The Theory Behind Setup Maps:

A Computational Tool to Position Parts for Machining

by

Nathan Kalish

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved May 2016 by the
Graduate Supervisory Committee:

Joseph K. Davidson, Co-Chair
Jami J. Shah, Co-Chair
Yi Ren

ARIZONA STATE UNIVERSITY

August 2016

ABSTRACT

When manufacturing large or complex parts, often a rough operation such as casting is used to create the majority of the part geometry. Due to the highly variable nature of the casting process, for mechanical components that require precision surfaces for functionality or assembly with others, some of the important features are machined to specification. Depending on the relative locations of as-cast to-be-machined features and the amount of material at each, the part may be positioned or 'set up' on a fixture in a configuration that will ensure that the pre-specified machining operations will successfully clean up the rough surfaces and produce a part that conforms to any assigned tolerances. For a particular part whose features incur excessive deviation in the casting process, it may be that no setup would yield an acceptable final part. The proposed Setup-Map (S-Map) describes the positions and orientations of a part that will allow for it to be successfully machined, and will be able to determine if a particular part cannot be made to specification.

The Setup Map is a point space in six dimensions where each of the six orthogonal coordinates corresponds to one of the rigid-body displacements in three dimensional space: three rotations and three translations. Any point within the boundaries of the Setup-Map (S-Map) corresponds to a small displacement of the part that satisfies the condition that each feature will lie within its associated tolerance zone after machining. The process for creating the S-Map involves the representation of constraints imposed by the tolerances in simple coordinate systems for each to-be-machined feature. Constraints are then transformed to a single coordinate system where the intersection reveals the common allowable 'setup' points. Should an intersection of the six-dimensional constraints exist, an optimization scheme is used to choose a single setup that gives the best chance for machining to be completed successfully. Should no intersection exist, the particular part cannot be machined to specification or must be re-worked with weld metal added to specific locations.

I would like to dedicate this thesis to my parents, who taught me how to think and how to reason. “School is not some place we drop you off while we work, it’s a place to be curious, ask questions and *learn*.”

ACKNOWLEDGMENTS

Foremost, I would like to acknowledge the tactful guidance from my academic advisor, Dr. Joseph Davidson. His attention to detail and structured approach to conducting research have been a great influence on me. The hours spent discussing kinematics, geometry, and trips to the mountains will be remembered.

My thanks are extended to my co-advisor Dr. Jami Shah for the useful, realistic and forward thinking approach to research in mechanical engineering design. His efforts in creating a renowned mechanical design department at Arizona State University are felt by many, including myself.

I would also like to thank Dr. Yi Ren for serving on my committee and offering his valuable advice to the optimization portion of this research.

I gratefully acknowledge the support and project guidance from Mr. Jiten Shah from Product Development and Analysis LLC, and the financial support from the Digital Manufacturing and Design Innovation Initiative (project DMDII-14-07-03).

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

LIST OF SYMBOLS AND NOTATION

(L', M', P, Q) – T-Map coordinates for an axis, represented in Plücker line coordinates.

(p', q', s) – T-Map coordinates for a planar feature representing the coefficients of the plane equation for a plane oriented with its normal along the z-axis.

$(\Delta x, \Delta y, \Delta z)$ – Small translational displacements along an x-, y- and z-axis, respectively.

(ϕ, ψ, θ) – Small rotational displacements about an x-, y- and z-axis, respectively.

$\left(a_{\phi}, a_{\psi}, a_{\theta}, a_{\Delta x}, a_{\Delta y}, a_{\Delta z}, b\right)$ – The coordinates of a "left" half-space or a hyperplane in 6-D small displacement space.

$\left[\$\$_{ij}\right]$ – The screw transformation matrix as described in Equation 4.52 of [1]

FxCS – Fixture Coordinate System

GCS – Global Coordinate System

GD&T – Geometric Dimensioning and Tolerancing

LCS – Local Coordinate System

Pre-superscript Notation (${}^{i}x$)—Element 'x' represented in coordinate system 'i'

Primed Notation (p', ϕ', etc.) – Angular deviations scaled by a length other than 1.0

S-Maps – "Setup Maps"

TBM – "to be machined"

T-Maps – "Tolerance Maps"

VCS – Vertex Coordinate System

CHAPTER 1 INTRODUCTION

1.1 Background and Motivation

Large, mechanical components are often made by the process of casting, or by welding smaller discrete pieces into one large weldment. While they are economical for creating intricate parts, these processes are plagued by low geometric and dimensional accuracy, often requiring further material removal through machining operations to create functional surfaces with higher dimensional accuracy. Often, the bulk of the material is left un-machined in its initial, as-fabricated state while only a select few surfaces are machined. Machined surfaces are typically ones that will mate with other parts in the overall assembly, or have some other functional necessity for dimensional accuracy. For simplicity and succinctness, I will refer to surfaces that are not machined as “as-cast” features. Surfaces that require machining or “cleanup” will be referred to as “to-be-machined” or “TBM” features in their as-cast state, and “machined” or “finished” features after machining has occurred.

