:

link 13: plane vector gap with distributed direction and lumped magnitude.

link 14: bivariate normal distributed center location.

link 15: bivariate rectangular distributed center location.

Entries 2 through 8 for a link of type 13 through 15 are listed identical in both number and content as those listed for links 5 - 8. A summary of the different record types is listed below in Table 3.2

Table 3.2 Summary of records in array lparams

link type 1:	A, X, UX, LX
link type 2:	A, XL, UXL, LXL, AND, UANG,LANG
link type 3:	A, X, UAR, LXR
link type 4:	(nil)
link types 5 - 8:	A, XB, XA, UXB, LXB, UXA, LXA
link types 9 - 12:	A, ANG, XB, XA, UXB, LXB, UXA, LXA
link types 13 - 15:	A, XB, XA, UXB, LXB, UXA, LXA

The tolerance chain data is read from the data file by invoking the "RE" command, i.e., "RETRIEVE CHAIN" command, from either the tolerance control or the tolerance distribution menu, which calls the subroutine *get_chain*. It can be seen in the main calling program *TASM* that *get_chain* has only one argument, and it is of variable type logical. This variable is a flag that regulates program operation after reading the data file by flagging whether the data file was without error. The flag is initially set to false, which will not let the program perform any other steps in a tolerance analysis or tolerance distribution before a chain has been read in. If the program were to try to perform some of the calculations that need the data in the tolerance chain data file before it had read it, it would in many cases "bomb", and in others give erroneous output. Once the chain has been read successfully however, the chain flag is set to true, and any other points in the program that need the information now know that it has been read in without error and is available.

The subroutine *get_chain* initially calls the subroutine *open_file*, which prompts the user to choose the chain to be analyzed and opens the file for reading in *get_chain*. If there is an error in opening the file specified, the user is prompted that the file is not in the directory and is asked to re-enter the file name. The convention used for tolerance chain data files is to use a ".CHN" file extension. The program prompts the user with a default data file which is currently set to the compression chamber example used in Chapter 5. The user need only hit return to accept the default. This default can be changed by editing the routine *open_file* source code.

After the file is opened, control returns to *get_chain*. The routine reads the tolerance chain data file until it encounters a "-1" which tells it that it is at the beginning of a chain file. With the sentinel, headers could be placed in the top of the data file that would not affect the reading of the file. After the "-1" is picked up, the type of sum dimension, i.e. stationary or stator - rotor, is read first. *Get_chain* then proceeds to read the type_params of the first link. It reads the six parameters described above and then continues on to read the lparams record. The routine then reads either four, seven, or eight parameters for the link, depending on its type that it has just read from entry 1 in type_params. The routine continues until it encounters another "-1", signalling the end of the file. Parameters of the links in the tolerance chain that are read in, as well as the counted number of links, are placed in the common block /chain_par/ to be accessed by the necessary routines. If at any time there is an error in the reading of the tolerance chain data file, an error message is displayed, and the value of chain_flag is returned as false.

3.2.2 User/Keyboard Input

The data necessary for the program to perform a tolerance control or tolerance distribution are not yet complete. So far only the geometrical data have been input via the tolerance chain data file created by the geometric modelling package. Statistical data needed for the analysis have not been provided.

Logically, the statistical data are not located in the geometric modelling data base. The statistical data therefore must be input by the user in the tolerance analysis module. In order to determine completely the parameters of the individual links, the means and variances of the links must be known to the link parameter routines.

The link means and variances are input after the tolerance chain has been read. Before the user can select "LS" for "Link Statistics" menu choice from either the tolerance control or tolerance distribution menu, a tolerance chain data file must have been successfully read, and the chain flag set true. With the chain flag set true, the link statistics can be read in. By choosing " LS" , the routine *crlmv* is invoked. This routine involves merely a calling routine to another routine *read_lmv* which actually reads the link means and variances. When the link means and variances are read in, they are placed in the common block /chain_par/ so that any routines that need them have only to access that common block. The main calling program *TASM* was composed in such a way that it is more a director of calculations and procedures than a performer of them. *Crlmv* was created was to ensure that procedural common blocks did not have reason to exist in the main program.

When the user chooses "LS", the module asks the user if the data are to be input or to be read from an existing file. Initially, when the tolerance chain data file is first received from the geometric modelling module, the link means and variances have not yet been introduced to the tolerance module. The user then proceeds to input the link means and variances from the keyboard.

After the link means and variances have been input by the user, the module asks if the user desires to save the link means and variances to a file. If the question is answered affirmatively, the module proceeds to save automatically the link means and variances to a file that has the same name as the tolerance chain data file, with the exception of the extension. For the case of the link means and variances data file, the extension appears as "*filename*.LMV" Once the data has been input by the user via the keyboard and saved to a file, the next time an analysis is run using the same means and variances, the user need only to retrieve them from the file that has been created, thus avoiding the labor of inputting them from the keyboard. Since the tolerance chain data file has been read prior to reading the link means and variances, the module knows the name of the file. When the user selects to retrieve an ".LMV" file, the module looks for a file with the file name of the current ".CHN" file with the extension ".LMV". If the module cannot find the appropriate file in the directory, it alerts the user and proceeds to take the user through a session to input the data from the keyboard.

The *read_lmv* routine that actually reads the data from the keyboard. Having access to the data in the common block /chain_par/, the routine knows how many links there are and what type of links they are. The routine prompts the user as to the link number and to what type of link the data are being read. The routine can prompt for data in two ways depending on what type of link data are being acquired. If the link is a span, then the routine prompts for only one mean and variance pair. If the link is a gap, however, the routine prompts for the means and variances of both components that make up the gap. As with all of the data input, the data input in the *read_lmv* routine are checked to see if it is the appropriate type that the routine is expecting. If it is not, the user is prompted to re-enter the data, thus insuring that erroneous data do not reach the analysis routines.

After all of the link means and variances have been read in from either the keyboard or a file, the program control returns to the main calling program where the link statistics flag "l_stat" is set to true. This flag is used at various other points in the program to check if the link means and variances have been read in. If they were not if would reek havoc on certain analysis routines and cause the program to crash or yield erroneous results both of which are highly unacceptable.

"Chain Link Routines" describes the implementation of section 2.4 into computer code. The link routines are located in the object module library named l_class.olb for "link classifications". A list of the names of the routines in this library and the link types to which they pertain is included as Table 3.3. Some of the link routines provide support for more than one type of chain link.

Table 3.3 Link routine names

Subroutine	Link type
lspan	line vector span
pspan	plane vector span
eccentricity	eccentricity
l_clearance	line vector clearance
l_transition	line vector transition
l_dist_ctr	line vector gap, normally distributed magnitude
	line vector gap rectangularly distributed magnitude
p_clearance	plane vector clearance
p_transition	plane vector transition
p_dist_ctr	plane vector gap, normally distributed magnitude
	plane vector gap, rectangularly distributed magnitude
pgap_dd_lm	plane vector gap, distributed direction, lumped magnitude
bivar	bivariate rectangularly distributed bivariate normally distributed

The links that do support more than one type of link branches internally depending on the link number which is available in the common block. The following pages will show how this can be done with the link pointer that is sent down to the routines.

Given the data input to the module as described in section 3.2, the effects of the chain links on the sum dimension can be calculated. The parameters of the links that are to be calculated are: distance to the middle of the range of the dimension, the range of the dimension, the expectation of the stochastic part of the dimension, and the variance of the stochastic part of the dimension. These parameters are transformed according to direction and location, and their effects on the sum dimension are summed up accordingly. The purpose of the link routines is only to determine the parameters of the individual links. Since the summing procedure is dependent on the geometry of the assembly and the link routines do not do any summing, they remain functional for any geometry. That is to say, the link routines can be called upon to deliver the parameters of the individual links by any summing routine for the geometry considered. Following sections will clarify how the summing routines for the geometry of a stationary part and the geometry of stator - rotor combination make use of the link routines.

Each of the link routines contains the common block /chain_par/ which holds the link's geometrical and statistical data. The arrays in the common block have been dimensioned to accommodate up to 25 chain links and the maximum number of parameters for any individual link. Element arrays were chosen to be dimensioned at 25 so as not to waste computer resources. For any of the examples considered, all arrays of 25 elements proved to be sufficient. The size of the arrays can easily be expanded if circumstances should require such a modification.

The information for an individual link is extracted from the common block /chain_par/ by way of a link pointer that is sent down to the link routine when the link routine is called. This pointer identifies to the values in the arrays in the common block that are associated with the particular link under consideration. Since all of the link data have been read into the arrays in order, the link routine can access the links values and correct number of values by way of the link pointer. For example, the array that holds the link means and variances is dimensioned as a 4 by 25 two dimensional array. In order to accommodate the links that have a set to two means and variances. It can be seen that for a link that has only one mean and variance pair, the third and fourth element are not filled after reading the data. The link routine does not have to know what size the array is or how many elements are present in it. Since the data were read in order and the link routines will be called in the same order, the correct number of values will be present for the link as pointed to by the link pointer. For example, a line vector span type link needs only one mean and one variance. The line vector span routine reads the mean and then the variance, and then proceeds, ignoring the last two empty elements in the array that would be reserved for a link that needs them.

The same holds true for the link parameters array "lparams" which is dimensioned as 8 by 25. As described in section 3.2.1, some of the links have four parameters, some have seven parameters, and some have eight parameters. The link routines that call for fewer than eight parameters ignore the remaining empty memory locations that have been set aside for the link in the dimensioning.

The first step that takes place in all of the link routines is the assignment of variables to the means and variances in the array lmv(4,25). Next, the routine assigns variables to the link's geometrical data present in the array lparams(8,25). Once the variable assignment has been performed, the routines have the data in a state that can be used to calculate the link's contribution to the sum dimension. Before the data can be input to the equations presented in section 2.4, some preliminary calculations carried out to transform the data obtained from the geometric modelling data base. The procedures used to obtain values that can be used in the link equation from the data in the array lparams will be described in section 3.3.1. The calculations for the link parameters, however, will not be repeated from section 2.4. The operation of two link routines that interact with the user for additional input will be discussed further detail in section 3.3.2.

3.3.1 Preliminary calculations.

The preliminary calculations that are carried out transform some of the data in the tolerance chain data file into a form useful for the link equations. For instance, the link routines need the tolerances of the dimensions of the links. This information is not found directly in the tolerance chain data file but can be determined from the upper and lower bounds of the dimensions that are located in the tolerance chain data file.

For span type links, there is only one dimension; therefore only one tolerance must be calculated from the dimensional limits. The equation is of the form:

$$TX = |(UX - LX)|.$$

For a gap type link that is made up of two distinct dimensions, the link equations need both the tolerance of the shaft (TX_A) and the tolerance of the bore (TX_B).

$$TX_A = |(UX_A - LX_A)|.$$

$$TX_B = |(UX_B - LX_B)|.$$

Similarly the middle of the range of the link's dimensions must be calculated from the existing data in the tolerance chain data file. The effect of a span type link has only one dimension and is calculated as follows:

$$MX = X + (UX + LX)/2.0.$$

The effect of a gap type link, however, must be determined from the middle of the range of both component parts of the gap. The calculation is comparable to the calculation for the span type link but includes an equation for each of the parts of the gap type link.

$$MX_A = X_A + (UX_A + LX_A)/2.0.$$

$$MX_B = X_B + (UX_B + LX_B)/2.0.$$

After the values for tolerances and the middle of the range of the link's component parts have been calculated, the link routine can continue to perform the calculations for the link's effect on the sum dimension, using the equations provided in section 2.4.

3.3.2 Interactive Link Routines

There are two link routines that need some additional data to determine their influence on the sum dimension. These link routines are for gaps with lumped magnitude, the line vector transition and the plane vector transition. The plane vector transition is identical to the line vector transition routine except for a substitution of "A cos(α)" for "A", as described in section 2.4.2.2. The discussion that follows here is valid for both types of chain links. In order to determine the effect of the transition link on the sum dimension, the values of ζ_{EX} and ζ_{VarX} must be determined from Figures 2.16 and 2.17. After the routine has performed the preliminary calculations and has determined TX_A, TX_B, and MX_A, and MX_B, the routine calculates EX_R and $VarX_R$, as explained in section 2.4.2.2. The value of ζ is then calculated and displayed on the screen. Given the value of Z, the user can retrieve the values of ζ_{EX} and ζ_{VarX}, as seen in Figures 2.16 and 2.17. The routine prompts the user for the input of the two values obtained from the plots and can proceed when they are input. The user must decide whether the transitions are rectangularly distributed or normally distributed to determine which curve to use in acquiring the values from the figures.

3.4 Tolerance Analysis (Control)

A tolerance analysis is performed when it is desired to determine the confidence level of an assortment of parts, whose tolerances are known or assumed, to be assembled to specification. By "assumed" it is meant that the tolerance is assigned for the analysis without really knowing what the tolerance is or what it should be. A later section of this paper will show how this module can be manipulated to be used as a tool to perform a "what if" type of analysis. That is to say, a designer could observe the direct consequence of altering a dimension's tolerance by making a change and performing another analysis.

The tolerance analysis section of the module is executed by the user's choosing the tolerance control selection "TC" from the main menu. The module then prompts the user with the "tolerance control" menu shown below in Figure 3.5.

```
TOLERANCE CONTROL

RE - Retrieve Chain
LS - Link Statistics
AN - Analyze Chain
SA - Save Results
```

Figure 3.5 Tolerance control menu.

Before the analysis can be carried out, the module must access an assembly's dimensional and statistical data in the form of a tolerance chain and the assembly link's means and variances, as mentioned in section 3.3. The module maintains the status of several data input flags to insure the user cannot call for an analysis before the appropriate data are available for the analysis routines.

When all of the necessary data has been input, the module allows the user to select the analyze chain entry "AN" from the tolerance control menu. By selecting "AN", the user invokes the routine *analyze_chain* which determines the parameters of the sum dimension and the confidence level of the assembly based on the functional requirement and calls the appropriate routines to display the results. Output of the results is displayed in graphic format of a plot of the sum dimension's probability distribution function with its confidence area filled in. The remainder of this section will discuss the user's interaction and the operation of the analysis routines in performing a tolerance analysis.

3.4.1 Analyze Chain routine

Analyze_chain calculates the parameters of the beta distribution model of the sum dimension by calling the appropriate summing routine to sum both the dimensional and statistical parameters of the individual dimensions. In order to do so, the routine must know what type of sum dimension is being considered, i.e., does the sum dimension lie between two relatively stationary parts or between a stationary part and a rotating part. The *analyze_chain* routine has in its code the common block /chain_par/ so it has access to all of the chain's link parameters including the value "sum_type" which is the sum dimension type that was read from the tolerance chain data file. By way of the FORTRAN "computed go to" statement, the routine branches to the proper summing routine based on the value of "sum_type". The two types available for analysis are routines *sum_ss* and *sum_sr* which stand for "sum stationary" and "sum stator - rotor" respectively. These two routines return the statistical parameters of the beta distribution to *analyze_chain* in order to model the sum dimension.

After the distribution of the sum dimension has been established, the last step in performing a tolerance analysis is to integrate the distribution over the desired tolerance zone of the sum dimension. The user inputs the desired tolerance zone for the sum dimension, and *analyze_chain* calls the integrating routines. Results of the analysis are then displayed on the terminal screen in graphics mode in the form of a plot of the distribution of the sum dimension with the input tolerance zone shaded and the confidence level displayed. A detailed discussion of the routines called by *analyze_chain* is included in the sections to follow.

3.4.1.1 Sum_ss routine

The *sum_ss* routine is called when the sum dimension occurs between two surfaces that are stationary relative to each other. The routine *sum_ss* also has access to the common block /chain_par/ so its input parameters are not included in the subroutines argument list; instead they are obtained directly from the common block that was filled when the link data was read in.

Subroutine *sum_ss* calls the link routines discussed in section 3.3 to calculate the individual link's parameters. A FORTRAN "computed go to" statement which controls this routine, is in a loop that counts through the number of links in the tolerance chain. The "computed go to" statement directs the flow of the *sum_ss* routine to the proper link routine for the current chain link being calculated. *Sum_ss* calls one of the eleven routines described in section 3.3 for each link in the tolerance chain and sums them up. The manner in which the parameters are summed up is discussed in section 2.4.3.

For the stationary situation that is considered in *sum_ss*, the individual link parameters calculated by the link routines are arithmetically summed as they are calculated. As was mentioned above, the subroutine *sum_ss* does not have any input in its argument list, but the output of the sum dimension's statistical parameters is returned through the argument list. The beta distribution parameters returned by the summing routine *sum_ss* for the sum dimension are:

A: lower bound of the range of the distribution.

B: upper bound of the range of the distribution.

EX: expectation of the distribution.

var: the variance of the distribution.

The parameters above are calculated from the sum totals of the link parameters upon completion of the loop that calls the link routines. The upper and lower limits of the range of the distribution of the sum dimension are calculated from the middle of the range of the sum dimension, and the range of the sum dimension.

$$A = MX_{\Sigma} - R\Delta X/2.0$$

$$B = MX_{\Sigma} + R\Delta X/2.0$$

The expectation and variance of the distribution can be calculated from equations 2.2 and 2.3.

$$EX = MX_{\Sigma} + E\Delta X_{\Sigma}$$

$$VarX = Var\Delta X_{\Sigma}$$

3.4.1.2 Sum_sr routine

The summing routine *sum_sr* is a bit more complex than the summing routine *sum_ss*; this increased complexity might be expected considering the development of the summing procedure for this case back in section 2.4.3.2. The procedure developed in section 2.4.3.2 assumes that all of the "sum slant" type links are all line or plane vector spans. A "sum slant" is a chain link that has an expectation different than zero. The equations derived in this section allow for the analysis of a sum dimension that is comprised of any kind of chain link for the

stator - rotor case. The detail of performing the summation of a stator - rotor case will not be repeated, but the derivations of the necessary equations will be included, and the operation of the routine *sum_sr* will be discussed.

The statistical parameters of the sum dimension that are returned by the summing routine *sum_sr* are the same as those returned by the summing routine *sum_ss*, namely, A, B, EX, VarX. The direction of concern for the stator - rotor case is in the radial direction. The parameters returned by *sum_sr* are calculated in the radial direction by the method described in 2.4.3.2. The rest of the analysis routines are independent of the location of the sum dimension. They calculate the distribution of the sum dimension and the tolerance zone's confidence level based on the parameters given by *sum_ss* or *sum_sr*.

The summing procedure for the stator - rotor case is carried out in a different order than is the stationary case in *sum_ss*. In the routine *sum_ss*, the individual link parameters were calculated and immediately added to the sum. By contrast, the summing routine *sum_sr*, all of the individual link parameters are calculated first and stored in arrays to be used in the more complex summing procedure. The routine *sum_sr* contains the common block \chain_par\ which permits it to call the appropriate link routines for each link in the chain.

After all of the individual link parameters have been calculated, the routine begins the task of determining the parameters of the sum dimension. The individual link statistics must be summed on the stator and rotor separately and then combined in the manner outlined in section 2.4.3.2. to obtain the statistics of the sum dimension (VΔX) in the radial direction. The parameters in the radial direction are summed by two subroutines called by *sum_sr*. The routines *sr_slant* and *sr_ecc* are used to calculate the contributions in the radial direction of the "sum slant" and "sum eccentricity" respectively. The routine loops over all of the links in the chain and calls the appropriate routine, *sr_slant* or *sr_ecc*, in a FORTRAN "computed go to" statement.

After the *sum_sr* routine has calculated all of the individual links' parameters influence on the sum dimension in the radial direction for the stator and rotor respectively, it combines them to determine that parameters of the beta distribution. The routine then returns the limits of the range of the distribution (A and B) and the expectation and variance of the distribution. The expectation will be returned for the large clearance for the sum dimension; i.e., the expectation of the stochastic part of the dimension is added to the middle of the range to obtain the expectation for the dimension.

Sum slants

If the link in question is of type 1, 2, 5, 6, 9, or 10 (see Table 3.1) , i.e., the link has an expectation different than zero, *sum_sr* calls *sr_slant* the input to this routine is the individual link parameters varΔX, EΔX, MX, RΔX, as well as a pointer to the link in question. The input is passed down to the routine via the subroutine argument list. The routine *sr_slant* calculates the parameters in the radial direction by breaking down the parameters into their X and Y components. If the link is a line vector, type 1, 5, or 6, the Y component of the parameters is zero, and the covariance between the X and Y component is also zero. In this case, only the X direction has an influence. The contribution of the link is added to the X sum.

For links that are plane vectors, 2, 9, or 10, the Y component must also be considered and the link may have a covariance between the X and the Y component. First the routine calculates the Y component of the expectation of the stochastic part of the dimension, EΔY, the variance, VarΔY, and the middle of the tolerance zone and the range MY, RΔY. After the Y parameters of the link are determined, the routine proceeds to determine the covariance of the link.

