Using C-Spaces for Tolerance Synthesis

R. Ramaswamy*

CMU-RI-TR-89-26

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania

October 1989

A thesis submitted to the Department of Mechanical Engineering in
partial fulfillment of the requirements for the degree of Master of Engineering.
Supervised by D.A. Bourne and D. Navinchandra, The Robotics Institute.

*Supported during this work by the Machinist Expert Consortium at The Robotics Institute, Carnegie Mellon University.

Table of Contents

List of Figures

List of Tables

Abstract

The objective of design is to create a specification for functionally acceptable parts. Tolerances must be assigned to the design dimensions and play an important role in deciding if the design really exhibits the desired behavior when it is manufactured. It is therefore important that tolerance allocation is based on an understanding of the semantics of tolerance rather than adherence to convention. In this paper we introduce a means of explicitly reasoning about the effect of tolerances on the kinematic behavior of a part. The method is based on the use of configuration spaces to represent kinematic constraints. We show how algorithms to automatically generate and reason about configuration spaces can form the basis of a useful computer-aided design (CAD) tool.

1. Introduction

The market for mass-produced goods is becoming increasingly competitive. The success of a product depends not only on its quality but also on the time taken to introduce it in the market. As a result, manufacturing industries are finding it necessary to streamline their activities and to eliminate redundancies. Shared databases, CAD/CAM systems and unified product and process representations are some methods that are used to achieve this objective. A problem that has defied usable solutions is the representation and interpretation of manufacturing tolerances. Tolerances, though a critical aspect of manufacturing, have received insufficient attention from designers, and this has often resulted in problems.

Manufacturing methods have a process capability or maximum precision; the dimensions of parts manufactured can only be controlled to within this precision. Each dimension has an inherent uncertainty or tolerance associated with it which must be taken into account when designing parts or assemblies. Designers use toleranced engineering drawings to define a variational class of parts that are functionally equivalent. The drawings are then passed on to the manufacturing department, which is responsible for producing the part in the required quantity.

Calculations are typically made with nominal dimensions, as most analytical models used by engineers are not created with variability of part dimensions in mind. However, dimensional variations can play an important role in a device's behavior. The effects may be easily detected, such as two parts that do not assemble or more subtle, such as misaligned bearings that cause a shaft imbalance and subsequent fatigue failure. Designers tend to handle potential problems of this nature by overdesigning, *i.e.* by using tight tolerances. Unfortunately, this often results in a steep increase in production cost and difficulty.

Every mechanical part has several *features* that enable it to perform its function. Walls, holes, ribs *etc* are examples of such features. Each feature has a set of basic properties: location, size, form, orientation and surface finish. The ANSI-Y14.5M [1] dimensioning and tolerancing standard lays down conventions for specifying the properties of each feature on a part. Properties such as location are often defined from a standard datum and hence a complete specification includes the value of the property variable and the datum from which this variable is to be measured. The ANSI-Y14.5M document also includes conventions for recording datum surfaces and features. For readers who are not familiar with ANSI-Y14.5M, a brief review is provided in Appendix I.

The objective of design is to develop a description of a part that performs a specified function. One aspect of the desired behavior of any part is that it should be manufacturable. Tolerances are used by a designer to ensure that the entire class of objects produced by a manufacturing process will acceptable from a functional point of view. ANSI-Y14.5M states that tolerance values should be specified so that this functional intent is achieved. However, since it is only a drafting standard, it does not suggest ways of deciding what tolerances are appropriate. It also does not establish conventions for inclusion of explicit information about manufacturing processes or desired behavior of the part. When such information is essential, the standard suggests that it be specified as a footnote or in another document referenced on the drawing. The choice of a tolerance is governed by factors such as the ease of manufacture or assembly and static or dynamic behavior of the design. However once the tolerance is assigned, all this information is discarded even though it may be significant during the life cycle of the product. This causes a communication gap between the original designer and others who are associated with the design. Examples of some resulting problems are:

1. Conservative allocation strategies used by designers often make it necessary for manufacturing engineers to consider loosening the tolerances in order to reduce process costs. A conventional engineering drawing does not contain enough information to help the engineers decide whether ignoring a tolerance specification will prevent the modified part from exhibiting the desired behavior, *i.e.* which tolerances or dimensions are critical to the part's functionality.
2. Complex drawings may have hundreds of numbers on them and, as a result, often contain errors. Ideally, most of these errors should be corrected on the shop-floor itself, without consulting the original designer. Lack of information on the drawing makes it necessary to re-consult the design department, resulting in costly delays.
3. Designers often allocate tolerances by reference to older designs. However, two geometrically similar designs can perform very different functions and may require different tolerances. Availability of functional information enables designers to compare designs using more appropriate criteria.

These and other similar problems have created an urgent need to provide designers with tools that can help them reason about the effects of tolerances on the function of a design. Before such tools can be created, it is necessary to be able to represent both behavior and tolerances and to understand the relationship between them.

1.1 Scope of the current work

The research presented in this report is most relevant to the tolerance synthesis problem, *i.e.* how to decide on appropriate tolerances. We are concerned with discovering the semantics of tolerance. This requires an investigation of the rationale behind tolerance allocation. Superficially, a set of tolerances is a set of complex constraints imposed on the geometry of a design. However, each of these constraints is necessitated by some aspect of the product's life cycle. Certain tolerances may be required for assemblability, others for manufacturability and still others for purely aesthetic reasons. We believe that it is important that this information is identified and represented because it will be useful when the product is redesigned or modified. Even during the design process, it will serve to increase the designer's awareness. This link between function and geometry is what we have attempted to discover through our research.

Our fundamental assumption is that the purpose of the mechanical design process is to create a device that exhibits a specified *behavior.* This desired behavior could be a kinematic trajectory, heat-transfer characteristic, or even that the device be manufacturable on a particular type of machine tool. A designer uses tolerances to define an entire class of devices that satisfy this behavioral specification. A valid tolerance allocation ensures that if produced to that tolerance, the device is guaranteed to be functionally acceptable.

As mentioned earlier, the relation between component tolerances and device behavior is not explicitly stated in conventional representations. There are a few notable exceptions such as the design of mating cylindrical parts or journal bearings. In these cases, the type of fit desired and the nominal diameter of the shaft alone can be used to determine the tolerances that are required. We would like to develop formal methods for relating tolerances and desired behavior. To reduce the problem to a manageable size, we only consider specifications on kinematic behavior.

We use configuration spaces (c-spaces) to represent kinematic behavior. C-spaces are higher-dimensional spaces that were first used in robot collision avoidance problems. They can be used to reason about geometry and behavior at multiple levels of resolution. We will demonstrate the use of of c-spaces to diagnose sensitivity to tolerances and also present some other applications, such as wear prediction.

The long term goal of this research is to explore the use of behavioral specifications in design and manufacturing. As a first step, we have attempted to identify the types of geometric controls exercised by designers through the use of tolerances and to relate these changes in geometry to the function of the device. We have adopted a representation that can be used to explicitly store information about device behavior. We believe that such specifications can be used to supplement the types of information already in use to develop a more efficient design and manufacturing cycle.

1.2 Outline of the report

We begin in Chapter 2 with a brief survey of research on tolerancing. The objectives of tolerance research are reviewed, and some of the important issues that arise when applying the available technology are discussed. A computer implementation that we have developed which provides some of these facilities is described.

Chapter 3 addresses the issue of behavioral and functional representations. Some existing representations are surveyed and c-spaces, with the associated formalisms, are introduced. The chapter concludes with a discussion of methods for exact calculation of c-spaces.

Chapter 4 introduces the tools necessary for designing with c-spaces. The significance of paths in the c-space is discussed and a number of operators that can be used for diagnosis of problems are introduced. Several examples are provided to illustrate the use of these operators.

Chapter 5 describes the existing computer implementation, problems encountered when using it for practical problems and possible solutions.

Chapter 6 summarizes the results achieved and provides a summary of future applications and possible extensions of the technique.

For those readers who are unfamiliar with the ANSI standard on tolerances, Appendix I provides a brief overview of the important features of the standard.

2. Tolerance Research

2.1 Introduction

There has been considerable interest in tolerance research since the concept of interchangeable manufacture was developed. Significant improvements in tolerancing practice were achieved during World War II because of the increased demands placed on production facilities during that period. More recently, researchers have worked on representation of tolerances in solid modeling and the application of mathematical programming techniques to tolerance analysis.

Research in this field can be broadly grouped into three categories: representation, analysis and synthesis. Each of these areas has different objectives which we will briefly outline and relate to our own research. We pay special attention to analysis and synthesis, as they are the most relevant. We have also included a discussion of the issues involved in choosing which of the available techniques to apply in a given situation.

2.2 Representation

Solid modeling is an emerging technology that has the potential to replace databases that use engineering drawings. Researchers in the area have been interested in representing tolerances because this capability is essential before solid models can be used for manufacturing and process planning.

A toleranced engineering drawing represents a variational class of objects. Conventional solid models represent only nominal objects and hence cannot be used to reason about tolerance-related problems. There are three popular solid modeling techniques [29], each with inherent properties that affect its suitability for representing tolerances. We will briefly discuss the constructive solid geometry (csg) and boundary representation (b-rep) models,

- A csg object is built up by using intersection and union operations on a set of simple primitives. The primitives usually include spheres, cylinders, cuboids, torii *etc.* Each primitive has a set of parameters that are used to define it; for example, a sphere has a radius and a cylinder has a radius and a length. The geometry of an object constructed using these primitives can be controlled by variations in the parameter values. However, the space of variations that can be obtained in this manner does not include the types of variations that often occur in industrial practice. For example, it would be impossible to represent a surface finish variation on a part without using an extremely complex primitive. Hence, the potential for using a purely csg-based representation to represent toleranced parts seems low.

- The b-rep solid modeling technique, defines an object in terms of its vertices, edges and faces. It is possible to represent even reasonably complex objects without having to store a large csg tree. A property of the b-rep model which is now being explored [17] is that it can be used to represent unrealizable entities such as ideal planes or lines. This can be of considerable value in representing datum surfaces for a tolerancing scheme. The ability to access individual edges makes it relatively straightforward to represent a variety of tolerances. Properties such as surface finish can be associated with each face, thus allowing a part to have different surface finishes in different areas.

These preliminary thoughts seem to indicate that the b-rep scheme is better suited to representing tolerances if existing solid modelers are adapted to conform to industrial tolerancing practice. A usable solution to the problem must allow a user to represent form, orientation, surface finish, and other tolerances that are not simple variations of linear dimensions, without excessive complexity. Many of the research efforts in this area have yielded results that are sufficient for producing engineering drawings but do not support reasoning about assembly or manufacture. One innovative approach is the offset boundary method proposed by Requicha [30]. Unfortunately, adopting this, or similar methods, would require that unacceptably large changes be made to current engineering practice.

2.3 Analysis

The tolerance analysis problem consists of deciding if the tolerances specified on a design are acceptable with respect to the applicable constraints. The allocation is satisfactory if no allowable variation in the geometry can lead to violation of a design constraint. Design constraints can be used to prevent redundant dimensioning, excessive tolerance stackup and other problems. Tolerance analysis is usually based on a worst case scenario as it is difficult to know the probability distribution of tolerance variables.

The initial work in tolerance analysis was aimed at computerization of existing manual techniques such as the tolerance chart method [22]. This technique allows a process engineer to simulate, with pen and paper, the machining of a part from rough stock. Using tolerance chart analysis, the engineer can check the datum surfaces used, amount of material removed, tolerance stackup and other critical variables. Tolerance chart analysis is applicable to one-dimensional problems.

More recently, mathematical programming techniques have been applied to the solution of the tolerance analysis problem by Hillyard [18] and Hoffmann [19]. Turner [36, 37] has developed methods to represent possible geometric variations as a multi-dimensional vector space which can be analyzed using existing mathematical techniques. Some simulation based solutions to the problem have also been proposed [6]. These techniques can be used to check for violation of a variety of design constraints.

Currently available computer-aided tolerancing systems such as BYU-CATS [16] provide both worst-case and statistical analysis of 1-D stackup problems. BYU-CATS requires the user to input the design constraints, which are then checked automatically. If constraint violations are detected, the program can generate suggestions such as scaling or reduction of tolerances. Turner and Wozny [37] describe the GEOTOL system that uses the vector space approach (describe in detail in Section 2.5.4. This system is integrated with the Geometric Design Process(GDP) solid modeling system developed at IBM.

Most tolerancing systems provide the capability of performing 1-D analysis for worst-case situations. A 3-D analysis runs into serious combinatorial problems because of the coupling between different dimensions. Statistical solutions are used, but they require extensive computation (*e.g.* Monte-Carlo methods) and must use unreliable models of tolerance variation. Greenwood and Chase [6] have shown that assuming the standard Gaussian distribution is usually incorrect. Before a practical and widely applicable tolerance analysis system is developed, it will be necessary to considerably refine the process models in use and also to extend tolerance analysis to three dimensions.

2.4 Synthesis

The tolerance synthesis problem deals with the optimal allocation of tolerances. The standard heuristic for tolerance allocation is to use as loose a tolerance as possible, subject to design constraints. Most tolerance synthesis research is directed at minimizing cost, but the objective function may also contain other terms (for a list of cost-tolerance models, see Greenwood and Chase [6]). One of the serious drawbacks of the cost-minimization approach is that most existing cost models are *ad hoc*; reliable statistics of this kind have not been maintained in industry.

Methods of mathematical programming have been used quite extensively to obtain least-cost solutions. Earlier work was restricted to linear constraints but non-linear programming is now used. With the introduction of techniques such as branch and bound search, it has also become possible to include complexities such as the availability of multiple production processes.

Research on tolerance synthesis based on functional requirements has been restricted to fields such as the design of journal bearings, where the coupling between tolerance and function is very strong. There has also been very little effort made to understand the relationship between the tolerances on a part and the function it performs. We hope to make a contribution to the knowledge in this area.

2.5 Important Issues in Tolerance Research

This section examines some of the issues relevant to using optimization techniques for tolerance analysis and allocation. The available mathematical programming and statistical techniques for the solution of these problems are reviewed in more detail and the relative advantages of the methods are discussed.

2.5.1 Analysis vs Synthesis

Analysis is concerned with checking designs that already have tolerances allocated. Every mechanical design has a set of critical dimensions and clearances without which it cannot perform its function. Tolerance analysis is used to check whether these critical dimensions are achieved when a particular tolerance allocation scheme is used.

The most common problem encountered is tolerance stackup. Often, a dimension is not created directly but by concatenating several component dimensions. If this is the case, the tolerance on the resultant dimension is the sum of the tolerances on all the component dimensions. Consider the situation in Figure 2-1. In this drawing, the dimension C is required to be 10 +/- 0.001 mm. Now, when the part is actually machined, C is created as the result of machining dimensions A and B. A simple calculation shows that if A and B are machined to 5 +/- 0.001 each, the specification on C will not be met. This is a one dimensional example of the tolerance stackup problem.