The problem of low dimensional accuracy in castings and weldments applies to size, form, position and orientation of as-cast features. For this reason, when a casting is “set up” or “configured” on a fixture for machining, a significant amount of adjustment must be undertaken to ensure that cleanup operations will create acceptable finished features. Typically, the machinist will start by aligning the part so that one machining operation will be performed successfully, then successively align others feature, considering the locations of pairs of TBM features one at a time, until all are lined up. For large parts with few TBM features, the time taken to configure may exceed that of the machining operations. For increasingly complex parts, the machinist must ensure that all TBM features will be positioned properly with respect to the machining tool cutters and tool paths. For large castings, setup takes place at the machine tool so that time spent adjusting the part is time that the machine sits idle.

Additionally, to account for complex parts with many TBM features, manufacturers may cast these features with an excess of material, called “machine stock”, to ensure that all machined features are created successfully. Conversely, for parts that are cast with insufficient machine stock or with large feature deviations, there may not be a set-up configuration that allows machining to successfully create a part to specification.

For the purposes stated above, a computational tool is needed to find an adjustment that: (a) locates a part such that *all* TBM features are positioned in a manner that each will be cleaned up by its corresponding machining operation; (b) splits the difference of allowable adjustment between TBM features; and (c) satisfies any specified tolerances. The tool must also identify the case in which no adjustment will allow the creation of an acceptable part due to excessive feature variations from the initial fabrication process. Such a tool is not specific to castings and weldments, and as such, can be used for any fabrication process that requires cleanup.

1.2 **Problem Statement**

The computational tool must describe the allowable or necessary small displacements of each individual TBM feature based on its actual location in order for the machining operation to create an acceptable finished feature. Displacements must be described in the six kinematic degrees of freedom of 3D space—three rotations and three translations—to capture all acceptable configurations of a given part. The tool must then find a single small displacement of the entire part body that satisfies the small displacement requirements at each TBM feature and splits the difference of deviation allowance between TBM features.

1.3 Approach

The principles for implementing a computational tool that can position a casting for cleanup machining are:

- The final cleaned up surfaces lie somewhere within the rough, as-cast body just below the surface of the machining stock. Acceptable setups will position the entire casting such that prescribed machining operations will trace out finished features that lie within tolerance zones.
- For ensuring the complete machining of any given feature, when more material is present at the TBM features, the model gives more latitude for positioning the feature with respect to the machining operation boundaries.

An analogous model, as shown in Figure 1.1 is that of clearance between parts in an assembly. Previous models have been created to determine the clearance of pin-hole assemblies. These models can be adapted to represent the machinability of parts with excess material. The goal then is to represent the machinability condition as an assemblability problem between then as-cast part (larger, red boundaries) and a virtual part corresponding to the machining operations (smaller, green boundaries). The expected finished feature must “fit” within the as-cast workpiece feature.

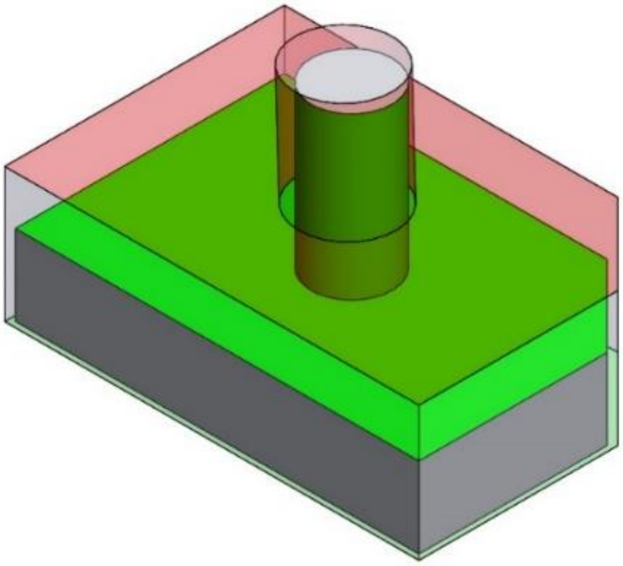


Figure 1.1 The Analogous Interpretation of the Machinability Problem to that of Assemblability Between Two Parts.