There are three different possible cases for the covariance calculation. The equations for the covariance of a plane vector span are included in the section 2.3.4.2 and will not be derived here and repeated. Covariance equations for a plane vector gap that is a clearance, and a plane vector gap that is a transition are derived in the same manner and the results are show below.

$$cov(x,y) = E(x,y) - E(x)E(y)$$

$$var(x) = E(x^2)-[E(x)]^2$$

<u>slant</u>

$$\Delta x = A_i\ TX_i \cos\alpha_i\ (z_i - 0.5)$$

$$\Delta y = A_i\ TX_i \sin\alpha_i\ (z_i - 0.5)$$

$$cov(\Delta x,\Delta y) = E(\Delta x \cdot \Delta y) - E(\Delta x)\ E(\Delta y)$$

$$= E[\{A_i\ TX_i \cos\alpha_i(z_i-0.5)\}\{A_i\ TX_i \sin\alpha_i(z_i-0.5)\}]- [E\{A_i\ TX_i \cos\alpha_i(z_i-0.5)\}\ E\{A_i\ TX_i \sin\alpha_i(z_i-0.5)\}]$$

$$= E[\{A_i^2\ TX_i^2 \cos\alpha_i\sin\alpha_i(z_i-0.5)^2\}]- [\{A_i\ TX_i \cos\alpha_i(Ez_i-0.5)\}\ \{A_i\ TX_i \sin\alpha_i(Ez_i-0.5)\}]$$

$$= A_i^2\ TX_i^2 \cos\alpha_i\sin\alpha_i\ \{E[(z_i - 0.5)^2 - (Ez_i-0.5)^2\}$$

$$= A_i^2\ TX_i^2 \cos\alpha_i\sin\alpha_i\ \{E(z_i^2) - [E(z_i)]^2\}$$

$$cov(\Delta x,\Delta y) = A_i^2\ TX_i^2 \cos\alpha_i\sin\alpha_i\ var(z_i)$$

<u>transition</u>

$$cov(\Delta x,\Delta y) = A_i^2 \cos\alpha_i\sin\alpha_i\ \zeta varX\ varX_R$$

$$= A_i^2 \cos\alpha_i\sin\alpha_i\ varX\Sigma$$

<u>clearance</u>

$$cov(\Delta x,\Delta y) = A_i^2 \cos\alpha_i\sin\alpha_i\ varX_L$$

When the link parameters are completed with the calculation of the covariance, the routine determines whether the link is on the stator or rotor. The links' influence

on the sum dimension are summed on the stator and rotor independently and must be added to the proper component. The link's type parameters "type_params" were read during the link retrieval from the tolerance chain data file and stored in the common block /chain_par/. ***Sum_slant*** determines whether the link is on the stator or rotor by checking the second entry in the link's type parameters array. If the second entry is a "1", the link is on the stator; if it is "2", the link is on the rotor. The routine returns an updated sum of either the stator or rotor for values of expectation, variance, range, and middle of the range for both the X and Y components and also the covariance.

Sr_ecc

For links that are considered sum eccentricities, i.e., those that have an expectation of zero, *sum_sr* calls the subroutine *sr_ecc*. Input to the routine include the values of varΔX, RΔX, that were calculated by the link routines, and a pointer to the link under consideration. The middle of the range, MX, and the expectation are both zero so they are not sent to the subroutine. Output through the argument list is the sum of the variance for the sum eccentricities called thus far. The final value for the variance of the assembly for sum eccentricities is the algebraic sum of the variance of the stator and that of the rotor. The variance for sum eccentricities is therefore not distinguished between for the stator and rotor and are summed up in one value. The range of the sum eccentricities is calculated in three different ways for the various link types. Link types 7 and 8 are both line vector type links which place the components in the X direction. Link types 11 and 12 are both plane vectors and have an X and a Y component. Components of the range are calculated in both the X and the Y direction and are stored in the summation variable rdx_sum and rdy_sum which are located in a common block named /radial_range/. Also present in routine *analyze_chain* is the common block

/radial_range/ which gives *analyze_chain* access to the values of the ranges of the sum eccentricities.

3.4.2 Remainder of analyze chain routine

The routines described thus far are called upon to provide the *analyze_chain* routine with the parameters of the sum dimension, namely A, B, EX, varX. The range of the distribution and the middle of the range of the distribution of the sum dimension are displayed on the terminal for the user. The user is then prompted to enter the desired tolerance for the sum dimension. By inputting the desired tolerance of the sum dimension, the user provides the module with the range of the distribution that the integration routines will integrate to yield the number of the assemblies actually within the desired bounds. The user inputs the tolerance as the size of the tolerance zone and also the middle of the tolerance zone. The parameters of the beta distribution γ (gamma) and η (eta) are then calculated from the equations 3.2 and 3.3. For a brief outline of the beta distribution see Appendix V.

THe routine check the tolerance zone values input by the user to see if they are within the limits of the range of the distribution. If either of the limits of the tolerance zone is outside of the range of the distribution, the corresponding limit of the range is substituted for the limit on the tolerance zone. The value of the integration of every probability density function over its entire range is known to be one. This fact is used to create an integration scheme that is very efficient and at the same time accurate. The scheme used to integrate the probability density function is the Gauss - Legendre method. For distributions that possess singularities in the form of spikes in the p.d.f., the number of gauss points is very important to maintain accuracy in the integration.

The integration routine begins with ten gauss points and compares the value of the integration to unity. If the value for the integration falls within a certain specified error value, the integration routine uses 10 gauss points when integrating between the limits of the tolerance zone. If the value of the integration is not within the specified error, the routine adds two more gauss points and performs the integration again. The integration scheme proceeds in this fashion until the value of the integration of the range of the distribution is close enough to unity. When the error tolerance has been matched, the routine integrates over the tolerance zone with the number of gauss points calculated for the entire range. This procedure provides enough gauss points in the tolerance zone to provide the accuracy desired without wasting computer time in calculating some preset number of gauss points. The details of the integrating scheme will be discussed in the sections to follow.

The last step the routine *analyze_chain* performs presents the results to the user. *Analyze_chain* calls upon subroutine *plot_dist* to display the results of the analysis. Subroutine *plot_dist* accepts the statistical parameters of the distribution and the limits of the tolerance zone and calculates the points on the curve of the distribution. With the points on the distribution calculated, *plot_dist* calls the plotting routine *tolplot* that displays the distribution, the input tolerance zone, and the results of the integration. The operation of these routines will be discussed in detail in their own section individual sections and their output will be demonstrated in Chapter 4.

When "AN" is selected from the tolerance control menu, the analyze chain menu shown below in Figure 3.6 is presented to the user. To perform a first time analysis, the user must choose "CD" for calculate distribution; that choice initiates the *analyze_chain*, as was discussed above. The distribution of the sum dimension is calculated from the input data known of the chain links.

ANALYZE CHAIN
CD - Calculate Distribution TZ - Tolerance Zone

Figure 3.6 Analyze Chain Menu

After the distribution has been determined for the sum dimension and the results have been displayed, the user may want to change the location or size of the sum dimension and re-display the new confidence. Since the distribution has already been calculated, it is unnecessary to completely execute *analyze_chain* again. Instead, by selecting "TZ" from the analyze chain menu, the user instructs the module to enter *analyze_chain* at a point after the distribution has already been calculated. A new value for tolerance zone and middle of the tolerance zone is prompted for by the module and upon input, the new confidence level is displayed. Different tolerance zones and locations can be experimented with by using this feature without wasting the time to recalculate the entire distribution of the sum dimension.

3.4.3 Saving Analysis Results

When the user is satisfied with the results of an analysis, all of the parameters of the distribution as well as the confidence level determined by the module can be saved to an output file. The parameters can be saved by selecting "SA" for "Save Analysis Results" from the tolerance control menu (Fig. 3.5). This file is written in the main directory from which the module is running. By default the user will be prompted with the name of the file "filename.out", where "filename" is the name of the tolerance chain data file originally read in. If the default file name is suitable, the user only has to hit the return key. However, if the user wants to change the name of the file, the change can be executed at the prompt. A sample of the saved output file is listed in Chapter 4.

Pragmatically, the confidence level of an assembly is the percentage of assemblies that coincide with the specification set forth so that the assembly performs satisfactorily. This specification is designated by way of assigning a tolerance to a specific functional dimension of a assembly. If the dimension falls within the tolerance range specified by the designer, the assembly is considered satisfactory. From a mathematical standpoint, the percentage of acceptable assemblies is represented as a reliability, the area of integration under a probability distribution function bounded by the tolerance zone. The manner in which the parameters of this function are determined has been presented in the preceding sections. This section continues to present the manner in which the parameters of the sum dimension are used to create the beta function distribution for the sum dimension, and the ways in which the function is integrated to yield the confidence level for the desired tolerance zone.

The beta probability distribution function on the interval [a,b] is of the from:

$$f(x) = \frac{1}{(b-a)\, B(\gamma,\eta)} \left(\frac{x-a}{b-a}\right)^{\gamma-1} \left(1 - \frac{x-a}{b-a}\right)^{\eta-1} \qquad \text{(eq. 3.1)}$$

The parameters γ and η are calculated after the either summing routine returns control to *analyze_chain* by the equations:

$$\gamma = \frac{(EX - a)^2\,(b - EX) - varX\,(EX - a)}{varX\,(b - a)} \qquad \text{(eq. 3.2)}$$

$$\eta = \frac{(EX - a)\,(b - EX)^2 - varX\,(b - EX)}{varX\,(b - a)} \qquad \text{(eq. 3.3)}$$

The beta function in equation 3.1, $B(\gamma,\eta)$ is known to be related to the gamma function by the relation:

$$B(\gamma,\eta) = \frac{\Gamma(\gamma)\,\Gamma(\eta)}{\Gamma(\gamma+\eta)}$$

For any $X > 0$, $\Gamma(X)$ is defined by

$$\Gamma(X) = t^{x-1}e^{-t}\,dt.$$

The function has an interesting property that is described as follows:

$$\Gamma(X+1) = X\,\Gamma(X) \quad \text{for all } X > 0.$$

This property can be proven by a simple integration by parts.

$$\Gamma(x+1) = \int_0^\infty t^x e^{-t}\,dt = \lim_{R\to\infty} \int_0^R t^x\, e^{-t}\,dt$$

$$= \lim_{R\to\infty}\left\{[t^x(-e^{-t})]_0^R - \int_0^R - e^{-t}\,dt\; x t^{x-1}\,dt\right\}$$

$$= x\int_0^\infty t^{x-1}\, e^{-t}\,dt = x\,\Gamma(x)$$

All of the routines involved in performing the integration are controlled by *analyze_chain*. Since the equation for the beta function requires the gamma function, the routine *gamma_func* was written. *Gamma_func* returns the value of the gamma function of the input given to it. Since the values of the gamma function can sometimes be quite large, the subroutine *gamma_func* returns the natural log of the gamma function. Details of numerically calculating the gamma function can be found in any numerical methods text. The discussion presented by Press et. al.[11] is quite clear and concise. The beta distribution function has been rearranged in the computer code for an efficient solution of the equations. The subroutines used for calculating the gamma function and the beta function were developed by Nathan Graham at M.I.T. [12]. The beta function exists as a FORTRAN function named *beta*, called by the integration routine *qgauss_gen* which stands for "generic Gaussian integration routine". In the function *beta*, the actual beta function has the form:

beta = exp(C+(gamma-1)*log((X-A)/(B-A))+(eta-1)*log(1-(X-A)/(B-A)))

where gamma and eta are calculated from equations (3.2 and 3.3). The constant C is calculated in *analyze_chain* and passed to *beta* along with gamma, eta, A, and B in the common block /pdf/. C is determined from the equation:

$$C = \Gamma(\gamma+\eta) - \Gamma(\gamma) - \Gamma(\eta) - \ln(B-A) \qquad \text{(eq. 3.4)}$$

It can be seen that, if the exponentiation is carried through in equation (3.4), the resulting equation is indeed equation (3.1).

The beta function is now fully established and needs only to be integrated. The choice for an integration scheme was a Gaussian quadrature scheme whose order is, essentially, twice that of the Newton - Cotes formula (e.g. Simpson's rule) with the same number of function evaluations. More specifically, the integration method chosen was the Gauss - Legendre integration which is often referred to as simply Gaussian integration.

The routine that actually performs the integration *qgauss_gen* is a modified version of subroutine *qgauss* from "Numerical Recipes, the Art of Scientific Computing", Press, et. al.[11] This routine accepts as input the function which is in this case *beta* and the limits on integration, and returns the value of the integral. The routine that exists in "Numerical Recipes" integrates the function with a predetermined number of Gauss points which could waste computer resources or, worse yet, prove to be insufficiently accurate. The routine was, therefore, modified to work with a variable number of Gauss points that is dependent on the particular function being integrated. The fact that the integral of all probability functions density functions is unity when integrated over their entire range is the key determinant of an appropriate number of Gauss points.

The integration scheme in *analyze_chain* begins by integrating over the entire range of the beta probability distribution of the sum dimension with ten Gauss points. If the value of the integration is not sufficiently close to one, two more Gauss points are calculated, and the integration is performed with twelve points. This scheme continues until the value of the integration over the entire range falls within the specified error. Once the sufficient number of Gauss points has been determined, the function is integrated over the range of the tolerance zone input by the user. To accomplish this integration this *analyze_chain* calls *qgauss_gen* with the number of Gauss points used for the entire range and the limits of the tolerance zone. Since the range of the tolerance zone can be no larger than the actual range of the function, the number of gauss points will be large enough to provide a sufficiently accurate integrand. In other cases in which the range of the tolerance zone is less than the entire range, the full number of Gauss points calculated for the entire range are used in a smaller range; increasing the accuracy even further. *Qgauss_gen* returns the value of the integration between the limits of the tolerance zone which is the confidence level of the assembly.

The Gauss points and weights are calculated from the routine *Gauleg* taken from "Numerical Recipes". Each time the section of *analyze_chain* decides that the value of the integration over the entire range is not within the specified error, it calls *Gauleg* to calculate two more gauss points and weights for the next iteration of integration.

The user can specify how close the integration over the entire range is to unity. The value defaults to an error of 10^{-5}. This value is initialized when the module is first executed, but if it is decided that more accuracy is needed, it can be modified by the user in the system parameters menu. The changes made in the system parameters menu are only for the current session; for subsequent executions, the original default value will be valid.

3.6 Plotting Scheme

The numerical solution of the confidence level of the assembly is the final characteristic that is calculated for the distribution of the sum dimension. Rather than presenting a list of numerical parameters of the distribution and confidence level, it is much more valuable to provide the user with some form of visual results. The plotting routine presents a plot of the distribution with its range and input tolerance zone clearly labeled. The X and Y axes of the plot are automatically scaled to accommodate the X and Y ranges of the distribution and make maximum utilization of the screen. The various statistics that are relevant to the distribution, such as the expectation, variance, middle of the range, and confidence level, are listed along with the plot. The graphic output gives the user a frame of reference through which to view the location of the tolerance zone and observe any skewness and the overall shape of the distribution. By visually observing the distribution, the user can determine if a shift in the tolerance zone might yield a higher confidence level. The graphic output of the distribution allows for a quick evaluation of the assembly in the performance of a "what if" type analysis. An illustration of a sample output is shown if Figure 3.7. The example shown if Figure 3.7 only has a few points along the curve to illustrate that the curve is actually a series of straight line segments.

There are actually two distinct subroutines that comprise the plotting capability of the tolerance analysis module, *plot_dist* and *tol_plot*. The first of these routines is called by *analyze_chain* after the numerical solution to the integration of the p.d.f. has been obtained. The routine *plot_dist* determines the X and Y coordinates of the points on the probability distribution. Once all of the points on the curve have been determined, including those that bound the tolerance zone, *plot_dist* calls *tol_plot* to toggle the terminal to graphics mode and actually plot the results.

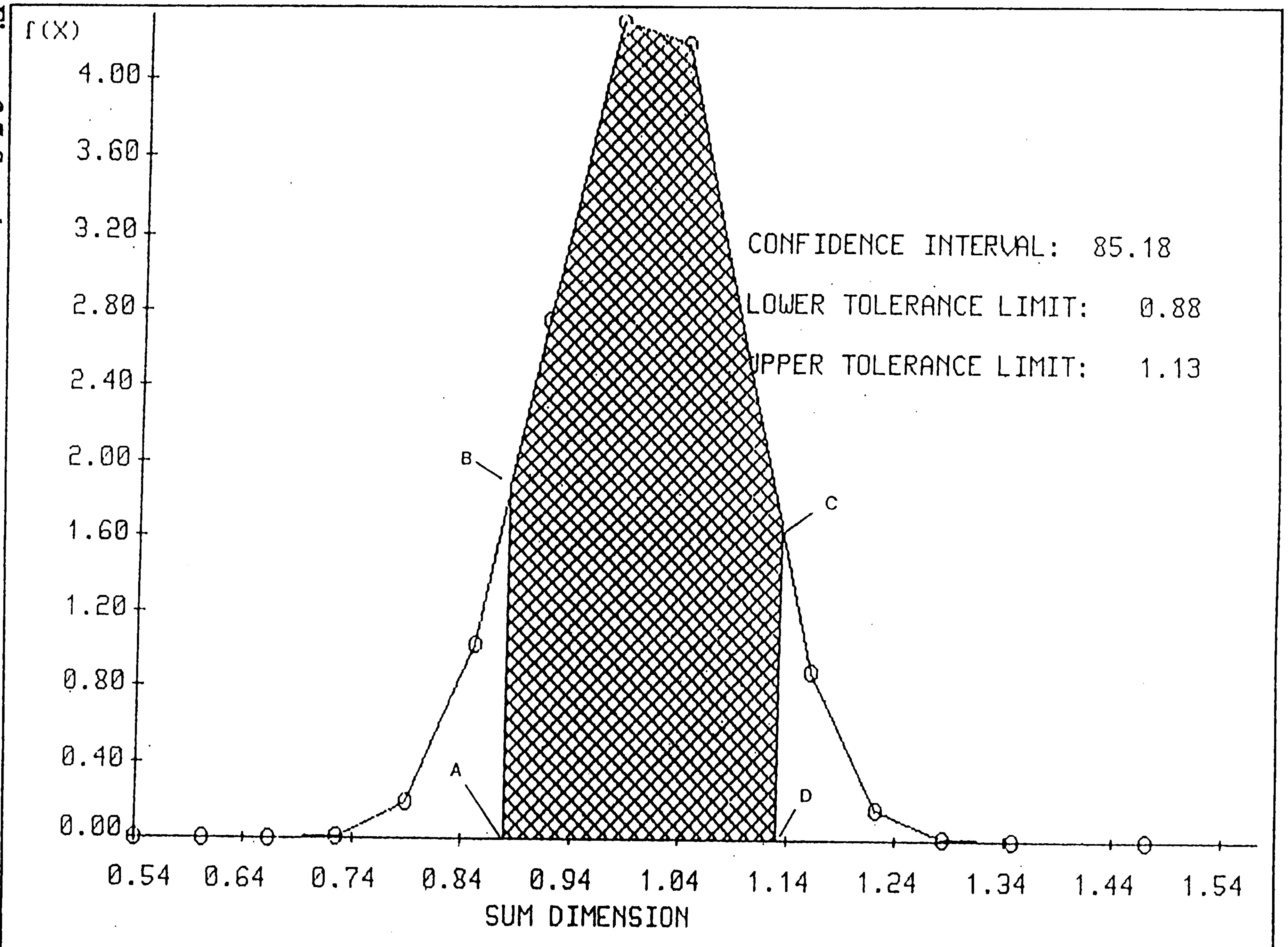

Figure 3.7 Sample output

3.6.1 Routine plot dist

The *plot_dist* routine accepts as input the limits of range of the distribution, the limits of the tolerance zone, a function that will be used to calculated the Y values of the distribution, and the calculated value for the confidence level. First, the routine divides the range of the distribution into a number of points for which to determine the Y values of the points along the distribution. The plotting routine actually draws straight lines between the points along the curve so the more points that are calculated for the curve, the smoother the curve will appear on the screen. The number of points into which the range of the distribution is broken up has a default value of 30 points set in the system parameters menu. If the user decides that the curve is not smooth enough for the assembly being viewed, the number of points can be increased from the system parameters menu. Conversely, if the user feels it is unnecessary to generate and plot the current number of points, the number of points can be reduced.

The plotting routines have been coded in such a way that they are independent of the distribution used. *Plot_dist* has the ability to accept any function which calculates the Y values of the X points within the range of the distribution. For the case here, the function passed to *plot_dist* is the beta function. *Plot_dist* calculates the size of the step between the positions of the X values of the points on the curve by dividing the magnitude of the range of the distribution by the number of specified points.

The beta function is undefined for values of X less than or equal to zero. If the sum dimension is a span, a case in which values of the range of the dimension are less than zero suggests a negative dimension which is a physical impossibility. If the sum dimension is a gap, negative values of the range infer an interference fit. For either of these cases, if an X value in the range is determined to be non-positive, it is assigned a Y value of zero. In none of the examples in which the module was tested did the sum dimension of a span have part of its range less than zero. If it had, it would suggest that for the other parts in the assembly given the ranges on their tolerances, for the negative part of the range, assembly of the parts would require a part with a negative dimension, making assembly impossible. The tolerance zone of the assembly would obviously never be placed in this region, but if it were, the analysis routines would yield what percentage of the assemblies fell into the specified area.

The first point on the distribution that the routine calculates is the Y value corresponding to the lower limit on the range of the integration. The routine tests to see if the value is positive. If it is, it calls the function to calculate the corresponding Y value. If the value is not positive, it assigns a Y value of zero. The routine then enters a loop to calculate the intermediate points on the curve. It begins with an X value of the lower limit and increments each time through the loop by the step size previously calculated. Each value of X is checked to see if it is positive, and it is either assigned a value of 0, or is sent to the function to determine its corresponding Y value. Upon completion of the loop, the Y value of the upper bound of the range is determined by the same method just described.

