OPTIMAL ALLOCATION OF TYPES AND MAGNITUDES OF GEOMETRIC TOLERANCES

By

ASHRAF MOHAMED OSAMA ABDEL-AZIZ NASSEF, B.Sc. Hons (Mech. Eng.), M.Sc.

A Thesis

Submitted to the School of Graduate Studies

in partial Fulfillment of the Requirements

for the Degree

Doctorate of Philosophy

McMaster University

(c) Copyright by Ashraf M. Osama Nassef, February 1996

DOCTOR OF PHILOSOPHY (1996) McMASTER UNIVERSITY

(Mechanical Engineering) Hamilton, Ontario, Canada

TITLE: Optimal Allocation of Types and Magnitudes of Geometric Tolerances

AUTHOR: Ashraf Mohamed Osama Abdel-Aziz Nassef
B.Sc. Hons. (Mech. Eng.) (Cairo University)
M.Sc. (Mech. Eng.) (Cairo University)

SUPERVISOR: Dr. Hoda A. ElMaraghy

NUMBER OF PAGES: xxvii,241

OPTIMAL ALLOCATION OF GEOMETRIC TOLERANCES

To my first engineering tutor; my father:

Osama Nassef

ABSTRACT

Manufacturing processes generate surfaces with variable dimensions and geometries. The produced surfaces deviate from their nominal geometry and consequently, critical dimensions and clearaces deviate from their designed values when the parts are assembled. Since it is impossible to check whether the functional requirements fall within their allowable range after the assembly process, geometric tolerances are specified on individual features during the design stage such that, parts whose geometric deviations would cause a violation of the functional requirements when assembled to other parts, would be rejected during the inspection process. Therefore, wrong choice of geometric tolerances could lead to either rejecting good parts, or the acceptance of bad parts leading to an assembly that violates the functional requirements. Furthermore, tolerance selection is not limited to the magnitudes of the geometric tolerances. Each feature in the assembly has four geometric variations that need to be controlled. These are size, position, form and orientation. Each of these variations, except variation in size, can be controlled by several types of tolerances and the selection of tolerance types affects the percentage of accepted or rejected parts during inspection.

This dissertation presents a novel computer–aided method for the synthesis of magnitudes and types of geometric tolerances in mechanical assemblies. Tolerance selection is formulated as a combinatorial optimization problem, where each feature has seven variables. These are the types of tolerances controling the orientation, form and position variations, the magnitudes of these tolerances as well as the magnitude size tolerance. A new criterion was developed to allocate geometric tolerances which is the minimum mismatch probability defined as the probability of rejecting a part during the inspection process which

satisfy the functional requirements when assembled to other parts, or accepting one violating the functional requirements when assembled to other parts.

In order to evaluate the objective function, a probabilistic analysis method, based on Monte Carlo simulation, was developed to calculate the rejection probabilities of assemblies with geometric tolerances. The surface of each feature in the assembly is represented with a number of points where the coordinates of these points are the random variables. In each simulation cycle, the coordinates of the surface points are generated using the probability distribution associated with the manufacturing process. The inspection process is then simulated where the geometric deviations on each feature are checked against the specified tolerances. Finally, the functional requirements are checked. Several methods for parts joining were examined and a new genetic algorithms based method is developed to evaluate the maximum and minimum values of critical clearances. The use of genetic algorithms ensures the arrival to the global minimum and maximum values of the clearance. Due to the large number of random variables, and since the probabilistic analysis is used in every optimization step in the tolerance allocation algorithm, two variance reduction techniques are incorporated with the standard Monte Carlo simulation to reduce the sample size.

A number of genetic algorithms based routines are used for checking of geometric deviations on the generated parts. In many cases the evaluation of geometric deviations involve optimization. The use of global optimizers ensures the correct evaluation of deviations and avoids the unnecessary rejection of good parts. The advantage of using genetic algorithms is demonstrated with several examples used by previous researchers. Furthermore, a new parametric surface interpolation method is developed to approximate the actual surface of the manufactured parts and help in the evaluation of some geometric deviations that cannot be evaluated directly using the generated points.

The new tolerance allocation method, presented in this dissertation, attempts to fill a void area in the tolerancing research, which is the selection of the "types" in addition to magnitudes of geometric tolerances. Although the skill of a tolerancing practitioner is still needed to specify candidate tolerance types for each geometric control, the developed robust mathematical formulation of the problem avoids the random human factors in the selection. The proposed methodology can be extended for incorporation within computer-aided tolerancing systems to assist designers in selecting geometric tolerances.

ACKNOWLEDGEMENT

I would like to thank my supervisor, Dr. Hoda A. ElMaraghy, for her guidance and support throughout the course of this research. I would like also to express my gratitude to the members of my supervisory committee Dr. J. Siddall and Dr. W.F.S. Poehlman. I am very grateful to Dr. Hoda ElMaraghy, the Mechanical Engineering Department, and McMaster University for the Research Scholarships, the Teaching Assistantships and scholarships I have received for financial support. I would like to thank the Mechanical Engineering Department secretaries Louise Perry, Jane Mah and Betty Bedell-Ryc.

I would like to thank everyone who helped me during my stay in the Flexible Manufacturing Center, McMaster University, namely Mr. Todd Pfaff the system engineer, Mr. Hubert Chu and Dr. Zhang Wu who provided me with a great amount of initial guidance in the area of tolerancing.

I would like to extend my thanks to everyone who helped me during my stay in the Intelligent Manufacturing Systems lab, namely Dr. Waguih ElMaraghy, Mr. Dan Corrin the system engineer and Mrs. Patricia Jackson the administrative secretary.

My sincere thanks go to the colleagues and dear friends of mine, Mr. El-Houssaine Waled who provided me with several mathematical routines for my program, Dr. Mohamed Gadallah who participated with me in several discussions and information exchange.

My final thanks go to my brother Mr. Hazem Nassef who provided me with an insight to several computer science areas that are related to my field, to my wife Sherin ElRafei who supported me in all her strength and helped me in the thesis write-up and, to my dear friends Dr. Ahmed S. Zaki, Dr. Atef Massoud and Mr. Anis Limaiem.

Finally I present this work to my great parents as well as my brother and my sister.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER ONE

INTRODUCTION

This chapter gives a brief review of the current tolerancing practice, the motivations behind the presented research, the approach followed during the research, the contributions to the tolerancing research and an overview of the dissertation.

1 .1 Review of Tolerancing Practice

Late in the nineteenth century, machine designers realized that manufacturing processes do not produce the ideal dimensions of designed parts. The idea of associating a lower and an upper bound with each dimension was introduced. These bounds were known as tolerances. During the inspection process, if a dimension exceeded the specified bounds, then that part would be rejected. Tolerances were allocated to ensure the proper function of mating features. Fits of mating features were classified as: (1) clearances, (2) location fits, and (3) interference fits, with various sub–grades in each classification. Each sub–grade was assigned a tolerance value depending on the nominal size of the mating features. A suitable type of fit is selected according to the application, and a tolerance value is selected for each of the two mating features.

1 .1 .1 Tolerance Chains

With the breakout of the Second World War, the accuracy requirements in assemblies became tighter. Designers had to consider other dimensions that affect parts and allocate tolerances to them as well. Thus the idea of tolerance charts was introduced to analyze the effect of various dimensions on a critical clearance or on a pair of mating parts. Eventually, critical dimensions were also considered for analysis and the term "functional requirements" was introduced. It refers to any limits specified by designers on any geometrical entities to

ensure the assembly's functionality. Tolerance charts were limited to linearly related tolerances. Since many applications had dimensions that are nonlinearly related, tolerance chains with nonlinear relationships were analyzed.

1 .1 .2 Probabilistic Analysis

The allocation of tight tolerances led to higher manufacturing cost. Therefore, tolerancing researchers in the early sixties made use of the fact that the variability by which machines produce dimensions can be modelled by certain probability distributions. They calculated the probability that an assembly will satisfy the functional requirements given a set of dimensional tolerances imposed on the assembly's features. The probabilistic analysis of tolerances led to the idea of allowing a small percentage of assembly rejection in order to allocate wider tolerances.

1 .1 .3 Tolerance Synthesis

The research done in tolerance analysis was paralleled in another field, the allocation of tolerances, known in the tolerancing literature as "tolerance synthesis". Manufacturing cost was found to decrease exponentially with the increase of tolerance magnitudes. At the same time, however, wider tolerances lead to higher rejection rates. Traditionally the tradeoff between cost and rejection rate has been solved by allocating tolerances to minimize the production cost while constraining the rejection rate to a certain value.

1 .1 .4 Geometric Tolerances

As early as the 1930s dimensional tolerances were found to be insufficient for controlling manufacturing variations. Geometric tolerances were introduced as limitations on controlling variations in features' geometries like perpendicularity, circularity, size, etc. Since the late 1950s geometric dimensioning and tolerancing standards have evolved. These standards specify the types of tolerances to be used to control certain geometric variations.

For example, the ANSI Y14.5M 1982 standards specify cylindricity, circularity and straightness as types of form tolerances to control the form variation of cylindrical features. Studies in the analysis and allocation of geometric tolerances started in the late eighties and are currently ongoing.

1 .1 .5 Current Tolerancing Practice

Figure 1 .1 outlines the steps in tolerancing practice and its associated processes:

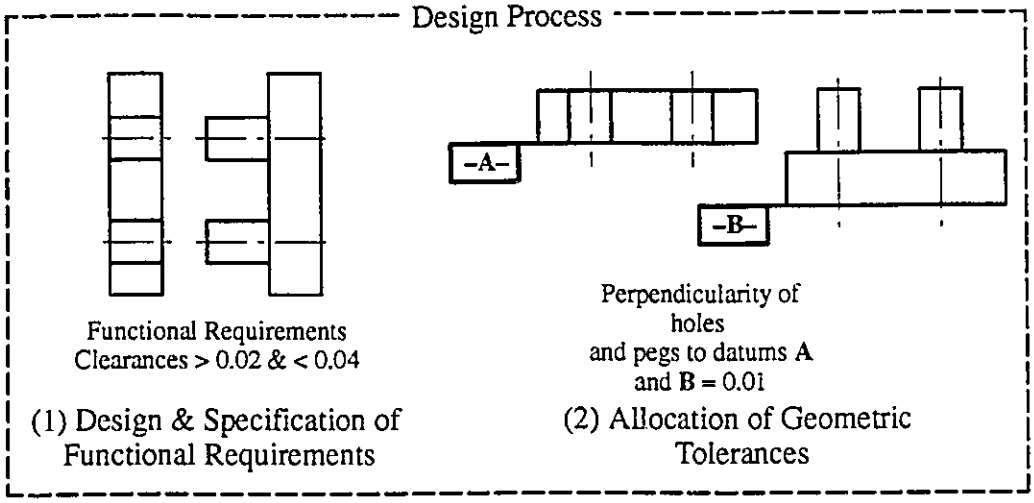
Step 1. **Design and Specification of Functional Requirements.** The design functional requirements are specified in this step.

Step 2. **Tolerance Allocation.** Since it is impossible to check the assembly functional requirements after the assembly process (Step 5), the geometric variations of each individual part are controlled by geometric tolerances such that parts which would not function as desired after assembly are rejected during the inspection process (Step 4) and parts which would function well after assembly are accepted.

Step 3. **Manufacturing.** On the shop floor, manufacturing processes are responsible for generating the various part features. The fixturing of each part is dictated by the datum reference frame of its position tolerances. There can be more than one combination of processes to generate the assembly features. Each has a different cost and a different effect on the rejection rate.

Step 4. **Inspection.** The manufactured parts are inspected for the imposed geometric tolerances in Step 2. A part whose geometric deviations fall out of the tolerance bounds is rejected otherwise, it is assembled to other parts in step 5.

Step 5. **Assembly.**



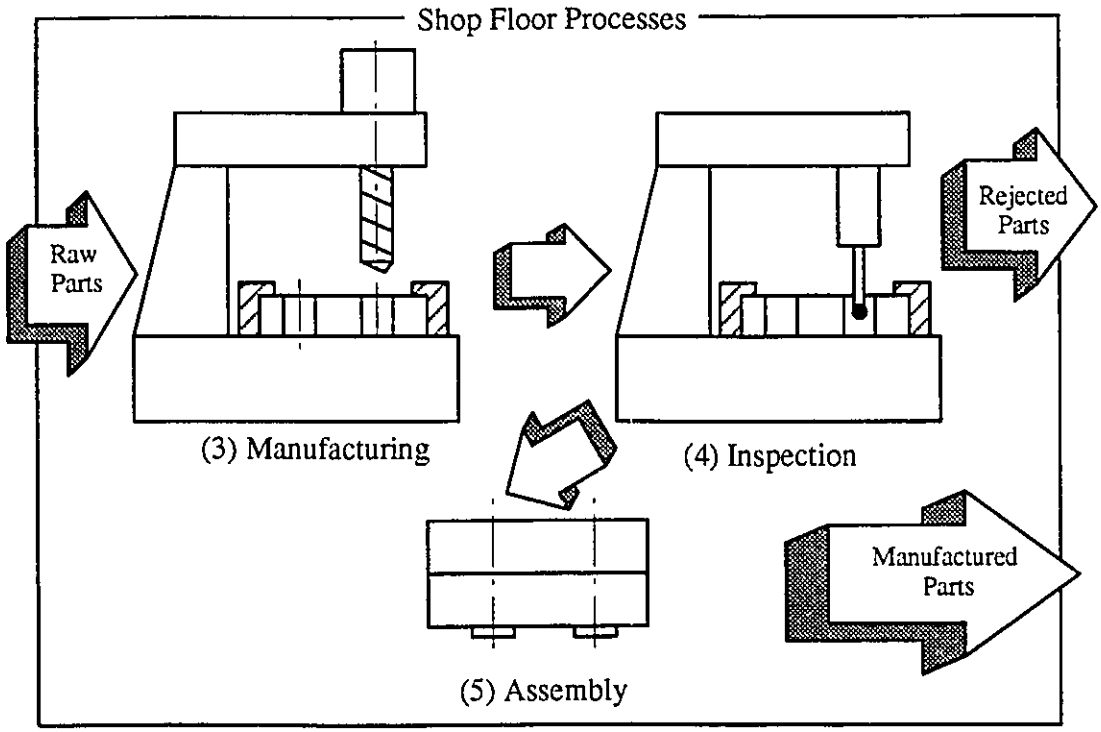


Figure 1 .1 Outline of Tolerancing Practice

1 .2 Motivation

The tolerancing literature lacks a comprehensive study of the allocation of geometric tolerances to assemblies. Tolerance synthesis has in large been confined to dimensional tolerances. At the same time, tolerancing standards provide a wide range of types of tolerances. The proper selection of the types of tolerances is a missing area in the available tolerancing literature. The presented work was mainly motivated by these two reasons. Since the area of tolerance synthesis depends on other related areas like tolerance analysis and evaluation of geometric deviations, these areas had to be investigated as well. This led to further motivations for the research work presented in the dissertation which are detailed in the following sections.

1 .2 .1 Tolerance Synthesis and Choice of Tolerance Types

Tolerances should be allocated to features such that the assembled parts would not violate the functional requirements and the parts which, would cause violation of functional requirements if assembled to other parts, would be rejected during inspection. Referring to Figure 1 .1 , this means that the geometric bounds imposed by the tolerances in step 2 should be selected such that parts which will assemble while conforming to the design functional requirements in step 5 would be accepted during inspection (step 4) and parts which will violate the functional requirements when assembled to other parts in step 5 are rejected during the inspection process. If the above criterion is not observed, either assemblies violating the functional requirements will be obtained, or good parts will be rejected. Previous work (Lee, W., 1988) in the area of tolerance synthesis focused on the selection of dimensional tolerances. The allocation of geometric tolerances to assemblies was tackled by few researchers. Traditionally, the criterion used to select tolerances is the minimum production cost, and the same criterion was used to allocate geometric tolerances. This

criterion guarantees the optimal selection of manufacturing processes to manufacture the various features in the assembly, but it does not address the inspection process and thus does not guarantee that tolerances are allocated to satisfy the assembly functional requirements. Therefore, a new criterion is needed for the allocation of geometric tolerances.

Another area that has received little attention is the selection of tolerance types. A feature in an assembly may be controlled for size position, form and orientation variations. Each of these variations can be controlled by a number of tolerance types. For example, an orientation variation of a feature can be controlled with a perpendicularity tolerance relative to one datum or by parallelism tolerance relative to another datum. The selection of one particular tolerance type can lead to a higher probability of accepting a part that will cause a violation of functional requirements when assembled to other parts (or vice versa). Thus a new formulation of the synthesis problem is needed in which the selection of tolerance types is taken into account.

1 .2 .2 Tolerance Analysis of Assemblies with Geometric Tolerances

Tolerance analysis of geometric tolerances in assemblies refers to calculating the probability of accepting (or rejecting) an assembly using a specific set of manufacturing processes and a specific manufacturing sequence (Step 3 in Figure 1 .1), a specific set of algorithms for checking geometric deviations on the manufactured parts (Step 4 in Figure 1 .1) and a specific assembly sequence (Step 5 in Figure 1 .1). The evaluation of these probabilities is needed for the allocation of tolerances as discussed in the previous section (**1 .2 .1**). Since, it is difficult to establish a closed form mathematical relation between the functional requirements and geometric tolerances, Monte Carlo simulation has been used for the probabilistic analysis of assemblies with geometric tolerances. The simulation of the manufacturing sequence, the inspection process and the assembly sequence is absent from the tolerance analysis literature. Since an accurate evaluation of the

probabilities of rejection and acceptance is impossible without their inclusion in the tolerance analysis process, a new formulation is needed where these processes and sequences are taken into account.

1 .2 .3 Evaluation of Actual Geometric Deviations

The simulation of the inspection process (Step 4 in Figure 1 .1) is needed to evaluate the probability of rejection as shown in the previous section. Hence a set of algorithms is required to evaluate geometric deviations on simulated manufactured surfaces. A considerable amount of research has been done in the development of algorithms that evaluate the geometric deviations on manufactured parts. Since tolerancing standards specify that deviations calculated from the points measured on a manufactured surface, should be at their minimum value, optimization is used to find the minimum deviations on the manufactured surfaces. All optimization methods used to date stop their search whenever a local minima is found. Shunmugam (1991) states that the presence of local minima can affect the optimization search adversely. A global optimization method is needed to evaluate geometric deviations. Another area that needs attention is the establishment of datums and datum reference frames on manufactured surfaces from which position and orientation deviations are measured.

1 .3 Objectives and Approach

The objective of the research reported in this dissertation is to develop a procedure for allocating geometric tolerance types and values to assembly features to ensure that the inspection process will reject parts that will cause a violation of the functional requirements, along with any necessary routines for tolerance analysis, assembly simulation and the evaluation of geometric deviations. In current tolerancing practice, geometric tolerances are allocated according to the expertise of the tolerance designer. The allocation process is a

matter of trial and error, where tolerances are re–allocated once a high rejection rate of parts is encountered. Hence, a mathematical formulation for the allocation of geometric tolerances is needed.

This goal is achieved in this dissertation, using the following approach:

1. Features in assemblies are assumed to have four types of geometric controls. These are: size, form, orientation and position. Except for controlling size, each control is assumed to have a finite set of discrete tolerance types for each feature. Form control can be achieved using straightness, cylindricity, flatness, circularity, etc.; orientation control can be achieved using perpendicularity, angularity and parallelism to one datum or another; and position control can be achieved by choosing one datum reference frame for position tolerance from a finite set of datum reference frames for each part. Each type of tolerance can have a range of possible magnitudes associated with it. Each feature is assumed to be produced by one manufacturing process chosen from a set of possible manufacturing processes, and each part can be produced using one manufacturing sequence selected from a set of possible manufacturing sequences.

2. An algorithm is developed for the optimal selection of a manufacturing process for each feature and a manufacturing sequence for each part, so that the production cost is minimized and the rejection rate is constrained to a certain predetermined value. Each manufacturing process and each manufacturing sequence is assumed to have a unique cost value, hence the total cost is the objective function to be minimized. The selected manufacturing sequences dictate the datum reference frames for the position tolerances.

3. The allocation of tolerances is formulated as a mixed programming optimization problem where each feature is associated with two discrete–valued variables and four real–valued variables. The discrete–valued variables are the types of controls on form and orientation (the type of position tolerance is determined in the previous step and size

control has one tolerance type, that is size tolerance). The real–valued variables are the magnitudes of the tolerances associated with each type.

4. The objective function to be minimized is the probability that a part will be rejected due to the violation of the specified geometric tolerances, while being acceptable to the assembly functional requirements or vice versa. This probability is called the probability of mismatch (**Chapter 7**). Genetic algorithms are used as function optimizers, to ensure the arrival at the globally minimum probability of mismatch (as well as the globally minimum cost in the previous step).

5. In order to evaluate the previous objective function, an algorithm is needed for the probabilistic analysis of geometric tolerances. The proposed analysis method is the Monte Carlo simulation aided with two variance reduction techniques. The surface of each feature is represented by a set of points. The coordinates of these points are the random variables. Since it is hard to obtain a closed form mathematical representation of the functional requirements as functions of the surface points, the simulation algorithm is proposed to simulate the manufacturing of parts, the inspection process and the assembly sequence. Then, the functional requirements and the geometric tolerances are checked in each simulation step.

6. In order to simulate the inspection process, the geometric deviations on the simulated manufactured parts have to be checked to determine whether they are within the tolerances selected by the optimization algorithm. A number of algorithms are proposed to evaluate these deviations using genetic algorithms to ensure arriving at the globally minimum deviations. In order to evaluate form deviations accurately, a parametric surface based on NURBS (Non–Uniform Rational B–Splines) is interpolated to the simulated surface points. This surface is used by the deviation calculation algorithms to evaluate form deviations.

1 .4 Contributions

The reported research makes the following contributions to the fields of tolerance analysis, tolerance synthesis and metrology:

1. A new criterion is developed for allocating geometric tolerances to satisfy functional requirements. The criterion is the minimum mismatch probability defined as the probability that the manufactured parts will be rejected during inspection while being acceptable according to the design functional requirements when assembled, or vice versa. The main motivation behind this criterion is that tolerancing to satisfy functional requirements has been recommended by many researchers without giving a mathematical formulation of such criterion.

2. Tolerance types are included for the first time in the allocation problem. A feature can be controlled for four geometric variations. These are: size, form, orientation and position. Each of these variations (except size) can be controlled by one of a number of tolerance types. For example a cylindrical feature's orientation can be controlled by parallelism to one datum or by perpendicularity to another datum. The choice of tolerance types has been absent from tolerancing research. The effect of choosing different tolerance types is investigated and a new formulation of the tolerance allocation problem is provided where tolerance types are taken into account.

3. An algorithm for tolerance analysis is developed using Monte Carlo simulation combined with variance reduction techniques. The algorithm reduces the number of samples needed for an accurate evaluation of the probability of assembly rejection.

4. Since it is difficult to define a closed form mathematical function relating the functional requirements to the imposed geometric tolerances, an algorithm is developed for 1) generating manufactured surfaces using the probability distributions of manufacturing

processes, 2) simulating the manufacturing sequence, 3) simulating the assembly sequence and 4) checking the conformance of the assembly to the imposed functional requirements and checking the conformance of the parts to the imposed geometric tolerances. The algorithm uses a graph representation for the assembly, where nodes are part features and links are the connections by which parts are fit together. A study of several methods for joining parts is given. A new algorithm for checking the conformance of clearances to their specified range is developed using genetic algorithms to search for the maximum and minimum values for the clearance.

5. A new set of algorithms is developed for evaluating the geometric deviations on manufactured parts. Genetic algorithms are used to find the minimum deviation zones to ensure that the global minimum values of the deviation zones are evaluated.

6. With the advent of the ANSI Y14.5.1M 1995 standards, some geometric deviations, such as the size tolerance, were re-defined. These deviations cannot be easily evaluated using the generated surface points alone. Hence, a new parametric surface interpolation method using NURBS was developed to approximate the actual generated surface. This method builds upon previous research in line interpolation to scattered points and extends it to surface patches. This surface is used to evaluate certain geometric deviations and to deduce the feature's actual axis.

1 .5 Overview of the Dissertation

The dissertation is divided into eight chapters and three appendices: Chapter one includes the motivation, research objective, approach and the list of contributions to the tolerancing field.

Chapter two presents a review of related research topics covering areas of tolerance representation in CAD, tolerancing theories, tolerance analysis, tolerance synthesis and

evaluation of geometric deviations from CMM data. It concludes by pointing out several key issues directly related to the research topic.

Chapter three describes a procedure for the interpolation of NURBS parametric surfaces to CMM data. The chapter gives an overview of NURBS, describes the interpolation method and concludes with examples.

Chapter four describes a set of algorithms for evaluating geometric deviations from CMM or simulated points. The chapter starts with a comparison of the results obtained from previous works with a formulation using genetic algorithms. It covers size, form, orientation and position deviation. An example with known deviations is used to verify the algorithms.

Chapter five describes algorithms for simulating machining and assembly sequences. It presents three methods for connecting parts to each other. It ends with an algorithm to evaluate the maximum and minimum clearances between holes and shafts.

Chapter six discusses the inclusion of variance reduction techniques with Monte Carlo simulation for tolerance analysis. The method is demonstrated with examples which include dimensional and geometric tolerances.

Chapter seven provides an algorithm for the optimal selection of manufacturing processes and manufacturing sequences to minimize the production cost. An example of a speed reducer is used to demonstrate the algorithms. The chapter discusses tolerancing cases where the choice of tolerance types affect assembly functional requirements. It then describes an algorithm for the optimal selection of tolerance values and types.

Chapter eight concludes the dissertation and provides suggestions for future research.

The dissertation has three appendices. Appendix A contains a description of using genetic algorithms for function optimization. Appendix B describes methods of building

geometric transformations. Appendix C describes methods of integrating functions with Monte Carlo simulation and variance reduction techniques.

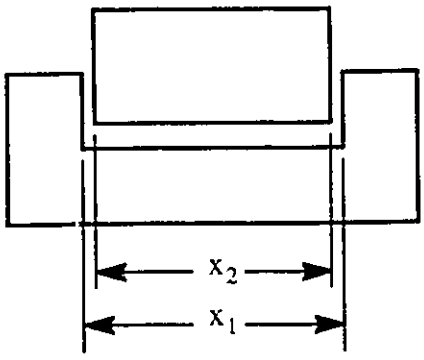
All procedures and algorithms were developed using C++ on a Sun Sparc workstation. The final runs were made on an SGI Crimson workstation. All graphs and surfaces were drawn using the MATLAB package.

CHAPTER TWO

LITERATURE SURVEY

The main research issues related to the selection of tolerances in assemblies are: tolerance analysis for calculating probabilities of assembly rejection, given the values and types of tolerances on the assembly features and tolerance synthesis for the optimal allocation of tolerances. Two other topics closely linked to either analysis or synthesis are computer modeling of tolerances and the evaluation of geometric deviations. The literature survey in this thesis covers previous work in the above mentioned topics and presents the research gaps which lead to the objective of this dissertation. The survey includes the analysis and synthesis of dimensional tolerances, then extends to work done on geometric tolerances.

2 .1 Tolerance Analysis



Functional Requirement:
$c = x_1 - x_2 \geq 0.003$

Design Function:
$F(\mathbf{x}) = x_1 - x_2 - 0.003$

Stack–up Condition:
$F(\mathbf{x}) \geq 0$

Figure 2 .1 An example of a Design Function

When designing an assembly designers specify restrictions on critical clearances, dimensions, etc. These restrictions are known as design functional requirements. Tolerance analysis involves the calculation of tolerance stack–up and design functional requirements,

which can be divided into two domains: 1) worst case analysis and 2) probabilistic analysis. Figure 2 .1 shows a simple peg and hole assembly. The limitation on the clearance c is the functional requirement. F(**x**) is called the design function and the vector **x** is known as the vector of design parameters.

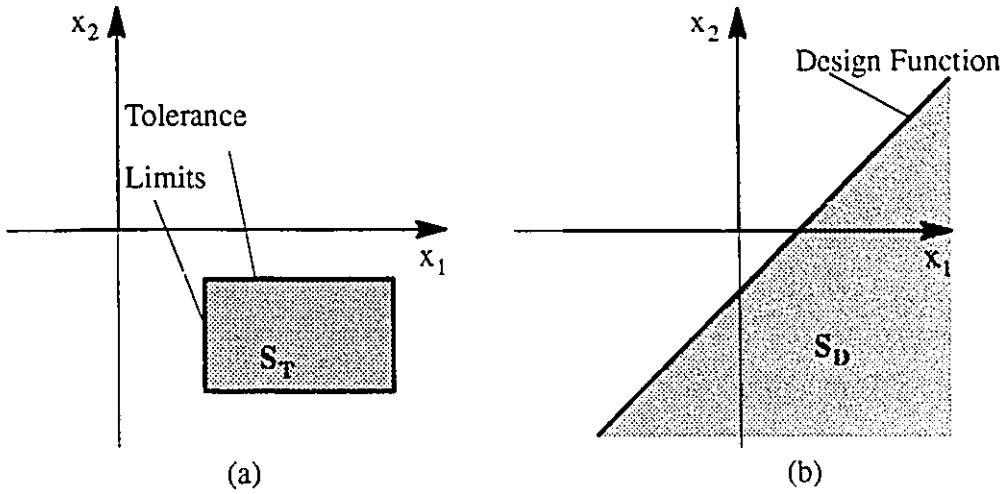
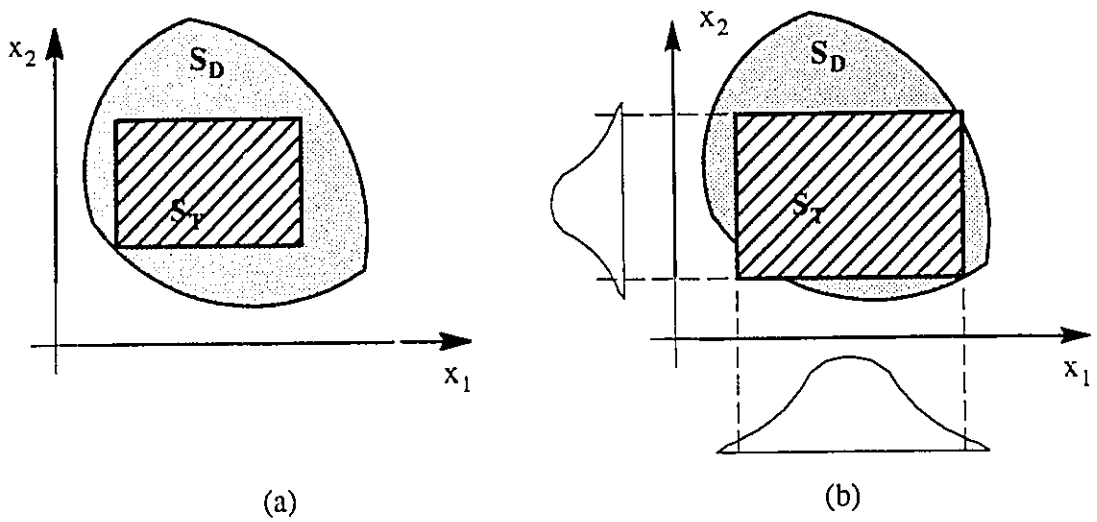


Figure 2 .2 Tolerance and Design Spaces

The tolerances and the design functions form two spaces over the design parameters domain. The first is the tolerance space (Figure 2 .2 .a) which refers to the bounded set of dimensions S_T acceptable according to the tolerance specifications. The second is the design space S_D (Figure 2 .2 .b) which refers to the dimensions acceptable according to the stack–up conditions. The worst case tolerance analysis checks if the specified tolerances satisfy the design function requirements (i.e. if S_T lies within S_D as in Figure 2 .3 .a). Since machining processes produce dimensions with variations that are random in nature, it is customary to associate with each dimension a probability density function representing the frequency with which that dimension is produced by the machining operation. In many cases this probability distribution function can be approximated by a normal distribution as stated by Mansoor (1963). Probabilistic tolerance analysis calculates the probability of satisfying the design function requirements (i.e. P(S_T lies within S_D) as shown in Figure 2 .3 (b). This

means the calculation of the multiple integral of the multivariate distribution function $f(\mathbf{x})$:

$$P(\mathbf{x} \in (S_T \bigcap S_D)) = \int_{\mathbf{x} \in (S_T \bigcap S_D)} f(\mathbf{x})\ d\mathbf{x} \qquad (2.1)$$



(a) (b)

Figure 2 .3 Tolerance Analysis

2 .1 .1 Linear Tolerance Analysis

The linear case of tolerance stack-up has been studied and discussed by several researchers (Bjorke, 1978, Evans, 1974 and 1975, Lee, W., 1988, Siddall, 1982 and Spotts, 1973). In all works tolerances were assumed to be independent random variables with normal distributions and equal to 6 times the standard deviation of the distribution. The

resultant clearance is equal to the root sum square of the component's tolerances:

$$t^2_{clearance} = \sum_{i=1}^{n} t^2_i \qquad (2.2)$$

2.1.2 Nonlinear Tolerance Analysis

Nonlinear tolerance analysis has been addressed using several methods which include: 1) Taylor series expansion, 2) Monte Carlo simulation, 3) quadrature techniques, 4) cell decomposition and 5) reliability index method. An extensive review of tolerance analysis methods is given by Wu et al. (1988).

2.1.2.1 Taylor Series Expansion

Given a design function $F(x_1, x_2, ..., x_n)$ where $x_1, x_2, ..., x_n$ are independent scalar random variables, the distribution of F is needed. If the moments of the distribution of F can be calculated, the distribution of F can be deduced. Taylor series expansion was used to expand the design functions around their mean values. The high order terms were then dropped. The method was investigated by Evans (1975) and Lee and Woo (1988). It was found to be ambiguous in some aspects, such as choosing the form in which the design function is expressed.

2.1.2.2 Quadrature Techniques

The quadrature technique is used to approximate the expected values of the moments of the design function distribution. Evans (1975) used the method and showed that its approximation is close to that of Taylor series.

2.1.2.3 Cell Decomposition

Michael and Siddall (1982) proposed the decomposition of the random variable space into orthogonal *n*–dimensional cubes. The probability was calculated simply by

summing up the probabilities of small cells dividing the feasible space. The method depends on examining the corner points of each cell, therefore in an n–dimensional problem the execution time needed is $O(2^n)$ and is thus limited to problems with a small number of tolerances.

2 .1 .2 .4 Reliability Index Method

The reliability index method converts the multivariate distribution into one univariate distribution. The method was discussed by Hasofer and Lind (1974) and Madsen et al. (1986), and applied to tolerance analysis by Lee and Woo (1988), Chase and Greenwood (1988) and Parkinson (1982,1984 and 1985). W.J. Lee (1988) tried tolerance analysis using the reliability index method and the Monte Carlo simulation, and the values he obtained for his examples results were close to those obtained by the Monte Carlo method. The advantage of the reliability index method is that it simplifies the calculation of the rejection probability in a single calculation without deviating from the correct value. Therefore, it is a powerful tool when used within a tolerance allocation algorithm decreasing the computation time significantly. Yet the method is limited to tolerances having normal probability distributions and to tolerances having differentiable limit functions for their functional requirements.

2 .1 .2 .5 Monte Carlo Simulation

Monte Carlo simulation is used in tolerance analysis where samples of the design parameters are generated; then for each generation, the design function is evaluated and checked for the stack–up condition. The process is repeated for a large number of samples. The method was used by Chase and Greenwood (1988) and H. ElMaraghy et al. (1991). The advantage of using Monte Carlo simulation is that it does not need a closed form mathematical function relating the functional requirements to the tolerances. It can be used

when the assembly is complex and it is difficult to evaluate the functional requirements except by simulating the assembly process. The drawback of the method is that it needs a large number of samples to evaluate the rejection probability accurately.

2 .2 Tolerance Synthesis

Tolerance synthesis has traditionally been concerned with allocating tolerance magnitudes to parts in order to minimize some criterion. Two criteria are frequently used for tolerance synthesis. These are: 1) the minimization of manufacturing cost, or 2) the minimization of the percentage of assembly rejections. In previous tolerance synthesis work, either one criterion was minimized and the other was used as a constraint on the minimization process, or vice versa. While the percentage of assembly rejections is the outcome of tolerance analysis, the determination of cost as a function of tolerance values is another area of research. Obsolete methods of tolerance allocation (1988) include allocation by proportional scaling and allocation by constant precision factor.

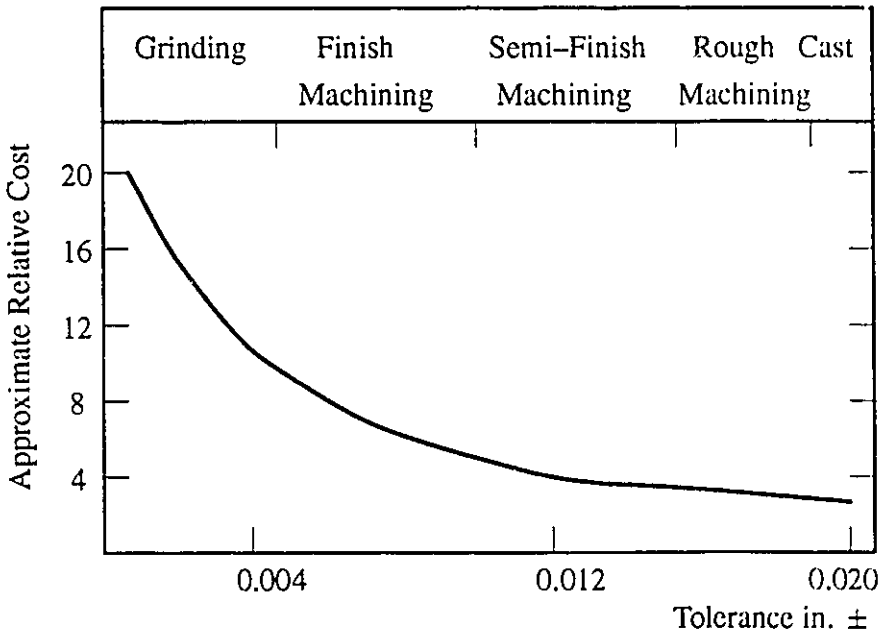


Figure 2 .4 Dieter's Cost–Tolerance Function (1983)

2 .2 .1 Cost–Tolerance Functions

Previous work in tolerance synthesis assumed cost to be a monotonically decreasing nonlinear function of the value of dimensional tolerances. An empirical curve (Figure 2 .4) for the cost of tolerance values was given by Dieter (1983). Many mathematical functions were proposed to fit manufacturing cost–tolerance data. Chase et al. (1990) made a survey of the proposed cost–tolerance functions. These models and the references in which they are found are given in Table 2 .1 . Wu et al. (1988) showed that Michael and Siddall's model (1981) is the best fit to Dieter's Curve. J.R. He (1991) proposed a cost function which combines the costs of material, scrap, rework, inspection and machining.

Cost Model	Author
$a + \frac{b}{t}$	Chase and Greenwood (1988)
$a + \frac{b}{t^2}$	Spotts (1973) and Parkinson (1985)
$a + \frac{b}{t^k}$	Sutherland and Roth (1975)
be^{-mt}	Speckhart (1972), Wilde and Prentice (1975)
$\frac{be^{-mt}}{t^k}$	Michael and Siddall (1981 and 1982)
$a_i - b_i t_i$	Bjorke (1978)
Discrete Points	Ostwald and Huang (1977) and Lee and Woo (1988)

Table 2 .1 Proposed Cost–Tolerance Functions

2 .2 .2 Models for Cost Minimization

Several optimization models have been proposed for tolerance synthesis. The main proposed models are:

2 .2 .2 .1 Lagrange Multipliers

Spotts (1973) used Lagrange multipliers to find a closed form solution for the tolerance allocation problem. He dealt with dimensional tolerances which were assumed to have normal distributions. Given the function:

$$L(\mathbf{t}) = \text{Cost}(\mathbf{t}) + \lambda \ \text{Constraint}(\mathbf{t}) \tag{2 .3}$$

where the cost function is:

$$\text{Cost}(\mathbf{t}) = \sum_i A_i + \frac{B_i}{t_i} \tag{2 .4}$$

and the constraint function is:

$$\text{Constraint}(\mathbf{t}) = \left| \sum_i t_i - t_{assembly} \right| = 0 \tag{2 .5}$$

The values of the tolerances corresponding to the minimum cost was found by deriving the Lagrange multiplier λ from equation (2 .3) as a function of individual tolerances then eliminating λ from equation (2 .3) and substituting for the component tolerances in the assembly tolerance equation:

$$t_{assembly}^{\ 2} = \sum_i t_i^{\ 2} \tag{2 .6}$$

2 .2 .2 .2 Geometric Programming

Wilde and Prentice (1975) treated the tolerance allocation problem as a geometric programming problem, which gave the same results as the Lagrange multipliers method. Their approach is restricted to exponential cost–tolerance functions.

2 .2 .2 .3 Discrete Optimization

Tolerance allocation was formulated as a discrete optimization problem by Ostwald and Huang (1977), who used the Balas zero–one method (1988) to solve the tolerance allocation problem. A more efficient method based on the Branch and Bound technique (1988), was developed by Lee and Woo (1988). Their method approaches the global minimum, but as the number of tolerances increases, the search space grows dramatically. The advantage of using a discrete optimization formulation of the synthesis problem is that each tolerance is associated with a specific manufacturing process and hence an optimal selection of manufacturing processes can be achieved along with the selection of tolerance values.

2 .2 .2 .4 Linear Programming

Patel (1980) and Ji (1993) formulated the allocation problem as a linear programming problem after linearizing the cost function and minimizing the total cost:

$$\text{Total Cost} = \sum_i a_i t_i + A \qquad (2.7)$$

The method is only applicable to problems where functional requirements have linear relations with the imposed tolerances.

2 .2 .2 .5 Nonlinear Programming

Nonlinear programming methods were used to allocate tolerances in several publications. (Balling et al., 1986, Bandler, 1976, Chase et al., 1990, Michael and Siddal, 1982 and Wu et al., 1988). In these methods a minimum is sought by iteratively testing local gradients and constraints. The methods of nonlinear programming used included Hooke–Jeeves (Reklaitis et al., 1983), Davidon–Fletcher–Powell (DFP) (Reklaitis et al., 1983) and Jackobson–Oksman method (Siddall, 1982). Dong and Soom (1990) used the

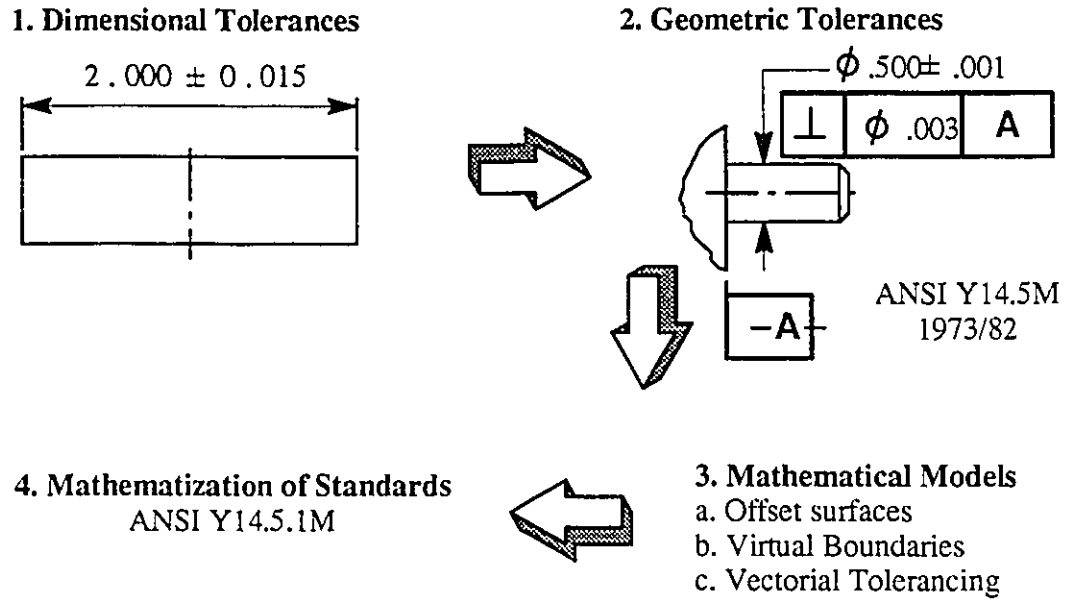
DFP method in conjunction with an assembly graph to allocate tolerances to multiple chains. Sequential quadratic programming was used by Parkinson et al. (1984) to allocate tolerances along with the selection of processes. The main drawback of nonlinear programming methods is that there is no guarantee that the global minimum will be reached.

2 .2 .2 .6 Stochastic Optimization Methods

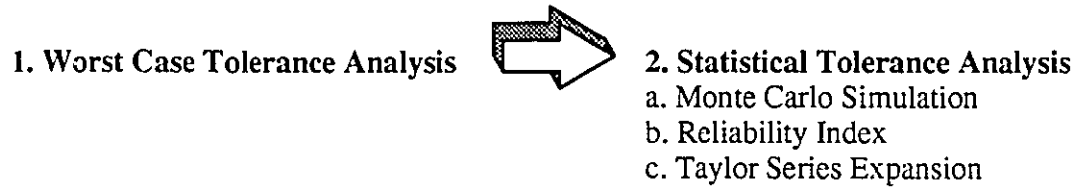
Methods of stochastic optimization were used for tolerance synthesis in a number of recent publications. This included the use of simulated annealing by Zhang and Wang (1993) and the use of genetic algorithms by Nassef and ElMaraghy (1993), Lee and Johnson (1993), Kanai et. al. (1995) and Iannuzzi and Sandgren (1995). Gadallah and ElMaraghy (1994) proposed a method for the discrete tolerance optimization with process selection based on experimental design techniques. These methods converge to global minima and hence became more widely used in recent tolerance synthesis literature.

2 .3 Computer Models of Tolerances

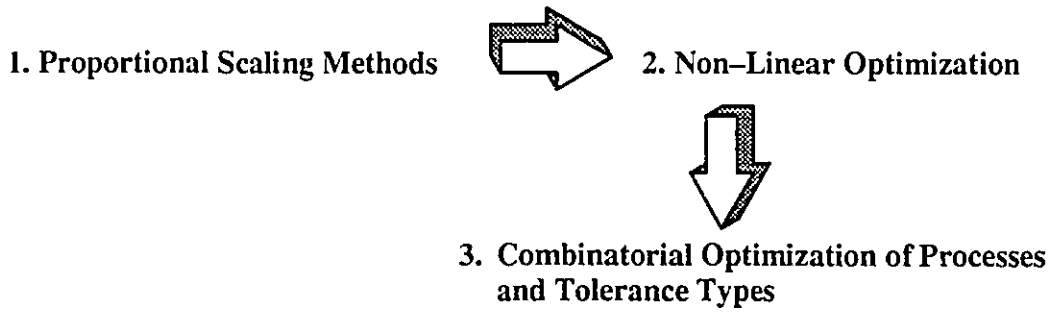
Tolerances were applied to mechanical design as a means of controlling dimensional and geometric variations. As mechanical designs became more sophisticated, the need for controlling parts' geometries evolved. This need led to the standardization of methods of controlling geometries by international standards organizations (ANSI Y14.5M 1982). The advent of computers encouraged some researchers to develop mathematical models and computer representations for tolerances and their stack-up. The logical and historical evolution of tolerances and their representations is shown in Figure 2 .5 .



(a) Tolerance representation



(b) Tolerance Analysis



(c) Tolerance Synthesis

Figure 2 .5 Evolution of Tolerancing Research

2 .3 .1 Mathematical Models of Tolerances

Much work has been done in developing mathematical models of geometric tolerances. The main reason for this research was to develop a mathematical representation of tolerances that is complete and unambiguous for computer applications. The main approaches used are described below:

2 .3 .1 .1 Offset Surfaces

Requicha (1983 and 1993) proposed a theory of tolerances based on offsetting surfaces in solid models. Each surface in a part is associated with a pair of offset surfaces that define its tolerance zone. He proposed combining all offset surfaces, thereby giving an overall tolerance zone within which an actual part should fall. The theory defines tolerance zones differently from the geometric tolerancing standards and treats tolerances as attributes to a part's surfaces. The ease of implementation is one advantage of this theory as pointed out by Juster (1992). Weill (1988) as well as Farmer and Gladman (1986) stated that this theory has major drawbacks and they showed that when the theory is applied to size tolerances it tends to reject acceptable parts.

2 .3 .1 .2 Virtual Boundaries

Jayaraman and Srinivasan (1989) addressed the issues of representing geometric tolerances in solid models from the functional requirements perspective. They introduced the concept of virtual boundaries, which are half-spaces that separate every pair of contacting surfaces in the mating parts. They used the virtual boundaries to convert the assembly requirement (that is the mating of two parts) into a set of requirements for each of the individual mating parts. To achieve this goal they defined a requirement known as the virtual boundary requirement where two mating features of size should have the same virtual boundary and each on one side of the boundary. They used these requirements as a tolerance

specification method. However, Turner (1993a) stated that their approach does not reflect design function requirements, which can be imposed independently on a feature's size, form, orientation and position.

2 .3 .1 .3 Vectorial Tolerancing

Wirtz (1991) as well as G. Turner and Anderson (1988) presented an approach in which tolerances constrained vector components relating a toleranced feature to another datum feature (Figure 2 .6). Their dependence on reference features made their approach unsuitable for form tolerances.

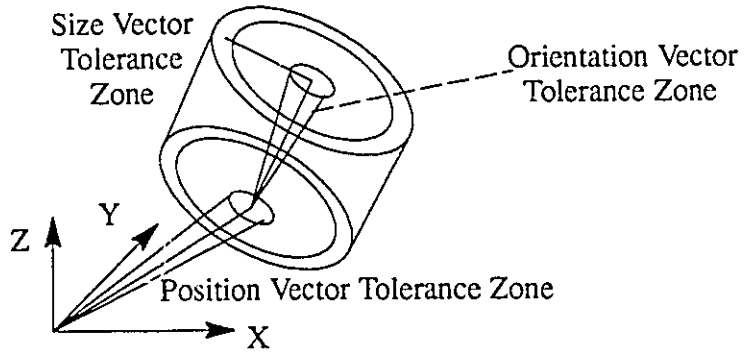


Figure 2 .6 Vectorial Tolerancing (Cited from Voelker, 1993)

2 .3 .1 .4 Feasibility Space

Turner (1988 and 1993) defined each allowable variation in a surface of the solid model as a model variable, which he defined as a real–valued measure of the extent to which a geometric property varies from normal. Using these model variables as well as tolerances as constraints on model variables he constructed a feasibility space within which an acceptable part should lie. The model variables were associated with edges and vertices for location and orientation tolerances, and he proposed the use of Bezier patches for form tolerances. Details about his use of Bezier patches are given by Guilford and Turner (1993). This approach was the nearest to the specifications given by the ANSI standards (1982).

2 .3 .1 .5 Remarks

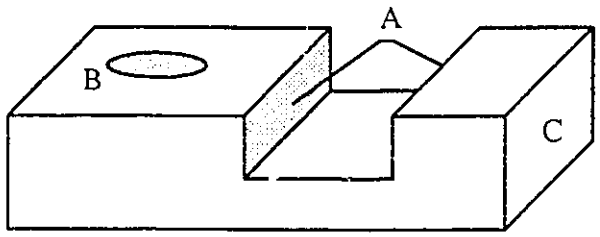
Although the issue of making mathematical models of tolerances is not resolved, the above models present insights into current tolerance practices, point out the ambiguities in the tolerancing standards and lead to the mathematization of tolerancing standards and to the evolution of the ANSI Y14.5.1M standards, where mathematical definitions of the different geometric tolerance zones, actual values and conformance of the manufactured parts to the specified tolerances are presented in unambiguous mathematical formulae.

2 .3 .2 Computer Representation Issues

The research done in the development of a mathematical theory of tolerances was paralleled by another closely-related area of research. That is representing tolerances in solid models, to be used in various applications such as checking conformance of parts to specified tolerances or tolerance analysis. Requicha et al. (1986) and Rossignac (1986) implemented their proposed offset surfaces theory within the CSG-modeler PADL-2. Light and Gossard (1982) proposed a scheme for modeling an object with its dimensions and tolerances based on variational geometry. Their implementation was limited to dimensional tolerances. Their work was extended to tolerances on surfaces by the work of Gossard et al. (1988), where a hybrid CSG/BREP modeler was proposed and where tolerance constraints were represented by a set operator that uses a scalar value to move one surface relative to another. A similar work by Roy and Liu (1988) proposed an assembly model based on a hybrid CSG/BREP structure, where tolerance information is associated with each face of the boundary representation. Bernstein and Preiss (1989) proposed a directed graph to represent dimensions in solid modelers, where each node on the graph represents a face of the object, and a transformation matrix that constrains the number of degrees of freedom of one face relative to the other is attached to the arc. The position of one face relative to another is found by multiplying the transformation matrices of the nodes in the path connecting the two nodes

representing the mentioned faces. Their work was extended by Cardew–Hall et al. (1993) to include tolerance matrices which allow the degrees of freedom of each surface to vary within limits. Juster (1992) argued that this method departs from the standards in that all dimensions propagate from a datum reference frame. Fleming (1988) proposed a procedure in which a toleranced part can be represented as a network of tolerance zones and datums connected by arcs to which inequality constraints are attached. A discussion of the general framework for representing ANSI Y14.5M standards (1982) was given by Shah and Miller (1990) as well as Guilford and Turner (1994). The above representations concentrated on the CAD representation issues and tolerances were represented by offset distances attached to each face in the boundary representations in a manner similar to the theory proposed by Requicha (1986) and Rossignac (1986). Research is still needed in this area to represent tolerances in CAD models which conform to the ANSI tolerancing standards (1982). Further details of representation issues can be found in the works of Juster (1992) and Roy et al. (1991).

2 .4 Review of the ANSI Y14.5M Standards



A is a pair of parallel planes
B is a cylindrical feature
C is a surface

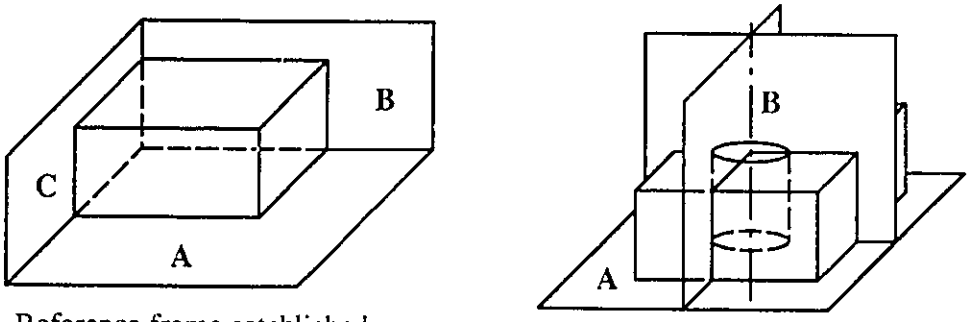
Figure 2 .7 Features of Size

So far, this review has covered issues related to dimensional tolerancing. Geometric tolerancing practice started as early as the 1960s and since then several standards for geometric tolerancing have been released. The following review is a brief description of the

ANSI Y14.5M (1982) standards. Standards divide part features into two types: 1) features of size which are cylindrical features, spheres or any set of parallel planes in a component and 2) surfaces. Features of size are controlled for size, form, orientation and position deviations, whereas surfaces are controlled for orientation and position deviations only.

2 .4 .1 Datums and Position Tolerance

The standards also provide guidelines for choosing datums. A datum is defined as a theoretically exact point, axis or plane derived from a true counterpart of a specified feature. For example if a cylindrical hole axis is specified as a datum its true counterpart on the manufactured part is the maximum inscribed cylinder within the hole, and the datum would be the axis of the inscribed cylinder. It is the origin from which the location or geometric characteristics of features are measured. The standards specify that a part should have at least one datum reference frame of points, axes and/or planes defining three mutually perpendicular planes (Figure 2 .8) to establish a reference frame from which position tolerance zones are located.



Reference frame established by planes **A,B,C**

Reference frame established by plane **A** and axis **B**

Figure 2 .8 Datum Reference Frames

2 .4 .2 Orientation and Form Tolerances

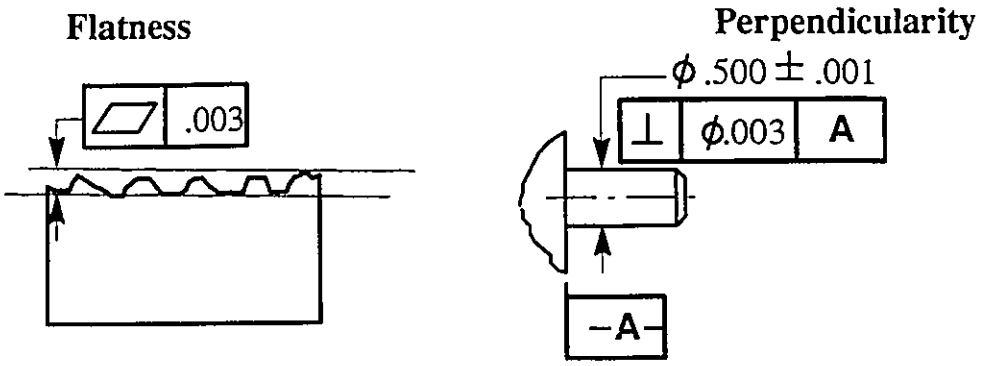


Figure 2 .9 Orientation and Form Tolerances

The standards then specify straightness, circularity, cylindricity and flatness as methods of controlling form. Orientation of features is controlled by parallelism, perpendicularity, angularity and runout. Position control is achieved by position tolerance. Both orientation and position tolerances are specified relative to datums. Figure 2 .9 shows some examples of orientation and form tolerances.

2 .4 .3 Size Tolerance

The standards specify a rule for size tolerances known as the envelope rule. A size tolerance is specified on a dimension of a feature of size. The rule specifies that the feature surface should be within an envelope of the maximum material condition (MMC) and that the size of the feature at any cross section should be within the size limits specified by the tolerance value. When the size tolerance does not provide enough form control on the feature, form tolerances can be applied to provide additional control.

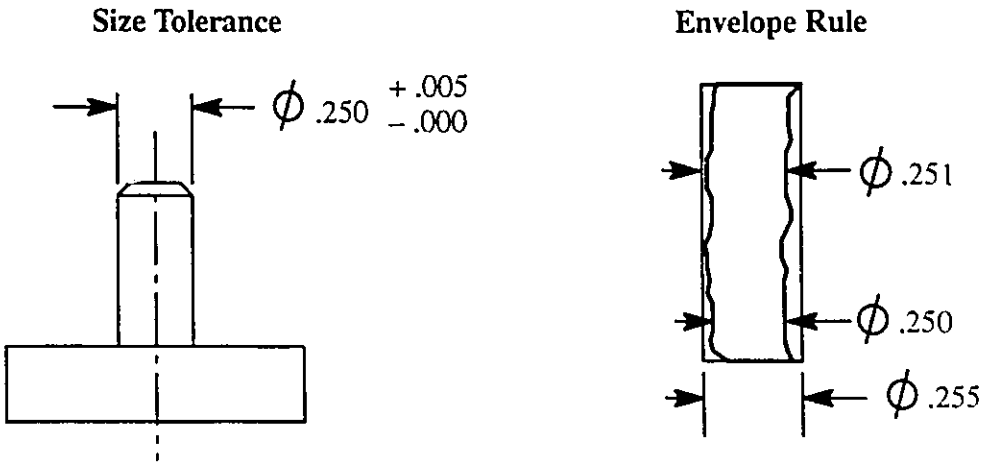


Figure 2 .10 Envelope Rule for Size Tolerance

2 .4 .4 ANSI Y14.5.1M Standards

The ANSI Y14.5M had some ambiguities concerning size tolerance as well as datum establishment. This lead to the release of the 1994 ANSI Y14.5M. Accompanying these new standards is a mathematical definition of the geometric tolerances standards (ANSI Y14.5.1M standards, 1995), as well as the B89.3.2 *Dimensional Measurement Methods* (to be released in 1996). Suresh and Voelker (1994) discussed the interpretation of the size tolerance in the new standards. Srinivasan and O'Connor (1994) addressed the inclusion of statistical symbols in the standards to produce features with statistical process control. They proposed including their work in the next standards release. The new mathematical standards give precise mathematical functions and/or objective functions to evaluate tolerance zones and to evaluate the conformance of a part to a tolerance zone.

2 .5 Evaluation of Geometric Deviations in Machined Parts

This section reviews the work done in the area of checking geometric deviations on machined parts. Surfaces obtained by machining processes deviate from the nominal

geometry therefore, the specified tolerances need to be verified. Traditionally functional gages have been used to check for geometric deviations. With the advent of coordinate measuring inspection machines (CMM) a new kind of data was obtained from the inspection process. These are a set of points sampled from the surface of manufactured parts. The sampled points need a number of algorithms to evaluate the various geometric deviations, as demonstrated by the following example. Figure 2 .11 shows a set of measured points and the minimum deviation zone for straightness. The problem can be viewed as fitting an ideal surface to a set of measured points. Given a set of measured points $\mathbf{P}_i$ and an equation for an ideal surface **S**, the objective of the fitting is to find the orientation of **S** which will minimize the distance function L:

$$L = \left[\frac{1}{n}\sum_{i=1}^{n} e_i^p\right]^{1/p} \qquad (2.8)$$

where:

e is the normal distance between **S** and $\mathbf{P}_i$

p is an exponent

n is the number of measured points

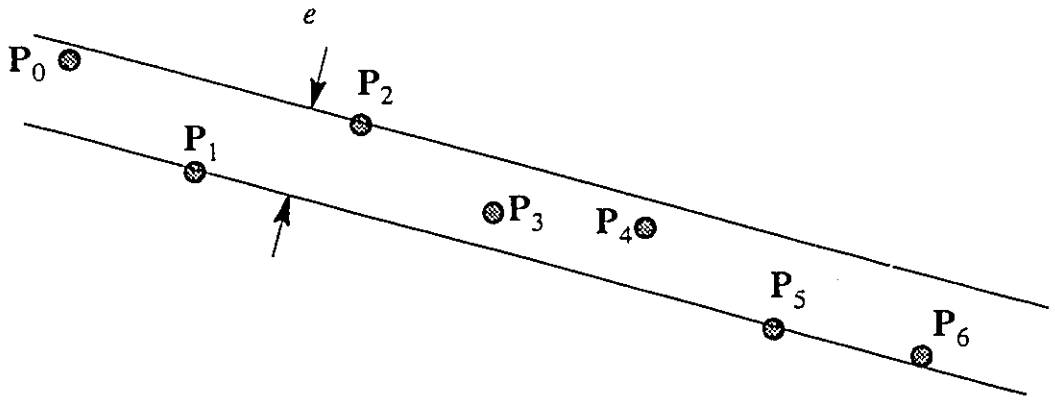


Figure 2 .11 Minimum Deviation Zone For Straightness

The exponent p in the distance function determines the type of fit. If p is equal to 2, then the fit corresponds to that of a least squares. If p is equal to ∞, then the distance function becomes:

$$L = \max_i(e_i) \tag{2.9}$$

The latter form of the distance function is the one used to find minimum geometric deviation zones. There is a tradeoff between the accuracy of the fit and its sensitivity to measurement errors, which might lead to more complex methods for estimating the exponent p of the distance function. This research is beyond the scope of this thesis but a good review of the problem is given by Hopp (1993). The following sections include a review of the methods used in evaluating the minimum zone deviations.