The proposed tool exploits geometry in six dimensions corresponding to the components of a small displacement. Each TBM feature imposes geometric constraints to a 6-D space. In the case when part cleanup is feasible, the intersection of all constraints will yield a set of points that satisfy all constraints. From this set, one "setup point" is chosen.

The following assumptions are made:

- Deviations are on the order of tolerances which are typically one or more orders of magnitude smaller than feature dimensions; therefore, small displacements are an acceptable representation of feature deviations and part adjustments.
- The exact machining processes used (milling, drilling, boring, etc.), the machine tools and tool paths are decided by the designer and machinist, and are assumed to be satisfactory for achieving all tolerances specified in the engineering drawing when the part is setup properly.

- One setup is used to machine all TBM features. Serial setups for specific machining operations are not considered, though the methods described would still be applicable for assessing the casting.

CHAPTER 2 RECONCILING T-MAPS AND DEVIATION SPACES

In order to represent the allowable deviations of features, I will bring attention to two existing models, created to represent the deviations of features within their tolerance zones. Devised for differing reasons, the two have converged on the geometrical representation for various tolerances as polytopes, though their methods leave a gap in understanding their interrelationship. In this chapter, I will describe the relationships between two prominent models for describing GD&T: the Tolerance-Map (T-Map) and the Deviation Space. From these two models, I will illuminate the necessity for a hybrid model to make them consistent.

Many similar models have been constructed to represent variations of surfaces that are allowable within their respective tolerance zones. Two joint papers by the various teams [2,3] describe the most prominent three: T-Maps©, Deviation Domains, and Technologically and Topologically Related Surfaces (TTRS). The two most alike are T-Maps and Deviation Domains. These two models differ in two major aspects: choice of coordinates, and their methods of construction and representation. They both represent convex spaces of points that correspond to a unique deviation of a feature, but out of the different contexts in which they were built, they each have significant advantages for representing feature deviations for tolerance analysis.

2.1 T-Maps

T-Maps were created to model the tolerance types consistent with the ASME Y14.5-2009 standard for GD&T ([4]). T-Maps are represented as convex shapes in an affine point-space. Simply, this means that the T-Map volume is represented as a continuous set of points in a space with similarity operations that preserve ratios of length in parallel line segments (such as shearing, rotating, translating and scaling). Also, the space is metric and therefore Euclidean because the coordinates have consistent units. An operation on the space can be represented as an operation on the set of points. Each point in the space corresponds to a set of feature coordinates that uniquely describe the location of deviated feature. If the point lies within the T-Map, the feature described lies within its tolerance zone.

2.1.1 **Construction** The T-Map space is an affine space defined by areal coordinates. A set of basis points (σ_i) correspond to basis features which are those that represent maximally deviated features within the tolerance zone including displaced features, and features that vary in size. A set of weights (λ_i) are used as coordinates to describe the location any particular feature, defined as point 'σ' in the affine space. The equation describing a point in the affine space is:

$$\sigma = \sum \lambda_i \sigma_i, \text{ such that } \sum \lambda_i = 1. \tag{2.1}$$

The benefit of using areal coordinates with basis features is that they allow specification of constraints regardless of a choice of coordinate frame in which the features are represented. They are invariant, just as the distance between two points does not depend on a defined coordinate system. When specifying basis points, T-Maps use coordinates that correspond to the coefficients of the equations that define the feature e.g. a plane or the axis of a cylinder.