When the probability distribution function is displayed, the user defines tolerance zone, which is the shaded area of integration. The limits on the integration of the tolerance zone are the lower and upper bounds of the tolerance zone. The routine that will shade the area under the curve bounded by the tolerance zone limits needs a closed polygon to shade. The closed polygon consists of the lower bound of the tolerance zone on the X - axis, the point on the curve corresponding to the lower limit, all of the points along the curve up to and including the point corresponding to the upper bound of the tolerance zone, and finally the upper bound of the tolerance zone on the X - Axis. The only points that are not known are the two points on the curve corresponding to the limits of the tolerance zone. These two points, shown as points B and C in Figure 3.7, are determined by linearly interpolating on the Y values between the two points that flank the tolerance zone limits. The Y values for the tolerance zone limits are not obtained by determining the beta function value for the two points. If they were, when the polygon to be filled, i.e. the area of integration, was drawn, the vertical lines that mark the boundary of the area might not end at the curve. The curve is actually a composition of straight line segments drawn between the points that were calculated to lie on the curve. A linear interpolation between the two points surrounding either of the tolerance zone limits produces the exact intersection point with a vertical line and the curve. The four points that bound the vertical sides of the integration area are sent to *tol_plot* in an array.

3.6.2 Routine tol_plot

The input to the routine *tol_plot* is the points that make up the curve,the points that create the boundary of the area of integration, and the parameters of the distribution that are to be placed on the screen with the plot. The output of the routine a Cartesian coordinate system with the axis labeled and a color plot of the distribution with a shaded area of integration.

The plotting routine used is a modified version of a terminal dependent plotting routine, *plot2d*, created by Audrey Griscavage [13] for a DEC VS11 type graphics terminal. The routine is a system of Graph3d.lu callable graphics subroutines. *Plot2d* was upgraded to be terminal independent and modified to operate in conjunction with the rest of the tolerance analysis module and renamed *tol_plot*. The basics remain the same, but the routine has been changed to interface with the system parameters menu, create a polygon bounding the area of integration, employ the Tektronix 41xx series terminal's capability to fill the area of integration, and display the parameters of the distribution.

Operation of tol plot

The routine begins by initializing the terminal to operate in graphics mode and to initialize the graph3d.lu routines so that they may be called from the module. The size of the graphics window is set to simulate the resolution of a VS11 graphics terminal for which the routine was originally written. By initializing the size of the window, any VS11 terminal dependent routines will function properly thereby making the plotting routine device independent.

Since the routine will always use the same size window, it must determine the smallest and largest values of both the X and Y values of the function to scale the plot to fit in the window. *Tol_plot* accomplishes this determination by calling two routines named *bigfind* and *smallfind* with the X and Y values of the curve as input. The range of the X and Y values must also be known to label the axes. *Tol_plot* calls on the routine *labcalc* to determine the values of the labels of the tic marks on the axes. Some of the features of the plotting routine are controlled by a flag and can be either turned on or off. The axes and tic marks on the axes have this feature. If the axis/tic mark flag is set to 1, then the axes will be displayed with tic marks and the tic marks labeled. If the axis/tic mark flag is set to zero, the axes and consequently the tic marks and their labels will not be displayed. This feature is controlled in the system parameters menu and defaults to featuring axes with labeled tic marks.

The routine has the capability to change the symbol displayed at the calculated points on the curve; seven possible choices can be chosen from the system parameters menu. The default value is a circle, but the symbol can be chosen by inputting a value from the list shown below in Table 3.4 into the point symbol selection in system parameters.

Table 3.4 Point symbols

number	character
0	no symbol
1	+
2	square
3	triangle
4	*
5	diamond
6	circle

The routine plots the points before the curve is actually connected through them. The routine draws the curve by calling subroutine *connect*. This routine connects straight lines through the coordinates that are passed to it in the order that they exist in the arrays that were passed. This process will not cause a problem because the arrays that hold the coordinates of the points along the curve were filled by starting at the lower limit of the range of the distribution and continued to the upper bound on the range.

If the routine were to stop here, the curve of the probability distribution function characteristic of this assembly with its given parameters would be displayed. The next step places the tolerance zone within the range, and shades the area. Initially the points that make up the vertical boundaries to the window of the screen are scaled (points A, B, C, D, in Figure 3.7) so that they are in the proper location on the curve. A vertical line is drawn between points A and B, and between points C and D in Figure 3.7. The area to be filled consists of the area bounded by X - axis on the bottom, the vertical lines on the sides, and the portion of the curve between the two vertical lines on the top. The routine determines which points along the curve lie between the two vertical lines. These points along with A, B, C, and, D in Figure 3.7 are sent to routine *poly2dfill*. This routine accepts the points as a closed polygon and fills the polygon with a specified pattern. The color and pattern of the fill have been preselected and cannot be changed by the user.

Once the curve is displayed and the area of integration is filled, the routine has completed its task. It returns the mode of the terminal to ANSI mode and returns control of program execution to *plot_dist* which in turn returns control to *analyze_chain*. The analysis process is now complete for one set of parameters.

3.7 Tolerance Synthesis (distribution)

Tolerance synthesis is the allocation of tolerances to the component parts of an assembly based on some assignment criteria such that the final assembly of parts functions in conjunction with the designated design specifications. Tolerance synthesis is performed on a tolerance chain that has one or more links with unassigned tolerances. At design time in the geometric modeling environment, the tolerances may not be specified for a dimension because either the tolerance is completely unknown, or because the link is flexible with respect to the tolerance. A link that possesses an untoleranced dimension is flagged in the tolerance chain data file in such a way that the tolerance analysis and synthesis module is aware that the particular link does not have an assigned tolerance. All of the links known to have unassigned are candidates to have their tolerances assigned so that the functional criteria of the assembly are met.

The method of assigning tolerances used in *TASM* is the complexity factor method discussed in Chapter 2. This method lends itself well to a wide variety of applications. The actual factor can be assigned any meaning because the system is one that is based on the relative weights of the factors. In other words, the factors can be based on cost, degree of difficulty of manufacturing, or any other application specific factor that would be affected by the tolerance assigned to a part. In most cases, somewhere down the line, the underlying factor is cost. The factors can be assigned values that are valid for a specific user, making the method very transportable. There is not one predetermined set of complexity factors that must be utilized for every user's application. As long as the user assigns realistic factors for each component, one set of factors is as good as another.

Bjorke's method of distributing the tolerances described in his text is expanded here to include chain links that are gap type links. A system has been devised that lets the user input a complexity factor for both components of the gap and divide the tolerance assigned to the link among its constituent components based on the individual parts complexity factor.

A tolerance synthesis is initiated from the main menu upon execution of the module. The approach to performing a tolerance synthesis and the operation of the module in accomplishing the tolerance synthesis are discussed in this section.

A tolerance synthesis session is begun by selecting menu choice "TD", for tolerance distribution, from the main menu. This choice directs the flow of program execution to the tolerance synthesis section of the main calling program. The next menu to be displayed is the tolerance distribution menu, shown below in Figure 3.8. Similar to a tolerance analysis, a tolerance chain must first be read into the module. This input can be done by selecting the menu entry "RE" for retrieve chain. The menu choice invokes the same series of subroutine calls as is carried out in performing a tolerance analysis. The tolerance chain data file is read in via a call to the *get_chain* routine. Once the tolerance chain data file has been successfully read in, the flag indicating that the chain has been read in is appropriately set as described in Section 3.4, and the module will again display the tolerance distribution menu.

TOLERANCE DISTRIBUTION

RE - Retrieve Chain
LS - Link Statistics
IE - Initial Estimate
TU - Tolerance Update

Figure 3.8 Tolerance Distribution Menu

Analogous to the process of performing a tolerance analysis, the next step in performing a tolerance synthesis is to read in the chain link's means and variances. Choosing the menu selection "LS" for link statistics initiates the same series of subroutine calls as those in the tolerance analysis. *Crlmv* is called as for a tolerance analysis, and the remaining steps in retrieving the link statistics follow exactly as in a tolerance analysis. Also, similar to the tolerance analysis procedure, the flag that maintains the input status of the link statistics must be updated upon successful completion of their being read. The flags are present in the tolerance synthesis portion for the same reasons they were incorporated in the tolerance analysis segment of the module. Certain analysis routines need specific data on which to operate. It is easier to make certain that the user has made this necessary data available than to attempt to anticipate and ameliorate all of the consequences resulting from a lack of data. Upon reading the link statistics either from the keyboard or from an existing ".LMV" file, the tolerance distribution menu is again displayed to the user.

After the tolerance chain data file and the link statistical data have been made available to the module, the similarities between the tolerance analysis approach and the tolerance synthesis approach end. The next step in determining the tolerances of the unassigned dimensions begins with the selection of "IE" ,for initial estimate, from the tolerance distribution menu. This selection transfers control of the main calling program *TASM* to the line that calls routine *inest*, which stands for "initial estimate". The routine prompts the user for additional input and performs the necessary calculations. Output is presented in tabulated form of the tolerances assigned to the originally unassigned links.

3.7.1 Routine inest

Subroutine *inest* walks the user through the steps involved for a tolerance distribution using the complexity factor method described in Chapter 2. The routine prompts the user for input on the desired tolerance zone of the functional dimension. With this information the routine knows how much of a tolerance it has to spread among the unassigned tolerances in conjunction with the tolerances that are pre-assigned. The routine next requires the normalized tolerance for the confidence level desired. This value is retrieved from a table of normalized beta parameters based on the normalized range, skewness, and confidence level. Such a table of normalized beta distribution parameters is listed in Appendix II. Since the tolerances have not yet been assigned to all of the parts, the actual range of the sum dimension is still unknown. An estimate of the normalized range is calculated by *inest* and presented to the user as a guide to determine a value for the normalized tolerance in the table. A second parameter "FW" in the table must also be known to determine the value of normalized tolerance for the sum dimension. "FW" is a normalized skewness parameter, and Bjorke suggests that a value of zero be used unless there is a known valid reason to use another. With the estimated value of the range, the specified confidence level, and given an "FW" of 0, the normalized tolerance can be extracted from the tables and input to the module.

Routine *inest* has the common block /chain_par/ resident in its code giving it access the chain link parameters. Each link tolerance assignment status is tested and the number of links with unassigned tolerances are counted. The individual effects of each of the links with predetermined tolerances must be determined so that Equation (2.56) can be solved for RHS. *Inest* loops over the links in the chain and calls the same link routines used in the tolerance analysis for each of the links with predetermined tolerances.

After the link parameters have been calculated by the link routines, *inest* prompts the user to input the complexity factors for the links with unassigned

tolerances. If the link is a gap type link, the routine prompts the user to input a complexity factor for each of the components of the gap. Immediately the routine calculates the weights of the links from the complexity factors. For a link that is a gap and has two complexity factors, the highest complexity factor must be realized; therefore *inest* uses the higher complexity factor of the two components comprising up the gap.

Given the weights of the links, the routine solves for the tolerances on the links using the equation (2.58). Although the tolerances have been spread across the links, the tolerance assignment is not yet complete. Knowing that a certain tolerance is to be assigned to a link if the link is a gap would be incomplete information. The routine checks to see if any of the links are gap type links that need their tolerances distributed to the component level. If the routine detects gap type links, it uses the two complexity factors read in for the link in question, recalculates the weights based on the two complexity factors for the link, and spreads the tolerances assigned to the link its two component parts. The tolerances that are assigned to the two component parts must be located on the parts.

Inest prompts the user to input an allowance for the gap. An allowance in the minimum allowable clearance that must exist between the parts, is generally available because of the functional requirements of the gap. Assignment of the tolerance locations of the tolerance zones on the parts is then based on the ISO system of fits with the hole basis classification H7. The classification of the shaft is dependent on the allowance input by the user. By the user stating the hole basis, one of the limits of the tolerance zone is known when placing the zone on the part. At this point the tolerances calculated for the gap type links are placed into the appropriate link parameter arrays of the common block /chain_par/.

The remainder of the procedure discussed in section 2.6 is now undertaken. *Inest* prompts the user for the desired expectation of the sum dimension and presents a list of assignable MX_i's. Only links whose MX are not equal to zero can be assigned an MX_i. The list presented consists of link numbers and the dimension's basic size. All the links but one selected link must have their middle of the range decided by the user. The routine asks the user to input which link will have its MX solved for. A series of checks are taken to insure that the user has chosen a selectable link. It first checks to see if it was a proper type, i.e. a span. It then checks to see if the span chosen had an unassigned tolerance. It finally proceeds to prompt for input for the rest of the available links mid-ranges. Tolerances that were calculate for the links can now be placed in a location on the span type links (Gaps have been already completely assigned). The parameters of these links can now also be updated in the common block /chain_par/.

The last step for *inest* is to calculate the links with previously unassigned tolerances effect on the sum dimension by calling the link routines and to determine the mid-range MX for the chosen link. After the link parameters have been calculated for the previously unassigned links and the common block /chain_par/ has been updated, MX for the chosen link is calculated, such that equation 2.57 is satisfied. Once MX is calculated, the parameters for this last link can be calculated and the common block /chain_par/ can be completed for the last link. The contents of the would be updated data base are then displayed to the user.

3.7.2 Analyzing the determined parameters

After all of the links have had their tolerances assigned, the now complete data can be analyzed in the analysis section of the module. All of the data are available for the analysis to run, and the parameters of the sum dimension can be displayed graphically.

3.7.3 Saving Assigned Tolerances

When the user is satisfied with the tolerances he has determined with the module, he can save them back to the original tolerance chain data file. This file can then be read back to the geometric modeling program to update its data base with the tolerances assigned by *TASM*. From the tolerance distribution menu, the user selects the menu entry "UD" for "update data base". Selecting "UD", the user invokes subroutine *update_dbase*. *Update_dbase* dumps the parameters in the arrays "type_params" and "lparams" back to the tolerance chain data file. The "lparams" array now contains values for the previously unassigned tolerances. This information is now available to the designer in the geometric modeling environment.

3.8 Systems Parameters

The system parameters menu, mentioned several times throughout this paper, has been included as a tool by which to change a selected number of parameters of the module. Certain parameters may be inadequate for the particular analysis being performed, or the user may prefer a variation on a default specification. Two subroutines formulate the structure of the system parameters capabilities. They are called subroutine *defaults* and *system_parameters*.

Subroutine *defaults* is read upon execution of the module. This routine reads in the default values from an file named "defaults.dat" that must be present in the main directory for the module to operate properly. If the file is not available to be read, upon module execution an error message is displayed to the user explaining the problem. When the file is successfully opened, the subroutine defaults begins to read in the default parameters. The opening of the defaults.dat file and the reading of the default parameters are done without any interaction with the user. The format of the file is set such that it can be read without error. If however the format is changed in any way that disrupts proper reading of the file, an error message is displayed to the user. It is critical to the execution of the module that the defaults.dat data file is opened and read correctly.

Changes can be made to the default values for any current session by selecting the "SP" menu choice for "system parameters" from the main menu. When this selection is made, the user is presented with the system parameters menu, shown below in Figure 3.9. Changes made to the parameters are changed only in the computers memory and not the defaults.dat file. Therefore any alterations made here will only be valid for the current session. When the user exits, the module any changes will be lost, and the original defaults.dat file will be read upon executing the module. Permanent changes to the defaults.dat file can be made by editing the file so that, when it is read, the desired parameter values are made available to the module.

SYSTEM PARAMETERS

HB - Hole basis
TI - Tolerance for integration
NP - Number of points on curve
PS - Point symbol
XI - X axis increment
YI - Y axis increment
AC - Axis color
LC - Labels color
LA - Labels (on or off)
TM - Tic marks (on or off)
TL - Tic labels (on or off)
CL - Connecting line (on or off)
AS - Same scale both axes

Figure 3.9 System Parameters Menu

4.0 Examples

Chapter 4 describes how *TASM* may be used to perform both tolerance analysis and tolerance synthesis on an assembly. Presently the module receives its needed input as a tolerance chain data file. Tolerance chain data files that are used for either an analysis or synthesis must be located in the directory from which *TASM* is executed. In the future, the data could be made available through a direct link with the geometric modelling environment.

This chapter serves two purposes. It illustrates the capabilities of the module and demonstrates how to perform a tolerance analysis and tolerance synthesis using *TASM*. The step-by-step procedure and generated output are presented. Section 4.1 will consist of tolerance analysis examples while section 4.2 presents an in depth tolerance synthesis.

4.1 Tolerance analysis

4.1.1 Compression chamber

The first example features the analysis of the compression chamber in the compressor shown in Figure 4.1. The geometry for this example is taken from Bjorke[7] and has had the tolerance chain automatically generated by the work done by Wang[8]. All of the links shown in Figure 4.1 have an effect on the height of the compression chamber which in this case is the sum dimension X_Σ. The fundamental equation for the sum dimension being considered is:

$$X_\Sigma = -X_1-X_2-X_3-X_4+X_5+X_6-X_7+X_8-X_9+X_{10}+X_{11}+X_{12}+X_{13}+X_{14}+X_{15}+X_{16}+X_{17}$$

Table 4.1 contains a list of the link descriptions and types that make up the tolerance chain for the sum dimension being considered in this example. A listing of the tolerance chain data file that contains the all of the tolerance chain data including dimensions is listed in Table 4.2. The records in the file are highlighted in section 3.2.1.

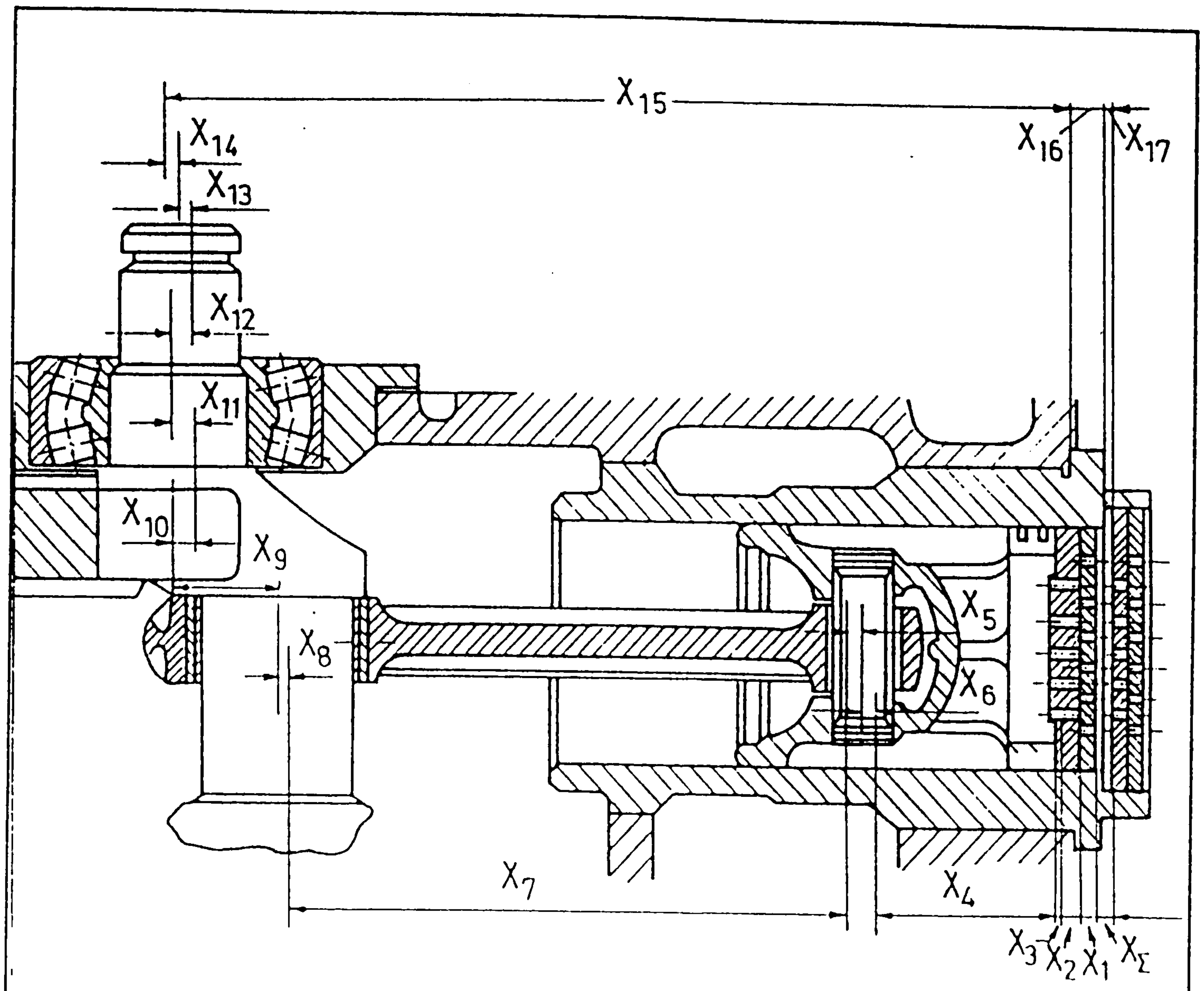

Figure 4.1 Compression chamber (reproduced from Bjorke).