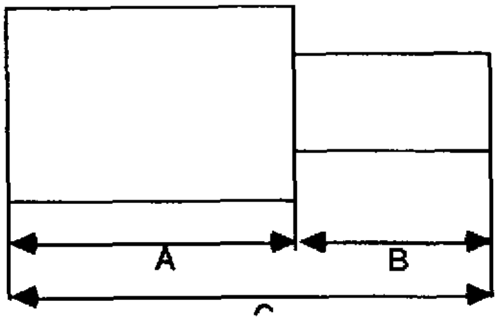


Figure 2-1: Simple Stackup Problem

In 1-D assemblies, like the one shown in Figure 2-1, the stackup of tolerances is linear,*i.e.*

$$T_{ASM}=\sum T_i$$

and the analysis is quite simple. In 3-D cases, the assembly function, which relates the overall tolerance to the component tolerances, is non-linear. The relation is,

$$T_{ASM}=\sum \left|\frac{\partial f}{\partial x_i}\right| T_i$$

where the partial derivatives are the sensitivities of the assembly tolerance to the component dimensions. Non-linear assembly functions require that non-linear programming methods be used and hence are significantly more difficult to handle than linear functions.

Allocation aims at optimal assignment of tolerances, subject to a set of design constraints. The most common objective function used has been the production cost, though now there is a move towards other measures of product suitability. Unfortunately, there is very little precise data available that relates manufacturing cost to tolerances achieved. Many production handbooks, *e.g.* Bralla [4], contain some approximate graphs such as the one shown in Figure 2-2. These are unsuitable for optimization because they have no analytical form. Some tolerance cost models developed for optimization are shown in Table 2-1.

A cost minimization problem can be formulated using these models and solved using mathematical programming packages such as GAMS [5]. It is important to note that the utility of these solutions is limited by the quality of the cost models. Expert committees that have examined the problem suggest that a better understanding of production processes is

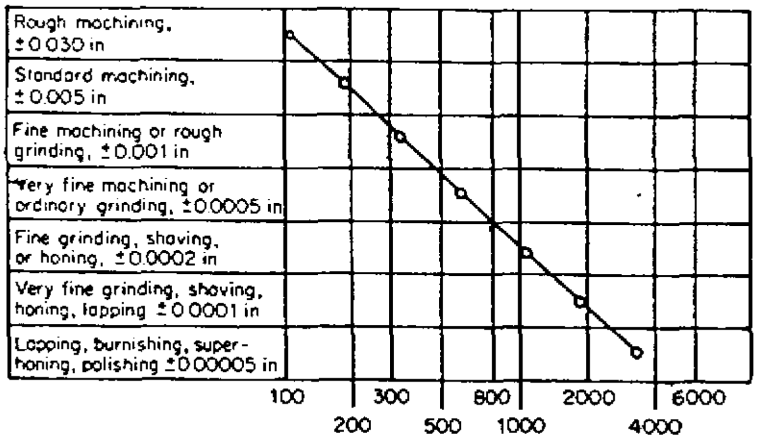


Figure 2-2: Tolerance Cost Curves

necessary before the actual cost of manufacturing a part can be understood [38]. Though a difficult task, some progress has already been made towards achieving this goal.

Reciprocal squared	$A+B/tol^2$	Spotts [35]
Reciprocal	$A+B/tol$	Chase & Greenwood [16]
Exponential	$A\exp(-B(tol))$	Speckhart [34]

Table 2-1: Tolerance Cost Models

2.5.2 Worst-Case vs Statistical

worst-case analyses deal exclusively with the upper and lower bounds on dimensions. A drawing that has satisfactorily passed a worst-case analysis is guaranteed to result in a 100% interchangeable assembly. Worst-case analysis is easy to apply because it requires only knowledge of the limits of a tolerance variation. Unfortunately, the constraint of 100% interchangeability is often stronger than necessary and tends to reduce the tolerance on the entire batch. There are situations where it is cheaper to perform a selective assembly and discard the units exhibiting extreme variation, than to tighten tolerances on the entire batch. Tight tolerances raise production times and thus increase costs. This is significant in the context of modern small-batch manufacturing where cycle-times must be kept low and even a small increase in price or drop in throughput results in loss of market share.

To perform a statistical analysis, it is necessary to assume or to estimate the probability distribution of each component tolerance using sampling techniques. Most of the widely used statistical techniques assume that the underlying probability distribution is Gaussian. Sums of normally distributed variables can be proven to follow a normal distribution and hence the assembly tolerance is also assumed to have this distribution. This assumption is incorrect in most practical situations [6]. Changes in dimensions of the product of a manufacturing process are intimately related to conditions in the production process and are often heavily skewed and shifted. This makes many techniques that use the Gaussian assumption unreliable.

If the distributions of the component tolerances are known to some certainty, *i.e.* all the production processes are well understood or "under control", it is possible to significantly reduce the required tolerances on an assembly while maintaining a high acceptance rate. For the stackup cases presented in the previous section, it can be shown that if the underlying distributions are normal and the worst-case tolerances are assumed to correspond to the 6σ limits, the component tolerances add as the root sum squared [6]. For 1-D cases, this is

$$T_{ASM}=[\sum T_i^2]^{1/2}$$

and for the multi-dimensional cases,

$$T_{ASM}=[\sum (\frac{\partial f}{\partial x_i})^2 t_i^2]^{1/2}$$

A recalculation of the stackup example of Figure 2-1 shows that looser tolerances are required if a statistical average is used.

Though they allow looser tolerances, statistical models often overestimate the assembly yield. Additionally, the presence of a single low-precision component in an assembly can cause the required tolerances on other components to become tighter than if computed using the worst-case model [6]. Tolerance analysis using statistical techniques is a Monte Carlo method and requires extensive simulation of the assembly process.[2] These simulations are often too expensive or simply impractical to perform.

[2] An interesting, though admittedly unrelated, observation in this regard is that most programs only simulate fit and not assemblability. A bolt may have a smaller diameter than a bolt hole intended for it but the hole may placed such that the bolt can never be actually inserted. because there is no approach path.

It is therefore important to decide before the tolerance allocation stage whether 100% interchangeability is really necessary. If so, a worst-case analysis must be performed, otherwise a conservative statistical study will probably yield better results.

2.5.3 1-D vs 3-D

Most texts that discuss tolerancing deal exclusively with one dimensional assemblies (see for example Peck [25]). The reason for this is that 1-D tolerance stackup problems can be solved by hand even in complex situations. The upper and lower bounds for each tolerance are available and it is possible, by considering only these bounds, to compute limits for the assembled dimensions. This is possible because of the linear assembly functions. When 2-D or 3-D cases are considered, the assembly function becomes non-linear and we cannot calculate the bounds on the assembly function by a simple interval arithmetic calculation; an extensive search must be performed before bounds can be determined.

It is possible, based on the rigid body assumption, to propagate tolerances in three dimensions through an assembly. However, if any optimization is attempted, the non-linearity of the assembly function poses a problem. The problem can be solved by non-linear programming methods if there are only a few, well conditioned, equations but in many cases only a brute force approach can yield the optimum. Heuristics can be used to greatly reduce the search space but we are not aware of any efforts to identify such heuristics. The most commonly adopted approach to solving the non-linear problem is to linearize the objective function and constraints about the nominal dimensions and use reliable linear programming methods. A commonly used justification for this step is that tolerances are usually small and linearization does not introduce gross inaccuracies.

2.5.4 The vector space approach

All the optimization approaches use the basic assumption that tolerances define a region in higher dimensional space. Each dimension of this space corresponds to a variable geometric dimension on the part. The design constraints dictate that only a part of this space is acceptable and the optimization algorithms attempt to find points in this space that maximize or minimize certain quantities such as cost.

Turner has proposed the vector space approach [36], which is a generalization of the above formulation. A "normed vector space over real numbers" is constructed by selecting a set of independent geometric variations that form the basis for this vector space. Three kinds of variables are defined by Turner in his thesis,

1. *Tolerance* variables represent the variation of a toleranced property from its nominal value. A toleranced property is usually a geometric dimension that can be explicitly controlled by the manufacturing engineer
2. *Design* variables represent the variation from the nominal of aggregate properties of the product that cannot be directly controlled by the manufacturing engineer
3. *Model* variables are measures of how much an elementary geometric property of a part varies from the nominal

The choice of model variables governs the variational properties of the design. For example, when manufacturing a cuboid if only height, length and width variations are chosen as model variables, it will not be possible to represent a parallelism error between two opposite faces. The model variables thus define a vector space and their relationship with the design variables can be used to specify an in-design region of this vector space. The tolerance analysis problem consists of checking whether any combination of tolerance variables satisfying a specification results in a point in the in-design region. In other words, the in-tolerance region must be a subset of the in-design region of the vector space.

It is very important that the choice of model variables be made correctly. Two important properties are,

- The model vectors must be linearly independent, *i.e.* it must not be possible to produce the geometric variation governed by one model variable by a linear combination of the other model variables.
- Vector space properties such as closure and associativity must hold.

Turner and Wozny [37] have reported a system that interfaces with a solid modeler to automatically extract a set of model variables.The c-space approach which we introduce later can be used to eliminate model variables that do not have a significant effect on the behavior of the device.

Dimensionality is a significant issue when using the vector space approach. To capture all the different variations in location, size,orientation and form it is necessary to have a large number of model variables. These have to be related to design and tolerance variables and only then can an optimization problem be formulated. Though the technique has been successfully applied to smaller examples, in a complicated assembly the large number of model variables may be a problem. For the designer to gain insight into the design and its functioning, methods for selecting the most important variables and the sensitivity of the design to them must be developed. These can be combined with the vector space approach to rapidly yield optimal solutions.

2.6 Summary

Significant advances in mathematical programming have made it possible to solve hitherto unsolvable problems. A wide range of linear and non-linear problems can be solved and least-cost tolerance allocation schemes calculated using either the worst-case or statistical analyses. However, there is still a need for tools that assist the designers in tolerance allocation at the preliminary stages of the design. After the basic design has been decided and shown to be feasible, the problem reduces to one of parametric design and is well suited for the application of math programming. An ideal design methodology uses the capabilities of a competent designer to create a suitable conceptual design which is then optimized automatically. This method takes advantage of the capabilities of both the designer and the optimization technique.

3. Representing Behavior

3.1 Introduction

The objective of the design process is to produce a device that exhibits a specified *behavior.* A device is said to perform its *function* correctly if it displays the behavior desired by the designer. Function may therefore be defined in terms of a desired behavior.

The subject of functional design has been of interest to practitioners of mechanical design for a long time. It is generally accepted that greater flexibility is retained in the design if the actual choice of components is made late in the design process [8]. To apply this "least commitment" strategy, designers must use abstractions in the initial stages of the design process. High level functional descriptions are often the logical choice.

In disciplines such as circuit design, it is possible to design almost completely with functional primitives and to consider specific components only when the design is actually fabricated. The reason for this is that interactions between components can be easily controlled. Two circuit elements only interact if the designer connects them. This is not the case in mechanical design; geometric elements may interact even if the designer has not consciously caused them to. Hence, reasoning in terms of behavioral primitives or high level entities is abandoned quite early and designers introduce a probable geometry which is suitably modified as the design progresses.

Researchers have made efforts to identify different types of behavior exhibited by devices. Collins [7] presents one such analysis based on helicopter engine parts. Some of the behavioral types or primitives identified by this study are shown in Table 3-1.

Researchers in Europe have developed standardized symbols for representing engineering functions. One set of standards for representing the handling of small parts has been published by the Society of German Engineers. A set of symbols for basic physical functions, due to R.Koller [8], are shown in Figure 3-1.

These symbols and descriptions can be of significant use for conceptual design. However, they would be of limited use if included with the final design drawings. Reasons for this include the fact that the symbols lack specifity, are often ambiguous and that covering a

1. Supporting
2. Attaching
3. Motion constraining
4. Force constraining
5. Friction reducing
6. Pivoting
7. Oscillatory sliding
8. Removable fastening
9. Oscillatory rolling
10. Permanent fastening
11. Motion damping
12. Force distributing
13. Clutching
14. Flexible spacing
15. Deflecting
16. Spacing

Table 3-1: Sample list of behavioral features

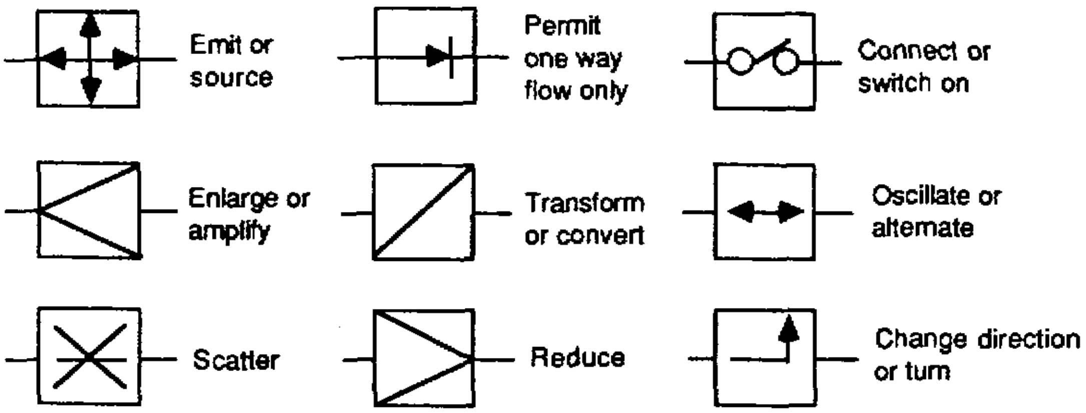


Figure 3-1: Symbols for basic functions

general design area would require hundreds of such symbols, most of which the designer would never use.

A functional description of a design must be more detailed and compact than the primitives described above. The representation must support reasoning about functional descriptions in as much detail as necessary; it should be possible to use qualitative reasoning for gross decisions and precise calculations for detailed ones. In rare cases, we can express the desired behavior of a part as a trajectory in real number space and the actual behavior as a mathematical function of the part's geometric parameters. This happens only in some well understood situations which are not considered in this report.

To reduce the research problem to a manageable size, we have limited the types of behavior considered. We have chosen to analyze kinematic behavior because it is a major objective in mechanical design. We attempt to use available results in the areas of qualitative kinematics, statics and motion planning to identify and develop representations of kinematic behavior suitable for use in a variety of inferencing tasks.

3.2 Qualitative representations

Several efforts have been made to develop qualitative descriptions of parts and assemblies. These have all aimed at the objective of performing useful inferencing based on approximate information. Availability of reliable qualitative reasoning techniques helps to ensure that gross errors or discrepancies may be detected and corrected without detailed calculations. Qualitative methods allow the designer to perform approximate analyses early in the design process, even if the design description is incomplete.

Significant progress has been made in developing theories of qualitative physics (see for example, [9]). It is now possible to reason about dynamic systems using qualitative differential equations and to predict trends and major transition points. We now review some of the published research in qualitative statics and kinematics.

3.2.1 Statics

Nielsen et al [24] have developed a theory of statics and attempted to use it for the analysis of real-world devices. The theory is based on qualitative descriptors of translation and rotation. The translation directions correspond to up-down, left-right and back-front. Translation along a positive axis is denoted by a "+" sign, translation along the negative direction of an axis by a "-" sign and a state of no motion is represented as "0". A similar convention is used for clockwise and counterclockwise rotations. Predicates are available to check if an object can rotate or translate about a given axis.