2.1.2 **Feature Coordinates** To define the basis points of a T-Map, the coefficients of the feature are used. For example, in [5] and [6], planar features are described using the equation of a plane when the coordinate system O*xyz* is located at the center of the nominal planar face with the z direction parallel to the plane normal (the "canonical", or most simple frame), as illustrated in Figure 2.1(a). The equation of the plane is:

$$px + qy + rz + s = 0 \tag{2.2}$$

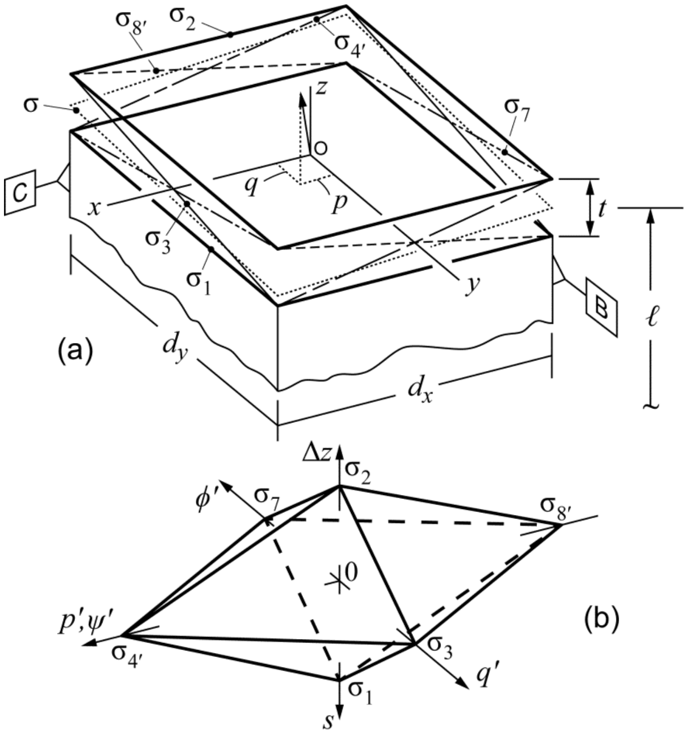


Figure 2.1 The Tolerance Zone of a Planar Feature and its Corresponding T-Map.

When coefficients p, q, and r are normalized so that $\sqrt{p^2 + q^2 + r^2} = 1$, they represent the unitless direction cosines of the plane normal, and s is the distance from the plane to the origin. Because the plane will experience only small variations, the assumption is that $r = 1$, and p and q are small values. Therefore, a unique plane with small deviations can be specified with the triad: (p, q, s). However, for defining a metric space of points, all coordinates for T-Maps must have consistent dimensions. The dimensionless coefficients p and q are multiplied by a length corresponding to half the largest size dimension of the planar feature—d_y in Figure 2.1(a). This value is known as the “characteristic length”. The primed notation variables (p' and q') have dimension [length], and the volume or content of the T-Map represents a quantity corresponding to the amount of variation allowed by the specified tolerance. The characteristic length scales the variables that correspond to angular deviations.

Feature coordinates have the advantage that they directly correlate with the feature and describe a mathematical surface. Their disadvantage comes when comparing the common deviation of different surfaces. For example, The T-Map for an axis, as explained in chapter 4.3, may be described with four independent coefficients for defining the line in a canonical reference frame (L', M', P, Q). When comparing the deviation of a *plane* attached to a part with respect to an *axis* attached to the same part, one must first associate the line coordinates of the axis with the plane coordinates to describe the relative position and orientation. It is possible, but not without tedious manipulation of variables first.

However, the use of feature coordinates allows the T-Map model to extend its reach to size variations as well. For example, in [7], when describing clearance between a pin and a hole, as the pin gets smaller, the T-Map indicates that there is more clearance. Allowing other dimensions to enter the model allows greater representation of the effects of standard tolerances.

2.2 Deviation Spaces, Clearance Spaces, and Small Displacement Coordinates

Deviation spaces and Clearance spaces are models that seek to quantify the amounts of deviation in assemblies and stack-ups. The deviation domain is used for the same purposes as T-Maps, but it is unique in its choice of coordinates, and its construction.

2.2.1 **Deviation Spaces and Clearance Spaces** The defining feature of Deviation Spaces and Clearance Spaces are the small displacement coordinates used to describe deviation of the feature relative to the part. Furthermore, a distinction is made in [8]: the deviation space represents allowable feature deviations in a tolerance-zone, while clearance spaces represent the possible displacements of parts that arise in assemblies with part clearance.