A tolerance analysis begins with the selection of the tolerance control entry from the main menu. When a tolerance control is selected, *TASM* displays the tolerance control menu shown in Figure 4.2.

TOLERANCE CONTROL

RE - Retrieve Chain
LS - Link Statistics
AN - Analyze Chain
SA - Save Results

Figure 4.2 Tolerance control menu

Table 4.1 Links in the compression chamber tolerance chain

link	type	description
X_1	line vector span	thickness of valve sleeve
X_2	line vector span	thickness of valve port plate
X_3	line vector span	thickness of gasket
X_4	line vector span	distance from piston head to wrist pin bore (on piston).
X_5	interference	gap between the wrist pin bore (on piston) and wrist pin.
X_6	clearance	gap between wrist pin and wrist pin bearing (on the connecting rod)
X_7	line vector span	center distance between wrist pin bearing and connecting rod bearing.
X_8	clearance	gap between connecting rod bearing and crank.
X_9	line vector span	throw of the crank.
X_{10}	transition	gap between crank shaft and inner ring of crank shaft bearing
X_{11}	eccentricity	eccentricity of the crank shaft bearing
X_{12}	transition	gap between outer ring of crank shaft bearing and bearing mount
X_{13}	eccentricty	eccentricity of the bearing mount
X_{14}	bivariate rect.	gap between bearing mount and distributed cylinder block
X_{15}	line vector span	distance between bearing mount bore in cylinder block and cylinder block head
X_{16}	line vector span	thickness of cylinder sleeve flange
X_{17}	line vector span	thickness of spacer

Table 4.2 Listing of tolerance chain data file for compressor.

```
-1
1
1,0,0,0,1,1
-1.00000,11.00000,0.00000,-0.05000
1,0,1,2,3,1
-1.00000,9.500000,0.00000,-0.03000
1,0,2,4,5,1
-1.00000,0.500000,0.00000,-0.10000
1,0,3,7,8,1
-1.00000,133.000000,0.00000,-0.10000
5,0,4,11,9,1
1.00000,43.000000,43.00000,0.023000,0.01600,0.016000,0.00900
1,0,5,12,13,1
-1.00000,380.00000,0.05000,-0.050000
5,0,6,15,16,1
1.00000,100.000000,100.00000,0.035000,0.00000,-0.04000,-0.07500
1,0,7,16,17,1
-1.00000,65.00000,0.05000,-0.050000
6,0,8,19,17,1
1.00000,90.000000,90.00000,0.035000,0.00000,0.02500,0.00300
3,0,9,19,18,1
1.00000,0.00000,0.012000,0.000000
6,0,10,21,18,1
1.00000,190.000000,190.00000,0.046000,0.00000,0.00200,-0.03800
3,0,11,21,20,1
1.00000,0.00000,0.025000,0.000000
15,0,12,22,20,1
1.00000,255.000000,255.00000,0.046000,0.00000,0.00000,-0.03000
1,0,13,22,23,1
1.00000,580.00000,0.00000,-0.140000
1,0,14,25,26,1
1.00000,19.00000,0.00000,-0.100000
1,0,15,29,30,1
1.00000,0.80000,0.10000,0.000000
-1,0,0,0,0,0
```

As mentioned in Chapter 3, a tolerance chain data file must be input to the module before any analysis can take place. For this example, the name of the file containing the tolerance chain to be read in is comp.chn. At present, the module defaults to this tolerance chain data file; with some slight modifications to the module, the default could be modified to be the last chain worked on. This revision would allow saving a file at the end of a session that would be read at the beginning of the next session, just as the defaults.dat file that is read to give the module the system parameters. In this case, however, since the default tolerance chain data file is comp.chn, the user only has to hit return to accept it as a default.

Next, the user must supply the link means and variances of the assembly. To initiate this process, the user selects "LS" from the tolerance control menu. Two methods, from the keyboard or from a pre-existing file, exist to input the link means and variances. This example shows how the link means and variances are input from the keyboard. Not all of the links but only the first few will be addressed, though the procedure is consistent for all.

Since, for this example, we are assuming that the link means and variances do not exist in an ".lmv" file, we answer no to the question "read link means and variances from a file [n] ?". This can be done by accepting the default of no by hitting return at the question. Link by link, the user goes through the chain and ask the user for the each links mean and variance. For the case of a gap, the module prompts the user for the mean and variance of each component of the gap. The link means and variances are selected from the list provided in Appendix I. Part tolerances are known, and the process dependent unit means and variances are selected from the table provided. All of the links' means and variances used in this analysis are listed in Table 4.3.

The module determines that the first link is a line vector span and relays this information to the user and prompts for the unit mean and variance for this

Table 4.3 **Compression chamber link means and variances.**

link	mean	variance
X_1	0.5	0.045
X_2	0.5	0.045
X_3	0.5	0.028
X_4	0.5	0.040
X_{5b}	0.4	0.028
X_{5a}	0.5	0.028
X_6	0.55	0.047
X_{7b}	0.5	0.047
X_{7a}	0.55	0.047
X_8	0.55	0.047
X_{9b}	0.5	0.047
X_{9a}	0.53	0.035
X_{10}	0.00	0.166
X_{11b}	0.5	0.047
X_{11a}	0.53	0.035
X_{12}	0.00	0.166
X_{13b}	0.5	0.047
X_{13a}	0.53	0.035
X_{14}	0.5	0.040
X_{15}	0.55	0.035
X_{16}	0.60	0.045

first link. Links 1 through 4 are all line vector spans and are handled in the same manner. Link 5 however is a clearance and therefore has two components. The module proceeds to prompt the user for both of the components means and variances. Parameters for all of the gap type links are input similarly as is the clearance. When all of the means and variances listed in Table 4.3 are input the module asks the user if he would like to save these values to a file. It is recommended that this is done in case the assembly being considered is to be analyzed again. A file is created having the name of the tolerance chain data file with the file extension ".lmv". The next time a tolerance analysis is going to be run for these statistical parameters, the module can retrieve this file. By answering yes to the question "read link means and variances from a file [n] ?", the user directs the module to search the current directory for a file with the same name as the tolerance chain being analyzed with an ".lmv" file extension.

After the statistical parameters of the links have been input to the module, the tolerance control menu is again displayed. At this point "AN" is selected to display the ANALYZE CHAIN menu shown below in Figure 4.3.

```
ANALYZE CHAIN

CD - Calculate distribution
TZ - Tolerance zone
```

Figure 4.3 Analyze chain menu.

Selecting "CD" instructs the module to determine the distribution of the sum dimension from the individual link parameters. "CD" must always be chosen before "TC". Once "CD" has been chosen and the distribution has been calculated, the chain can be analyzed for any number of tolerance zones. The "TC" choice allows the user to move the tolerance zone around within the distribution without re-calculating the distribution of the sum dimension each time.

When "CD" is selected, the module first calculates the individual link parameters effect on the sum dimension. If the module encounters a transition type link in the chain, it stops and prompts the user for additional information. This additional information is the expectation ζ_{ex} and variance ζ_{varx} of the transitions distribution, determined from Figures 2.16 and 2.17, as was explained in Section 2.4.2.2. The value of ζ is calculated for the link is question and displayed to the user such that the values of ζ_{ex} and ζ_{varx} can be determined from the plots. Either curve on the plots can be used depending on which curve the user feels better describes the transition.

In this compression chamber example, two of the links are transitions. Link number 9 is a transition that has ζ = 0.329. From the normal curves in the diagrams in Figure 2.16 and Figure 2.17:

$$\zeta_{ex} = 0.61 \qquad\qquad \zeta_{varx} = 0.46$$

Link 11 is also a transition with ζ is calculated to be 3.19. The value of expectation and expectation are extrapolated from the diagrams to be:

$$\zeta_{ex} = 3.19 \qquad\qquad \zeta_{varx} = 1.0$$

After both of the transition links have been calculated the module continues uninterrupted to solve for the remainder of the individual link parameters and determine the distribution of the sum dimension. When the module has completed solved for the parameters of the beta distribution of the sum dimension, the user is prompted for the desired tolerance zone of the sum dimension. Since the tolerance zone is not static, i.e., it can be located anywhere within the bounds of the range of the sum dimension, the user must input the tolerance in the form of the tolerance zone and its midpoint. To enable the user to make a realistic choice for the tolerance zone, the parameters of the sum dimension are presented. For the compression chamber example, the message displayed for the parameters of the sum dimension presented are shown in Figure 4.4.

Middle of range = 0.940
Range = 1.070
Lower limit of range = 0.405
Upper limit of range = 1.475
Expectation = 0.937
Variance = 0.004

Figure 4.4 Parameters of sum dimension.

At the prompt for the tolerance zone in this example a value of 0.353 was entered. The midpoint of the tolerance zone was chosen to be the expectation of the distribution, 0.937.

TASM now has all of the information it needs to calculate the confidence level for the assembly based on the input tolerance zone. The integration routines are called, and the results are displayed on the screen. A confidence level of 99.73 was calculated for the tolerance data input. Figure 4.5 illustrates the graphical output generated on the terminal by *TASM*. In this plot it is difficult to see the limits on the tolerance zone at 0.76 and 1.11. Hard copy graphical output is generated by dumping the screen to an available plotter. A file of the input and calculated parameters can be saved by selecting "SA" from the tolerance distribution menu.

The user may want to see what effect tightening the tolerance of the sum dimension will have on the confidence level of the assembly.. A "what if" type of analysis can be performed by recalculating the confidence level for a different tolerance on the sum dimension. If the tolerance is tightened, the confidence level decreases. Since the distribution has already been calculated, the module has only to integrate the distribution over a new tolerance zone. A new tolerance zone can be specified by selecting "TZ" from the analyze chain menu. When "TZ" is chosen, the user is again prompted to input a tolerance zone and the middle pointof the tolerance zone. For this example, a tolerance zone of .2 was chosen with a middle point again at the expectation of the distribution. This time the confidence is level is at 90.44%. Figure 4.6 shows the limits on the tolerance zone for this example are more apparent.

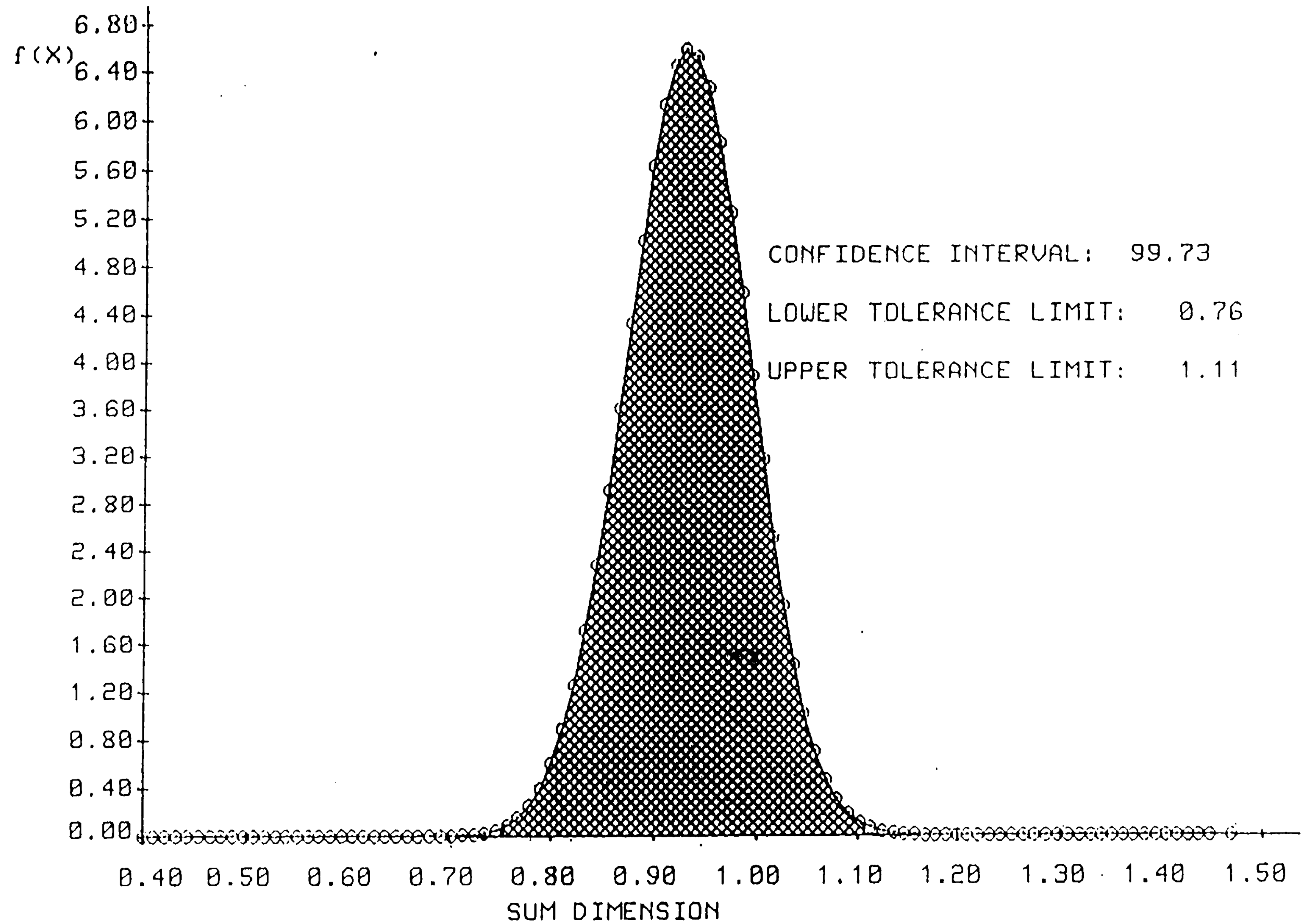

Figure 4.5 Plot of distribution of compression chamber sum dimension with tolerance zone of .353 and middle point .937.

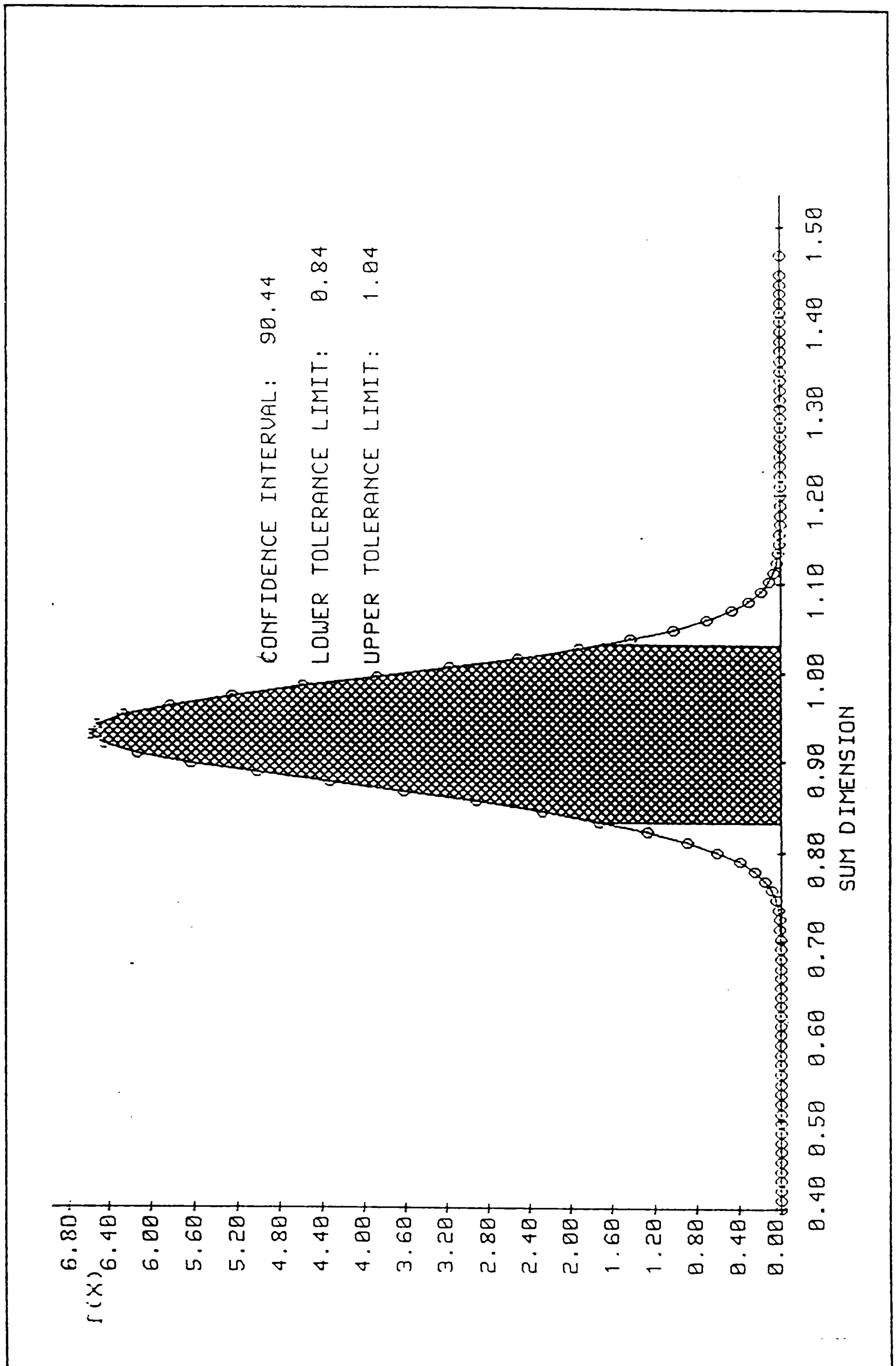

Figure 4.6 Plot of distribution of compression chamber sum dimension with tolerance zone of 0.2 and middle point at 0.937.

It may also be necessary to see what would happen if the same tolerance were applied at a different location within the range of the sum dimension. For th example a tolerance zone of 0.2 was used again but instead of being centered at the expectation, it has a middle point value of 1.0. This tolerance only has a confidence of 72.46 %. It can be seen that for the best results the middle point of the tolerance zone should be as close to the expectation of the distribution as possible. Figure 4.7 shows the distribution with the tolerance zone of 0.2 moved to a middle point value of 1.0.

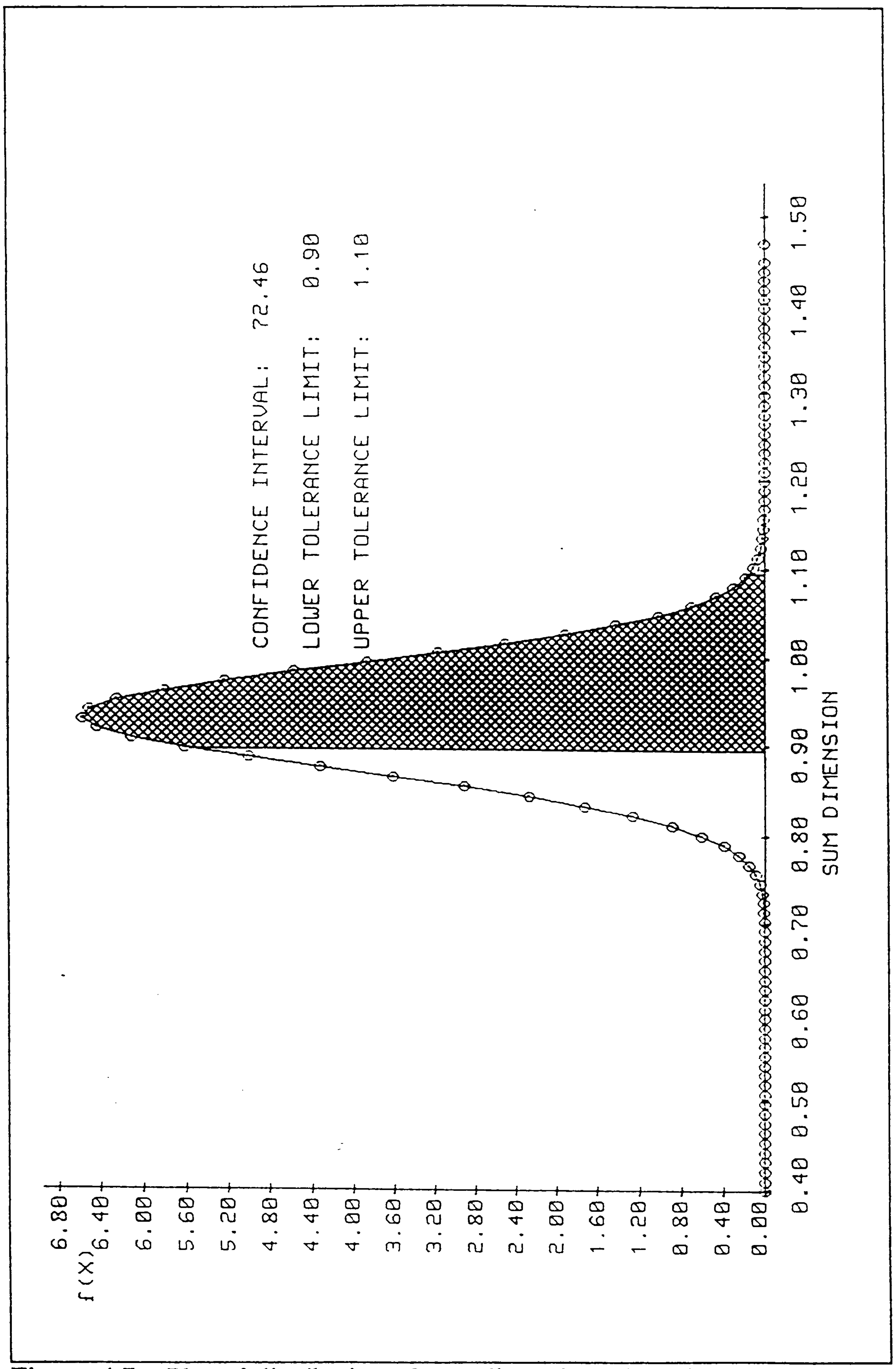

Figure 4.7 Plot of distribution of sum dimension with tolerance zone of .2 centered at 1.0.