The representation allows the user to represent contact as well as external forces and torques. Based on a description of the assembly and an input motion of one of the components, the program attempts to predict the motion of the other objects in the assembly.

By assuming that two objects interact only when in direct contact, motion effects may be propagated to other objects in the assembly.

Because of the limited information content of this qualitative representation, it is often not possible to unambiguously predict kinematic behavior. For example, consider two blocks on a table such that a vertical face of one block is in contact with one of the vertical edges of the other. If one of the blocks is moved, it is not possible to predict whether the other block will translate, rotate or both without more detailed information about masses, friction coefficients *etc*. When such unpredictable situations occur, all the cases must be considered and a set of possible motions generated.

Other qualitative representations of contact and position have been introduced, especially in the domain of assembly planning for robots [28]. The utility of these representations is also limited by their information content. This is another manifestation of the knowledge-search duality.

3.2.2 Qualitative Kinematics

Forbus et al [15] have proposed a framework for qualitative kinematics. One of their basic assertions, the Poverty Conjecture, is :

> There is no purely qualitative general purpose kinematics.

This means that to perform an arbitrary reasoning task, it is necessary to maintain specific information. When a qualitative description is not sufficient to answer a question, a detailed description must used. The qualitative description is called the place vocabulary, and the detailed description, the metric diagram. These terms are defined by Forbus *et al* [15] as follows,

- *Metric Diagram*: A combination of of symbolic and quantitative information used as an oracle for simple spatial questions.
- *Place Vocabulary*: A purely symbolic description of shape and space, grounded in the metric diagram.

Though it is always possible to derive the place vocabulary from the metric diagram, this can be a time consuming process and the place vocabulary is usually maintained explicitly.

The definitions given above do not specify the form of the metric diagram and place vocabulary. Depending on the type of problem, the metric diagram could use real numbers, algebraic equations *etc.* A place vocabulary that is appropriate for each individual problem must be defined and then derived from the metric diagram. We now discuss the use of configuration spaces as the basis for deriving place vocabularies. This approach has been demonstrated by Faltings [11] and independently by Joskovicz [20], who uses the term state diagram.

3.3 Configuration spaces

The term configuration space (c-space) was introduced by Lozano-Perez [26, 27] in the domain of motion planning for robots. More recently, this representation has been applied to the analysis and design of mechanisms by Joskovicz [20] and Faltings [10].

3.3.1 Basic Theory

A configuration of an object is a vector of six parameters - three positions and three orientations. The configuration space or c-space consists of all the possible values of this vector. If there was only a single object in the universe, its position along the x,y and z axes could vary from $-\infty$ to $+\infty$ while its rotation about these axes could vary from 0 to 2π. However, in reality, this range is reduced by the presence of other objects.

Consider a mechanism with two links, each with six degrees of freedom. As two objects cannot overlap in space, some configurations are illegal and the c-space of each link is partitioned into two subsets, corresponding to legal and illegal placements. The illegal or forbidden region is shown as the shaded area in all the figures in this report. Since motion of the mechanism consists of passing through a series of legal states, any motion can be represented as a continuous curve in the legal region of the c-space.

A simple mechanism with two links can be constructed such that each link has exactly one degree of freedom. Figure 3-2(a) shows such a mechanism. The disk is constrained to rotate about point O and the rod to translate along the direction L indicated. The configuration vector for this mechanism thus has two elements: Θ, the angular position of the disk and X, the linear position of the rod. Taken separately, the ranges for X and Θ would have been $[0,+\infty]$ and $[0,2\pi]$. However, since the rod cannot overlap the disk, its range of motion becomes confined to the intervals $[R,+\infty]$. The range of rotations for the disk remains unaffected. The c-space is divided into regions corresponding to the legal and illegal placements of the links, depicted as the unshaded and shaded regions respectively.

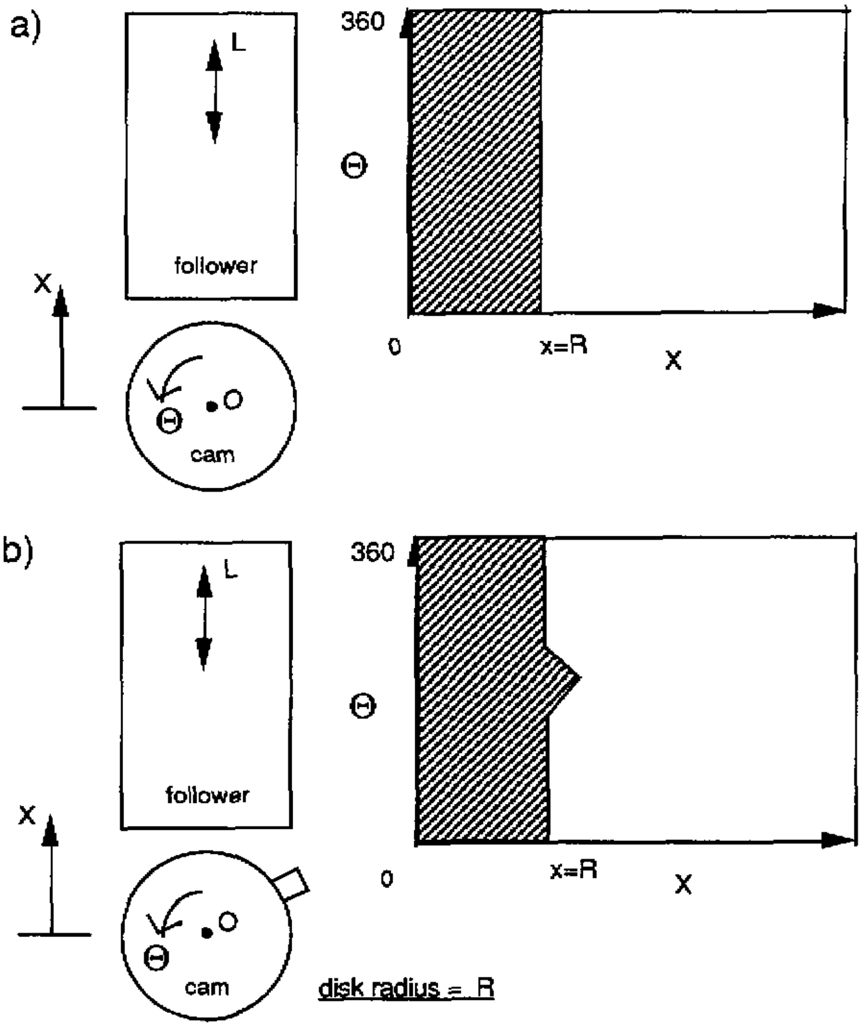


Figure 3-2: Simple Configuration Spaces

The next example, shown in Figure 3-2(b), is slightly more complicated. The form of the mechanism is similar to that of part (a) except for the addition of a projection on the disk. This introduces a change in the behavior of the device which is indicated by the appearance of a notch in the c-space. To understand why this change occurred, trace the sequence of events as the cam rotates counter clockwise. As the left upper tip of the projection touches the follower face, it pushes the follower upward causing the X value of the follower to increase. This is the beginning of the notch in the c-space (Figure 3-2(b)). The motion continues until a position is reached where the top face of the projection and the bottom face of the follower are parallel and in contact. This sequence of events is then reversed as the cam continues to rotate and the follower descends. If the height of the projection is gradually reduced, and the c-space repeatedly recomputed, the depth of the notch will decrease until the original c-space of part (a) is reached in the limiting case where the projection disappears completely.

It is usually impractical to calculate the entire c-space of a mechanism in a single step because of the high dimensionality of the space (6n, where n is the number of links). A useful approach is to consider the mechanism to be composed of smaller functional units. The c-space for each individual unit is calculated and these are composed to give the c-space for the entire mechanism. Methods for doing this are suggested by Joskovicz [20] who rigorously proves the validity of this procedure. Our discussion limits itself to the domain of two-axis mechanisms because they can be analyzed without excessive computation but still serve to illustrate the important underlying principles.

3.3.2 Important Properties

Joskovicz [20] has identified several interesting properties of c-spaces and provides a detailed listing of these in his thesis. The most relevant for tolerance analysis is that the c-space of a mechanism can be considered as the union of several connected subsets. There is a one-to-one correspondence between these subsets and possible motion types. For fixed axis mechanisms considered, the possible motions are linear, rotational and helical. Complex mechanisms consist of a closed chain of kinematic pairs (see Reuleaux [31]), which are the smallest functional sub-assemblies of the mechanism. The c-space of a kinematic chain can be calculated by composing the c-spaces of all possible kinematic pairs. A useful simplification is that only those elements that can actually come in contact need be considered as kinematic pairs.

Qualitatively similar subsets of the c-space are referred to as *regions* by Joskovicz. He used the c-space as the basis for calculating a *region diagram* which shows all the regions and possible transitions between them. Faltings [12] has used the similar idea of *places*. This term was originally suggested by Forbus [13, 14] to refer to possible regions of motion of a point mass amidst polygonal obstacles. Faltings has extended the idea to c-spaces and defines a place as a set of qualitatively equivalent points in the c-space. Behavior can then be defined by listing all the places and the transitions between them in a representation called a *place graph*. He has explored methods of making a transition between the place graph and c-spaces. The concept of qualitatively similar classes of behaviors is used later in the paper to study the effects of tolerance specifications.

All motions of a mechanism must occur within the legal region of the c-space. Any motion can be represented by a start and an end point in the c-space which are joined by a continuous curve. This curve may be arbitrarily complex but, to represent a legal motion, it must be continuous. We refer to such curves as *paths* in the c-space. Most interesting paths occur along the boundary of the legal region of the c-space. Introduction of external tendencies, such as springs, cause the mechanism to follow certain paths preferentially. This is seen in the design examples presented in the next section.

One of the major benefits of using the c-space approach is that we can *automatically* provide designers with insights into the behavior of a device that are not evident from its geometry. A critical property for such reasoning is that each subset of the c-space is associated with a qualitatively similar set of behaviors. Much of the analysis we present here relies on this basic property of topology (Euler number) of the c-space. The topology of the configuration space is a property that the designer would like to preserve since a modified topology is indicative of a qualitatively different behavior. However, as seen in the following sections, if topology alone is used as a descriptor, significant changes in the behavior may be overlooked. Hence, it is necessary to also maintain other geometric information, such as the number of segments constituting the boundary, the convexity of individual regions *etc.*

Perturbation of geometry to test for changes in behavior has been demonstrated by Faltings [11] in his thesis, but he provides no method for selecting which parts of the geometry to perturb. This is significant, because exhaustively modifying all combinations of geometric parameters is not feasible. We propose heuristics that can be used to determine candidate dimensions for perturbation.

Once the perturbation has been performed and the c-space recomputed, it is necessary to *perform diagnosis, i.e.* to *determine* if a major qualitative change has occurred in the behavior. The following are some specific changes that occur in the c-space and can be used to predict unacceptable variations in the functioning of the device. They are, listed in order of increasing importance,

1. *Changes in the topology of the c-space.* Appearance or disappearance of regions indicates that the basic type of motion is changing.
2. *Changes in the boundaries of the c-space.* Though topology may be unaffected, a geometric variation may cause the appearance of new boundary segments. Since this indicates a change in the nature of contacts and hence the place vocabulary, it is probably noteworthy.
3. *Changes in geometric parameters.* The above categories do not account for changes in the geometric properties such as the height to width ratio of a region. This type of change may also be indicative of undesirable changes in behavior.

The c-space can be used to monitor changes in the qualitative behavior of the device, the nature of the physical contact occurring and the magnitude of relative motions along different axes. In addition, it is possible to directly perturb c-space boundaries to predict the effect of phenomena like wear and failure.

3.3.3 Exact calculation of c-spaces

To use c-spaces as a useful analysis and representational tool, we require techniques to rapidly calculate the c-space from geometry and motion specifications. This is a difficult problem, because of the high dimensionality of the c-space of any real world mechanism. Calculation of c-spaces for lower dimensional cases has been studied extensively in the motion planning literature. A review of algorithmic approaches to motion planning is presented by Yap [39].

The *most general method for calculation of c-spaces,* developed by Shwartz and Sharir [32], is based on the calculation of boundaries between legal and illegal placements. An object is described in terms of its faces, edges and vertices. Consider two objects A and B with a vertex of A in contact with a face of B. This contact would eliminate an entire half plane of translational motions for object A if the face on B were infinite. Due to the finite size of all faces on B, the half plane is delimited by applicability constraints that are governed by the size of the face. Calculation of the configuration space involves finding the constraints corresponding to all possible contacts between the two objects and intersecting the half planes of motion defined by them. This requires algebraic techniques for determining the zeroes of higher order polynomials. The large number of feature contacts possible between two objects, the fact that the order of the constraint polynomials increases with that of the object boundaries and the complexity of the algebraic methods required make this method of theoretical interest only.

The problems with handling higher dimensional cases require that we directly calculate low-dimensional cases and use composition or other techniques for the complete c-space. This is the approach we have adopted in our research. For 2-D cases, we have investigated the calculation and use of approximate c-spaces. Approximate c-spaces are calculated by simulation and can be of varying resolution. The use of special purpose hardware makes it to recompute the c-space a large number of times during a design.We have proposed a simulation approach which is discussed later along with several design examples.

3.4 Summary

We have reviewed some behavioral representations and their limitations. The concept of using c-spaces to represent kinematic behavior has been introduced along with methods of calculation and important properties of c-spaces. The reader is referred to the original articles for more detail on all the topics discussed.

Upto this point, we have described the problems faced by designers during tolerance allocation and proposed that behavioral representations based on c-spaces can be used to overcome some of these problems. In the next chapter, we present a number of useful operators that can be used in the c-space to aid analysis and design. These techniques can be used to create the basis for a design and manufacturing cycle that uses functional information to supplement engineering drawings. This methodology is shown to be superior to the current practice which is based purely on tolerances. The chapter contains a number of examples of practical mechanisms which have been analyzed or designed using c-spaces.

4. Working in C-Space

4.1 Introduction

To perform useful inferences based on the c-space, it is necessary to have operators that can be used to check for a range of potential problems. In this chapter, we introduce a basic set of c-space operators. (This set includes some of the abstraction operators introduced by Joskovicz [21].) We illustrate, with examples, the applicability of the operators to tolerance analysis.

4.2 Paths

Before using c-spaces for analysis, it is important to understand the significance of a path in the legal region of a c-space. The axes of a c-space represent motion parameters and hence a continuous curve in the legal region represents a sequence of states that the mechanism can pass through. All possible motions of a mechanism may be represented by such paths. A legal path has the following properties,

- It lies completely within the legal region of the c-space, *i.e.* no illegal states can ever be attained.
- It is continuous, *i.e.* it is not physically possible for parameter values to change discontinuously.

Paths P1 and P2 in Figure 4-1 are examples of legal paths.