2 .5 .1 Monte Carlo Method

Murthy and Abdin (1980) assumed that the minimum deviation zone mean surface lies close to the least squares mean surface, and within the least squares deviation zone. The minimum deviation zone parameters are then randomly selected until they satisfy the minimum zone conditions.

2 .5 .2 Spiral Search

The spiral search is a method used to find a global minimum for two or three variables. The search is conducted in a spiral manner starting from the least squares minimum. Murthy and Abdin (1980) used the spiral search, compared its results with the Monte Carlo method, and found it to be more time-efficient.

2 .5 .3 Simplex Search

Simplex search starts from the minimum obtained by the least squares and moves iteratively to the minimum zone with the aid of a simplex triangle. This method was used

by Murthy and Abdin (1980) who found its results comparable with the spiral search and by Shunmugam (1991) and Dhanish (1991), who used it in conjunction with a linear Chebyshev approximation of the actual surface.

2 .5 .4 Nonlinear Programming

Methods of nonlinear programming were used for the evaluation of minimum zone deviations, by W. ElMaraghy et al. (1990) and Yu (1992). The drawback of nonlinear programming methods is that there is no guarantee of converging to the global minimum.

2 .5 .5 Genetic Algorithms

Genetic algorithms have been used by Nassef and H. ElMaraghy (1994) to evaluate the values of the actual geometric deviations using CMM data. The results showed that the methods which depended on the gradient descent search lingered in local minima and the deviations obtained by using genetic algorithms were less than those obtained by other methods. This has the advantage of avoiding the rejection of good parts. A full detail of the method is reviewed in Chapters 3 and 4.

2 .5 .6 Voronoi Diagrams

Methods of computational geometry have been applied to the evaluation of size and perpendicularity deviations by Jackman et al. (1994). They constructed Voronoi diagrams using the measured points to evaluate the centre of the largest inscribed cylinder contained within the measured points. A Voronoi diagram (1985) is a planar polygon whose edges are the loci of proximity of a set of points. A locus of proximity of a point p is the locus of all points (x,y) that are closer to p than any other point in the plane. Thus given a set of points measured on a cylinder, the largest inscribed cylinder within those points and perpendicular to a certain plane (which is equal to the perpendicularity deviation) has its centre at one of the vertices of the Voronoi diagram. The advantage of this method is that it gives the exact

value of the geometric deviation in $O(N\log N)$ time where N is the number of points. The method's use has been limited to the evaluation of position and perpendicularity deviations.

2 .6 Issues Related to Research Motivations

Section **1 .2** , chapter one, stated the motivations behind the research presented. This section reviews the work related to these motivations and shows the gaps that the dissertation addresses.

Little research has been conducted in the area of allocation of geometric tolerances to assemblies since it started in the late eighties. Research in this area is confronted with the following problems:

1. What criteria would be used to allocate geometric tolerances ?
2. Since a single geometric variation can be controlled by several tolerance types (for example a cylinder's variation in form can be controlled by any of the following tolerances: cylindricity, circularity or straightness), how can the tolerance types be included in the tolerance allocation algorithm ?
3. If rejection rates of assemblies are to be evaluated, what method of tolerance analysis would be the most appropriate for this task ?
4. Would geometric tolerances be stacked–up in the same manner as dimensional tolerances ?
5. What algorithms would be used to check the conformance of parts to the allocated tolerances and how would the functional requirements be checked ?

2 .6 .1 Criteria for Tolerance Allocation

The most widely used criterion (Chase et al., 1990, Lee, W., 1988 and Wu et al., 1988) for allocating dimensional tolerances is the minimum manufacturing cost. Hence, the

tolerance allocation (synthesis) problem was formulated as an optimization problem where the magnitudes of the tolerances were the independent variables and the total cost is the objective function. Another criterion that was used to allocate tolerances is the minimum rejection rate.

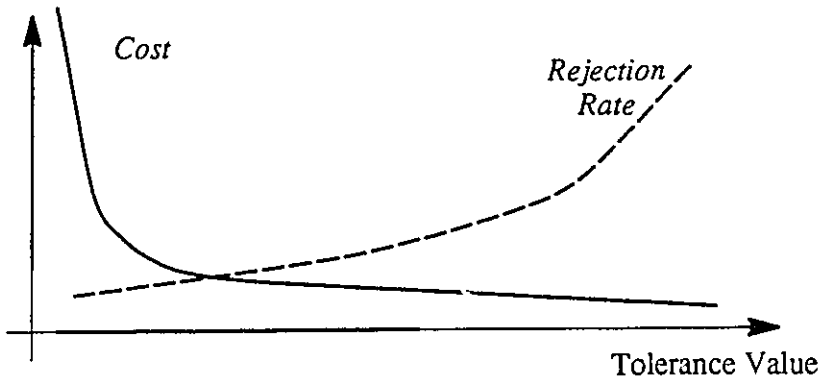


Figure 2 .12 Tradeoff between Cost and Rejection Rate

Figure 2 .12 shows the tradeoff between the cost and rejection rate over the tolerance values. Loose tolerances decrease production cost, but causes the rejection rate to increase (Lee, W., 1988). This tradeoff was solved by either:

1. Minimizing the cost and constraining the rejection rate, or

2. Minimizing the rejection rate and constraining the cost.

When researchers in tolerance synthesis considered to the allocation of geometric tolerances, cost was again used as a criterion (ElMaraghy et al., 1991), where a cost vs. tolerance values function is associated with every type of allocated tolerance. However, it is impractical to obtain a cost tolerance function for each type of tolerance imposed on a single feature. A discrete cost vs. tolerance value model has been proposed by Lee and Woo (1988) for dimensional tolerance analysis, where the cost is associated with the limits of the manufacturing processes and the tolerance values are set to be equal to the processes limits. This model was used by Nassef and H. ElMaraghy (1993) and Kanai et al. (1995) to allocate

geometric tolerances. This approach assumes that equal values of several tolerance types have the same cost. For example, a cylindricity tolerance value of 0.003" would have the same cost as of a position tolerance value of 0.003". Farmer and Gladman (1986) state that geometric tolerances should be allocated to assembly features in order that the design functional requirements not be violated. Hence, those parts which are not rejected during assembly should not cause a violation of the assembly functional requirements once they are assembled to the other parts. The minimum cost does not guarantee such functioning, and the tolerancing research lacks a mathematical formulation of a criterion that would guarantee that tolerances are allocated such that parts accepted during the inspection process are the ones that satisfy the functional requirements and parts rejected during the inspection process are the ones which will cause a violation of the functional requirements when assembled to other parts. A formulation of such a criterion is given in chapter seven.

2 .6 .2 Choice of Tolerance Types

According to the ANSI Y14.5M standards (1982), a feature of size has four geometric variations that need to be controlled. These are size, form, orientation and position. Each of these variations can be controlled with a number of tolerance types as specified by the standards. The proper choice of tolerance types is missing in the tolerancing research. The inclusion of tolerance types in the allocation problem is addressed in chapter seven.

2 .6 .3 Analysis of Assemblies with Geometric Tolerances

Previous work in tolerance analysis of assemblies having geometric tolerances used either of two methods: 1) An approach based on the concept of feasibility space, where a method of representing boundaries on model variables was proposed by Turner (1993a) and Gupta and Turner (1993). Their work was limited to the evaluation of worst case tolerance

analysis and did not include the statistical analysis of geometric tolerances and 2) Monte Carlo simulation was used by H. ElMaraghy et al. (1991). Although Monte Carlo simulation does not need a closed form mathematical representation of the analysis problem, its main drawback is that it needs a large number of samples to get accurate results. This problem is addressed in chapter six, by the use of variance reduction techniques.

2 .6 .4 Tolerance Stacking

Researchers in the analysis of dimensional tolerances developed closed form mathematical functions relating the functional requirements to the imposed tolerances. It is difficult to develop the same closed form mathematical tolerance chains for assemblies with geometric tolerances imposed on their features. Hence, Monte Carlo simulation becomes the only possible method for analyzing geometric tolerances, where samples of the manufactured features are generated, the assembly process is simulated (with the generated parts), then the conformance of the functional requirements to their designed limits is checked. This conclusion was pointed out by Turner (1993b) in his review of tolerancing packages currently available. The next question that arises is, how will the generated parts be positioned in space according to their joining method? Little work has been done in the area of modelling the assembly of manufactured parts to each other. Turner (1991) used linear programming to find the optimal configuration of manufactured parts. Methods of connecting parts together include shrink fits, bolted fasteners, location fits, etc. Some of these connections are tackled in chapter five in addition to the simulation of the assembly sequence.

2 .6 .5 Checking Geometric Deviations

The checking of geometric deviations on generated surfaces is needed for the analysis of assemblies with geometric tolerances. Research in this area has been covered in

section 2 .5 . However, there are some issues that still need to be addressed. The first is that most of the algorithms used do not guarantee the evaluation of the global minimum deviation zone, hence a global optimization method is needed. This issue is addressed in chapter four. On the other hand the new ANSI Y14.5.1M standards (1995) define some geometric deviations, such as the local size, in a way that is not easy to calculate directly from the generated (or measured) points. Thus a surface representing the actual manufactured surface needs to be interpolated to the generated points. This issue is addressed in chapter three.

2 .7 Tolerancing Packages

Several commercial tolerancing packages had been developed by the late eighties. These packages are intended for tolerance analysis and do not support tolerance magnitude allocation and all of them use Monte Carlo simulation for the analysis. Furthermore none of these packages support tolerance types allocation. Turner (1993b) reviewed these packages and divided them into three categories: (1) Procedural–based packages, such as the point–based VSA package, which depend on a set of 3D points to determine the location of a toleranced feature, (2) dimension–driven packages that analyze 2–D dimensional sketches such as SDRC–IDEAS tolerance module, Anlytix and Mechanical Advantages packages and, (3) packages that facilitate the analysis of a few common special cases like the VALYSIS package which is designed to analyze the mating of two parts. Turner (1993b) concludes his review by stating that these packages may be suitable according to the tolerance problem solved and that none of the current packages can substitute the human insight. The development of tolerancing software is conducted in academic circles as well. The GEOS system (1993b) uses the feasibility space approach proposed by Turner (1993a). The package has the capability of analyzing form tolerances in addition to size, location and orientation tolerances, but on the other hand it is limited to the worst–case analysis. The TASS system (1991) uses Monte Carlo simulation to analyze assemblies with geometric

tolerances. It supports the simulation of assembly sequences, but does not address form tolerances.

CHAPTER THREE

INTERPOLATION OF MEASURED SURFACES

The process of tolerance allocation described in subsequent chapters utilizes a routine which checks the conformance of the generated features to the allocated tolerances. The generated surfaces are represented by a set of points. These points are equivalent to points obtained from Coordinate Measurement Machines (CMMs) in the actual manufacturing. A feature may have to be checked for size, form, orientation and position deviations. In the past, the evaluation of geometric deviations (ElMaraghy et al., 1990 and Yu, 1992) used the measured points to evaluate the deviations. With the advent of the new ANSI Y14.4.1M standards, some of the form tolerances, as well as the size tolerance, are difficult to evaluate without a surface representation of the actual surface. This chapter presents a procedure for creating parametric surfaces passing through measured (or generated) points, using the non-uniform rational B-Spline (NURBS) parametric representation of curves and surfaces.

3 .1 Motivation

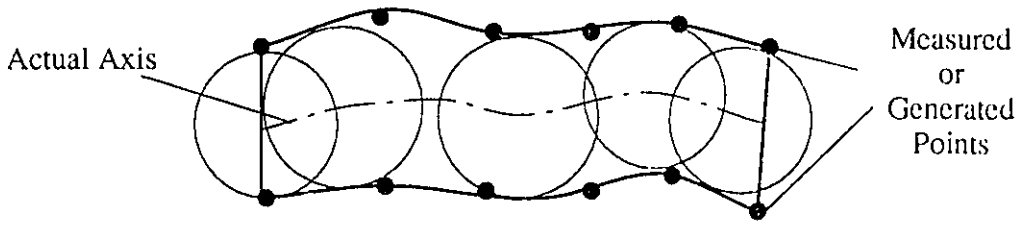


Figure 3 .1 Deriving a feature's axis

Figure 3 .1 shows an exaggerated view of a manufactured cylindrical feature with a set of points either generated by simulation or obtained from CMM measurements.

Srinivasan (1993) describes local size of the feature at any cross–section to be equal to the diameter of the largest inscribed sphere within the feature at that cross–section. The derived sphere's centre–point is a point on the feature's actual axis. When a tactile probe CMM is used, the number of sampled points is relatively small and the evaluation of the sphere's position and radius would be impossible without a representation approximating the actual surface and filling in the gaps between the available points. At the same time, no tolerance associated with the feature's axis can be evaluated without finding the sphere's centre point. The necessity of a parametric surface interpolated through the generated point is not limited to the evaluation of the geometric deviations, but is needed for checking critical clearances in assemblies as shown in **Chapter 5**. Since the purpose of the presented method is to help in the evaluation of geometric deviations on surfaces generated by simulation (**Chapter 5**) the measurement errors (i.e. errors inherent to the CMM probe, geometry of the CMM moving elements, thermal errors, etc.) are not taken into account.

3 .2 Choice of Representation

The surface representation will be used to evaluate geometric deviations on features of size. As defined by the ANSI Y14.5M (1982) standards, features of size comprise three shapes: cylinders, pairs of opposite and parallel planes, and spheres. Hence, the surface representation would be used to approximate a slightly deformed cylinder, plane or sphere. Other types of surfaces are generally controlled by profile tolerance measured from a specified datum. They do not need a surface representation of their manufactured surface and their deviations can be directly evaluated using the measured points. Literature on geometric modelling includes several methods for interpolating a parametric surface through a set of points. However, the representation to be chosen must have the following characteristics: (1) The surface representation needed should be capable of representing the ideal feature (i.e. capable of representing planes, cylinders and spheres) as the deviations are small relative to

the feature's dimensions. (2) The surface representation should have a local variation property such that each measured point affects a local patch of the surface. The later criterion is satisfied by Bezier surfaces, B–Splines and Non–Uniform Rational B–Splines (NURBS) (Mortenson, 1985), but the first criterion is only satisfied using NURBS, so it is chosen as the method for surface interpolation.

3 .3 A NURBS Review

A NURBS is a parametric representation of a curve or a surface. A curve is defined by an approximation polygon (control polygon), the corner points of which are known as control points (Figure 3 .2). Each control point $\mathbf{P}_i$ is associated with a positive real number, known as weight w_i , which pulls the curve towards, or pushes it away from, the control point. The NURBS equation is a function of an independent parameter u which ranges from zero to one.

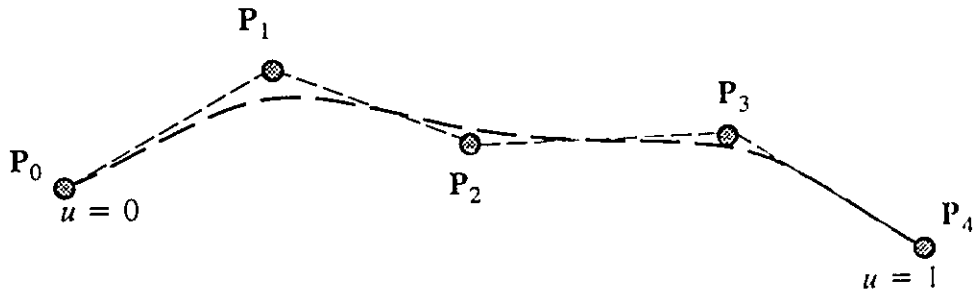


Figure 3 .2 A NURBS Curve

To derive a NURBS representation of a curve (Piegl and Tiller, 1987, and Piegl, 1991) let:

$$\mathbf{T} = \{t_0, t_1, \ldots, t_i, t_{i+1}, \ldots, t_m\} \qquad (3.1)$$

be a non–decreasing real numbered sequence (known as the knot vector), dividing the independent parameter's interval into knot spans [t_i , t_{i+1}). For every control point $\mathbf{P}_i$ there is a corresponding B–Spline function (de Boor, 1978) $N_{i,p}$ (where p is the curve's degree) defined recursively as :

$$N_{i,p}(u) = \frac{u - t_i}{t_{i+p} - t_i} N_{i,p-1}(u) + \frac{t_{i+p+1} - u}{t_{i+p+1} - t_{i+1}} N_{i+1,p-1}(u) \qquad (3.2)$$

$$N_{i,0}(u) = \begin{cases} 1 & \text{if } u \geq t_i \ \& \ u < t_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

The NURBS equation is then given by:

$$\mathbf{P}(u) = \sum_{i=0}^{n} R_{i,p}(u) \ \mathbf{P}_i \qquad (3.3)$$

where n = number of control points + 1, and $R_{i,p}(u)$ is known as the rational basis function given by:

$$R_{i,p}(u) = \frac{w_i \ N_{i,p}(u)}{\sum_{i=0}^{n} w_i \ N_{i,p}(u)} \qquad (3.4)$$

The rational basis function is non–zero in the interval [t_i , t_{i+p+1}), and thus affects the curve equation (3 .3) in a local segment of the curve. The shape of a NURBS curve depends on the control point coordinates, the weights, the knot vector and the degree of the curve.

3 .3 .1 Conics

A conic curve is a curve interpolated to its end points, the tangents at its end points and an intermediate point all lying in the same plane (Mortenson, 1985). Sets of conic

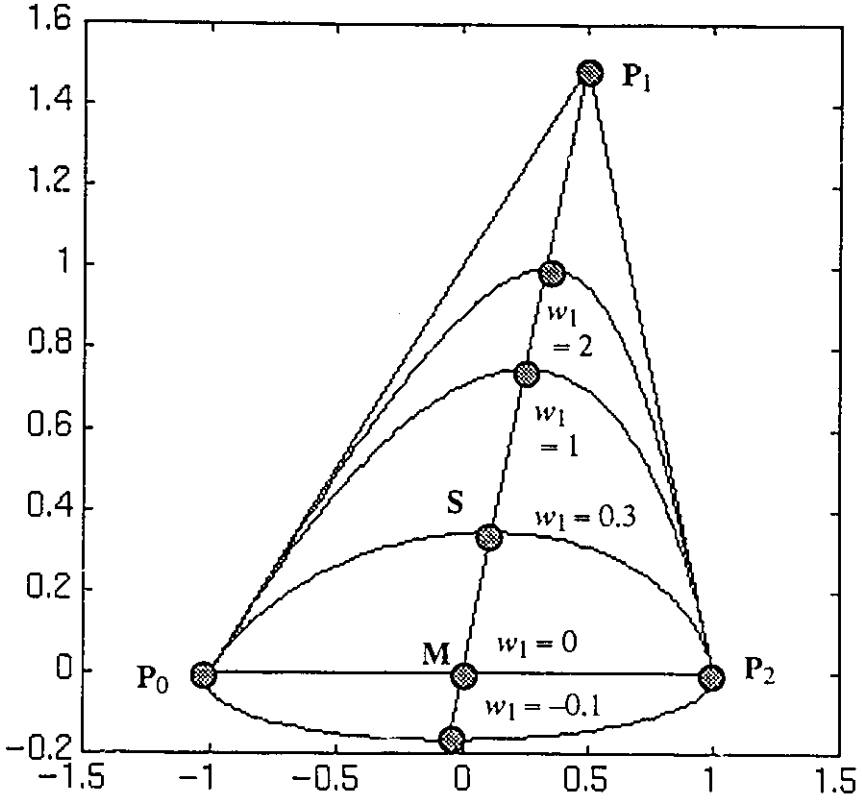


Figure 3 .3 NURBS Representation of a Conic Arc

sections have been used to smooth scattered data (Piegl, 1987). In the later sections conics will be used for interpolating curves to generated points. A conic can be represented by a rational B–Spline equation:

$$\mathbf{C}(u) = \frac{N_{0,2}(u)w_0\mathbf{P}_0 + N_{1,2}(u)w_1\mathbf{P}_1 + N_{2,2}(u)w_2\mathbf{P}_2}{N_{0,2}(u)w_0 + N_{1,2}(u)w_1 + N_{2,2}(u)w_2} \qquad u \in [0, 1) \qquad (3.5)$$

where the knot vector is given by:

$$\mathrm{T} = \{0, 0, 0, 1, 1, 1\} \qquad (3.6)$$

Equation (3 .5) defines a family of conics (ellipses, hyperbolas and parabolas). Each conic passes through the end points $\mathbf{P}_0$ and $\mathbf{P}_2$ (Figure 3 .3), and is tangent to the chords

$[\mathbf{P}_2,\mathbf{P}_1]$ and $[\mathbf{P}_0,\mathbf{P}_1]$. Faux and Pratt (1979) showed that the value w_0w_2/w_1^2 determines the type of the conic, not the individual values of w_i. Thus if the weights of the end points are fixed:

$$w_0 = w_2 = 1 \tag{3.7}$$

then the value of w_1 will determine the type of the conic curve. Table 3.1 shows the different types of conic sections determined by the value of w_1.

Range of w_1	Curve Type
$w_1 > 1$	Hyperbola
$w_1 = 1$	Parabola
$0 < w_1 < 1$	Ellipse
$w_1 = 0$	Straight Line
$w_1 < 0$	Complementary Arc

Table 3.1 Types of Conic Sections

Faux and Pratt (1979) showed that the value of w_1 is determined by the ratio

$$w_1 = \frac{\mathbf{MS}}{\mathbf{SP}_1} \tag{3.8}$$

where:

$\mathbf{S} = \mathbf{C}(0.5)$ and $\mathbf{M}$ is the midpoint of the cord $[\mathbf{P}_0, \mathbf{P}_2]$. Equation (3.8) can be used to derive the value of w_1 for a circular arc (Figure 3.4).

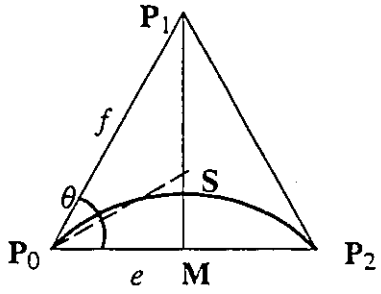
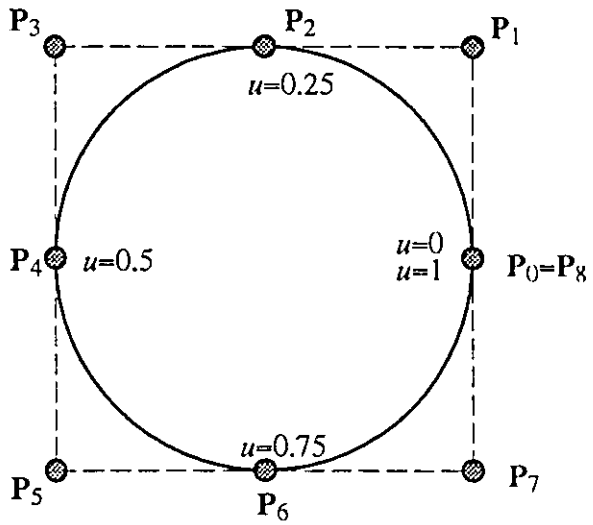


Figure 3 .4 A Conic Circular Arc

With some algebraic manipulation, equation (3 .8) can be reduced to :

$$w_1 = \cos(\theta) \tag{3 .9}$$



weights = { 1 , 0.7071 , 1 , 0.7071 , 1 , 0.7071 , 1 , 0.7071 , 1 }

T = { 0 , 0 , 0 , 0.25 , 0.25 , 0.5 , 0.5 , 0.75 , 0.75 , 1 , 1 , 1 }

Figure 3 .5 Representation of Circle

A circle can be represented by a repetition of conic sections. Figure 3 .5 shows a circle represented by second degree NURBS curve along with its control points, weights and knot vector. The range of the independent parameter u covered by each rational basis function is shown in Figure 3 .6 . It is obvious that each function covers a local span of the independent parameter u and thus has a local effect on the curve. This property will be used in surface interpolation to the generated points, where each point will affect a local span of the surface.

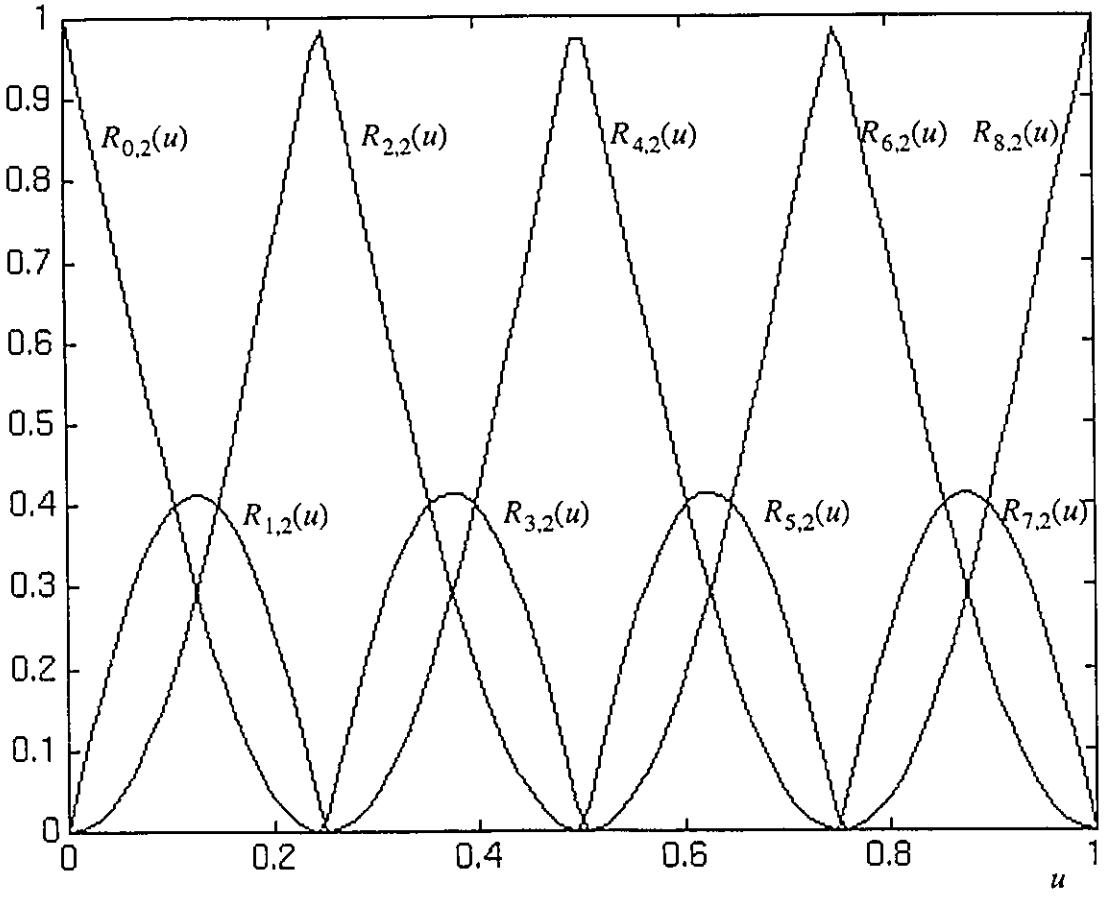


Figure 3 .6 Rational Basis Function of a Circle

3 .3 .2 NURBS Surfaces

A parametric surface is a function of two independent variables u and v. NURBS surfaces are given by the equation:

$$\mathbf{P}(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} R_{i,p;j,q}(u,v)\ \mathbf{P}_{i,j} \qquad (3.10)$$

where $\mathbf{P}_{i,j}$ are the control points of the approximating control net. A control net has (n+1) $\times$ (m+1) control points. The rational basis function of a NURBS surface is given by the equation:

$$R_{i,p;j,q}(u,v) = \frac{w_{i,j}\ N_{i,p}(u)\ N_{j,q}(v)}{\sum_{i=0}^{n} \sum_{j=0}^{m} w_{i,j}\ N_{i,p}(u)\ N_{j,q}(v)} \qquad (3.11)$$

where p and q are the degrees of $N_{i,p}(u)$ and $N_{j,q}(v)$ respectively.

3 .4 Fitting a NURBS Curve to a Set of Generated Points

This section investigates methods of interpolating curves to generated (or measured) points. Curve interpolation will be used for surface interpolation discussed in the next section. The following inputs are needed for interpolating a curve to generated points (Figure 3 .7).

1. A parametric equation of the nominal curve $\mathbf{C}(u)$ shown by the continuous curve in Figure 3 .7 .

2. A set of N+1 target points $\mathbf{G}_i$ $i \in \{0,1,2,....,N\}$ located on $\mathbf{C}(u)$.

3. The set of parameter values U_i $i \in \{0,1,2,....,N\}$ corresponding to $\mathbf{G}_i$.

4. A set of N+1 generated points $\mathbf{M}_i$ $i \in \{0,1,2,....,N\}$ obtained from the feature simulation routine, and generated between two offset ideal curves, shown by the dashed lines in Figure 3 .7 .

The generated points represent a discrete representation of the actual curve. A good approximation of the actual curve minimizes the deviation from the designed curve C(u), so that it does not contribute an unnecessary deformation to the actual representation. A good representation should have a local variation property so that the actual curve can be locally pulled towards the generated points, such that each point affects a curve segment whose length depends on the relative spacing between the generated points. The mentioned reasons make NURBS an ideal representation to interpolate the actual curve through the generated points. NURBS can represent ideal curves used in features of size (straight lines and circles), thus starting from an ideal NURBS representation of the curve, the control points can be re–evaluated to ensure that the actual curve passes through the generated points, keeping the knot values and the weights the same. Since NURBS can be divided into segments (using conics in circles for example) , the local effect of every generated point can be taken into consideration. Since features of size are the only features under consideration, interpolating generated points whose ideal curves are either lines or circles will be considered.

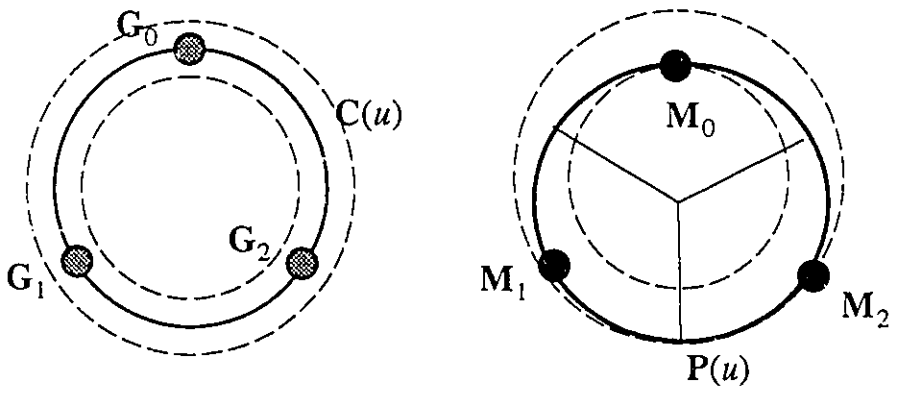


Figure 3 .7 Inputs to the interpolation algorithm

Thus the points $\mathbf{M}_0, \mathbf{M}_1, \mathbf{M}_2$, in Figure 3 .7 are to be interpolated by a NURBS curve $\mathbf{P}(u)$, shown by the bold line, divided into three segments, each affected by a generated point with the whole curve having a minimum deviation from the ideal curve C(u). The basic entity making the proposed NURBS representation of the interpolated curve is the conic. Conics represent ideal circles and can also be used to represent straight lines. Thus a change in their control points can be made to pass them through the generated points. At the same time each conic will be affected by one generated point.

3 .4 .1 Interpolating Conics to Generated Points

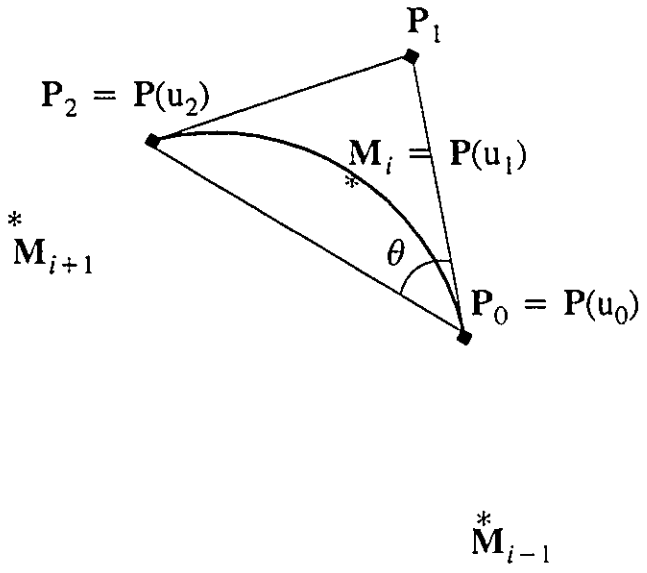


Figure 3 .8 Interpolating a Conic

Figure 3 .8 shows a conic whose ideal curve is a circular arc, interpolated between three generated points $\mathbf{M}_{i-1}, \mathbf{M}_i, \mathbf{M}_{i+1}$. The conic equation is given by:

$$\mathbf{P}(u) = \frac{(1-u)^2 w_0 \mathbf{P}_0 + 2u(1-u)w_1 \mathbf{P}_1 + u^2 w_2 \mathbf{P}_2}{(1-u)^2 w_0 + 2u(1-u)w_1 + u^2 w_2} \qquad u \in [u_0, u_2) \qquad (3.12)$$

where :

$\mathbf{P}_0$, $\mathbf{P}_1$, $\mathbf{P}_2$ are the control points,

w_0, w_1, w_2 are the weights, and

u_0 and u_2 are the limits of the independent parameter range corresponding to the conic.

The following calculations are carried out to interpolate the conic:

3 .4 .1 .1 Parameter Values

There are several methods of estimating the parameter values of the generated points. Some of these methods are discussed by Lee (1989), but since the target points are very close to the nominal curve, the following assumption can be made. Assuming that the points are generated for curves on machined surfaces, the specified tolerances would be very small in value relative to the curve's length. It can be concluded that the target points have the same parameter values as the generated points. Thus at the generated point $\mathbf{M}_i$ the parameter value u_1 is approximately equal to the parameter value U_i corresponding to the target point. The conic starts and ends between successive generated points. The parameter values at which the conic begins and ends can be averaged between those of the generated points:

$$u_1 = U_i \tag{3 .13}$$

$$u_0 = (U_i + U_{i-1})/2 \tag{3 .14}$$

$$u_2 = (U_i + U_{i+1})/2 \tag{3 .15}$$

3 .4 .1 .2 Interpolated Points

The coordinates of the points where the conic starts and ends have to be interpolated between generated points. The type of interpolation depends on the type of the ideal curve. If the ideal curve is a circle, then the interpolation will be quadratic. Let :

$$\mathbf{C}(u) = Au^2 + Bu + C \tag{3 .16}$$

then the constants A, B and C can be determined by substituting for u and $\mathbf{C}(u)$ by U_{i-1}, U_i, U_{i+1} and $\mathbf{M}_{i-1}$, $\mathbf{M}_i$, $\mathbf{M}_{i+1}$ respectively and solving three linear equations. The coordinates of a point $\mathbf{P}_0$ at the beginning of the conic can be calculated by substituting the corresponding parameter value in equation (3 .16). If the ideal curve is a straight line, then the starting points can be obtained by linear interpolation between the generated points.

3 .4 .1 .3 Weights

The weights of the interpolated curve are the same as those for a conic representing an ideal curve. If the ideal curve is a straight line, then all weights are equal to one. If the ideal curve is a circular arc (Figure 3 .8) then the weights of the beginning and end points are equal to one and the weight of the mid point is equal to $\cos(\theta)$.

3 .4 .1 .4 Control Points

Only the middle control point of the interpolated conic remains to be calculated. Since the control points of the start and end points coincide with the curve, they have been already calculated using equation (3 .16). The middle control point $\mathbf{P}_1$ can be calculated by substituting u_1 and $\mathbf{M}_i$ for u and $\mathbf{C}(u)$ in (3 .12).

3 .4 .1 .5 End Points

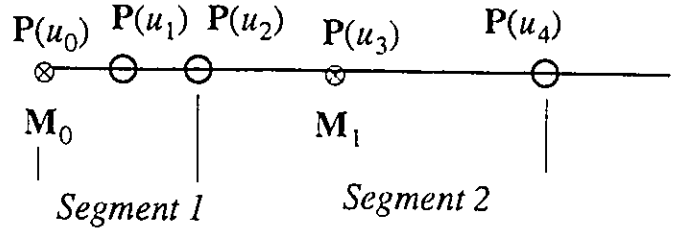


Figure 3 .9 Straight Line End Segment

If the generated points are generated along a circle, the first control point on the first conic and last point of the last conic coincide. Both points can be interpolated using equation (3 .16) between $\mathbf{M}_{N-1}$, $\mathbf{M}_0$ and $\mathbf{M}_1$. If the generated points lie along a straight line, then an extra point is inserted in the first (or last) segment. In Figure 3 .9 this point corresponds to $\mathbf{P}(u_1)$, where $u_1 = (u_0 + u_2)/2$. The coordinates of the inserted point $\mathbf{P}(u_1)$ can be obtained by substituting u_1 for u in (3 .16), and the middle control point of the segment can be calculated by substituting u_1 and $\mathbf{P}(u_1)$ for u and $\mathbf{P}_1$ in (3 .12).

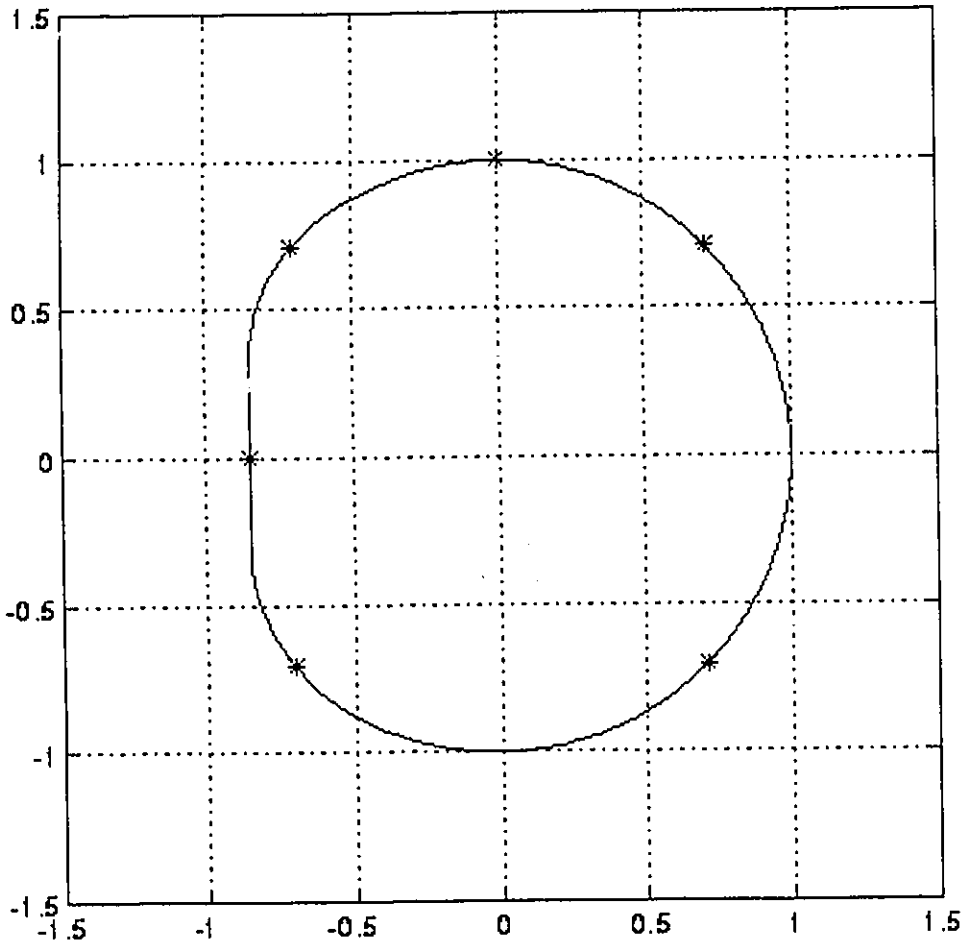


Figure 3 .10 Interpolated Curve for Points along a Circle

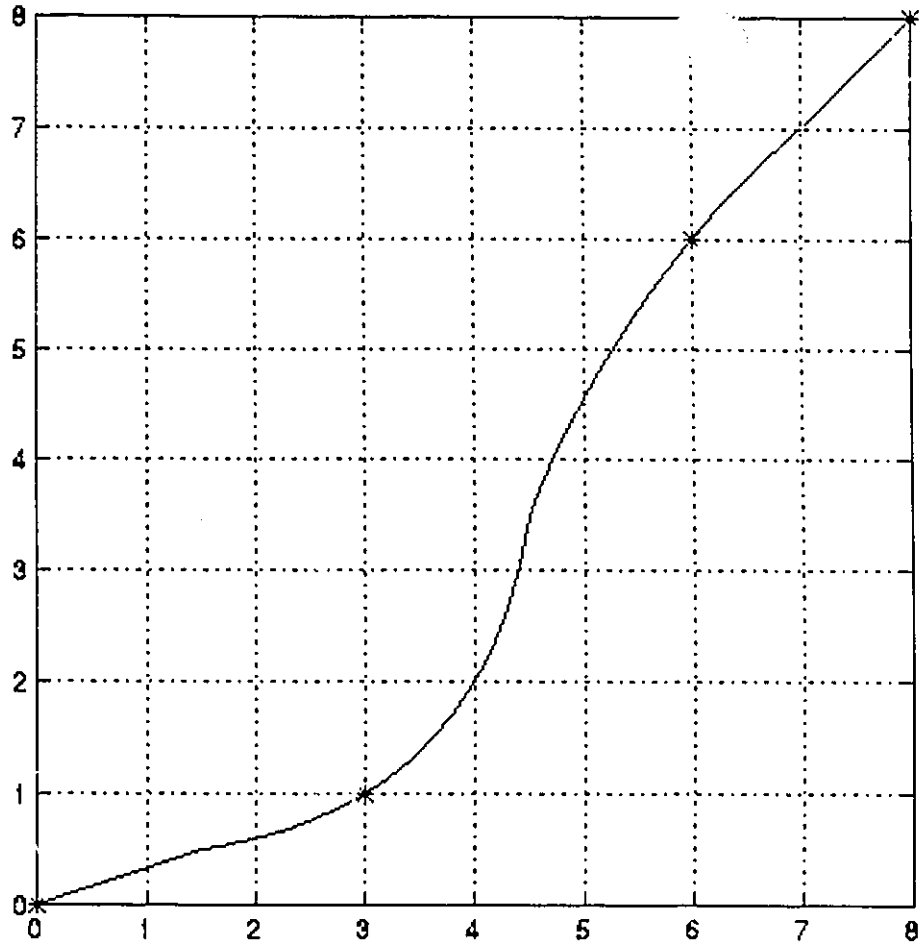


Figure 3 .11 Interpolated Curve for Points along a Straight Line

3 .4 .1 .6 Knot Vector

The knot vector of the interpolated curve is constructed from the parameter values of the start and end points of the curve segments.

$$t = \{0, 0, u_0, u_2, u_2, u_4, \ldots u_{2N-2}, u_{2N-2}, u_{2N}, 1, 1\} \qquad (3.17)$$

3 .4 .2 Examples of Curve Interpolation

Figure 3 .10 shows an interpolated curve for six points measured along a circle and Figure 3 .11 shows another interpolated curve for four points measured along a straight line. In both examples all points were kept at their nominal position except for one point.

3 .5 Fitting a NURBS Surface to a Set of Generated Points

Surface interpolation can be done by surface lofting (Piegl, 1991) for points generated along planes or cylinders.

3 .5 .1 Cylinders

Given a set of N × M generated points along a cylindrical surface (N along the circular sections and M along the longitudinal sections), an actual surface can be interpolated along the generated points by first interpolating successive curves along the circular sections, then interpolating the longitudinal sections through the control points of the circular curves. Examples of interpolation to points on a cylindrical surface are shown in Figure 4 .9 and Figure 4 .34 .

3 .5 .2 Planar Surfaces

Given a set of M × N generated points along a planar surface, the procedure for interpolating a surface through these points is to interpolate a set of NURBS curves through the width–wise points, then interpolate the length–wise control points.

3 .6 Conclusion

This chapter presents a method for the interpolation of parametric surfaces to measured points using NURBS conic sections. The use of NURBS conic sections to interpolate lines through scattered data has been demonstrated previously by Piegl (1987). The approach shown in this chapter augments his method to interpolate surface patches to

measured points. This method can be used as a tool for aiding the evaluation of geometric deviations such as local size and axis straightness deviations shown in the next chapter. The method is limited to interpolating surface patches to points measured on manufactured planes and cylinders. This limitation does not affect the method's use in the evaluation of geometric deviations since form tolerances are normally specified on cylindrical and parallel–plane features. Other free form surfaces are controlled by profile tolerance which can be evaluated using the measured points directly. Since any deviation will be small relative to the feature's dimensions, a manufactured feature would have a shape very close to the ideal feature. Therefore any surface representation method which is not capable of representing the ideal feature cannot be used for interpolation. Representation methods like Bezier patches and B–Splines fail to represent ideal cylinders. Therefore, the NURBS–based method presented in this chapter is more appropriate for the intended interpolation.

CHAPTER FOUR

EVALUATION OF GEOMETRIC DEVIATIONS

This chapter describes a set of procedures for checking geometric deviations on generated parts features. The main purpose for evaluating actual geometric deviations is to check the compliance of a manufactured part (or a part generated by simulation) with the specified tolerances reliably and accurately, as well as provide a much needed input to computer software and systems designed for geometric tolerance design and simulation (ElMaraghy et al., 1990). Features are generated in the form of points as shown in the next chapter. The methods described in this chapter are used to check whether the geometric deviations of the generated points are within the acceptable range. Methods of evaluating deviations range in their computational complexity according to the deviation type. While size and form tolerances are computationally complex to evaluate, orientation and position tolerances are relatively easier. The developed method for interpolating a parametric surface to the generated points, described in the previous chapter, is used to help evaluate some form tolerances. Genetic algorithms are used to evaluate the geometric deviations of size and form.

4 .1 Metrology Overview

The development of algorithms for the evaluation of actual geometric deviations on manufactured parts lies within the area of metrology. Voelker (1993) stated that there are three generic measurement technologies in use today (Figure 4 .1). These are: (i) functional gages where a high precision gage is used to check whether a manufactured part conforms to the specified tolerance, (ii) manual methods and (iii) sampled coordinate metrology (CMMs). The last method has two advantages over the first two. First, sampled coordinates

can be worked upon by software to emulate the hard gages, hence eliminating their high cost. Second, the use of computer software allows for the implementation of non physically gage–able criteria.

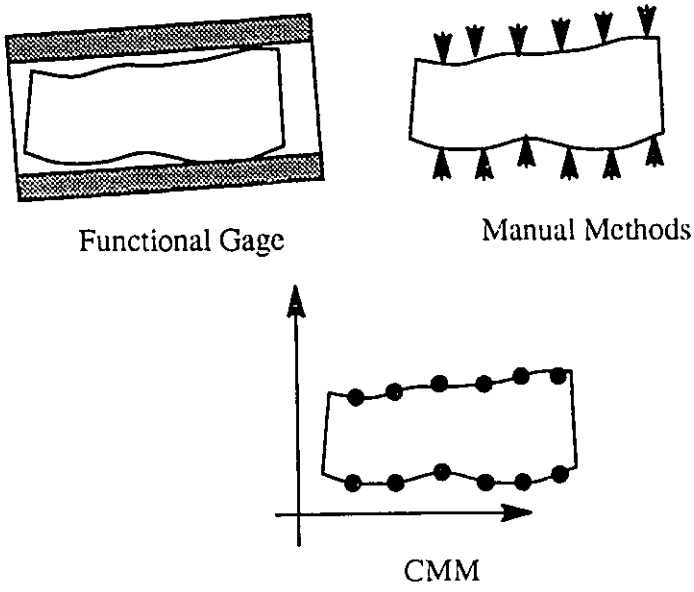


Figure 4 .1 Measurement Methods (Voelker 1993)

There are two types of CMM sampling technologies currently in use. The first applies tactile p ensing technology and the second uses laser camera technology. The second has the advantage of sampling a large number of points in a short time, but on the other hand the sampled points need to be reduced in order for the software using the sampled points to be time–efficient. The sampled points are used to evaluate the geometric deviations with the aid of computer software. There are two categories of algorithms that evaluate geometric deviations. The first is the one–step least squares method and the second category uses optimization to find the minimum geometric deviation zone. Although the least squares method is the one widely used by current CMM software due to its simplicity, it yields incorrect values for the geometric deviations (Yu, 1992).

4 .2 Addressed Problems

This chapter presents a number of optimization algorithms to evaluate actual geometric deviations, and addresses the following problems:

1. The methodologies developed to date depend on algorithms that do not guarantee the evaluation of globally minimum deviation zones. This problem is solved by using genetic algorithms. The new algorithms are applied to examples presented by other researchers for verification and comparison.

2. With the advent of the new ANSI Y14.5.1M standards and its mathematical definition of geometric tolerances, some form tolerances as well as the size tolerance have new definitions. This requires that an approximation of the manufactured (or generated) surface be evaluated if the number of sampled points is not large (as in the case of the widely used tactile sensing CMMs). The method for interpolating a parametric surface to the generated points (**Chapter 3**) is used to approximate the actual surface.

The evaluation of geometric deviations raises other problems such as the inclusion of measuring machine errors, and the specification of the number of sampled points needed for a correct evaluation. However, these are beyond the scope of this dissertation, although the later problem is addressed briefly at the end of this chapter.

4 .3 Problem Formulation

Given a set of n points, measured by a CMM or generated by computer simulation:

$$\mathbf{P}_1, \mathbf{P}_2, ..., \mathbf{P}_n \tag{4.1}$$

measured on a surface and a function z,

$$z = z(p_1, p_2, ... p_m \ , \ \mathbf{P}_1, \mathbf{P}_2, ..., \mathbf{P}_n) \tag{4.2}$$

defining the deviation zone on the measured points, where (p_1, p_2, ... , p_m) are parameters of the function z, find the values (p^*_1, p^*_2, ... , p^*_m) for which:

$$z^* = \min(z) \tag{4.3}$$

4 .3 .1 Choice of Optimization Method

Early work in the area of evaluating geometric deviations used the least squares method (Hopp, 1993), which was proved incapable of calculating the minimum zone deviation (Dhanish and Shunmugam, 1991, Hopp, 1993, and Yu, 1992). Several optimization methods were then proposed. These are summarized in Table 4 .1 :

Method	Reference
Linear Chebyshev Approximation	Dhanish and Shunmugam (1991)
Direct Search	W. ElMaraghy et. al. (1990)
Constrained Optimization with Augmented Lagrangian Function	Yu (1992)
Simplex Search	Murthy and Abdin (1980)

Table 4 .1 Optimization Methods used for Minimum Zone Evaluation

Since all of the above methods do not guarantee the arrival at the global minimum value of the deviation zone (the simplex search being an exception for the optimization of two or three variables), genetic algorithms were used to evaluate the minimum zone deviations. Genetic algorithms (Appendix A) are guaranteed to arrive at a near-global optimal solution provided that the genetic parameters were suitably selected and the problem is properly coded. The examples used in the above references were compared with those obtained using the genetic algorithms formulation, as shown in the following sections. A full

description of how genetic algorithms are used as function optimizers is provided in Appendix A.

4 .3 .1 .1 Example 1: Straightness

Figure 4 .2 shows a set of measured points that was used by Dhanish and Shunmugam (1991) to evaluate straightness. The same example was evaluated using genetic algorithms to minimize the function:

$$z^* = \min_{y_1,y_2}\left(\max_i(d_i) - \min_i(d_i) \right) \qquad (4.4)$$

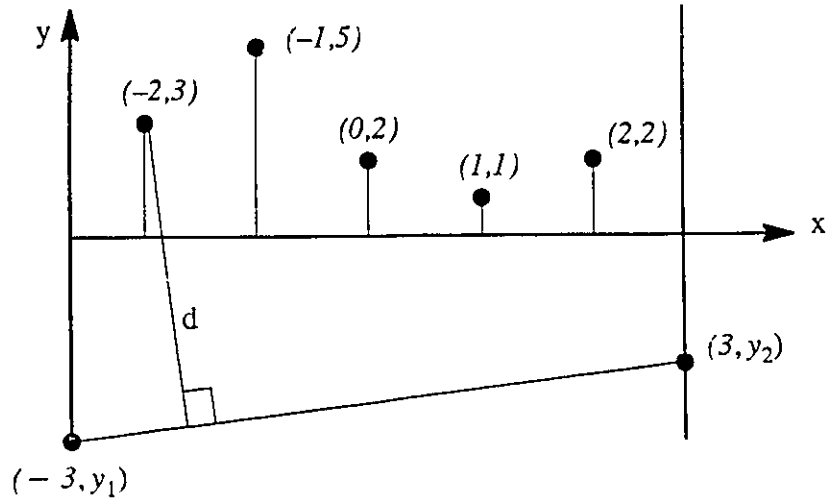


Figure 4 .2 Straightness Example

The genetic algorithm was run using the following parameters: population size = 30, cross–over probability = 0.8 and mutation probability = 0.05. The result of the run is shown in Figure 4 .3 along with the straightness value obtained by Dhanish and Shunmugam (1991). Results show that the genetic algorithms approach converged to a value 20% less

than that obtained by Dhanish and Shunmugam (1991).

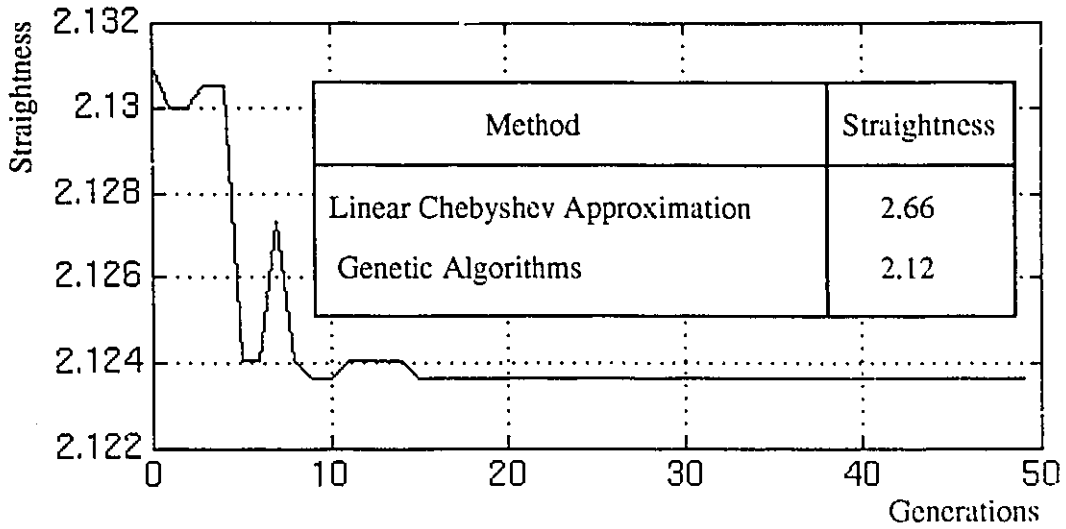


Figure 4 .3 Results of Straightness Example using GAs

4 .3 .1 .2 Example 2: Circularity

Murthy and Abdin (1980) used the spiral search method to evaluate the circularity deviation of an example consisting of 24 points measured in the same plane. The method can make a complete scan of the search space for the global minimum provided that the number of independent variables is limited to two or three. Hence, their work was limited to problems having fewer than four independent variables such as circularity and sphericity. Their example for circularity was composed of 24 measurements along a circular periphery equally separated by 15 degrees. The measured radii are shown in Table 4 .2 . The minimum circularity deviation zone was defined by the objective function:

$$z = \min_{\mathbf{p}}\left(\max_{i}\left(R_i \right) - \min_{i}\left(R_i \right) \right) \qquad (4.5)$$

where **p** is the centre of the circularity zone z. The same example was solved using genetic algorithms, whose results are shown in Figure 4 .4 .

i	01	02	03	04	05	06	07	08
R_i	13.68	13.68	13.68	13.44	14.40	14.88	15.10	16.40

i	09	10	11	12	13	14	15	16
R_i	16.80	17.20	17.20	18.00	18.40	18.80	18.00	18.00

i	17	18	19	20	21	22	23	24
R_i	18.00	17.44	16.60	15.80	15.40	15.04	14.40	13.80

Table 4 .2 Radii for the Circularity Example

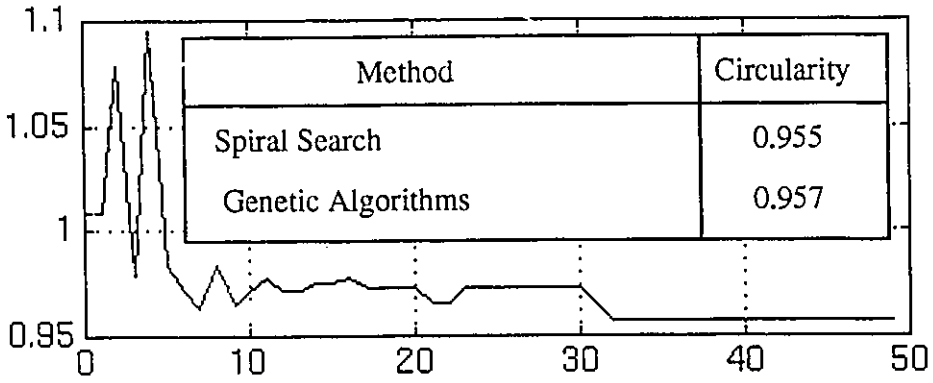


Figure 4 .4 Circularity Example Results

Results show that the two methods arrived at the same results. Considering that genetic algorithms are more computationally intensive than the spiral search, the spiral search method would be more efficient in cases of two independent variables.

4 .3 .1 .3 Example 3: Cylindricity

The following example was tried by Dhanish and Shunmugam (1991) and later by Yu (1992) for the evaluation of cylindricity. The example points are given in Table 4 .3 in

the form of radii measured at certain angles around the cylinder's periphery and at certain heights along the cylinder's axis.

z_i \ θ_i	0	45	90	135	180	225	270	315
1	5	3	4	3	1	2	2	3
0	4	4	3	3	3	2	2	3
−1	3	4	3	2	3	1	2	2

Table 4 .3 Cylindricity Example

The same example was solved to obtain a minimum cylindricity zone using genetic algorithms. Figure 4 .5 shows the formulation of the objective function used. The x,y of the tolerances of size, form, orientation and position tolerancescoordinates of the cylindricity zone end points are the independent variables. The algorithm minimized the function

$$z = \min_{p_1,p_2}\left(\max_i(d_i) - \min_i(d_i) \right) \qquad (4.6)$$

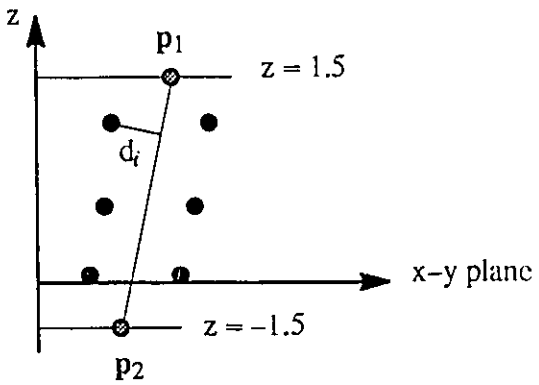


Figure 4 .5 Objective Function for Cylindricity Example

Figure 4 .6 shows the result of the genetic algorithm compared with the cylindricity values obtained by other methods (Dhanish and Shunmugam, 1991, and Yu, 1992).

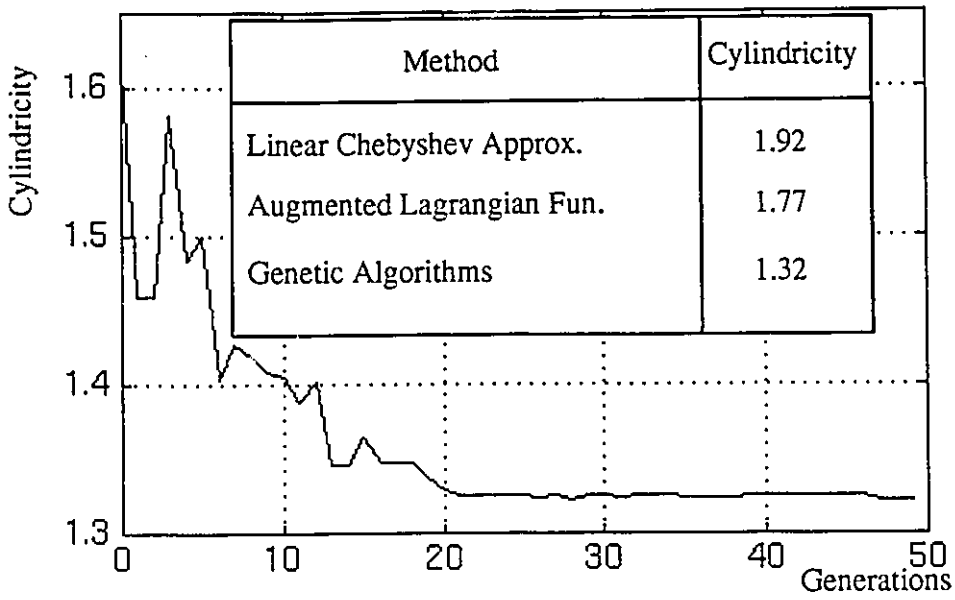


Figure 4 .6 Cylindricity Example Results

These results show that the use of a genetic algorithms approach resulted in a reduction of the minimum cylindricity value obtained to about 25% of that obtained by the use of the augmented Lagrangian function. This reduction indicates that the use of earlier methods can lead to the rejection of good parts which are within the specified tolerances.