4.1.2 Printed circuit board model

In this example an assembly process is actually used to model a manufacturing process. Figure 4.8 shows a cross section of a modeled segment of a printed circuit board. The dimensions in the geometry are unitless values that do not represent any particular circuit board's dimensions. They exist only to illustrate the technique and put some values on the dimensions.

In this model, there are four horizontal plates denoting the layers of a printed circuit board. The holes in the plates represent the pads that a chip would be inserted into. Before any chips can be placed on a board, the board must be drilled. The process involves drilling holes through the pads that lie on the layers of the board. In order for the board to function properly, the holes drilled through the board must hit all of the pads without "breaking out". Breakout is a condition where the drill only hits a part of a pad. This phenomenon results from the shifting of the layers which causes the centers of the pads along a vertical line to be eccentric to one another. Even if the hole is drilled directly through the center of the top pad, a layer below may shift enough to allow the drill to miss the pad entirely or, just as bad, only partially hit the pad creating a breakout condition.

In the model devised here by Ozsoy, Wang, and the author, the pad drilling process is modeled after a pin insertion assembly process through a series of holes. In order for a pin to be inserted successfully through a series of horizontal plates, the holes in those plates must be suitably concentric. If any one of the holes along a vertical line through the plates is not properly in line, the pin will fail to go all the way through the plates. For the printed circuit board model, a pin is analogous to the drill bit, and the pads on a layer are analogous to a hole in a plate. A clearance condition exists at each plate so that a pin can penetrate all the way through the plates. This clearance condition is similar to the breakout condition in drilling the circuit boards. The drill must pass through all of the pads

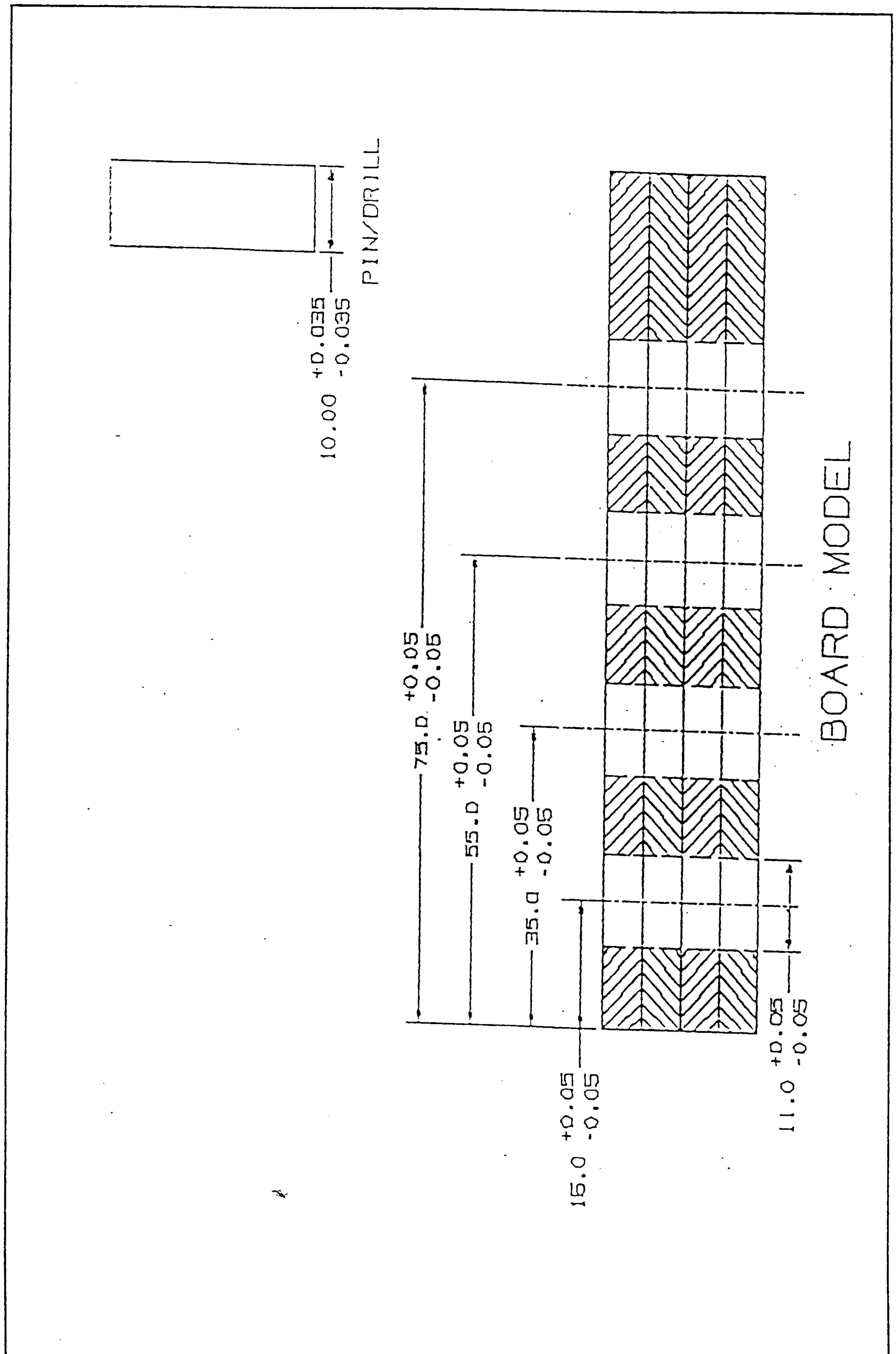

Figure 4.8 Printed circuit board model

in a vertical line and be within each of the pads in a layer the same way a pin must be within each of the holes in a plate.

One major difference exits between this example and others discussed throughout the thesis. This example is under the constraint that it has multiple tolerance chains that must be simultaneously satisfied. For the example shown in Figure 4.8 the four holes through the four plated require twelve tolerance chains. An example of one of these tolerance chains are shown in Figure 4.9. There are three tolerance chains for each hole i.e., a chain for each layer in reference by the to the top layer with the sum dimension as shown. Figure 4.9 shows the mechanical assembly with the plates at each of the extremes. The three chains per hole and a total of four holes provides a total of twelve tolerance chains.

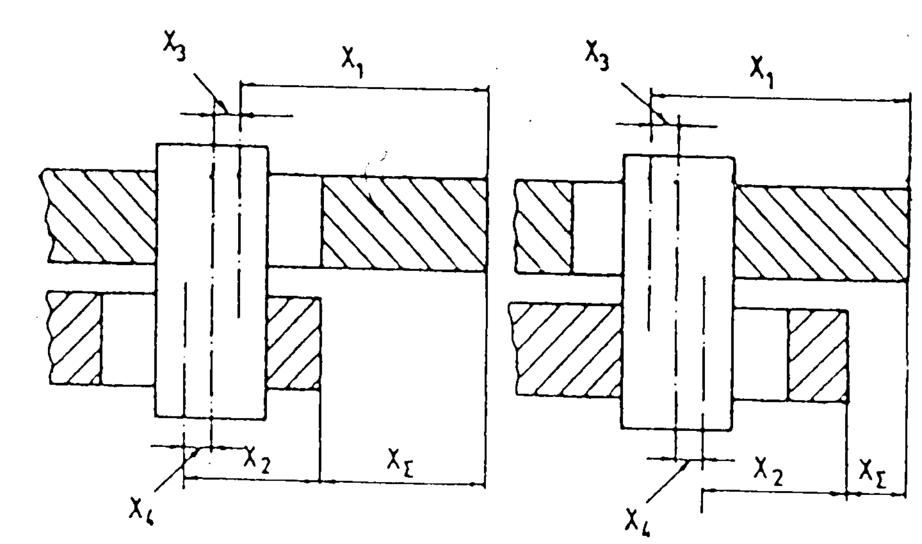

Figure 4.9 Single tolerance chain in printed circuit board model (reproduced from Bjorke).

The link statistics for each of the tolerance chains can be assumed to be the same because the same manufacturing process is used for each of the similar components for each of the chains. In this example it is therefore necessary only to perform the calculations for one of the tolerance chains and determine the confidence level for one tolerance chain. Since each of the chains does not effect the others, the probabilities of each of the chains are mutually independent, and the total confidence of the assembly is the product of all of the individual confidence levels. Since the statistics for each of the chains are the same, the final confidence level of the entire assembly is the confidence of one of the chains raised to the number of chains n, which in this example is twelve.

$$\text{assembly confidence} = (\text{chain confidence})^{n=12} \quad \text{(eq. 4.1)}$$

The analysis of one of the tolerance chains in the model is begun by reading the tolerance chain data file pcb.chn. The link means and variances used for the chain are listed in a file named pcb.chn. Both the dimensions of the components in the chain and their means and variances are listed in Table 4.4.

Table 4.4 Nominal dimensions and statistical parameters of the components in the printed circuit board example.

Variable	Description of variable	nominal dimension	mean	variance
X_1	distance to hole	15	0.5	0.028
X_2	hole diameter	11	0.55	0.028
X_3	pin diameter	10	0.5	0.04

The analysis is performed by selecting "AN" from the tolerance distribution menu. The statistics calculated for the sum dimension that are displayed to the viewer are shown listed below in Figure 4.10 as they appear on the terminal.

Middle of Range	1.000
Range	0.930
lower limit of range	0.535
Upper limit of range	1.465
Expectation	0.997
Variance	0.007

Figure 4.10 Sum dimension statistics presented to the user prior to a tolerance zone selection.

With the above statistical information on the sum dimension, a tolerance zone of 0.48, centered at 1.0 was input to the module at the appropriate prompts. The confidence level calculated for the individual chain was 99.71%. The graphical results displayed by the module are reproduced in Figure 4.11.

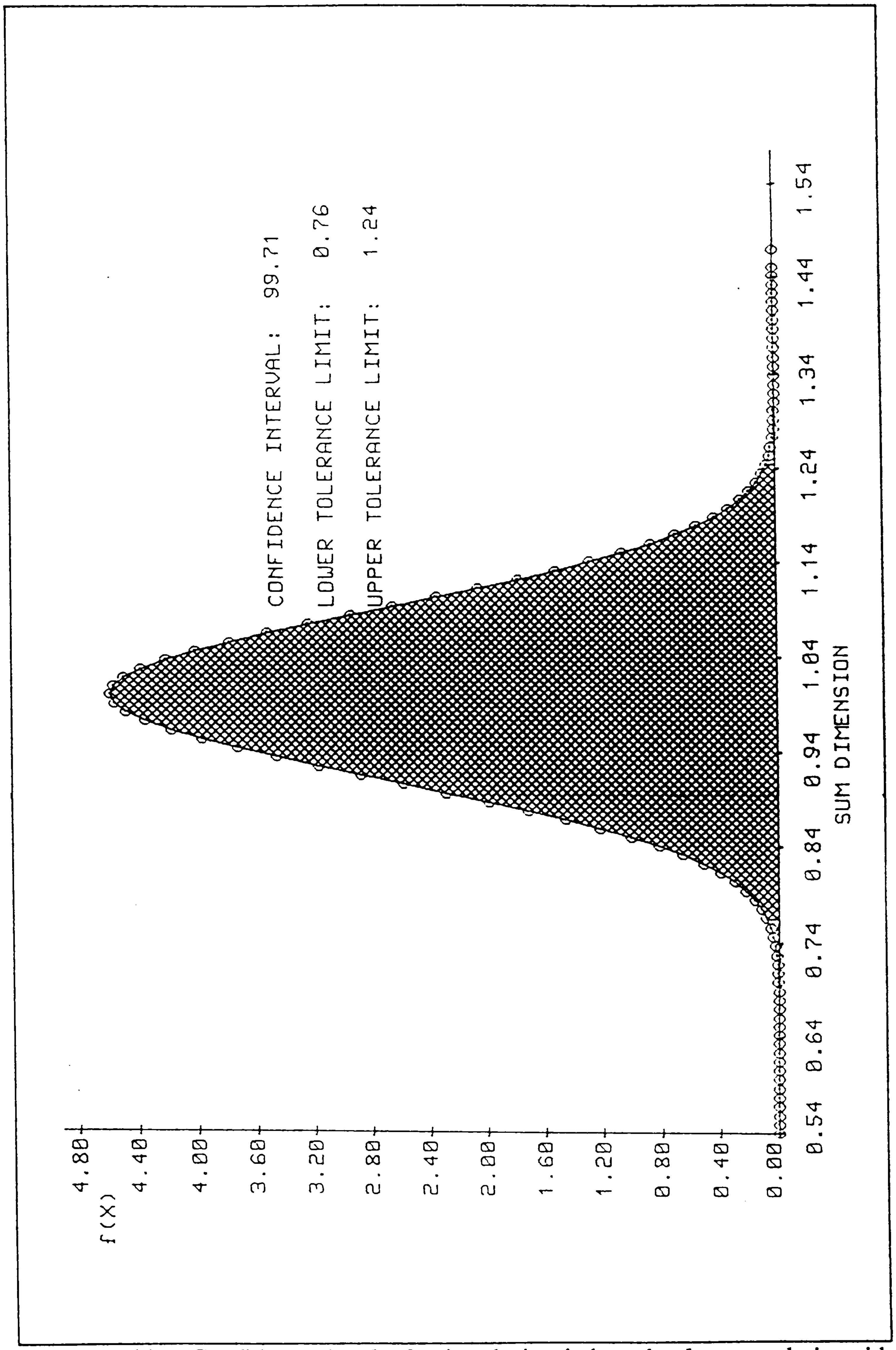

Figure 4.11 Confidence level of printed circuit board tolerance chain with a tolerance zone of 0.48 centered at 1.0.

Another confidence level was calculated and plotted by selecting "TZ" from the analyze chain menu and inputting a new tolerance zone. This time the tolerance zone was given a value of 0.3 and was again centered at 1.0. The confidence level yielded by this tolerance zone was 92.34% A plot of this out put is illustrated in Figure 4.12. Both of these distributions are for one of the twelve distributions. By placing the constraint that all of the chains will need the same tolerance specifications, the confidence of the entire assembly is calculated by equation 4.1. The results of the confidence levels of the assembly for the two cases run are shown in Table 4.5 along with the a summary of the analysis.

Table 4.5 summary of analysis of printed circuit board model.

Assembly Attributes

- o 12 tolerance chains
- o Assembly confidence = (Chain confidence)$^{n=12}$

		Case 1	Case 2
o	chain confidence	92.34%	99.71%
	Assembly confidence	38.43%	96.57%

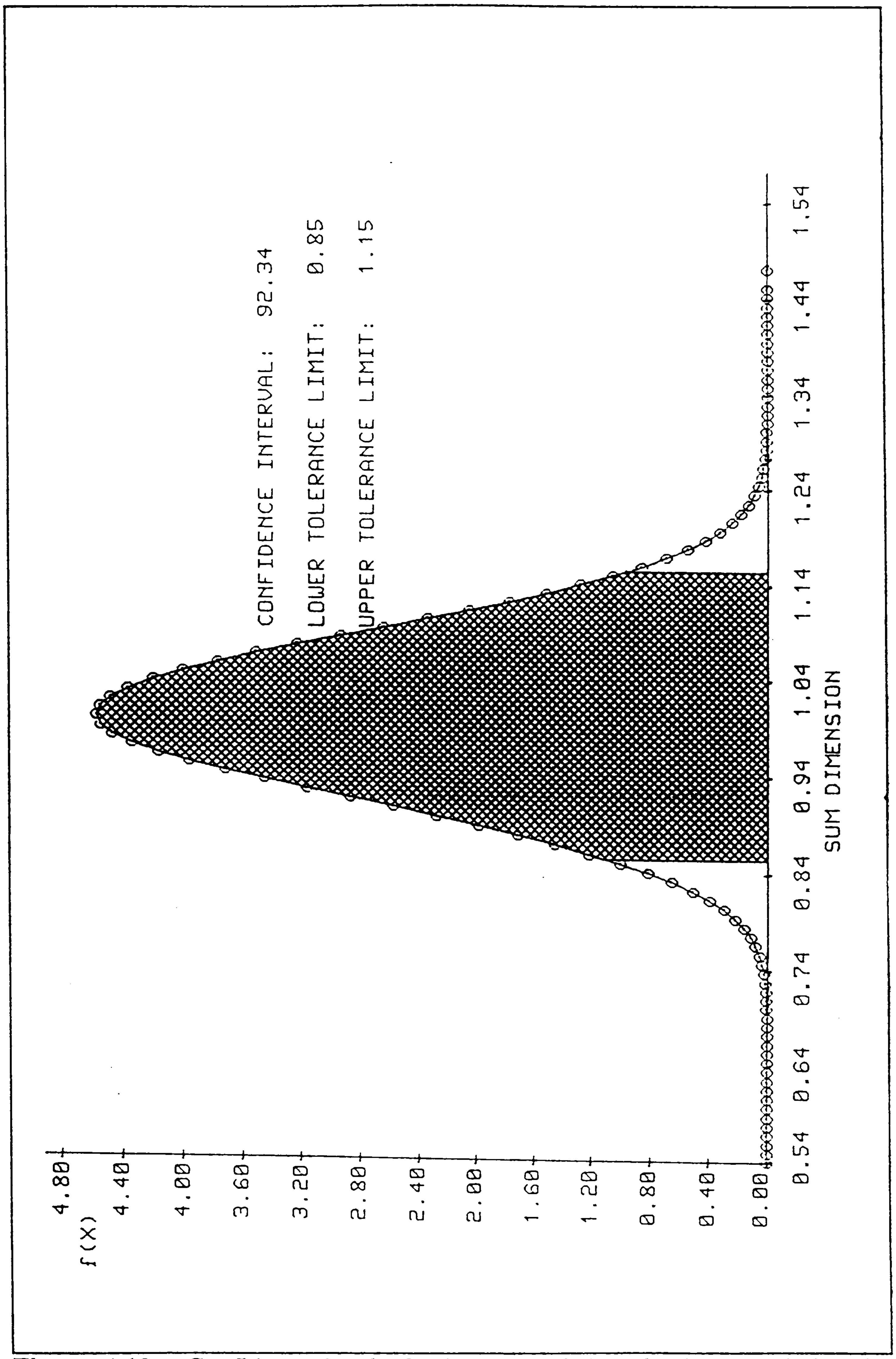

Figure 4.12 Confidence level of printed circuit board tolerance chain with a tolerance zone of 0.3 centered at 1.0.

4.2 Tolerance synthesis

4.2.1 Transmission

The example given for performing a tolerance synthesis uses the transmission shown in Figure 4.13

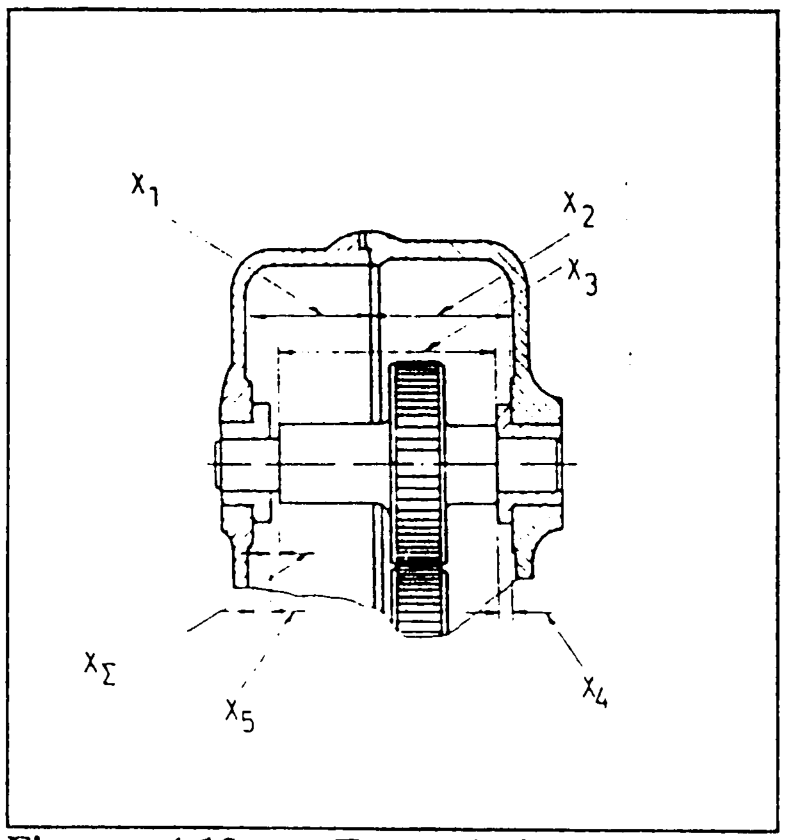

Figure 4.13 Transmission gear box (reproduced from Bjorke).