In this report, we consider only two dimensional c-spaces, typically associated with kinematic pairs. Since the c-space is a kinematic representation, it cannot be used to predict motion. However, in practice there are often external *tendencies* such as gravity, springs or driving forces that act on mechanisms. An external tendency usually attempts to drive a parameter to its extreme value. Consider the c-space in Figure 4-1. If the mechanism is placed in the initial state O and there are no external tendencies, it will remain in that state. Without any knowledge of these tendencies, it is not possible to predict the path that will be followed. If, however, there is a tendency in the +X direction then path P1 will be followed until a contact occurs and no further legal motion can occur. Note that the Y value remains unchanged. Similarly, a -Y tendency will cause a motion along path P2 on which X is constant. It is interesting to note that forces and torques, because of the analogy between translational and rotational motion, can be both represented as tendencies on the appropriate parameters in c-space.

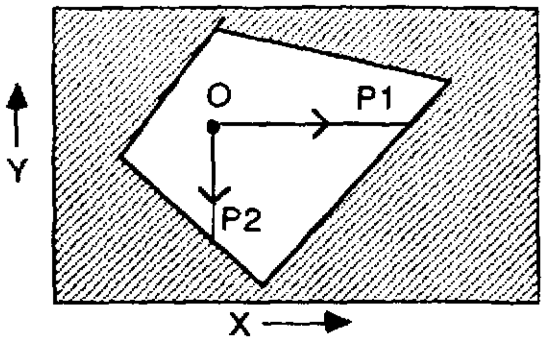


Figure 4-1: C-Space Tendencies

In most kinematic pairs, one link provides the driving force, the second link moving because of its contact with the first. Contact must be maintained for motion to occur and hence paths followed by the mechanisms are along the boundaries of the legal and illegal regions. The boundary represents all the legal configurations which involve contact between the two bodies. Often, gravity or springs are used to ensure that this contact is maintained. Figure 4-2(a) shows a cam-follower mechanism. Consider the behavior of this mechanism if a massless follower is placed in contact with the cam surface and an input rotation applied to the cam. The path followed is shown in the c-space of part(a). As there is no force acting on the follower, it maintains its position at a constant distance from the center of the cam. Compare this to part (b) which shows the path when the follower is loaded with a vertical spring. The spring loading forces the follower to always maintain contact and therefore to enter the depression in the cam. Similar behavior would result if the spring was removed but the follower was considered to have mass and be under the action of gravity.

The design of a path in the c-space is often of critical importance. The designer must often impose constraints on the relation between the two motion parameters that must be satisfied on the path . Once a suitable path has been found, it may be necessary to consider other constraints. The designer is free to modify the geometry of the design as long as the desired path remains feasible. In fact, the manner in which the path becomes infeasible when a geometric change is made can be used to determine active constraints on the design. The

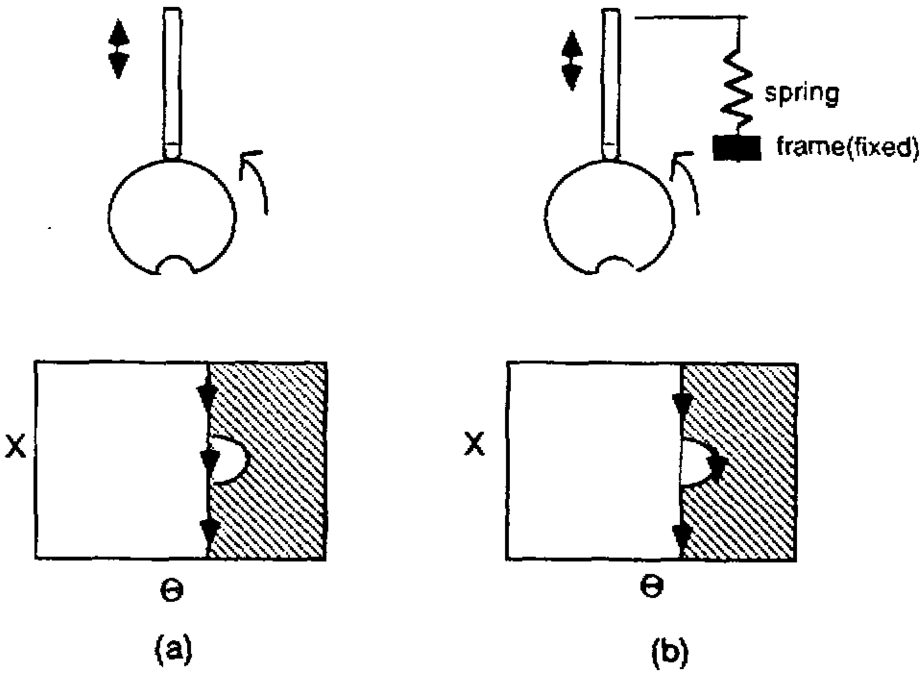


Figure 4-2: Paths along c-space boundaries

c-space of a hypothetical device is shown in Figure 4-3 with a desired path marked on it. Assume that it is necessary to make a wall of the device thicker to satisfy a stress condition; this causes boundary B to come closer and closer to path P until it finally disrupts it. The designer now knows that stress is one of the binding constraints in the design and can use this information to advantage. This also provides a useful way of visualizing the effect of non-geometric factors on the behavior of a device.

In Chapter 5, we describe a simple path generation algorithm. This algorithm uses properties such as slope of c-space boundaries, strength of the prevailing tendencies and momentum gathered during the motion to simulate the path that a mechanism would follow from an initial point in the c-space.

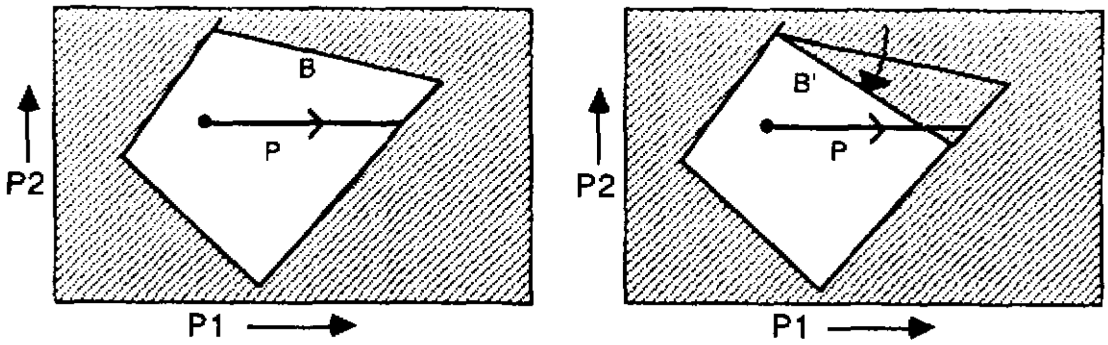


Figure 4-3: Effect of changing boundaries on paths

4.3 C-space Operators

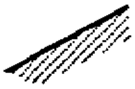
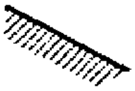
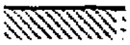
The operators introduced in the following sections are intended for use as safety checks on a design. The effects of these operators are similar to those of real world phenomena such as wear, misalignment *etc.* Designers can use them to check that the feasible path is not highly sensitive to these factors. These operators are inexpensive to apply and may be applied repeatedly without increasing the cost of the design. Most of the operators enable the designer to detect problems that will occur if changes are made to the geometry of the design. Examples of these are the operators that search for narrowly separated regions. Other operators can be used as abstraction mechanisms or to see if the c-space has a particular property. We have attempted to define all the operators in a manner suitable for application to bitmap versions of the c-space. This provides a significant speed advantage over operators that use a constraint-based c-space description.

4.3.1 Query Operators

These provide for ways of checking whether the c-space has a particular property. This capability is also useful when modifying existing designs to satisfy new specifications. Some useful questions that could be asked about a c-space are as follows,

- Is there a boundary segment with specified properties? Figure 4-4 shows a set of possible relationships between two parameters of a c-space. The parameters P1 and P2 may be either rotational or translational, giving another dimension to this table. At the top level, the existence of one or more boundary segments satisfying these properties can be queried. One level lower, if a segment is chosen in which P1 increases with P2, the rate at which this change occurs can be specified. The diagram only shows linear variations but, in principle, any monotonic function may be specified. This query operator can be easily be extended to handle checking for a series of adjacent segments, each with specified properties.

- Is there a place in the c-space with a specified force-displacment characteristic? Often, device function can be specified terms of properties of the c-space. For example,a ratchet can be described as a device that has a place (or state) such that a clockwise torques result in a motion but anti-clockwise torques have no effect. Hence, if a ratchet design is to be created by modifying one of a number of candidate designs, this operator can be used as a basis. Figure 4-5 shows the c-space for a locking device. The motion specification is that if the lock arm is maintained in a particular position then it must not be possible for the slider to move more than a small amount δ. The part of the c-space that satisfies this specification is marked in the diagram.

P2 Increases	
P2 Decreases	
P2 Constant	
P2 Unrelated	

Variation of P2 as P1 increases

Figure 4-4: Possible Parameter Relations

4.3.2 Perturbation Operators

Unlike the query operators which only provide information about the c-space, the perturbation operators check for special situations and modify the c-space accordingly. The resulting changes in the c-space must then be interpreted by the designer. Perturbation operators must be applied intelligently so that the designer can make useful inferences. Some features that are usually good candidates for application of perturbation operators are

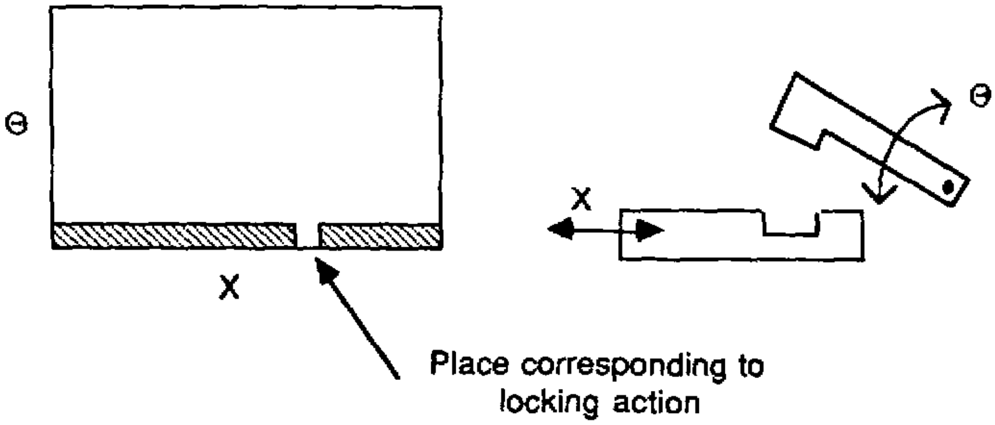


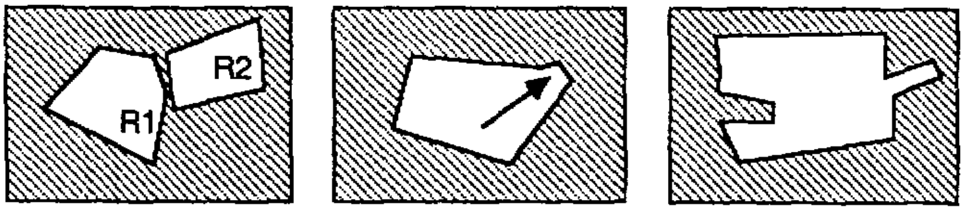
Figure 4-5: Existence of Places in C-space

as follows,

- *Regions that are narrowly separated.* Two unconnected regions in a c-space indicate that the mechanism has two distinct zones of motion which are unreachable from each other. The actual motion depends on the way the device is assembled. For example, in the c-space shown in Figure 4-6(a) the two legal regions, R1 and R2, are separated by a narrow illegal region. Since there is only a narrow strip separating them, it is possible that a minor change might cause the legal regions to merge.
- *Short boundary segments.* Such segments indicate a fleeting contact between the two elements of the kinematic pair. This contact is also potentially sensitive to small changes in the geometry. This is shown in Figure 4-6(b).
- *Projections.* Narrow projections are often seen in the legal or illegal region (Figure 4-6(c)). They correspond to drastic changes in the nature of contact and are hence good candidates for further investigation.
- *Point Contacts.* Boundaries of the c-space that correspond to point contacts usually require some examination as they are prone to excessive wear and breakages; good design practice dictates that they be avoided

Once these areas are identified, the perturbation operators must be applied to check if any drastic changes have occurred. Currently, we have identified only two major classes of perturbations,

1. *Tolerance Variation.* Selected dimensions of the device are changed and the c-space recomputed to check if undesirable changes have occurred. Mechanisms have a large number of dimensions and therefore exhaustive enumeration is impractical,so it is necessary to have a logical basis for this decision. The strategy we use is to concentrate on potential trouble spots of



Narrowly Separated Regions

Short Boundary Segments

Narrow Projections

Figure 4-6: Interesting C-Space Features

the c-space that have been identified using the heuristics listed earlier. If a constraint description of the c-space is available, then the dimensions that appear directly in the offending constraint are the best candidates for perturbation. Determination of the candidate dimensions is a more difficult when the bitmap version of the c-space is in use as it requires a partial re-simulation to determine the intersecting edges. The heuristics are not guaranteed to yield all the sensitive dimensions because active constraint subsume constraints involving other variables. Therefore, better heuristics for selection of candidate dimensions need to be determined.

2. *Wear Perturbation.* Abrasive wear is characterized by a progressive growth of the legal region of the c-space. Hence, a general wear test is to simply grow the legal region uniformly. However, object features coming into contact during the duty cycle of the mechanism are subject to greater wear, as are points or places where there is a sharp change in the nature of motion of the objects. Hence, the legal region can be grown at differential rates to give a more realistic picture of the wear patterns. Once again, if undesirable effects occur, it is possible to locate the constraints, and consequently the geometric features that are responsible.

Perturbation operators affect c-space properties such as topology but can also affect paths in the c-space. A topology change in the c-space may be acceptable if the path representing the duty cycle of the mechanism is unaffected.

4.3.3 Abstraction Operators

It is often necessary to compare c-spaces. This may be because the designer wants to know how well a mechanism meets the design specification or simply to assess the magnitude of changes that have occurred in the c-space. Abstraction operators hide unimportant details and are very useful in this context. These operators have been identified by Joskovicz [21] and, because of their relevance, are briefly reviewed here. The reader is referred to the original report for more detail on the definition of these operators.

- *Linearization.*This operator simplifies non-linear boundaries of the c-space by replacing them with piece-wise linear approximations. An important issue in implementing this operator is that linearization must not change the topology of the c-space. For example, two non-linear non-intersecting boundaries of the c-space must remain non-intersecting when they are linearized.
- *Qualitative* Abstraction. This operator merges adjacent c-space boundaries and creates "qualitative configuration space boundaries". The only information retained after this operator is applied is the nature of the motion parameter relation(Figure 4-4). This can be checked using the query operators that were defined earlier.
- *Gap Closing* Abstraction. A major difference between ideal and real mechanisms is play or backlash. A real mechanism does not have any perfect fits and there must be some clearances available for the device to be assemblable. Play, clearances and backlash are seen in the c-space of a mechanism as narrow channels of legal placements. The gap-closing operator merges the boundaries of these two-dimensional channels into a line.
- *Periodicity* Abstraction. A device such as a gear pair exhibits periodic behavior because it has a number of identical teeth. To capture the behavior of mechanisms that exhibit such periodicity, it is necessary to know the c-space of the smallest functional unit that can be repeated to get the entire mechanism. The periodicity abstraction finds patterns of repeated behavior and yields a more compact description of device behavior. It is computationally expensive as it involves searching for graph isomorphisms in the region diagram.