4 .4 Measurement Axis vs. Feature's Actual Axis

The examples shown in sections **4 .3 .1 .1** to **4 .3 .1 .3** compared CMM measurements made along the measurement axis (Figure 4 .7). This axis differs from the actual axis of the manufactured feature. This difference adds an error to the calculation of some geometric deviations. For example, Foster (1986) describes the circularity tolerance as a zone enclosed by two concentric circles lying in a plane at any cross–section of the feature and perpendicular to the feature's axis. In practice points are taken in a plane normal to the CMM measurement axis. In the coming sections the measurement axis is referred to as the CMM axis.

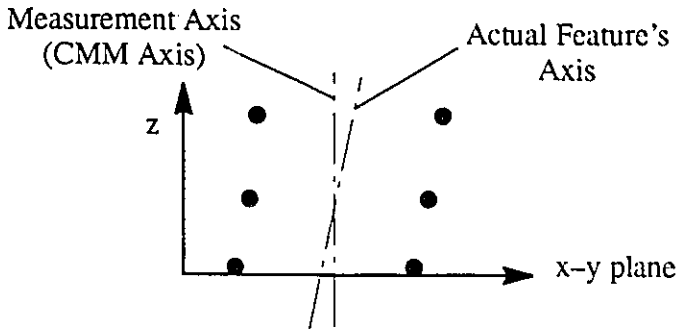


Figure 4 .7 Feature's Axis vs. CMM Axis

The method described in the previous chapter for interpolating parametric surfaces to generated points can be used to interpolate a NURBS surface to the generated points which in turn is used in the generation of the part's actual axis.

4 .5 Evaluation of Geometric Deviations

The following sections describe a set of procedures used to evaluate geometric deviations of a manufactured cylindrical feature. The example shown in Figure 4 .9 is used to verify the algorithms used to calculate geometric deviations. The cylinder surface is a NURBS representation interpolated through 4×5 points. The shown cylinder is ideal in shape except for a depression of maximum value of 0.2 inches introduced at its middle section. Thus the actual geometric deviations are known beforehand, and are given in Table 4 .4 .

4 .5 .1 Evaluation of Size Deviation

Size deviation is checked for two criteria according to the ANSI Y14.5M 1982, standards. The first criterion is that the perfect form envelope of the feature should have a diameter (or a cross sectional size in case of a pair of parallel planes) less than the maximum material condition (MMC) of the specified size tolerance. This means that the least

circumscribing perfect cylinder (or maximum inscribing for holes) to the feature should have a diameter less than the MMC of the specified size tolerance.

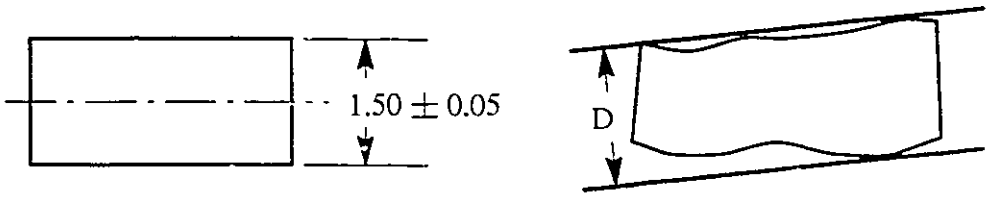
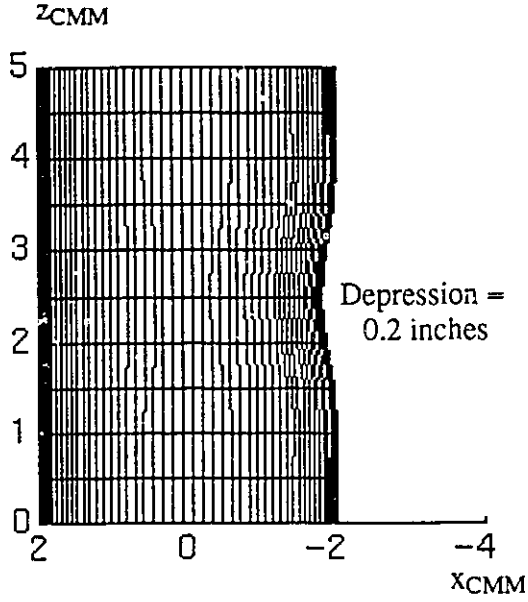


Figure 4 .8 Perfect Form Size

Thus the least circumscribing cylinder to the actual feature shown in Figure 4 .8 should have a diameter less than 1.55 inches. The second criterion for which a size tolerance should be checked is local size. Foster (1986) states that the actual size of an individual feature at any cross–section shall be within the specified tolerance of size. This definition of local size indicates that at any cross–section a two–point grip of a micrometer or a caliper should read a diameter within the limits of size. Neumann (1993) and Srinivasan (1993) indicated that the ANSI Y14.5.1 standards modified this definition of local size, to one where at any cross section the largest inscribed sphere should have a diameter within the limits of size. Thus the diameter of the sphere shown in Figure 4 .8 should be within $1.45 \leq D \leq 1.55$.



r_i Radii of Measured Points (inches)

z_i \ θ_i	0.0	90	180	270
0.00	2.0	2.0	2.0	2.0
1.25	2.0	2.0	2.0	2.0
2.50	2.0	2.0	1.8	2.0
3.75	2.0	2.0	2.0	2.0
5.00	2.0	2.0	2.0	2.0

Figure 4 .9 A Deformed Cylindrical Feature

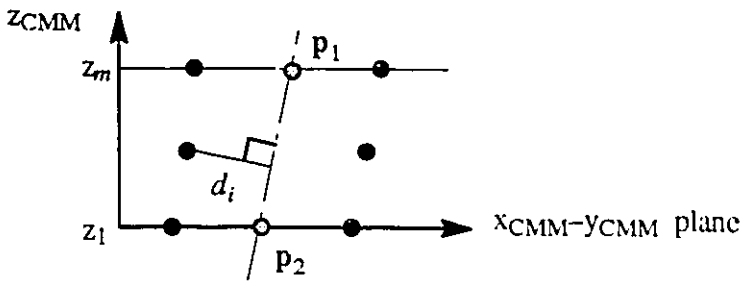
Deviation	Value
1. Perfect Form Envelope	4.0 inches
2. Local Size at Mid–Section	3.8 inches
3. Cylindricity	0.2 inches
4. Axis Straightness	0.1 inches
5. Straightness	0.2 inches

Table 4 .4 Geometric Deviations for the Cylinder Shown in Figure 4 .9 .

4 .5 .1 .1 Perfect Form Envelope

Given a set of $N = n \times m$ points $\{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_N\}$ representing a manufactured surface, and measured at m different planes perpendicular to a measurement axis z_{CMM}

(Figure 4 .10), the perfect form envelope to these points is evaluated using the minimax optimization objective function shown in Figure 4 .10 .



$$2 \times \max_{p_1,p_2}\left(\min_{i}\left(|d_i| \right) \right) \qquad \textit{Hole} \tag{4.7}$$

$$2 \times \min_{p_1,p_2}\left(\max_{i}\left(|d_i| \right) \right) \qquad \textit{Peg} \tag{4.8}$$

Figure 4 .10 Perfect Form Envelope Objective Function

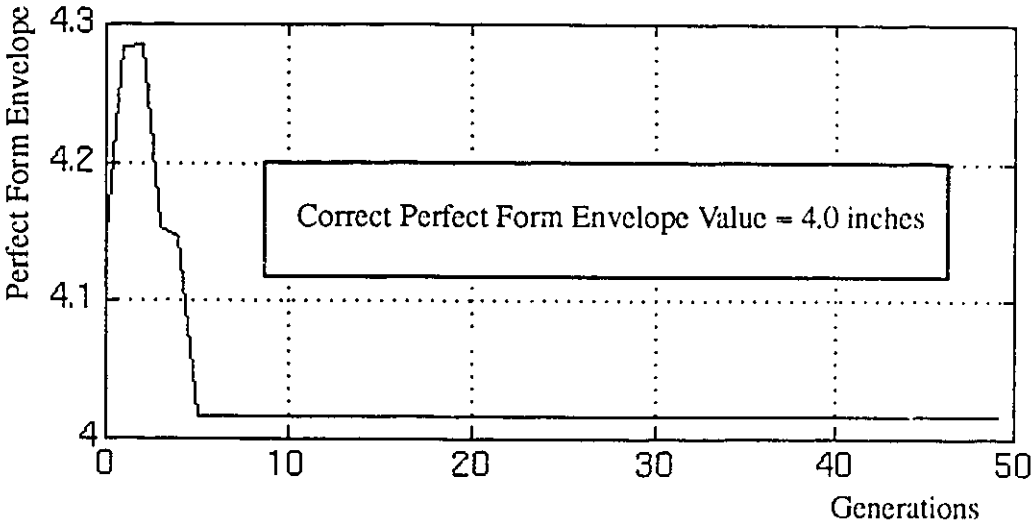


Figure 4 .11 Perfect Form Envelope Result for Cylinder in Figure 4 .9

The z_{CMM}–value of the points $\mathbf{p}_1$ and $\mathbf{p}_2$ are equal to z_1 and z_m respectively. This leaves four independent variables to be optimized (x_{CMM} and y_{CMM} for the points $\mathbf{p}_1$ and $\mathbf{p}_2$). The genetic algorithms results for the perfect form size of the cylinder shown in Figure 4 .9 is given in Figure 4 .11 .

4 .5 .2 Local Size

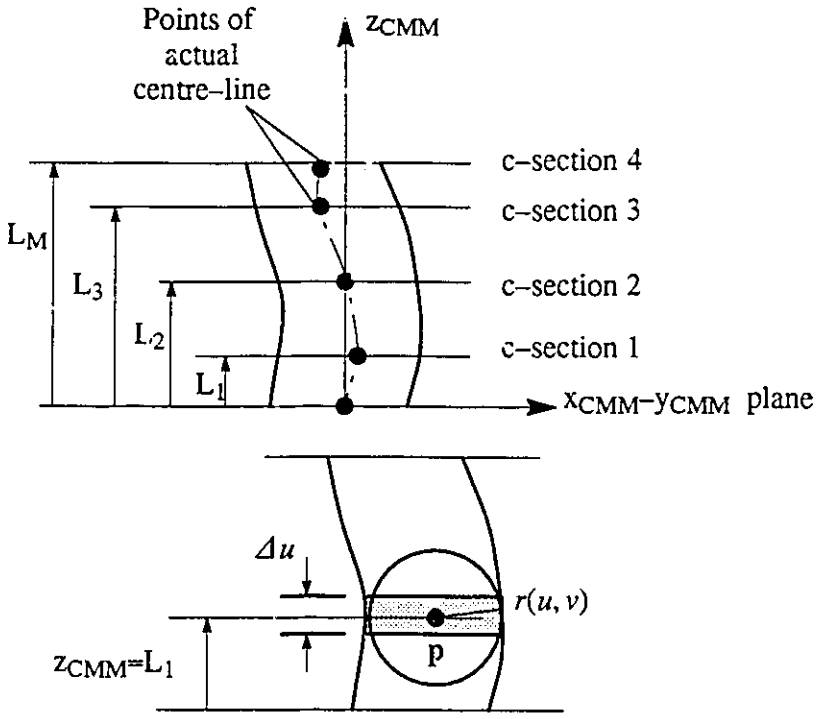


Figure 4 .12 Evaluation of Local Size

The local size of a feature is evaluated at a specified cross–section along the feature's length (Figure 4 .12). The local size is evaluated by finding the largest inscribed sphere to the feature at the specified cross–section. The feature is divided into M sections at planes perpendicular to the feature's nominal axis. At each plane, optimization is used to search for the position of the centre–point **p** of the largest sphere inscribed to the cylinder at the given cross–section. The independent variables are the x_{CMM} and y_{CMM} values of the point **p**. The objective function is:

$$2 \times \max_{\mathbf{p}}\left(\min_{u,v}\left(r(\mathbf{p}, u, v) \right) \right) \qquad (4.9)$$

where u and v are the independent parameters of the NURBS surface. The evaluated

centre–points are the points on the actual axis of the feature. Figure 4 .13 , Figure 4 .14 and Figure 4 .15 show the genetic algorithms results for the local size at z_{CMM}=0.0, 2.5 and 5.0 inches respectively along with the centre points of the inscribed spheres.

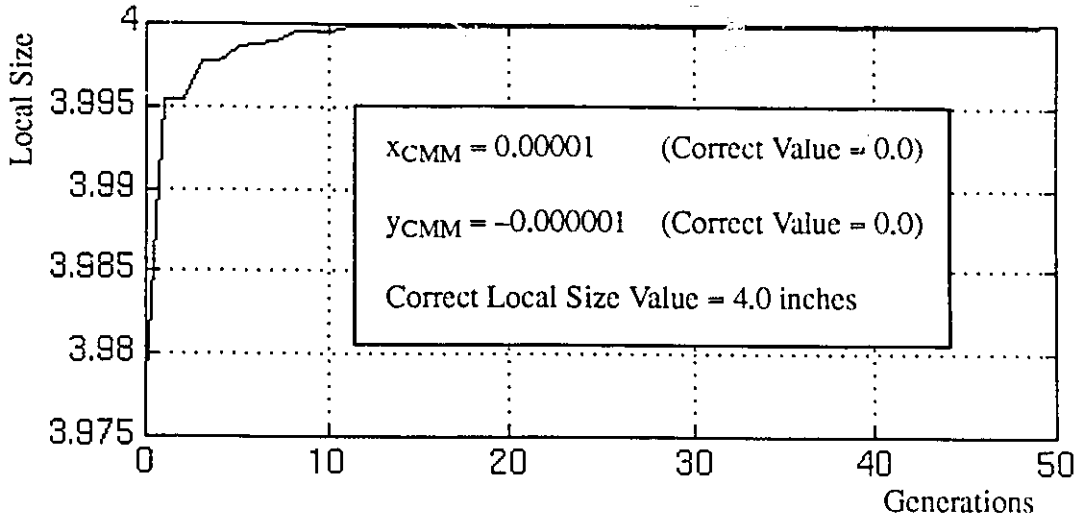


Figure 4 .13 Local Size Result for Cylinder in Figure 4 .9 at z_{CMM} = 0.0

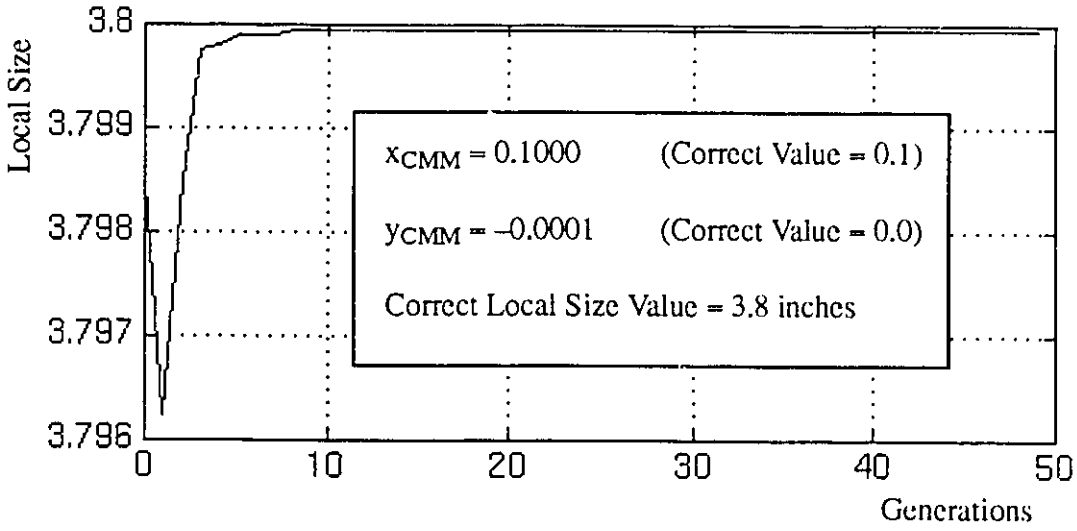


Figure 4 .14 Local Size Result for Cylinder in Figure 4 .9 at z_{CMM} = 2.5

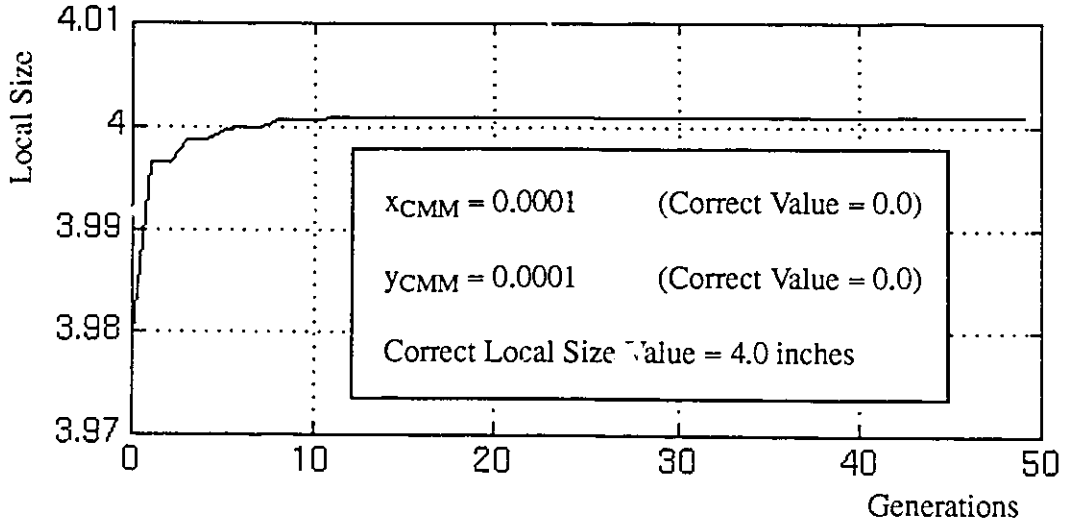


Figure 4.15 Local Size Result for Cylinder in Figure 4.9 at z_{CMM} = 5.0

4.6 Evaluation of Form Deviations

Tolerances of form are specified on features of size, a single surface or an element of a surface. No datums are needed to evaluate form deviations. The following sections describe methods of evaluating form deviations on a cylindrical feature.

4.6.1 Straightness

Straightness is a condition where an element of a surface or an axis is a straight line (Foster, 1986).

4.6.1.1 Surface Straightness

Straightness tolerance of a surface element is a zone enclosed by two parallel lines spaced by the specified tolerance and where both lines share a common plane with the nominal axis z_{CMM} (Foster, 1986). Optimization is used to find the orientation of the two parallel lines enclosing the surface elements points and whose straightness deviation is

minimum (Figure 4 .16). The objective function used is:

$$\min_{\mathbf{p}_1,\mathbf{p}_2}\left(\max_i(d_i) - \min_i(d_i)\right) \qquad (4.10)$$

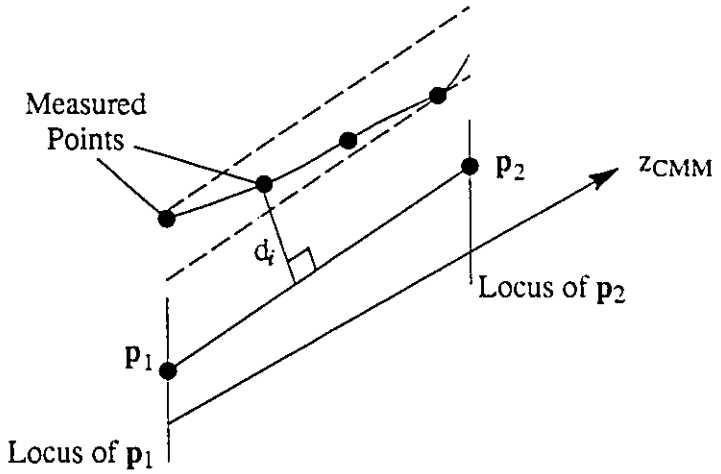


Figure 4 .16 Evaluation of Surface Straightness

where $\mathbf{p}_1$ and $\mathbf{p}_2$ lie in the same plane containing the measured points and z_{CMM}. The loci of $\mathbf{p}_1$ and $\mathbf{p}_2$ are perpendicular to z_{CMM}. The genetic algorithm results for the surface element straightness for the example in Figure 4 .9 is shown in Figure 4 .17 and Figure 4 .18 .

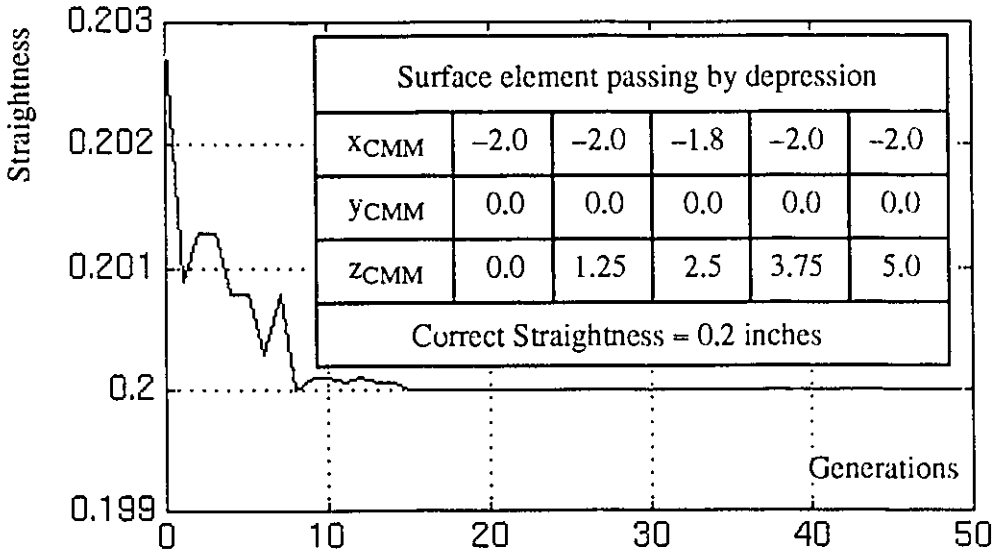


Figure 4 .17 Surface Element Straightness Result for Cylinder in Figure 4 .9

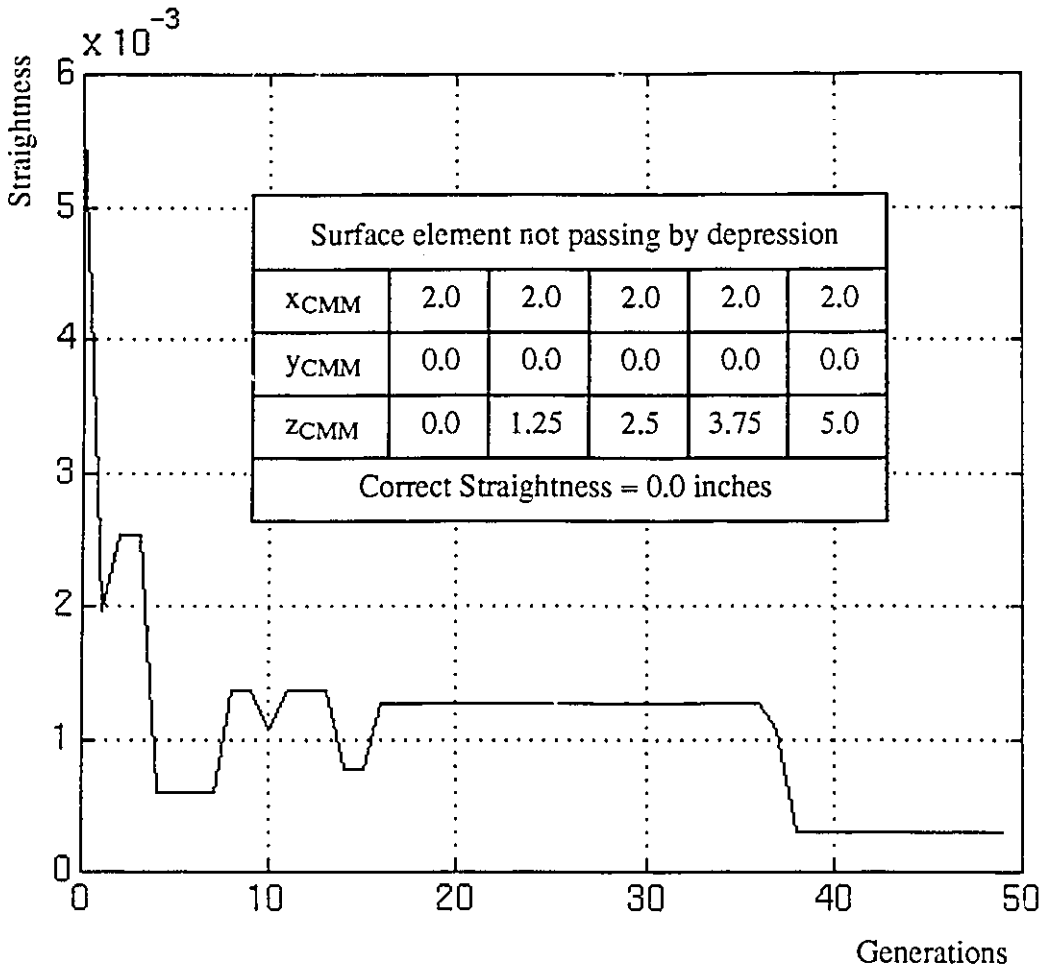


Figure 4 .18 Surface Element Straightness Result for Cylinder in Figure 4 .9

The first result (Figure 4 .17) corresponds to a surface element passing by the depression, while the second (Figure 4 .18) corresponds to a perfectly straight surface element. In both figures, x_{CMM}, y_{CMM} and z_{CMM} correspond to the coordinates of the measured points. However both results converged to the correct value.

4 .6 .1 .2 Feature Axis Straightness

Straightness can be specified to control the form deviation of a cylindrical feature's axis (Figure 4 .19). In such a case the actual axis of the feature must lie within a cylinder whose diameter is less than the specified tolerance.

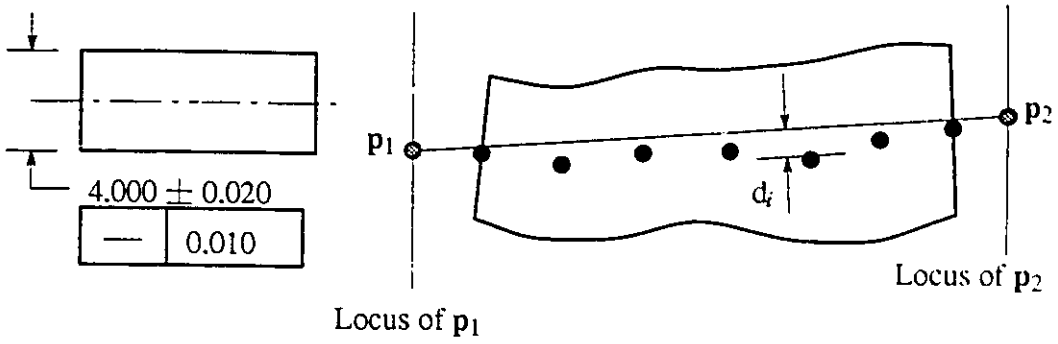


Figure 4 .19 Straightness of a Feature's Axis

Optimization using genetic algorithms is used to evaluate the cylindrical deviation zone. The centre–points of the local size spheres obtained in section **4 .5 .2** are used as the actual axis points. The objective function used is:

$$\min_{p_1,p_2}\left(\max_{i}\left(d_i\right)\right) \qquad (4.11)$$

Figure 4 .20 shows the genetic algorithms results for the straightness of the cylinder's axis shown in Figure 4 .9 .

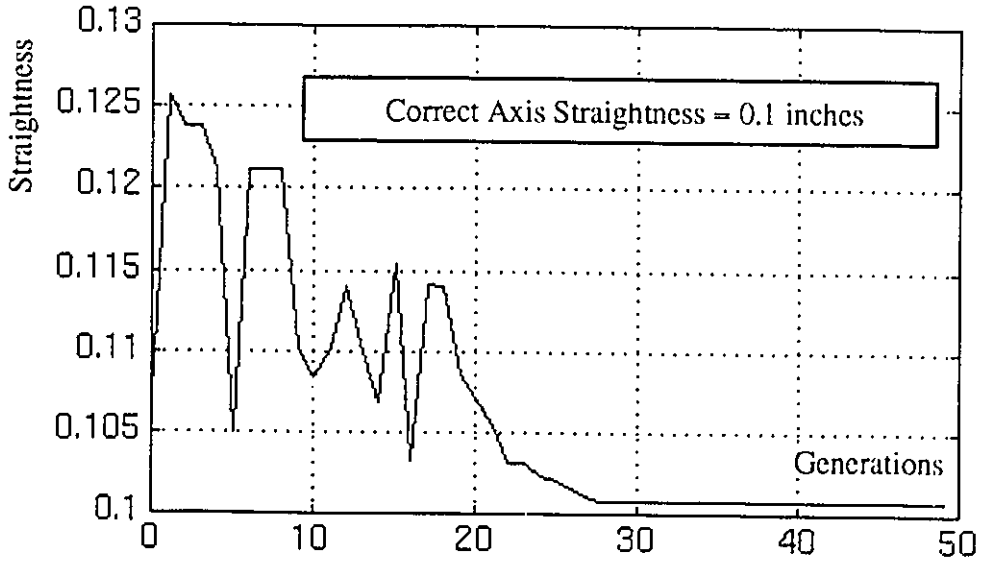


Figure 4 .20 Axis Straightness Result for Cylinder in Figure 4 .9

4 .6 .2 Circularity

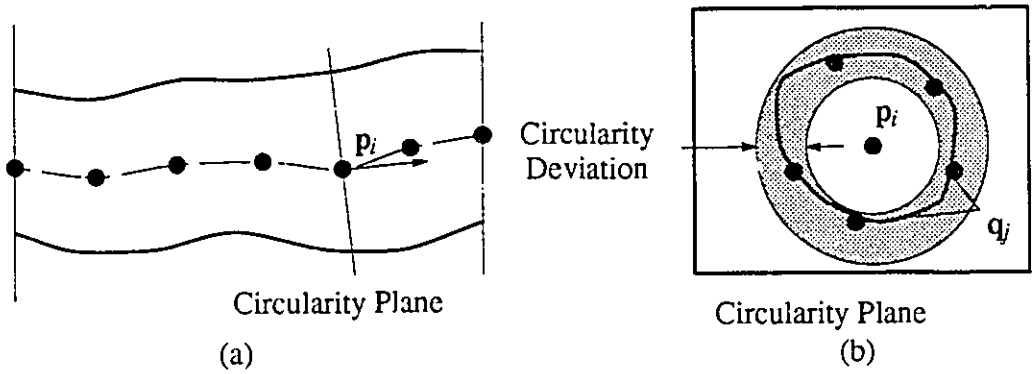


Figure 4 .21 Evaluation of Circularity

The definition of the circularity tolerance changed from the ANSI Y14.5M standards to a new definition by the ANSI Y14.5.1 standards. Walker and Srinivasan (1993) stated that the new definition of circularity is "*...finding a tangent continuous curve through each point of which cross–sectional planes are constructed normal to the curve. Within each plane, the centre of the annular tolerance zone must be located at the point of intersection of the curve with the plane*". Therefore, the following procedure is used to check circularity according to the definition:

1. Fit a NURBS curve to the centreline points evaluated in section **4 .5 .2 .**
2. At each centre–line point $\mathbf{p}_i$ (Figure 4 .21 a), construct a plane normal to the axis curve at $\mathbf{p}_i$.
3. In each plane generate a number of points $\mathbf{q}_j$ $j \in \{1,2,....,N\}$ (Figure 4 .21 b), at equal angles and positioned at the intersection of rays emanating from $\mathbf{p}_i$ with the NURBS surface.
4. The circularity deviation is equal to:

$$\max_j(\ |\mathbf{p}_i - \mathbf{q}_j| \ ) - \min_j(\ |\mathbf{p}_i - \mathbf{q}_j| \ ) \qquad (4.12)$$

Figure 4 .23 and Figure 4 .24 show the evaluated circularity for the cylinder shown in Figure 4 .9 at the depression (z_{CMM}=2.5) and at a perfectly circular cross section (z_{CMM}=5.0). The NURBS curve interpolated to the actual axis points is shown in Figure 4 .22 .

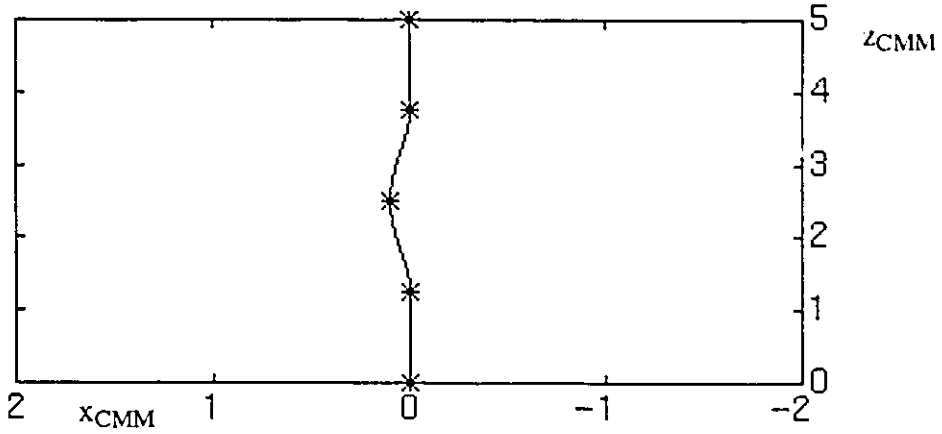


Figure 4 .22 Actual Axis of Example shown in Figure 4 .9 .

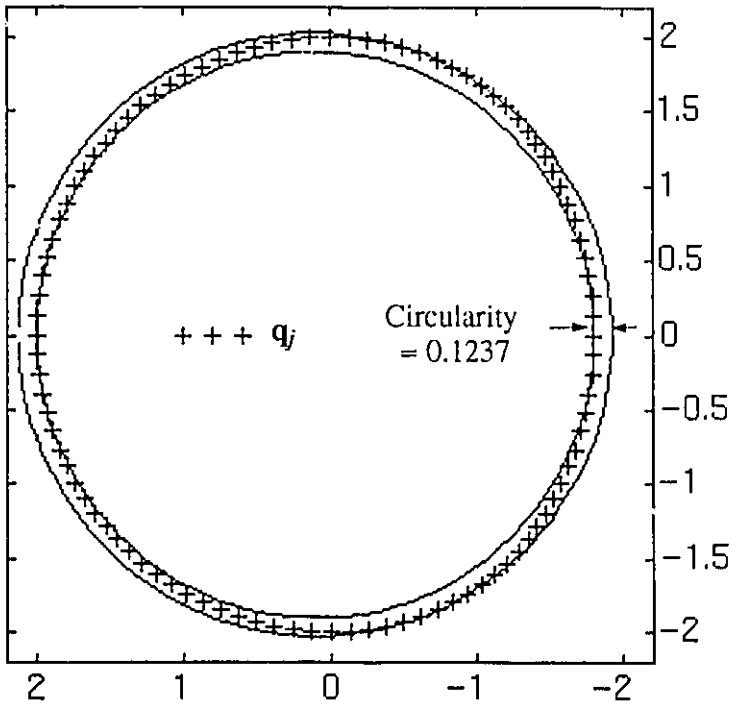


Figure 4 .23 Circularity at z_{CMM}=2.5 for the Example shown in Figure 4 .9 .

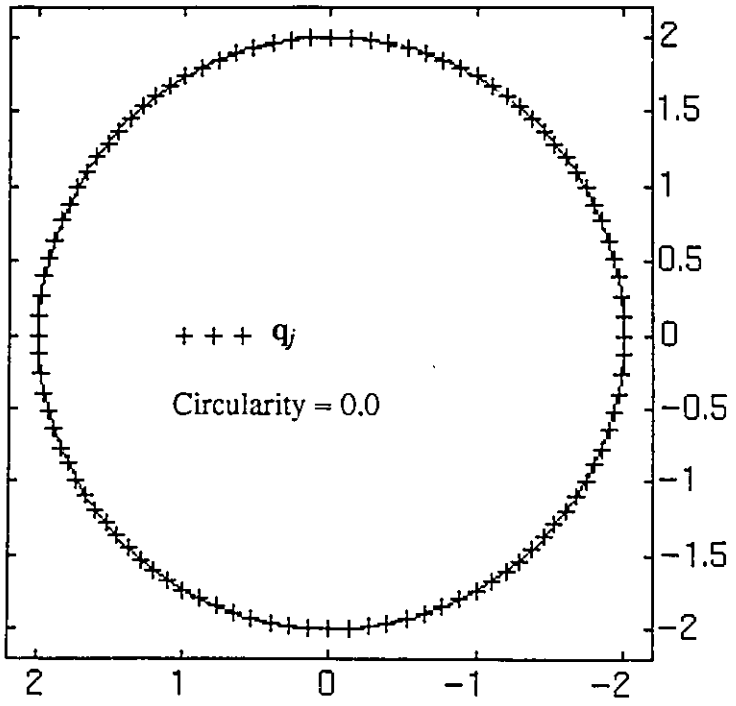
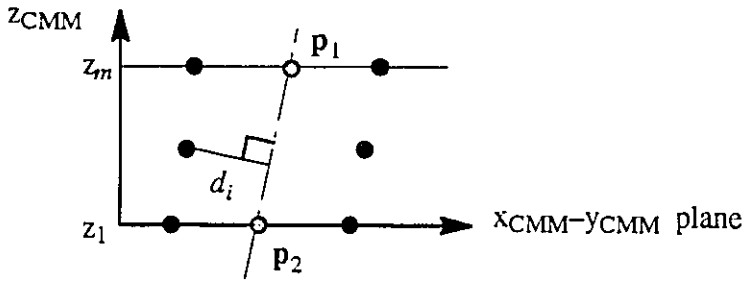


Figure 4 .24 Circularity at z_{CMM}=5.0 for the Example shown in Figure 4 .9 .

4 .6 .3 Cylindricity



$$\min_{p_1,p_2}\left(\max_i(d_i) - \min_i(d_i) \right) \qquad (4.13)$$

Figure 4 .25 Evaluation of Cylindricity

A cylindricity tolerance zone is bounded by two concentric cylinders within which the actual surface must lie (Foster, 1986). Optimization can be used to evaluate the cylindricity deviation directly from the measured points. The objective function is shown

in Figure 4 .25 . Figure 4 .26 shows the genetic algorithms results for evaluating the cylindricity of the example shown in Figure 4 .9 .

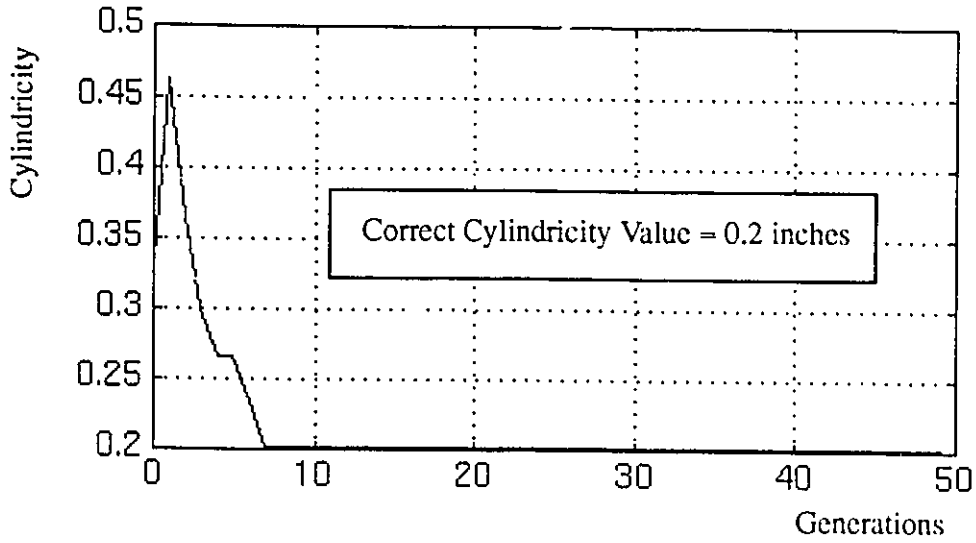


Figure 4 .26 Cylindricity Result for Cylinder in Figure 4 .9 .

4 .7 Datum Establishment

ANSI Y14.5M (1982) standards defines a datum as a theoretically exact point, axis or plane derived from the true geometric counterpart of a specified datum feature. If a plane is specified as a datum feature, then the datum is simulated by placing the manufactured surface on a perfect plane. Section **2 .4 .1** showed that a datum reference frame can consist of a plane and a cylinder perpendicular to the plane. A cylindrical feature's axis can be used as a datum feature. If the cylinder is a hole, the datum is the largest inscribed cylinder contained in the hole and perpendicular to the plane. The following section describes the establishment of a datum axis from a cylindrical feature.

4 .7 .1 Datum Axis as a Secondary Datum

Figure 4 .27 shows a cylindrical feature used to establish a secondary datum axis **B** perpendicular to a primary datum **A**. The measured points are projected on the datum plane **A**, then the centre **c** of the largest inscribed circle is obtained using the objective function:

$$\max_{\mathbf{c}}\left(\min_{i}\left(d_i\right)\right) \qquad (4.14)$$

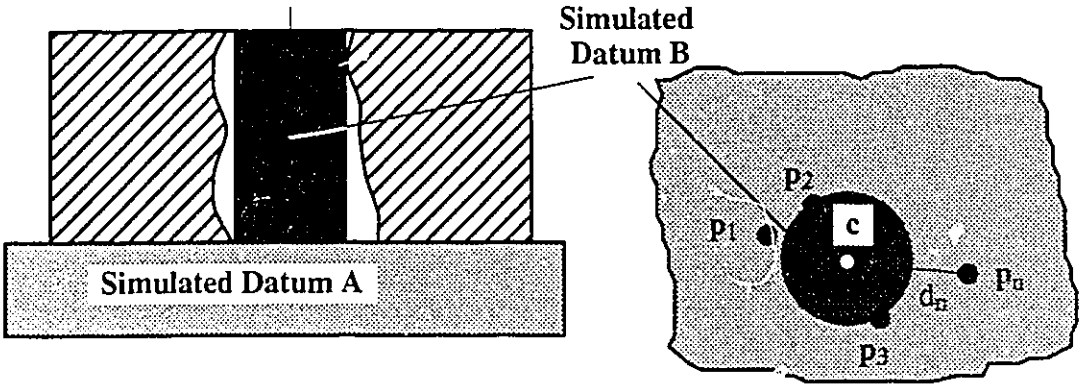


Figure 4 .27 Datum Establishment

The datum axis **B** passes through **c** and is perpendicular to the plane **A**. The cylindrical feature in Figure 4 .9 is used to establish a datum perpendicular to the feature's x_{CMM}–y_{CMM} plane. Figure 4 .28 shows the genetic algorithms result for evaluating the centre **c** of the datum axis.

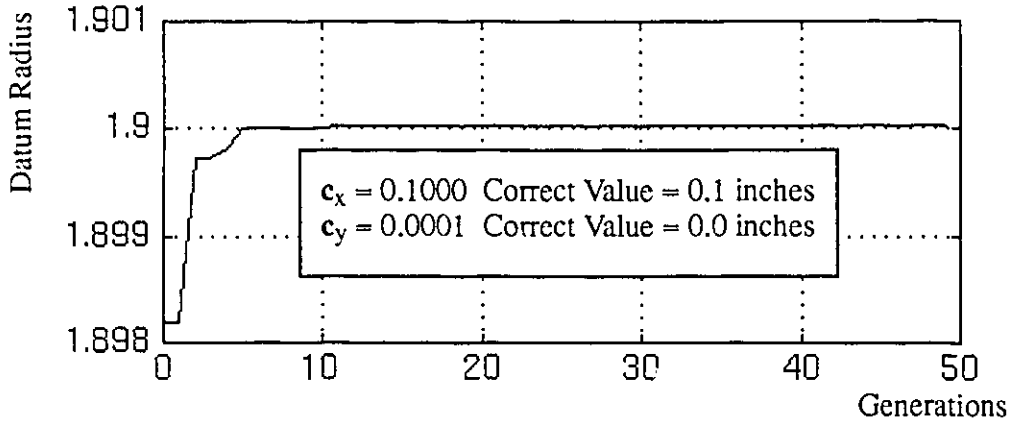


Figure 4 .28 Datum Establishment for Cylinder in Figure 4 .9 .

4 .8 Evaluation of Orientation Deviations

Orientation tolerances are specified to control the space orientation of a surface element, a surface or the axis of a feature of size. Orientation tolerances are always related to datums. ANSI Y14.5M (1982) standards classify orientation tolerances into perpendicularity, parallelism and angularity tolerances.

4 .8 .1 Perpendicularity

A perpendicularity tolerance for a cylindrical feature (Foster, 1982) specifies a cylindrical tolerance zone perpendicular to a datum plane within which the axis of a feature must lie. Perpendicularity deviation is calculated in two steps. The first projects the points $\{p_1, p_2, \ldots, p_n\}$ to the specified datum plane or line (Figure 4 .29 a). The second step evaluates the minimum diameter circle in the plane enclosing the projected points (Figure 4 .29 b).The diameter of the circle is equal to the perpendicularity deviation. The objective function used is:

$$\max_{c}\left(\min_{i}\left(d_i\right)\right) \qquad (4.15)$$

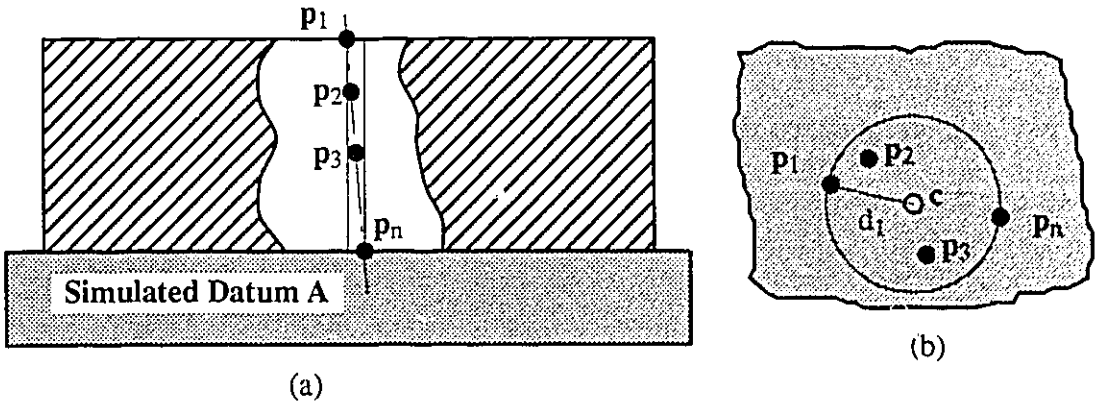


Figure 4 .29 Evaluation of Perpendicularity Deviation

Figure 4 .30 shows the genetic algorithms result for the evaluation of perpendicularity for the cylinder shown in Figure 4 .9 . The x_{CMM}–y_{CMM} plane is used as a datum for perpendicularity.

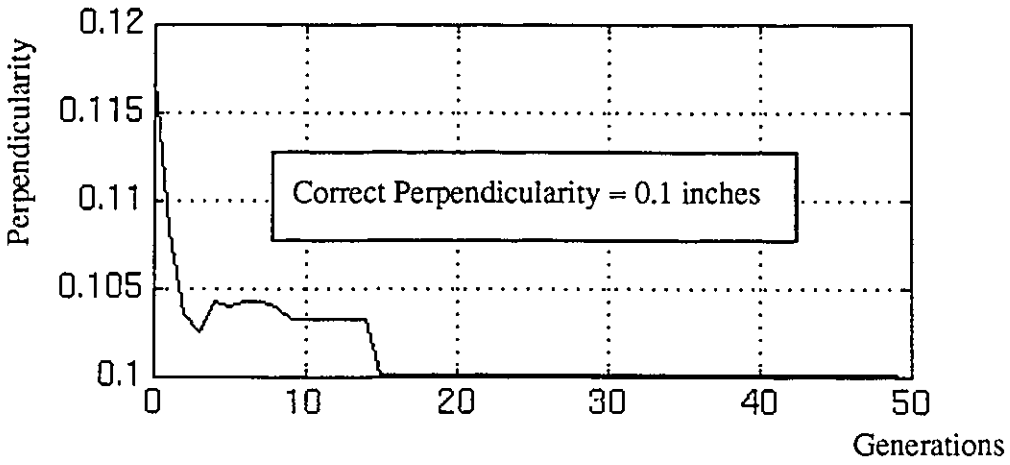


Figure 4 .30 Perpendicularity Run for Example in Figure 4 .9 .

4 .8 .2 Parallelism

A parallelism tolerance for a cylindrical feature (Foster, 1986) specifies a cylindrical tolerance zone parallel to a datum plane or line within which the axis of a feature must lie.

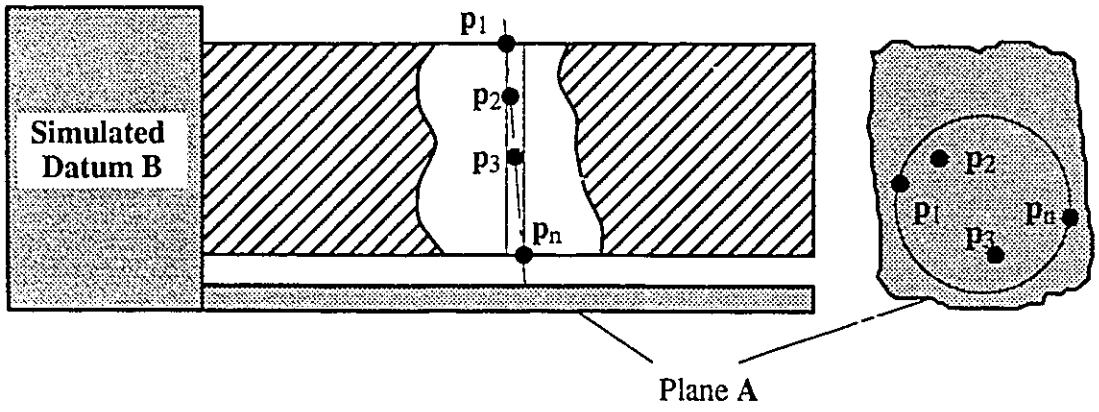


Figure 4 .31 Evaluation of Parallelism Deviation

Figure 4 .31 shows a feature whose orientation is to be controlled by parallelism to a datum plane **B**. The deviation can be evaluated by constructing a plane **A** perpendicular to the datum plane **B** and then projecting the actual axis points on **A**. The same objective function used for evaluating perpendicularity (4 .15) can then be used to evaluate the parallelism deviation.

4 .9 Evaluation of Position Deviations

Position tolerances control the position of a feature of size measured from a datum reference frame. Figure 4 .32 shows a feature toleranced by a position tolerance. A position tolerance of a cylindrical feature specifies a cylindrical tolerance zone located by basic dimensions from the datum reference frame within which the feature's axis should lie.

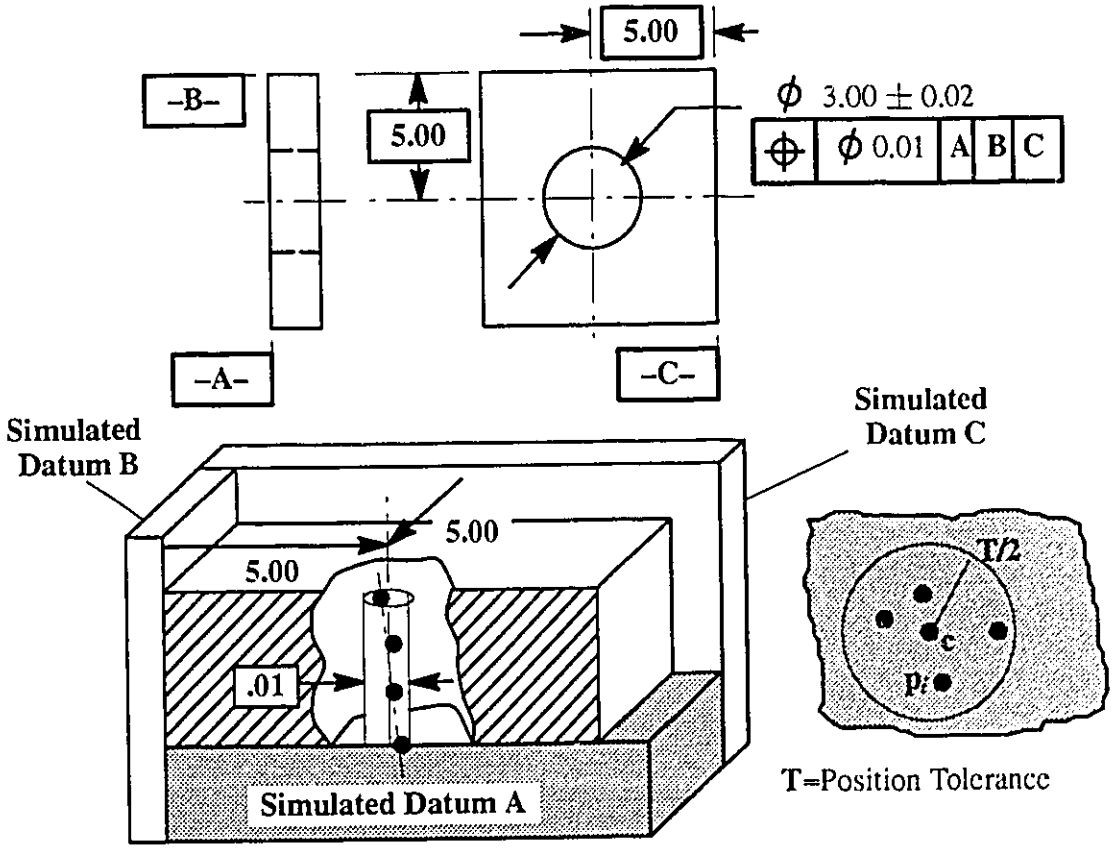


Figure 4 .32 Evaluation of Position Deviation

The feature's axis points are projected on the primary datum $\mathbf{p}_i$. Let point **c** (Figure 4 .32) be the point of intersection of the tolerance zone axis with the primary datum. The position deviation is equal to:

$$2 \times \max_i(\ |\mathbf{c} - \mathbf{p}_i| \ ) \qquad (4.16)$$

4 .10 Choice of Genetic Algorithms Parameters

The previous sections used genetic algorithms as a method for global optimization to evaluate the geometric deviations. All objective functions had either two or four independent variables. Since the parameters of the genetic algorithms affect the search (Appendix A), a study was carried out to find the suitable values of the population size and chromosome length. The example shown in Figure 4 .34 is used to evaluate the effect of these parameters on a genetic algorithms search for the minimum perfect form feature (four independent variables).

4 .10 .1 Coding of Independent Parameters

Genetic algorithms use strings to conduct searches instead of the real values of the independent variables. The conversion of the parameter real values into strings is known as coding. Coding discretizes the real domain of the independent variable between minimum and maximum attainable values. Section **A.1 .1** in Appendix **A** describes a general methodology for coding real independent parameters into binary strings that genetic algorithms can use. Instead of using the chromosome length as a parameter, the difference between the maximum and minimum possible values for each variable is used as a parameter along with the minimum difference between every pair of consecutive values for the independent parameters. Thus, for every independent variable the following parameters are required:

1. MAX = The maximum attainable value.

2. MIN = The minimum attainable value.

3. DIFF = The minimum difference between each pair of consecutive discrete values of the parameter.

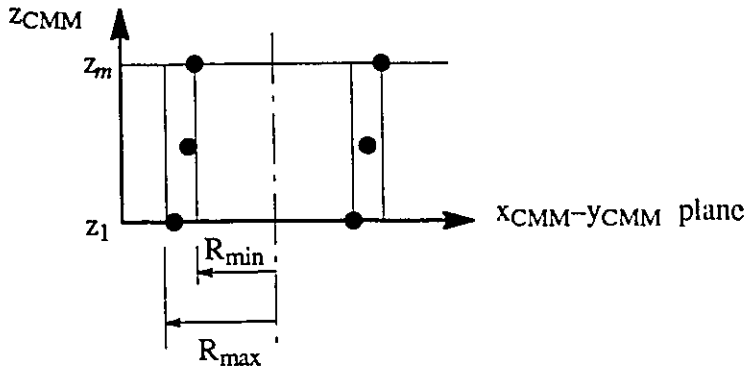
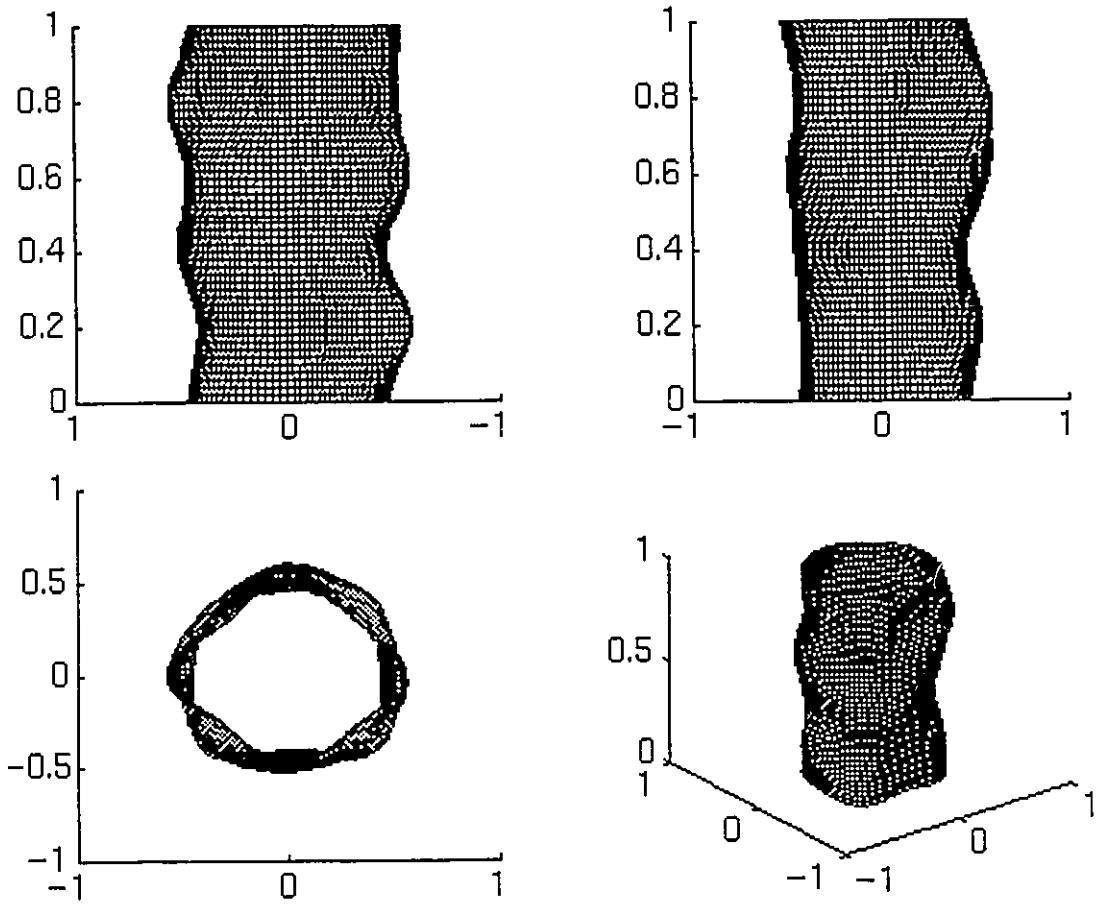


Figure 4 .33 Minimum and Maximum Radii of a Generated Feature

The objective functions described in this chapter optimized the x_{CMM} and the y_{CMM} values as the independent parameters. These values are located near to the feature's nominal axis (i.e. at x_{CMM} =0 and y_{CMM} =0). Since geometric deviations are controlled by the measured radii (Figure 4 .33) representing the manufactured feature, the independent parameter's domain is affected by the same radii. If the independent parameters domain covers a large range of values (for example if $MIN(x_{CMM})=-R_{max}$ and $MAX(x_{CMM})=R_{max}$), a larger number of genetic generations will be needed to reach convergence. On the other hand, a very small domain of values might lead the genetic optimization to miss the global deviation value. A good approximation of the domain values can be taken slightly larger than the difference between the largest and smallest measured radii.

$$MAX(x_{CMM} \text{ or } y_{CMM}) = 1.2 \times (R_{max}-R_{min}). \quad (4.17)$$

$$MIN(x_{CMM} \text{ or } y_{CMM}) = -1.2 \times (R_{max}-R_{min}). \quad (4.18)$$



Measured Points : $R_{i,j}$

θ_i \ z_j	0.0000	0.2000	0.4000	0.6000	0.8000	1.0000
000	0.4793	0.4319	0.5208	0.4810	0.5576	0.4609
045	0.5681	0.4767	0.5165	0.5715	0.4531	0.4304
090	0.4707	0.5382	0.4540	0.5884	0.5966	0.4675
135	0.4893	0.4118	0.4781	0.5326	0.4614	0.4775
180	0.4637	0.5800	0.4587	0.5693	0.5202	0.5287
225	0.5773	0.4327	0.5485	0.4006	0.5217	0.5507
270	0.4031	0.4318	0.4597	0.4925	0.4425	0.5207
315	0.5168	0.5066	0.4151	0.5065	0.5772	0.5063

Figure 4 .34 A Generated Feature Used for Testing Different Genetic Parameters

The only parameter left to be checked is the minimum difference between each pair of consecutive discretized values for the independent parameters (DIFF). Figure 4 .34 shows an example of a randomly generated cylindrical feature which is used for studying of genetic parameters. The feature shown in Figure 4 .34 is optimized for the perfect form envelope using different DIFF values and a population size of 30 chromosomes.