In Figure 4.13 the sum dimension $X\Sigma$ is the gap between the bushing on the left and the gear hub. This gap must be larger than zero to prevent jamming of the shaft, but less than a certain value to prevent axial motion of the gears. The actual length of the gear hub X_3 is not as important as the size of the gap between the hub and the bushing. The length of the gear hub is, therefore, not the sum dimension; rather the combination of the lengths of the dimensions X_1 - X_5 determines the size of the gap that is the sum dimension. For the sum dimension shown for the gear assembly in Figure 4.13, the fundamental equation is:

$$X\Sigma = X_1 + X_2 - X_3 - X_4 - X_5$$

where:

X_1: width of the left side of the gear box

X_2: width of the right side of the gear box

X_3: distance between the fear hub faces

X_4: thickness of the right side bushing flange

X_5: thickness of the left side bushing flange

It has been determined that, for the assembly to function properly, the sum dimension must be assigned a value of $1\ ^{+}\ 0.125$, which makes the size of the tolerance zone of the sum dimension $TX_\Sigma = 0.25$.

To start the tolerance synthesis, the "TD" choice is selected from the main menu. The terminal then displays the tolerance distribution menu (Figure 4.14).

```
  TOLERANCE DISTRIBUTION

RE - Retrieve Chain
LS - Link Statistics
IE - Initial Estimate
TU - Tolerance Update
```

Figure 4.14 Tolerance distribution menu.

At this point, the tolerance chain data file has not been read into the module yet. "RE", selected to retrieve a tolerance chain, prompts the user to input the file name for the tolerance chain. For the assembly in Figure 4.13, the tolerance chain data file is named trans.chn. After the module has successfully read in the tolerance chain data file, the tolerance distribution menu is again displayed. The link means and variances must now be read into the module. The manner in which they are read from the keyboard is the same as that demonstrated in the compression chamber example and will not be repeated here. For this example a ".lmv" file is already existing with the values listed in Table 4.6 for each of the links.

Table 4.6 Link means and variances used in the transmission example.

$Ez_1 = 0.4$	$varz_1 = 0.056$
$Ez_2 = 0.4$	$varz_2 = 0.056$
$Ez_3 = 0.6$	$varz_3 = 0.056$
$Ez_4 = 0.5$	$varz_4 = 0.028$
$Ez_5 = 0.5$	$varz_5 = 0.028$

With all of the necessary link data read in, the procedure of assigning tolerances begins. "IE" is chosen from the menu to begin assignment of the tolerances. It was decided to determine the tolerance distribution based on a 90 % confidence level of the sum dimension. An estimate of the normalized range is calculated by the module and presented to the user, and the module prompts for the normalized tolerance. Appendix II contains a list of normalized beta distribution parameters. On a 90% confidence level and for a normalized skewness of zero, the normalized tolerance is found from Appendix II to be 3.28.

The module detects the number of unassigned tolerances when it reads in the tolerance chain data file. For this assembly, three of the tolerances are unassigned and two are assigned. It is assumed that the bushing are standard parts, with the following predetermined tolerances on the flanges.

$$TX_4 = TX_5 = 0.15$$

Complexity factors for the assembly must be designated for each of the links with unassigned tolerances. The complexity factors used for this example are listed in Table 4.7.

Table 4.7 Complexity factors used in the assignment of tolerances in the transmission example

Variable	Complexity factor
X_1	1.4
X_2	1.4
X_3	1.0

The module prompts the user for the complexity factor for each unassigned link. The complexity factors listed above in Table 4.7 are entered at the prompt and the module calculates the weighting factors and distributes the tolerances and displays the values of both on the screen.

A value for the expectation of the sum dimension is required for the module to calculate the middle of the tolerance zones for the links that just had tolerances assigned. The obvious choice to make for the expectation is the value of the sum dimension that has been specified for the functional dimension. In this case, it was stated that, for the assembly to operate properly, the sum dimension is specified to be 1 $^{+}$0.125. Ideally, the most suitable value of the expectation would be 1.0. With most of the values around 1.0, it can be expected that the confidence level would be appropriately high. Therefore, at the prompt for the expectation, the value of 1.0 is input.

A list of all of the determinable links accompanied with their basic sizes is displayed, from which the user chooses one of the links to have its middle of the range solved for. In this example, link three was chosen to have its MX calculated. Values for the middle of the range must be entered by the user to complete the tolerance assignment and to solve for the MX for link three. Dimensions X_1 and X_2

were chosen to have symmetric tolerances by selecting the basic size for the middle of the tolerance zone. Finally the calculated dimensions are displayed and the tolerance distribution menu is displayed again. The results for this example are shown below in Table 4.8.

Table 4.8 Tolerance assigned by *TASM* for the complexity factors in Table 4.7

variable	dimension
X_1	40 ± 0.086
X_2	50 ± 0.086
X_3	$79 \begin{smallmatrix} + 0.024 \\ - 0.122 \end{smallmatrix}$

Once the tolerance chain data file is complete, i.e., all of the dimension have associated tolerances, the assembly can be analyzed by the analysis part of the module and have the distribution of the sum dimension displayed for the tolerances that have been assigned. The graphical output of this process is shown below in Figure 4.15. The confidence level calculated for the tolerances assigned at a 90% confidence level is very close at 89.85%. The discrepancy can be expected because table consulted for the normalized tolerance employed the use of an estimate of the normalized range. The confidence displayed in Figure 4.15 has been calculated from the actual range made up of the assigned tolerances.

If the user is not satisfied with the tolerance distribution, he can adjust the complexity factors, the link chosen to be solved for, the values assigned to the middle of the tolerance zones for the links that didn't have the middle points solved for, or any combination of the three. To demonstrate the effect of the modifying complexity factors, another case was run, and the results are shown in Table 4.9. For this case, all of the parameters were left as they were in the previous case,

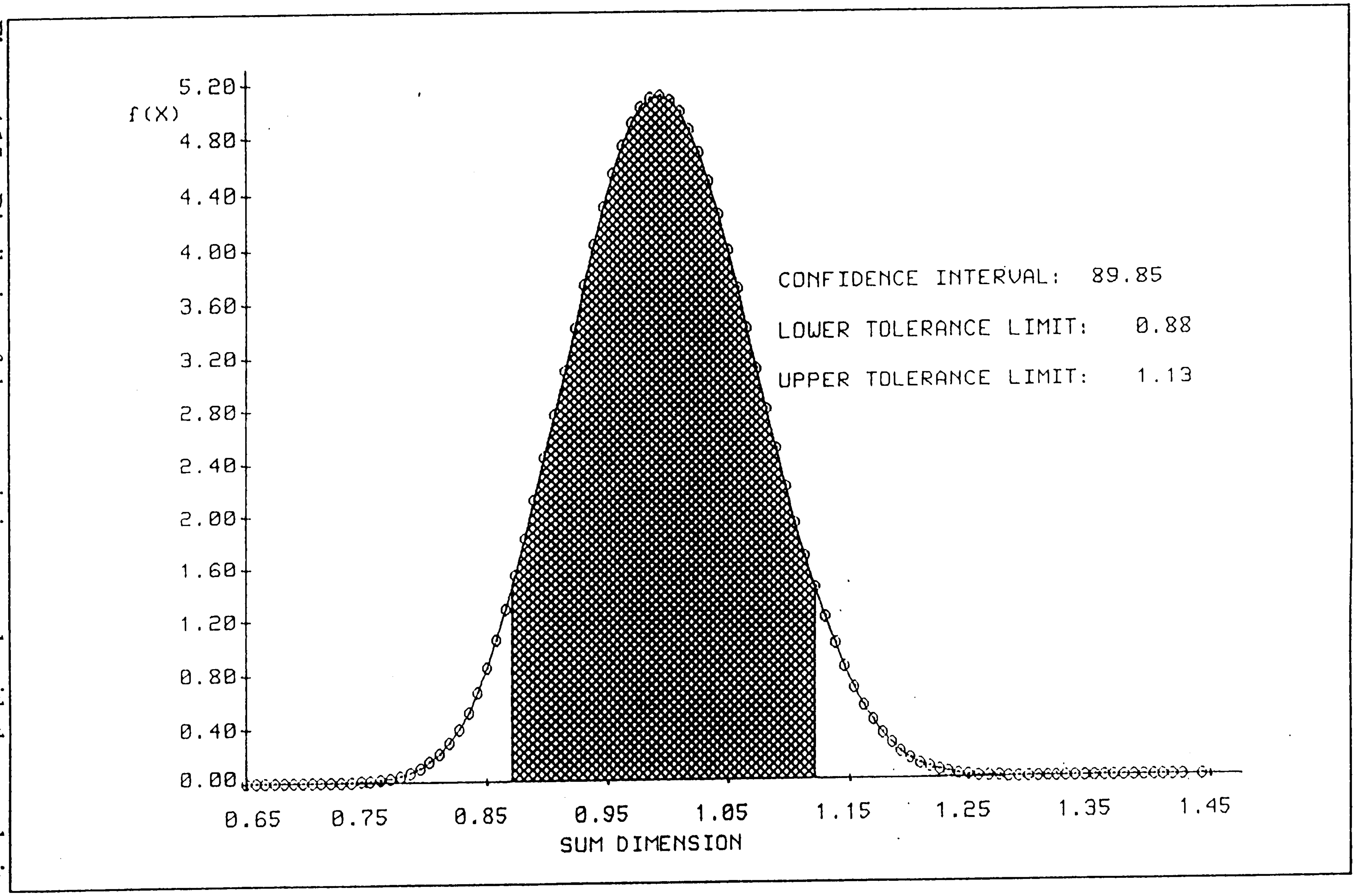

Figure 4.15 Distribution of the transmission example with the complexity factors of Table 4.7.

except for the complexity factors to illustrate the direct impact of altering the complexity factors. The general rule is that assigning a higher complexity factor to a component will result in the module assigning a looser tolerance to the component. Table 4.9 lists the modified complexity factors with their corresponding tolerances zones calculated by the module.

Table 4.9 Modified complexity factors and tolerances for the transmission example.

variable	complexity factor	tolerance zone
X_1	1.7	.184
X_2	1.4	.167
X_3	1.0	.141

For each tolerance synthesis case that is performed, the data corresponding to the case is saved in an output file named case_study.out. In this file the following parameters of the case are recorded.

1) the sum dimension tolerance zone specified by the user

2) the estimated normalized range

3) the normalized tolerance zone input by the user

4) the link chosen to have MX solved for

5) the final values of MX for all of the undetermined links

6) the updated contents of the data base if the results were saved.

A listing of the file case_study.out for the two cases performed above in the transmission example is shown below in Figure 4.16.

```
TZ: 0.05        RW: 10.56     TW: 3.28

CF: 1.40  1.40  1.00

INPUT EX: 1.00        CHOSEN LINK: 3

MXi: 40.00  50.00  78.89

UPDATED CONTENTS OF THE WOULD BE DATABASE
 1.000  40.000  0.190  -0.190
 1.000  50.000  0.190  -0.190
-1.000  79.000  0.530  -0.269
-1.000   5.000  0.750  -0.750
-1.000   5.000  0.750  -0.750


TZ: 0.05        RW: 10.56     TW: 3.28

CF: 1.70  1.40  1.00

INPUT EX: 1.00        CHOSEN LINK: 3

MXi: 40.00  50.00  78.89

UPDATED CONTENTS OF THE WOULD BE DATABASE
 1.000  40.000  0.202  -0.202
 1.000  50.000  0.183  -0.183
-1.000  79.000  0.470  -0.263
-1.000   5.000  0.750  -0.750
-1.000   5.000  0.750  -0.750
```

Figure 4.16 Case_study.out file for the transmission tolerance synthesis example.

5. Conclusions

This thesis has presented an experimental software module that is capable of both evaluating and assigning tolerances on component parts of an assembly. The project goals have been met in that a tool has been successfully implemented that interfaces a geometric model with the tolerance analysis. As the module stands it has eliminated much of the tedium of creating the tolerance chains in Bjorke's method by interfacing with the geometric modelling data base created by Ms. Wang. Although the user does have to provide certain data for the module such as the means and variances of the components of the assembly, the module makes the data entry process easy by allowing the user to save the link means and variances in a file if they will be used again. Although the module is only a prototype, a sucessful user interface has been developed.

Bjorke has presented a clear and concise method most suitable for linking the geometric modelling environment with the concerns of tolerancing in mechanical engineering. The scheme presented in his text is not complete for many assemblies that can be conceived. In the work done here, augmentations have been added such as spreading tolerances across links that are gaps to the individual components. Presently the selection of a gap as having assignable links is limited to having both components of the link with unassigned tolerances. A future enhancement to the module might allow for independence in the assignment status of the individual components of a gap type link.

TASM as it exists is a useful tool that does yield important information about the effects of the individual components on an assembly functioning to specification. During the creation and testing of the module it was observed how varying an individual component's mean and variance affected the confidence of the overall assembly. It was not the intention of this study to perform a massive testing of the module. All of the cases that were run, however, served to confirm our confidence in the analysis and synthesis routines developed. It is recommended that a future project consist of testing the module further to determine sensitive links in assemblies and to further delve into statistical attributes of machining and manufacturing processes.

For a smooth transition into future additions to the concept of an overall design package, the *TASM* module has been coded to operate as tool capable of interfacing with a larger scheme. By taking a modular approach additions and alterations to the code can be implemented without disrupting the entire module.

Throughout the formulation of the method to analyze and assign the tolerances, a number of peripheral support concepts were uncovered. Some of them are actual enhancements of the existing module, while others are distinct additions that would serve to expand the module's capabilities. Because of the nature of developing a prototypical module within the time constraints, several capabilities that were envisioned were not fully developed.

One such instance that should be noted exists in the tolerance synthesis section of the module. At present the user selects which link the module will solve for in placing the tolerances on the unassigned parts while the user places the remaining calculated tolerances. A suitable enhancement would be to allow the module to "spread" this delta MX across all of the unassigned links that have an MX that is not zero either evenly or according to some developed scheme.

Much of the success or certainty in the results of a tolerance analysis or synthesis resides in the confidence one has in the values of the statistical data input to the routine. Not much work has actually been done in the area of statistical analysis of manufactured parts. With the onslaught of computer monitoring of machines in industry, more and more information about individual machines' statistical idiosyncrasies will become available. With either more confidence in statistical data for common machining practices, or better yet, with accurate statistical data for a factory's particular machines, analysis conventions such as the one described in this paper will find greater acceptance in the market of industrial design and planning.

Applications of certain Artificial Intelligence (A.I.) techniques such as knowledge based or expert system technology might prove to be suitable for interfacing with *TASM*. Presently based on a parts geometry and the tolerance necessary for proper performance of the part, an engineer knowledgeable in machining processes selects a process that is congruent with both the part's design and also the facility constraints placed upon him. Statistical data for the process is either known for the individual machine selected, or they are extracted from the table of typical machining capabilities. This work could be performed by an expert system module that would have access to the geometric model data base as well as the statistical data base. With access to the geometric data, the system could determine whether the part would be turned, milled, etc. Depending on the tolerance necessary for the part the system could then search the statistical data base for the machines in the factory to determine which process is to be used and also what statistical parameters of the selected process. Not only could the expert system be used to determine which processes are optimally compatible with facilities ability to manufacture the parts, it could also at the same time generate consistent process plans for the parts. A schematic of how such a system might be laid out is briefly illustrated in Figure 5.1.

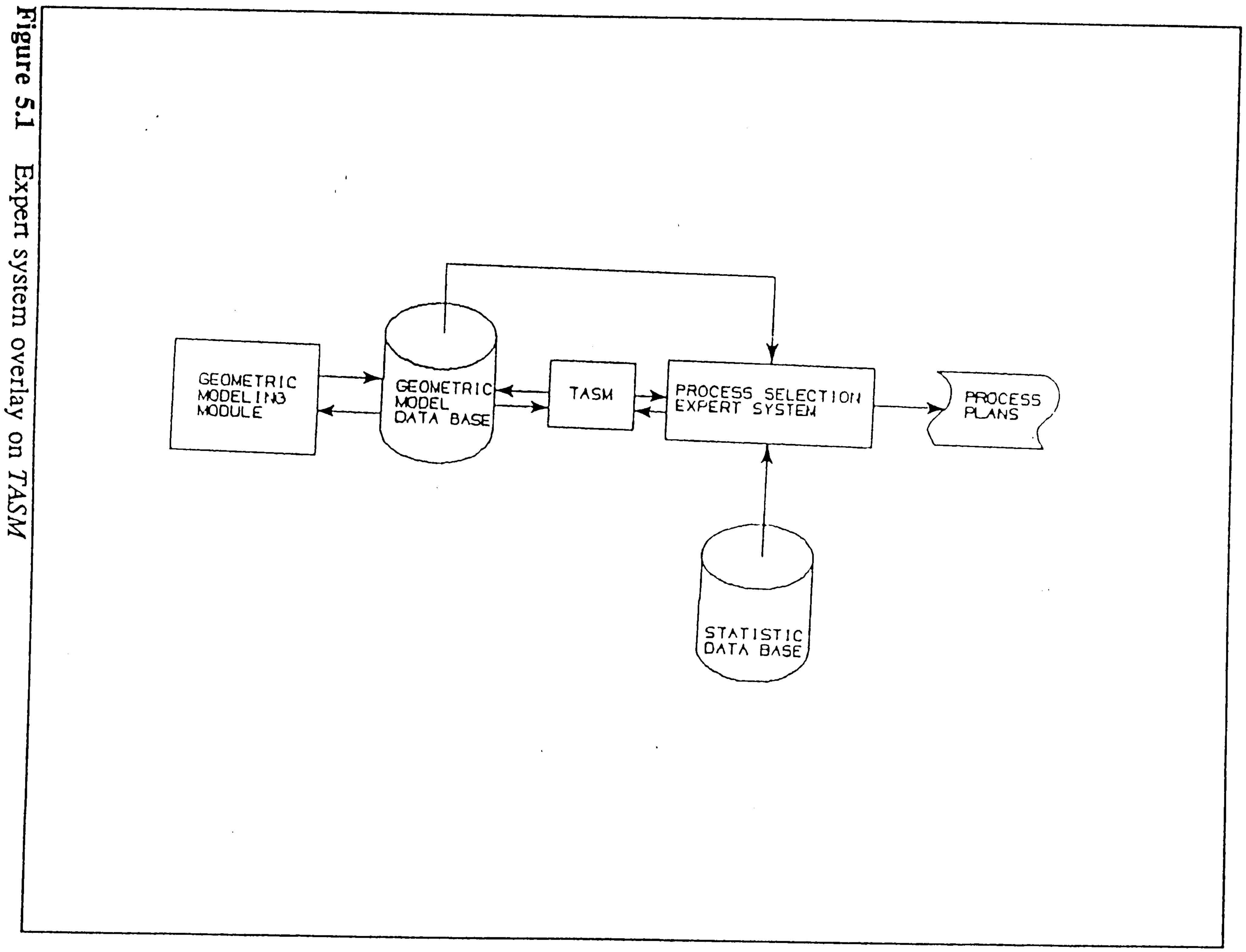

Figure 5.1 Expert system overlay on *TASM*

Another area where A.I. applications may yield positive results is in the selection of possible functional dimensions. A system could be envisioned to determine the dimensions where a variation in size may pose an ill effect on the assembly's functionality. Such dimensions could be presented to the design engineer for inspection followed by the automatic generation of the tolerance chains for analysis in *TASM*.

Surely future work will continue in the area of analyzing and optimally assigning tolerances for assembly. Computer automation and a growing pool of design data existing in geometric model data bases will aid in providing the necessary data and a medium necessary for automated tolerance analysis. This paper describes only a small segment of all of the necessary data and procedures for a complete tolerance assessment at design time.