The order of application of these abstraction operators can strongly affect the simplified description that is produced. Joskovicz has provided a ordering of these operators that attempts to rapidly discard unreachable behaviors and to rapidly reduce the number of regions. The abstraction operators are thus applied in increasing order of coarseness,*i.e.* linearization, qualitative abstraction, gap-closing and periodicity.

4.4 Composition

Most devices used in practice have more than two degrees of freedom. Hence, to obtain a global view of the mechanism, it is necessary to construct the c-space of the entire mechanism and not just the c-spaces of individual kinematic pairs.Single step computation of a high dimensional c-spaces, though theoretically possible, is too computationally intensive to be applied. An alternative method is to calculate a series of 2-D c-spaces and compose them to yield the c-space of the entire mechanism.

A composition operator has been defined by Joskovicz as follows,

The composition of two sub-sets of Cspace(M_1) and Cspace(M_2), $S_1 \subseteq$ Cspace(M_1) and $S_2 \subseteq$ Cspace(M_2), is a subset of Cspace($M_1 \otimes M_2$) defined as :

$$S_1 \otimes S_2 = \{p \in \text{Cspace}(M_1 \otimes M_2) \mid \text{projection}(M_1,p) \in S_1 \wedge \text{projection}(M_2,p) \in S_2\}$$

where projection(M_1,p) is the projection of the point p in Cspace($M_1 \otimes M_2$) to a point in Cspace(M_1).

Using this definition, the c-space of any mechanism with *n* links can be formed by composing the c-spaces of all its *n*(*n*-1)/2 object pairs. In most mechanisms, the components have one or two degrees of freedom and hence the c-spaces of object pairs can be calculated without running into unmanageable complexity. The problem is further simplified by the fact that the c-spaces of all objects that do not come into contact need not be calculated.

Unfortunately, this definition is not procedural, *i.e.* even if the component c-spaces of a mechanism are provided, it will not be possible to calculate the overall c-space using this definition. Joskovicz has provided a procedural definition applicable to the reduced class of fixed axis mechanisms. Unfortunately, these constitute only a small fraction of the mechanisms in use today; examination of a representative set of mechanisms [33] shows that most of them involve moving axes. For example, to analyze a simple four bar linkage it is necessary to compute a five dimensional configuration space. (The input and output links of the mechanism have a single degree of freedom each and the connecting link has three degrees of freedom.) The inapplicability of the c-space technique to situations like this one is a serious flaw and ways of overcoming it must be investigated and if c-space analysis is ever to be used in practice.

In special cases it is possible to cascade c-spaces [2]. For example, consider a gear set involving three gears A, B and C. The input motion is applied to A and transmitted through B to the output gear C. If the c-spaces of the gear pairs (A,B) and (B,C) are calculated, it is possible to use the common parameter B to calculate the c-space of the pair (A,C). It is easy provide a procedural definition for this and a simple case is presented by Bourne *et al* [2].

The lack of a procedural definition for composition is a significant drawback to the c-space approach. Even if such an operator was available, it is not possible to visualize higher dimensional c-spaces and this restricts their utility. However, this does not prevent 2-D spaces from being a useful tool for designers.

4.5 Examples

In this section, we present a series of examples that illustrate the application of the c-space analysis to problems in design and manufacturing. Special emphasis is laid on detecting tolerance related problems.

4.5.1 Example 1: Circuit Breaker

This example illustrates the use of c-spaces as an aid for designers. The artifact to be designed is a circuit breaker that uses a bi-metallic strip. The basic principle is that the electrical circuit is maintained by contact between two links, one of which is attached to a bi-metallic strip. When the temperature rises, the bi-metallic strip deflects, pulling the links apart and breaking the circuit. This state must be maintained until the device is manually reset.

A typical design is shown in Figure 4-7. The electrical circuit is completed by contact between the two links **L1** and **L2**. **L1** is connected to the bi-metallic strip and can translate in the **X** direction and **L2** rotates about **O**. The angle of rotation is denoted as Θ. A torsion spring applies a constant clockwise torque to **L2**.

Some constraints to be satisfied are as follows,

1. The device must break the contact only when an appreciable upward movement of L1 occurs.
2. The contact must be broken cleanly to prevent sparking, *i.e.* the contact must either occur over an appreciable area or not at all; point contacts are to be avoided.

We now present a scenario in which the designer starts from an initial rough design and modifies it to satisfy the behavioral specifications by reasoning about its c-space. The first design attempt is shown in Figure 4-7(a). The translational axis of **L1** passes through the center of rotation of **L2**. The c-space for this mechanism is shown in Figure 4-7(b).(The ranges of Θ and **X** have been restricted to [0,180] and [0,Xm] respectively). The initial position of the mechanism is marked **a**. The spring loading of **L2** ensures that it moves to the maximum value of Θ while traveling on a line parallel to the Θ axis and without passing through any illegal regions. If no such line exists, movement occurs along boundaries of the legal region that allow an increase in Θ. For this design, the initial position **a** is at the bottom of the notch. Hence as **X** increases, motion occurs along the segment **ab** and then, under the action of the spring, along **bc**.

The path followed by the mechanism while breaking the contact is **abc**. This suffers from the drawbacks that only a marginal upward movement of **L1** is requried for the circuit to be broken. Hence, the mechanism is extremely sensitive to the relative vertical locations of **L1** and **L2**. A small change in the vertical position of **L1** can cause the contact to be broken. When wear occurs, the features corresponding to the sharp point **b** in the c-space will become rounded and make it easier for the mechanism to slip. In addition, the segment **ab** corresponds to a vertex-edge contact constraint and will induce sparking. As only the ends of L1 and **L2** are utilized, the contact area is quite low.

The next version of the design is shown in Figure 4-7(c). The axis of the translating link has been shifted to the right and the c-space recomputed. It can be seen that the design change has produced a significant change in the shape of the c-space. The initial state of the mechanism has been located and once again marked as **a**. In this design a much larger increase in the value of the **X** is required before the spring can push Θ to it's extreme value. This corresponds to the segment **ab** in Figure 4-7(c). This solves the problem of sensitivity to vertical displacements but the segment **bc** marked in the figure is still caused by a point contact and will cause sparking. As the circuit is broken **L2** reaches point **d** along **cd**. When the device cools off, **X** decreases and state **e** is reached. It is from this position that the device must be reset. One possible path for this, **efcba** is shown in Figure 4-7(c).

To rectify the sparking problem, a new design with the chamfered ends shown in Figure 4-7(d) is proposed. The path followed during the duty cycle of the mechanism is **abc**.

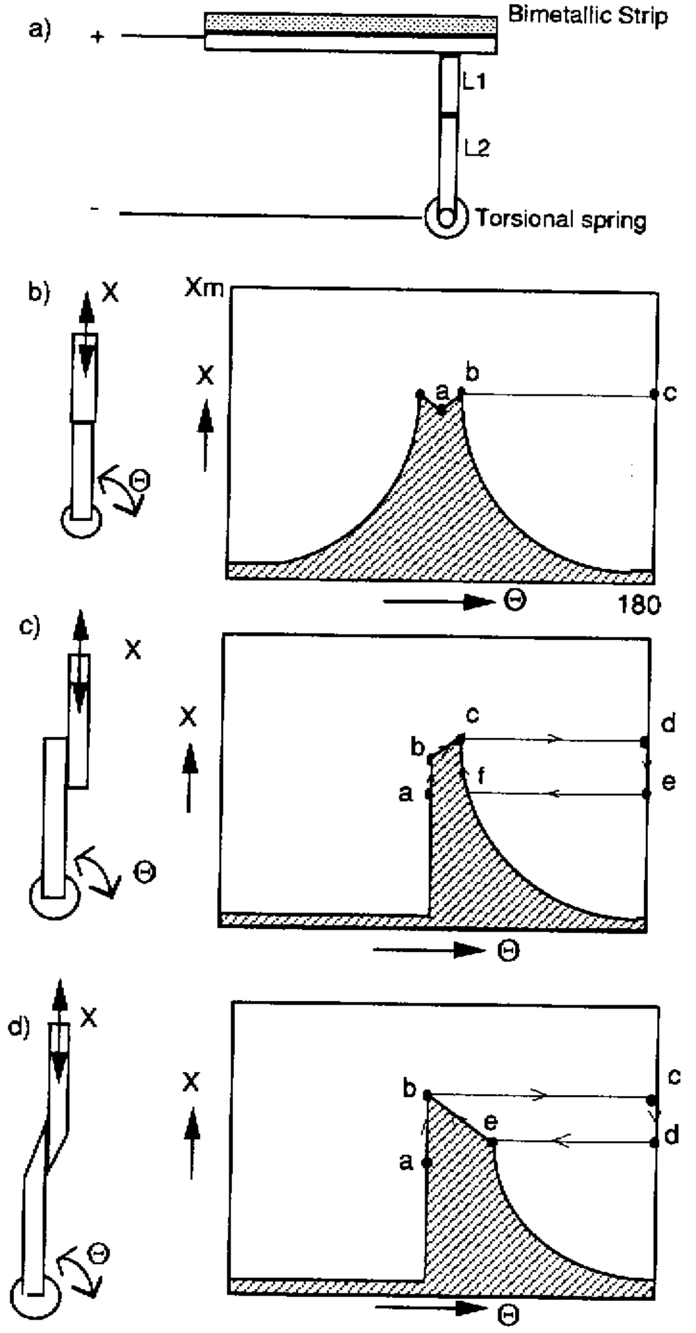


Figure 4-7: Example 1: Circuit Breaker

It can be seen that the state first moves along the vertical edge **ab** in Figure 4-7(d). This corresponds to face contact between the two links. When the vertex at the top of that edge is reached, the contact is abruptly broken with only a brief point contact occurring. Hence the mechanism is seen to satisfy both of the design requirements. The path followed when the circuit is broken is **abc**. A possible reset path **deba** is also shown.

The critical tolerances on the device can now be determined. The parameters considered are the thickness of the two links and the positioning of the axis of the translating links. When these variations are made and the c-space recalculated, a drastic change in the shape of the c-space is seen. The new c-space is similar to that of one of the rejected designs. The reason for this is that the alignment of the links has a strong effect on the behavior of the device. It is therefore possible to infer that the dimensions affecting alignment are important for the functioning of the device.

To investigate the effects of wear on the device, the entire illegal region of the c-space is slightly reduced (this corresponds to a widening of the legal region of the configuration space) and all sharp corners are rounded out. One effect on the mechanism is that the increase in X required to trigger the device is reduced. The original sharp change in the contact state at **b** has now become a brief segment of point contact. This can lead the designer to recommend that the tip of links 1 and 2 be made harder and more wear resistant and that an adjustment device be provided for the rest position of link 1.

This example has shown some of the important uses of the c-space in troubleshooting a design. We were able to observe a major change in the shape of the c-space as the result of a geometric change, and minor changes as the result of variation in dimensions of parts and routine wear. A desired path was defined and the c-space constraints used to evaluate the suitability of the path.

4.5.2 Example 2: Window Regulator Mechanism

A mechanism used in automobiles to raise and lower windows is shown in Figure 4-8. This mechanism has been used as a design case study by the Engineering Design Research Center, Carnegie Mellon University. It consists of a small gear (gear 1) rotated by a handle (not shown) and meshed with a partially geared segment (gear 2) that is linked to the bottom

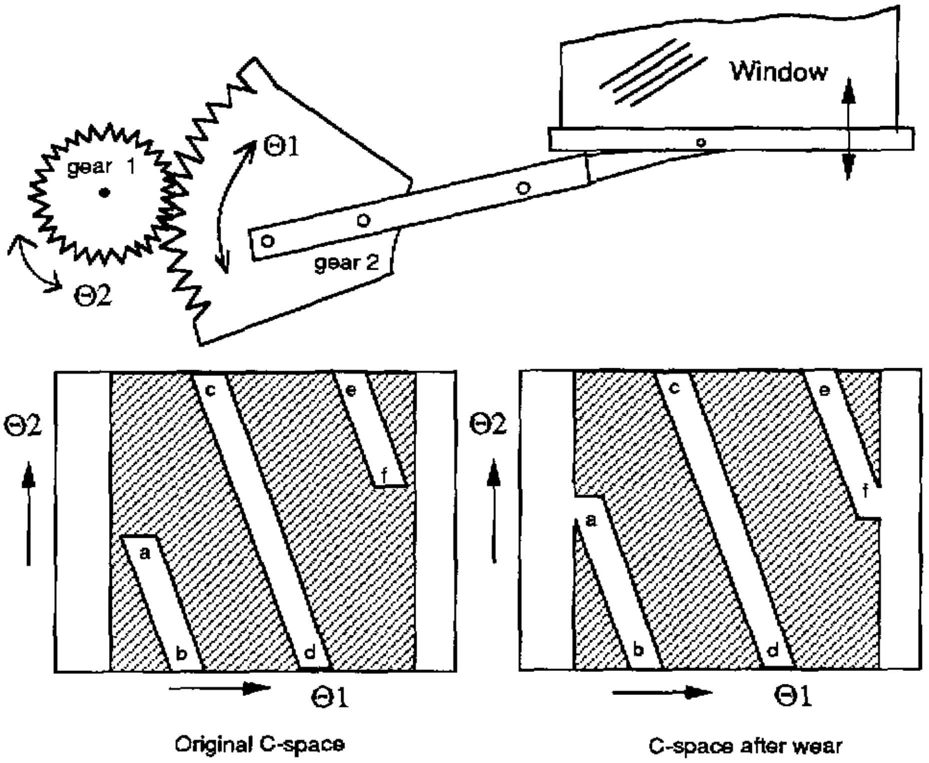


Figure 4-8: Example 2 : Window Regulator Mechanism

of the window pane. When the handle is turned, gear 1 rotates and causes gear 2 to raise the window.

This example examines the effects of wear on positional parameters. The operation of the mechanism involves passing sequentially through the states **a**,**b**,**c**,**d**,**e** and **f** and repeating this sequence in the reverse order. Parts of the mechanism that are prone to wear are identified by two heuristics. These are follows,

1. Wear occurs along all the boundaries of the region in which the motion is occurring. This is seen as a progressive widening of the unshaded areas of the original c-space shown in Figure 4-8.
2. The wear is more severe if there are sharp changes of slope in the boundaries

of the c-space. This would occur in the regions labeled **a** and **f** in the c-space Figure 4-8.

Once these regions are identified, the corresponding features of the device are modified to account for the wear. The c-space must then be recomputed to confirm that no changes in the qualitative behavior have occurred as a result of the modification.

The effect of wear is shown by the c-space drawn after wear has occurred. Two, initially distinct, unshaded regions of the c-space have merged into a single connected region. This shows that if wear occurs, gear 2 will be able to rotate freely and the device will no longer raise the window. As the c-space boundaries responsible for this change are known, the corresponding geometric features on the device can also be identified. Steps may then be taken to make those particular areas of the part thicker or more wear resistant.