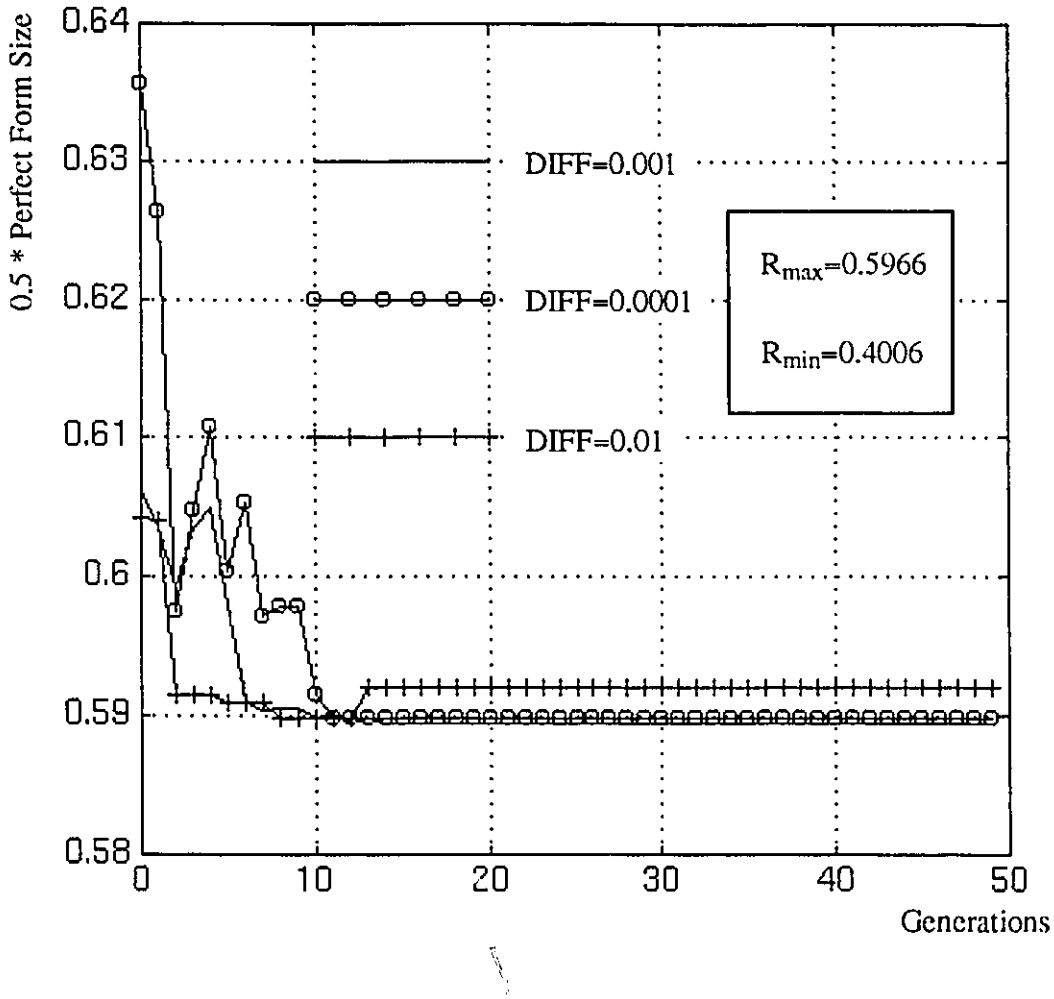


Figure 4 .35 Evaluation of Perfect Form Size for Different Chromosome Lengths

The value of DIFF=0.01 resulted in a an early loss of chromosomes with low objective function value. On the other hand the optimization with DIFF=0.0001 converged to the same value as the run with DIFF=0.001. The coefficient C_a is constructed to evaluate the

chromosome length of a geometric deviation problem:

$$C_a = \frac{0.001}{0.5996 - 0.4006} = 0.005 \qquad (4.19)$$

Thus for any new problem given (R_{max}–R_{min}) the DIFF value is evaluated by

$$DIFF = (R_{max} - R_{min}) \times C_a \qquad (4.20)$$

Knowing the three values of MIN, MAX and DIFF, the chromosome length can be evaluated as shown in section **A.1 .1** (Appendix **A**).

4 .10 .2 Population Size

Population size is the number of chromosomes (instances of the independent parameters) used in the genetic search (Section **A.3** , Appendix **A**). Large population sizes require more computations than smaller ones. On the other hand, extremely small population sizes lead to a larger possibility of early loss of chromosomes with low objective function value. Figure 4 .36 shows two tests for the evaluation of perfect form size of the cylindrical feature shown in Figure 4 .34 with population sizes equal to 10 and 20 chromosomes. Both runs were conducted with DIFF=0.001.

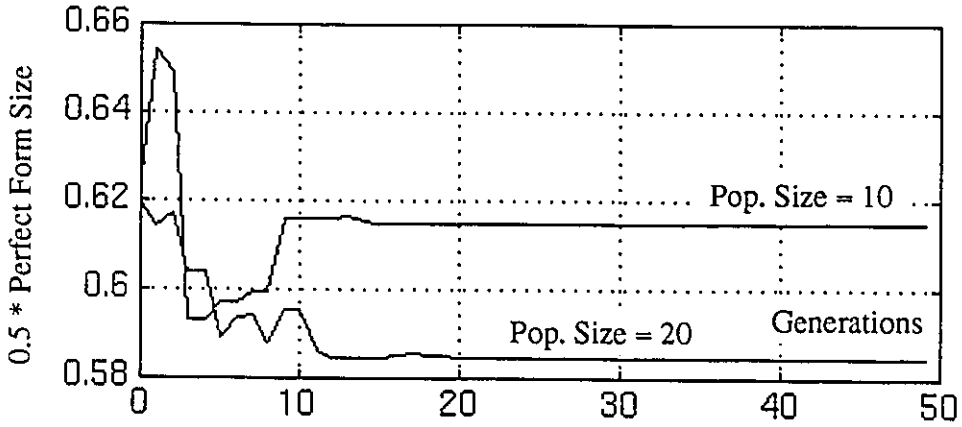


Figure 4 .36 Evaluation of Perfect Form Size for Different Population Sizes

Results show that a very low population size (10) led to an early loss of low objective function value chromosomes.

4 .11 Point Density

The number of points used to represent a feature affects the evaluated values of geometric deviations. To investigate the effect of the number of points used per unit length, the following example is used Figure 4 .37 . A cylinder having a height equal 1 inch and a mean radius equal to 0.5 inches is tested for the perfect form envelope size deviation. Let N be the number of points on the cylinder's periphery, and M be the number of points along the cylinder's length. Random values for the points radii are generated from a normal distribution with a standard deviation equal to 0.2/6.0 inches (i.e. most radii would vary from 0.4 to 0.6 inches).

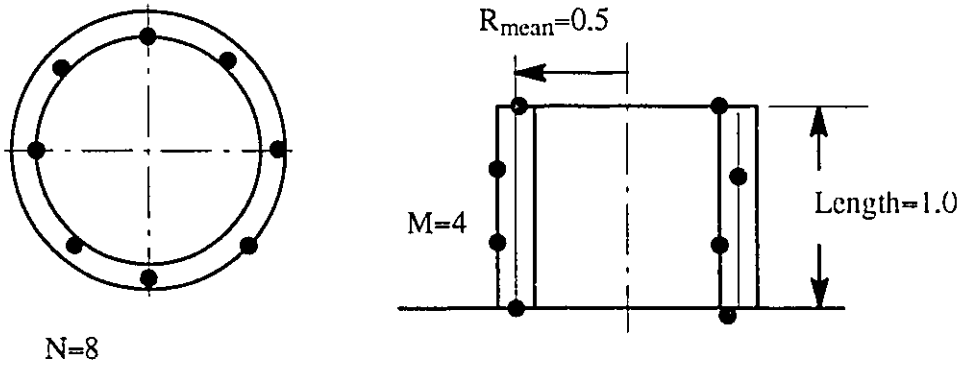


Figure 4 .37 Example for Point Density

Given the values of N × M, the radii are generated and the feature's perfect form size is evaluated. The process is repeated 200 times and the mean perfect form size is evaluated. The process is repeated for different values of N (M is taken to be equal to N–2), and the result is shown in Figure 4 .38 .

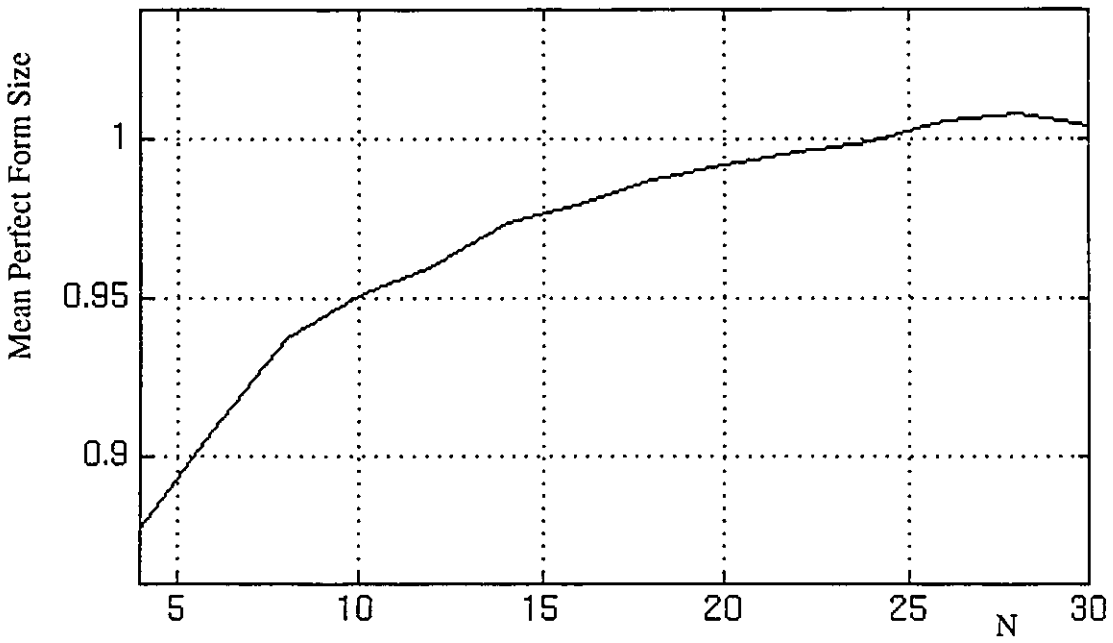


Figure 4 .38 Effect of Number of Points on the Mean Perfect Form Size Value

Figure 4 .38 shows that with the increased number of points representing the surface, the mean value of the evaluated deviation becomes semi–constant. Since the mean size is equal to 1 inch. Hocken e al. (1993) recommend a minimum number of 10 points to represent a cylindrical feature.

4 .12 Conclusions

The chapter presented a number of algorithms for the evaluation of geometric deviations on manufactured features. Most of these algorithms used genetic algorithms as function optimizers while some used the NURBS surface interpolation presented in **Chapter 3** as an aiding tool. Previous work in the subject of evaluating geometric deviations used optimization methods which did not guarantee reaching the global minimum value of the deviations. The geometric deviations evaluated by those methods were larger than the real ones, and hence led to unnecessary rejection of good parts. A comparison between the

formulation based on genetic algorithms, used as global optimizers, and methods used in previous research was presented and the superiority of the GA–based algorithms was demonstrated. The presented work was also motivated by another issue: The new ANSI Y14.5.1M standards have a new definition for size tolerance whose deviation cannot be easily evaluated using the measured points directly. The algorithms presented in this chapter have two novel aspects. The first is the use of parametric surfaces, interpolated to the measured points, that made the evaluation of the newly defined size deviation possible and consequently the actual centre–line of the manufactured feature could be derived. Second, the use of genetic algorithms as global function optimizers avoids local minima and hence parts are not rejected unnecessarily. Since genetic algorithms are more computationally intensive than other methods, a study was conducted to choose the best genetic parameters and population size (**Appendix A**) which would lead to the minimum number of computations. Although the algorithms presented concentrated on cylindrical features, the objective functions of the various algorithms could be easily modified for other types of features of size (pair of opposite and parallel planes and spheres).

CHAPTER FIVE

ASSEMBLY BUILD–UP

The evaluation of the probabilities of accepting or rejecting an assembly of manufactured parts is a crucial step in tolerance allocation. In the case of assemblies with dimensional tolerances these probabilities are evaluated using a function, that relates the functional requirements to the imposed tolerances, known as the stack–up function. In the case of assemblies with geometric tolerances, the development of a stack–up function is not as straightforward as in the case of assemblies with only dimensional tolerances. This chapter describes the methods by which a stack–up function for an assembly with geometric tolerances is formulated. The algorithms developed in this chapter are used within the method described later in Chapter Six for the evaluation of probabilities of acceptance and/or rejection of an assembly of parts.

5 .1 Introduction

Figure 5 .1 shows the steps through which parts pass to produce an assembly. These steps are:

1. **Manufacturing Step:** Manufacturing processes generate the surfaces of the part features. The dimensions and geometries of the generated surfaces are not exact due to the errors inherent in the manufacturing processes. Each process is associated with a cost value. Another cost value is associated with the fixturing set–up of the part. The set–up is dictated by the manufacturing sequence where one feature is manufactured before the other.

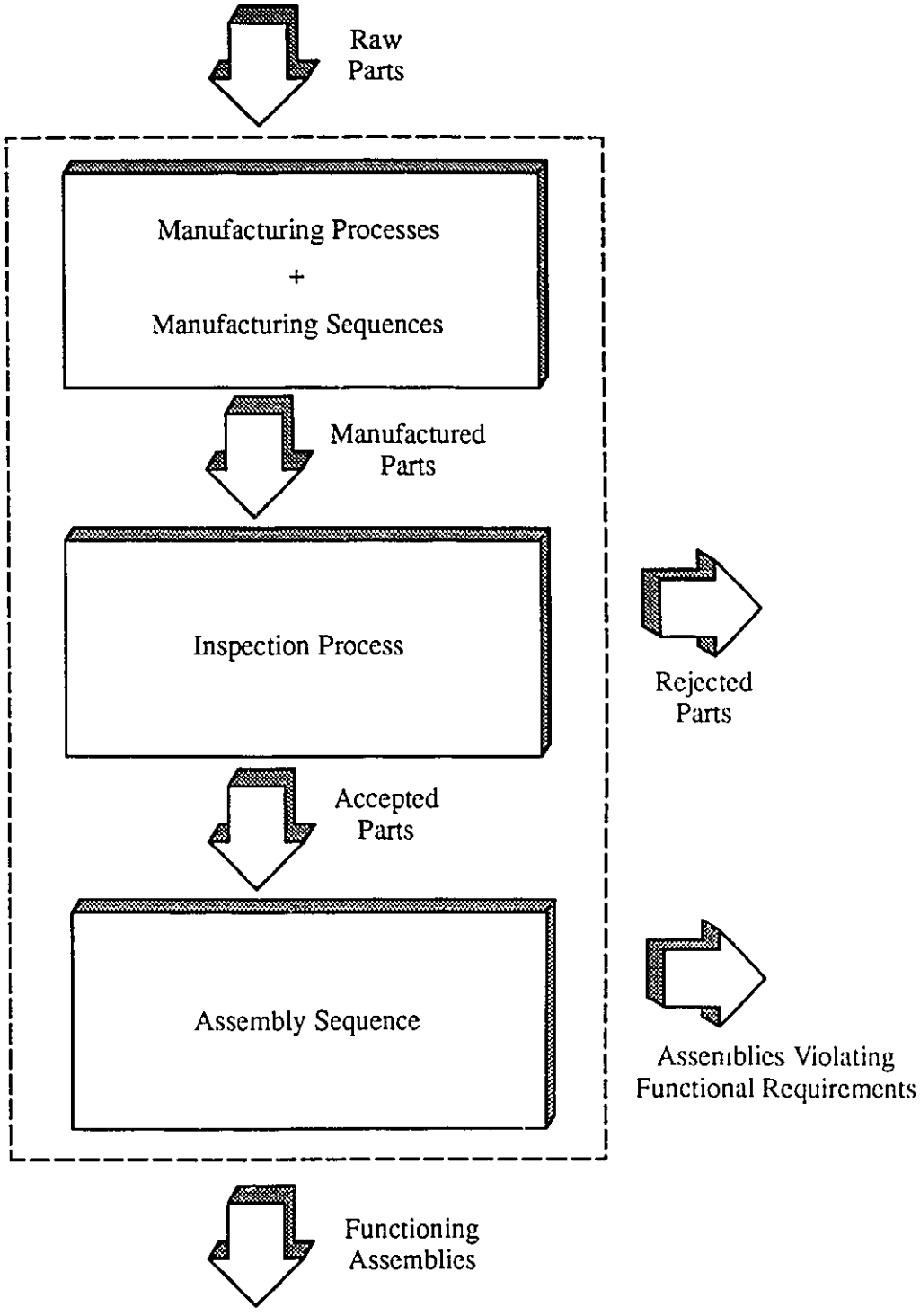


Figure 5 .1 Typical Manufacturing and Assembly Steps

2. **Inspection Step:** The manufactured parts are inspected for the dimensions and tolerances imposed during the design phase. Since it is hard to check the functional requirements after the assembly process, tolerances are imposed on the individual parts such that a "bad" part (a part that will cause a violation of assembly functional requirements when assembled to other parts) is rejected during the inspection step.

3. **Assembly Step:** Parts which successfully pass the inspection process are assembled. If the tolerances were not properly specified and bad parts were not rejected during the inspection step, a number of assemblies violating the functional requirements will result.

Therefore, the input for a stack-up function is the set of generated (either measured or simulated) surfaces, a manufacturing sequence, a tolerance specification and an assembly sequence. Its output is an integer indicating whether the set of parts is or is not rejected during inspection, and whether the assembly does or does not violate the functional requirements.

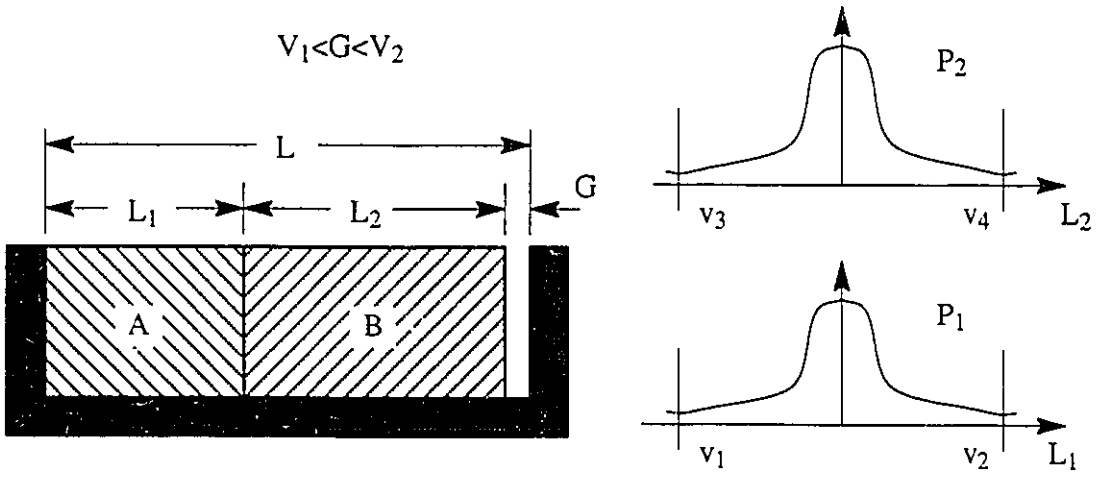


Figure 5 .2 An example of a simple assembly of three parts

The development of a stack–up function for an assembly with dimensional tolerances is demonstrated by the example shown in Figure 5 .2 . Here it is seen how the above steps are reflected in the function development. The parts A and B are produced by two manufacturing processes P_1 and P_2 . Due to the errors inherent in the manufacturing processes, the dimensions L_1 and L_2 of the produced parts range from a lower limit (v_1 and v_3) to a higher one (v_2 and v_4) and according to a probability distribution for each dimension defined for its whole range of values. The limits on the clearance G are the functional requirements. G is related to the dimensions L_1 and L_2 by the equation:

$$G = L - L_1 - L_2 \qquad (5.1)$$

Let two tolerance limits be associated with each dimension (t_1 and t_2 for L_1 and, t_3 and t_4 for L_2). These limits are used during inspection for the identification of bad parts. Previous work in tolerance analysis and synthesis considered the tolerance limits to be equal to the processes limits (i.e. $t_i = v_i$). The stack–up function for the assembly shown in Figure 5 .2 is described by the following steps:

1. **Manufacturing Step:** Using the probability distribution of the manufacturing processes P_1 and P_2, generate two instances of the dimensions L_1 and L_2.

2. **Inspection Step:** If L_1 or L_2 falls outside the tolerance limits, then set an output value indicating that the rejection is due to the violation of tolerance specification; otherwise, go to step 3.

3. **Assembly Step:** Calculate the value of G using equation (5 .1). If G falls outside its designed limits, then set output value indicating that the functional requirements are violated; otherwise, indicate indicate an acceptable assembly.

The above steps form the main outline for developing a stack–up function for geometric tolerances as well except for the following differences:

1. **Manufacturing Step:** The generation of a manufactured dimension is a straightforward process. In the case of assemblies with geometric tolerances the generation of a manufactured surface would be needed. With geometric tolerances, certain surfaces are manufactured using other surfaces as datum reference frames. Hence the manufacturing sequence should be followed, to generate the surfaces used as datums before other surfaces are produced.

2. **Inspection Step:** A single feature can have more than one geometric tolerance imposed on it. The algorithms described in Chapter Four should be used to check for the geometric deviations on a manufactured surface.

3. **Assembly Step:** The development of a closed form mathematical equation relating the functional requirements to the manufactured surface is difficult. Hence, the assembly sequence by which parts are assembled together should be simulated, and the functional requirements checked at the end.

The following sections describe the various procedures needed in each of the above mentioned steps for developing a stack–up function for assemblies with geometric tolerances.

5 .2 Manufacturing Step

5 .2 .1 Generation of Parts Surfaces

A manufactured part can be represented by a set of points generated from the manufacturing process probability distribution (Figure 5 .5). For example the feature shown in Figure 5 .5 is represented by 8 radial $\times$ 3 axial points measured from the datum reference frame **A**,**B** and **C** (Figure 5 .4), where the points' radii are the random variables. The radii vector $\boldsymbol{R}$ is generated from the probability distribution representing the random variation of the machined surface. For an existing part, such points would be obtained by

measurement. Hence, simulation here means generating points on the manufactured surfaces typical of those produced by the manufacturing process. It is assumed that points on the manufactured (or generated) surface fall between two ideal offset surfaces which form the limits of the manufacturing process. The position of any point on the surface is determined by three coordinates, one of which varies along the offset distance. In the case of a cylindrical feature these coordinates are z,θ,r (Figure 5 .3). The radius varies between the two offset surfaces of the process limits, following the machining process probability distribution. The same methodology is applicable to other types of features of size. The emphasis in this section is on cylindrical features as it is one of the most common features and is used in the examples given in subsequent chapters. The generated points are measured from the feature's frame of axes which is defined by three points $\mathbf{p}_1$, $\mathbf{p}_2$, and $\mathbf{p}_3$, where the distance $|\mathbf{p}_1\ \mathbf{p}_2|$ equals the feature's height and the distance $|\mathbf{p}_1\ \mathbf{p}_3|$ equals the feature's nominal radius.

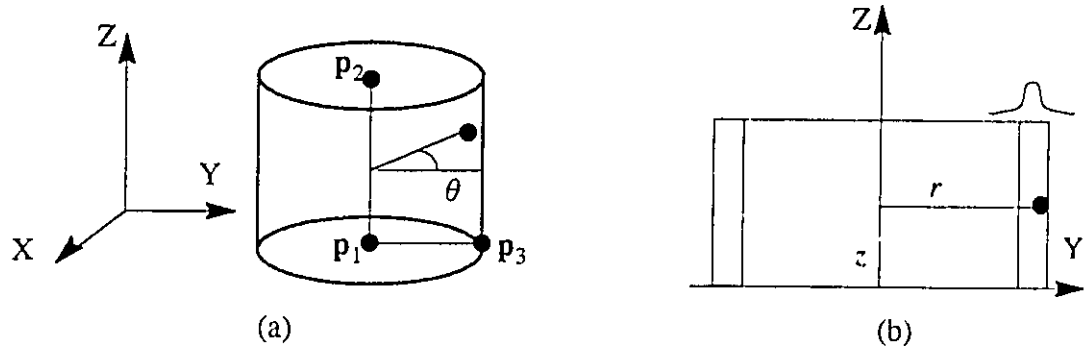


Figure 5 .3 Feature Generation

Given a probability distribution $f(r)$ and a probability function $F(r)$ of a random radius r, random samples of r can be generated using the inverse transform method (Rubinstein, 1981):

1. Generate a random number U between [0,1]
2. Find $r=F^{-1}(U)$.

If the inverse function $F^{-1}(U)$ cannot be derived (e.g. for normal distribution), then the value

of r can be found using a dichotomous search (Reklaitis et al., 1983) as shown in the following procedure:

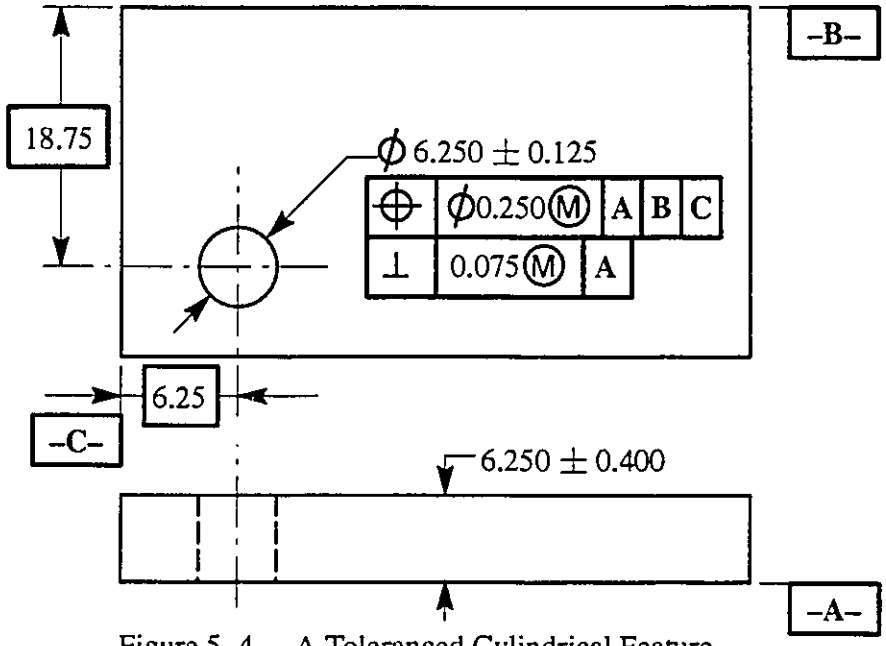


Figure 5 .4 A Toleranced Cylindrical Feature

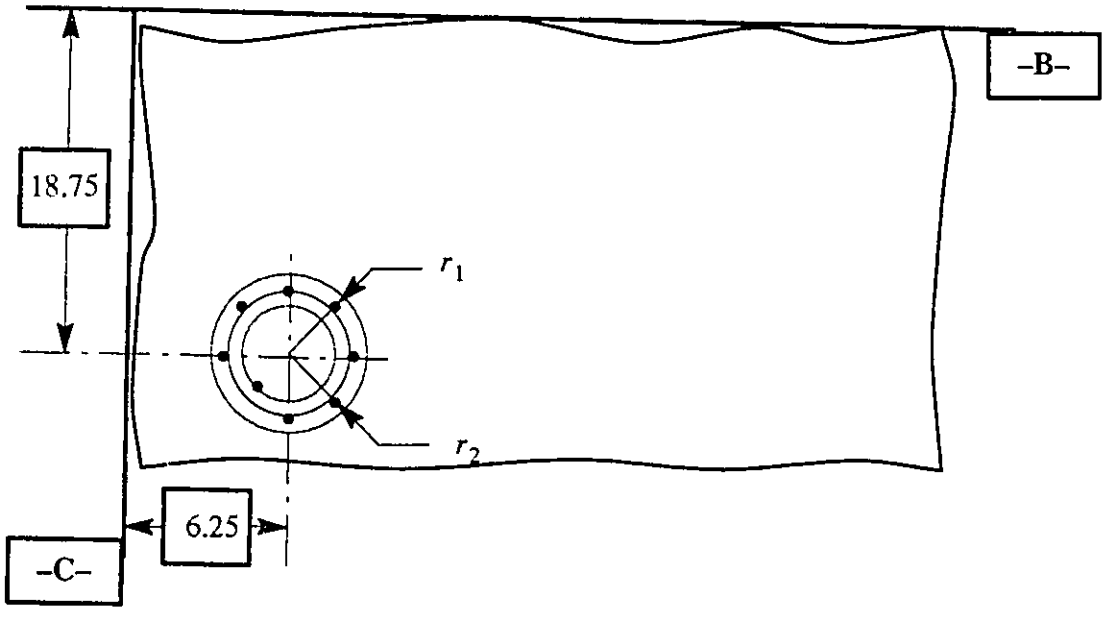


Figure 5 .5 Feature Generation

1. Calculate r_0 corresponding to $F \simeq 0$, r_1 corresponding to $F \simeq 1$ and $r_m=(r_0+r_1)/2$.

2. **repeat** until convergence

   a. **if** $F(r_m) > U$ then

   $r_1 = r_m$

   $r_m=(r_0+r_1)/2$.

   else

   $r_2 = r_m$

   $r_m=(r_0+r_1)/2$.

   end–if

   end–repeat

5 .2 .1 .1 Manufacturing Process Probability Distribution

Previous work in dimensional tolerance analysis assumed that the random variables (the dimensions) are statistically independent. However, the statistical independence might not be valid for points measured on manufactured parts' surfaces. The multinormal distribution can be used to represent the manufacturing variability, taking into account the covariance between the random dimensions (radii in the case of a cylindrical feature) of the points representing the manufactured part. The parameters of the manufacturing process probability distribution (mean value and standard deviation of radii) are usually obtained by taking several measurements of points on the actual parts and estimating the population parameters. Thus, if a manufactured cylindrical feature is represented by M points and N parts were manufactured, then the variance of the measured radii would be equal to:

$$\mathrm{VAR}(r) = \frac{\left[\sum_{i=1}^{N}\sum_{j=1}^{M}(\mathrm{R}_{i,j} - \overline{\mathrm{R}})\right]^2}{N \times M - 1} \tag{5.2}$$

where $\mathrm{R}_{i,j}$ is the jth generated radius of the ith manufactured part, and

$$\overline{\mathrm{R}} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M}\mathrm{R}_{i,j}}{N \times M} \tag{5.3}$$

The same measurements can be used to calculate the covariance matrix of the generated points

$$\mathbf{C} = \begin{bmatrix} \sigma_{1,1} & \sigma_{1,2} & \cdots & \sigma_{1,n} \\ \sigma_{2,1} & \sigma_{2,2} & \cdots & \sigma_{2,n} \\ \vdots & \vdots & \vdots & \vdots \\ \sigma_{n,1} & \sigma_{n,2} & \cdots & \sigma_{n,n} \end{bmatrix} \tag{5.4}$$

where

$$\sigma_{i,j} = \frac{\sum_{k=1}^{N}(\mathrm{R}_{i_k} - \overline{\mathrm{R}}_i)\ (\mathrm{R}_{j_k} - \overline{\mathrm{R}}_j)}{N - 1} \tag{5.5}$$

and

$$\overline{\mathrm{R}}_i = \frac{\sum_{k=1}^{N}\overline{\mathrm{R}}_{i_k}}{N} \tag{5.6}$$

The multinormal distribution can then be used to represent the random nature of the manufacturing process, taking into account the random variables' dependence.

$$f_r(r) = \frac{1}{(2\pi)^{n/2}|\mathbf{C}|^{1/2}} \exp\left[-\frac{1}{2}(r - \overline{\mathbf{R}})^{\mathrm{T}}\mathbf{C}^{-1}(r - \overline{\mathbf{R}})\right] \quad (5.7)$$

where:

$r = (r_1, r_2, \ldots r_n)$ is the random vector of the feature's radii (5 .8)

$\overline{\mathbf{R}} = (\overline{R}_1, \overline{R}_2, \ldots \overline{R}_n)$ is the vector of distribution means (5 .9)

5 .2 .1 .2 Feature Generation from a Multinormal Distribution

The simulation of the manufactured parts depends on the type of the distribution representing the random variation in the process leading to the geometric deviations. If the manufactured surface points can be considered independent random variables, then the inverse transform method (Rubinstein, 1981) can be used for the generation of surface points. If the variation is represented by a multinormal distribution (5 .7), then the following procedure is used (Rubinstein, 1981).

Let:

$$\mathbf{c} = \begin{bmatrix} c_{1,1} & 0 & \cdots & 0 \\ c_{2,1} & c_{2,2} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ c_{n,1} & c_{n,2} & \cdots & c_{n,n} \end{bmatrix} \quad (5.10)$$

be the lower triangular matrix of the covariance matrix $\mathbf{C}$ such that :

$$\mathbf{C} = \mathbf{c}\mathbf{c}^{\mathrm{T}} \quad (5.11)$$

then the vector R of the simulated points is equal to:

$$R = \mathbf{c}\mathbf{Z} + \overline{\mathbf{R}} \quad (5.12)$$

where **Z** is an independent generation of a random vector from the standard normal distribution. Thus the algorithm (Rubinstein, 1981) for generating points from the multinormal distribution is:

1. Generate $\mathbf{Z} = (Z_1, Z_2, ..., Z_n)$ (5 .13)
2. Calculate

$$c_{ij} = \frac{\sigma_{ij} - \sum_{k=1}^{j-1} c_{i_k} c_{j_k}}{\left[\sigma_{jj} - \sum_{k=1}^{j-1} c_{j_k}^2 \right]^{1/2}} \qquad (5 .14)$$

where :

$$\sum_{k=1}^{0} c_{i_k} c_{j_k} = 0, \qquad 1 \le j \le i \le n$$

3. $R = \mathbf{c}Z + \overline{\mathbf{R}}$ (5 .15)

5 .2 .1 .3 Actual Axis of Features

Once the points representing a feature of size are generated, the feature's actual axis can be derived. The optimization function described in section **4 .5 .2** for the evaluation of the local size deviation can be used to locate points on the actual axis at different cross-sections. The end points of the actual axis are used to define the actual orientation of the feature in space. This orientation is later used in the assembly process when dealing with shrink fits (section **5 .4 .2 .1**).

5 .2 .1 .4 Global (Assembly) Axes

The points generated for the feature shown in Figure 5 .5 are measured from the datum reference frame **A, B** and **C** (Figure 5 .4), as specified by the position tolerance of the feature. Hence the coordinates of the feature's generated points are measured from that

frame of axes. Since every feature can have different frames of axes according to its position tolerance specification, a frame of axes has to be specified for the whole part to which all points on different features are to be referred (Figure 5 .6). The coordinates of these points are then transformed to a global frame of axes when the assembly sequence is simulated.

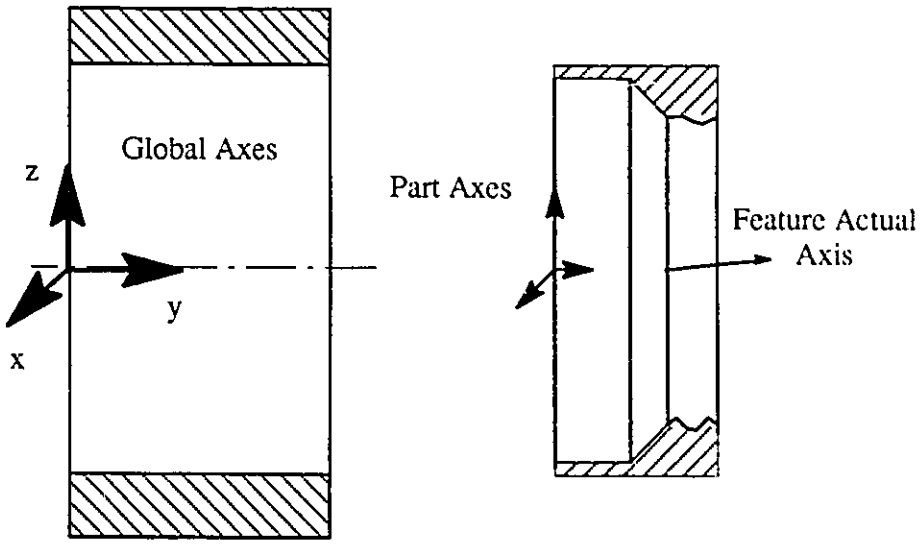


Figure 5 .6 Assembly, Part and Feature Axes

5 .2 .2 Simulation of the Manufacturing Sequence

The sequence in which the part features are manufactured depends on the geometric tolerances imposed on the part. Figure 5 .7 shows a bushing controlled by a set of geometric tolerances. The part should be machined in the following sequence (illustrated in Figure 5 .8):

1. The surface S1 is referenced by datum **D** which is used with the tolerances specified in other features (position tolerance of feature F1) and at the same time is not toleranced with a type of tolerancing that refers it to another datum. The

same argument applies to the cylindrical surface S2 and to the planar surface S3. Since the mentioned surfaces are used as datums in other tolerances, they should be machined first.

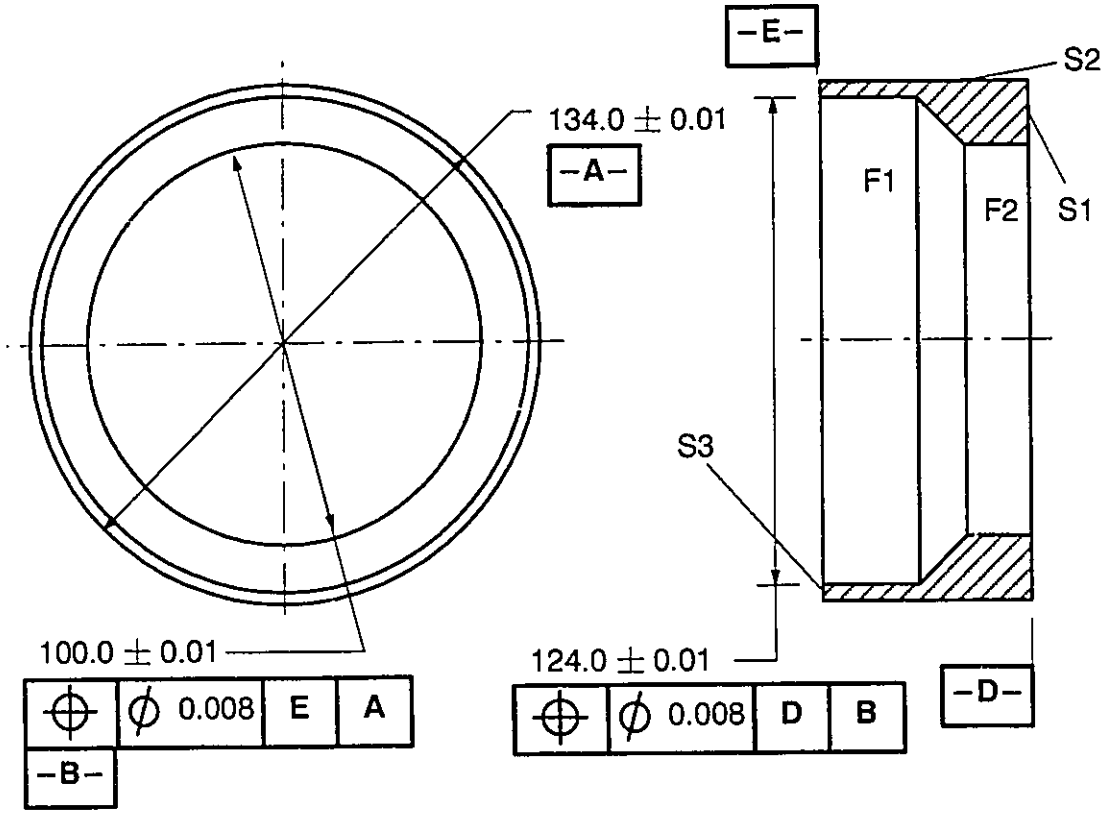


Figure 5 .7 A Toleranced Bushing

2. Feature F2 comes second in the machining sequence as the surfaces used to establish its position tolerance datums are machined in step 1.

3. Once feature F2 is machined, it can be used to establish datum **B** which is in turn used in the position tolerance for feature F3, hence F3 comes last in the machining sequence.

The given example differentiates between two types of surfaces: (1) those which are not controlled with tolerances that refer them to other surfaces and (2) those surfaces which

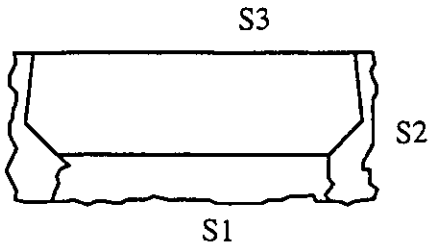
are controlled by tolerances that refer them to other datums. This differentiation is used by the following algorithm which derives the manufacturing sequence from the specified tolerances.

Procedure: MCSEQ

1. Place all surfaces and features of type (1) in a list called FREE.
2. Place all surfaces and features of type (2) in a list called TOLERANCED.
3. Generate all the surfaces and features in FREE and place them in a list called GENERATED.
4. **for** every feature in TOLERANCED
    1. **if** TOLERANCED is empty **then** exit loop.
    2. **if** the feature is referred to datums derived from surfaces or features found in GENERATED **then**
        1. Generate the feature, then remove it from TOLERANCED and add it to GENERATED.

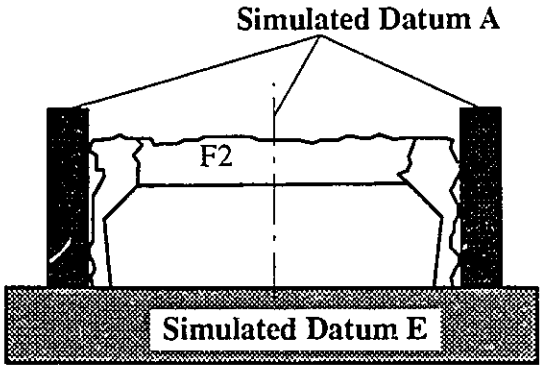
    end–if

end–for



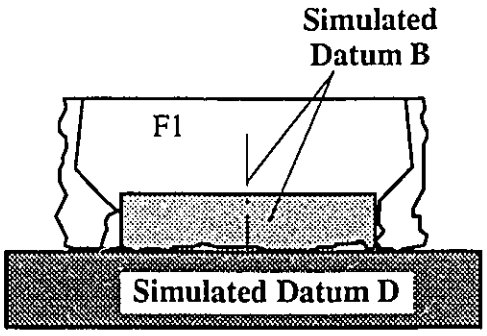
Step 1

1. Surfaces S1, S2 and S3 are machined with no reference to any datums.



Step 2

2. Surface S3 is placed on a theoretically exact plane to simulate datum **E**, while datum **A** is established by finding the least inscribing exact cylinder to surface S2. The part is then clamped and feature F2 is generated.



Step 3

3. Surface S1 is placed on a theoretically exact plane to simulate datum **D**, while datum **B** is established by finding the largest circumscribing exact cylinder to the feature F2. The part is then clamped and feature F1 is generated.

Figure 5 .8 Machining Sequence for Part in Figure 5 .7

5 .3 Inspection Step

Having the parts surfaces generated in the form of points, their geometric deviations have to be evaluated and checked for the specified tolerances. The algorithms of these evaluations have been shown in Chapter Four.

5 .4 Assembly Step

5 .4 .1 Simulation of the Assembly Sequence

The assembly sequence of the machined part needs to be simulated, in order to check a critical clearance or dimension in an assembly for a specified value. The simulation is performed using the assembly sequence graph described in the next section.

5 .4 .1 .1 Assembly Sequence Graph

Figure 5 .9 shows an assembly sequence graph for a simplified punching machine and Figure 5 .11 shows another graph for the speed reducer (Niemann, 1975) detailed earlier in Figure 5 .10 . The nodes of the graph are the parts included in the assembly. Each part has a set of attributes associated with it:

1. A number identifying the part. A part numbered "1" is the part with which the assembly sequence begins.
2. A set of sub–nodes representing the part's features and/or surfaces.

The links represent either the physical connections (solid lines) between the features, or the functional requirement between each feature and the other (dashed links). Every connection has the following attributes:

1. TYPE: A number indicating the type of connection as shown in Table 5 .1 .

Connection Type	TYPE
Shrink Fit (2 Features)	1
Shrink Fit (4 Features)	2
Bolted Connection	3

Table 5 .1 Connection Types

2. PRECEDENCE: A number indicating the priority of performing the connection in the assembly sequence. Connections with PRECEDENCE=1 are to be performed first in the sequence.

The links representing the functional requirements have the following attributes:

1. CLASS: A number indicating the type of functional requirement between features. CLASS=1 indicates a restriction on a clearance and CLASS=2 indicates a restriction on a dimension.

2. A list of restrictions on the value of the functional requirement. For example, a clearance which should range from 0.05 mm to 0.10 mm will have two restrictions: one for the "less–than" relationship and another for the "greater–than" relationship.

The punching machine assembly in Figure 5 .9 has one critical clearance. Insufficient clearance may cause short tool life and distortion of the stamped part, while excessive clearance will cause excessive burrs on the manufactured part's edge. The speed reducer assembly shown in Figure 5 .10 has four critical dimensions representing the centre–distance between the upper and lower shafts. Excessive deviation of the centre–distance from the nominal value will increase backlash and gear noise.

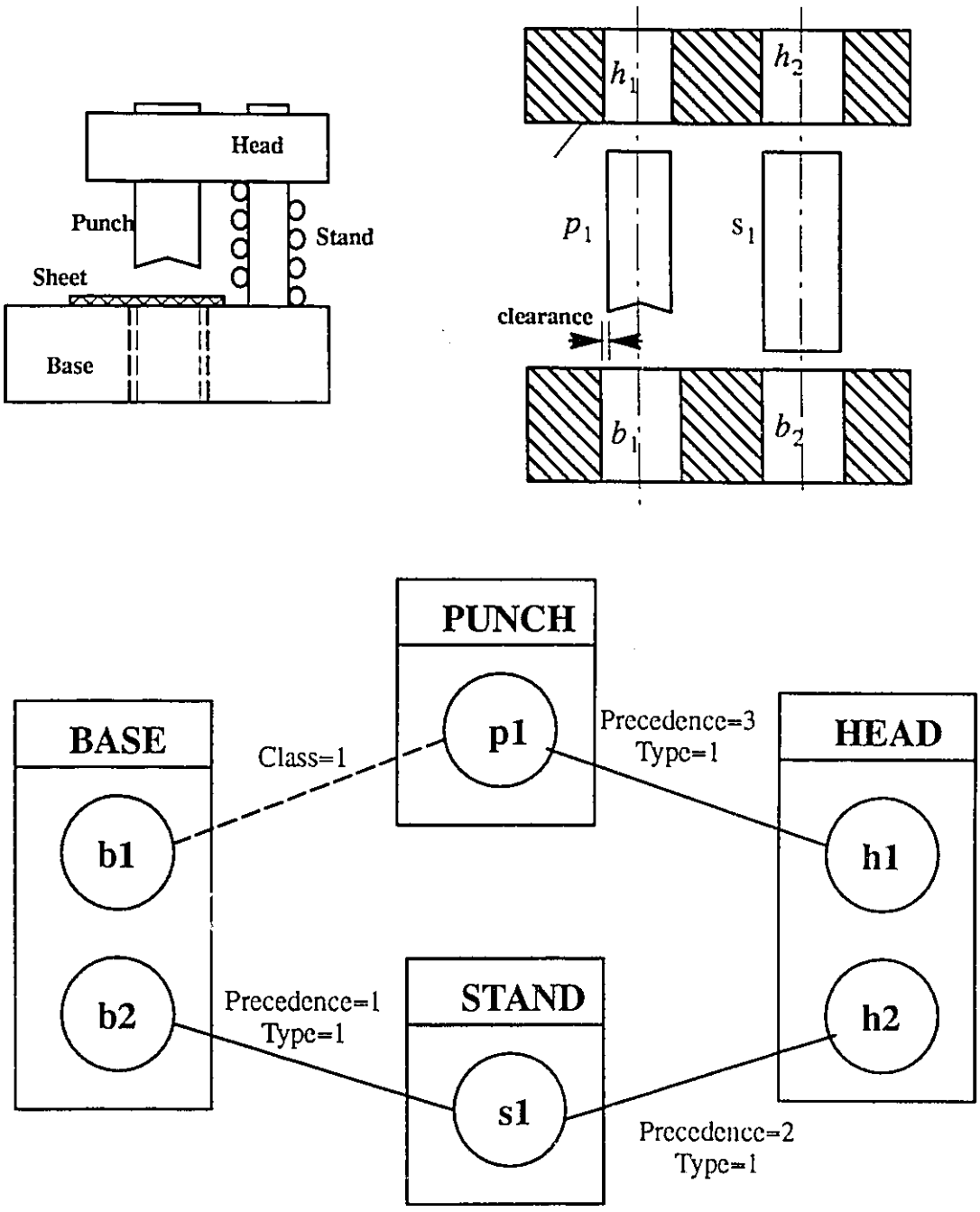


Figure 5 .9 Assembly Sequence Graph of a Punching Machine

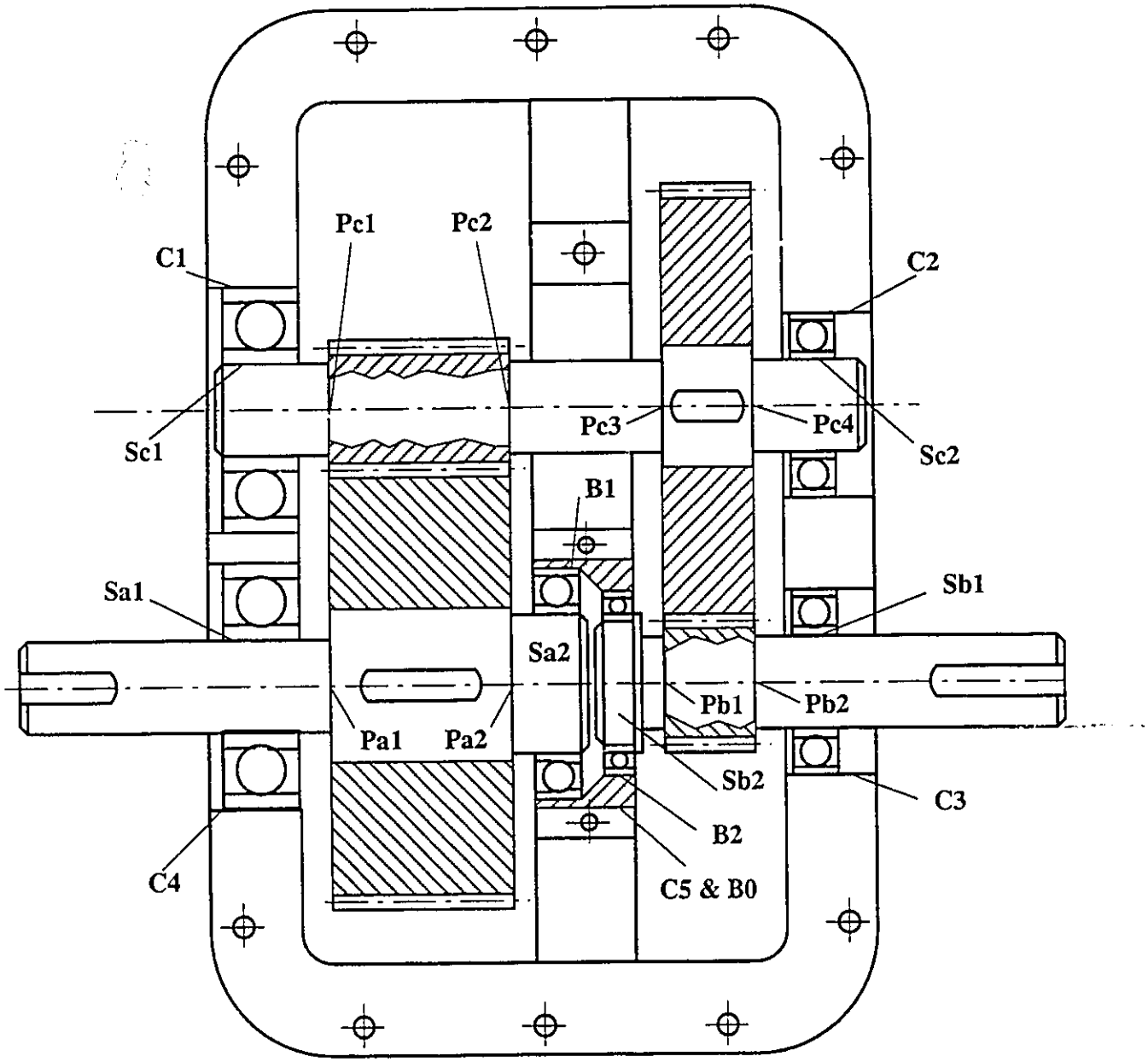


Figure 5 .10 A Speed Reducer

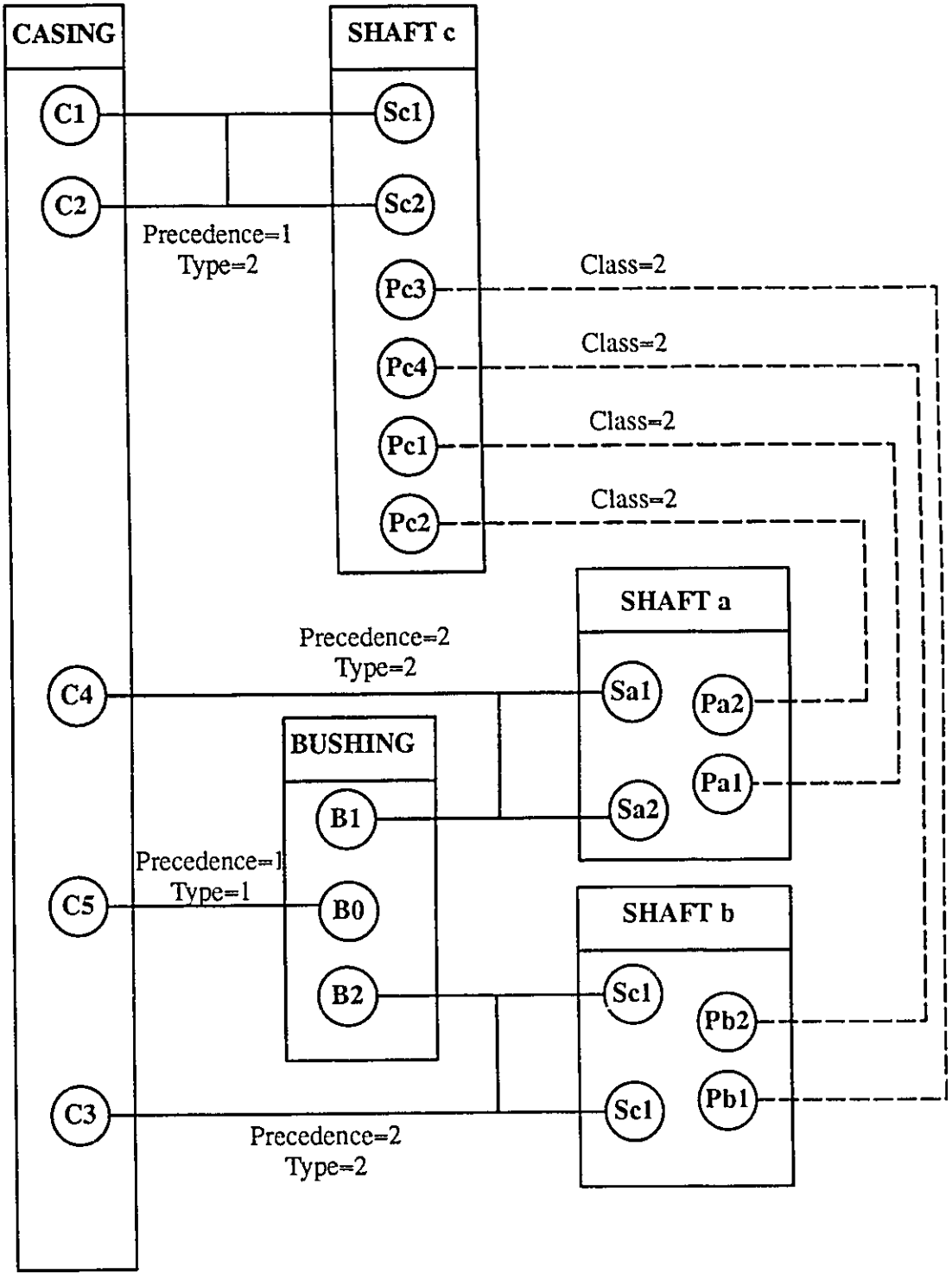


Figure 5 .11 Assembly Sequence Graph for a Speed Reducer

5 .4 .1 .2 Assembly Sequence Simulation Algorithm

The objective of the assembly sequence simulation algorithm is to find for each manufactured part its actual position in space relative to a set of global axes (Figure 5 .12), according to its method of connection to other parts.

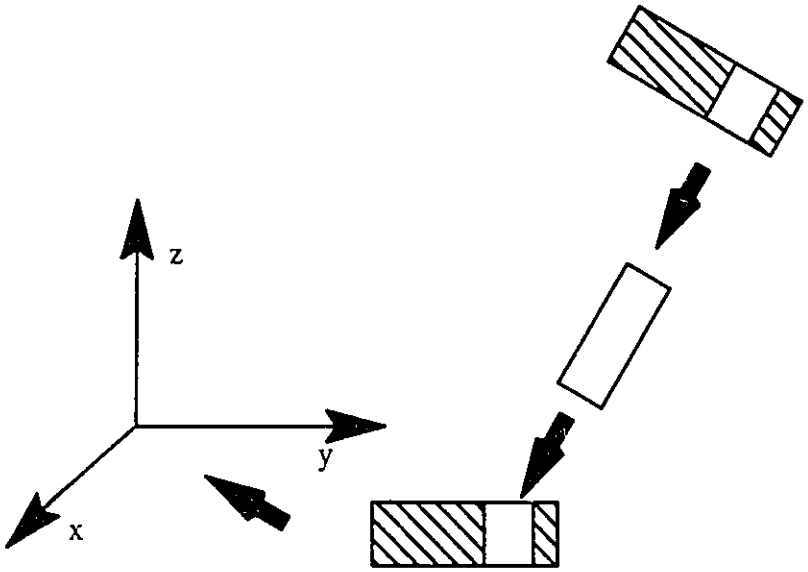


Figure 5 .12 Assembly Sequence and Global Axes

The following variables and lists are used by the algorithm:

1. POS: A list of parts already positioned in the assembly relative to the global axes.
2. PEND: A list of parts pending for positioning in the assembly.
3. RANK: A number indicating the active priority number, and whose connections are to be performed.

Given the above mentioned variables and lists, the assembly sequence simulation algorithm proceeds as follows:

Procedure: ASMSEQ

1. Position part number 1 at the global axes
2. Place Part number 1 in POS and the other parts in PEND.
3. RANK=1, **repeat**

   if no connection has a PRIORITY equal to RANK

   stop **repeat.**

   else

   for every connection having PRIORITY=RANK

   a. Position all pending parts associated with the connection according to the connection type.

   b. Remove the parts used in (a) from PEND and place them in POS.

   end–for

   end–if

end–repeat

Part positioning in space relative to the global axes depends on the types of connections by which the manufactured parts are joined together. Types of connections are discussed in the following sections.

5 .4 .2 Types of Connections

The following sections describe a classification of connections between manufactured parts. Although this classification does not cover every method of joining parts together, the described connections cover a wide range of application. The first two described connections are used in the examples in Chapters Six and Seven.

5 .4 .2 .1 Shrink Fit Connections (Two Features)

Shrink and transition fits are established physically by either of two methods: (1) the hole is heated, thus its diameter expands, and then the shaft is placed in it and the fit is established by the hole's contraction upon the shaft, or (2) the shaft is pressed into the hole. For the purpose of simulating the connection it is supposed that the shaft's actual axis assumes the same orientation in space as the hole's actual axis in both cases. Figure 5 .13 shows two parts P1 and P2 to be fit together, at features F1 and F2 respectively.

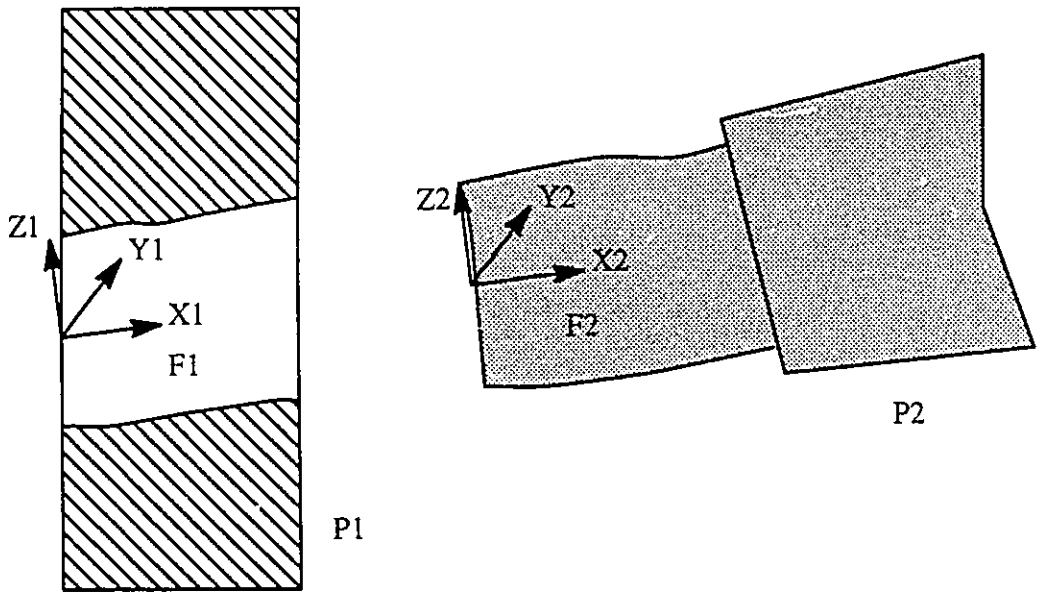


Figure 5 .13 Shrink Fit Connections (Two Features Involved)

Assuming that part P1 is already positioned with respect to the global axes and that P2 is pending for positioning, P2 is repositioned in space such that F2's frame of axes (X2,Y2,Z2) coincides with F1's frame of axes (X1,Y1,Z1). The transformation matrix **T** between both frames is evaluated using the procedure described in section **B.4** . Any point **p** on the part P2 is relocated to a new position $\mathbf{p}^* = \mathbf{p}\,\mathbf{T}$.