Appendix I

Unit Distribution Parameters of Typical Machining Processes

Form element	Technological process	TX		EZ		varZ	
		min	max	min	max	min	max
External cylinders	Rough turning	IT11	IT13	0.40	0.60	0.034	0.054
	Finishing turning	IT7	IT9	0.40	0.60	0.030	0.054
	Grinding	IT5	IT7	0.50	0.55	0.028	0.047
Internal cylinders	Rough turning	IT11	IT13	0.40	0.50	0.034	0.054
	Drilling	IT10	IT14	0.40	0.50	0.034	0.066
	Finishing turning	IT8	IT10	0.40	0.50	0.030	0.054
	Reaming	IT7	IT9	0.45	0.50	0.028	0.040
	Grinding	IT5	IT7	0.40	0.50	0.028	0.047
	Honing	IT4	IT6	0.45	0.50	0.028	0.034
Radial runout of cylindrical sufaces	Turning	0.02	0.10	0.30	0.40	0.035	0.040
	Grinding	0.01	0.02	0.30	0.40	0.035	0.038
The distance between external parallel planes	Cut off	IT12	IT16	0.30	0.70	0.040	0.083
	Turning	IT10	IT13	0.50	0.65	0.028	0.054
	Milling, planing	IT10	IT15	0.40	0.60	0.028	0.054
	Grinding	IT5	IT7	0.45	0.55	0.028	0.050

Form element	Technological process	TX		EZ		varZ	
		min	max	min	max	min	max
The distance between internal parallel planes	Turning Milling, planing Grinding	IT10 IT10 IT5	IT13 IT15 IT7	0.40 0.38 0.38	0.60 0.62 0.62	0.034 0.034 0.040	0.054 0.054 0.050
The distance between external and internal planes	Turning Milling, planing Grinding	IT10 IT10 IT5	IT13 IT15 IT7	0.40 0.50 0.48	0.60 0.55 0.52	0.034 0.034 0.034	0.054 0.054 0.050
Parallelism, perpendicularity and angularity between surfaces	Planing Milling Grinding	$\frac{0.1}{300}$ $\frac{0.1}{300}$ $\frac{0.02}{300}$	$\frac{0.2}{300}$ $\frac{0.3}{300}$ $\frac{0.1}{300}$	0.50 0.50 0.50	0.50 0.50 0.50	0.034 0.034 0.040	0.054 0.040 0.054

Form element	Technological process	TX		EZ		varZ	
		min	max	min	max	min	max
The distance from a center line to a reference plane	Boring	IT7	IT10	0.48	0.52	0.028	0.040
	Face milling	IT9	IT12	0.50	0.65	0.028	0.047
	Face grinding	IT6	IT9	0.65	0.70	0.047	0.056
Parallelism, perpendicularity and angularity between a center line and a reference plane	Boring	$\frac{0.05}{300}$	$\frac{0.1}{300}$	0.50	0.50	0.034	0.047
	Face milling	$\frac{0.1}{300}$	$\frac{0.3}{300}$	0.50	0.50	0.034	0.040
	Face grinding	$\frac{0.05}{300}$	$\frac{0.1}{300}$	0.50	0.50	0.040	0.047
The distance between center-lines	Boring	IT7	IT10	0.50	0.55	0.028	0.047

Form element	Technological process	TX		EZ		varZ	
		min	max	min	max	min	max
Parallelism, perpenducularity and angularity between center lines	Boring	$\frac{0.05}{300}$	$\frac{0.2}{300}$	0.5	0.5	0.034	0.047

Appendix II

Normalized Beta Unit Distribution Parameters

CONFIDENCE LEVEL = 90.00 0/0

RW \ FW	-.4	-.3	-.2	-.1	.0	.1	.2	.3	.4
5	-	-	3.22	3.27	3.28	3.27	3.22	-	TW
	-	-	-.72	-.38	.00	.38	.72	-	MW
6	-	-	3.25	3.29	3.30	3.29	3.25	-	-
	-	-	-.98	-.51	.00	.51	.98	-	-
7	-	3.19	3.27	3.29	3.30	3.29	3.27	3.19	-
	-	-1.74	-1.22	-.62	.00	.62	1.22	1.74	-
8	-	3.21	3.28	3.29	3.30	3.29	3.28	3.21	-
	-	-2.10	-1.45	-.73	.00	.73	1.45	2.10	-
9	-	3.23	3.28	3.29	3.30	3.29	3.28	3.23	-
	-	-2.44	-1.67	-.84	.00	.84	1.67	2.44	-
10	-	3.24	3.28	3.29	3.30	3.29	3.28	3.24	-
	-	-2.77	-1.88	-.95	.00	.95	1.88	2.77	-
11	-	3.25	3.28	3.29	3.30	3.29	3.28	3.25	-
	-	-3.09	-2.10	-1.05	.00	1.05	2.10	3.09	-
12	3.10	3.26	3.29	3.29	3.30	3.29	3.29	3.26	3.10
	-4.35	-3.41	-2.31	-1.16	.00	1.16	2.31	3.41	4.35
13	3.12	3.26	3.29	3.29	3.29	3.29	3.29	3.26	3.12
	-4.79	-3.73	-2.51	-1.26	.00	1.26	2.51	3.73	4.79
14	3.14	3.27	3.29	3.29	3.29	3.29	3.29	3.27	3.14
	-5.22	-4.04	-2.72	-1.37	.00	1.37	2.72	4.04	5.22
15	3.16	3.27	3.29	3.29	3.29	3.29	3.29	3.27	3.16
	-5.65	-4.35	-2.93	-1.47	.00	1.47	2.93	4.35	5.65
16	3.18	3.27	3.29	3.29	3.29	3.29	3.29	3.27	3.18
	-6.07	-4.66	-3.13	-1.57	.00	1.57	3.13	4.66	6.07
17	3.19	3.27	3.29	3.29	3.29	3.29	3.29	3.27	3.19
	-6.49	-4.97	-3.33	-1.67	-.00	1.67	3.33	4.97	6.49
18	3.20	3.28	3.29	3.29	3.29	3.29	3.29	3.28	3.20
	-6.91	-5.28	-3.54	-1.77	.00	1.77	3.54	5.28	6.91
19	3.21	3.28	3.29	3.29	3.29	3.29	3.29	3.28	3.21
	-7.33	-5.59	-3.74	-1.87	-.00	1.87	3.74	5.59	7.33
20	3.22	3.28	3.29	3.29	3.29	3.29	3.29	3.28	3.22
	-7.74	-5.89	-3.95	-1.98	.00	1.98	3.95	5.89	7.74
25	3.24	3.28	3.29	3.29	3.29	3.29	3.29	3.28	3.24
	-9.79	-7.41	-4.96	-2.48	.00	2.48	4.96	7.41	9.79
30	3.26	3.28	3.29	3.29	3.29	3.29	3.29	3.28	3.26
	-11.83	-8.93	-5.96	-2.98	.00	2.98	5.96	8.93	11.83

TW = 3.29 NORMALIZED NORMAL
MW = 0.00 DISTRIBUTION

CONFIDENCE LEVEL = 95.00 0/0

FW / RW	-.4	-.3	-.2	-.1	.0	.1	.2	.3	.4
5	-	-	3.63	3.69	3.71	3.69	3.63	-	TW
	-	-	-.58	-.32	.00	.32	.58	-	MW
6	-	-	3.74	3.78	3.79	3.78	3.74	-	-
	-	-	-.86	-.45	.00	.45	.86	-	-
7	-	3.71	3.79	3.82	3.83	3.82	3.79	3.71	-
	-	-1.55	-1.12	-.58	.00	.58	1.12	1.55	-
8	-	3.76	3.83	3.85	3.86	3.85	3.83	3.76	-
	-	-1.92	-1.36	-.69	.00	.69	1.36	1.92	-
9	-	3.79	3.85	3.87	3.87	3.87	3.85	3.79	-
	-	-2.28	-1.59	-.81	-.00	.81	1.59	2.28	-
10	-	3.82	3.86	3.88	3.88	3.88	3.86	3.82	-
	-	-2.63	-1.81	-.92	.00	.92	1.81	2.63	-
11	-	3.83	3.87	3.89	3.89	3.89	3.87	3.83	-
	-	-2.96	-2.03	-1.03	.00	1.03	2.03	2.96	-
12	3.71	3.85	3.88	3.89	3.89	3.89	3.88	3.85	3.71
	-4.09	-3.29	-2.24	-1.13	.00	1.13	2.24	3.29	4.09
13	3.73	3.86	3.89	3.90	3.90	3.90	3.89	3.86	3.73
	-4.54	-3.62	-2.46	-1.24	.00	1.24	2.46	3.62	4.54
14	3.75	3.87	3.89	3.90	3.90	3.90	3.89	3.87	3.75
	-4.99	-3.94	-2.67	-1.34	.00	1.34	2.67	3.94	4.99
15	3.77	3.87	3.90	3.90	3.90	3.90	3.90	3.87	3.77
	-5.43	-4.26	-2.88	-1.45	-.00	1.45	2.88	4.26	5.43
16	3.79	3.88	3.90	3.90	3.91	3.90	3.90	3.88	3.79
	-5.87	-4.57	-3.09	-1.55	.00	1.55	3.09	4.57	5.87
17	3.80	3.88	3.90	3.91	3.91	3.91	3.90	3.88	3.80
	-6.30	-4.89	-3.29	-1.65	.00	1.65	3.29	4.89	6.30
18	3.81	3.89	3.90	3.91	3.91	3.91	3.90	3.89	3.81
	-6.73	-5.20	-3.50	-1.76	.00	1.76	3.50	5.20	6.73
19	3.82	3.89	3.91	3.91	3.91	3.91	3.91	3.89	3.82
	-7.15	-5.51	-3.70	-1.86	.00	1.86	3.70	5.51	7.15
20	3.83	3.89	3.91	3.91	3.91	3.91	3.91	3.89	3.83
	-7.57	-5.82	-3.91	-1.96	.00	1.96	3.91	5.82	7.57
25	3.86	3.90	3.91	3.91	3.91	3.91	3.91	3.90	3.86
	-9.66	-7.36	-4.93	-2.47	.00	2.47	4.93	7.36	9.66
30	3.88	3.91	3.91	3.92	3.92	3.92	3.91	3.91	3.88
	-11.72	-8.88	-5.94	-2.97	.00	2.97	5.94	8.88	11.72

TW = 3.92 NORMALIZED NORMAL
MW = 0.00 DISTRIBUTION

CONFIDENCE LEVEL = 99.00 0/0

RW \ FW	-.4	-.3	-.2	-.1	.0	.1	.2	.3	.4
5	-	-	4.24	4.30	4.32	4.30	4.24	-	TW
	-	-	-.35	-.20	-.00	.20	.35	-	MW
6	-	-	4.51	4.57	4.59	4.57	4.51	-	-
	-	-	-.63	-.34	.00	.34	.63	-	-
7	-	4.61	4.68	4.73	4.74	4.73	4.68	4.61	-
	-	-1.16	-.90	-.48	.00	.48	.90	1.16	-
8	-	4.72	4.79	4.83	4.84	4.83	4.79	4.72	-
	-	-1.56	-1.16	-.60	.00	.60	1.16	1.56	-
9	-	4.80	4.87	4.90	4.91	4.90	4.87	4.80	-
	-	-1.94	-1.40	-.73	.00	.73	1.40	1.94	-
10	-	4.87	4.92	4.95	4.96	4.95	4.92	4.87	-
	-	-2.31	-1.64	-.84	.00	.84	1.64	2.31	-
11	-	4.91	4.96	4.99	4.99	4.99	4.96	4.91	-
	-	-2.67	-1.88	-.96	-.00	.96	1.88	2.67	-
12	4.93	4.95	4.99	5.01	5.02	5.01	4.99	4.95	4.93
	-3.52	-3.02	-2.10	-1.07	.00	1.07	2.10	3.02	3.52
13	4.94	4.98	5.02	5.03	5.04	5.03	5.02	4.98	4.94
	-4.00	-3.36	-2.32	-1.18	.00	1.18	2.32	3.36	4.00
14	4.96	5.00	5.04	5.05	5.05	5.05	5.04	5.00	4.96
	-4.47	-3.70	-2.54	-1.29	.00	1.29	2.54	3.70	4.47
15	4.97	5.02	5.05	5.06	5.07	5.06	5.05	5.02	4.97
	-4.94	-4.03	-2.76	-1.40	-.00	1.40	2.76	4.03	4.94
16	4.98	5.04	5.06	5.07	5.08	5.07	5.06	5.04	4.98
	-5.40	-4.36	-2.98	-1.50	.00	1.50	2.98	4.36	5.40
17	5.00	5.05	5.07	5.08	5.09	5.08	5.07	5.05	5.00
	-5.85	-4.69	-3.19	-1.61	.00	1.61	3.19	4.69	5.85
18	5.01	5.06	5.08	5.09	5.09	5.09	5.08	5.06	5.01
	-6.30	-5.01	-3.40	-1.71	.00	1.71	3.40	5.01	6.30
19	5.02	5.07	5.09	5.10	5.10	5.10	5.09	5.07	5.02
	-6.74	-5.33	-3.61	-1.82	.00	1.82	3.61	5.33	6.74
20	5.03	5.08	5.09	5.10	5.10	5.10	5.09	5.08	5.03
	-7.18	-5.65	-3.82	-1.92	.00	1.92	3.82	5.65	7.18
25	5.07	5.10	5.12	5.12	5.12	5.12	5.12	5.10	5.07
	-9.34	-7.22	-4.86	-2.44	.00	2.44	4.86	7.22	9.34
30	5.09	5.12	5.13	5.13	5.13	5.13	5.12	5.12	5.09
	-11.45	-8.77	-5.88	-2.95	.00	2.95	5.88	8.77	11.45

TW = 5.15 NORMALIZED NORMAL
MW = 0.00 DISTRIBUTION

CONFIDENCE LEVEL = 99.50 0/0

RW \ FW	-.4	-.3	-.2	-.1	.0	.1	.2	.3	.4
5	- -	- -	4.41 -.28	4.46 -.17	4.48 .00	4.46 .17	4.41 .28	- -	TW MW
6	- -	- -	4.75 -.55	4.81 -.30	4.83 .00	4.81 .30	4.75 .55	- -	- -
7	- -	4.91 -1.03	4.97 -.82	5.02 -.44	5.04 -.00	5.02 .44	4.97 .82	4.91 1.03	- -
8	- -	5.05 -1.42	5.12 -1.08	5.16 -.57	5.18 .00	5.16 .57	5.12 1.08	5.05 1.42	- -
9	- -	5.16 -1.81	5.22 -1.33	5.26 -.69	5.27 .00	5.26 .69	5.22 1.33	5.16 1.81	- -
10	- -	5.23 -2.18	5.29 -1.57	5.33 -.81	5.34 .00	5.33 .81	5.29 1.57	5.23 2.18	- -
11	- -	5.30 -2.55	5.35 -1.81	5.38 -.93	5.38 .00	5.38 .93	5.35 1.81	5.30 2.55	- -
12	5.41 -3.29	5.34 -2.90	5.39 -2.04	5.41 -1.04	5.42 .00	5.41 1.04	5.39 2.04	5.34 2.90	5.41 3.29
13	5.40 -3.78	5.38 -3.25	5.42 -2.27	5.44 -1.15	5.45 .00	5.44 1.15	5.42 2.27	5.38 3.25	5.40 3.78
14	5.42 -4.26	5.41 -3.60	5.45 -2.49	5.47 -1.26	5.47 .00	5.47 1.26	5.45 2.49	5.41 3.60	5.42 4.26
15	5.42 -4.74	5.44 -3.94	5.47 -2.71	5.49 -1.37	5.49 .00	5.49 1.37	5.47 2.71	5.44 3.94	5.42 4.74
16	5.43 -5.20	5.46 -4.27	5.49 -2.93	5.50 -1.48	5.51 .00	5.50 1.48	5.49 2.93	5.46 4.27	5.43 5.20
17	5.45 -5.66	5.48 -4.60	5.50 -3.14	5.52 -1.59	5.52 .00	5.52 1.59	5.50 3.14	5.48 4.60	5.45 5.66
18	5.46 -6.12	5.49 -4.93	5.51 -3.36	5.53 -1.69	5.53 -.00	5.53 1.69	5.52 3.36	5.49 4.93	5.46 6.12
19	5.47 -6.57	5.50 -5.25	5.53 -3.57	5.53 -1.80	5.54 .00	5.53 1.80	5.53 3.57	5.50 5.25	5.47 6.57
20	5.48 -7.02	5.51 -5.57	5.53 -3.78	5.54 -1.90	5.55 .00	5.54 1.90	5.53 3.78	5.51 5.57	5.48 7.02
25	5.52 -9.20	5.55 -7.16	5.56 -4.83	5.57 -2.42	5.57 .00	5.57 2.42	5.56 4.83	5.55 7.16	5.52 9.20
30	5.54 -11.33	5.57 -8.71	5.58 -5.85	5.58 -2.94	5.58 .00	5.58 2.94	5.58 5.86	5.57 8.71	5.54 11.33

TW = 5.60 NORMALIZED NORMAL
MW = 0.00 DISTRIBUTION

CONFIDENCE LEVEL = 99.73 0/0

FW / RW	-.4	-.3	-.2	-.1	.0	.1	.2	.3	.4
5	-	-	4.52	4.57	4.59	4.57	4.52	-	TW
	-	-	-.23	-.14	.00	.14	.23	-	MW
6	-	-	4.92	4.98	5.00	4.98	4.92	-	-
	-	-	-.48	-.27	.00	.27	.48	-	-
7	-	5.14	5.19	5.24	5.26	5.24	5.19	5.14	-
	-	-.92	-.75	-.41	.00	.41	.75	.92	-
8	-	5.31	5.37	5.42	5.44	5.42	5.37	5.31	-
	-	-1.31	-1.01	-.54	.00	.54	1.01	1.31	-
9	-	5.44	5.50	5.54	5.55	5.54	5.50	5.44	-
	-	-1.69	-1.27	-.66	.00	.66	1.27	1.69	-
10	-	5.53	5.59	5.63	5.64	5.63	5.59	5.53	-
	-	-2.07	-1.51	-.78	.00	.78	1.51	2.07	-
11	-	5.60	5.66	5.69	5.70	5.69	5.66	5.60	-
	-	-2.44	-1.75	-.90	.00	.90	1.75	2.44	-
12	5.79	5.66	5.71	5.74	5.75	5.74	5.71	5.66	5.79
	-3.10	-2.81	-1.99	-1.02	.00	1.02	1.99	2.81	3.10
13	5.81	5.71	5.76	5.78	5.79	5.78	5.76	5.71	5.81
	-3.58	-3.16	-2.22	-1.13	.00	1.13	2.22	3.16	3.58
14	5.81	5.75	5.79	5.81	5.82	5.81	5.79	5.75	5.81
	-4.07	-3.51	-2.44	-1.24	.00	1.24	2.44	3.51	4.07
15	5.80	5.78	5.82	5.83	5.84	5.83	5.82	5.78	5.80
	-4.56	-3.85	-2.67	-1.35	-.00	1.35	2.67	3.85	4.56
16	5.81	5.80	5.84	5.85	5.86	5.85	5.84	5.80	5.81
	-5.03	-4.19	-2.89	-1.46	.00	1.46	2.89	4.19	5.03
17	5.82	5.83	5.86	5.87	5.88	5.87	5.86	5.83	5.82
	-5.50	-4.52	-3.10	-1.57	.00	1.57	3.10	4.52	5.50
18	5.83	5.84	5.87	5.88	5.89	5.88	5.87	5.84	5.83
	-5.96	-4.86	-3.32	-1.68	.00	1.68	3.32	4.86	5.96
19	5.84	5.86	5.89	5.90	5.90	5.90	5.89	5.86	5.84
	-6.42	-5.18	-3.53	-1.78	.00	1.78	3.53	5.18	6.42
20	5.86	5.87	5.89	5.91	5.91	5.91	5.90	5.87	5.85
	-6.87	-5.51	-3.75	-1.89	.00	1.89	3.75	5.51	6.87
25	5.90	5.92	5.93	5.94	5.94	5.94	5.93	5.92	5.90
	-9.08	-7.10	-4.80	-2.41	.00	2.41	4.80	7.10	9.08
30	5.92	5.94	5.95	5.96	5.96	5.96	5.95	5.94	5.92
	-11.23	-8.67	-5.83	-2.93	.00	2.93	5.83	8.67	11.23

TW = 6.00 NORMALIZED NORMAL
MW = 0.00 DISTRIBUTION

CONFIDENCE LEVEL = 99.90 0/0

RW \ FW	-.4	-.3	-.2	-.1	.0	.1	.2	.3	.4
5	-	-	4.66	4.70	4.72	4.70	4.66	-	TW
	-	-	-.17	-.10	.00	.10	.17	-	MW
6	-	-	5.15	5.21	5.23	5.21	5.15	-	-
	-	-	-.39	-.23	.00	.23	.39	-	-
7	-	5.46	5.49	5.55	5.57	5.55	5.49	5.46	-
	-	-.76	-.65	-.36	.00	.36	.65	.76	-
8	-	5.67	5.72	5.78	5.80	5.78	5.72	5.67	-
	-	-1.14	-.91	-.49	.00	.49	.91	1.14	-
9	-	5.83	5.89	5.94	5.96	5.94	5.89	5.83	-
	-	-1.52	-1.17	-.62	.00	.62	1.17	1.52	-
10	-	5.96	6.02	6.06	6.07	6.06	6.02	5.96	-
	-	-1.91	-1.42	-.74	.00	.74	1.42	1.91	-
11	-	6.05	6.11	6.15	6.16	6.15	6.11	6.05	-
	-	-2.28	-1.66	-.86	.00	.86	1.66	2.28	-
12	6.32	6.13	6.18	6.22	6.23	6.22	6.18	6.13	6.32
	-2.84	-2.65	-1.90	-.98	.00	.98	1.90	2.65	2.84
13	6.41	6.19	6.24	6.27	6.28	6.27	6.24	6.19	6.41
	-3.29	-3.01	-2.14	-1.10	.00	1.10	2.14	3.01	3.29
14	6.37	6.24	6.29	6.31	6.32	6.31	6.29	6.24	6.37
	-3.80	-3.37	-2.37	-1.21	.00	1.21	2.37	3.37	3.80
15	6.37	6.28	6.32	6.35	6.35	6.35	6.32	6.28	6.37
	-4.29	-3.72	-2.60	-1.32	.00	1.32	2.60	3.72	4.29
16	6.39	6.32	6.35	6.37	6.38	6.37	6.35	6.32	6.39
	-4.77	-4.06	-2.82	-1.43	.00	1.43	2.82	4.06	4.77
17	6.39	6.35	6.38	6.40	6.40	6.40	6.38	6.35	6.39
	-5.25	-4.40	-3.04	-1.54	.00	1.54	3.04	4.40	5.25
18	6.40	6.37	6.40	6.42	6.42	6.42	6.40	6.37	6.40
	-5.72	-4.74	-3.26	-1.65	.00	1.65	3.26	4.74	5.72
19	6.41	6.39	6.42	6.43	6.44	6.43	6.42	6.39	6.41
	-6.18	-5.07	-3.48	-1.76	-.00	1.76	3.48	5.07	6.18
20	6.42	6.41	6.43	6.45	6.45	6.45	6.43	6.41	6.42
	-6.64	-5.40	-3.69	-1.87	.00	1.87	3.69	5.40	6.64
25	6.46	6.47	6.49	6.50	6.50	6.50	6.49	6.47	6.46
	-8.89	-7.02	-4.75	-2.39	.00	2.39	4.75	7.02	8.88
30	6.49	6.50	6.52	6.52	6.52	6.52	6.51	6.50	6.49
	-11.06	-8.59	-5.79	-2.91	.00	2.91	5.80	8.59	11.06

TW = 6.58 NORMALIZED NORMAL
MW = 0.00 DISTRIBUTION

Appendix III

Development of Eccentricity and Distributed Center Location

eccentricities

A plane vector span with distributed direction will be denoted an eccentricity, and it is a dimension which can form any angle to the sum direction. In order to clarify the behaviour of eccentricities, we will give the following example.