4.5.3 Example 3: Self-Positioning Mechanisms

This example illustrates the steps involved in the design of a simple positioning device and how the c-space can be used to improve the design. The mechanism has two parts, the lower part is essentially a frame and the upper part has a spring loaded locating pin. The design objective is to create a device that will automatically position the two parts with respect to each other, in both horizontal and vertical directions.

Figure 4-9(a) illustrates an initial design and its c-space. The spring loading of the pin is manifested in the c-space by a tendency in the -Y direction. The path followed by the mechanism during its duty cycle is marked by arrows. Two problems can be diagnosed from this c-space,

1. The mechanism will not slide into the the locked position unless it is quite accurately positioned to begin with. This can be verified using a path generation algorithm.
2. The momentum gathered by the pin on its vertical descent into the slot may lead to a strong impact when the mechanism locks.

In an attempt to solve the problem, the slot is replaced by a v-notch (shown in Part (b)). This solves the problems with the first design but introduces two new ones,namely

1. To achieve the same vertical distance between upper and lower parts as the first design, the v-notch has to be much deeper.
2. The path followed during locking (and the final position) corresponds to point contact.

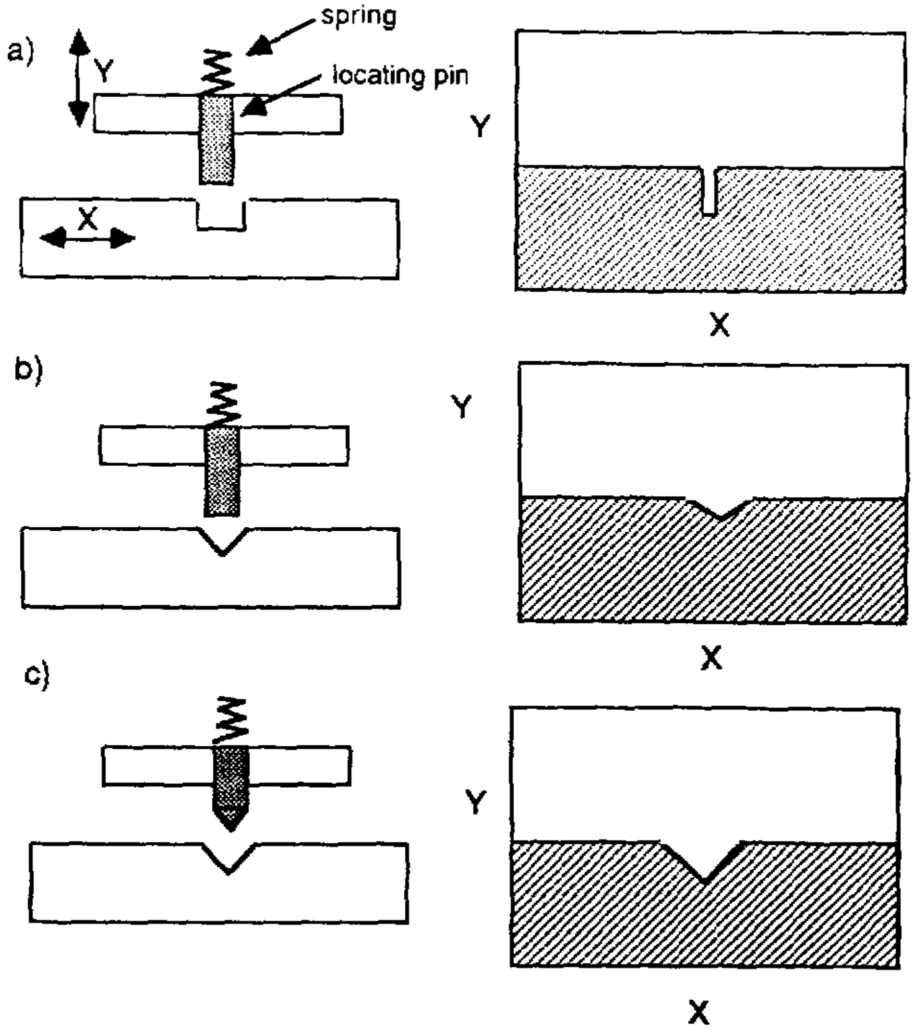


Figure 4-9: Example 3: Self Positioning Mechanism

calculation.

The previous calculation would not be valid if the cam and valve could touch each other directly because that would introduce a new interaction to be accounted for. It is interesting to note that modern engines have the cam mounted directly on the valve (overhead cams), thus achieving the same input-output relation without the play caused by multiple intervening components.

This example is very similar to that of a multi-gear set used earlier to explain composition of

The final design, shown in Figure 4-9(c) also includes a chamfer on the pin. The resulting c-space shows none of the disadvantages of the previous two designs.

If the problem in this example was merely to choose between the three designs shown, identical criteria and reasoning could have been used arrive at the same choice for the best design. The advantage of using the c-space methodology is that most of the steps described in this example could conceivably be automated. Hence, once the operators and some heuristics are encoded, we would have a powerful tool that would enable even novice designers to identify potential trouble spots and take corrective action. It should be noted that c-space operators and tools are used only to identify problems and do not provide means of correcting them. We feel at this stage that this is best left to the expertise of the designer.

4.5.4 Example 4: Automobile Valve Train

Under certain conditions, c-spaces can be cascaded with useful consequences. We use this example to explain the required conditions and present some results of cascading c-spaces. Figure 4-10 shows a valve mechanism used in four-stroke internal combustion engines. The mechanism has four major components: cam, pushrod, rocker and valve, and these are marked in the figure. The cam is rotated by the engine and causes the pushrod to oscillate linearly. At the top of its travel, the pushrod displaces the rocker arm which briefly opens the valve. When the pushrod descends, the return spring closes the valve.

The c-spaces of the three major kinematic pairs involved are labelled 1,2 and 3 in Figure4-10. Using these c-spaces we can to deduce the position of the valve for each position of the input cam and construct the c-space of the cam and valve degrees of freedom directly. We now illustrate this calculation for a sample value of T1. Since the pushrod and tghe cam must be in contact, the value of X1 is found by noting where the vertical line through T1 in c-space 1 intersects the boundary. We can now look up the value of T2 corresponding to this in c-space 2. Next, the value of X2 is read off from c-space 3. Hence, it is possible to deduce the value of X2 corresponding to a particular value of X1 by concatenating c-spaces. If a more general relation is required, the all the boundaries of c-space 4 can be calculated and used directly instead of going through three itermediate c-spaces each time. Note that all the inaccuracies and play get automatically included in the c-spaces. The common property of these two mechanisms and others that can be analyzed by cascading is that they consist of a chain of components, each of which is only in contact with the next and the previous links in the chain. This property enables us to neglect a number of c-spaces that would have to be calculated in the general case. For a chain of n elements, only *n*-1 c-spaces need be calculated whereas for a general mechanism, *n*(*n*-1)/2 pairwise c-spaces would be required.

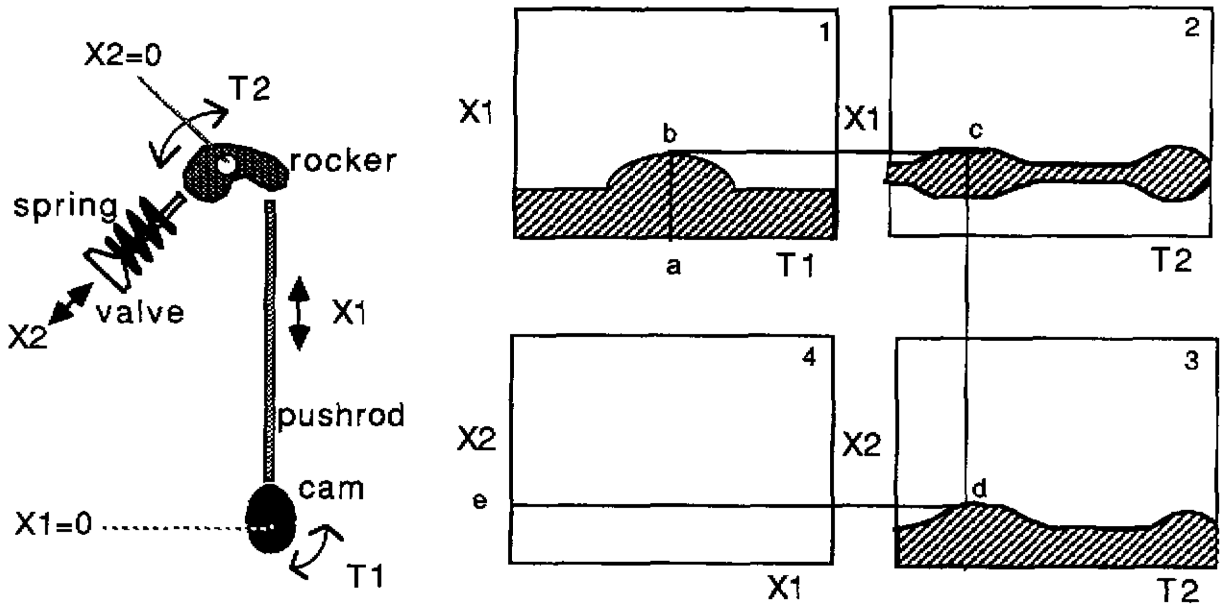


Figure 4-10: Example 4: Automobile Valve Train

4.6 Summary

We have introduced a number of operators that can be applied in the c-space to perform useful inferences. These operators can help designers gain insight into aspects of the design that they would otherwise have missed. The operators are applicable both on exact (constraint) and approximate (bitmap) representations of the c-space. In the next chapter, we look at ways of rapidly calculating the c-space of a kinematic pair and algorithms for implementing each of the operators.

We have presented several examples of using the c-space for analysis and design of mechanisms. These examples show the potential for using c-spaces as the basis of an automated diagnostic aid for designers.

5. Implementation Issues

5.1 Introduction

For the mode of analysis used to analyze the examples in the last chapter to be practical, the following capabilities are required,

1. Support for topological and geometric descriptors of the c-space.
2. Rapid generation of the c-space from the geometry and motion specification.
3. Heuristics for detection and explanation of undesirable changes in the c-space.

We now describe a computer implementation that provides some of the above functionality.

5.2 Representation

The topology and geometry of the c-space are most relevant to our analysis. The topology of the c-space gives a qualitative description of the behavior of the device. For example, the number of distinct legal regions of the c-space indicates the number of distinct classes of motions that the mechanism can exhibit. Therefore, a change in the topology indicates a fundamental change in the behavior, as seen in the window regulator mechanism presented earlier (Figure 4-8).

Though topology and high level geometric parameters provide a rapid and inexpensive way of specifying a class of behaviors, they are not sufficient for a detailed analysis. For example, the designer may want to know if a mechanism parameter can ever attain a specific value. This information is not available from a topological description and hence it is necessary to store the exact geometry of the c-space to answer such questions.

We plan to use the Noodles solid modeling system for this task (see Gursoz and Prinz [17] for details on Noodles). It provides all the functions of a standard boundary representation (b-rep) based modeler. It also has the additional capability of handling other, normally illegal, features such as dangling edges or faces and isolated nodes in a uniform manner. This is possible because it explicitly extracts and stores detailed information about the connectivity of the different regions of the model. The topology is stored explicitly in the model and the user can query the system for this information. This is useful for detection and diagnosis of topological changes. Since the b-rep scheme is used, other information such as number of c-space boundaries *etc* is also easily available. Additionally, though a minor convenience, the same representation is used for objects and c-spaces.

5.3 A New Approach to C-space Calculation

Current approaches to this problem attempt to solve the problem symbolically. Knowing the equations for the boundaries of the two objects and their motion paths, it is possible to formulate constraints governing possible contacts between the two bodies and to solve them using symbolic algebra. The computation required for this problem rises with the complexity of the object boundaries and the number of degrees of freedom. Hence, even for two degree of freedom cases, the process must be expedited by restricting the scope to fixed-axis mechanisms and using simple heuristics to avoid complex calculation. This has been suggested by Joskovicz [20].

The approach that we apply for two dimensional c-spaces is to calculate the configuration space by simulation [3]. Each object is represented as a bitmap, and motion is simulated by changing the location of these objects in a bit plane. A simple bisection algorithm is used to determine the precise point of contact. This approach has certain advantages including,

1. It is possible to obtain rapid results using special purpose hardware. For two dimensional cases, the computation required to calculate the c-space at a given resolution is independent of the complexity of the object.
2. The granularity of the simulation can be controlled so that results of the desired precision are obtained. The simulation may be limited to specific orientations of the components which are of interest to the designer.
3. Symmetry of the objects can be exploited to reduce the amount of computation. For example, analysis of a gear pair is reduced to simulation of contact between a single pair of teeth.
4. Intelligent strategies can be used to update previously calculated spaces rather than recomputing them entirely. This is possible because we can keep track of all the c-space boundaries affected by a particular geometric feature. If that feature is modified we only need to re-simulate all the contact points involving that feature; all others remain unaffected.

It is not possible to work with bitmaps when the c-space has a dimension exceeding two. Intersection of full three dimensional parts must be performed using the solid modeler. This extension of the simulation approach to the domain of three dimensional objects seems reasonable. This is because translation and rotation are relatively inexpensive operations in solid modeling when compared to computation of intersections. Our approach only requires that interference between objects be detected and this can be performed rapidly. However strategies to identify the most interesting degrees of freedom of a part are required.

The major problem encountered in applying this approach has been choosing the correct resolution to capture all the interesting features of a c-space. When a low resolution is used, small changes in the geometry of the object will not be reflected by changes in the c-space. It is therefore necessary to develop strategies to estimate the proper resolution to be used. This involves deciding whether a geometric change is likely to be significant and then choosing a proper resolution to reveal the change.

The analytical approach for calculating c-spaces provides detailed equations of the c-space boundaries in terms of the mechanism dimensions. Hence, it is possible to evaluate the relative effect of different dimensions on a particular boundary of the c-space. Joskovicz [20] creates a "configuration space boundary map" to relate geometry and c-space constraints. This records, for each boundary segment, information about the geometric features that produced it and c-space changes that will occur if these features are removed. Availability of these equations also facilitates simple analyses using linearized forms of constraints. Using simulation, it is possible to record high level information about the dependencies of c-space boundaries on features. However, the exact algebraic relations are not retained. As a result, perturbation analysis must also be done by simulation. To overcome this problem, we may consider calculating exact boundary equations in special situations.

The speed with which c-spaces can be updated and recomputed is a critical factor in making a c-space based system usable. Thought the simulation approach yields rapid results, it is that some user-interaction may be necessary for construction of c-spaces in complex cases. This could yield substantial improvements in speed since inferences that seem trivial to the user often require considerable computation. However as it may limit the use of the analysis tool, this trade-off must be carefully evaluated.

The current system, CSpace v1.0, is implemented on Personal Iris workstations. The code is written in C and the graphics is done under X Windows. The current version has the capability of handling two dimensional objects which may be input by the user in a variety of formats : as bitmap objects, as polygons or drawn interactively on the screen using the mouse. The user can also set the resolution at which the c-space simulation is performed.