5 .4 .2 .2 Shrink Fit Connections (Four Features)

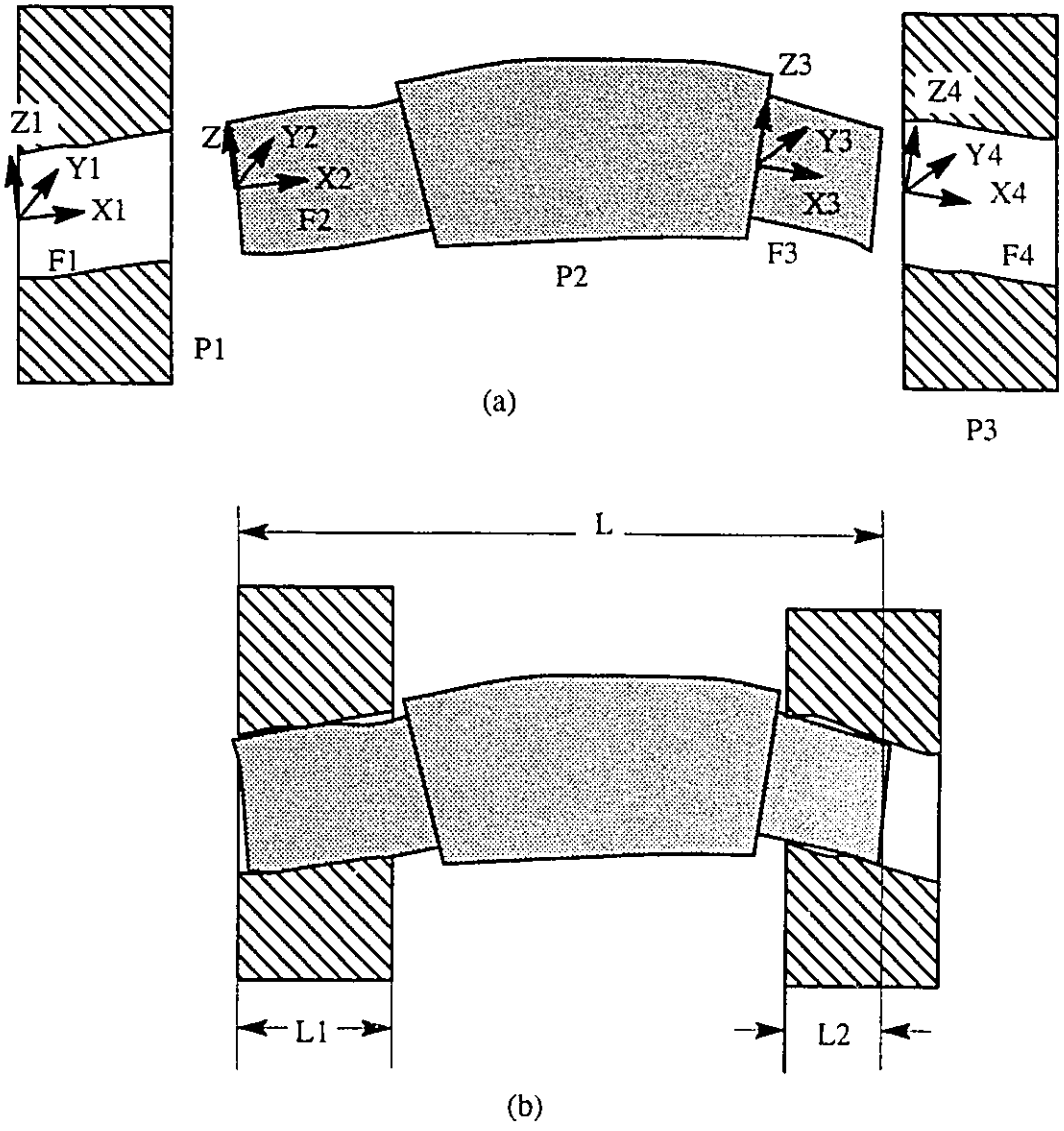


Figure 5 .14 Shrink Fit Connections (Four Features Involved)

This kind of connection joins two (or three) parts. An example is a shaft joined to a pair of ball bearings which are in turn fit into two holes in one or two parts. The speed reducer shown in Figure 5 .10 shows three examples of such a connection. The following

paragraphs describe the method by which this connection is simulated for assembly. The shaft rotation is not taken into account during simulation. This type of connection is physically established in the same manner as the previous type, except that the shaft axis gets an extra bend due to the misalignment of the two holes (Lateral deformation in the shaft is assumed to exceed that of the holes as the shaft flexibility is much higher). The two shaft features which are fit into the holes are assumed to acquire the same position and orientation as their respective holes they are fit into. The extra bend in the shaft axis depends on the shaft's material and cross--section size, but the discussion of these factors is beyond the scope of this dissertation. Figure 5 .14 shows an example of a connection where parts P1 and P3 are already placed in the assembly and part P2 is pending for assembly. Feature F2 is to be fit into feature F1 while F3 is to be fit into F4. The connection is established by the following steps:

1. Finding the transformation matrices $\mathbf{T}_1$ and $\mathbf{T}_2$ fitting F2 to F1 and F3 to F4 respectively.

2. Having the end features F1 and F2, approximate the shaft bend by a parametric equation using the following procedure:

Figure 5 .15 Approximation of the shaft's axis

Given the shaft's axis end points $\mathbf{p}_1 = \mathbf{p}(u=0)$ and $\mathbf{p}_4 = \mathbf{p}(u=1)$ calculate the

axis end tangent vectors:

$$\mathbf{p}_0^u = \frac{d\ \mathbf{p}(0)}{du} \simeq \frac{\mathbf{p}_2 - \mathbf{p}_1}{\mathrm{L1/L}} \tag{5.16}$$

$$\mathbf{p}_1^u = \frac{d\ \mathbf{p}(1)}{du} \simeq \frac{\mathbf{p}_4 - \mathbf{p}_3}{\mathrm{L2/L}} \tag{5.17}$$

The axis equation is approximated by the 3rd degree equation (Mortenson. 1985):

$$\mathbf{p}(u) = F1(u)\ \ \mathbf{p}_0 + F2(u)\ \ \mathbf{p}_1 + F3(u)\ \ \mathbf{p}_0^u + F4(u)\ \ \mathbf{p}_1^u \tag{5.18}$$

where:

$$F1(u) = 2u^3 - 3u^2 + 1 \tag{5.19}$$

$$F2(u) = -2u^3 + 3u^2 \tag{5.20}$$

$$F3(u) = u^3 - 2u^2 + u \tag{5.21}$$

$$F4(u) = u^3 - u^2 \tag{5.22}$$

Equation (5.18) and its first derivative define a locus of frames of axes to which all cross–sections of generated points along the shaft are transferred.

5.4.2.3 Bolted Connections

Bolted connections consist of two parts joined together with a screw bolt. Sections B3 and B4 in the ANSI Y14.5M (1982) standards divide them into two types: (1) fixed fasteners and (2) floating fasteners. A fixed fastener is shown in Figure 5.16 a. The lower member has a screwed hole in which the bolt is fastened. Physically the connection is established by moving the upper part on top of the lower part till a position is found in which

the bolts can be sunk in both holes (Figure 5 .16 b). This connection can be simulated using the following steps:

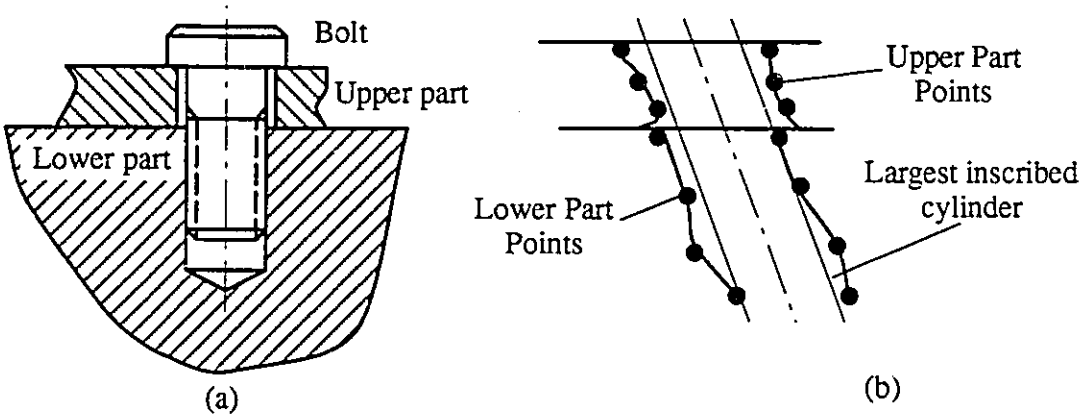


Figure 5 .16 Fixed Fastener

1. Find the largest inscribed cylinder contained within the lower part's hole, minimizing the function shown in Figure 5 .17 . The radius of the cylinder is the radius of the largest bolt that can be contained in the hole of the lower part. If that radius is greater than the MMC of the size tolerance of the lower part, then the connection would be acceptable; otherwise, it is rejected.

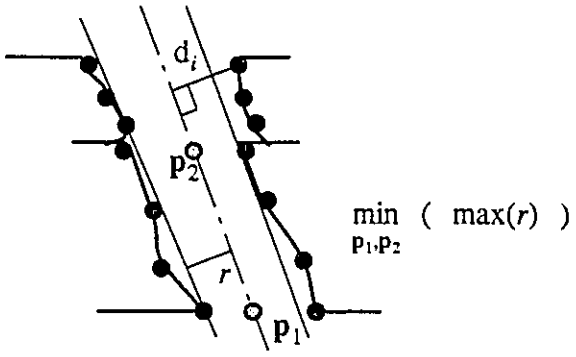
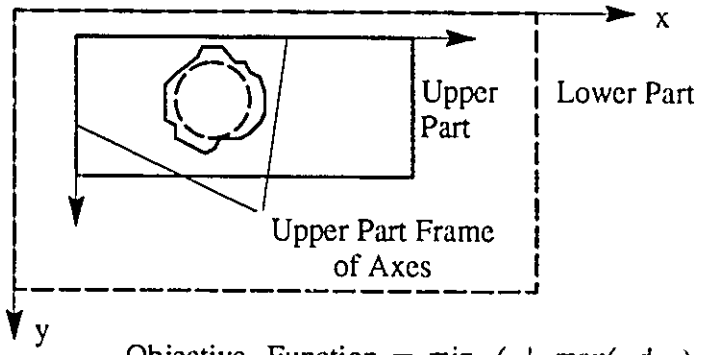


Figure 5 .17 Largest Inscribed Cylinder of the Lower Hole

2. If all points in the upper part's hole are separated from the cylinder's axis with a normal distance d (Figure 5 .17) less than the cylinder's radius, then the connection is acceptable.

3. Searching for a position for the upper part that can make the connection acceptable can be done using several search methods such as genetic algorithms or univariate search. The search is conducted by minimizing the minimum difference between d (Distance between the upper hole's points and the lower hole's largest inscribed cylinder axis) and the radius r (the radius of the lower hole's largest inscribed cylinder). The independent variables are the x,y locations of the upper part's frame of axes. The objective function is given in Figure 5 .18 . The optimization function places the upper part in a position having the minimum clearance between the upper hole points and the largest inscribed cylinder to the lower hole's points. If at the end of the search the value of (d_i–r) is negative for any upper hole point, then an interference is detected and the upper part cannot be placed in a position where the largest inscribed cylinder to the lower part's points can pass through the upper hole without interference.



$$\text{Objective Function} = \min_{x,y} \left(\left| \max_{i}(d_i) - r \right| \right) \qquad (5.23)$$

Figure 5 .18 Search for an Acceptable Position for the Upper Part

5 .4 .2 .4 Example

An example in Figure 5 .19 demonstrates the search for the optimum placement of an upper part on top of a lower one for a fixed fastener. The upper part has 4×5 points with radii equal to 2.2 inches, while the lower one has the same number of points but with a different radius value (2 inches). The optimal placement for the top part would be at x=0 and y=0 with a clearance equal to 0.2 inches. The search for the optimal placement was conducted using genetic algorithms. The result is shown in Figure 5 .20 .

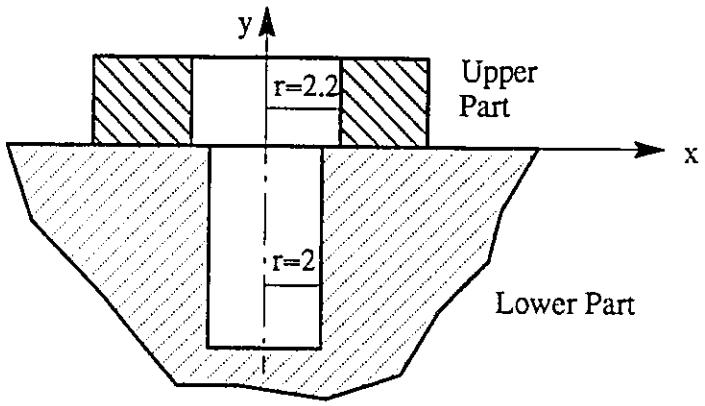


Figure 5 .19 Example for a fixed fastener

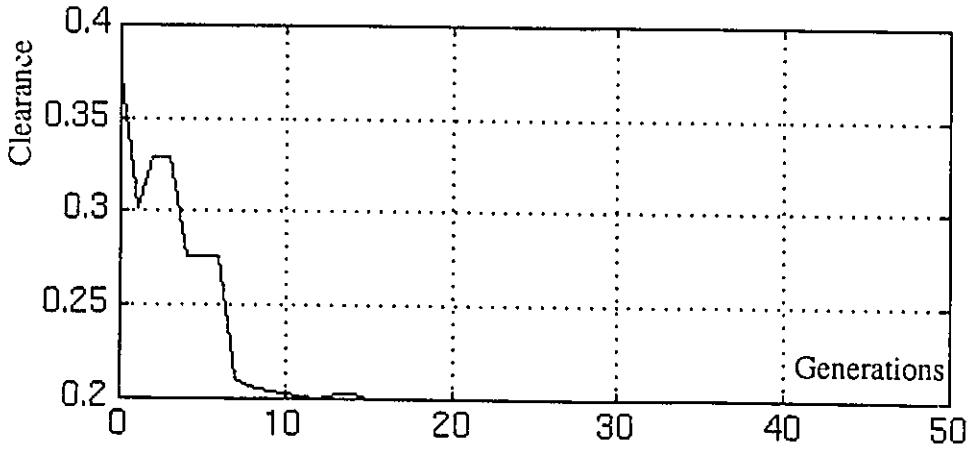


Figure 5 .20 Genetic Algorithms Result for Example in Figure 5 .19 .

5 .4 .3 Checking of Functional Requirements

The last step in simulating the assembly sequence is to determine whether the critical dimensions or clearances are within the acceptable ranges. These ranges are known in the tolerancing literature as functional requirements. The following sections discuss methods of evaluating functional requirements, once the assembly sequence has been completed.

5 .4 .3 .1 Clearances

A clearance is a gap between two surfaces of mating parts; examples range from gaps between car doors and the body frame to a gap between a hole and a shaft. The later application is considered in this section since it will be used in Chapter Seven. A clearance in an assembly should be checked for three criteria. The first criterion is that the minimum value for the gap should not be less than the specified minimum limit. The second criterion is that the maximum value for the gap should not exceed the specified upper limit. The third criterion is that the shaft does not interfere with the hole.

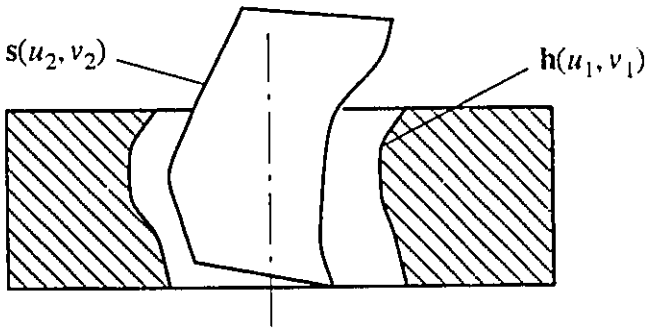


Figure 5 .21 Evaluating a Gap between a Hole and a Shaft.

To check the three criteria, two parametric surfaces are interpolated to the points of the shaft and the hole using the method described in Chapter 3 and as shown in Figure 5 .21 . The following objective function is minimized to find the minimum gap between the hole and the shaft:

$$\min_{u_1,v_1,u_2,v_2} \left(\left| \mathbf{s}(u_2, v_2) - \mathbf{h}(u_1, v_1) \right| \right) \tag{5.24}$$

The maximum value of the same function gives the value of the maximum gap between the hole and the shaft.

$$\max_{u_1,v_1} \left(\min_{u_2,v_2} \left(\left| \mathbf{s}(u_2, v_2) - \mathbf{h}(u_1, v_1) \right| \right) \right) \tag{5.25}$$

The given function is non–differentiable and can have several local maxima and minima. Thus, genetic algorithms can be used to find its global maximum and minimum values. To check for interference, the normal distances between the nominal axis of the hole and the surfaces **h** and **s** are calculated (d_h and d_s respectively). If ($d_h - d_s$) is negative at any time during the search for the maximum clearance, then an interference is detected and the objective function value can be set to a very large value to which the genetic search converges. This large value indicates that an interference is detected.

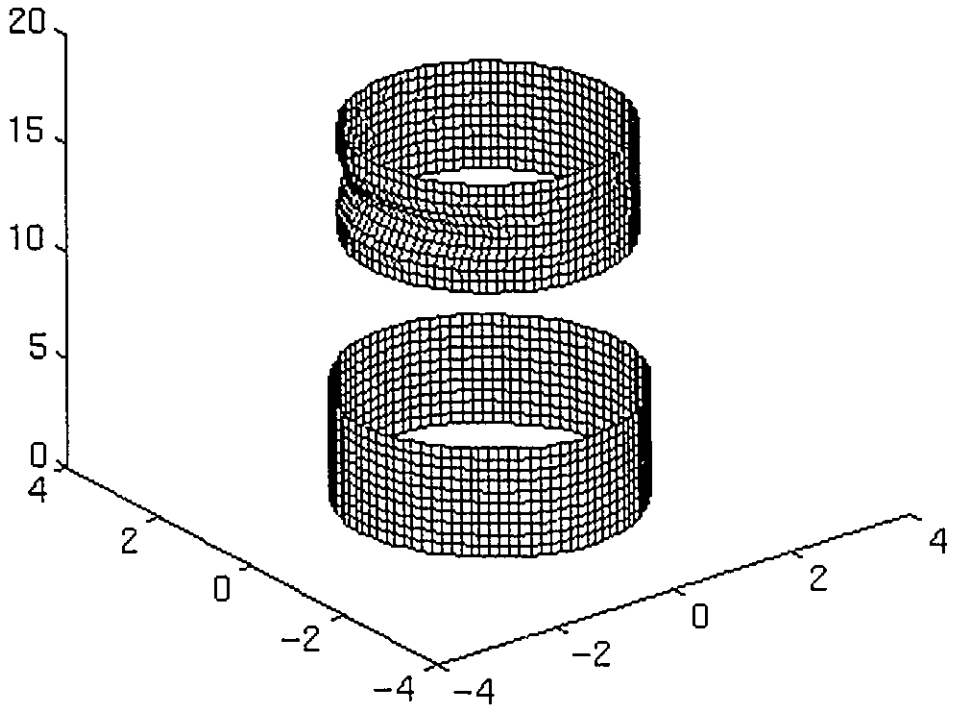


Figure 5 .22 An Example for a Clearance

Figure 5 .22 shows two concentric cylinders (the inner one is translated upwards for the purpose of viewing). The inner cylinder is the same cylinder with a depression in its surface, that was used in Figure 4 .9 in Chapter Four. The outer cylinder is a perfect form cylinder with a radius equal to 2.2 inches. The maximum clearance is equal to 0.4 inches and the minimum clearance is equal to 0.2 inches. The genetic algorithms results for the maximum and minimum clearances are shown in Figure 5 .23 and Figure 5 .24 respectively.

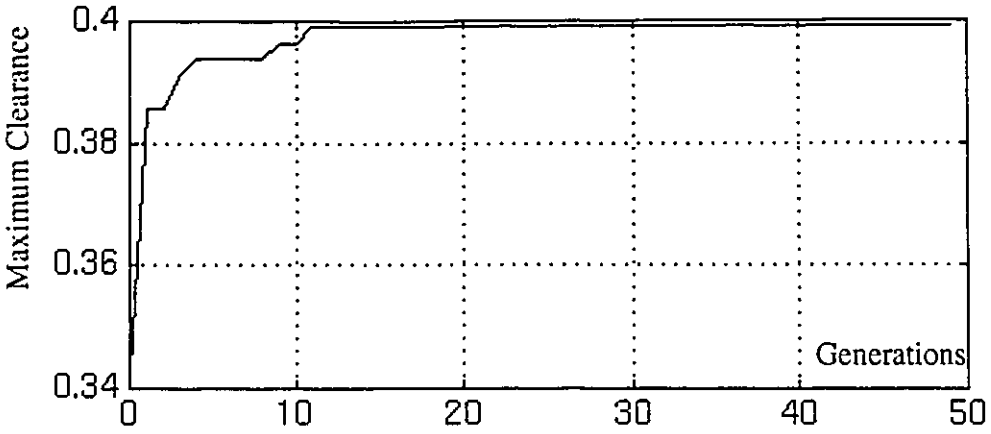


Figure 5 .23 Results for Minimum Clearance shown in Figure 5 .22 .

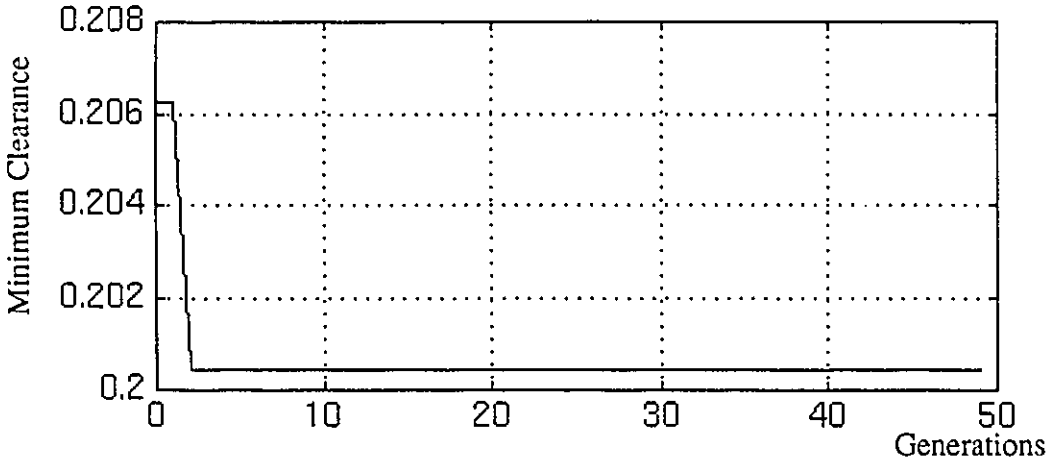


Figure 5 .24 Results for Maximum Clearance in Figure 5 .22 .

The coordinates of the inner cylinder are translated 0.2001 inches in the X–direction to touch the outer cylinder. The genetic algorithms results for the maximum clearance is shown in Figure 5 .25 . The search converged to a huge number (5×10^7) as soon as the interference was detected.

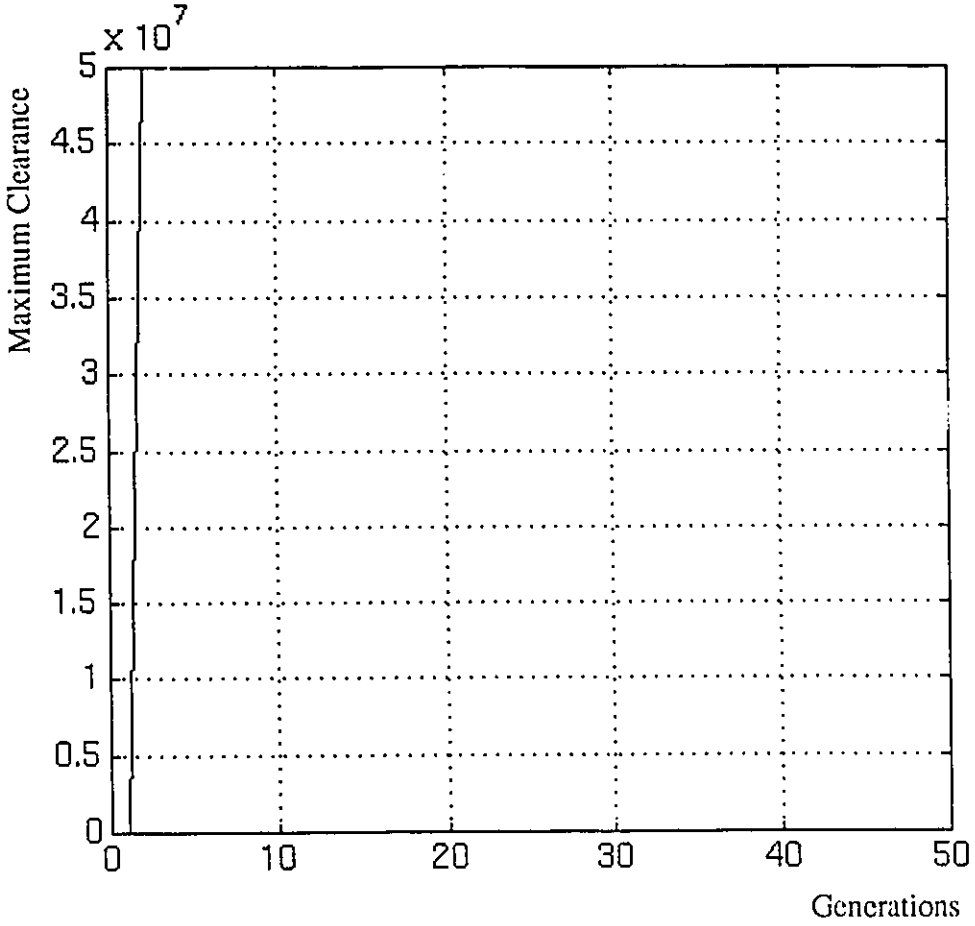


Figure 5 .25 Results for Interference in Figure 5 .22 .

5 .4 .3 .3 Dimensions

Critical dimensions are specified between two points in the assembly. Given two points $\mathbf{p}_1$ and $\mathbf{p}_2$, the distance between both points is given by the equation:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \qquad (5.26)$$

5 .5 Conclusions

This chapter demonstrated a simulation methodology for building assemblies of manufactured parts. This step is needed for checking whether or not a set of manufactured

parts would conform to the specified functional requirements when assembled together. The reason for simulating the assembly is that it is difficult, if not impossible, to develop a closed form tolerance stack–up function for geometric tolerances; hence the surfaces are generated from the manufacturing processes distributions, the assembly sequence is simulated and the functional requirements are checked. An algorithm was developed for simulating the assembly sequence based on an assembly graph and a study was carried out for some types of assembly connections. A new algorithm was developed to check limitations on clearances using genetic algorithms. The use of genetic algorithms ensured the detection of the globally maximum and minimum values of the clearance between two manufactured surfaces and a provision was made for detecting interference. The use of a global optimization method ensures that actual values of the clearance are evaluated. The presented algorithms can be used within a Monte Carlo simulation routine to evaluate the probabilities of rejection and acceptance. The investigation of assembly connections was limited to a certain number of assembly modes that were necessary for the examples in Chapters Six and Seven. A thorough investigation would need an enumeration of all assembly modes, a task that can be carried out in future research.

CHAPTER SIX

TOLERANCE ANALYSIS OF ASSEMBLIES

Tolerance analysis is the calculation of the probabilities of acceptance and rejection of an assembly due to the violation of the specified tolerances or design functional requirements. Monte Carlo simulation is used in this chapter to evaluate these probabilities. Since the problem of geometric tolerance analysis is a complex one dealing with a large number of random variables, simple Monte Carlo simulation might fail to detect instances of assemblies where a violation of function requirements or the specified tolerances is encountered. This problem is compounded by a large expected variance of the calculated probabilities leading, to larger errors in the values of the calculated probabilities. These problems are solved by incorporating two variance reduction techniques with the Monte Carlo simulation.

6 .1 Probabilistic Analysis of Tolerances

Given is a set of dimensional tolerances:

$$\mathbf{t} = \{\ t_1,\ \ t_2,\ \ t_3,\ \ \ldots,\ \ t_n\ \} \tag{6.1}$$

specified on n dimensions in an assembly, and a set of m functional requirements:

$$F_j\ (t_1, t_2, t_3, \ldots t_n) \leq v_j \quad j \in \{1, 2, \ldots, m\} \tag{6.2}$$

where v_j is an arbitrary value specified by the designer that F_j should not exceed. Knowing that the tolerances are independent random variables, each having a probability distribution $P_i(t_i)$, the joint probability distribution of these tolerances becomes:

$$P(t_1, t_2, t_3, \ldots t_n) = \prod_{i=1}^{n} P_i(t_i) \qquad (6.3)$$

Tolerance analysis is the calculation of the probability of rejection:

$$P_{rej} = \int\int \cdots \int_{F(t)>\nu} P(t_1, t_2, t_3, \ldots t_n)\ dt_1 dt_2 \ldots dt_n \qquad (6.4)$$

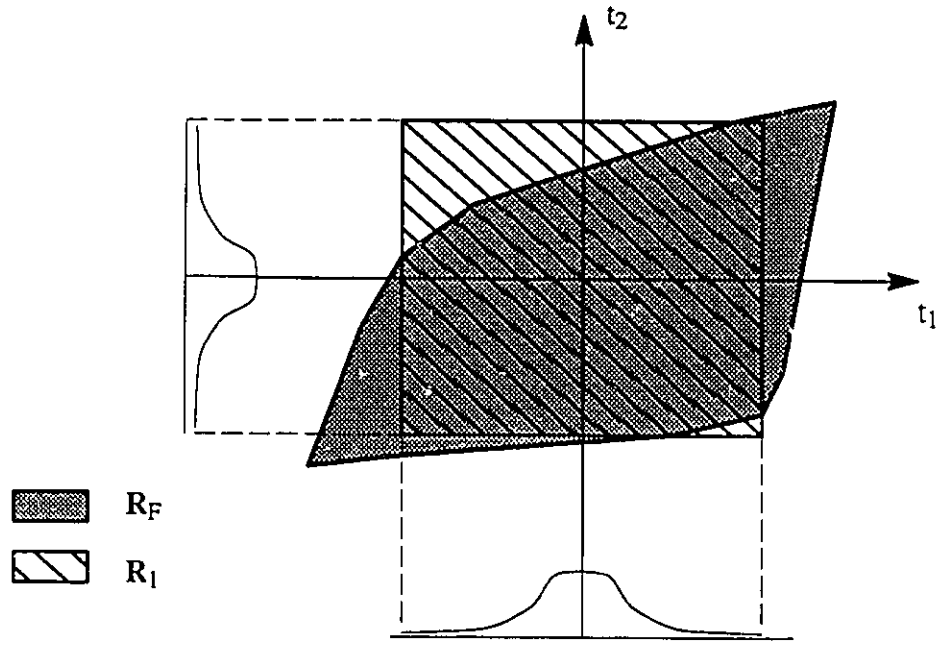


Figure 6.1 Probabilistic Analysis of Dimensional Tolerances

Figure 6.1 shows a diagram of the regions formed by the tolerances and the functional requirements over two tolerance values. The region $\mathbf{R_F}$ (enclosed by the bold line) defines the set of dimensions acceptable to the functional requirements. The region $\mathbf{R_1}$ refers to the set of dimensions generated within the limits of the manufacturing processes

probability distributions. The probability of rejection referred to in equation (6 .4) is equal to 1 minus the probability of accepting the manufactured dimensions. The acceptance probability is equal to the volume enclosed by the joint probability distribution and the intersection of regions $\mathbf{R}_F$ and $\mathbf{R}_1$.

6 .1 .1 Geometric Tolerances

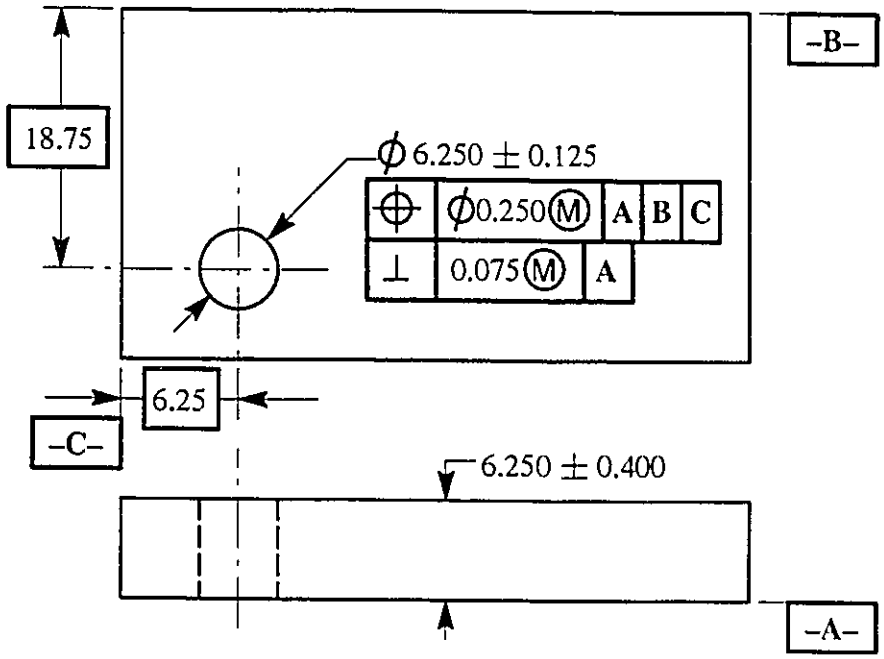


Figure 6 .2 Probabilistic Analysis of Dimensional Tolerances

The probabilistic analysis of geometric tolerances is not as straightforward as that of dimensional tolerances. Consider the part shown in Figure 6 .2 . The imposed tolerances on the cylindrical feature (size, position and perpendicularity) control the surface variation of the feature. At the same time, assume that the part is assembled to another part where an assembly function requirement is imposed (e.g. limits on a clearance). A manufactured feature is acceptable if its deviations fall within the limits of the specified geometric tolerances and if, when assembled, the functional requirements are satisfied; otherwise, it

is rejected. On a feasibility diagram (such as the one shown in Figure 6 .1), the geometric tolerances form a region of feasible values for the coordinates of the manufactured surface points. Figure 6 .4 shows an illustration of such a feasibility region.

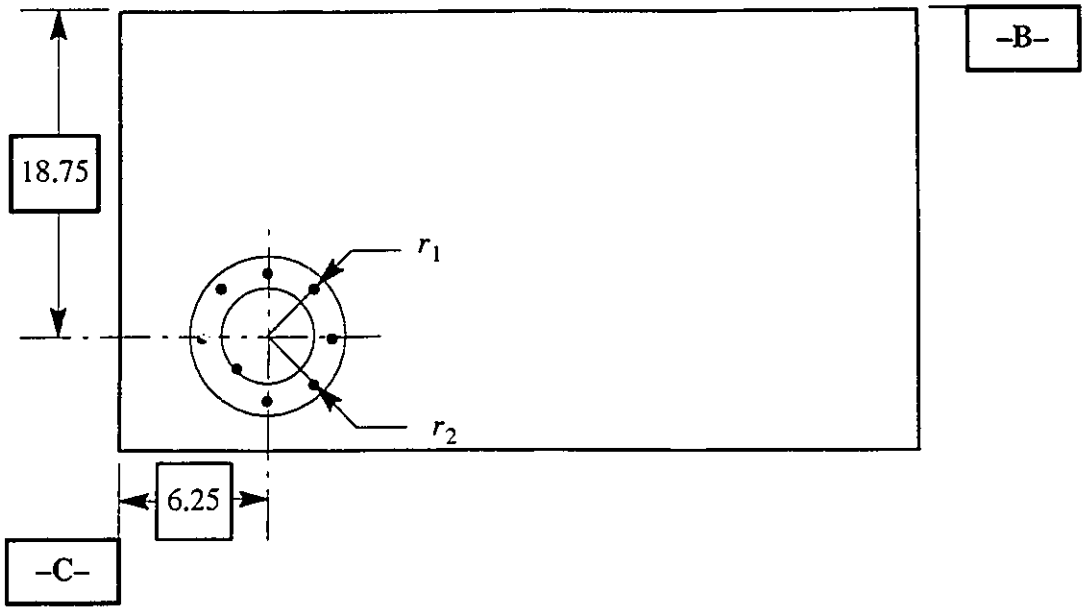


Figure 6 .3 Manufactured Cylindrical Feature

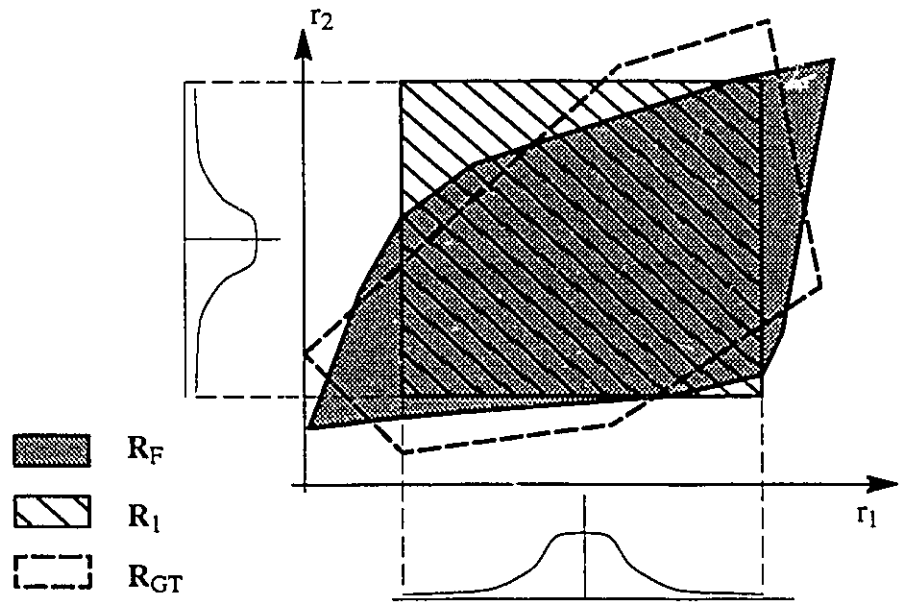


Figure 6 .4 Feasible Regions for a Geometric Feature

Assume that the feature's manufactured surface is represented by a number of points generated from the machining process probability distribution. Assume that the feature shown in Figure 6 .2 is represented by 8×3 points positioned at equal angles and measured from the datum reference frame **A**, **B** and **C** as shown in Figure 6 .3 . The radii of the points representing the manufactured surface are generated from the probability distribution representing random variation of the machined surface. Figure 6 .4 shows the region $\mathbf{R}_{GT}$ enclosed by the dotted bold lines representing the set of feasible radii values satisfying the imposed tolerances. The shaded region $\mathbf{R}_F$ represents the set of feasible radii values satisfying the assembly functional requirements, and the region $\mathbf{R}_I$ represents the set of radii values that can be generated from the manufacturing process distribution. The probability of accepting a feature is thus equal to the volume enclosed by the joint probability distribution over the intersection of $\mathbf{R}_F$ and $\mathbf{R}_{GT}$.

$$P_{acc} = \int_{\{r_1, r_2, \ldots, r_n\} \in (\mathbf{R}_F \cap \mathbf{R}_{GT})} \int \int \cdots \int P(r_1, r_2, r_3, \ldots r_n)\ dr_1 dr_2 \ldots dr_n \qquad (6.5)$$

The stack–up function developed in chapter five can be used to determine whether an instance of assembly parts will be acceptable to the imposed geometric tolerances and to the assembly functional requirements or not, i.e. whether its surface points belong to the regions $\mathbf{R}_F$ and $\mathbf{R}_{GT}$ or not.

6 .1 .2 Choice of Analysis Method

Methods of probabilistic tolerance analysis used in previous work included Taylor series expansion (Evans, 1975), quadrature method (Evans, 1975), cell decomposition (Michael and Siddall, 1981), reliability index method (Lee, W., 1988) and Monte Carlo simulation (Chase and Greenwood, 1988). Monte Carlo simulation needs a large number of samples to evaluate the integral in equation (6 .5). On the other hand, all other methods,

except for the cell decomposition method, need a closed form mathematical representation of the design functions, which is an infeasible requirement for geometric deviations of manufactured surfaces since the majority of these are evaluated with optimization routines. The cell decomposition method is also ruled out because the method depends on checking the cells' corners which increase exponentially with the increase in number of radii and features. This leaves Monte Carlo simulation, but it requires a large number of samples for accurate results. The number of samples needed for simulation can be reduced using variance reduction techniques as shown in the following sections.

6 .2 Monte Carlo Simulation

Given an assembly with the following characteristics:

1. A set of parts, where each part consists of a number of features (total number of features in the assembly = N_f). Each feature is produced by a manufacturing process with a known probability distribution representing the variability in the surface production. Each feature is represented by a set of points (total number of points for each feature equals N_p). Thus the number of random variables equals $N = N_f \times N_p$.
2. A manufacturing sequence for each part.
3. A set of connections between the parts' features.
4. An assembly sequence.

The sample mean Monte Carlo simulation method (section **C.1**) can be used to evaluate the integral of equation (6 .5) for the assembly using the following procedure:

Procedure: TOL–AN1

1. Using the manufacturing process probability distribution associated with each feature and the manufacturing sequence simulation procedure **MCSEQ** (section

5 .2 .2), generate N_s samples of the random vector $\{R_1, R_2, R_3, \ldots, R_{Ns}\}$, where

$$R_i = \left[r_1, r_2, r_3, \ldots r_N\right]_i \tag{6 .6}$$

2. **for** all N_s random vectors **do**

   a. Calculate the geometric deviations on all features using the algorithms described in chapter **4** and check them for the imposed tolerances.

   b. Simulate the assembly sequence using the assembly sequence procedure **ASMSEQ** (section **5 .4 .1 .2**), and check the assembly functional requirements.

   c. **if** the assembly requirements and the geometric tolerances are in spec **then**

      Increment Na (Counter for accepted instances of the assembly).

   else

      Increment Nf (Counter for rejected instances of the assembly).

   end–if

end–for

3. Calculate the acceptance and rejection probabilities

$$P_{acc} = \frac{Na}{Na + Nf} \tag{6 .7}$$

$$P_{rej} = \frac{Nf}{Na + Nf} \tag{6 .8}$$

To calculate the probabilities of rejection and acceptance accurately, the number of random vector samples "N_s" should be large. A third decimal place accuracy necessitates

that N_s be around 100,000 (Thisted, 1988). This large number can be reduced using variance reduction techniques.

6 .3 Variance Reduction Techniques

Assume that the probability of rejection P_{rej} is evaluated using the sample mean Monte Carlo simulation described above. If the variance of the calculated probability is lessened, then the value of P_{rej} can be more accurately calculated. On the other hand, if two Monte Carlo methods are required to achieve the same accuracy of calculation, then the one with less variance will need less time to calculate the probability of rejection. Variance reduction techniques are incorporated with Monte Carlo simulation to reduce the variance of the evaluated probabilities.

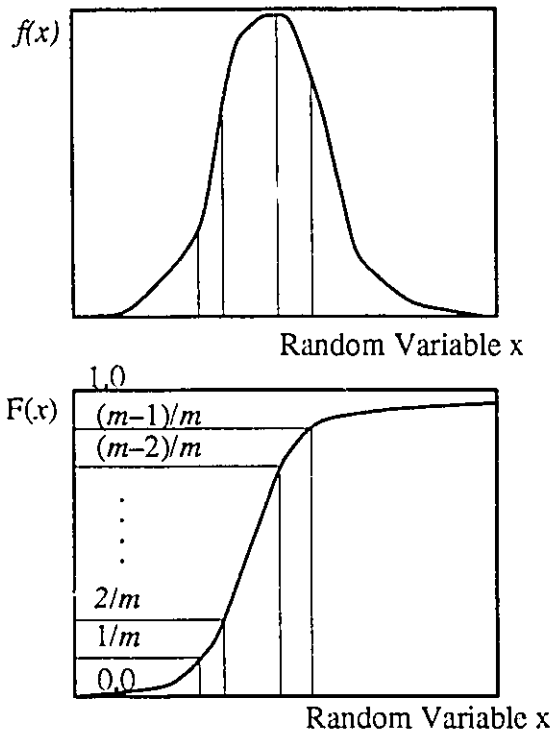


Figure 6 .5 Latin Hypercube Sampling

6 .3 .1 Latin Hypercube Sampling

Latin hypercube sampling is a constrained sampling method which covers the range of the sampled random variable (Rubinstein, 1981) by breaking it into intervals, then distributing the sampling process over these intervals.

6 .3 .1 .1 Sampling For One Variable

Consider a random variable x whose probability distribution function $f(x)$ and probability function $F(x)$ are shown in Figure 6 .5 . The probability function is divided equally into m non–overlapping intervals. All intervals have the same probability of occurrence, and each interval can be used to generate m values of the random variable for each Monte Carlo simulation cycle. The m values can be generated by generating m random numbers from a uniform distribution over the range [0,1] and allocating a random number U_i to the ith interval, thus for each random number $U \in [0,1]$ generated for the ith interval, there is a value U_i such that:

$$U_i = \frac{U}{m} + \frac{(i-1)}{m} \qquad i \in \{1, 2, \ldots, m\} \tag{6 .9}$$

Equation (6 .9) shows that U_i ranges from $(i-1)/m$ to i/m. The random numbers U_i represent a set of m probability values for the random variable x. The equivalent values of the random variable can be produced using the inverse transform method:

$$x_i = F^{-1}(U_i) \tag{6 .10}$$

where F^{-1} is the inverse probability function.

6 .3 .1 .2 Sampling for More than One Variable

The Latin hypercube sampling method delivers a set of m values in each Monte Carlo simulation cycle. For more than one random variable, the method divides the probability

space into m^n probability subregions, where n is the number of of random variables. Figure 6 .6 shows the division of the probability space for two random variables into 16 intervals. The sampling method delivers in each cycle x_{1j} and x_{2j} , $j \in \{1,2,....,m\}$, which is cquivalent to generating random variates in the regions $R_{1,1}$, $R_{2,2}$, ... , $R_{m,m}$. The generated random variates are randomly assigned to different regions, to eliminate the generation bias to the diagonal regions. This is done by rearranging the variables x_1 and x_2 such that:

$$\left\{x_{ij} \ : \ i \in \{1,2\} \ , \ j \in P\right\} \tag{6 .11}$$

where P is a random permutation of the set $\{1,2,.....,m\}$.

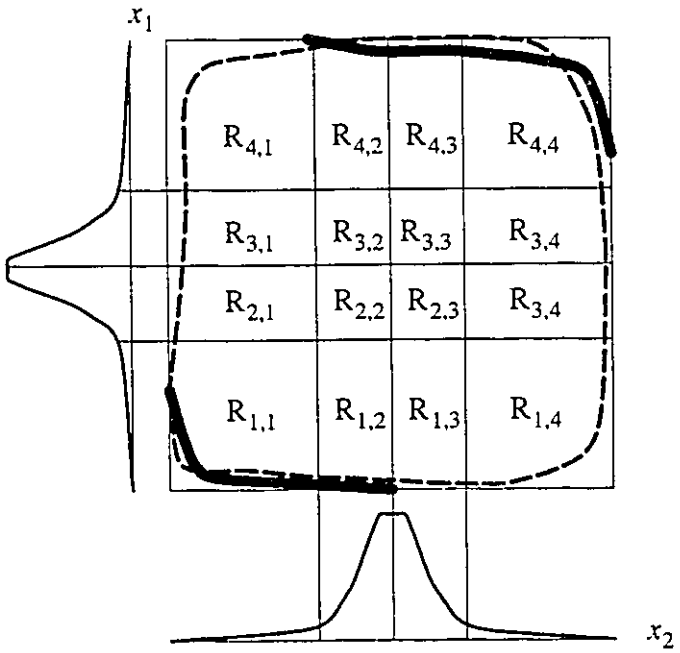


Figure 6 .6 Latin Hypercube Sampling for Two Variables

6 .3 .1 .3 Advantages of using Latin Hypercube Sampling

Latin hypercube sampling is one of the stratified sampling methods (Appendix C, section C.3). There are two main advantages to using them over the standard sample mean Monte Carlo simulation random sampling. The first advantage is that the variance of the calculated probability is less than that of the probability calculated using standard Monte Carlo (Rubinstein, 1981). In tolerance analysis problems the calculated probability of rejection is very small, and the limit functions tend to lie near the tails of the probability distributions. If the number of simulated variables is large, the standard Monte Carlo method tends to generate samples in the central regions of the probability space thus missing the rejection regions and giving a zero probability of failure (Figure 6 .7).

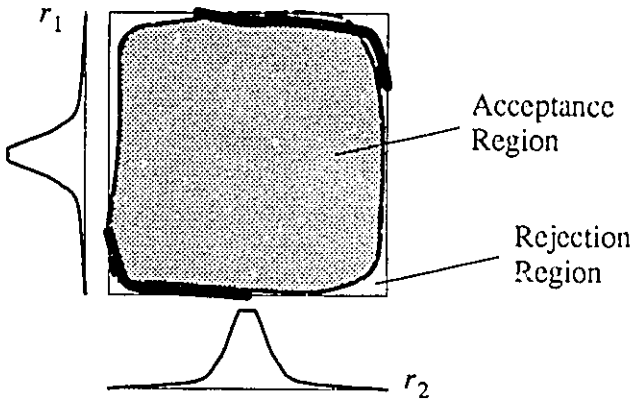


Figure 6 .7 Acceptance Region for Monte Carlo Simulation

The second advantage of Latin hypercube sampling is that it constrains the sampling within regions such that for each sample a value is generated at the tail of the distribution.

6 .3 .2 Antithetic Variates

Figure 6 .8 shows the probability distribution function and the probability function of a random variate x. Let a random number U be used to generate an instance of X of the random variate x using the inverse transform method and let another number 1–U be used to generate X'. If the process is repeated for N times the covariance between X and X' will be:

$$\mathrm{COV}(X, X') = \frac{\sum_{i=1}^{N}(X_i - \overline{X})(X'_i - \overline{X})}{N - 1} \qquad (6.12)$$

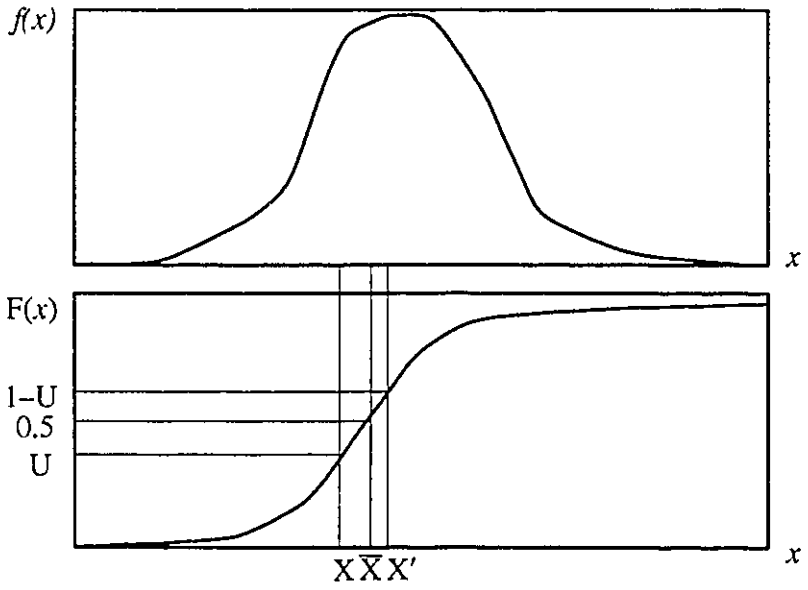


Figure 6 .8 Antithetic Variates

Since $(X_i - \overline{X})$ and $(X'_i - \overline{X})$ can never have the same sign, the covariance is strongly negative. If a probability of rejection is calculated then let U be a random number drawn from a uniform probability distribution, let *Pr'* be the probability of rejection calculated from N random generations of U, and let *Pr"* be the probability of rejection calculated from N

generations of (1–U). The probability of rejection has a mean equal to:

$$\overline{Pr} = \frac{Pr' + Pr''}{2} \tag{6.13}$$

and a variance equal to:

$$\mathrm{VAR}(Pr) = \frac{\mathrm{VAR}(Pr') + \mathrm{VAR}(Pr'') + \mathrm{COV}(Pr', Pr'')}{4} \tag{6.14}$$

Since COV(Pr',Pr'') is strongly negative, then the variance of Pr is reduced (Rubinstein, 1981).

6 .3 .3 Algorithm for Tolerance Analysis

The inclusion of Latin hypercube sampling and antithetic variates within the tolerance analysis algorithm is shown in the following procedure:

Procedure: TOL–AN2

1. Divide the manufacturing process probability function associated with each feature into m equal probability intervals. Let N_s be the number of samples required for simulation and let N be the number of random variables (radii on features) to be simulated.

2. **repeat** N_s times,

   a. Using the manufacturing process probability distribution associated with each feature and the manufacturing sequence simulation procedure **MCSEQ** (section **5 .2 .2**), generate m values of all random variables using Latin hypercube sampling. Thus an $N \times m$ matrix r of the generated values is

obtained:

$$r = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_i \\ \vdots \\ r_N \end{bmatrix} = \begin{bmatrix} r_{1,1} & r_{1,2} & r_{1,3} & \cdots & r_{1,m} \\ r_{2,1} & r_{2,2} & r_{2,3} & \cdots & r_{2,m} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ r_{i,1} & r_{i,2} & r_{i,3} & \cdots & r_{i,m} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ r_{N,1} & r_{N,2} & r_{N,3} & \cdots & r_{N,m} \end{bmatrix} \qquad (6.15)$$

b. For each random variable r_i rearrange the m generated values in a random permutation, i.e. rearrange the rows of matrix r in a random permutation.

c. Find the antithetic variate of each element of the r matrix, thus forming another matrix of $\bar{r}$ of antithetic variates.

d. Since each column of the r and $\bar{r}$ matrices form an instance of a set of manufactured parts, then **for** each column in the r matrix **do**

   i. Calculate the geometric deviations on all features using the algorithms described in chapter **4**, then check them for the imposed tolerances.

   ii. Simulate the assembly sequence using the assembly sequence procedure **ASMSEQ** (section **5.4.1.2**), and check the assembly functional requirements.

   iii. **if** the assembly requirements and the tolerances are in spec **then**

      Increment Na = (Counter for accepted instances of the assembly)

   else

      Increment Nf (Counter for rejected instances of the assembly).

   end–if

e. **end–for**

f. Repeat the loop of step "d" for the $\bar{r}$ obtaining $\overline{Na}$ and $\overline{Nf}$.

3. **end–repeat**

4. Calculate the acceptance and rejection probabilities

$$P_{acc} = 0.5\left(\frac{Na}{Na + Nf} + \frac{\overline{Na}}{\overline{Na} + \overline{Nf}}\right) \tag{6.16}$$

$$P_{rej} = 0.5\left(\frac{Nf}{Na + Nf} + \frac{\overline{Nf}}{\overline{Na} + \overline{Nf}}\right) \tag{6.17}$$

Appendix **C** provides a review of the relations by which the discussed variance reduction techniques decrease the variance of the evaluated probabilities. A combination of Latin hypercube sampling with m=10 and antithetic variates decreases the variance of the calculated probability by a factor of 1/200, and thus number of samples needed to achieve a third decimal place accuracy is equal 500.

6.4 Example 1 -- A Toleranced Hole

The example shown in Figure 6.2 is used to test the proposed probabilistic analysis method. The cylindrical feature is to be analyzed for the imposed geometric tolerances only, since there are no assembly functional requirements shown. Points on the feature are generated from a multinormal probability distribution having a mean radius of 3.125 mm measured from a centreline lying 18.75 mm from the datum **B** and 6.25 mm from datum **C**. Eight points are used on the feature's circumference at three levels along the feature's length (Figure 6.9), i.e. the number of random variables N is equal to $8 \times 3 = 24$. The number of Latin hypercube sampling intervals m is equal to 10. The radii of the points are assumed to have a normal probability distribution with a mean value equal to the nominal radius (3.125 mm), and thus the covariance matrix has zero non–diagonal elements, while the diagonal variances are equal to the square of the largest imposed tolerance (0.25) divided by 6. Procedure **TOL–AN2** is used with a number of samples N_s = 500 to evaluate the rejection

probability. Results are shown in the following section.

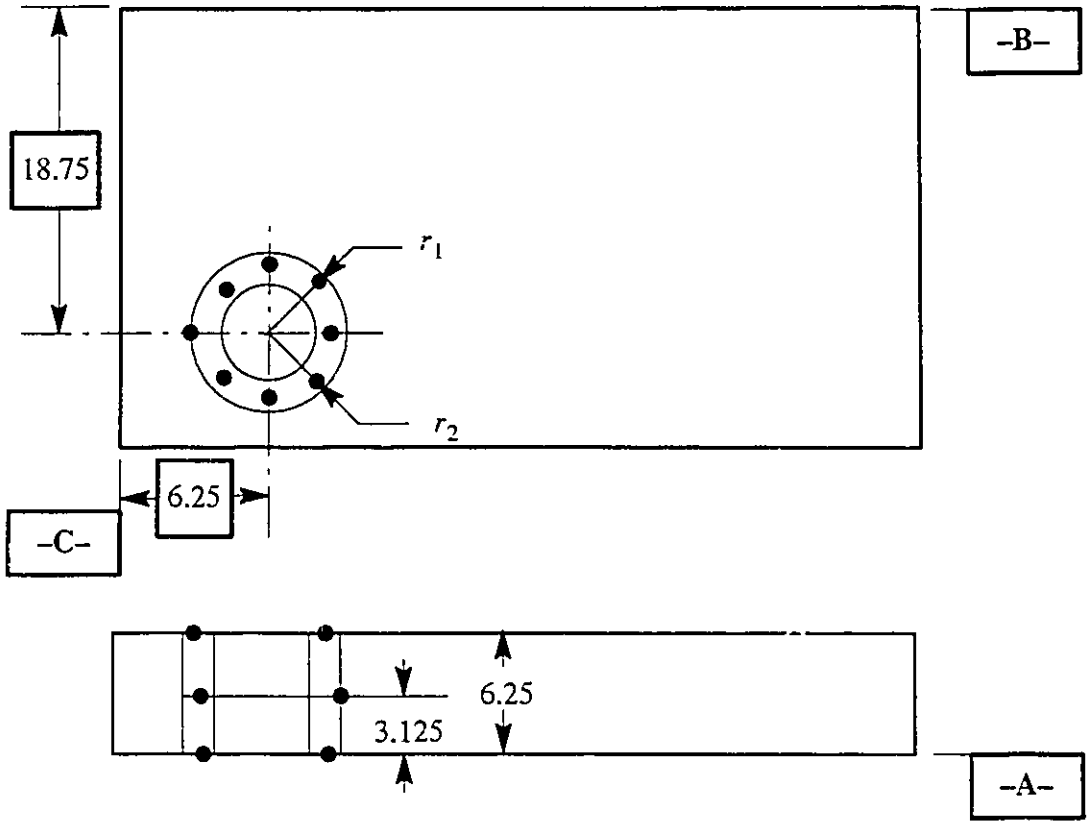
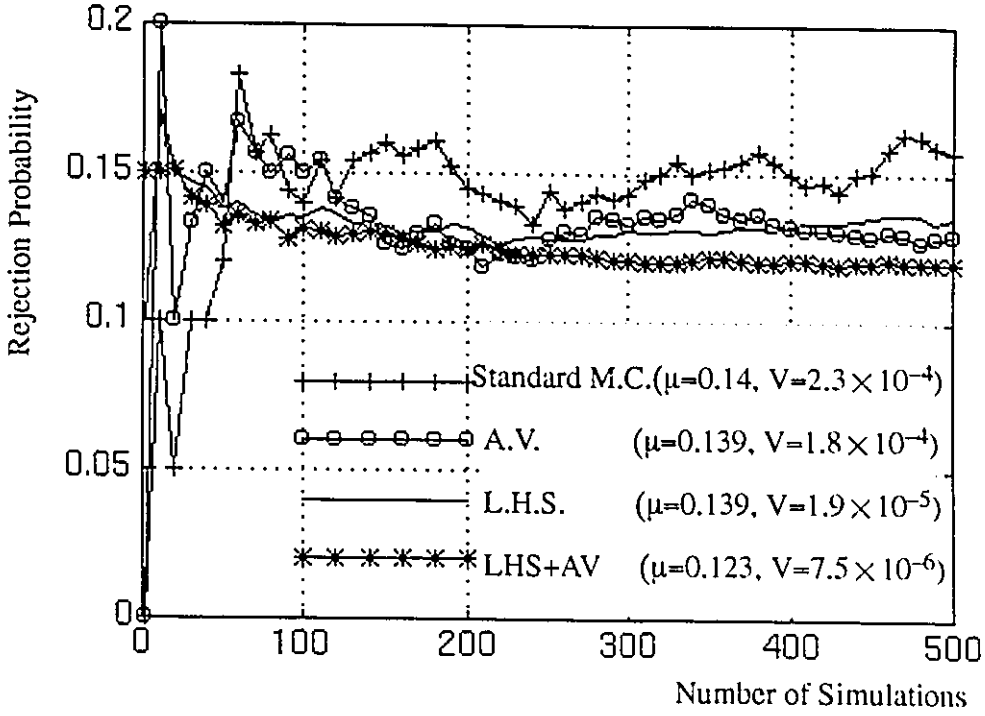


Figure 6 .9 Generation of Example's Points

6 .4 .1 Observations

Figure 6 .10 shows the simulation results from combining Monte Carlo simulation and variance reduction techniques for up to 500 generated samples. In addition, the figure shows the mean value and the variance of the probability of rejection for 100 runs of the different simulations. Results show that a combination of antithetic variates and Latin hypercube sampling converged after 200 samples with a low variance.