Let the distance between the center line of the shaft and that of the tapered pin, be the sum dimension in the assembly shown in fig. 3.4. This dimension will be influenced by other parts, among which is the bushing. Let us assume that the cylindrical surfaces of the bushing are eccentric to each other, and that the eccentricity is the only geometric deviation in the assembly. Then the dispersion of the sum dimension will be given from the magnitude and the direction of the eccentricity alone. The magnitude of the eccentricity is a quality of the bushing as a part, and it came into existence during the machining of the workpiece. The direction of the eccentricity is, on the other hand, a result of the assembly of the workpiece. It may therefore, be concluded that the magnitude and the direction of an eccentricity are uncorrelated.

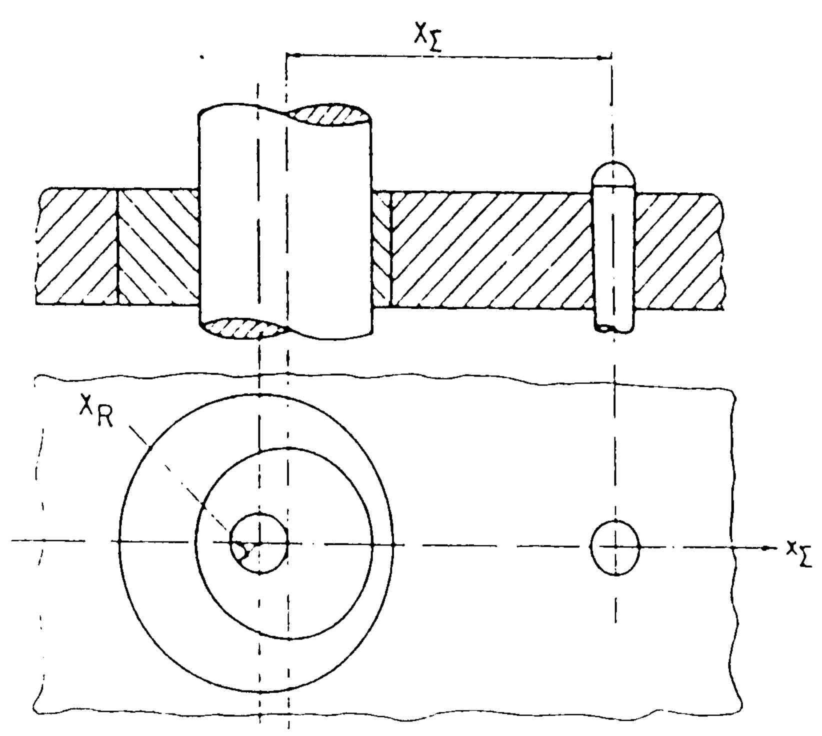

Fig. 3.4. Bushing with eccentricity

The magnitude of an eccentricity is a result of errors occurring during machining of a workpiece. A detailed analysis of these errors is outside the scope of this text. We need, however, a basis for estimation of the

distributions of eccentricities. In order to illustrate such a basis, we will analyze the effect of the setting up errors in a four jaw chuck as an example.

During the setting up procedure of a bushing in the chuck as shown in fig. 3.5, we make a positioning error in X - direction and Y - direction independently. Due to the symmetry in the picture, we will expect the probability density of the positioning error to be the same in both X- and Y- direction. From practical experience, we know that these distributions are almost normal. The question we then ask is,

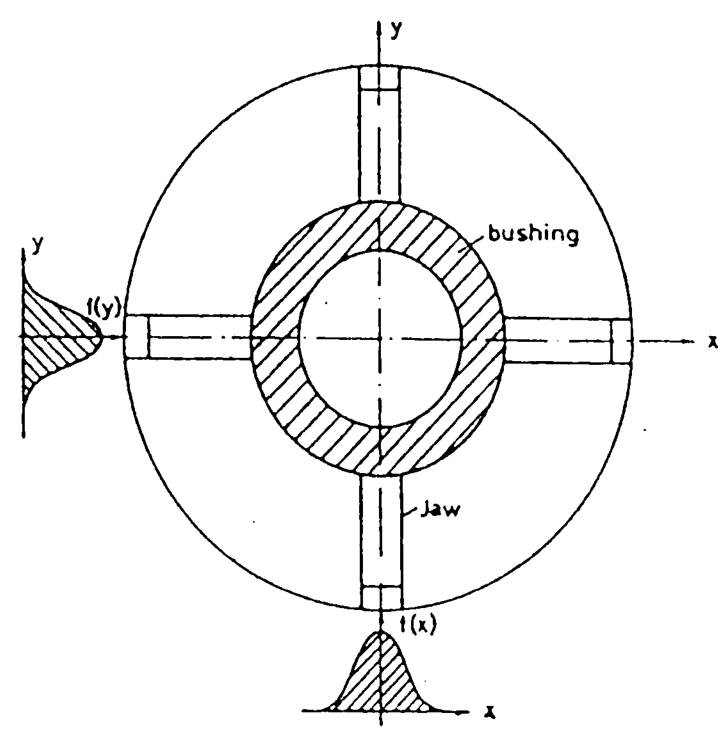

Fig. 3.5. Setting up a bushing in a chuck

what distribution does the eccentricity of the cylindrical surfaces of the bushing follow, after internal turning? The mathematical formulation of this question is, what is the distribution of:

$$X_R = \sqrt{X^2 + Y^2}$$

when the independent variates X and Y both are normally destributed $(0,\sigma)$.

The bivariate density of X and Y is:

Appendix III

$$f(x,y) = \frac{1}{2\pi\sigma^2} \exp[-(x^2 + y^2)/2\sigma^2]$$

that is, the variate $(x^2 + y^2)/\sigma^2$ is χ^2 distributed with 2 degrees of freedom. As a consequence of that the density of X_R may be given from:

$$f_1 = \frac{X_R}{\sigma^2} \exp\left[-\frac{X_R^2}{\sigma^2}\right]$$

which is the density of a Rayleigh distribution (see fig. 3.6). From this deduction we may draw the important conclusion that the magnitude of an eccentricity is Rayleigh distributed to the same level of accuracy as its components are normally distributed. This result is general and it is valid not only for the case analyzed above.

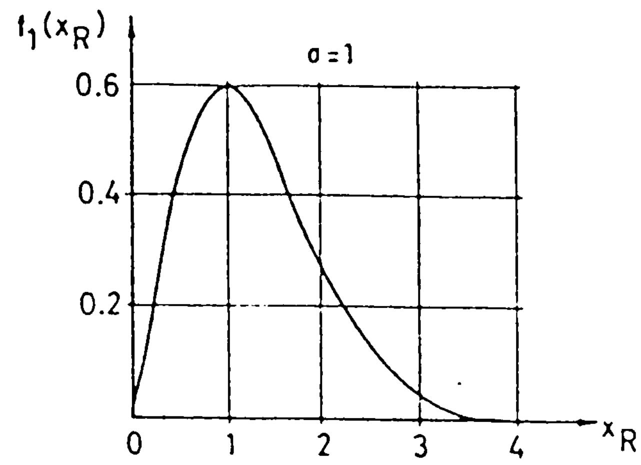

Fig. 3.6. Rayleigh distribution

The direction of an eccentricity is a result of the angular location of the bushing when it is put into the bore. If we study fig. 3.4, it is easy to realize that no direction could have any preference, and we may conclude that every direction ought to have the same probability. That is, the distribution of X_ϕ has to be rectangular in the range $[0,2\pi]$, with the probability density:

$$f_2(X_\phi) = \frac{1}{2\pi}$$

The influence of the eccentricity on the sum dimension in fig. 3.4, may be given approximately by the projection of the eccentricity on the sum

direction. The fundamental equation of an eccentricity will, therefore, in general, be given by:

$$X_\Sigma = A\ X_R \cos X_\phi$$

where

A - a signed constant

X_R - the magnitude of the eccentricity

X_ϕ - the direction of the eccentricity

Equation (3.10) contains a product of two stochatastic, but uncorrelated variables, X_R and $\cos X_\phi$. The expectation and the variance of that product may be found by using (A.4) In order to do so the expectation and the variance of $\cos X_\phi$ have to be computed from:

$$EY = EX_1 + EX_2$$

$$\mathrm{var}Y = \mathrm{var}X_1\ \mathrm{var}X_2 + \mathrm{var}X_1 (EX_2)^2 + \mathrm{var}X_2 (EX_1)^2$$

(A.4)

$$E(\cos X_\phi) = \int_0^{2\pi} \cos X_\phi\ f_2(X_\phi) dX_\phi = \frac{1}{2\pi}\int_0^{2\pi} \cos X_\phi\ dX_\phi = 0$$

$$E(\cos^2 X_\phi) = \int_0^{2\pi} \cos^2 X_\phi\ f_2(X_\phi)\ dX_\phi = \frac{1}{2\pi}\int_0^{2\pi} \cos^2 X_\phi\ dX_\phi = 1/2$$

(3.11)

$$\mathrm{var}(\cos X_\phi) = E(\cos^2 X_\phi) - (E(\cos X_\phi))^2 = 1/2$$

By substitution of (3.11) into (A.4) we get:

$$EX_\Sigma = EX_R\ E(\cos X_\phi) = EX_R\ 0 = 0$$

$$\mathrm{var}X_\Sigma = \mathrm{var}X_R\ \mathrm{var}(\cos X_\phi) + \mathrm{var}X_R (E(\cos X_\phi))^2 + \mathrm{var}(\cos X_\phi)$$

$(EX_R)^2$

$$= \mathrm{var}X_R\,\tfrac{1}{2} + \mathrm{var}X_R(0)^2 + \tfrac{1}{2}\,(EX_R)^2 \tag{3.12}$$

$$= \tfrac{1}{2}\,(\mathrm{var}X_R + (EX_R)^2) = \tfrac{1}{2}\,\mathrm{var}_o X_R$$

where $\mathrm{var}_o X_R$ is the variance of the magnitude of the eccentricity measured relative to the origin (remenber $\mathrm{var}X_R$ is measured relative to the expectation EX_R).

As a summary of the influence of an eccentricity on the parameters of the sum dimension, we give the formulas where unit dimensions have been substituted:

$$\begin{aligned} MX_{\Sigma R} &= 0 \\ R\Delta X_\Sigma &= |A|\ 2TX_R \\ E\Delta X_\Sigma &= 0 \\ \mathrm{var}\Delta X_\Sigma &= A^2\ \tfrac{1}{2}TX_R^2\,\mathrm{var}_o Z_R \end{aligned} \tag{3.13}$$

Where $\mathrm{var}_o Z_R$ is the variance of the unit distribution measured relative to the origin.

Due to symmetry, both $MX_{\Sigma R}$ and $E\Delta X_R$ are zero. From fig. 3.4, it can be seen that the range $R\Delta X_\Sigma$ has te be twice that of X_R, and since we use a model of the individual dimensions saying that the range is equal to the tolerance, equation (3.13) results.

Normally distributed center location

The probability density of a normal distribution is:

$$g_1(x) = \frac{1}{\sqrt{2\pi}\ \sigma}\ \exp\left[-\frac{(x-\zeta)^2}{2\sigma^2}\right] \tag{3.25}$$

Let us assume that the center location of the internal part is given by the density (3.25), and that the range $[-X_R, X_R]$ corresponds to 3σ limits of the distribution. That is:

$$\zeta = 0 \qquad\qquad 3\sigma = X_R$$

Substituting into (3.25), we get the conditional probability density of X_Σ given X_R:

$$g_1(X_\Sigma|X_R) = \frac{3}{\sqrt{2\pi}\, X_R} \exp\left[-\frac{9}{2}\left(\frac{X_\Sigma}{X_R}\right)^2\right] \qquad (3.26)$$

If we substitute (3.26) into (3.24) we get:

$$g(X_\Sigma) = \int_0^{X_R} \frac{3}{\sqrt{2\pi}\, X_R} \exp\left[-\frac{9}{2}\left(\frac{X_\Sigma}{X_R}\right)^2\right] f_1(X_R)\, dX_R \qquad (3.27)$$

Let us for a moment, suppose that the variate X_Σ and X_U:

$$X_\Sigma = X_R\, X_U \qquad (3.28)$$

and that

$f_1(X_R)$ - the probability density of X_R

$h(X_U)$ - the probability density of X_U

By using the multiplication formula:

$$Y = X_1\ X_2$$

$$g(y) = \int |\tfrac{1}{z}|\, f_1(z)\, f_2\left(\frac{y}{z}\right) dz$$

where

$f_1(X_1)$ is the probability density of X_1

Appendix III

$f_2(X_2)$ is the probability density of X_2

we may under the above assumptions, express the probability density of X_Σ as:

$$g(X_\Sigma) = \int_0^{X_R} \left|\frac{1}{X_R}\right| h\left(\frac{X_\Sigma}{X_R}\right) f_1(X_R)\, dX_R \qquad (3.29)$$

If we compare the integrals in (3.27) and (3.29), we may easily see that X_Σ can be written as the product (3.28), if the probability density of X_U is:

$$h(X_U) = \frac{3}{\sqrt{2\pi}} \exp\left[-\frac{9X_U^2}{2}\right] \qquad (3.30)$$

Which is the density of a normal distribution having:

$$\zeta = 0 \qquad \sigma = \frac{1}{3} \qquad (3.31)$$

As mentioned above, the aim of this analysis is to determine the statistical parameters of the variate X_Σ. This can now be done by using our knowledge of the statistical parameters of the variates in the product (3.28), and it is, consequently, no longer necessary to perform the integration (3.27). By substitution of (3.31) into (A.4) we get:

$$EX_\Sigma = EX_R\ EX_U = EX_R\ \zeta = 0$$

$$\mathrm{var}X_\Sigma = \mathrm{var}X_R\ \mathrm{var}X_U + \mathrm{var}X_R(EX_U)^2 + \mathrm{var}X_U\ (EX_R)^2$$

$$= \mathrm{var}X_R\ \frac{1}{9} + \mathrm{var}X_R(0)^2 + \frac{1}{9}\ (EX_R)^2$$

$$= \frac{1}{9}\ (\mathrm{var}X_R + (EX_R)^2) = \frac{1}{9}\ \mathrm{var}_O X_R \qquad (3.32)$$

As can be seen from this equation, we have succeeded in expressing the variance of the sum dimension as a function of the variance of the

mating parts, given by the variate X_R.

We are now in a position to express the influence on the parameters of the sum dimension from a line vector gap with normal distributed magnitude:

$$\begin{aligned}
&MX_{\Sigma R} = 0 \\
&R\Delta X_{\Sigma} = |A|\ 2TX_R \qquad (3.33) \\
&E\Delta X_{\Sigma} = 0 \\
&\mathrm{var}\Delta X_{\Sigma} = A^2\ \tfrac{1}{9} TX_R^2\, \mathrm{var}_o Z_R
\end{aligned}$$

Both $MX_{\Sigma R}$ and $E\Delta X_{\Sigma}$ are zero , due to the symmetry of the gap. The range $R\Delta X_{\Sigma}$ has to be twice that of the variate X_R, as was the case with eccentricities. The range of X_R, may be found from:

$$RX_R = TX_R = \tfrac{1}{2}\,(UX_b - LX_a) \qquad (3.34)$$

Appendix IV

List of routines

Main calling program
TASM

Menu.olb
dsply_menu
get_option
read_menus
up_case

Lclass.olb
bivar
eccentricity
lspan
l_clearance
l_dist_ctr
l_transition
p_gap_dd_lm
pspan
p_clearance
p_dist_ctr
p_transition
read_lmv

Support.olb
analyze_chain
axisl
axticd
beta
bigfind
clear
connect
crlmv
defaults
gamma_func
gauleg
get_chain
inest
labcalc
labeldat
open_file
plot_dist
qgauss_gen
read_lmv
save
smallfind
sr_ecc
sr_slant
sum_sr
sum_ss
system_parameters
ticl
tolplot
updat_dbase

Appendix V

Beta distribution

$$f(x) = \frac{1}{(b\text{-}a)\beta(\gamma,\eta)} \left(\frac{(x\text{-}a)}{(b\text{-}a)}\right)^{\gamma\text{-}1} \left(1\text{-}\left(\frac{x\text{-}a}{b\text{-}a}\right)\right)^{\eta\text{-}1}$$

$$a = MX_{\Sigma R} - R\Delta X$$

$$b = MX_{\Sigma R} + R\Delta X$$

$$\gamma = \frac{(EX\text{-}a)^2(b\text{-}EX) - varX(EX\text{-}a)}{varX(b\text{-}a)}$$

$$\eta = \frac{(EX\text{-}a)(b\text{-}EX)^2 - varX(b\text{-}EX)}{varX(b\text{-}a)}$$

$$EX = MX_{\Sigma R} + E\Delta X$$

$$varX = var\Delta X$$

$$\beta(\gamma,\eta) = \frac{\Gamma(\gamma)\Gamma(\eta)}{\Gamma(\gamma+\eta)}$$

$$\Gamma = \int_0^\infty t^{x\text{-}1}\, e^{\text{-}t}\, dt$$

References

[1] Parkinson, D.B.,"The Application of Probability Methods to Tolerancing" Trans.ASME J. Mech Design Vol 104 (1982) pp 612-618

[2] Parkinson, D.B., "Tolerancing of Component Dimensions in CAD", Computer Aided Design, Vol. 16, No. 1, Jan. 1984, pp 25-32

[3] Parkinson, D.B., "Assessment and Optimization of Dimensional Tolerances" Computer Aided Design, Vol. 17, No. 4, May 1985, pp 191-197

[4] Turner, J.U., Wozny, M.J., and Hoh, D.D., "Tolerance Analysis in a Solid Modeling Environment", Computers in Engineering, Proc. of the 1987 ASME International Conference and Exhibition, Aug. 1987, New York, NY, pp 169-175

[5] Michael, W. and Siddall, J.N., "The Optimization Problem with Optimal Tolerance Assignment and Full Acceptance", ASME Trans., Journal of Mechanical Design, Vol 103, Oct. 1981, pp 842-848

[6] Michael, W. and Siddall, J.N., "The Optimal Tolerance Assignment with Less Than Full Acceptance", ASME Trans., Journal of Mechanical Design, Vol 104, Oct. 1982, pp 855-860

[7] Bjorke, O., "Computer_Aided Tolerancing", Tapir publisher, Trondheim, Norway 1978.

[8] Wang, N., "Program of Study for the Degree of Doctor of Philosophy in Mechanical Engineering", Lehigh University, 1988.

[9] Ozsoy, T.,"Graph3d.lu Graphics Package Reference Manual", Lehigh University, 1985.

[10] Haugrud,B., "Tolerances in Mechanical Assemblies". Norwegian Defense Research Establishment FFIS S-10, Kjeller, 1961.

[11] Press, W., "Numerical Recipes the Art of Scientific Computing", Cambridge University Press, New York, NY 1987

[12] Graham, N. "Computer-Aided Tolerance Analysis of Boundary Geometry Representations of Assemblies", B.S. Thesis, M.I.T June 1982.

[13] Griscavage, A. "Plot2d", Senior project report, Lehigh University, 1983.

[14] Shigley, Joseph, and, Mitchell, Larry, "Mechanical Engineering Design", McGraw-Hill, Inc., New York, NY 1983.

VITA

The author was born in Levent Idaho, the first of seven children to the descendants of an Irish potato share-cropping family. A fascination with mechanisms at an early age eventually led to the study of mechanical engineering. A long trip and a descent SAT score led the author to study mechanical engineering at Villanova University, Villanova Pa. The author attended the national championship game in 1985 at Lexington Kentucky between Villanova and Georgetown in which the Wildcats won 66-64. The work ethic experienced at childhood payed off at college helping the author to graduate *Summa Cum Laude*. After graduation continued study led to attending Lehigh University where the author received his M.Sc. in mechanical engineering in Jan. 1989.