The results are output to a file and and are also displayed on the screen. The basic implementation relies heavily on the completely transportable X-Windows graphics functions and is quite slow. By creating a no-graphics version of the program, we achieved a speedup of an order of magnitude. When the algorithm is implemented on special purpose hardware we expect another similar improvement. Version 1.0 uses only locally defined graphics entities and the interface with the Noodles solid modeler has not been completed. We are awaiting a new, more efficient version of Noodles which will be used as the basis for geometry representation.

There is still considerable doubt as to whether the simulation approach to c-space calculation will be practical. Though the approximate c-spaces can be calculated extremely fast, the problem of choosing the proper resolution still exists. When using the bitmap approach, the c-space is calculated by detection of overlap; the details of the overlap are considered unimportant. However, many of the operators defined in the last chapter use the c-space boundary map. For example, if a c-space boundary interferes with a desired path, it is important to be able to identify the geometric features that control the boundary. As a result, the bitmap algorithm must be modified to calculate this information each time an overlap is detected. For two objects with n boundary features each, this requires that n^2 intersections be checked. This number can be reduced by using information about the position and geometry of the objects involved but still remains significant. Searching for intersections reduces the speed advantage that the bitmap algorithm originally had and also makes it dependent on object complexity. These problems may make it necessary to use other representations to supplement or even replace the current one and this possibility is currently under consideration.

5.4 Path Generation Algorithms

It is also possible to simulate the behavior of a device based on its c-space and a description of the prevailing tendencies. This can be achieved using simple, fast algorithms that operate on the bitmap versions of the c-space. We present here a simple algorithm that describes motion under a +X tendency of medium strength. The algorithm, shown in Figure 5-1, uses a grid with nine squares in it. The path generated by the algorithm is the locus of the center of this grid as it moves from left to right. Based on where it is in the c-space, some of the squares of the grid may be in the no-go region. The number of squares in the no-go region is used to estimate the nature of the c-space boundary and to decide how the motion parameters will vary. Specific criteria for deciding where to move next are shown in the Figure 5-1. The algorithm shown assumes a moderate tendency in the X direction and hence can overcome only gentle slopes in the c-space boundaries. It can be modified to handle a stronger X tendency by changing the termination condition suitably. The resolution of the tendency variable can be increased by increasing the number of squares in the grid.

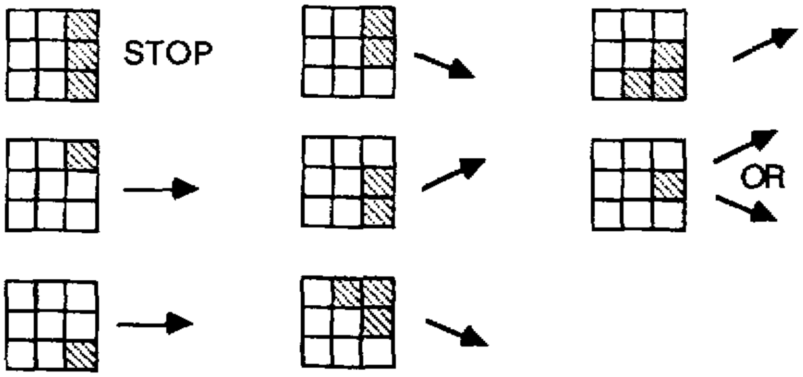


Figure 5-1: A simple path generation algorithm

C-space path generation algorithms provide a reliable way of simulating the behavior of a device. If linearized or otherwise abstracted versions of the c-space are used, we can deduce the qualitative differential equations of the system. The algorithms can be made more reliable by adding features such as momentum conservation. This feature would allow the path to ascend even steep c-space boundaries if considerable momentum had been built up in previous stages of the motion.

The path generation algorithm can provide the basis for simulating a variety of behaviors exhibited by a mechanism. The c-space need be calculated only once and, using this information, paths corresponding to various initial conditions and external tendencies can be calculated. The entire process can be automated to yield a powerful visualization tool.

5.5 Conventional Tolerancing Utilities

It is useful to have available some of the standard cost-tolerance models and associated optimization techniques. There are often situations where these models are either directly applicable or their application can yield a useful insight into the problem. Hence, we have included them as part of the tolerancing tools provided.

We have created a basic tolerance optimization facility using GAMS, a generalized algebraic manipulation system [5, 23]. GAMS provides a range of optimization capabilites including linear programming, non-linear programming and mixed-integer-non-linear programming. GAMS uses its own input language but our software eliminates the need for the user to know this language. Our package takes input in equation form and converts it to a form suitable for GAMS. This data is fed to GAMS and the results are retrieved and displayed.

The tolerancing package requires the user to enter a set of design equations along with appropriate bounds for all the variables. The user must also choose a suitable cost-tolerance model for each variable. The values for each variable that result in a minimum cost tolerance allocation are calculated using GAMS and displayed on the screen. The packages also provides additional utilities such as proportional scaling, symbolic or numeric sensitivity analysis based on linear approximations and functions for statistical summation of tolerance variables. The package is written in Common Lisp and should be run from within the EMACS editing environment.

5.6 Summary

We have introduced a novel method of calculating c-spaces. This is based on simulation and has several advantages over the analytical calculation approach. One major benefit is the speed with which the c-spaces can be calculated. This allows the designer to freely explore the effect of geometric changes on the c-space. The operators introduced in the last chapter can also be easily applied to the c-spaces calculated by simulation. The major drawback of the approach appears to be that choosing an appropriate resolution to reveal all important details is extremely difficult.

A simple package that provides some conventional tolerance optimization facilities is also described. This package allows the designer to achieve guaranteed optimal solutions if the problem can be described in terms of non-linear equalities and inequalities. This is very similar, in principle, to some of the commercially available tolerance optimization packages.

6. Conclusions

6.1 Role of Tolerances in Design

A satisfactory design must satisfy constraints on various properties such as mass, kinematic behavior, heat transfer characteristic *etc.* The only means of controlling these properties that designers usually possess are modifications in the geometry of the design. Specifications on the behavior of the design must be translated into constraints on geometry, or tolerances. The variability inherent in even a simple nominal geometry is extremely large and many different types of controls are necessary (form, orientation *etc*). As each of these controls has an impact on the manufacturability of the design, they must be kept to a minimum. This is possible, because in modern manufacturing it is safe to assume that all unspecified geometric properties will be produced to reasonable tolerances (*e.g.* 0.01 *in* for linear dimensions). In addition, it is also desirable that the tolerances be justified and documented. Existing engineering representations must be refined and augmented to perform this task.

Typically, only a small subset of the geometric properties is pertinent to the desired behavior. For example, the surface roughness does not affect the function of a machine foundation as it affects a convective cooling surface. The problem is to determine which of the geometric properties are significant and to exercise explicit control only over these properties. Good designers are able to identify these critical properties of the design through experience and insight. However, inexperienced designers must be provided with tools that assist them in this task. In this report, we have laid the foundation for a tool that provides some the functionality described above in the limited realm of designing for kinematic behavior.

6.2 Using C-spaces in Design

We have adopted configuration spaces as a representation of kinematic behavior and shown how it can be used to draw useful conclusions about variations in geometry. This draws on earlier work by Joskovicz and Faltings and extends it to explicitly reason about tolerances. We have reviewed the theory of c-spaces and proposed a set of operators that can be used for diagnosing mechanisms based on their c-space description. We use simple heuristics to identify potentially important dimensions and describe ways of verifying this by perturbation of the c-space. A new method of rapidly calculating the c-space for fixed-axis mechanisms has been implemented. This algorithm functions at variable resolution and its complexity is independent of the complexity of the objects involved. Several useful operators that can be used to extract useful information from the c-space have also been described and partially implemented. In summary, we believe that we have shown how the c-space representation can be used to partially automate reasoning about tolerances.

However, significant obstacles to using this technique still remain and it must remain the subject of ongoing research. Briefly, these problems are

1. Earlier work on using c-space for mechanism analysis assumes that most practical mechanisms are of the fixed-axis type. A study of handbooks and texts [33] on mechanisms indicates that this is not really true. Hence the limitation of being able to analyze only fixed axis mechanisms is quite a strong one

2. For succesful mechanism design, analysis has to be performed both at the global and local level. However, no procedure exists for composing the c-spaces of individual kinematic pairs to get the c-space of the entire mechanism. A procedural definition of composition exists only for fixed-axes mechanisms.

3. Existing techniques for c-space calculation are slow and require powerful algebraic computation systems like MACSYMA. The complexity increases with the boundaries of the objects and, as a result, is only manageable for two dimensional objects composed of arcs and lines. Some speedup is possible by taking advantages of properties like symmetry but this is insufficient.

6.3 Future Directions

We briefly describe two potential extensions of this work.

1. *Link with qualitative simulation.* We have described abstraction operators that reduce the detail in the c-space. After applying operators like linearization and qualitative abstraction, it should be possible to get the qualitative equations that describe a particular path in the c-space. Conversely, a set of qualitative differential equations can be used to construct paths or boundaries in c-space. This will provide a powerful capability for designers to successively refine designs and add more detail to them.

2. *Considering more general behavior.* The design process involves consideration of properties besides kinematic behavior. Each of these can is independent variable that must be controlled via geometry, material and other properties of the design under the explicit control of the designer. The concept of c-spaces can be generalized to parameter spaces where any properties of the design may be studied. Heuristics for manipulation of parameter spaces can then be developed to automate a number of design decisions.

During the course of this work, we have become increasingly aware of the of the real significance of tolerance specifications. It is necessary to develop a design methodology that carefully justifies and correctly documents each tolerance specification that is applied to a design. Designers must be provided with tools that enable them to explore the results of tolerance variations before the design is ever really built. This will help them develop an appreciation of the impact of tolerance specications and lead to a more efficient design and manufacturing cycle.

I. Appendix I

I.1 Introduction to ANSI-Y14.5M

This is intended to be a supplement to the complete ANSI-Y14.5M standard. We explain the basic principles and terminology used in the standard without getting into the finer details that an actual standards document must necessarily contain. Hence, issues such as placement and size of symbols are ignored and only their information content, relevant to the research presented in this report, is analyzed.

Each part consists of a set of *features*. Any edge, face or group of faces on a part may be regarded as a feature. Examples of features are walls, holes and slots. A correctly drafted drawing of a part must provide manufacturers with enough information to create each feature on the part. ANSI-Y14.5M provides a standard way of displaying this information on engineering drawings.

A feature may have up to five properties that must be defined before it can be produced. These properties are,

- Location
- Size
- Orientation
- Form
- Surface finish

Location, size and orientation must be defined in terms of *datum* features. When a planar surface is used as a datum, it is assumed to be a perfect surface and no variations in it are explicitly accounted for. In some cases, cylindrical datum features are also used. A cylindrical datum feature is associated with two theoretical planes intersecting at right angles on the datum axis. All non-datum features on a part are specified with respect to datum features. The datums to be used for locating the non-datum features are specified in feature control frames (FCF) associated with each non-datum feature. (The FCF is described in more detail later).

Figure I-1(a) shows two datum features, A and B, marked on a rectangular part. They are both plane faces and hence can be directly used for locating other features. Sometimes the entire surface of a feature cannot be used as a datum because of inherent irregularities in it. For example, a curved surface used to define a plane. In these situations, datum targets, which are positions, areas or lines on the irregular feature, are specified. To define a plane on a curved surface A, three datum targets A1,A2 and A3 are necessary and these are marked by the symbol X on the feature A. The symbol in Figure I-1(b) is associated with each datum target, with the appropriate target number in the lower half of the circle. If the datum target has a finite area, this is indicated in the upper half of the circle otherwise that half is left blank.

Hence, datums are established either by directly using regular features such as planar faces or by indicating sufficient datum targets on an irregular feature.

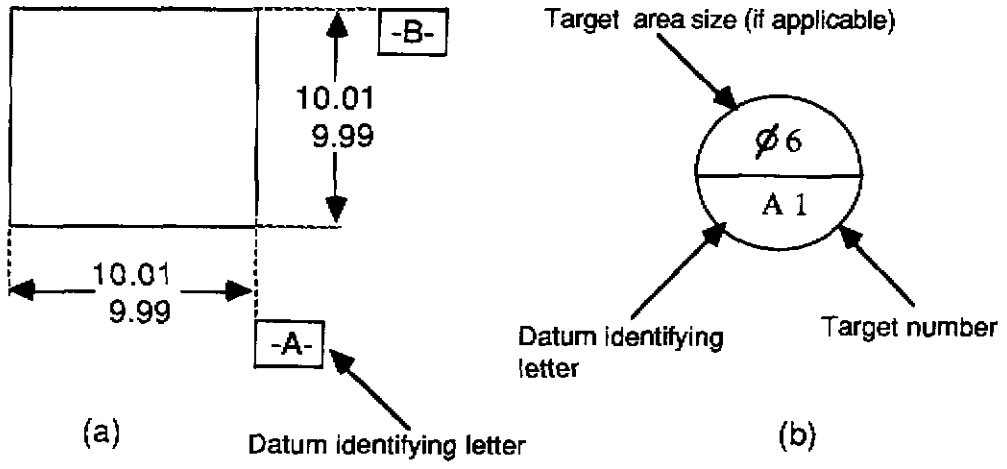


Figure I-1: Datum surface and target symbols

We now provide more detail on each of the feature properties and introduce the associated terminology.

I.1.1 Position

According to ANSI-Y14.5M

> A positional tolerance defines a zone within which the center, axis, or center plane of a feature of size is allowed to vary from true (theoretically exact) position.

Consider the example in Figure I-2 where a hole must be located on a plate. It is important to realize that we are considering only the *location* of the hole and not its *size* (diameter). Basic dimensions are used to locate the center of the hole from the two datum surfaces A and B. In part (a) of the figure, A and B are given +/- tolerances and this results in a rectangular tolerance zone. In part (b) a true position tolerance is specified in accordance with ANSI-Y14.5M and leads to a circular tolerance zone for the center of the hole. This is achieved by using a feature control frame (labeled FCF for purposes of illustration).