Legend: μ=mean, V=variance, M.C.=Monte Carlo simulation.
A.V.=Antithetic Variates, LHS=Latin hypercube sampling

Figure 6 .10 Tolerance Analysis Results. Example-1

6 .5 Example 2 – A Speed Reducer

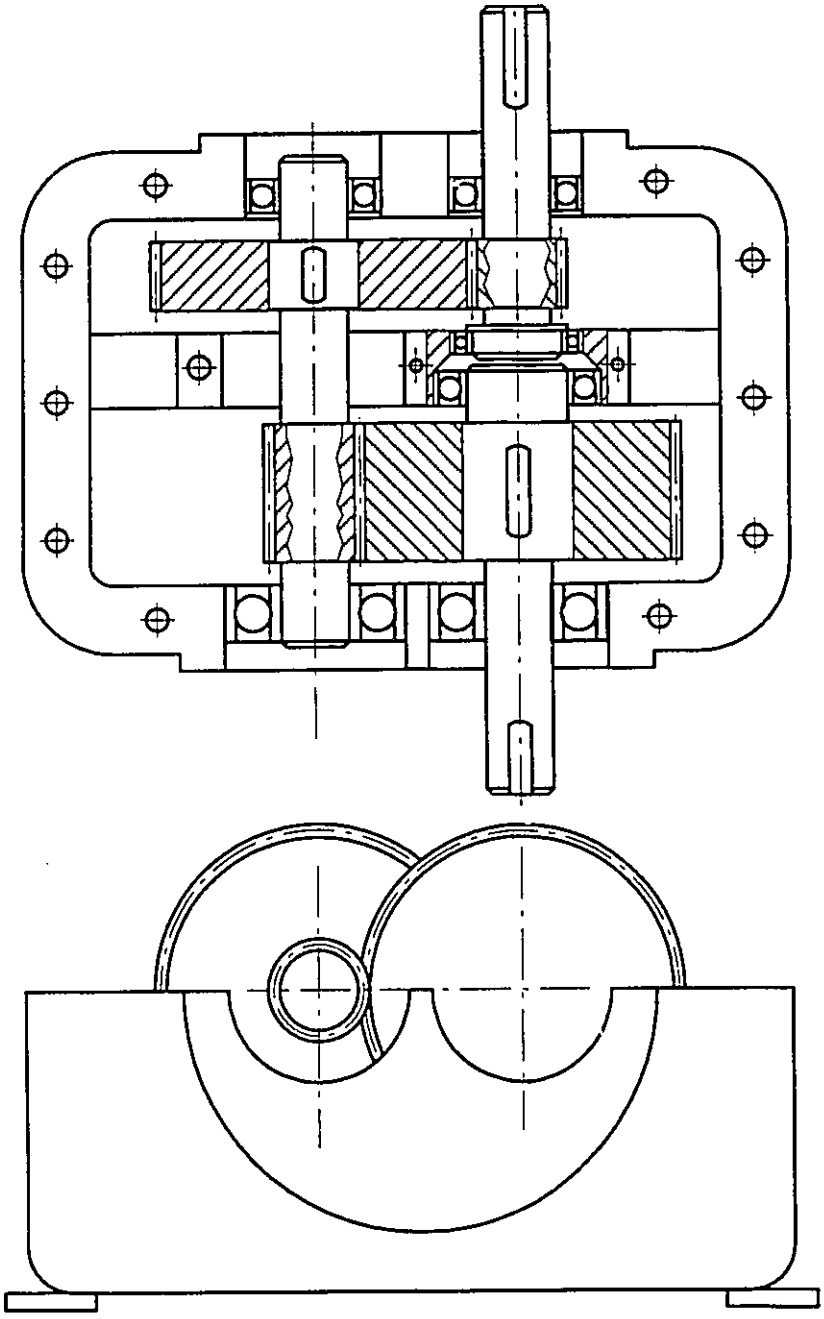


Figure 6 .11 Example 2: A Speed Reducer

The speed reducer assembly shown in Figure 6 .11 (obtained from Niemann, 1975) is used to test the proposed tolerance analysis procedure in the following sections.

6 .5 .1 Dimensional Tolerances

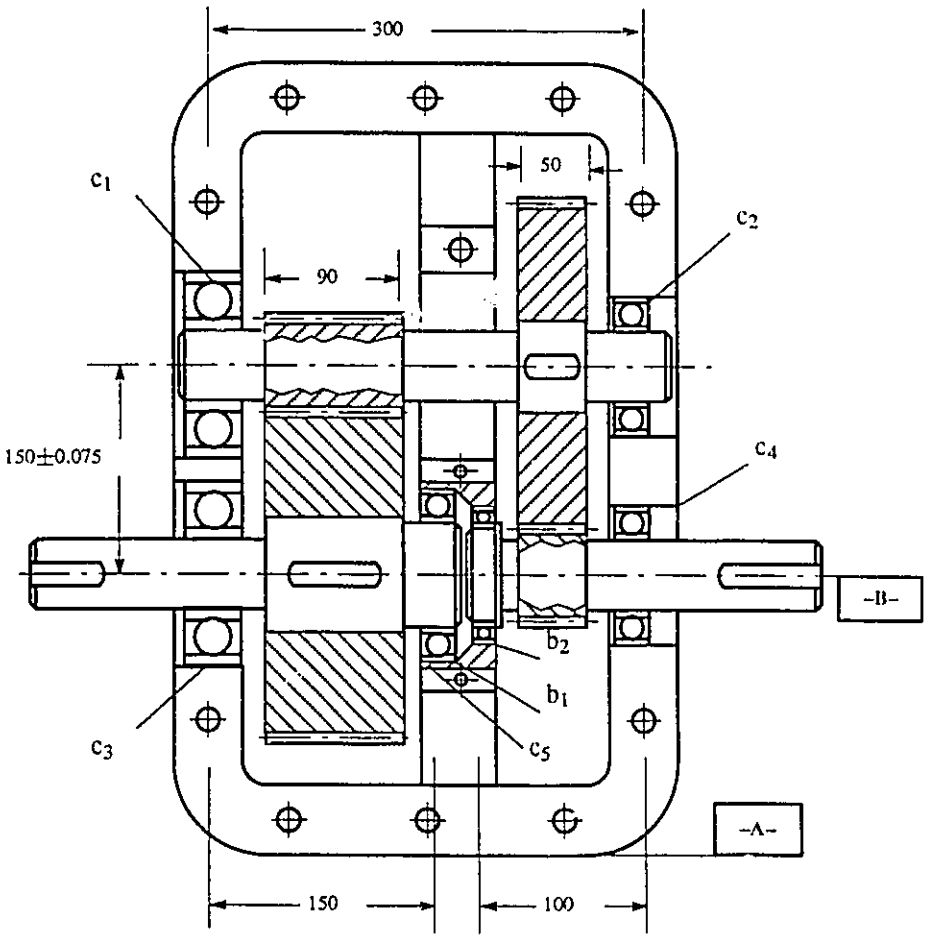


Figure 6 .12 Example 2: (Dimensional Tolerances)

Figure 6 .12 shows the speed reducer with some dimensional tolerances imposed on it. The centre distance between the gears' axes should not exceed certain design values equal to 0.15 mm, i.e. ±0.075mm (Dudley, 1991); otherwise, backlash between the gear teeth will increase, leading in turn to an increase in the gear noise. The tolerances under consideration

are the position tolerances of the holes in the casing (c_1,c_2,c_3.c_4,c_5) measured from datum **A** and the position tolerance of the holes in the bushing (b_1,b_2) measured from the datum **B**. The centre distance is to be checked at the start and end of the face width of every pair of gears; therefore, there are four functional requirements:

$$-0.075 \leq CD_j \leq 0.075, \quad j \in \{1,2,3,4\} \tag{6.18}$$

where CD_j is the calculated deviation in the jth centre distance,

$$CD_1 = \frac{30}{300}(c_1 - c_2) + c_2 + \frac{30}{150}(c_3 - b_1 - c_5) + b_1 + c_5 \tag{6.19}$$

$$CD_2 = \frac{120}{300}(c_1 - c_2) + c_2 + \frac{120}{150}(c_3 - b_1 - c_5) + b_1 + c_5 \tag{6.20}$$

$$CD_3 = \frac{75}{300}(c_2 - c_1) + c_1 + \frac{75}{100}(c_4 - b_2 - c_5) + b_2 + c_5 \tag{6.21}$$

$$CD_4 = \frac{25}{300}(c_2 - c_1) + c_1 + \frac{25}{100}(c_4 - b_2 - c_5) + b_2 + c_5 \tag{6.22}$$

Table 6.1 shows the values of the position tolerance of each feature considered in the analysis:

Feature	c_1	c_2	c_3	c_4	c_5	b_1	b_2
Tol.	0.149	0.149	0.150	0.149	0.150	0.150	0.150

Table 6.1 Position Tolerance Values

Assuming that each tolerance has a normal probability distribution with a zero mean and a standard deviation equal to 1/6 the tolerance value, the exact solution of the problem is found using the root sum square method and is equal to 0.0659. (A linear problem was chosen to compare the simulation results to the exact solution). The problem is simulated using procedure **TOL–AN2** and the results for $N_s = 500$ are shown in the following section.

6 .5 .1 .1 Observations

Figure 6 .13 shows the simulation results of combinations of Monte Carlo simulation and variance reduction techniques for up to 500 generations, while Table 6 .2 shows the mean value and the variance of the probability of rejection for 100 runs of the different simulations. Results show that a combination of antithetic variates and Latin hypercube sampling converged to the correct value with a low variance after a small number of generations.

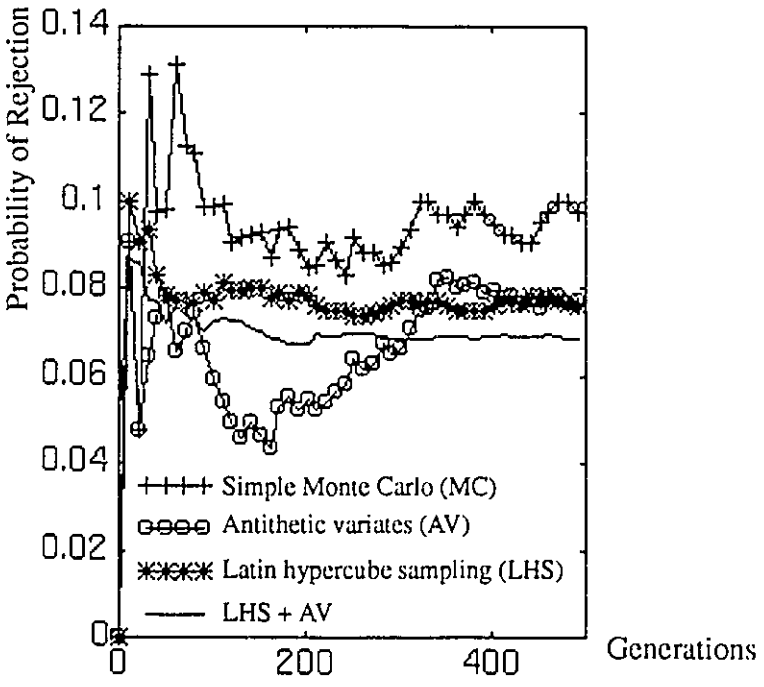


Figure 6 .13 Example 2: Simulation Results

Method	(MC)	AV	LHS	AV + LHS
Variance	1.36×10^{-4}	1.16×10^{-4}	1.18×10^{-5}	5×10^{-6}
Mean	0.0754	0.0788	0.0801	0.0696

Table 6 .2 Means and Variances of Calculated Probabilities

6 .5 .2 Geometric Tolerances

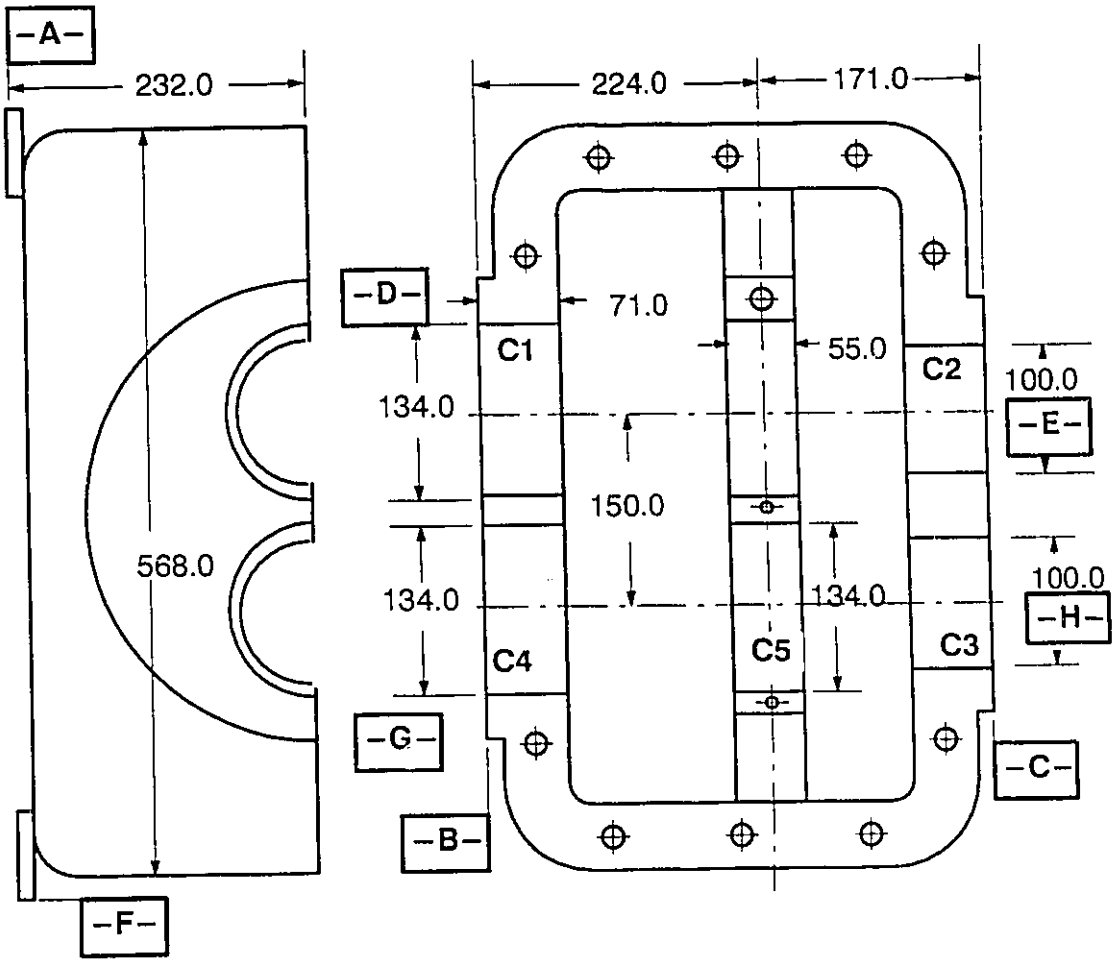
The speed reducer shown in Figure 6 .11 is used to test procedure **TOL–AN2** with a set of geometric tolerances imposed on its features. The reducer's parts are shown in Figure 6 .14 , Figure 6 .15 and Figure 6 .16 along with their dimensions and tolerances. Tolerances on the holes in the casing and the bushing are the only ones considered. The shafts and gears are assumed to be at their nominal geometry.

6 .5 .2 .1 Manufacturing Sequences

The manufacturing sequence for features in the casing and the bushing are determined from the position tolerancing schemes imposed on them using procedure **MCSEQ** (section **5 .2 .2**). The manufacturing sequence of the casing is shown in Table 6 .3 , and the manufacturing sequence of the bushing is shown in Table 6 .4 .

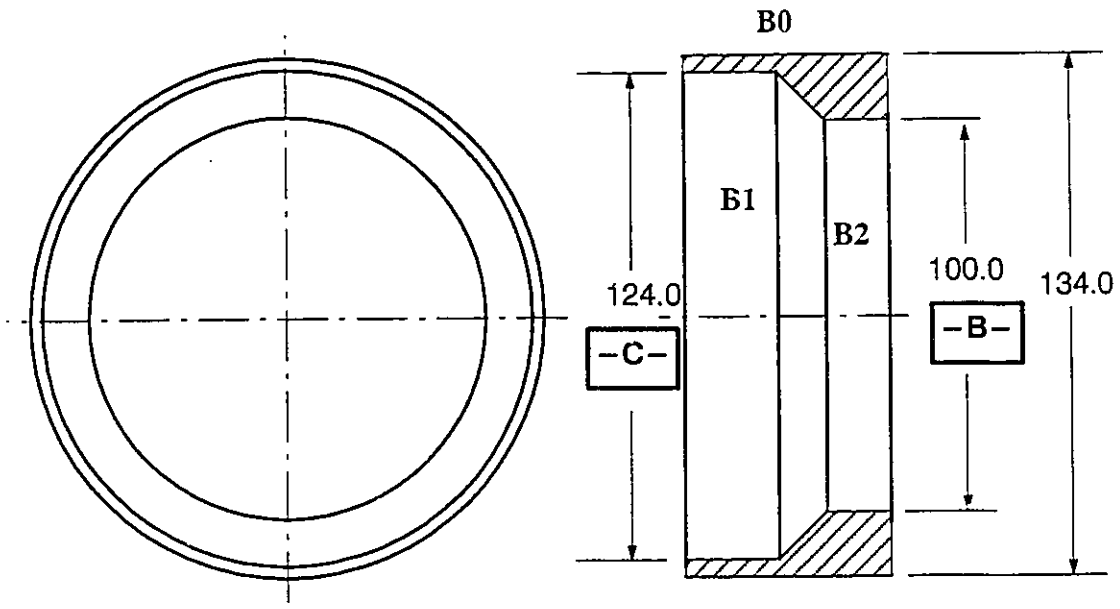
Step	Feature Generated	Datum Reference Frame
1	**C1**	**A–C–F**
	C2	**A–B–F**
2	**C3**	**B–D**
	C4	**C–E**
	C5	**B–D**

Table 6 .3 Manufacturing Sequence of Casing's Features



Feature	Size		Location ⊕		Orientation		
	Min.	Max.	Dat. Ref. Fr.	Val.	Type	Dat.	Val.
C1	134.000	134.023	A–C–F	.014	⊥	B	.012
C2	100.000	100.025	A–B–F	.014	//	D	.010
C3	100.000	100.025	B–D	.015	//	G	.014
C4	134.000	134.023	C–E	.011	⊥	B	.010
C5	134.000	134.023	B–D	.011	//	B	.010

Figure 6 .14 Speed Reducer's Casing



Feature	Size		Location ⌖		Orientation		
	Min.	Max.	Dat. Ref. Fr.	Val.	Type	Dat.	Val.
B1	134.000	134.023	**D–A**	.011	//	**B**	.010
B2	100.000	100.025	**D–A**	.016	//	**C**	.012

Figure 6 .15 Speed Reducer's Bushing

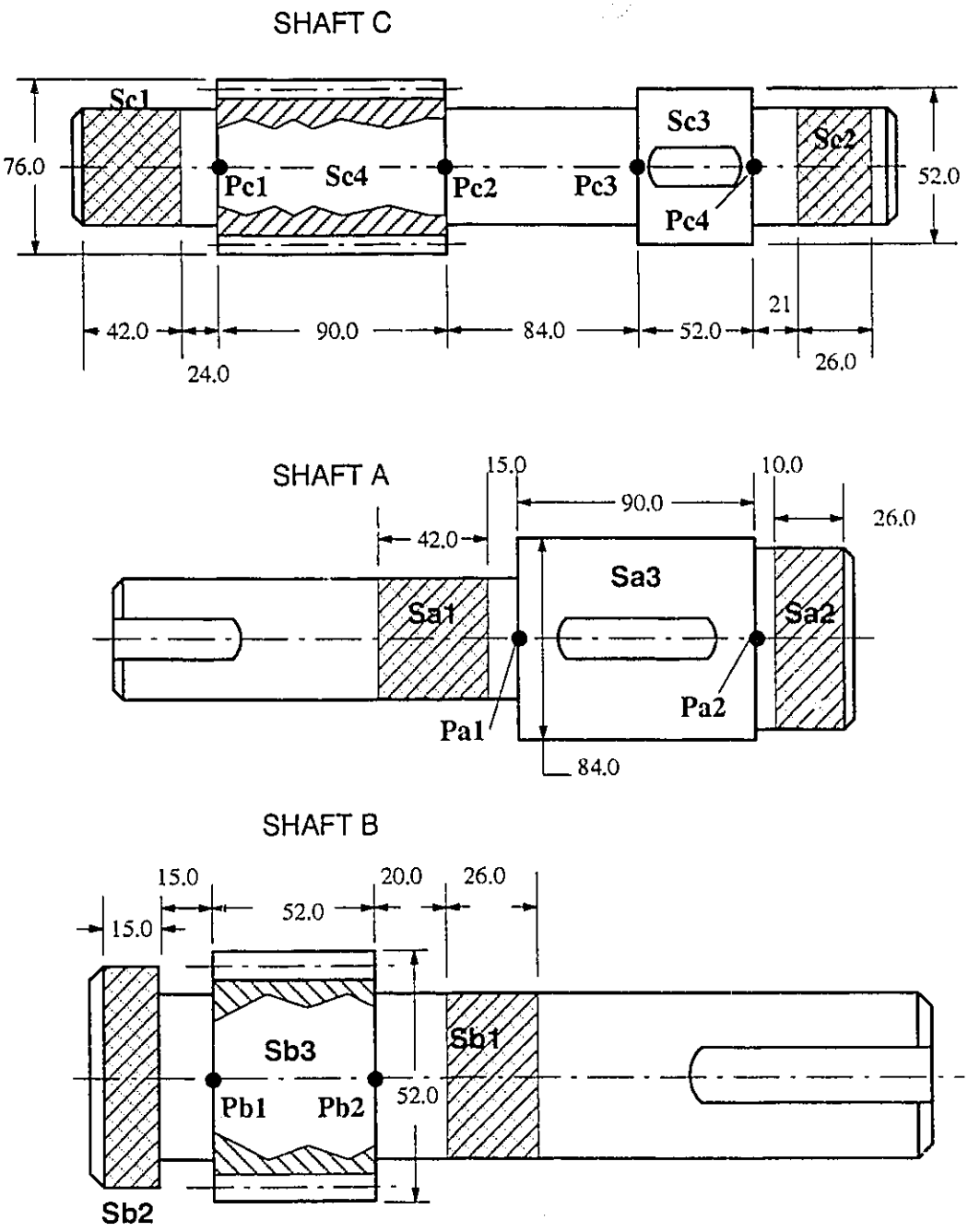


Figure 6 .16 Speed Reducer's Shafts

Step	Feature Generated	Datum Reference Frame
1	**B1**	**D–A**
	B2	**D–A**

Table 6 .4 Manufacturing Sequence of Bushing's Features

6 .5 .2 .2 Assembly Sequence

The assembly sequence of the speed reducer follows the assembly sequence graph shown in Figure 5 .11 . Procedure **ASMSEQ** (section **5 .4 .1 .2**) is used to simulate the assembly sequence, which is described in the following steps:

1. The bushing is press fit into the hole **C5** in the casing.
2. Shaft A is assembled to its bearings at the cross–hatched sections in Figure 6 .16 and the whole sub–assembly is press–fit into holes **C1** and **C2** in the casing.
3. Shaft B is assembled to its bearings at the cross–hatched sections in Figure 6 .16 and the whole sub–assembly is press–fit into holes **C4** of the casing and **B1** in the bushing.
4. Shaft C is assembled to its bearings at the cross–hatched sections in Figure 6 .16 and the whole sub–assembly is press–fit into holes **C3** of the casing and **B2** in the bushing.

6 .5 .2 .3 Functional Requirements

The assembly functional requirements are:

1. The centre distance between the shafts should be within the limits 150 mm $\pm$ 0.01 mm. The centre distance is measured along the centre–lines of the shaft

between the following pairs of points: **Pc1–Pa1, Pc2–Pa2, Pc3–Pb1** and **Pc4–Pb2.**

2. The holes in the casing and the bushing should be maintained within their size tolerance to ensure a proper fit with the bearings.

6 .5 .2 .4 Observations

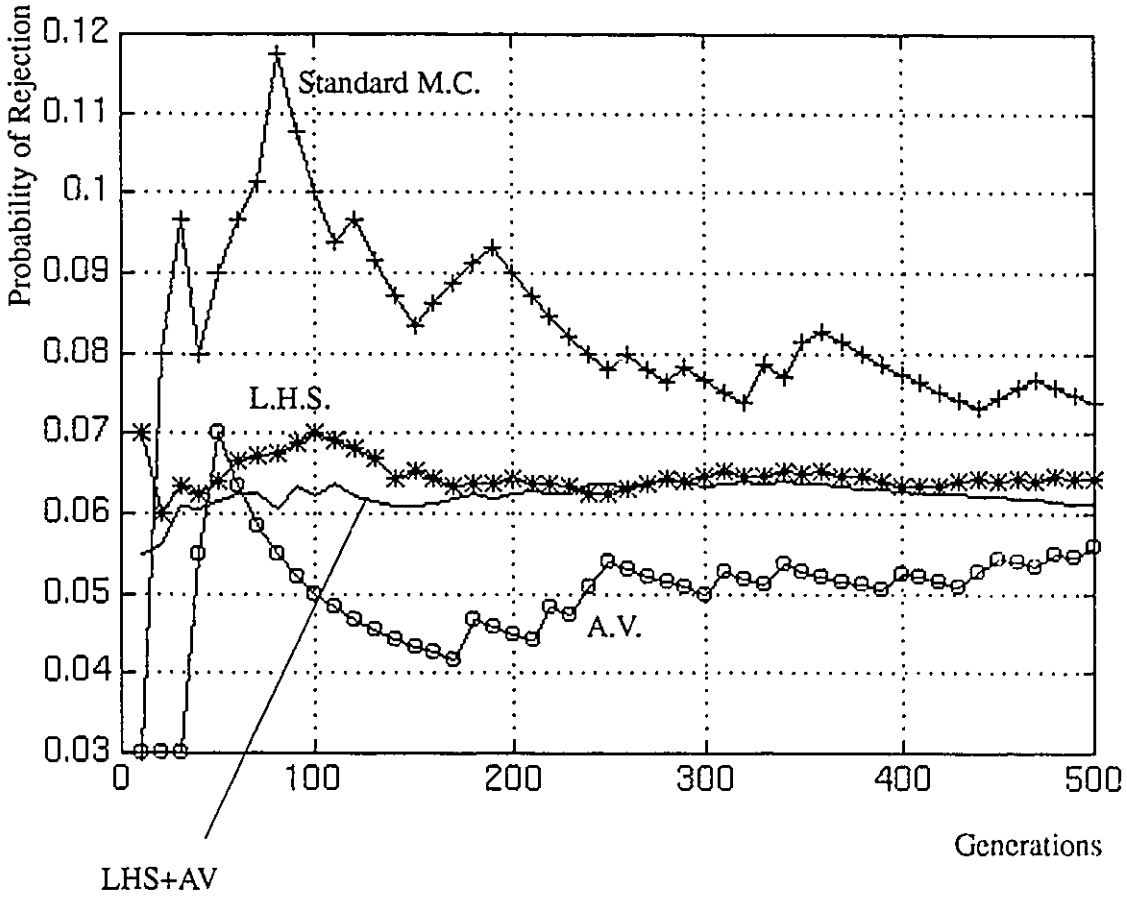


Figure 6 .17 Tolerance Analysis Results. Example–2 (Geometric Tolerances)

The example is simulated by several combinations of the variance reduction techniques (Latin hypercube sampling is used with the number of equal partitions of the

probability functions used m=10) and the simulation is shown in Figure 6 .17 . Figure 6 .17 shows that the standard Monte Carlo simulation or a combination of antithetic variates with standard Monte Carlo simulation need a large number of simulations to converge to the values of the other two runs, showing the increased efficiency obtained by using the Latin hypercube sampling. The number of random variables in the example is much larger than in the two previous ones (N= 8×4 points $\times$ 7 features = 224 random variables). With the increase of the number of random variables, simulations which did not use Latin hypercube sampling showed a lower rate of convergence than those which were used in problems with smaller numbers of random variables.

6 .6 Conclusions

The chapter introduces the idea of incorporating variance reduction techniques with Monte Carlo simulation for tolerance analysis. These techniques are Latin hypercube sampling and antithetic variates. The sample size was decreased to 500 compared to 10000 normally used with standard Monte Carlo simulation.

CHAPTER SEVEN

TOLERANCE ALLOCATION

The work described in the previous chapter had one objective: to describe how to calculate the probability that an assembly does not violate the design functional requirements, given a set of tolerances specified on parts in an assembly, the assembly sequence, and the production variational parameters of each feature. This chapter discusses methods of allocating geometric tolerances to assemblies. The chapter differentiates between selecting manufacturing processes and allocating geometric tolerances to satisfy the assembly functional requirements. In the first case an optimization algorithm is proposed which selects manufacturing processes that minimize the production cost. In the second case, a new criterion for allocating tolerances is proposed. An optimization algorithm using this new criterion is developed for the allocation of geometric tolerance types and values, where each feature has two discrete-valued variables and four continuous ones. The discrete variables are the tolerances of orientation and form. The continuous variables are the magnitudes of these tolerances as well as the magnitude of the size and position tolerances. Genetic algorithms are u r optimization.

7 .1 Tolerance Selection Criteria

7 .1 .1 Dimensional Tolerances

The term tolerance synthesis refers to a class of problems where the values of tolerances imposed on dimensions are selected to satisfy an assembly criterion. Consider the tolerancing problem in Figure 7 .1 of two dimensions x and x' controlled by the tolerances t and t' respectively. The bold lines enclose a region $\mathbf{R_F}$ which represents the feasible set of values of x and x' which satisfy the functional requirements.

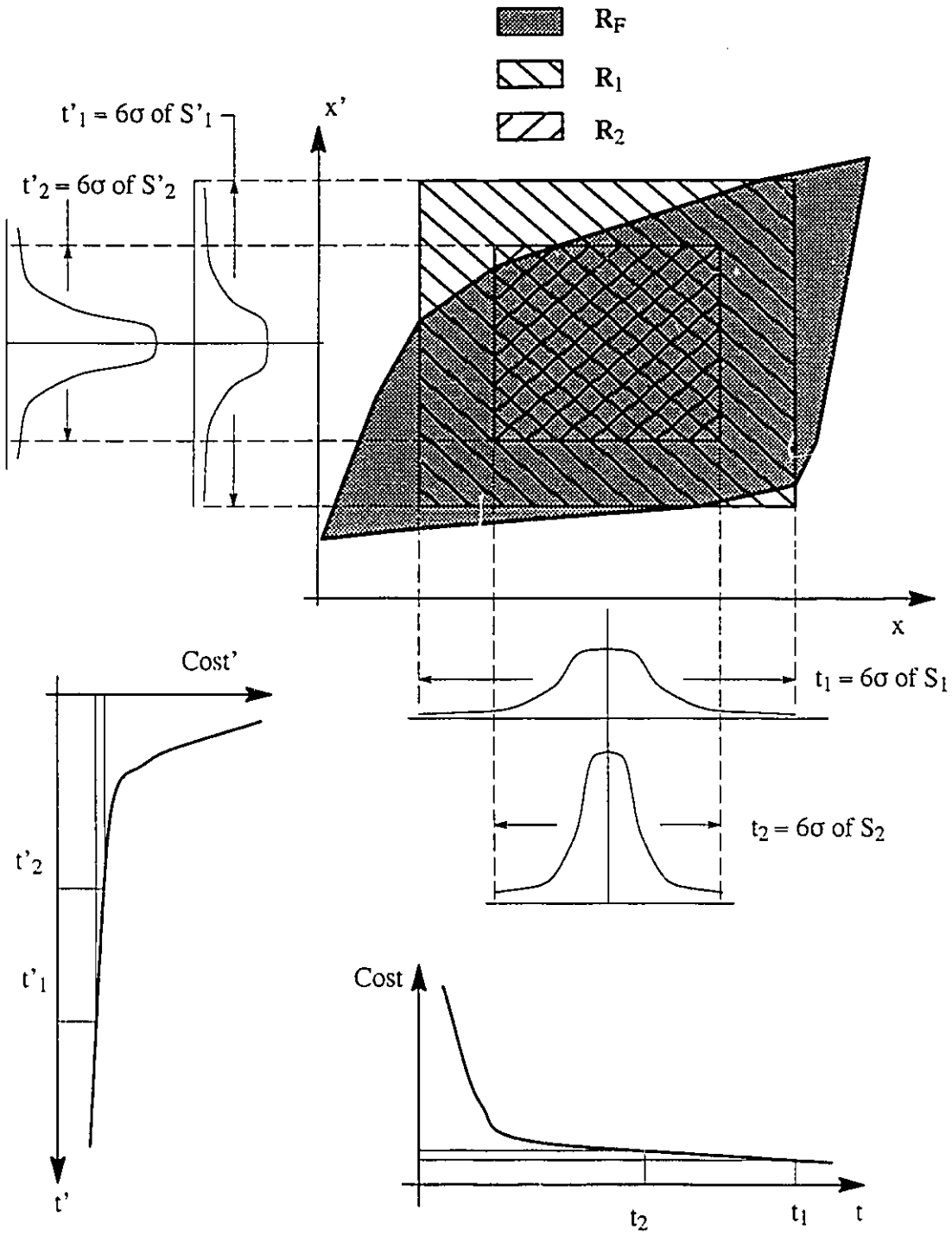


Figure 7 .1 General Formulation of Tolerance Synthesis

Assume that each dimension can be produced by either of two manufacturing processes. Each process is associated with a maximum and a minimum value for the dimensions it produces as well as a probability distribution of the values of the dimensions it produces:

Dimension	Possible Processes and their Limits	
x	S_1 $(x_{1,\min} - x_{1,\max})$	S_2 $(x_{2,\min} - x_{2,\max})$
x'	S'_1 $(x'_{1,\min} - x'_{1,\max})$	S'_2 $(x'_{2,\min} - x'_{2,\max})$

Table 7 .1 Processes Limits

In addition, each process is associated with a unique cost value. Consider the following production cases:

Dimension	Processes of Case 1	Processes of Case 2
x	S_1	S_2
x'	S'_1	S'_2

The range of values of x and x' produced by the processes used in Case 1, cover the area $\mathbf{R_1}$ shown in Figure 7 .1 . The area $\mathbf{R_2}$ covers the range of values produced by the processes in Case 2. Although the manufacturing cost in Case 1 is less than that in Case 2, Case 1 will cause a higher rejection rate of parts with dimensions violating the functional requirements. Hence, to select the proper manufacturing process there is a tradeoff between the manufacturing cost and the rejection rate. At this point the relationship between cost (or rejection rate) and the limits of the manufacturing processes is established. In previous tolerance synthesis literature (Lee, W., 1988 and Kanai et al., 1995) tolerance values were related to the production cost, where each process was associated with a tolerance value. The tolerance value associated with the processes was equal to the difference between the limits

between which each dimension was produced. The reason behind this choice is demonstrated by Figure 7 .2 .

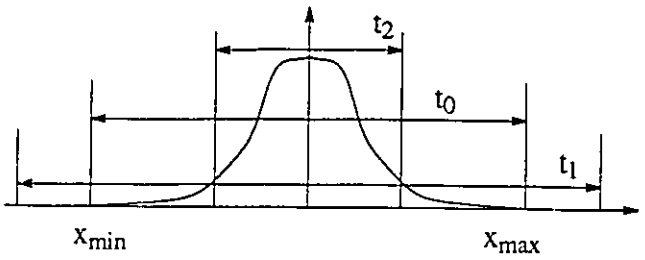


Figure 7 .2 Process limits and dimensional tolerance

It shows a probability distribution of a manufacturing process which produces a dimension x. The values of the generated dimensions range from x_{min} to x_{max}. If the tolerance value is less than the range of dimensional values covered by the distribution (i.e. $t_1 < [x_{max}-x_{min}]$), a number of parts would be unnecessarily rejected during inspection. On the other hand, there is no need to choose a tolerance value greater than the range covered by the distribution (i.e. $t_2 > [x_{max}-x_{min}]$), since no dimensions will be generated outside the process range. Hence instead of relating the cost value to the limits on the processes, they are related directly to the tolerance values. Tolerance values were selected (Lee, W., 1988) by minimizing the production cost while constraining the rejection rate.

7 .1 .2 Geometric Tolerances

The previous section showed how the minimum production cost was used as a criterion for the selection of dimensional tolerances. In this section, a new criterion for selecting geometric tolerances will be proposed. Little research was done in the area of geometric tolerance allocation. The minimum cost criterion was used by previous researchers (Kanai et al. 1995, Nassef and ElMaraghy, 1993 and Turner, 1993a) to allocate

geometric tolerance in an analogy to the selection of dimensional tolerances. There are two main drawbacks for the direct use of cost in the case of geometric tolerances.

The first drawback is that there are many types of geometric tolerances, and relating cost to tolerance values means that for every manufacturing process, a cost–tolerance value for all types of geometric tolerances should be obtained for the features it generates. For example, there should be a cost vs straightness value function, cost vs perpendicularity value (in relation to a specific datum) function, cost vs circularity value function, etc. for a drilling process generating a cylindrical feature. It is hard to obtain a cost vs tolerance value function for all these types of tolerances.

The second drawback is demonstrated by the following example. Consider the punching machine shown in Figure 7 .3 . Assume that all features are produced with perfect geometry except hole B, then consider only two points on B's surface. These are points X_1 and X_2. At the same time assume that the only tolerance imposed on the hole is the position tolerance measured from the datum reference frame **A–B**. The tolerance zone is demonstrated by the dashed lines in Figure 7 .3 . Hence an acceptable hole should have its centre point $x_c=0.5\times(X_1+X_2)$ within the tolerance zone. The functional requirement is that the distance between either of the points X_1 or X_2 and the punch surface should fall within a certain limit. How can the manufacturing cost be used to find the best value of the position tolerance ?

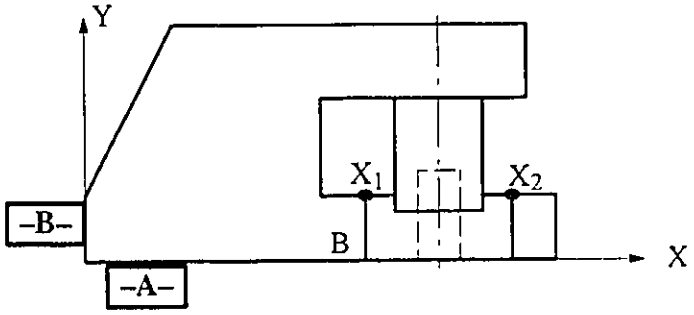


Figure 7 .3 A Simplified Punching Machine

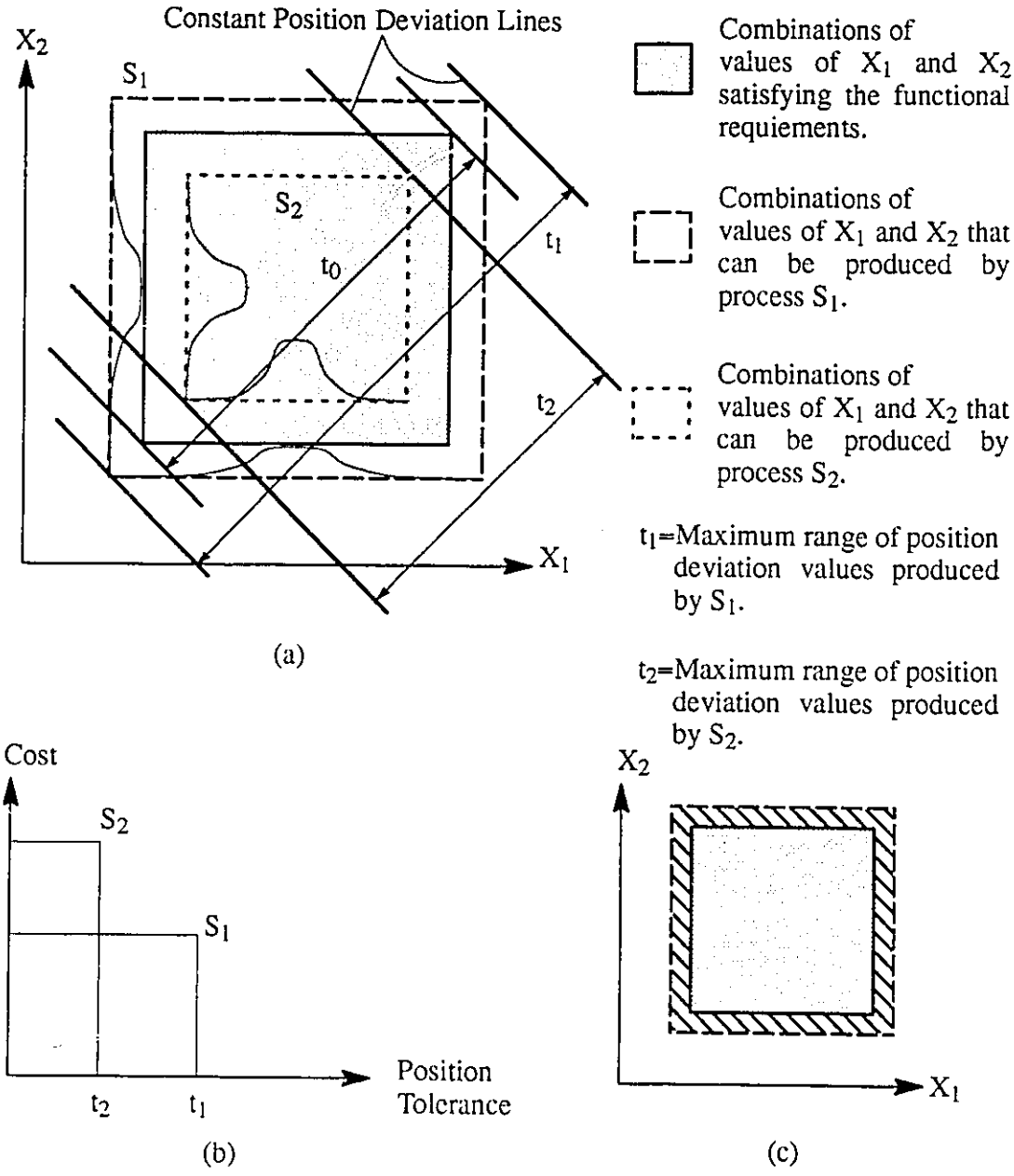


Figure 7 .4 Two Alternative Processes to Produce Hole B

Assume that there are two alternative manufacturing processes to produce hole B (Figure 7 .4), where each process is associated with the maximum range of position

deviation values measured on its produced parts. In analogy to the research done in dimensional tolerances synthesis, let the position tolerance value be equal to the deviations range. Process S_1 has a lower cost (Figure 7 .4 b) and hence would be selected to produce the hole. At the same time process S_1 would produce parts with dimensions that would violate the functional requirements, and these parts would still be accepted during the inspection process. The best value of position tolerance would be equal to t_0. The set of dimensions of these parts is shown in Figure 7 .4 c by the hatched region. Hence, in the case of assemblies with geometric tolerances cost can only be used to select the optimal set of processes, but a new criterion is needed for allocating geometric tolerances.

Figure 7 .5 shows the industrial processes which are used to manufacture assemblies from raw parts. Through these steps, the following processes have different costs associated with them: 1) manufacturing processes, which include the cost of machines, tools, operators and scrap, 2) manufacturing sequences, which dictate the fixturing set-up of parts, 3) inspection processes which include the cost of the inspection machines and operators, and 4) assembly processes, which include the assembly tools and operators. At the same time manufacturing processes are the ones which cause the geometric variability in parts' surfaces that need to be controlled by tolerances. Hence, a manufacturing process which produces parts' surfaces with high variability will produce parts that have a higher probability of violating the assembly functional requirements when assembled to other parts. Since it is a hard task to check whether the functional requirements are within their allowable ranges after the assembly process, tolerances are specified during the design phase on surfaces of individual parts such that a "bad" part (one which will cause a violation of the functional requirements when assembled to other parts) is rejected during inspection. Hence, if geometric tolerances are properly specified in the design phase, then all bad parts will be rejected during inspection, and the assemblies obtained after the assembly process are those

which do not violate the functional requirements. The above discussion is elaborated by the feasibility diagram shown in Figure 7 .6 .

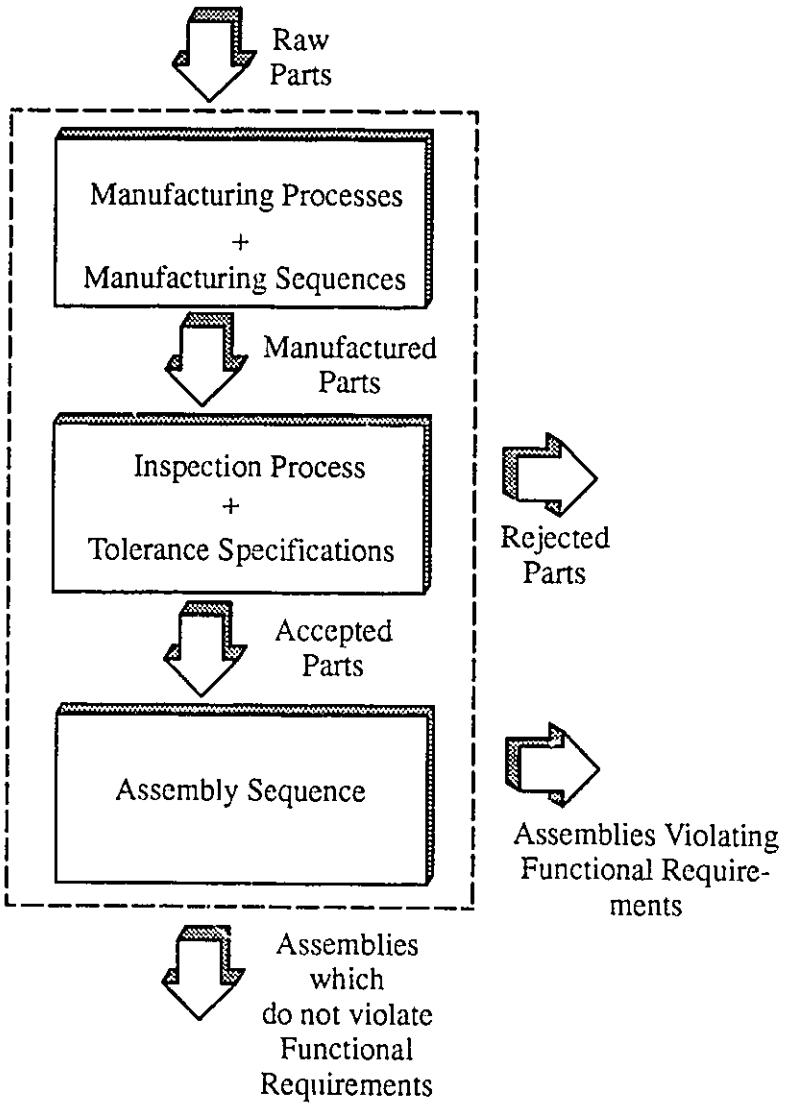
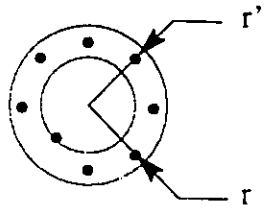
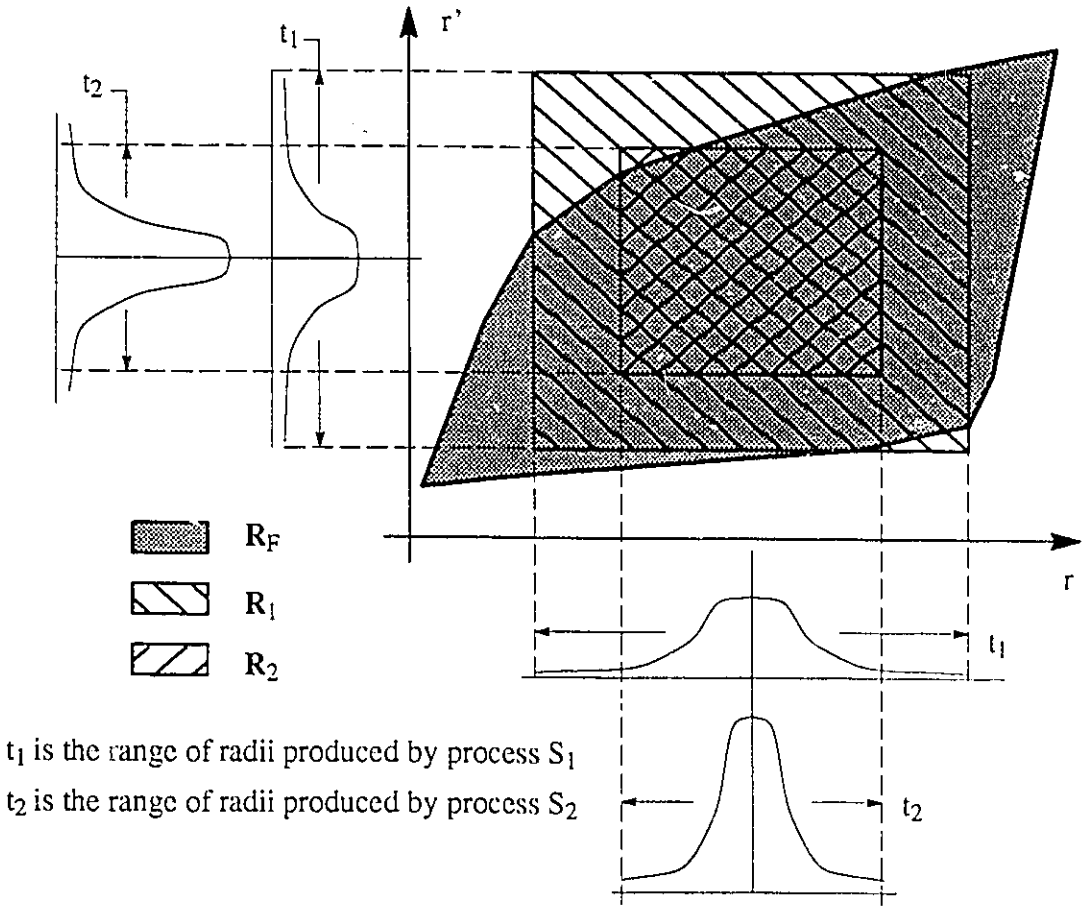


Figure 7 .5 Flow of parts through Industrial Processes



(a)



(b)

Figure 7 .6 Feasibility Diagram for a Cylindrical Feature

Figure 7 .6 (a) shows a top view of a cylindrical feature represented by a set of points

generated from a process distribution. The feature is included in some assembly having a set of functional requirements. Figure 7 .6 (b) shows the effect of the functional requirements on two points with radii r and r' from the points representing the feature. The area enclosed by the bold lines $\mathbf{R_F}$ represents the set of feasible values of r_1 and r_2 satisfying the functional requirements. If the feature is produced with a process S_1, then the values r and r' will vary between the limits of S_1. Since process S_2 has less variance than S_1, the probability that r and r' will satisfy the functional requirements by using process S_2 is higher than that evaluated by using S_1. The tradeoff between maximizing the probability of satisfying the functional requirements and minimizing the cost applies. What is selected here is the process generating each feature. Because it would be difficult to check the assembled product, for the functional requirements, after the assembly process, the geometry of the individual manufactured parts is controlled with a set of geometric tolerances that, when violated a bad part is rejected. The phrase "bad part" stands for a part, when assembled to other parts, will lead to an assembly violating the design functional requirements. The understanding of the relationship between geometric tolerances, assembly functional requirements and manufactured parts is facilitated by drawing the feasible areas over two radii (two points are used for demonstration since it is impossible to view a diagram of the whole feature in hyperspace) of a manufactured feature as shown in Figure 7 .7 . The area enclosed by the dashed lines in Figure 7 .7 ($\mathbf{R_{GT}}$) represent the set of feasible values r and r' satisfying the imposed geometric tolerances. If the imposed geometric tolerances are allocated to capture the design intent, then the set of feasible values of r and r' satisfying the functional requirements should satisfy the geometric tolerances and the set of values r and r' unsatisfactory to the functional requirements should be rejected when checked for the geometric tolerances. In other words, geometric tolerances should be selected to make:

$$\left(\mathbf{R_F} \bigcup \mathbf{R_{GT}}\right) - \left(\mathbf{R_F} \bigcap \mathbf{R_{GT}}\right) = \emptyset \qquad (7.1)$$

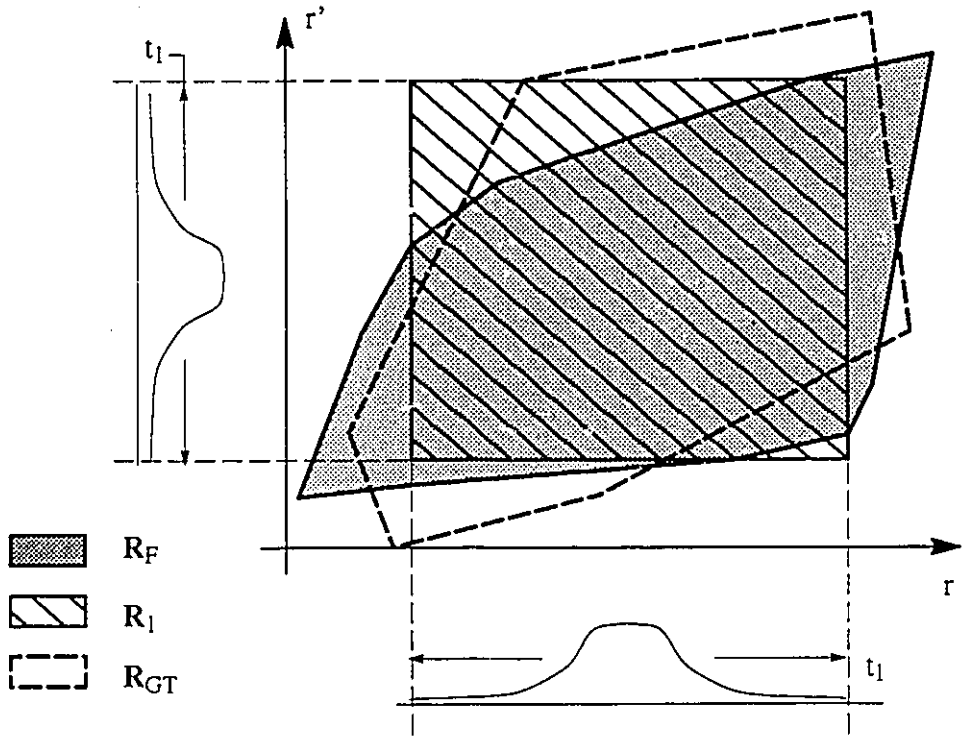


Figure 7 .7 Feasible Spaces for Geometric Tolerances and Functional Requirements

Hence, the selection of geometric tolerances can be done in two steps:

1. Select the manufacturing processes that will minimize the production cost and does not exceed a certain probability of rejection (due to the violation of functional requirements).

2. Select the types and values of geometric tolerances that will minimize:

$$\min_{\text{geometric tolerances}} \left[\left(\mathbf{R}_F \bigcup \mathbf{R}_{GT} \right) - \left(\mathbf{R}_F \bigcap \mathbf{R}_{GT} \right) \right] \qquad (7.2)$$

7 .2 Process Selection

The selection of manufacturing processes is modelled as a combinatorial optimization problem as shown in the following sections. The selection of the manufacturing sequence of each part is incorporated in the formulation.

7 .2 .1 Cost vs. Process Limits Function

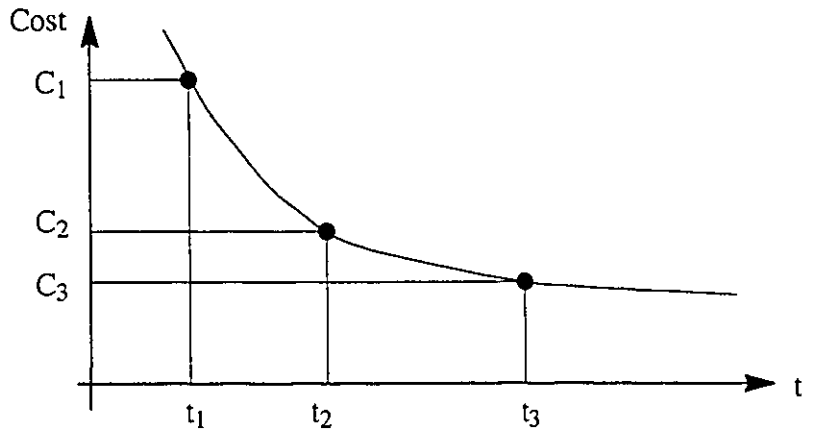


Figure 7 .8 Cost vs Process Limits

The discrete cost – tolerance model shown in Figure 7 .8 is proposed by Lee, W. J. (Lee, W., 1988). Assuming that the features produced are cylindrical features, the values t_1, t_2 and t_3 in the model refer to the limits between which each process generates its surface. For example, process S_1 has a range t_1 within which the radii of a feature are produced. Each process S_i has a cost C_i associated with it. The cost value reflects the combination of man, machine and material (Lee, W., 1988).

7 .2 .2 Manufacturing Sequence

Manufacturing sequences affect the assembly functional requirements as well as the total cost of production. Consider the simplified model of a punching machine shown in

Figure 7 .9 . The base can be machined by two sequences dictated by the position tolerancing schemes shown in Figure 7 .9 (a) and Figure 7 .9 (b). The manufacturing sequences for both schemes are shown in Figure 7 .10 with the datum simulators shown by dashed lines. In the first scheme, the part should be clamped in a way such that basic locations of features are measured from the origin established by the datum reference frame **A–B–C**. Features b_2 and b_1 are then manufactured at axes located with basic dimensions 40 cm and 190 cm respectively and measured from the reference frame origin O_1. In the second scheme, the part is first clamped at the datum reference frame **A–B–C**. Feature b_2 is manufactured, at an axis located by basic dimensions of 40 cm in the X direction and 60 cm in the Y direction from the origin of the frame of axes O_1. Feature b_2 is then used as a secondary datum feature (i.e. the axis of the the largest inscribed cylinder to the actual feature and normal to datum A is used as a datum from which measurements are taken). A datum axis is established from b_2 with origin O_2 as the origin from which the location of feature b_1 is measured (150 cm in the X direction). The following assumptions are made, to investigate the effect of manufacturing sequences on the clearance:

1. The holes b_1 and b_2 are manufactured using manufacturing processes which produce features whose radii have normal probability distributions with the parameters shown in Table 7 .2 :

b_1	Process 1	$\mu_1 = 21.0$	$6\sigma_1 = 0.05$
b_2	Process 2	$\mu_2 = 32.5$	$6\sigma_2 = 0.05$

Table 7 .2 Manufacturing Processes Parameters for Punch

2. Each feature is represented by 8×4 points.

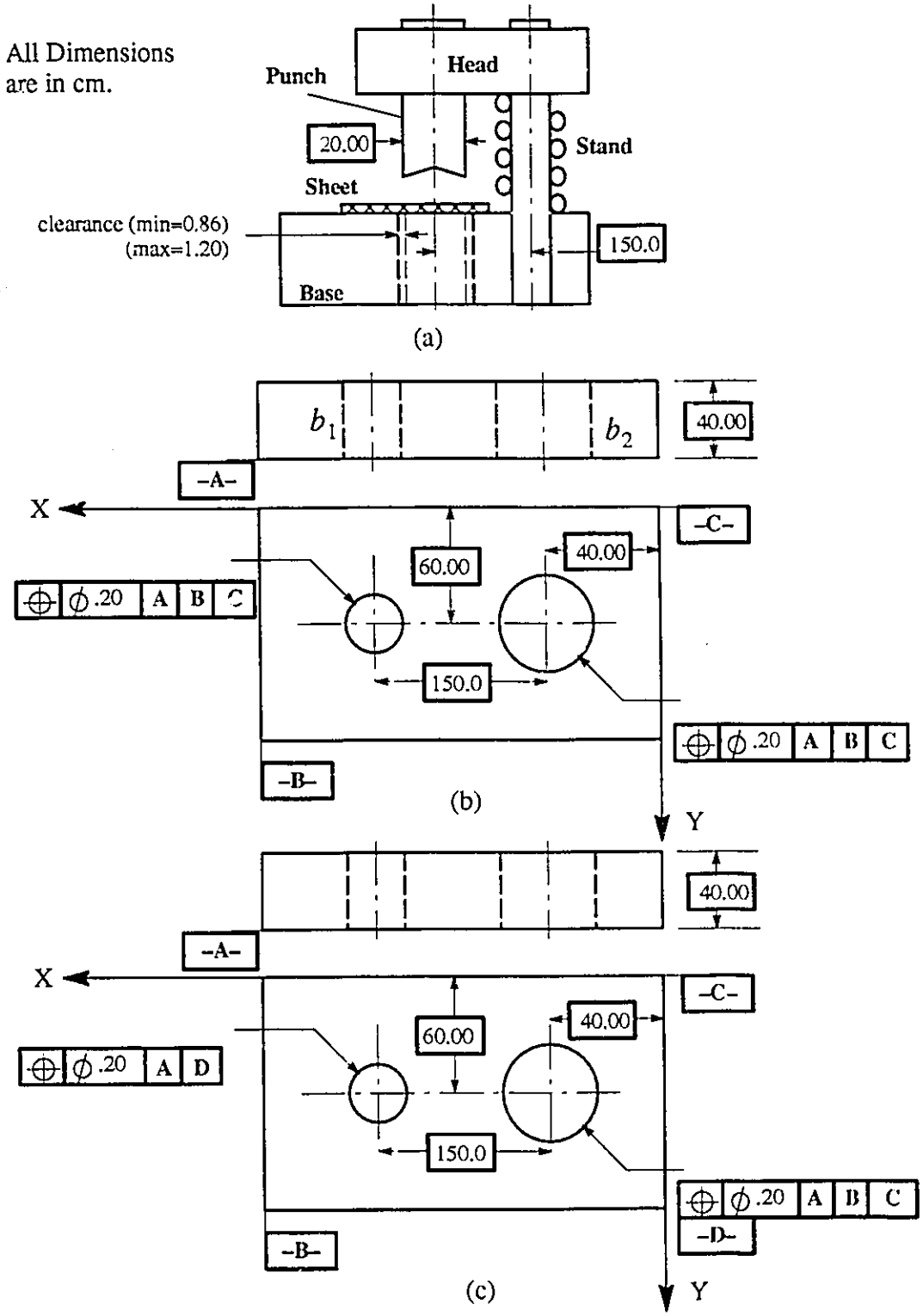
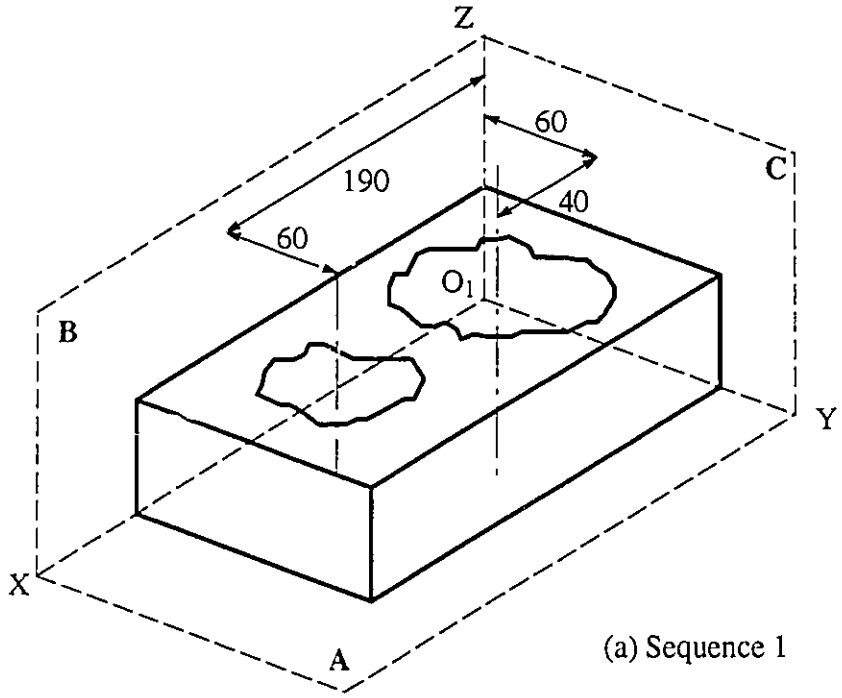
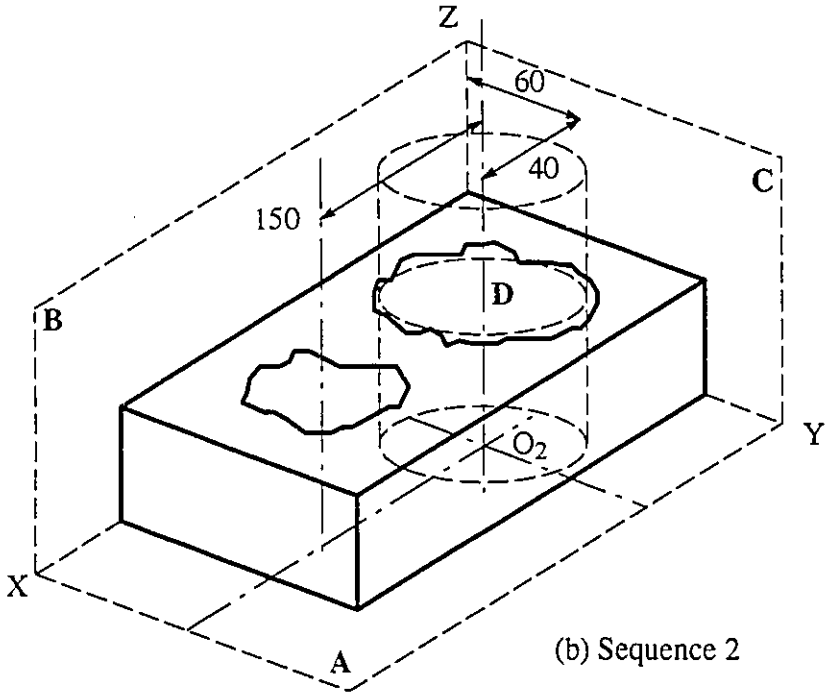


Figure 7 .9 Alternative Position Tolerance Schemes for Punch's Base



(a) Sequence 1



(b) Sequence 2

Figure 7 .10 Manufacturing Sequences for Base

3. The stand is assumed to be press-fit into b_1. All other surfaces are assumed to be at their nominal geometry.

The example is simulated using Monte Carlo simulation described in procedure **TOL-AN2** with 500 samples. The probability that each sequence will produce parts violating the limits on the clearance (i.e. the assembly functional requirement) is evaluated along with the mean values and the variances of the maximum and minimum clearances. The results of the simulation is shown in Table 7 .3 .