2.2.2 **Small Displacement Coordinates** To describe the displacements of a feature within a tolerance zone, or the available freedom in an assembly pair, Deviation and Clearance Spaces employ the small displacement “torsor.” As explained in [9], torsors acknowledge that a small displacement is not dependent on location of the coordinate system. The torsor notation represents a mathematical interrelationship of features established by tolerances without explicating the intricacies of computing actual deviations. For this reason, torsors are useful for explaining connectivity with relative position vectors, but are less useful when computing actual displacements. Unlike the areal coordinates defined in T-Maps, the method requires the choice of a coordinate reference frame. In [9], the authors assign an origin (Oxyz) to the space at the time of displacement calculation. Once a coordinate frame is established, the displacement is described with up to six dimensions corresponding to three independent rotations and three independent translations. While the authors of [9] use other symbols to describe this displacement, for the sake of keep notation straight, I will call the three rotational components φ, ψ, θ for small rotation variables about x, y, and z axes, respectively; and the three translations will be Δx, Δy, and Δz. The small displacement of a frame attached to a deviated feature from its nominal frame will be represented in the vector notation:

$$^{O}[\phi, \psi, \theta, \Delta x, \Delta y, \Delta z]$$

In this notation, the pre-superscript symbol corresponds to the coordinate frame in which the displacement is measured.

The small displacement coordinates have the advantage of describing the small relative movement of any feature in a uniform way, whether it is a plane, axis, cylinder, or any other feature. When a feature displaces from its nominal position, it is described by the rotation and translation of the perfect form feature in coordinates that are not dependent of the feature itself. In addition, small displacement coordinates can describe displacements that do not change the values of feature coordinates. For example, if a planar feature is shifted in a direction parallel to its surface, feature coordinates will indicate no change. Similarly, if a cylinder is rotated about or translated along its axis, the coordinates of the axis-line will not change. With small displacements, one can compare the displacements of planes, cylinders, and any other feature with common coordinates. For further discussion, see Appendix A on how feature coordinates and small displacement coordinates are related. An equally valid method would be to use small displacement screws, however the small displacement coordinates cleanly separate rotations and translations of a feature or body with respect to the chosen coordinate system.

The disadvantage to using purely small displacement coordinates for representing tolerances is that it ignores the influence of other variations such as feature size. In [8], the author acknowledges that feature size impacts the amount of allowable variation, but denies that the polytope representation can be extended to higher dimensions that represent other feature variables. In addition, small displacement coordinates do not have consistent variables. Three of the coordinates represent angles, and are therefore unitless, while the other three have dimension [length]. Because of this, the volume of the bounded space does not have the same interpretable nature as T-Maps.

2.2.3 **Deviation Space Construction** As best explained in [9] and [10], the approach for constructing deviation spaces is to compute the linear inequalities that arise due to tolerance specifications. Each point on a surface will contribute a pair of half-spaces that bound the six-dimension space in one direction. As it turns out, only those points that lie on the convex hull of the surface will contribute to the final shape of the deviation space polyhedron. A coordinate system is chosen, and all half-spaces are computed for all features of interest in this one reference frame. There is no guarantee of a completely bounded 6D space, and in most cases, the deviation space polyhedron for a single feature will be open ended for unconstrained directions of the surface.

In order to handle the possibility of open-ended polyhedra, Homri et al. [11] suggest the use of “cap half-spaces” to create a fully consistent computational tool that can handle polytope operations for all possible operand polyhedra. In this method, polytopes are represented by the set of half-spaces and vertices at the intersections of half-spaces, called the “HV-description”. For tolerance analysis methods, this full description is necessary for computing the Minkowski sums of polytopes—a useful operation for tolerance stack-up calculations; however, as will be demonstrated later, only the half-spaces are necessary for computing intersections and optimal casting positions.

2.3 Comparing Coordinates

It is apparent that T-Maps and Deviation Spaces are quite alike. Both arrive at a polytope representation of a set of points that correspond to feature deviations. However, neither those creating Deviation Spaces nor those creating T-Maps have previously shown the connection of T-Map and small displacement coordinates, not even in [2] or [3]. Feature coordinates, specific to the type of feature (plane, axis, etc.) can be transformed to small displacement coordinates, or vice versa. For example, the plane σ in Figure 2.1 is represented by feature coordinates (p, q, s). The projection of the unit normal onto the x-axis of the coordinate system gives p. Assuming small rotations, when the plane is rotated about the y-axis in the right-handed sense by angle ψ in radians, the value of p increases by the same value.

$$p = \sin(\psi) \approx \psi \tag{2.3}$$

In contrast, when the plane in Figure 2.1(a) is rotated about the x-axis by an angle ϕ in radians in the right-handed sense, the projection of the unit normal onto to the y-axis, labelled q, decreases by the same value.