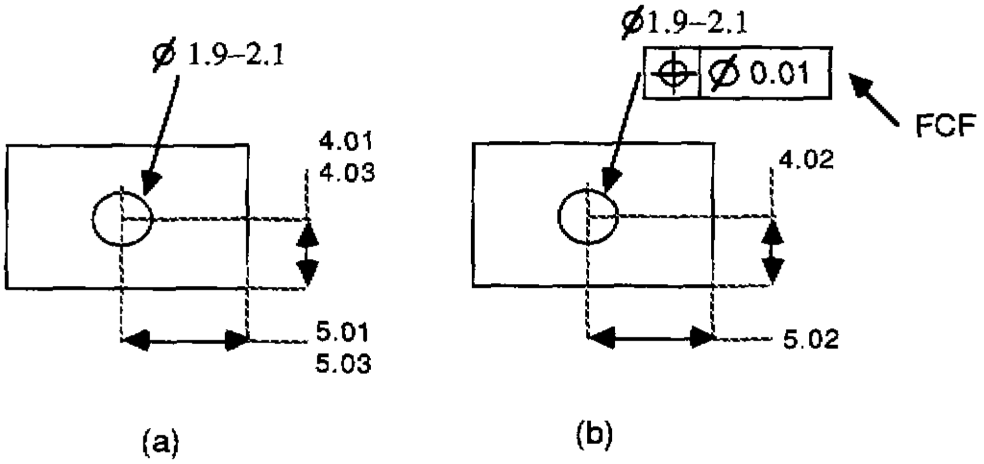


Figure I-2: Position tolerances in ANSI-Y14.5M

A feature control frame consists of at least two compartments. The leftmost compartment gives the geometric characteristic symbol which indicates the type of control being applied (flatness, straightness, position, parallelism *etc*). The next compartment contains a tolerance value which may optionally be preceded by the diameter symbol,Φ, and followed by a material condition symbol. An example is shown in Figure I-2(b). The material condition symbol indicates whether the specified tolerance is applicable at the upper limit of feature size (MMC), lower limit of feature size (LMC) or independent of feature size (RFS).

Complex FCFs may have upto three datum references, each with its own associated material condition letter. When multiple datums, referred to as primary, secondary and tertiary datums, are specified, they indicate the desired order in which datum references are to be made when the feature is located. An example of this is shown in Figure I-3. The FCF shown specifies the tolerance on the center of a hole, indicated by the icon in the leftmost compartment. The tolerance zone is a circle of diameter 0.01 centered at the nominal position. Three datum surfaces A, B and C (all RFS) are specified. There is additional information that can be entered in the FCF and the reader is referred to pages 30-33 of the ANSI-Y14.5M document for further details.

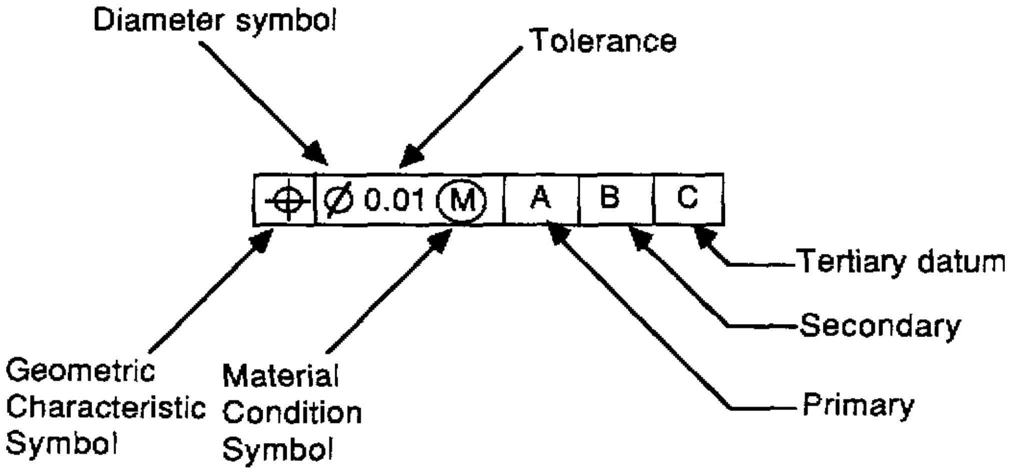


Figure I-3: Feature Control Frames

I.1.2 Size

Some features, such as holes, have a size attribute associated with them. For example, the diameter is the only size attribute of a hole and the maximum and minimum diameter are the limits this attribute.

Upper and lower bounds on the size of a feature can also be described in terms of the material condition. The size tolerances on features decide how much material may be safely removed during machining. Consider a hole with lower and upper bounds on its diameter. If the hole is manufactured to the maximum diameter, more material has to be cut away than if the lowest diameter is achieved. Hence, for holes, the lowest diameter correponds to the Maximum Material Condition (MMC) and the highest diameter to the Least Material Condition (LMC). For a shaft, the situation is reversed, *i.e.* maximum diameter corresponds to MMC and least diameter to LMC.

It is possible for a designer to use a material condition specification in conjunction with a tolerance to provide greater flexibility to a machinist while still ensuring that parts are assemblable. When locating a hole that is to mate with a shaft, errors in locating the center of the hole can be compensated for by increasing the diameter of the hole; the precision to which the center is to be located is highest when the hole is at its lower limit of size. Hence, the designer may specify a location tolerance that is quite tight if the hole is made to the smallest allowable size but may be progressively relaxed as the hole is made larger. This is an example of a tolerance specification made at the Maximum Material Condition or MMC. Tolerance specification may be made at Least Material Condition or LMC. If it is required that the feature be located to the same tolerance irrespective of its actual size, a Regardless of Feature Size or RFS tolerance specification is used. MMC, LMC and RFS are the three material condition specifications used in ANSI-Y14.5M. If no material condition is specified, RFS is assumed.

The freedom to vary location tolerances based on the actual size of a feature allows the designer to avoid specifying a blanket tolerance applicable to all feature sizes. By using the material condition specification, designers can provide the manufacturer with some flexibility to adjust depending on how individual parts turn out, but still be assured that the part will assemble or otherwise perform satisfactorily.

I.1.3 Form

Form tolerances control straightness, flatness, circularity, and cylindricity. It is customary to specify form and orientation tolerances when size and location tolerances do not provide sufficient control. For example, consider the situation in Figure I-4 which illustrates how specifying only size and position tolerances can lead to unexpected results. The lower half of the figure shows the case that satisfies the tolerance specification of the upper half. If the designers do not find such variations acceptable, they must further constrain the geometry by imposing a *straightness* tolerance. The reader may refer to the ANSI-Y14.5M document for more examples of cases where form tolerances become necessary.

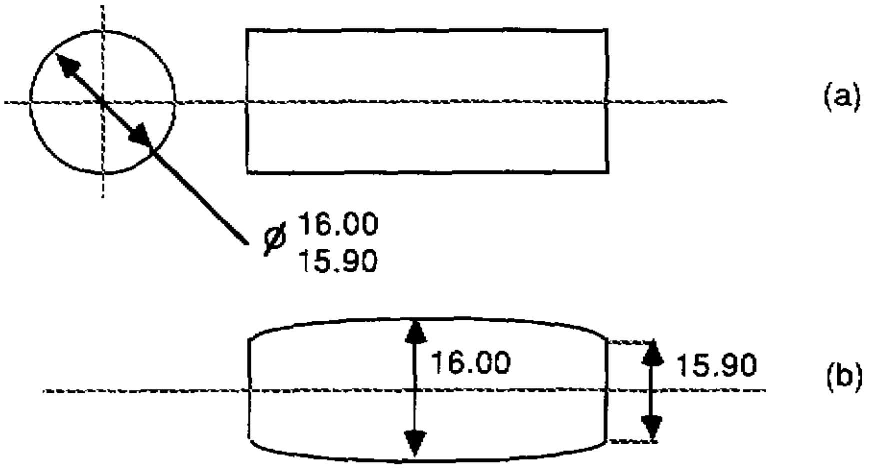


Figure I-4: Effect of pure size tolerance

Straightness is defined by ANSI-Y14.5M as

> "a condition where an element of a surface or an axis is a straight line. A straightness tolerance specifies a tolerance zone within which the considered element or axis must lie"

The document also contains similar definitions for *flatness*, *circularity* and *cylindricity*. Form tolerances are denoted by using a FCF with an appropriate geometric characteristic symbol followed by the tolerance and a metal condition symbol.

I.1.4 Orientation

Angularity, *parallelism* and *perpendicularity* are the commonly used orientation tolerances. They are also referred to as attitude tolerances. Orientation tolerances are also specified using FCFs that specify tolerances, material conditions and datum surfaces or axes. The Y14.5 document gives numerous example of orientation tolerances.

I.1.5 Surface Finish

Designers may specify a surface finish when it is critical to the function of the part. The ANSI-Y14.5M standard does not cover surface finish specifications; these are described in ANSI-Y14.36.

I.2 Conclusion

Tolerancing is a way of controlling the variation of geometry in relevant ways. Depending on the application, different types of geometric controls are necessary and a tolerancing standard provides standard ways of applying these controls and a set of figures and symbols for representing them on an engineering drawing.

We have briefly introduced the key features of the ANSI-Y14.5M dimensioning and tolerancing standard. The basic philosophy is to provide means of measuring the intrinsic properties of each feature on a part from convenient datum surfaces. This involves the use of datum and non-datum features, feature control frames, material conditions and various types of tolerances. The original document contains detailed definitions of all the terms used in this summary and explanations of their significance. It also presents hundreds of examples of their proper use in special and standard situations. Readers are urged to study these examples to improve their knowledge of this standard.

References

[1] American Society of Mechanical Engineers.
Dimensioning and Tolerancing Standard Y14.5M.
1982

[2] Bourne, D.B., Ramaswamy, R. and Navinchandra, D.
Tolerance Free Design for Manufacturing.
In *Proceedings of the Winter Annual Meeting of the ASME, 1989.* American Society of Mechanical Engineers, 1989.

[3] Bourne, D.B., Navinchandra, D and Ramaswamy, R.
Relating Tolerances and Kinematic Behavior.
In *4th International Conference on Applications of Artificial Intelligence in Engineering, Cambridge, U.K..* Computational Mechanics Institute, 1989.
Also published in Artificial Intelligence in Design, J.Gero (ed), Springer Verlag.

[4] Bralla, J.G.
Handbook of Product Design for Manufacturing.
McGraw-Hill Book Company, 1986.

[5] Brooke, A. et al.
GAMS- A Users Guide.
The Scientific Press, Redwood City, California, 1987.

[6] Chase, K.W. and Greenwood, W.H.
Design Issues in Mechanical Tolerance Analysis.
Manufacturing Review, 1988.

[7] Collins, J.A., Hagan, B.T. and Bratt, H.M.
The Failure Experience Matrix - A Useful Design Tool.
Transactions of the ASME, Aug, 1976.

[8] Crossley, E.
A Shorthand Route to Design Creativity.
Machine Design, April, 1980.

[9] de Kleer, J. and Brown, J.
A Qualitative Physics Based on Confluences.
Artifical Intelligence (24), 1984.

[10] Faltings, B.
A Theory of Qualitative Kinematics in Mechanisms.
Technical Report UIUCDCS-R86-1274, Department of Computer Science, University of Illinois at Urbana-Champaign, 1986.

[11] Faltings, B.
Qualitative Place Vocabularies for Mechanisms in Configuration Space.
Technical Report UIUCDCS-R-1360, Department of Computer Science, University of Illinois at Urbana-Champaign, 1987.

[12] Faltings, B.
Qualitative Kinematics of Mechanisms.
Technical Report No.88-01, AI laboratory, Department of Information,EPF Lausanne, 1988.

[13] Forbus, K.
Spatial and Qualitative Aspects of Reasoning about Motion.
In *Proceedings of the National Conference on AI.* 1980.

[14] Forbus, K.
A Study of Qualitative and Geometric Knowledge in Reasoning about Motion.
Technical Report AI TR 615, MIT, 1981.

[15] Forbus, K. et al.
Qualitative Kinematics: A Framework.
Technical Report UIUCDCS-R-87-1352, University of Illinois, 1987.

[16] Greenwood, W.H. and Chase, K.W.
Computer Aided Tolerance Selection: CATS.BYU User Guide.
1986

[17] Gursoz, E.L. and Prinz, F.B.
Node-based Representation of Non-manifold Surface Boundaries in Geometric Modeling.
Technical Report , Design for Manufacturability Laboratory, Engineering Design Research Center, Carnegie Mellon University, 1988.

[18] Hillyard, R.C. and Braid, I.C.
Analysis of Dimensions and Tolerances in Computer-Aided Mechanical Design.
Computer-Aided Design 10(3), May, 1978.

[19] Hoffmann, P.
Analysis of Tolerances and Process Inaccuracies in Discrete Part Manufacturing.
Computer Aided Design 14(2), March, 1982.

[20] Joskovicz, L.
Reasoning about Shape and Kinematic Function in Mechanical Devices.
Technical Report TR No. 402, Courant Institute of Mathematical Sciences, New York University, 1986.

[21] Joskovicz, L.
Mechanism Comparison and Classification for Redesign.
Technical Report RC 14573, IBM, April, 1989.

[22] Karolin, A.
Computer-Aided Tolerance Analysis.
In *Proceedings of the Autofact 6 Conference.* Society of Manufacturing Engineers, October, 1984.

[23] Murtagh, B.K. and Saunders, M.A.
MINOS 5.1 Users Guide.
Technical Report SOL 83-20R, Stanford University, 1988.

[24] Nielsen, P.
The Qualititive Statics of Rigid Bodies.
Technical Report No.1354, University of Illinois at Urbana-Champaign,Dept. of Computer Science, 1987.

[25] Peck, H.
Allocating Tolerances and Limits.
Longmans Green and Co., 1968.

[26] Lozano-Perez, T. and Wesley, M.
An Algorithm for Planning Collision-free Paths Among Polyhedral Obstacles.
Communications of the ACM 22, 1979.

[27] Lozano-Perez, T.
Spatial Planning: A Configuration Space Approach.
IEEE Transactions on Computers C-32(2), 1983.

[28] Popplestone, R.J. and Ambler, A.P. and Bellos, I.
An Interpreter of a Language for Describing Assemblies.
Artificial Intelligence 14(1), 1980.

[29] Requicha, A.A.G. and Voelker, H.B.
Solid Modeling: Current Status and Research Directions.
IEEE Computer Graphics and Applications , October, 1983.

[30] Requicha, A.A.G.
Toward a Theory of Geometric Tolerancing.
International Journal of Robotics Research 2(4), 1983.

[31] Reuleaux, Franz.
The Kinematics of Machinery: Outline of a Theory of Machines.
Dover Publications, 1963.

[32] Schwartz, J.T. and Sharir, M.
On the Piano Movers II. General Techniques for Computing Topological Properties on Real Algebraic Manifolds.
Advances in Applied Mathematics (4), 1983.

[33] Shigley, J.E. and Mischke, C.R.(editors).
Standard Handbook of Machine Design.
McGraw Hill Book Company, 1986.

[34] Speckhart, F.H.
Calculation of Tolerance Based on a Minimum Cost Approach.
Journal of Engineering for Industry 94(3), 1972.

[35] Spotts, M.F.
Allocation of Tolerance to Minimize Cost of Assembly.
Journal of Engineering for Industry 95(3), 1973.

[36] Turner, J.U.
Tolerances in Computer-Aided Geometric Design.
Technical Report TR-87018, Rensselaer Polytechnic Institute, 1987.

[37] Turner, J.U. and Wozny, M.
Tolerances in Computer-Aided Geometric Design.
Technical Report TR-87063, Rensselaer Polytechnic Institute, 1987.

[38] Wright, P. et al.
Characterization, Planning and Execution of Manufacturing Processes.
1988
Informal summary, ASME/NSF Workshop on Tolerancing.

[39] Yap, C.K.
Algorithmic Motion Planning.
In Schwartz, J.T. and Yap, C.K. (editors), *Algorithmic Approaches to Motion Planning.*
Lawrence Earlbaum Associates, Hillside HJ, 1987.