	P_{rej}	Minimum Clearance	Maximum Clearance
Sequence 1	P_{rej}=0.0114	μ=0.9896 σ^2=0.0022	μ_1=1.0707 σ^2=0.0021
Sequence 2	P_{rej}=0.0024	μ=1.0020 σ^2=0.0071	μ_2=1.0589 σ^2=0.0071

Table 7 .3 Means and Variances of Maximum and Minimum Clearances

Results in Table 7 .3 show that when using the second manufacturing sequence, the variance of the clearance between the punch and the hole b_1 decreases, leading to a lower probability of rejection. The above example showed how different manufacturing sequences affect the assembly functional requirements. In addition, different manufacturing sequences necessitate different fixturing methods and consequently would have different costs associated with them.

7 .2 .3 Selection of Manufacturing Processes and Sequences

Assume that an assembly has a set of *Np* parts *p* and a set of *Nf* features *f*.

$$p = \{p_i\} \quad , \quad i \in \{1, 2, \ldots, Np\} \tag{7 .3}$$

$$f = \{f_j\} \quad , \quad j \in \{1, 2, \ldots, Nf\} \tag{7 .4}$$

Each feature f_j can be produced by a set of possible manufacturing processes s where feature f_j is allowed to be produced by one process s_k. The term manufacturing process stands for one machine or a combination of machines used to generate the feature, along with other effects like personnel and material.

$$s = \{s_k\} \quad , \quad k \in \{1, 2, \ldots ., Ns_j\} \tag{7.5}$$

where Ns_j is the number of possible processes for feature f_j. At the same time each part p_i has a set of possible manufacturing sequences $\boldsymbol{d}$

$$\boldsymbol{d} = \{d_l\} \quad , \quad l \in \{1, 2, \ldots, Nd_i\} \tag{7.6}$$

where Nd_i is the number of possible sequences for part p_i. Each process s_j has a manufacturing cost value MC_k as shown in section **7.2.1**. Another cost value DC_l is associated with each manufacturing sequence, thus the total cost of producing an assembly where each feature f_j is produced by a specific process s_k and where the production of every part p_i follows a specific manufacturing sequence d_l is:

$$\text{Cost} = \sum_{i=1}^{Np} \text{DC}_l(p_i) + \sum_{i=1}^{Nf} \text{MC}_k(f_j) \tag{7.7}$$

The problem of selecting processes and manufacturing sequences is combinatorial in nature. The number of possible combinations of processes and sequences for an assembly is equal to:

$$\text{Number of combinations} = \prod_{i=1}^{Np} Nd_i \times \prod_{i=1}^{Nf} Ns_j \tag{7.8}$$

possible combinations of datum reference frames and manufacturing processes.

7 .2 .3 .1 Formulation

The problem of selecting manufacturing processes and manufacturing sequences can be formulated as a combinatorial optimization problem, where each feature f_j has one discrete independent variable X_j (The manufacturing process by which it is produced) and each part p_i has another discrete independent variable Y_i (The manufacturing sequence by which it is produced). Hence the total number of independent variables is equal to $Np+Nf$. Since the datum reference frame of the position tolerancing scheme of each part must follow the manufacturing sequence of the part, another formulation can be devised where each feature f_j can have a variable X_j, which stands for the manufacturing process and another variable Z_j which is the datum reference frame used in its position tolerance. Since some reference frames cannot be used with others, each candidate reference frame can have a set of constraints on its usage as an attribute. Given the reference frame of each feature in a part, procedure **MCSEQ** (section **5 .2 .2**) can be used to deduce the manufacturing sequence for each part. This formulation is further elaborated in the next example.

The objective function to be minimized is the total production cost. The probability that the manufactured parts and features will violate the assembly functional requirements is constrained to a specified value v. The objective function is given by the following equation.

$$\min_{\mathbf{X},\mathbf{Z}}(\text{Cost}(\mathbf{X},\mathbf{Z})) \quad , \quad \text{subject to} \quad P_{rej}(\mathbf{X},\mathbf{Z}) \leq v \tag{7 .9}$$

Since genetic algorithms have been used in combinatorial optimization successfully (Goldberg and Kuo, 1986) and lead to results better than those obtained by other methods (e.g. branch and bound method), they are used to minimize the function in equation (7 .9). The use of genetic algorithms for combinatorial optimization is discussed in appendix **A**.

The following procedure gives a general outline of the use of genetic algorithms in selecting manufacturing processes and manufacturing sequences:

Procedure: SEL_PROC

1. Given an assembly with a set of parts and features, a set of possible manufacturing processes to manufacture each feature and a set of possible datum reference frames for each feature, code the independent variables into binary strings as shown in section **A.1 .2** .
2. Generate a random population of the strings.
3. Use the following objective function to evaluate the fitness of each string:
   a. For all independent variables **X** and **Z** calculate the total cost C(**X**,**Z**) using equation (7 .7).
   b. Evaluate the probability of rejecting the assembly due to the violation of the assembly functional requirements (P_{rej}), using procedure **TOL–AN2.**
   c. If P_{rej} is less than the allowable limit *v*, let the fitness function value be equal to zero; otherwise deliver the value V – C(**X**,**Z**) as the fitness function value where V is some arbitrary large number (larger than the maximum value of C(**X**,**Z**)).
4. Apply the genetic operators discussed in Appendix **A** on the population, generating a new population.
5. Repeat steps 3 and 4 for the specified number of maximum generations.

7 .2 .3 .2 Example

The speed reducer shown in section **6 .5 .2** is used to test the proposed procedure with the same functional requirements and assembly sequence. The following cost functions

are used for the selection of manufacturing processes, to manufacture the holes in the casing and the bushing.

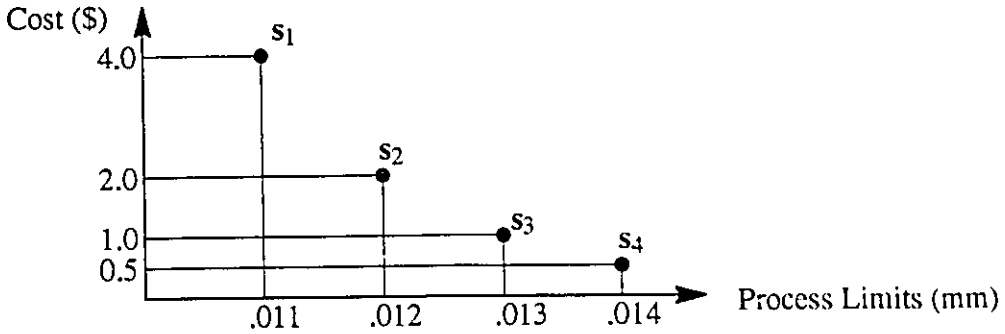


Figure 7 .11 Cost vs. Process Limits (Features **C1, C4** and **C5**)

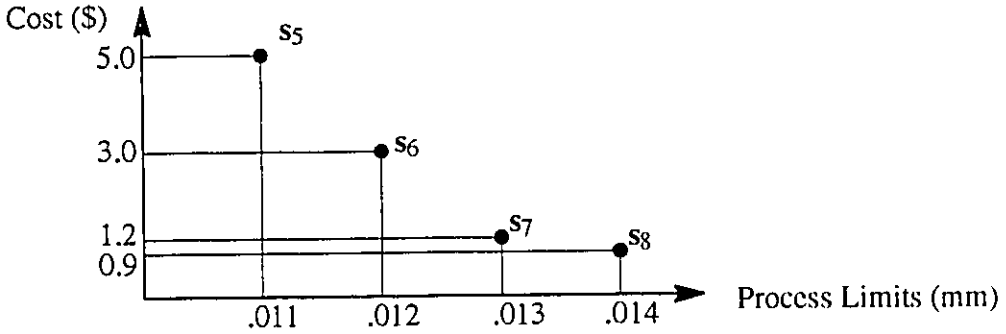


Figure 7 .12 Cost vs. Process Limits (Features **C2** and **C3**)

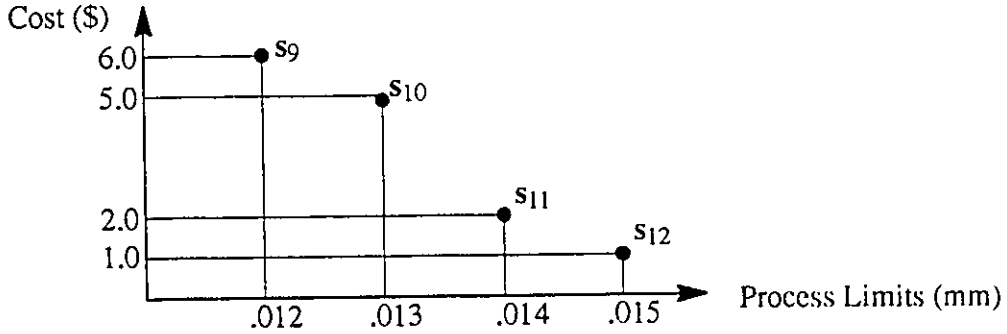
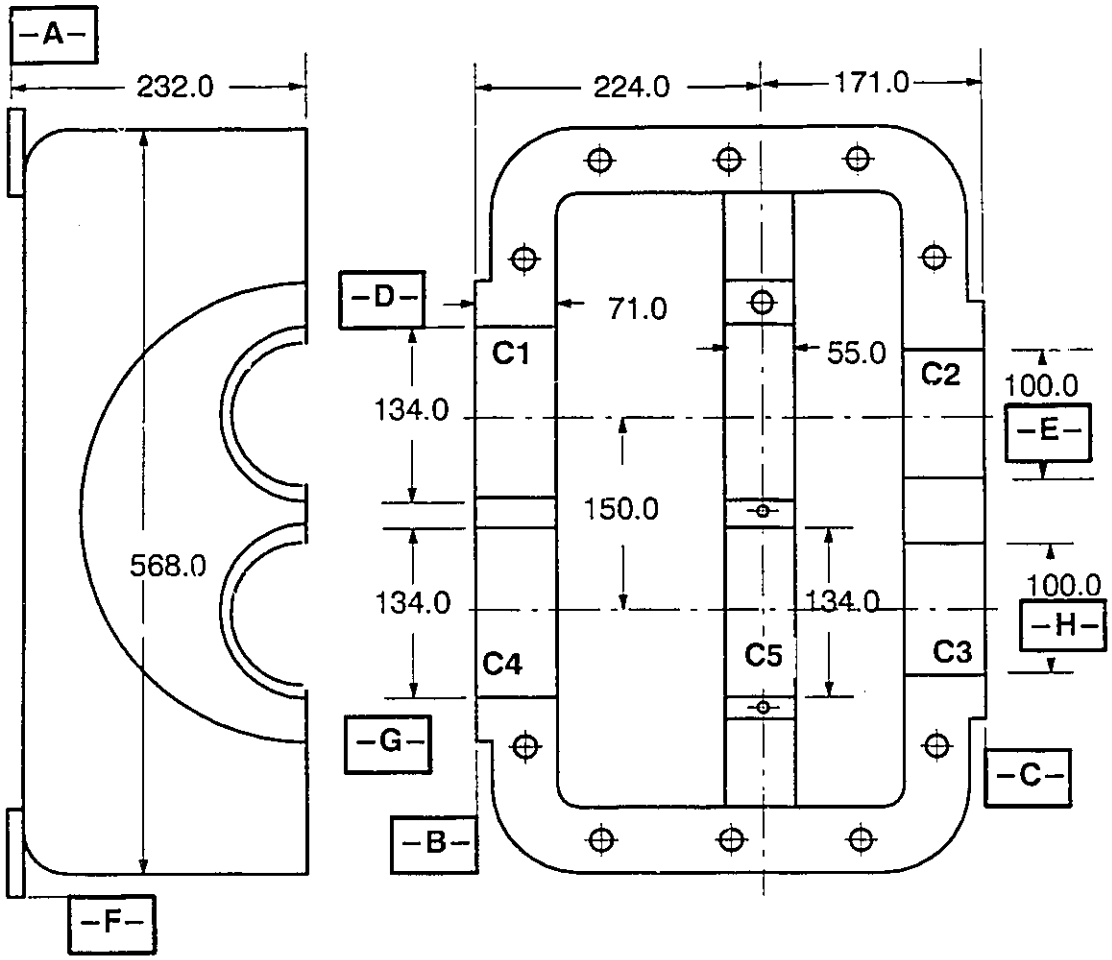


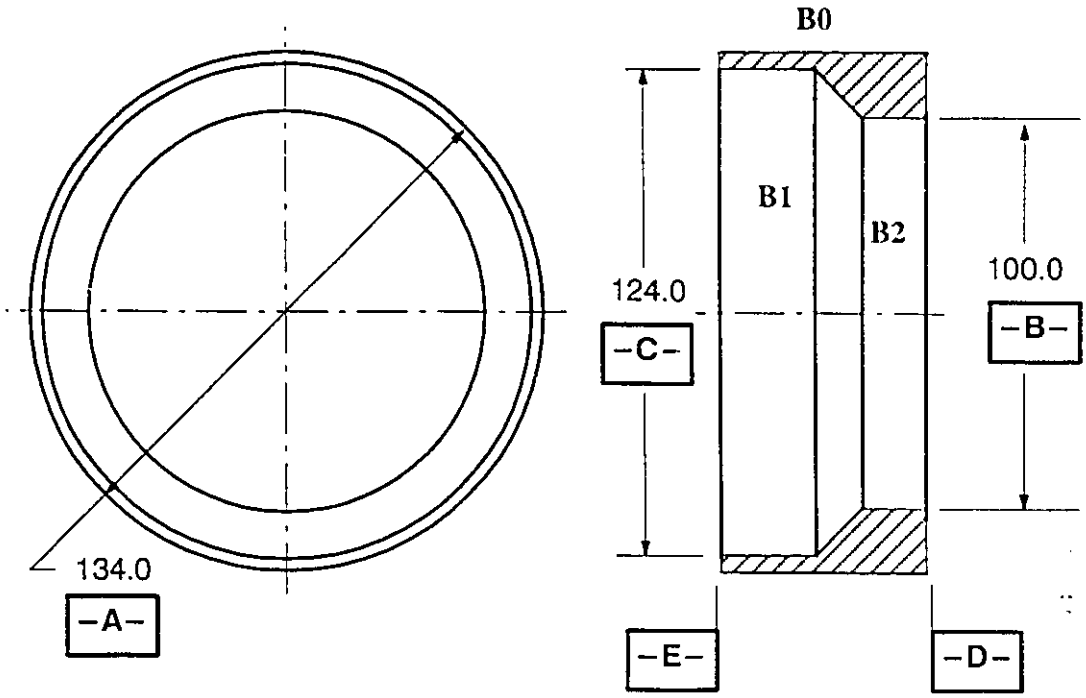
Figure 7 .13 Cost vs. Process Limits (Features **B1** and **B2**)



Feature	Candidate 1	Candidate 2
C1	⊕ A-C-F	⊕ C-E *C2
C2	⊕ A-B-F	⊕ B-D *C1
C3	⊕ B-D	⊕ C-E
C4	⊕ B-D	⊕ C-E
C5	⊕ B-D	⊕ C-E

CONSTRAINT
***C2** means that this reference frame can be used only if feature **C2** is manufactured

Figure 7 .14 Candidate Reference Frames for Casing



Feature	Candidate 1	Candidate 2
B1	⌖ D-A	⌖ D-B *B2
B2	⌖ D-A	⌖ D-C *B1

Figure 7 .15 Candidate Reference Frames for Bushing

Figure 7 .14 and Figure 7 .15 show the candidate datum reference frames for the features of the casing and the bushing respectively. Note that some datum reference frames cannot be used unless other features are manufactured. This is demonstrated by feature **C1**, where the second candidate datum reference frame (**C–E**) cannot be used unless the feature **C2** is manufactured before **C1**. If such a case is encountered during the genetic algorithms search, the fitness function (section **A.2**) is set to zero, thus diverting the genetic algorithms search from such a case.

It is assumed that the use of manufactured holes as datums makes the manufacturing setup harder than when all datums are planar. Thus whenever one of the holes **C1**, **C2**, **B1** or **B2** is used as a datum an extra cost of $0.5 is added to the total cost. The allowable range of probability of rejection is:

$$0 \leq P_{rej} \leq 0.0063 \qquad (7.10)$$

Procedure **SEL_PROC** is used to find the optimal processes and datum reference frames using the following genetic parameters:

1. DIFF value (section **A.1 .1**) = 0.001 (i.e. chromosome length = 9).
2. Population size = 100 (a number recommended by Reeves (1993) for chromosome lengths in the range of 8–10).
3. Uniform cross–over with a probability = 0.4 (a low cross–over probability is used to eliminate possible early loss of high fitness chromosomes).
4. Tournament selection.
5. Mutation probability = 0.01.

Figure 7 .16 shows the curve of cost vs. genetic generations while Table 7 .4 gives the values of the optimum selections of manufacturing processes and datum reference frame.

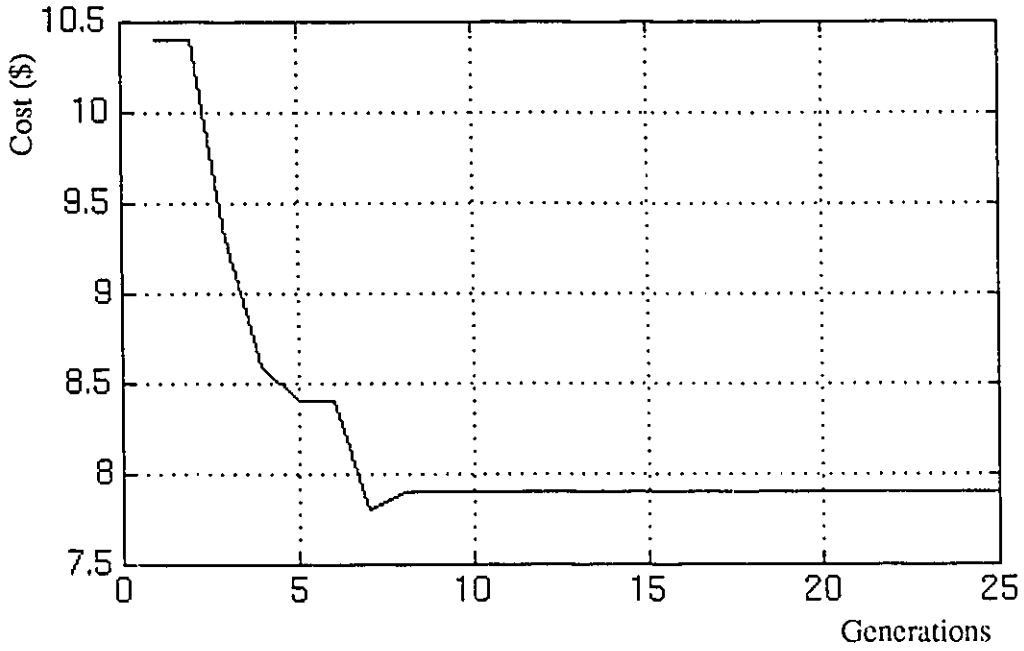


Figure 7 .16 Cost vs. Generations Result

Feature	C1	C2	C3	C4	C5	B1	B2
Manufacturing Process	s_4	s_7	s_7	s_4	s_4	s_{11}	s_{11}
Datum Reference Frame	A–C–F	A–B–F	B–D	C–E	B–D	D–A	D–A
Probability of rejection due to the violation of functional requirements = 0.0057							

Table 7 .4 Optimal Selections of Manufacturing Processes and Reference Frames

The rejection probability obtained by using the manufacturing processes and datum reference frames shown in Table 7 .4 equals 0.0057. The reason that this probability is not exactly equal to the desired value of 0.0063 is that the problem is of a discrete nature, where the search can find solutions near to the desired ones. The manufacturing sequence for the casing is deduced from the obtained datum reference frames using procedure **MCSEQ**:

1. Clamp the casing at the datum reference frame **A–C–F**, then machine hole **C1**. Next, clamp the casing at the datum reference frame **A–B–F** then machine hole **C2**.
2. Establish a secondary datum **D** at the hole **C1**, then use the reference frame **B–D** as a reference of measurement to machine the holes **C3** and **C5**.
3. Establish a secondary datum **E** at the hole **C2**, then use the reference frame **C–E** as a reference of measurement to machine the holes **C4**.

7 .3 Allocation of Geometric Tolerances

Section 7.2 differentiated between the the selection of manufacturing processes and the selection of geometric tolerances. Geometric tolerances should be chosen such that assemblies would satisfy the design functional requirements and "bad" parts would be rejected during the inspection process. This statement necessitates the formulation of the following objective function.

7 .3 .1 Objective Function

Geometric tolerances should be selected to minimize the region $\mathbf{R_m}$. (Section 7 .1 .2):

$$\min(\mathbf{R_m}) = \min_{\text{geometrictolerances}} \left[\left(\mathbf{R_F} \bigcup \mathbf{R_{GT}} \right) - \left(\mathbf{R_F} \bigcap \mathbf{R_{GT}} \right) \right] \qquad (7.11)$$

where $\mathbf{R_m}$ is the region where the imposed geometric tolerances do not match the assembly

functional requirements, such that an instance of the assembly may be acceptable to the geometric tolerances but unacceptable to the assembly functional requirements or vice versa.

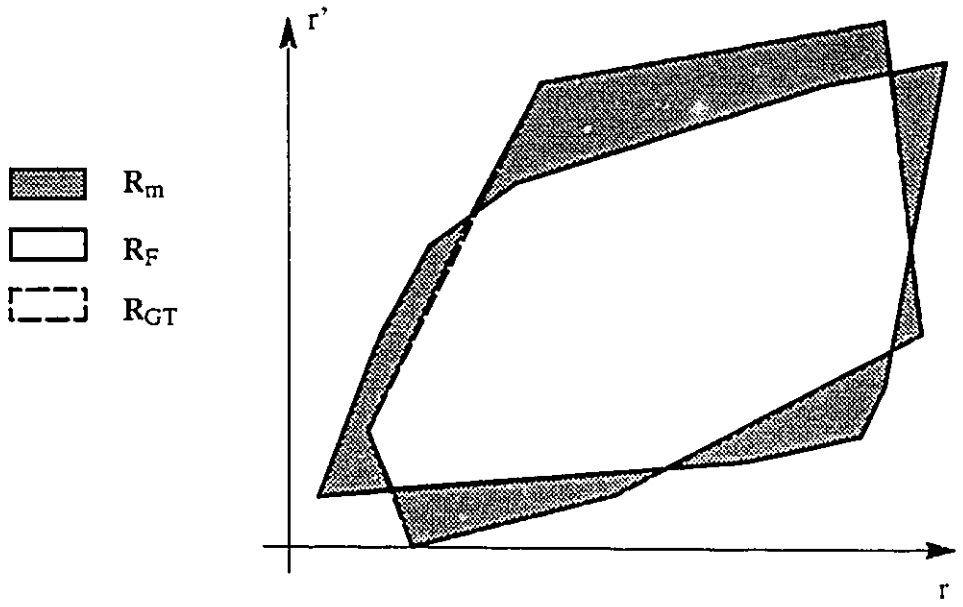


Figure 7 .17 Mismatch Region

Figure 7 .17 shows the region $\mathbf{R}_m$ under consideration. Assume that the range of values of each radius r_i is covered by a uniform distribution, then the integral:

$$P_m = \int\int\cdots\int_{(r_1,r_2,\ldots,r_n)\in \mathbf{R}_m} u(r_1,r_2,\ldots,r_n)\ dr_1 dr_2 \ldots dr_n \qquad (7.12)$$

is equal to the probability that the assembly will belong to the region $\mathbf{R}_m$, where $u(r_1,r_2,\ldots,r_n)$ is the joint uniform distribution over all radii and n is the number of radii (random variables). In the following sections P_m would be referred to as the mismatch probability. Hence, if the geometric tolerances are allocated such that all parts, that are accepted during inspection, will produce assemblies that satisfy the design functional requirements, P_m would be equal to zero. A combination of Latin hypercube sampling and antithetic variates can be used to evaluate equation (7 .12) as described in the following procedure. The reason for choosing

the uniform distribution to associate with each radius is that unless the associated probability distributions form a joint probability distribution having a shape similar or at least close to the region $\mathbf{R}_m$ (Rubinstein, 1981), the sample mean Monte Carlo method would be less efficient than the hit-and-miss method using uniform distributions. Since the shape of $\mathbf{R}_m$ is not known before-hand, then the uniform distribution was associated with every random variable. Yet the evaluation of P_m can be enhanced by incorporating Latin hypercube sampling and antithetic variates within the simulation.

procedure: EVAL_REG

1. Assume a uniform distribution over the range of possible values of radii produced by the manufacturing processes. Divide the probability functions of the distributions into m equal probability intervals. Let N_s be the number of samples required for simulation and let N be the number of random variables (radii on features) to be simulated.

2. **repeat** N_s times,

   a. Using the uniform probability distribution associated with each feature and the manufacturing sequence simulation procedure **MCSEQ** (section **5 .2 .2**), generate m values of all random variables using Latin hypercube sampling. Thus an $N \times m$ matrix r of the generated values is obtained:

$$r = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_i \\ \vdots \\ r_N \end{bmatrix} = \begin{bmatrix} r_{1,1} & r_{1,2} & r_{1,3} & \cdots & r_{1,m} \\ r_{2,1} & r_{2,2} & r_{2,3} & \cdots & r_{2,m} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ r_{i,1} & r_{i,2} & r_{i,3} & \cdots & r_{i,m} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ r_{N,1} & r_{N,2} & r_{N,3} & \cdots & r_{N,m} \end{bmatrix} \qquad (7.13)$$

   b. For each random variable r_i rearrange the m generated values in a random permutation, i.e. rearrange the rows of matrix r in a random permutation:

c. Find the antithetic variate of each element of the r matrix, thus forming another matrix of $\bar{r}$ of antithetic variates.

d. Since each column of the r and $\bar{r}$ matrices forms an instance of a set of manufactured parts, then **for** each column in the r matrix **do**

i. Calculate the geometric deviations on all features using the algorithms described in chapter **4** and check them for the imposed tolerances.

ii. Simulate the assembly sequence using the assembly sequence procedure **ASMSEQ** (section **5 .4 .1 .2**), then check the assembly functional requirements.

iii. **if** the assembly requirements **are** in spec and the tolerances **are not** in spec **or if** the assembly requirements **are not** in spec and the tolerances **are** in spec **then**

Increment Na = (Counter for belonging to $\mathbf{R}_m$)

else

Increment Nf (Counter for not belonging to $\mathbf{R}_m$).

end–if

e. **end–for**

f. Repeat the loop of step "d" for the $\bar{r}$ obtaining $\overline{\mathrm{Na}}$ and $\overline{\mathrm{Nf}}$.

3. **end–repeat**

4. Calculate the probability that the assembly instances will belong to $\mathbf{R}_m$.

$$P_m = 0.5 \times \left(\frac{\mathrm{Na}}{\mathrm{Na} + \mathrm{Nf}} + \frac{\overline{\mathrm{Na}}}{\overline{\mathrm{Na}} + \overline{\mathrm{Nf}}} \right) \qquad (7.14)$$

The value of P_m can be used as an objective function to allocate geometric tolerances.

7 .3 .2 Tolerance Types

Assume that the orientation of a cylindrical feature is to be controlled. A perpendicularity tolerance with respect to one datum can be one tolerance type to use, or a parallelism tolerance with respect to another datum can be another type. Some tolerance types can lead to a higher acceptance rate of parts that will not satisfy the assembly functional requirements, or else lead to unnecessary rejection of functional parts. In other words, one tolerance type can lead to higher values of $\mathbf{P}_m$ than others as shown in the following sections.

7 .3 .2 .1 Orientation Control

Table 7 .5 shows two tolerancing schemes for controlling the orientation of holes b_1 and b_2 in the simplified punch example shown in Figure 7 .9 . They both have the same tolerance value of ϕ 0.2 cm.

Scheme	b_1		b_2		P_m
	Type	Datum	Type	Datum	
1.	⊥	A	⊥	A	0.032
2.	⊥	A	//	D	0.023

Table 7 .5 Effect of Types of Orientation Tolerances on P_m

The value of $\mathbf{P}_m$ was evaluated for both schemes using procedure **EVAL_REG**. Scheme 2 has a smaller value $\mathbf{P}_m$ and hence is more acceptable than scheme 1.

7 .3 .2 .2 Form Control

Table 7 .6 shows two tolerancing schemes to control the form of hole b_1 of the simplified punch shown in Figure 7 .9 . They both have the same tolerance value of ϕ 0.1

cm. Procedure **EVAL_REG** is used to evaluate $\mathbf{P_m}$ for both schemes. Scheme 2 shows that controlling the feature's form with cylindricity does not produce as much mismatch between the rejection due to the violation of the functional requirement as circularity.

Scheme	Type	P_m
1.	○	0.063
2.	⌭	0.059

Table 7 .6 Effect of Types of Form Tolerances on P_m

7 .3 .2 .3 Formulation

The selection of geometric tolerances to satisfy the assembly functional requirements can be formulated as a mixed programing problem, where each feature f_i in the assembly has two discrete–valued independent variables (O_i and F_i) for the types of tolerance controlling its orientation and form deviations respectively and four real–valued independent variables (X_i, Y_i, Z_i and U_i) for the magnitudes of the tolerances of size, position, orientation and form respectively. The datum reference frame for the position tolerance is already determined during the selection of manufacturing processes (section **7 .2 .3 .1**). Controlling the size of a feature needs only one type of tolerance (size tolerance). The objective function to be minimized is the mismatch probability $\mathbf{P_m}$ which is calculated using procedure **EVAL_REG.**

$$\min_{\mathbf{O,F,X,Y,Z,U}} (P_m(\mathbf{O, F, X, Y, Z, U})) \qquad (7 .15)$$

Since both real and discrete variables can be coded into genetic strings (section **A.1**) genetic algorithms are used for the minimization of $\mathbf{P_m}$ with the following procedure:

procedure: SEL_TOL

1. Given an assembly with a set of parts and features, a set of possible types of form and orientation tolerances to control its features and a range of allowable values for the magnitudes of size, form, orientation and position tolerances, code the independent variables into binary strings as shown in section **A.1 .2** .
2. Generate a random population of the strings.
3. Use the following objective function to evaluate the fitness of each string:
   a. For all independent variables **O, F, X, Y, Z** and **U**, calculate the value of P_m using procedure **EVAL_REG**, then use it to evaluate the fitness of every string in the population.
4. Apply the genetic operators discussed in Appendix **A** to the population, generating a new population.
5. Repeat steps 3 and 4 for the specified number of maximum generations.

It is worth noting that a set of constraints concerning the inspection time for checking every geometric tolerance can be used within the above procedure. An estimate of the time needed to inspect the part using every geometric tolerance can be made and the total inspection can be constrained to not exceed a certain value.

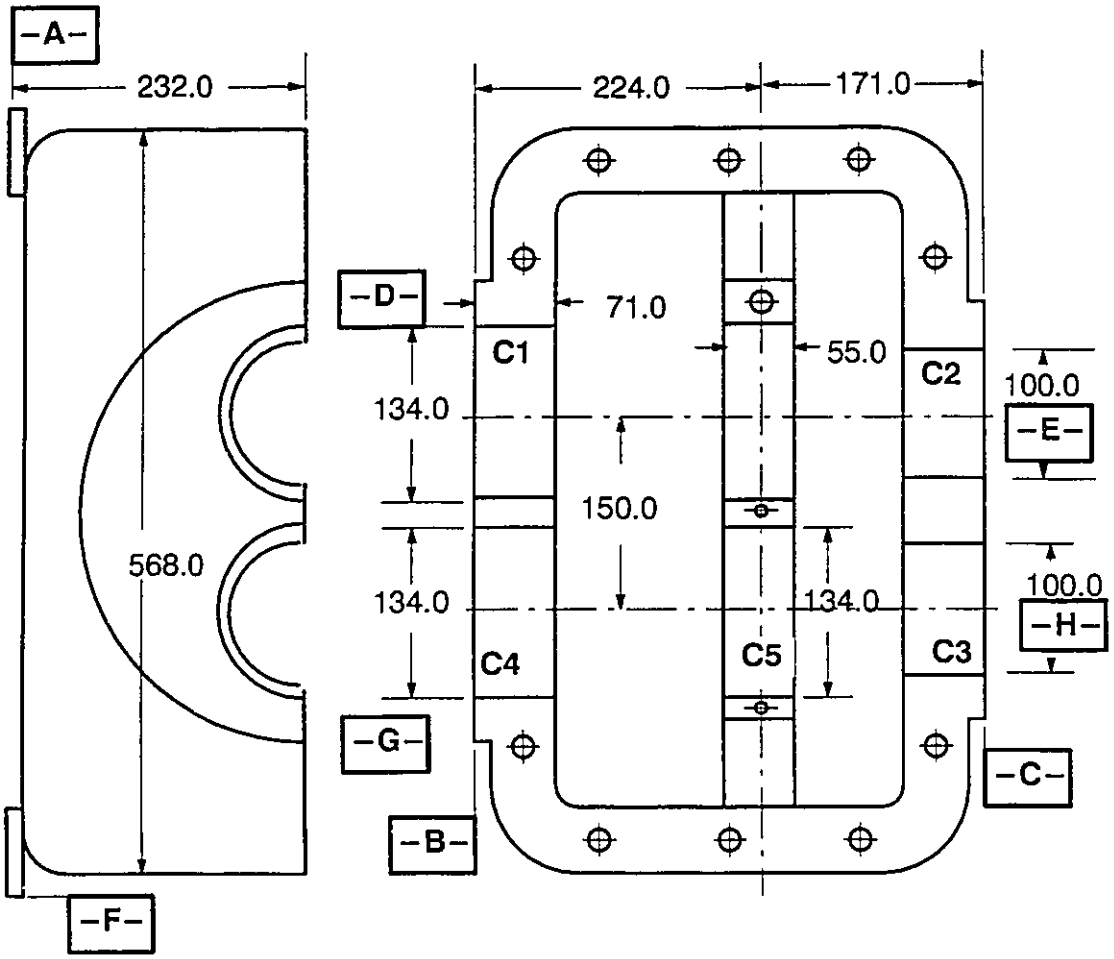
7 .3 .3 Example

The speed reducer shown in section **6 .5 .2** is used to test procedure **SEL_TOL** using the same functional requirements and assembly sequence described in section **6 .5 .2** . The optimal position tolerancing scheme obtained in Table 7 .4 is used to control the location of the features in the candidate tolerance types for the controlling the orientation of the various

features in the reducer. Figure 7 .18 and Figure 7 .19 show the candidate types of orientation tolerances as well as the ranges of tolerance magnitudes for each type of feature control. Since the size tolerance of all holes is used as a functional requirement, it is not included in the independent variables, and consequently the form control is specified directly by the size tolerance. The optimal value of the independent variables is given in Table 7 .7 , while the curve of P_m vs. the genetic algorithms search generations is given in Figure 7 .20 . The genetic algorithms were used with the following parameters:

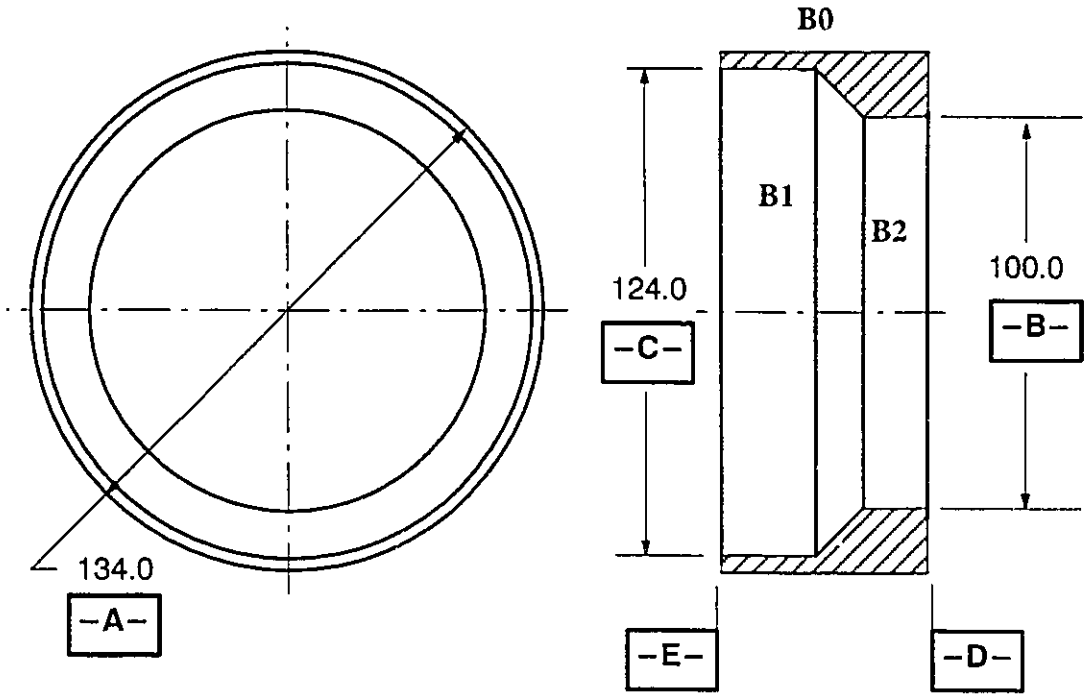
1. DIFF value (section **A.1 .1**) = 0.001.
2. Population size = 200.
3. Uniform cross–over with a probability = 0.4.
4. Tournament selection.
5. Mutation probability = 0.01.

The evaluated mismatch probability P_m equals 0.0067 for the selected tolerance types and values. The reason that the value does not equal zero is due to the discretization of the real independent variables by the genetic algorithms coding, thus a near optimal solution is obtained rather than the optimal one. The tolerance magnitudes in Table 7 .7 show some features with a position tolerance magnitude less than the magnitude of the orientation tolerance.



Feature	Tolerance Types Candidates				Range of Tolerance Magnitudes for Position and Orientation Control	
	1	2	3	4	Min.	Max.
C1	⊥B	⊥E			.008	.016
C2	⊥B	//D				
C3	⊥B	//D	//G			
C4	⊥B	//D				
C5	⊥B	//D	//G	//H		

Figure 7 .18 Candidate Tolerances for Casing



Feature	Tolerance Types Candidates				Range of Tolerance Magnitudes for Position and Orientation Control	
	1	2	3	4	Min.	Max.
B1	⊥D	//A	//B		.008	.016
B2	⊥D	//C				

Figure 7 .19 Candidate Tolerances for Bushing

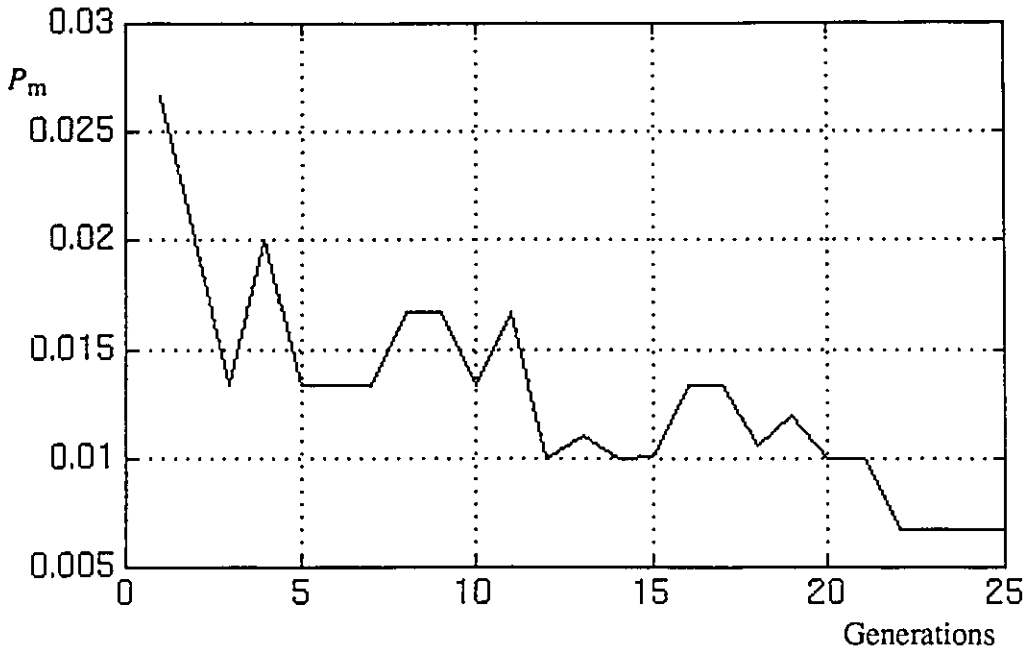


Figure 7 .20 P_m vs Generations Result

Feature	C1	C2	C3	C4	C5	B1	B2
Type of Orientation Tolerance	⊥B	//D	//G	⊥B	⊥B	//B	//C
Magnitude of Orientation Tolerance	.012	.011	.016	.016	.015	.011	.011
Magnitude of Position Tolerance	.014	.014	.015	.011	.011	.011	.016
$P_m = 0.0067$							

Table 7 .7 Optimal Selections of Tolerance Types and Magnitudes

Since the orientation tolerances should act as a refinement for the position tolerance, their magnitudes must be less than those of the position tolerance. Thus it can be deduced that for features with position tolerances having smaller magnitudes than orientation tolerances, the position tolerance was enough to control the feature's position and orientation and thus the use of an orientation tolerance is redundant and can be neglected.

7 .4 Conclusions

A formulation for the optimal selection of manufacturing processes and sequences for the minimum production cost using genetic algorithms has been developed. The algorithm is an extension of a formulation presented in previous work by W. Lee (1988), and includes manufactured surface variations as well as manufacturing sequences. The optimal manufacturing sequence is used to deduce the datum reference frame for each part's position tolerance. The algorithm was tried on a speed reducer example and converged to a rejection rate value near to the desired one. A new criterion for allocating geometric tolerances has been developed. That is the minimum mismatch probability, which is defined as the probability that assembly parts will be rejected during inspection while being acceptable to the design functional requirements when assembled, or vice versa. A study was carried out to demonstrate the effect of different tolerance types on the functional requirements. It showed the necessity of including tolerance types into the allocation problem. A new algorithm has been developed for the allocation of geometric tolerance types and magnitudes to minimize the mismatch probability using genetic algorithms and was tried on a speed reducer example. The algorithm found the optimal set of tolerances equivalent to a near zero mismatch probability. The developed algorithms demonstrate a methodology which can be used to convert the geometric tolerance allocation problem from a human-based trial and error process into a mathematically formulated problem searching for the best solution. The proposed algorithms do not completely exclude the expertise of a tolerance designer. The

choice of the set of possible tolerance types for each feature is still made by the designer. Because the objective function was quite complex, the algorithm took a relatively long duration (six hours on an SGI workstation). This long duration does not pose any engineering drawback as the problem solved is a problem which is solved once at the design stage.

CHAPTER EIGHT

CONCLUSIONS

8 .1 Introduction

The objective of the research presented in this dissertation was to develop an algorithm for the allocation of geometric tolerances types and magnitudes to assembly features which satisfy the assembly functional requirements and ensure that good parts are not rejected unnecessarily or bad parts are assembled causing the assemblies to violate the functional requirements. To achieve this objective, several sub–problems had to be addressed which include:

1. The formulation of a suitable mathematical criterion to allocate geometric tolerances such that parts which are accepted during inspection are those which conform to the functional requirements when assembled and parts which are rejected during assembly are those which will violate the design functional requirements if assembled.
2. The formulation of an algorithm which allocates tolerance types in addition to tolerance magnitudes.
3. The development of an algorithm for the probabilistic analysis of assemblies with geometric tolerances.
4. The development of a set of algorithms for stacking tolerances and checking the conformance of the manufactured assemblies to the functional requirements.
5. The development of a set of algorithms, for checking the conformance of manufactured parts to the imposed geometric tolerances, which evaluate the global minimum geometric deviations as indicated by the tolerancing standards.

How these problems were addressed is summarized in the following section along with some concluding remarks and statements about possible future research.

8 .2 Summary and Conclusions

A set of algorithms was developed to address the different problems mentioned in the previous section. These algorithms are for:

A. Evaluation of Geometric Deviations.

B. Simulation of Manufacturing Sequences.

C. Simulation of Assembly Sequences.

D. Tolerance Analysis.

E. Selection of Manufacturing Processes.

F. Selection of Tolerance Types and Magnitudes.

8 .2 .1 Evaluation of Geometric Deviations

Algorithms were developed, implemented and simulated to evaluate geometric deviations based on CMM data or data generated from a Monte Carlo simulation. The geometric deviations investigated were those of size, form, orientation, and position. Genetic algorithms were used whenever the deviation was calculated. A NURBS parametric surface is interpolated to the generated points, in order to enhance the calculation of some deviations.

8 .2 .1 .1 Concluding Remarks

The use of a global optimization method such as genetic algorithms ensured the calculation of the global minimum deviation zones. This was proved by conducting comparisons with other methods. Optimization methods which linger in local minima caused the evaluation of some geometric deviations to be higher than the actual value, which

may cause an unnecessary rejection of good parts. The use of a parametric surface to represent the actual generated surface made the evaluation of the local size deviation possible according to its new definition in the ANSI Y14.5.1M (1995) standards. However, the interpolated surface is needed most when the number of generated points is small as in the case of tactile sensing CMMs; otherwise, if the number of points is huge (as in the case of scanning with a laser camera), the surface points can be used directly for evaluating those deviations.

8 .2 .1 .2 Future Research

The new ANSI Y14.5.1M (1995) standards provide a mathematical definition for geometric tolerance zones. Algorithms for some of those deviations were developed in this dissertation, but a large number of deviations is still not yet addressed. Of particular importance is the new definition of primary datums. The standards differentiate between two mathematical criteria for tolerances. These are the conformance of a manufactured part to a specified tolerance value and the actual value of geometric deviation. In the analysis of geometric tolerances, a part's conformance to tolerance is sufficient to conduct the analysis, and since checking a conformance is a search problem which does not necessitate the evaluation of global minimum for the geometric deviations, its use would enhance the efficiency of the tolerance analysis. Hence a set of algorithms is needed to verify parts conformance to tolerances. Another issue that needs to be taken into account is the presence of measurement errors within the measured points. Hence, if a parametric surface representation is used to approximate the actual manufactured feature, it should not be made to pass through the measured points, but rather fit within their space according to the expected value of the measurement error.

8 .2 .2 Simulation of Assembly Sequence

Since a closed form mathematical representation relating the functional requirements to the geometric tolerances is difficult to develop, the approach followed to verify that the generated surfaces conform to the allocated tolerances and whether the assembly conforms to the functional requirements is to: 1) simulate the manufacturing sequence and generate the surfaces using the manufacturing process probability distribution, 2) evaluate the geometric deviations on the generated surfaces using the algorithms mentioned in section **8 .2 .1** , 3) simulate the assembly sequence, and 4) check the assembly's conformance to the functional requirements.

8 .2 .2 .1 Simulation of the Manufacturing Sequence

The manufacturing sequence by which the part features are generated dictates the fixturing set–up of the part, and hence dictates the datum reference frame for the part's position tolerance. Since each manufacturing sequence dictates a unique fixturing set–up and therefore has a unique cost value associated with it and has a unique effect on the functional requirements, it is necessary the various features be generated in the same sequence as that specified by the manufacturing sequence. An algorithm is implemented for simulating such a sequence.

8 .2 .2 .2 Generation of Surface Points

An algorithm was implemented to generate surface points from the manufacturing process probability distribution. It was assumed that each process generated a surface which varied between two ideal offset surfaces and that a probability distribution of the actual surface points location within the offset distance is associated with the process. The surface points were the random variables, and the imposed geometric tolerances as well as the assembly functional requirements were functions of these random variables. The

multinormal distribution was proposed for representing the variation in the surface points to take into account the covariance of the generated points. Once the parts surfaces are generated, they are checked for the imposed tolerances using the algorithms mentioned in section 8 .2 .1 .

8 .2 .2 .3 Assembly Simulation

After the parts are generated they are connected to each other using the designed assembly sequence. An assembly graph was proposed, where parts are the nodes and assembly connections are the links to be used in the simulation and an algorithm was proposed for deducing the assembly sequence from the graph. Several types of assembly connections were modeled for the purpose of positioning parts in the assembly which include shrink fits and bolted fasteners. An algorithm for searching for the optimal placement of bolted parts was developed using genetic algorithms.

8 .2 .2 .4 Checking Functional Requirements

Two types of functional requirements were considered. These are limits on critical dimensions and limits on clearances. An algorithm was developed for the evaluation of the maximum and minimum values of clearances using genetic algorithms. The algorithm had the capability of detecting surface interferences.

8 .2 .2 .5 Concluding Remarks

Since Monte Carlo simulation is the most appropriate method of probabilistic analysis of assemblies with geometric tolerances, the simulation of the assembly sequence is inevitable to check for the assembly's conformance to the functional requirements. The functional requirements are related to the generated surface points and tolerances should be treated as constraints on the deviations evaluated on the surface points. This formulation reflects the real world and differentiates between parts rejected due to the violation of the

imposed tolerances and others rejected due to the violation of the functional requirements. Such a differentiation is essential for allocating tolerances to satisfy the functional requirements. The algorithm developed for the relative placement of bolted parts has the advantage of finding the global optimum placement of bolted parts due to the use of genetic algorithms. Using genetic algorithms to find the global maximum and minimum values of a clearance ensures that the clearance limits are properly evaluated.

8 .2 .2 .6 Future Research

The dissertation addressed assembly modes which were used in the examples in its last two chapters. The area of modeling assembly connections has been unexplored in the tolerancing research. A thorough investigation of this issue would have to enumerate all types of assembly modes and develop algorithms for their simulation. The functional requirements addressed in this dissertation are ones which can be quantified. However, there are functional requirements that are not quantified in practice (e.g. a location fit). Methods for quantifying these requirements are needed.

8 .2 .3 Probabilistic Analysis of Assemblies with Geometric Tolerances

An algorithm for the probabilistic analysis of assemblies with geometric tolerances was developed using Monte Carlo simulation combined with two variance reduction techniques (Latin hypercube sampling and antithetic variates), which used the assembly simulation algorithm mentioned in section 8 .2 .2 .

8 .2 .3 .1 Concluding Remarks

Although the Monte Carlo simulation has long been used in tolerancing, the introduction of variance reduction techniques is new to the field. Previous researchers either used other methods which are unsuitable for geometric tolerances (Lee, W., 1988), or limited the simulation to a small number, arguing that a completely accurate evaluation is not needed

(Lee and Johnson, 1993). The developed algorithm decreased the number of samples needed for the analysis, and ensured that the rejection probabilities would not be miscalculated.

8 .2 .3 .2 Future Research

Monte Carlo simulation has been the most widely used method for tolerance analysis. In each simulation step, all features are generated and if the geometric deviation of one of the features does not fall within the range of the specified tolerance, the whole instance of parts is rejected. This method is more conservative than reality. A new formulation of the probabilistic analysis of geometric tolerances is needed to avoid the unnecessary rejection of parts within the simulation.

8 .2 .4 Allocation of Geometric Tolerances

Two algorithms were developed for the allocation of geometric tolerances to assemblies. The first algorithm selects the optimum set of manufacturing processes for features and the optimal set of manufacturing sequences for parts to minimize the production cost and keep the rejection rate of assemblies due to the violation of the functional requirements at a certain limit. The evaluation of the rejection rate is performed by the algorithm mentioned in section **8 .2 .3** . The optimal manufacturing sequence was used to deduce the datum reference frame of the position tolerances of each part, hence this algorithm selects the type of position tolerance. A second algorithm was developed to allocate tolerance values and magnitudes to assembly features to satisfy the assembly functional requirements. These tolerances constrain the surface variation of the parts' features. A new criterion for allocating geometric tolerances has been developed. That is the minimum mismatch probability, which is defined as the probability that assembly parts will be rejected during inspection while being acceptable to the design functional requirements when assembled or vice versa. This criterion ensures that the geometric tolerances are

allocated such that the inspection process will reject parts that will violate the design functional requirements when assembled. The allocation problem was formulated as a combinatorial optimization problem where every feature has six variables associated with it. These are the types of tolerances to control the feature's form and orientation as well as the magnitudes of the size, form, orientation and position tolerance (the type of position tolerance is already determined by the optimal manufacturing sequence) and the mismatch probability is the objective function. In both algorithms, genetic algorithms were used as function optimizers since they were proven to outperform methods like the branch–and–bound and Ballas zero–one in combinatorial optimization.

8 .2 .4 .1 Concluding Remarks

Several researchers (Farmer and Gladman, 1986, Weill, 1988 and Voelker, 1993) noted that geometric tolerances should be allocated to satisfy the functional requirements. In current geometric tolerancing practice a proper tolerancing of a part would not cause a good manufactured part (one which will not violate the functional requirements when assembled to other parts) to be rejected during inspection, and would not cause a bad manufactured part (one which will violate the functional requirements when assembled to other parts) to be accepted. This criterion, although mentioned in several works was never formalized. At the same time it is very hard to obtain a cost tolerance–value relation with each type of tolerance on the same feature. The presented algorithms introduces for the first time the concept of choosing tolerance types as well as tolerance magnitudes in assemblies. An improper choice of tolerance types can increase the rejection rate of good parts.

8 .2 .4 .2 Future Research

In tolerance synthesis the number of independent variables should be kept as low as possible to ensure an efficient optimization. Methods of experimental design can be used to

predict which surface variations have more effect on the assembly functional requirements. Hence only the relevant variations are taken into consideration.

8 .2 .5 Overall Review

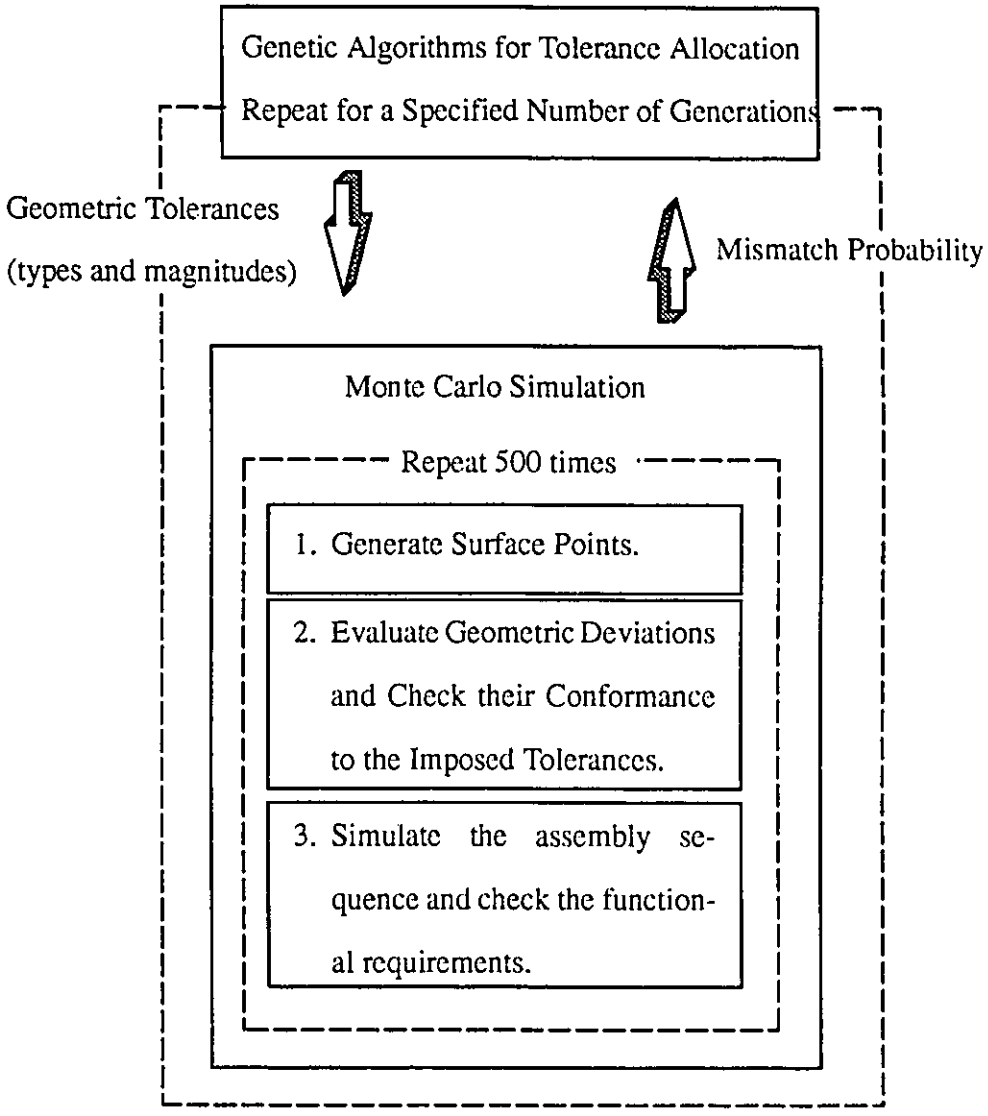


Figure 8 .1 The Developed Tolerance Allocation Algorithm

Figure 8 .1 shows the overall structure of the tolerance allocation algorithm. The algorithm does not ignore the insight of an experienced tolerance practitioner in choosing

tolerance types, but rather directs his effort to suggesting candidate tolerance types for each feature then selects the optimal set of tolerances from these candidates. The algorithm's methodology can be incorporated in a computer–aided tolerancing software to aid designers in selecting geometric tolerances. Although the time required for the program execution is long, due to the use of Monte Carlo simulation within the objective function, this does not pose a serious problem, since tolerance selection is conducted once during the design process.

REFERENCES

American Society of Mechanical Engineers, 1982, New York, N. Y., "*Dimensioning and Tolerancing*", ANSI Y14.5M–1982.

American Society of Mechanical Engineers, 1995, New York, N. Y., "*Mathematical Definition of Dimensioning and Tolerancing Principles*", ANSI Y14.5.1M–1994.

Ayyub, A. M. and Lai, K.L., 1991, "Selective Sampling in Simulation–Based Reliability Assessment", *International Journal of Pressure Vessels & Piping*, Vol. 46., pp. 229–249.

Ayyub, A. M. and Chia, C.Y., 1992, "Generalized Conditional Expectation for Structural Reliability Assessment", *Structural Safety*, Vol. 11, pp. 131–146.

Balling, R. J., Free, J.C. and Parkinson, 1986, A. R., "Consideration of Worst–Case Manufacturing Tolerances in Design Optimization", *Transactions of the ASME, Journal of Mechanisms, Transmissions and Automation in Design*, Vol. 108, pp. 438–441.

Bandler, J., 1976, "Optimization of Design Tolerances Using Nonlinear Programming", *Journal of Optimization Theory and Applications*, Vol. 14, No. 1, pp. 99–114.

Bernstein, N. S. and Preiss, K., 1989, "Representation of Tolerance Information in Solid Models", *Advances in Design Automation–1989*, Vol. 1: *Computer Aided and Computational Design, ASME*, pp. 37–48.

Bjorke, O., 1978, *Computer Aided Tolerancing*, Tapir Press, Norway.

Cardew–Hall, M. J., Labans, T., West, G. and Dench, P., 1993, "A Method of Representing Dimensions and Tolerances on Solid Based Freeform Surfaces", *Robotics and Computer Integrated Manufacturing*, Vol. 10, No. 3, pp. 223–234.

Chase, K.W. and Greenwood, W.H., 1988, "Design Issues in Mechanical Tolerance Analysis", *Manufacturing Review, ASME,* Vol. 1, No. 1, pp. 50–59.

Chase, K.W., Greenwood, W.H., Loosli, B.G. and Hauglund, L.F., 1990, "Least Cost Tolerance Allocation for Mechanical Assemblies with Automated Process Selection", *Manufacturing Review, ASME,* Vol. 3, No. 1, pp. 49–59.

Cogun, C., 1990, "A Correlation Between Deviations in Circularity, Cylindricity, Roughness and Size Tolerances", *International Journal of Machine Tools and Manufacture*, Vol. 30, No. 4, pp 561–567.

de Boor, C., 1978, "A Practical Guide to Splines", *Springer–Verlag.*

Desrochers, A. and Clement, A., 1994, "A Dimensioning and Tolerancing Assistance Model for CAD/CAM", *International Journal of Advanced Manufacturing Technology,* Vol. 9, pp. 352–361.

Dhanish, P.B. and Shunmugam, M.S., 1991, "An Algorithm for Form Error Evaluation – Using the Theory of Discrete and Linear Chebyshev Approximation", *Computer Methods in Applied Mechanics and Engineering*, Vol. 92, pp. 309–324.

Dieter, G. E., 1983, *Engineering Design: A Materials and Processing Approach,* McGraw–Hill.

Dong, Z. and Soom, A., 1990, "Automatic Optimal Tolerance Design for Related Dimension Chains", *Manufacturing Review, ASME,* Vol. 3, No. 4, pp. 262–271.

Dudley, D., 1991, 1991, "*Gear Handbook*", McGraw–Hill, New York.

ElMaraghy, H.A., Wu, Z. and ElMaraghy, W.H., 1991, "A Tolerancing System for Mechanical Design", *Proc. ASME Conf. on Int. Design and Manuf. for Prototyping, ASME,* Atlanta, pp. 113–179.

ElMaraghy, W.H., ElMaraghy, H.A. and Wu, Z., 1990, "Determination of Actual Geometric Deviations Using Coordinate Measuring Machine Data", *Manufacturing Review, ASME*, Vol. 3, No. 1, pp. 32–39.

Etesami, F., 1993, "A Mathematical Model for Geometric Tolerances", *Transactions of the ASME, Journal of Mechanical Design*, Vol. 115, pp. 81–86.

Evans, D.H., 1974, "Statistical Tolerancing. The State of the Art, Part I, Background", *Journal of Quality Technology*, Vol. 6, No. 4, pp. 188–195.

Evans, D.H., 1975, "Statistical Tolerancing. The State of the Art, Part II, Methods of Estimating Moments", *Journal of Quality Technology*, Vol. 7, No. 1, pp. 1–12.

Fainguelernt, D., Weill, R. and Bourdet, P., 1986, "Computer Aided Tolerancing and Dimensioning in Process Planning", *Annals of CIRP*, Vol. 35/1, pp. 382–286.

Farmer, L. E. and Gladman, C. A., 1986, "Tolerance Technology Computer–Based Analysis", *Annals of CIRP*, Vol. 35, No. 1, pp 7–10.

Faux, I. and Pratt, M., 1979, "*Computational Geometry for Design and Manufacture*", Ellis–Horwood, Chichester.

Fleming, A., 1988, "Geometric Relationships between Toleranced Features", *Artificial Intelligence*, Vol. 37, pp. 403–412.

Foster, L.W., 1986, "*GEO–METRICS II, The Application of Geometric Tolerancing Techniques*", Addison Wesley.

Gadallah, M. and ElMaraghy, H.A., 1994, "A New Algorithm for Discrete Tolerance Optimization", *Proceedings of the 4th International Conference on Computer Integrated Manufacturing and Automation Technology.*

Goldberg, D. and Kuo, C.H., 1986, "Genetic Algorithms in Pipeline Optimization", *Journal of Computing in Civil Eng.*, Vol. 1, No. 2, pp. 128–141.

Goldberg, D., 1989, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison–Wesley.

Gossard, D. C., Zuffante, R. P. and Sakurai, H., 1988, "Representing Dimensions, Tolerances and Features in MCAE systems", *IEEE Computer Graphics and Applications*, Vol. 8, No. 2, pp. 51–59.

Grefenstette, J.J., 1986, "Optimization of Control Parameters for Genetic Algorithms", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. SMC–16, No. 1, pp. 122–128.

Guilford, J. and Turner, J. U., 1993, "Advanced Tolerance Analysis and Synthesis for Geometric Tolerances", *Proceedings of the 1993 International Forum on Dimensional Tolerancing and Metrology, ASME.* pp. 187–198.

Guilford, J. and Turner, J. U., 1994, "Representational Primitives for Geometric Tolerancing", *Computer Aided Design*, Vol. 29, No. 9, pp. 577–586.

Gupta, S. G. and Turner, J. U., 1993, "Variational Solid Modeling for Tolerance Analysis", *IEEE Computer Graphics and Applications*, Vol. 13, No. 3, pp. 64–74.

Hasofer, A. M. and Lind, N. C., 1974, "Exact and Invariant Second Moment Code Format", *Journal of the Engineering Mechanics Division*, Vol. 100, pp. 111–121.

He, J.R., 1991, "Tolerancing for Manufacturing via Cost Minimization", *International Journal of Machine Tools and Manufacture*, Vol. 31, No. 4, pp. 455–470.

Hocken, R.J., Raja, J. and Babu,U., 1993, "Sampling Issues in Coordinate Metrology", *Proceedings of the 1993 International Forum on Dimensional Tolerancing and Metrology, ASME.* pp. 97–111.

Holland, J.H., 1975, *Adaptation in Natural and Artificial Systems*, University of Michigan Press.

Hopp, T.H., 1993, "Computational Metrology", *Proceedings of the 1993 International Forum on Dimensional Tolerancing and Metrology, ASME.* pp. 207–217.

Iannuzzi, M. and Sandgren, E., 1995, "Tolerance Optimization using Genetic Algorithms: Benchmarking with Manual Analysis", *Proceedings of the 4th CIRP Seminars on Computer Aided Tolerancing, Tokyo, Japan.*

Jackman, J., Deng, J., Ahn H., Way K. and Vardeman, S., 1994, "A Compliance Measure for the Allignment of Cylindrical Part Features", *Institute of Industrial Engineers Transacations*, Vol. 26, No. 1, pp. 2–10.

Jayaraman, R. and Srinivasan, V., 1989a, "Geometric Tolerancing 1: Virtual Boundary Requirements", *IBM Journal of Research and Development*, Vol. 33, No. 2, pp. 90–104.

Jayaraman, R. and Srinivasan, V., 1989b, "Geometric Tolerancing 2: Conditional Tolerances", *IBM Journal of Research and Development*, Vol. 33, No. 2, pp. 105–124.

Ji, P., 1993, "A Linear Programming Model for Tolerance Assignment in a Tolerance Chart", *International Journal of Production Research*, Vol. 31, No. 3, pp. 739–751.

Ji, P., 1993, "A Tree Appraoch for Tolerance Charting", *International Journal of Production Research*, Vol. 31, No. 5, pp. 1023–1033.

Ji, P., 1994, "An Automatic Tolerance Assignment Approach for Tolerance Charting", *International Journal of Advanced Manufacturing Technology*, Vol. 9, pp. 362–368.

Juster, N. P., 1992, "Modeling and Representation of Dimensions and Tolerances: a Survey", *Computer Aided Design*, Vol. 24, No. 1, pp. 3–17.

Kanai, S., Onozuka, M. and Takahashi, H., 1995, "Optimal Tolerance Synthesis by Genetic Algorithm under the Machining and Assembling Constraints", *Proceedings of the 4th CIRP Seminars on Computer Aided Tolerancing, Tokyo, Japan.*

Lee, E.T.Y., 1989, "Choosing Nodes in Parametric Curve Interpolation", *Computer Aided Design*, Vol. 12, No. 6, pp. 363–370.

Lee, J. and Johnson, G.E., 1993, "Optimal Tolerance Allotment using a Genetic Algorithms and a Truncated Monte Carlo Simulation", *Computer Aided Design*, Vol. 25, No. 9, pp. 601–611.

Lee, Woo–Jong, 1988, *Tolerancing: Computation on Geometric Uncertainties*, Ph.D. dissertation, University of Michigan, Ann arbor.

Light, R. A. and Gossard, D. C., 1982, "Modification of Geometric Models Through Variational Geometry", *Computer Aided Design*, Vol. 14, No. 4, pp. 209–214.

Madsen, H. O., Krenk, S. and Lind, N. C., 1986, "*Methods of Structural Safety*", Prentice–Hall Inc., Englewood Cliffs.

Mansoor, E.M., 1963, "The Application of Probability to Tolerances Used in Engineering Designs", *Proc. Inst. Mech. Eng.*, Vol. 178, No.1, pp.29–51.

Michael, W. and Siddall, J.N., 1981, "The Optimal Tolerance Assignment and Full Acceptance", *Transactions of the ASME, Journal of Mechanical Design*, Vol. 103, pp. 842–848.

Michael, W. and Siddall, J.N., 1982, "The Optimal Tolerance Assignment with Less Than Full Acceptance", *Transactions of the ASME, Journal of Mechanical Design*, Vol. 104, pp. 855–860.

Mortenson, M.E., 1985, "*Geometric Modeling*", John Wiley and Sons.

Murthy, T.S.R. and Abdin, S.Z., 1980, "Minimum Zone Evaluation of Surfaces", *International Journal of Machine Tool Design and Research*, Vol. 20, pp. 123–136.

Nassef, A.O. and ElMaraghy, H.A., 1993, " Allocation of Tolerance Types and Values Using Genetic Algorithms ", *Proceedings of the 3rd CIRP Seminars on Computer Aided Tolerancing, CACHAN, France.*

Nassef, A.O. and ElMaraghy, H.A., 1994, "Evaluation of Actual Geometric Deviations using NURBS Surface Interpolation and Genetic Algorithms", *Proceedings of the 3rd IASTED Computer Applications in Industry Conference*, Cairo, Egypt.

Nassef, A.O. and ElMaraghy, H. A., 1995, "Tolerance Analysis of Assemblies Using Monte Carlo Simulation with Variance Reduction Techniques", *Proc. of the 15th Canadian Congress of Applied Mechanics, CANCAM 95*, Victoria, B.C., Canada.

Nassef, A.O. and ElMaraghy, H. A., 1995, "Probabilistic Analysis of Geometric Tolerances", *Proceedings of the 4th CIRP Seminars on Computer Aided Tolerancing, Tokyo, Japan.*

Nassef, A.O. and ElMaraghy, H. A., 1995, "Statistical Analysis and Optimal Allocation of Geometric Tolerances", *Proceedings of the 15th ASME Conference on Computers in Engineering, Boston, MASS, USA.*

Neumann, A., 1993, "The New ASME Y14.5M standard on Dimensioning and Tolerancing", *Proceedings of the 1993 International Forum on Dimensional Tolerancing and Metrology, ASME.* pp. 7–17.

Niemann, G., 1975, "*Maschinenelemente. 2*", Translated to English, Springer–Verlag, Berlin.

Ostwald, P. F., and Huang, J., 1977, "A Method for Optimal Tolerance Selection", *Transactions of the ASME, Journal of Engineering for Industry*, Vol. 99, pp. 558–565.

Parker, G.R., 1988, *Discrete Optimization*, Academic Press, Boston.

Parkinson, A.R., Balling, R.J. and Free, J.C., 1984, "OPTDES.BYU: A Software System for Optimal Engineering Design", *Computers in Engineering, ASME*, Vol. 1, pp. 429–434.

Parkinson, D.B., 1982, "The Application of Reliability Methods to Tolerancing", *Transactions of the ASME, Journal of Mechanical Design*, Vol. 104, pp. 612–628.

Parkinson, D.B., 1984, "Tolerancing of Component Dimensions in CAD", *Computer Aided Design*, Vol. 16, No.1, pp. 25–32.

Parkinson, D.B., 1985, "Assessment and Optimization of Dimensional Tolerances", *Computer Aided Design*, Vol. 17, No.5, pp. 191–199.

Patel, A. M., 1980, "Computer Aided Assignment of Manufacturing Tolerances", In: *Proceedings of the 17th Design Automation Conference, ASME*, Minneapolis.

Piegl, L., 1987, "A Technique for Smoothing Scattered Data with Conic Sections", *Computers in Industry*, Vol. 9, pp. 223–237.

Piegl, L., 1989, "Modifying the Shape of Rational B–Splines. Part 1: curves", *Computer Aided Design*, Vol. 21, No. 8, pp. 509–518.

Piegl, L., 1989, "Modifying the Shape of Rational B–Splines. Part 2: surfaces", *Computer Aided Design*, Vol. 21, No. 8, pp. 538–546.

Piegl, L., 1991, "On NURBS: A Survey', *IEEE Transactions on Computer Graphics and Applications*, pp. 55–71.

Piegl, L. and Tiller, W., 1987, "Curve and Surface Constructions Using Rational B–Splines", *Computer Aided Design*, Vol. 19, No. 9, pp. 486–498.

Preparata, F. and Shamos, M., 1985, "*Computational Geometry, An Introduction*", Springer–Verlag, New York.

Reeves, C. R., 1993, "Using Genetic Algorithms with Small Population Sizes", *Proceedings of the Fifth International Conference on Genetic Algorithms, Univ. of Illinois, Urbana–Champaign, Ill. USA*, pp. 92–99.

Reklaitis, G. V., Ravindran, A., Ragsdell, K.M., 1983, *Engineering Optimization, Methods and Applications*, J. Wiley.

Requicha, A. A. G., 1983, "Towards a Theory of Geometric Tolerancing", *International Journal of Robotics Research*, Vol. 2, pp. 45–60.

Requicha, A. A. G., 1993, "Mathematical Meaning and Computational Representation of Tolerance Specifications", *Proceedings of the 1993 International Forum on Dimensional Tolerancing and Metrology, ASME*. pp. 61–68.

Requicha, A. A. G. and Chan, S. C., 1986, "Representation of Geometric Features, Tolerances and Attributes in Solid Modelers Based on Constructive Geometry", *IEEE Journal of Robotics and Automation*, Vol. RA–2, pp. 156–166.

Rice, J.R., 1969, "*The Approximation of Functions*", Addison–Wesley Pub. Co., Reading, Mass.

Rivest, L., Dupinet, E., Fortin C. and Morel, C., 1994, "Analysis of Product Tolerances for Process Plan Validation", *Manufacturing Review, ASME*, Vol. 7, No. 4, pp. 312–331.

Rossignac, J. R. and Requicha, A. A. G., 1986, "Offsetting Operations in Solid Modeling", *Computer Aided Geometric Design*, Vol. 3, pp. 129–148.

Roy, U. and Liu, C. R., 1988, "Feature–Based Representational Scheme of a Solid Modeler for Providing Dimensioning and Tolerancing Information", *Robotics and Computer Integrated Manufacturing*, Vol. 4, No. 3/4, pp. 335–345.

Roy, U., Liu, C. R. and Woo, T. C., 1991, "Review of Dimensioning and Tolerancing: Representation and Processing", *Computer Aided Design*, Vol. 23, No. 7, pp. 466–483.

Rubinstein, Y.R., 1981, "*Simulation and the Monte Carlo Method*", John Wiley & Sons Inc., New York.

Shah, J. and Miller, D. W., 1990. "A Structure for Supporting Geometric Tolerances in Product Definition Systems for CIM", *Manufacturing Review, ASME*, Vol. 3, No. 1, pp. 23–31.

Shunmugam, M.S., 1991, "Criteria for Computer–Aided Form Evaluation", *Transactions of the ASME, Journal of Engineering for Industry*, Vol. 113, pp. 233–238.

Siddall, J.N., 1982, *Optimal Engineering Design : Principles and Applications*, Marcell Dekker.

Siddall, J.N., 1983, *Probabilistic Engineering Design: Principles and Applications*, Marcell Dekker.

Speckhart, F. H., 1972, "Calculation of Tolerance Based on a Minimum Cost Approach", *Transactions of the ASME, Journal of Engineering for Industry*, Vol. 94, pp. 447–453.

Spotts, M.F., 1973, "Allocation of Tolerances to Minimize Cost of Assembly", *Transactions of the ASME, Journal of Engineering for Industry*, pp. 762–764.

Srinivasan, V., 1993, "Role of Sweeps in Tolerancing Semantics", *Proceedings of the 1993 International Forum on Dimensional Tolerancing and Metrology, ASME*. pp. 69–78.

Srinivasan, V. and O'Connor, M.A., 1994, "On Interpreting Statistical Tolerancing", *Manufacturing Review, ASME* Vol. 7, No. 4, pp. 304–311.

Suresh, K. and Voelcker, H.B., 1994, "New Challenges in Dimensional Metrology: A Case Study Based on Size", *Manufacturing Review, ASME*, Vo. 7, No. 4, pp. 291–303.

Sutherland, G. H. and Roth, B., 1975, "Mechanism Design: Accounting for Manufacturing Tolerances and Costs in Function Generating Problems", *Transactions of the ASME, Journal of Engineering for Industry, ASME*, Vol. 97, pp. 283–286.

Tang, G.R., Fuh, Y.M. and Kung, R., 1993, "A List Approach to Tolerance Charting", *Computers in Industry*, Vol. 22, pp. 291–302.

Thisted, R.A., 1988, "*Elements of Statistical Computing*", Chapman & Hall, New York.

Turner, G. and Anderson, D., 1988, "An Object-Oriented Approach to Interactive, Feature–Based Design for Quick Turnaround Manufacturing", *Proceedings of the 1988 ASME Computers in Engineering Conference*, San Fransisco, CA.

Turner, J. U., 1991, "Relative Positioning of Parts in Assemblies using Mathematical Programming", *Computer Aided Design*, Vol. , pp. .

Turner, J. U., 1993a, "A Feasibility Space Approach for Automated Tolerancing", *Transactions of the ASME, Journal of Engineering for Industry*, Vol. 115, pp. 341–346.

Turner, J. U., 1993b, "Current Tolerancing Packages", *Proceedings of the 1993 International Forum on Dimensional Tolerancing and Metrology, ASME*. pp. 241–248.

Turner, J. U. and Wozny, M. J., 1988, "A Mathematical Theory of Tolerances", *Geometric Modeling for CAD Applications*, M. Wozny, J. Encarnacao and H. McLaughlin, Eds., North–Holland.

Voelker, H.B., 1993, "A Current Perspective on Tolerancing and Metrology", *Proceedings of the 1993 International Forum on Dimensional Tolerancing and Metrology, ASME.* pp. 49–60.

Walker, R. K. and Srinivasan, V., 1993, "Creation and Evolution of the ASME Y14.5.1M Standard", *Proceedings of the 1993 International Forum on Dimensional Tolerancing and Metrology, ASME.* pp. 19–30.

Weill, R., 1988, "Integrating Dimensioning and Tolerancing in Computer–Aided Process Planning", *Robotics and Computer Integrated Manufacturing*, Vol. 4, No. 1/2, pp. 41–48.

Whitley, D., 1993, "Tutorial 2: Advanced Genetic Algorithm Topics", *ICGA–93, 5th International Conference on Genetic Algorithms, Univ. of Illinois, Urbana–Champaign*, Conference Tutorial.

Wilde, D. and Prentice, E., 1975, "Minimum Exponential Cost Allocation of Sure–Fit Tolerances", *Transactions of the ASME, Journal of Engineering for Industry*, Vol. 97, pp. 1395–1398.

Wirtz, A., 1991, "Vectorial Tolerancing for Production Quality Control and Functional Analysis in Design", *CIRP INternational Working Seminar on Computer Aided Tolerancing*, Penn State Univ., University Park, PA.

Wu, Z., ElMaraghy, W.H. and ElMaraghy, H.A., 1988, "Evaluation of Cost–Tolerance Algorithms for Design Tolerance Analysis and Synthesis", *Manufacturing Review, ASME*, Vol. 1, No. 3, pp. 168–179.

Yu, W., 1992, "Minimum Zone Evaluation of Form Tolerances", *Manufacturing Review, ASME*, Vol. 5, No. 3, pp. 213–220.

Zhang, C. and Wang, H.P., 1992, "Tolerance Analysis and Synthesis for Cam Mechanisms", *International Journal of Production Research*, Vol. 31, No. 5, pp. 1229–1245.

Zhang, C. and Wang, H.P., 1993, "The Discrete Tolerance Optimization Problem", *Manufacturing Review, ASME*, Vol. 6, No. 1, pp. 60–71.

APPENDIX A

FUNCTION OPTIMIZATION WITH GENETIC ALGORITHMS

Genetic algorithms have been used with wide acceptance as a method of global optimization. The algorithms work on a population of values for the independent variables with functions that emulate the biological genetic operators (Holland, 1975). Researchers (Goldberg, 1989) in the area of genetic algorithms tested them on several functions and were shown to outperform other methods in optimizing combinatorial and multimodal functions. The method has been applied in several engineering areas as function optimizers (Goldberg and Kuo, 1986).

A.1 Coding

Consider a function $f(x)$ of the variable x. If genetic algorithms are to be used to find the value of x^* such that :

$$f(x^*) \leq f(x), \quad \forall x \qquad \text{minimization} \tag{A.1}$$

$$f(x^*) \geq f(x), \quad \forall x \qquad \text{maximization} \tag{A.2}$$

the first step is to code the variable x into a binary string, that can be used by the genetic operators. This operation is known in the genetic algorithms literature as the coding of the independent variable. Coding discretizes the possible values of the independent variables. Assume that x ranges from zero to 31, and that each consecutive possible values of x differ by one, i.e.

$$x \in [0, 1, 2, ..., 30, 31] \tag{A.3}$$

A binary string representing x would have five locations, where each location i can assume a value of zero or one and where the value of each location $i \in \{0,1,2,3,4\}$ is multiplied by 2^i

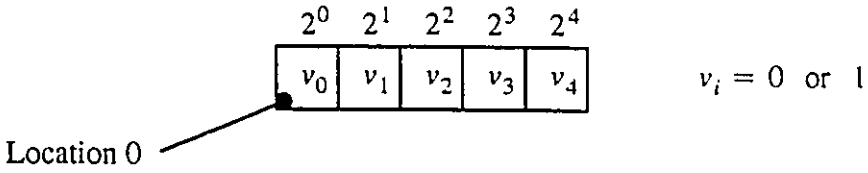


Figure A.1 Genetic Chromosome

It is obvious that if all locations are equal to one the value of the string becomes 31 and if all location are equal to zero the value of the string becomes zero.

A.1 .1 Coding of Continuous Variable

Assume that a variable x is to be coded where:

$$x \in \mathbb{R}, \qquad \mathbb{R} = \text{real values line}. \tag{A.4}$$

and that x ranges from a value a to a value b

$$a \leq x \leq b \tag{A.5}$$

Assume that x is to be discretized to a set of n possible values x_i such that:

$$x_i \in \{x_1, x_2, ..., x_n\} \tag{A.6}$$

$$x_1 = a \tag{A.7}$$

$$x_n = b \tag{A.8}$$

$$x_{i+1} - x_i = \text{DIFF} \tag{A.9}$$

where DIFF is a predetermined difference between x_{i+1} and x_i. The variable x is to represented by a binary string having $n+1$ locations, where each location i has a value:

$$\text{Val}(i) = r2^i \qquad i \in \{0, 1, 2, ..., n\} \tag{A.10}$$

and where the whole string assumes the value

$$\text{Val(string)} = a + \sum_{i=0}^{n} \text{Val}(i) \tag{A.11}$$

The values of r and n are determined as follows:

$$\text{DIFF} = r2^1 - r2^0 \tag{A.12}$$

$$\text{DIFF} = r \tag{A.13}$$

If the value of all locations is equal to one, the value of the string becomes a geometric series:

$$\text{Val(string)} = a + \sum_{i=0}^{n} r2^i \tag{A.14}$$

$$= a + r(2^{n+1} - 1)$$

but,

$$\text{Val(string)} = b \tag{A.15}$$

Let

$$v = b - a \tag{A.16}$$

then

$$n = \log_2\left[\frac{v}{r} + 1\right] \tag{A.17}$$

Since $n \in \mathbb{R}$, and n should be integer, the value:

$$N = \text{integer}(n) \tag{A.18}$$

is used and a new value for r is computed

$$r = \left[\frac{v}{2^{N+1} - 1}\right] \tag{A.19}$$

A.1 .2 Coding of an Integer Variable

If the possible value of the variable x belongs to the integer line:

$$x \in \{0, 1, 2, 3, ..., m\} \tag{A.20}$$

then x is already discretized with a minimum value of zero, a maximum value of m and a difference between every consecutive values equal to one (i.e. DIFF=1). The coding method described in the previous section can be used. This may lead to strings of small length which can lead to an early loss of strings with high fitness once the genetic operators are in action. This problem can be solved by using a smaller value of DIFF and the following function for string evaluation

$$\text{Val(string)} = a + \text{integer}\left[\sum_{i=0}^{m} r2^i\right] \tag{A.21}$$

A.2 Fitness Function

Genetic algorithms are supposed to maximize a function known as the fitness function. Thus any objective function to be optimized by genetic algorithms should be converted into a fitness function form. If the optimization problem at hand is a maximization problem, then the fitness function F equal to the objective function f. If the problem at hand is a minimization problem, the value of the fitness function is equal to some large value V

minus the value of the objective function.

$$F = V - f \tag{A.22}$$

A.2 .1 Fitness Function Scaling

In order to avoid the early loss of high strings, fitness function scaling is a common practice (Goldberg, 1989). Several methods are proposed for scaling the fitness function. The objective of the scaling is to make the chromosomes with maximum fitness have a re-scaled fitness value twice as large as the re-scaled average fitness. Linear scaling is the most widely used method Figure A.2 .

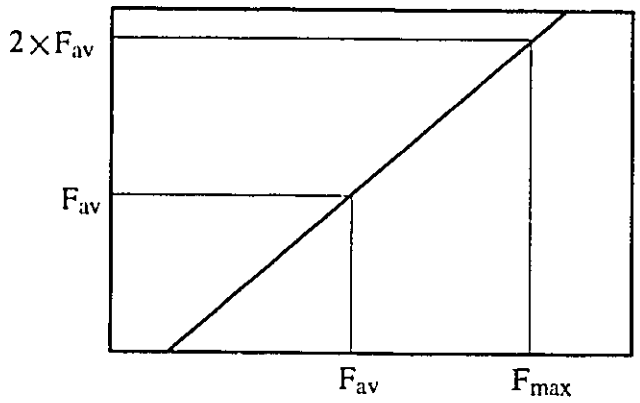


Figure A.2 Fitness Function Scaling

Linear scaling has one disadvantage. It may lead to negative re-scaled values. This problem can be solved by re-scaling the less than average chromosomes such that the minimum fitness chromosome is re-scaled to zero fitness.

A.3 General Procedure of Genetic Algorithms

After a problem is coded into a binary string, a population of random valued strings is formulated and is used for making new generations of strings iteratively. The strings are known in the genetic algorithms literature as "*chromosomes*", and the location of each binary

digit in the chromosomes is known as a *gene*. Assume that genetic algorithms are used to optimize a function of one variable *f(x)*. The general genetic algorithms optimization procedure is:

Procedure: GEN–ALG

A. Code the range of possible values of the variable *x* into a binary string as shown in section **A.1**

B. Construct the fitness function for the objective function as shown in section **A.2** .

1. Generate a population of *N* chromosomes. Each chromosome has a randomly chosen binary digit in each of its genes as shown in Figure A.3 . Each chromosome represents a random instance of the independent variable *x*.

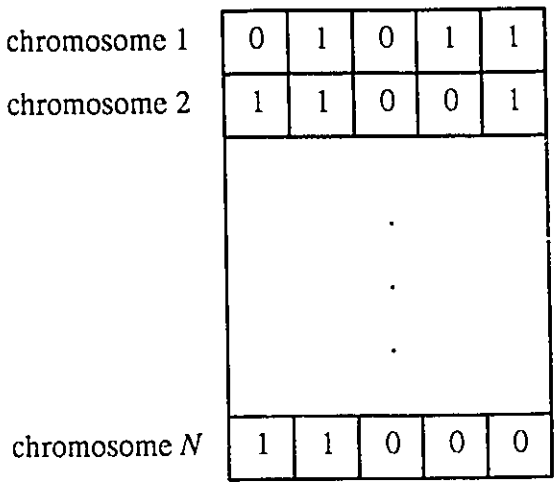


Figure A.3 A Genetic Population

2. Generate a new population of chromosomes using the reproduction operator.

3. Apply the cross–over operator on the new generation.

4. Apply the mutation operator on the new generation.

5. If the generation number is less than the maximum number of generations go to step 2, otherwise stop.

6. Find the chromosome with the largest fitness value in the final generation. Decode the chromosome and deliver its equivalent value of the independent to the optimum value of the objective function.

In the case of a multidimensional optimization, each independent variable is enclosed into a sub–chromosome and put together in a raw to form a larger chromosome as shown in Figure A.4 .

2^0	2^1	2^2	2^3	2^4	2^0	2^1	2^2	2^3	2^4	2^0	2^1	2^2	2^3	2^4
0	1	0	1	1	1	1	1	1	1	0	1	0	0	1

Figure A.4 A chromosome for 3 variables

In such a case the reproduction operator is applied to the larger chromosome, while the mutation and cross–over operators are applied to each sub–chromosome.

A.4 Genetic Operators

The previous section referred to three genetic operators used iteratively to maximize a fitness function. These operators are reproduction, mutation and cross over. The operators emulate the action of equivalent genetic operations in biology.

A.4 .1 Reproduction

Given a population of N chromosomes, the fitness function f_i is evaluated for each chromosome. Each chromosome is then assigned a value:

$$v_i = \frac{f_i}{\sum_{i=1}^{N} f_i} \tag{A.23}$$

forming a discrete probability distribution over the population. The purpose of the reproduction operator is to replace the old population with a new population of chromosomes selected from the old population such that:

$$\overline{f_{\text{new}}} > \overline{f_{\text{old}}} \tag{A.24}$$

where $\bar{f}$ is the average fitness value of the population. The reproduction operator is a method of chromosome selection based on its fitness value. The main method used for selection is the roulette wheel selection (Goldberg, 1989). The selection process proceeds as follows:

1. The interval [0,1] is divided into subintervals I_i whose length is equal to the value v_i of its corresponding equal chromosome.

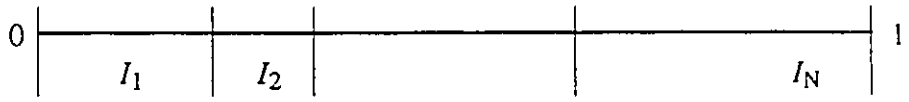


Figure A.5 Roulette Wheel Selection

2. A random number U∈[0,1] is generated and the value of U is checked for the subinterval within which it lies. The chromosome corresponding to the subinterval is delivered as the selected chromosome.

The selection process is repeated N times for the whole population.

A.4 .2 Cross Over

After the old population is replaced by a new one, an even number of chromosomes are selected from the new population to undergo cross–over. The number of chromosomes selected for cross–over is determined by the cross–over probability P_c. Every successive pair of chromosomes in the population is picked up and a random number $U \in [0,1]$ is generated. If U is less than P_c, the pair of chromosomes undergo cross–over. The cross–over operator acts as follows:

1. A location l in the pair of chromosomes is picked at random

$$l \in \{1, 2, 3, ..., L\} \tag{A.25}$$

where L is the number of genes in the chromosomes.

2. All binary bits in the genes right of the position l are swapped between the pair of chromosomes under cross–over (Figure A.6).

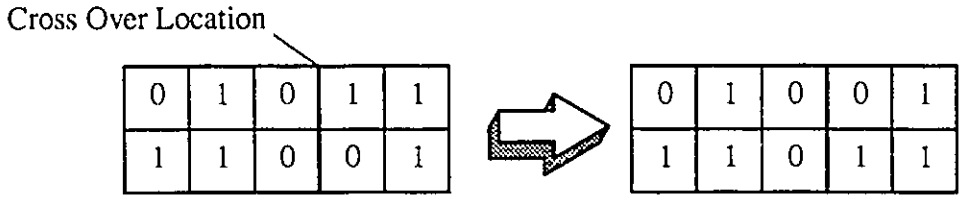


Figure A.6 Cross Over

A.4 .3 Mutation

A number of chromosomes is picked from the new population to undergo mutation based on a mutation probability P_m. For every chromosome in the population a random number $U \in [0,1]$ is generated. If U is less than P_m, the chromosome undergoes mutation. The mutation operator is applied as follows:

1. A random position l is picked where:

$$l \in \{1, 2, 3, ..., L\} \tag{A.26}$$

2. The binary value at the gene positioned at l is flipped.

A.5 Schema Theorem

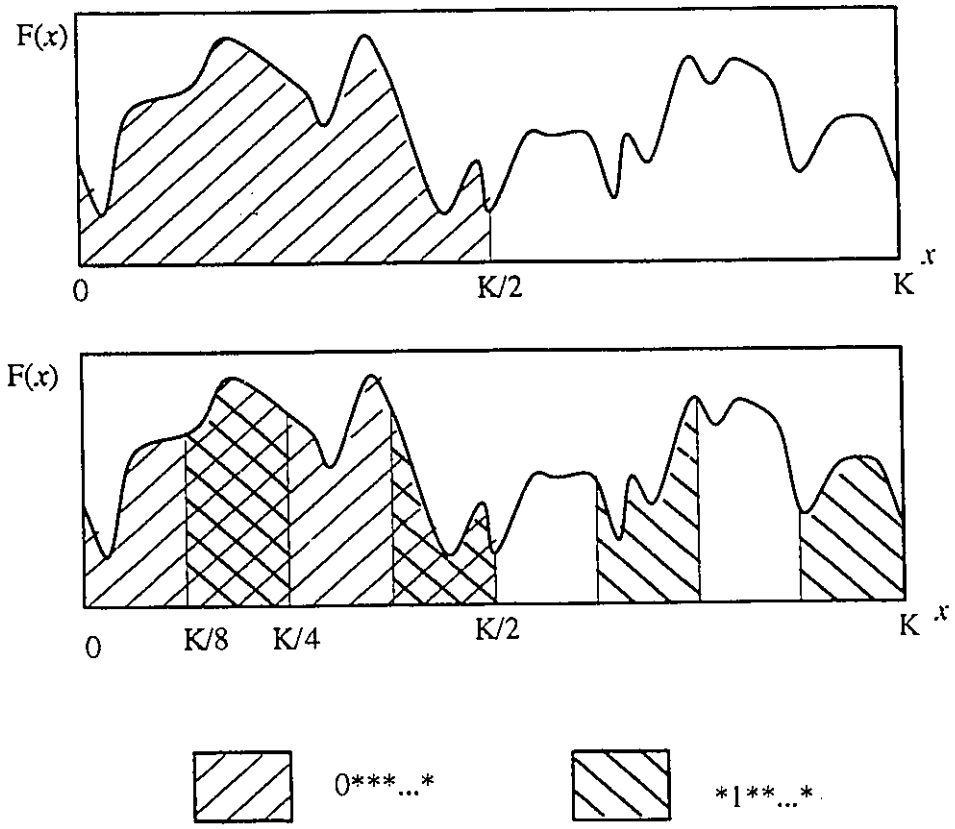


Figure A.7 A Function with Schemata

How genetic algorithms arrive at the global optimum of the functions they optimize is understood in view of what is known as a chromosome whose genes can have one of the following three values: "0", "1", or "*", where "*" is a wild–card value that can have "0"

or "1" in its place. Figure A.7 shows a function F(x) of one variable x which ranges from zero to an arbitrary value K. The schema 0***...* represent all values of x less than K/2 and more than zero, while the schema *1**...* represents all values of x between iK/8 and (i+1)K/8, $i \in \{1,3,5,7\}$. Figure A.7 shows that the schema 0***...* has an average fitness value larger than the average fitness value of the schema *1**...*. In a population of chromosomes, each chromosome has l genes, each is a member of 2^l schemata. Let $M(H,t)$ be the number of chromosomes representing a schema H at time t+1 under the effect of reproduction is equal to:

$$M(H, t+1) = M(H, t)\frac{f(H)}{\bar{f}} \tag{A.27}$$

where $f(H)$ is the average fitness of the chromosomes representing H and $\bar{f}$ is the average population fitness. Equation (A.27) shows that a schema with a high average fitness would have more representatives in a population than a schemata with a lower average fitness. If reproduction is solely used, only replications of the original population will be selected. Cross–over is responsible for introducing new chromosomes representing new schemata in the search. The number of chromosomes representing a schema H at time t+1 is thus equal to

$$M(H, t+1) = M(H, t)\frac{f(H)}{\bar{f}} p_s \tag{A.28}$$

where p_s is the probability that H survives to time t+1. Holland (1975) shows that p_s is given by the relation

$$p_s \geq 1 - p_c\frac{\delta(H)}{l-1} \tag{A.29}$$

where p_c is the cross–over probability and $\delta(H)$ is the defining length of the schema H given by the number of genes between the every consecutive non *–valued genes. Holland (1975) showed that using a combination of reproduction and cross–over, schemata with above average fitness and small defining length will be tried at an exponential rate.

A.6 Advanced Operators

Variations of the standard genetic algorithms operators have been investigated in previous research (Whitley, 1993). Two of these variations are discussed in the following sections.

A.6 .1 Tournament Selection

Some researchers proposed alternative schemes for the selection of chromosomes from population,other than the roulette wheel selection. One of these schemes that proved to be less biased is the tournament selection. Tournament selection work as follows (Goldberg, 1989): Draw two chromosomes from the old generation using tournament selection, then accept the one with the higher fitness as the winner and insert it in the new population. The process is continued until the population is full.

A.6 .2 Uniform Cross–Over

Uniform cross–over is a generalized version of the one point cross–over discussed in the previous sections. Uniform cross–over works as follows: for each gene in the two chromosomes undergoing cross–over, randomly decide if the values of the genes are to be swapped or not. Uniform cross–over is unbiased operator. At the same time uniform cross–over is more disruptive in processing schemata than one point cross–over. Researchers in the area of genetic algorithms (Whitley, 1993) suggested that uniform cross–over is best used with small population sizes.

APPENDIX B

GEOMETRIC TRANSFORMATIONS AND RELATIONSHIPS

In this appendix methods of translating and rotating bodies in space as well as the construction of the transformation matrices are described.

B.1 Translation

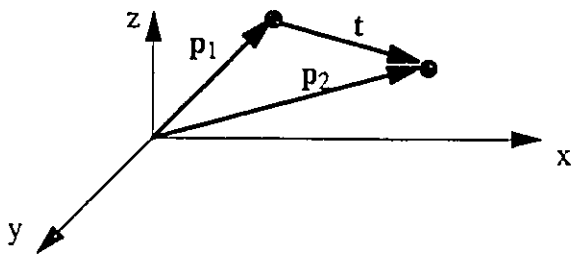


Figure B.1 Point Translation

Figure B.1 shows the translation of a point from one set of coordinates to another. The translation is given by the following equation:

$$\mathbf{p}_2 = \mathbf{p}_1 + \mathbf{t} \tag{B.1}$$

where

$$\mathbf{t} = [(x_2 - x_1)\ (y_2 - y_1)\ (z_2 - z_1)] \tag{B.2}$$

B.2 Rotation

Figure B.2 shows point rotation in a x–y plane around the z–axis. Point $\mathbf{p}_1$ is to be rotated to point to the coordinates at point $\mathbf{p}_2$. The coordinates of $\mathbf{p}_2$ is given by:

$$x_2 = |p_2| \cos(\alpha + \theta) \tag{B.3}$$

$$y_2 = |p_2| \sin(\alpha + \theta) \tag{B.4}$$

From basic trigonometry:

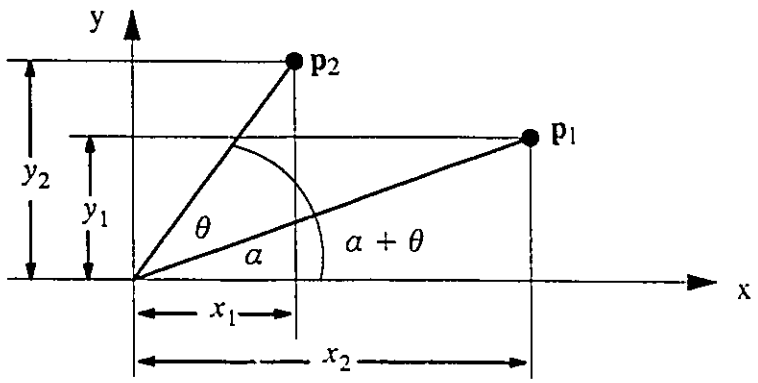


Figure B.2 Point Rotation about z Axis

$$\cos(\alpha + \theta) = \cos\alpha \cos\theta - \sin\alpha \sin\theta \tag{B.5}$$

$$\sin(\alpha + \theta) = \sin\alpha \cos\theta + \cos\alpha \sin\theta \tag{B.6}$$

$$\cos\alpha = \frac{x_1}{|p_1|} \tag{B.7}$$

$$\sin\alpha = \frac{y_1}{|p_1|} \tag{B.8}$$

using some simple substitution the following formulas are obtained:

$$\cos(\alpha + \theta) = \left(\frac{x_1}{|p_1|}\right)\cos\theta - \left(\frac{y_1}{|p_1|}\right)\sin\theta \tag{B.9}$$

$$\sin(\alpha + \theta) = \left(\frac{x_1}{|p_1|}\right)\sin\theta + \left(\frac{y_1}{|p_1|}\right)\cos\theta \tag{B.10}$$

hence

$$x_2 = x_1 \cos\theta - y_1 \sin\theta \tag{B.11}$$

$$y_2 = x_1 \sin\theta + y_1 \cos\theta \tag{B.12}$$

The rotation matrix for rotation around the z–axis is:

$$\mathbf{T}_z = \begin{bmatrix} \cos\theta_z & \sin\theta_z & 0 \\ -\sin\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{B.13}$$

Rotations around the y–axis and the x–axis can be similarly derived:

$$\mathbf{T}_y = \begin{bmatrix} \cos\theta_y & 0 & -\sin\theta_y \\ 0 & 1 & 0 \\ \sin\theta_y & 0 & \cos\theta_y \end{bmatrix} \tag{B.14}$$

$$\mathbf{T}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & \sin\theta_x \\ 0 & -\sin\theta_x & \cos\theta_x \end{bmatrix} \tag{B.15}$$

Thus the rotation matrix around the axes x,y and z is given by:

$$\mathbf{T}_{rot} = \mathbf{T}_z\mathbf{T}_y\mathbf{T}_x \tag{B.16}$$

B.3 Transformation Matrix

The position vector **p** can be represented in the 3–dimensional space by the vector:

$$\mathbf{p} = [x \;\; y \;\; z \;\; 1] \tag{B.17}$$

Using the above form, the transformation to another position vector **q** can be established

using the 4×4 transformation matrix T.

$$\mathbf{q} = \mathbf{pT} \tag{B.18}$$

$$\mathbf{T} = \begin{bmatrix} T_{rot_{1,1}} & T_{rot_{2,1}} & T_{rot_{3,1}} & 0 \\ T_{rot_{1,2}} & T_{rot_{2,2}} & T_{rot_{3,2}} & 0 \\ T_{rot_{1,3}} & T_{rot_{2,3}} & T_{rot_{3,3}} & 0 \\ t_x & t_y & t_z & 1 \end{bmatrix} \tag{B.19}$$

B.4 Transformation Matrix Construction

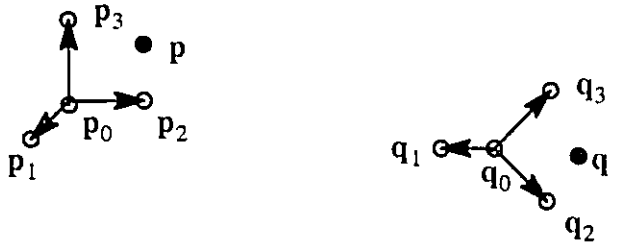


Figure B.3 Three Point Transformation

Figure B.3 shows a reference frame represented by two unit vectors $\overline{p_0p_1}$, $\overline{p_0p_2}$ & $\overline{p_0p_3}$. The frame is to be transformed into the new reference coordinates $\overline{q_0q_1}$, $\overline{q_0q_2}$ & $\overline{q_0q_3}$. The transformation matrix elements can be derived using procedure **P1** (Mortenson, 1985). Although the points shown in Figure B.3 make a right hand orthogonal frame of axes, the procedure is described in general for the transformation of any set of 3 points

Procedure P1:

1. Let

$$\mathbf{V}_1 = \mathbf{p}_2 - \mathbf{p}_1 \tag{B.20}$$

$$\mathbf{W}_1 = \mathbf{q}_2 - \mathbf{q}_1 \tag{B.21}$$

$$\mathbf{V}_3 = \mathbf{V}_1 \times (\mathbf{p}_3 - \mathbf{p}_1) \tag{B.22}$$

$$\mathbf{W}_3 = \mathbf{W}_1(\mathbf{q}_3 - \mathbf{q}_1) \tag{B.23}$$

$$\mathbf{V}_2 = \mathbf{V}_3 \times \mathbf{V}_1 \tag{B.24}$$

$$\mathbf{W}_2 = \mathbf{W}_3 \times \mathbf{W}_1 \tag{B.25}$$

The vectors $\mathbf{V}_1$, $\mathbf{V}_2$ and $\mathbf{V}_3$ form a right–hand orthogonal frame. The same argument applies for the set of vectors $\mathbf{W}_1$, $\mathbf{W}_2$ and $\mathbf{W}_3$.

2. Evaluate the unit vectors:

$$\mathbf{v}_1 = \frac{\mathbf{V}_1}{|\mathbf{V}_1|} \tag{B.26}$$

$$\mathbf{v}_2 = \frac{\mathbf{V}_2}{|\mathbf{V}_2|} \tag{B.27}$$

$$\mathbf{v}_3 = \frac{\mathbf{V}_3}{|\mathbf{V}_3|} \tag{B.28}$$

$$\mathbf{w}_1 = \frac{\mathbf{W}_1}{|\mathbf{W}_1|} \tag{B.29}$$

$$\mathbf{w}_2 = \frac{\mathbf{W}_2}{|\mathbf{W}_2|} \tag{B.30}$$

$$\mathbf{w}_3 = \frac{\mathbf{W}_3}{|\mathbf{W}_3|} \tag{B.31}$$

3. Let,

$$[\mathbf{v}] = [\mathbf{v}_1 \ \ \mathbf{v}_2 \ \ \mathbf{v}_3] \tag{B.32}$$

$$[\mathbf{w}] = [\mathbf{w}_1 \ \ \mathbf{w}_2 \ \ \mathbf{w}_3] \tag{B.33}$$

4. The rotation matrix $\mathbf{T}_{rot}$ is equal to:

$$\mathbf{T}_{rot} = [\mathbf{v}]^{-1}[\mathbf{w}] \tag{B.34}$$

5. The translation vector $\mathbf{t}$ is equal to:

$$\mathbf{t} = \mathbf{q}_1 - \mathbf{p}_1[\mathbf{v}]^{-1}[\mathbf{w}] \tag{B.35}$$

The translation vector as well as the rotation matrix can be substituted into equation (B.19) to obtain the transformation matrix **T**. The derived transformation places point $\mathbf{p}_1$ at the coordinates of $\mathbf{q}_1$, coincides the direction of the vector $\overline{\mathbf{p}_1\,\mathbf{p}_2}$ with the direction of the vector $\overline{\mathbf{q}_1\,\mathbf{q}_2}$ and coincides the plane formed by the points $\mathbf{p}_1$, $\mathbf{p}_2$ & $\mathbf{p}_3$ with the plane formed by the points $\mathbf{q}_1$, $\mathbf{q}_2$ & $\mathbf{q}_3$. Thus a point $\mathbf{p}$ in the **v** system is transformed into the point $\mathbf{q}$ in the **w** system.

B.5 Normal Distance Between a Point and a Line

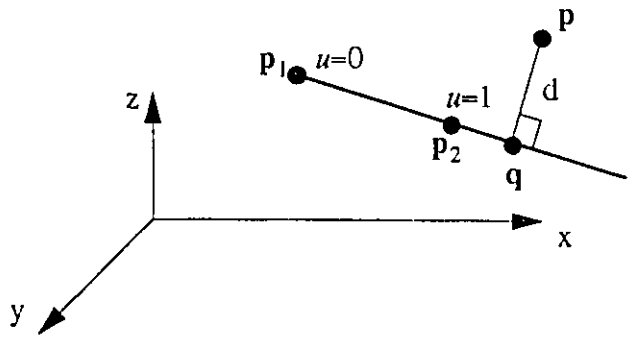


Figure B.4 Normal Distance between a Point and a Line

The minimum distance between a point and a line in space is equal to the perpendicular distance between the point and the line. Figure B.4 shows a line represented

two given points $\mathbf{p}_1$ and $\mathbf{p}_2$. The line parametric equation is given by:

$$\mathbf{p}(u) = \mathbf{p}_1 + u(\mathbf{p}_2 - \mathbf{p}_1) = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} + u\left(\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} - \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}\right) \tag{B.36}$$

The distance between $\mathbf{p}$ and the line is given by:

$$d = \|\mathbf{p} - \mathbf{q}\| = \left\| \begin{bmatrix} x - x_q \\ y - y_q \\ z - z_q \end{bmatrix} \right\| \tag{B.37}$$

The point $\mathbf{q}$ on the line is such that the vectors $\overline{\mathbf{p}_1\,\mathbf{p}_2}$ and $\overline{\mathbf{p}\,\mathbf{q}}$ is equal to zero.

$$(x_2 - x_1)(x - x_q) + (y_2 - y_1)(y - y_q) + (z_2 - z_1)(z - z_q) = 0 \tag{B.38}$$

Since

$$\mathbf{q} = \mathbf{p}_1 + u(\mathbf{p}_2 - \mathbf{p}_1) \tag{B.39}$$

where u is the value of the parameter at the point $\mathbf{q}$, equation (B.39) can be substituted into equation (B.38), which results in

$$u = \frac{(x_2 - x_1)(x - x_1) + (y_2 - y_1)(y - y_1) + (z_2 - z_1)(z - z_1)}{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \tag{B.40}$$

Using Matlab, the value of u is found from equation (B.39) and consequently the minimum distance is found from equation (B.37).

B.6 Normal Distance Between a Point and a Plane

Consider a point $\mathbf{p}$ and a plane represented by three points $\mathbf{p}_1$, $\mathbf{p}_2$ and $\mathbf{p}_3$. Using the procedure described in section B.5, a point $\mathbf{q}$ is found on the vector $\overline{\mathbf{p}_2\,\mathbf{p}_3}$, such that the vectors $\overline{\mathbf{p}\,\mathbf{q}}$ and $\overline{\mathbf{p}_2\,\mathbf{p}_3}$ are perpendicular.

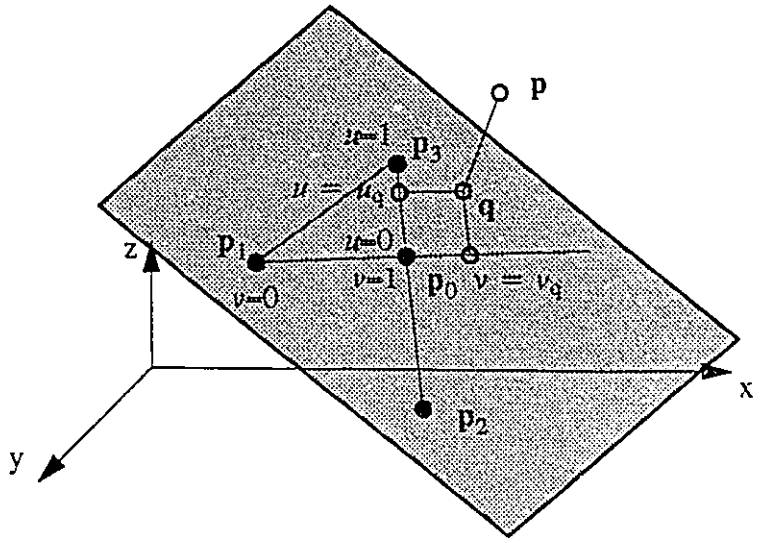


Figure B.5 Minimum distance between a point and a plane

The points $\mathbf{p}_1$, $\mathbf{p}_0$ and $\mathbf{p}_3$ are used to construct the following parametric equation of the plane:

$$\mathbf{p}(u, v) = \mathbf{p}_1 + u(\mathbf{p}_0 - \mathbf{p}_1) + v(\mathbf{p}_3 - \mathbf{p}_0) \tag{B.41}$$

Point $\mathbf{q}$ can be found by projecting $\mathbf{p}$ on the vector $\overline{\mathbf{p}_2\,\mathbf{p}_3}$ to obtain u_q and on the vector $\overline{\mathbf{p}_1\,\mathbf{p}_0}$ to obtain the v_q and substituting into equation (B.41). The minimum distance between $\mathbf{p}$ and the plane is equal to the magnitude of the vector $\overline{\mathbf{p}\,\mathbf{q}}$.

APPENDIX C

INTEGRATION WITH MONTE CARLO SIMULATION

Monte Carlo simulation is used to find integrals of joint probability distributions. The following methods describe the theoretical basis of Monte Carlo integration.

C.1 Sample Mean Monte Carlo Method

Consider the integral

$$I = \int_{\mathbf{x}\in\mathbf{D}_f} f(\mathbf{x})\ d\mathbf{x} \tag{C.1}$$

where $\mathbf{x}$ is an independent random vector, $f(\mathbf{x})$ is the joint probability distribution of $\mathbf{x}$ and $\mathbf{D}_f$ is the set of feasible values of $\mathbf{x}$. The integral can be rewritten in the form:

$$I = \int g(\mathbf{x}) \tag{C.2}$$

where,

$$g(\mathbf{x}) = \begin{cases} f(\mathbf{x}) & \mathbf{x} \in \mathbf{D}_f \\ 0 & \text{otherwise} \end{cases} \tag{C.3}$$

The above form can be converted to a form calculating the expected value of a function of a random variable $\mathbf{x}$.

$$I = \int \frac{g(\mathbf{x})}{f(\mathbf{x})}\ f(\mathbf{x})\ d\mathbf{x} \tag{C.4}$$

$$\text{i.e.} \quad I = \mathrm{E}\left[\frac{g(\mathbf{x})}{f(\mathbf{x})}\right] \tag{C.5}$$

Knowing that,

$$g(\mathbf{x}) = f(\mathbf{x}) \qquad \text{for } \mathbf{x} \in \mathbf{D}_f \tag{C.6}$$

then,

$$I = \mathrm{E}[B(\mathbf{x})] \tag{C.7}$$

where:

$$B(\mathbf{x}) = \begin{cases} 1 & \mathbf{x} \in \mathbf{D}_f \\ 0 & \text{otherwise} \end{cases} \tag{C.8}$$

An unbiased estimate of I is:

$$\theta_1 = \frac{1}{\mathrm{N}} \sum_{i=1}^{\mathrm{N}} B(\mathbf{X}_i) \tag{C.9}$$

where N is a large number and $\mathbf{X}_i$ is the ith random generation of the random vector $\mathbf{x}$. The following procedure describes the method by which θ_1 is evaluated:

1. Generate a sequence of random vectors

$$\mathbf{X}_i \qquad i \in \{1, 2, ..., \mathrm{N}\} \tag{C.10}$$

from the probability distribution $f(\mathbf{x})$

2 Compute

$$B(\mathbf{X}_i) = \begin{cases} 1 & \mathbf{x} \in \mathbf{D}_f \\ 0 & \text{otherwise} \end{cases} \tag{C.11}$$

3 compute θ_1 using equation (C.9).

C.2 Efficiency of Monte Carlo Method

If two Monte Carlo methods are used to evaluate the integral I, using the estimators θ_1 and θ_2, then the first method is more efficient than the second if:

$$\varepsilon = \frac{t_1 \ \mathrm{Var}(\theta_1)}{t_2 \ \mathrm{Var}(\theta_2)} < 1 \tag{C.12}$$

where t_1 and t_2 are the units of computing time for evaluating θ_1 and θ_2. If t_1 and t_2 are equal then $\mathrm{Var}(\theta_1) < \mathrm{Var}(\theta_2)$, thus the less the variance of the estimators, the more efficient is the simulation.

C.3 Stratified Sampling

Let **D** be the domain of the random vector **x** such that:

$$\mathbf{D}_f \subset \mathbf{D} \tag{C.13}$$

D can be divided into m subregions,

$$\mathbf{D} = \bigcup_{i=1}^{m} \mathbf{D}_i \quad , \ \mathbf{D}_k \bigcap \mathbf{D}_l = \emptyset \quad , \ k \neq l \tag{C.14}$$

then,

$$I = \int g(\mathbf{x}) \ d\mathbf{x} = \sum_{i=1}^{m} \int_{\mathbf{x} \in (\mathbf{D}_f \bigcap \mathbf{D}_i)} g(\mathbf{x}) \ d\mathbf{x} \tag{C.15}$$

Let,

$$I_i = \int_{\mathbf{x} \in (\mathbf{D}_f \bigcap \mathbf{D}_i)} g(\mathbf{x}) \ d\mathbf{x} \tag{C.16}$$

then,

$$I_i = \int_{\mathbf{x} \in (\mathbf{D}_f \bigcap \mathbf{D}_i)} \frac{g(\mathbf{x})}{f(\mathbf{x})} \; f(\mathbf{x}) \; d\mathbf{x} \tag{C.17}$$

$$I_i = \int B_i(\mathbf{x}) \; f(\mathbf{x}) \; d\mathbf{x} \tag{C.18}$$

where:

$$B(\mathbf{x}) = \begin{cases} 1 & \mathbf{x} \in (\mathbf{D}_f \bigcap \mathbf{D}_i) \\ 0 & \text{otherwise} \end{cases} \tag{C.19}$$

$$I_i = \mathrm{E}\left[B_i(\mathbf{x})\right] \tag{C.20}$$

An estimator of I_i is:

$$I'_i = \frac{1}{\mathrm{N}} \sum_{j=1}^{\mathrm{N}} B_i(\mathbf{X}_j) \tag{C.21}$$

Since

$$I = \sum_{i=1}^{m} {}'_i \tag{C.22}$$

An estimator of I would be:

$$\theta_2 = \sum_{i=1}^{m} \frac{1}{\mathrm{N}} \sum_{j=1}^{\mathrm{N}} B_i(\mathbf{X}_j) \tag{C.23}$$

Rubinstein (1981) shows that the variance of the stratified sampling method is more efficient

than the sample mean method with a factor

$$\varepsilon = \frac{1}{m^2} \tag{C.24}$$

C.4 Antithetic Variates

Consider two random variables V1 and V2, the variance of (V1+V2)/2 is equal to:

$$\mathrm{Var}(\tfrac{1}{2}(\mathrm{V1} + \mathrm{V2})) = \tfrac{1}{4}\mathrm{Var}(\mathrm{V1}) + \tfrac{1}{4}\mathrm{Var}(\mathrm{V2}) + \tfrac{1}{2}\mathrm{Cov}(\mathrm{V1}, \mathrm{V2}) \tag{C.25}$$

if the last term in the above equation is negative, then the variance of the average of V1 and V2 follows the following equation:

$$\mathrm{Var}(\tfrac{1}{2}(\mathrm{V1} + \mathrm{V2})) \leq \frac{\mathrm{Var}(\mathrm{V1}) + \mathrm{Var}(\mathrm{V2})}{2} \tag{C.26}$$

Consider the integral,

$$I = \int_{\mathbf{x} \in \mathbf{D}_f} g(\mathbf{x})\ d\mathbf{x} \tag{C.27}$$

which is evaluated with the sample mean Monte Carlo estimator:

$$\theta_1 = \frac{1}{\mathrm{N}} \sum_{i=1}^{\mathrm{N}} B(\mathbf{X}_i) \tag{C.28}$$

where

$$B(\mathbf{X}_i) = \begin{cases} 1 & \mathbf{X}_i \in \mathbf{D}_f \\ 0 & \text{otherwise} \end{cases} \tag{C.29}$$

and $\mathbf{X}_i$ is drawn from the distribution $f(\mathbf{x})$

$$\mathbf{X}_i = \mathrm{F}^{-1}(\mathrm{U}_i) \qquad , \ \mathrm{U}_i \in [0, 1] \tag{C.30}$$

Another estimators of the same integral is equal to

$$\theta_3 = \frac{1}{2N}\sum_{i=1}^{N}\left[B(\mathbf{X}_i) + B(\mathbf{X'}_i)\right] \tag{C.31}$$

where

$$\mathbf{X'}_i = F^{-1}(1 - U_i) \qquad , \quad U_i \in [0, 1] \tag{C.32}$$

Since $\mathbf{X}_i$ has a negative correlation with $\mathbf{X}_i$, then:

$$\mathrm{Var}(\theta_3) \leq \frac{1}{2}\mathrm{Var}(\theta_1) \tag{C.33}$$