$$q = -\sin(\phi) \approx \ -\phi \tag{2.4}$$

And finally, when the plane is displaced in the positive z-direction, the value of s decreases by the same value.

$$s = \ -\Delta z \tag{2.5}$$

As mentioned before, T-Maps use consistent units. The final conversion is to multiply the p and q values by the characteristic length ℓ_c (d_y in Figure 2.1) to give p' and q'. In matrix notation, the transformation and the inverse transformation take the forms in Table 2.1 when there is no displacement along invariant directions (translations in the plane, rotation about the normal). The polytope in Figure 2.1(b) includes both the coordinate systems for feature coordinates and the small displacement coordinates to show their correlation. The primed ϕ' and ψ' indicate that the angular small displacements were scaled by a characteristic length other than 1.0 as well.

Table 2.1 Relating T-Map and Small Displacement Coordinates for a Plane

Small Displacements to Plane Coordinates	Plane Coordinates to Small Displacements
$\begin{bmatrix} p' \\ q' \\ s \end{bmatrix} = \begin{bmatrix} 0 & \ell_c & 0 \\ -\ell_c & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} \phi \\ \psi \\ e_z \end{bmatrix}$ (2.6)	$\begin{bmatrix} \phi \\ \psi \\ \Delta z \end{bmatrix} = \begin{bmatrix} 0 & -1/\ell_c & 0 \\ 1/\ell_c & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} p' \\ q' \\ s \end{bmatrix}$ (2.7)

A similar process can be undertaken with the T-Map for an axis. The explanation for how the transformations were created can be found in Appendix A. The resulting transformations are shown in Table 2.2 for the case when there is no displacement along invariant directions. Appendix A also explains the full matrix representation for transformation between coordinates when there may exist displacement along invariant directions. Table 2.2 shows the transformations for line coordinates to small displacements for a cylinder represented in its canonical coordinate system.

Table 2.2 Relating T-Map and Small Displacement Coordinates for an Axis

Small Displacements to Axis Coordinates	Axis Coordinates to Small Displacements
$\begin{bmatrix} L' \\ M' \\ P \\ Q \end{bmatrix} = \begin{bmatrix} 0 & \ell_c & 0 & 0 \\ -\ell_c & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} \phi \\ \psi \\ e_x \\ e_y \end{bmatrix}$ (2.8)	$\begin{bmatrix} \phi \\ \psi \\ e_x \\ e_y \end{bmatrix} = \begin{bmatrix} 0 & -1/\ell_c & 0 & 0 \\ 1/\ell_c & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} L' \\ M' \\ P \\ Q \end{bmatrix}$ (2.9)

2.4 Setup-Maps: A Hybrid Model for Casting Position

Returning to the problem of positioning castings for cleanup machining, it is apparent that neither the T-Map nor the Deviation Space models can satisfactorily be used without adaptation. The proposed model will incorporate elements of both models. The model will map the setup configurations of a part to points in a higher-dimension space, and will therefore be called the Setup-Map, or S-Map.

2.4.1 **Coordinate Selection** A simple way to represent a unique setup of a casting is by the small displacement with respect to a reference coordinate system corresponding to its initial fixtured position. For comparing the allowable deviations of TBM features of different geometry, small displacement coordinates are a unifying choice. In addition to small displacement coordinates, because feature sizes will influence the amount of allowable position variation, their contribution must be included. For these reasons, the six small displacement coordinates combined with any number of feature-size coordinates will be used to describe the allowable positions of castings in the Setup-Map (S-Map) model. It is convenient to remember that S-Maps use Small displacement and feature Size coordinates.

2.4.2 **Setup-Map Representation** There are many challenges in finding the intersection of higher-dimensional objects, and many models approach the problem in various ways. One possibility would be to discretize the higher dimensional space. Sets of ‘voxels’ or ‘volumetric pixels’ could be used for performing basic set algebra inclusion or exclusion operations. This has the problem of resolution and oversized models. The number of elements needed to span the working portion of six or greater dimension space increases quickly as dimension increases. For this reason, a continuous method is preferred. A boundary representation (B-Rep)[12] is useful for intersecting generally shaped objects, but requires the association of topological boundary elements for each higher dimension element in the model—each line is bounded by two points; each plane is bounded by lines; each volume is bounded by planes; each n-‘hyper’volume is bounded by (n-1)-‘hyper’planes; ad infinitum. Even in 3-D, we run into difficult problems with topology connectivity for intersecting shapes with reentrant angles. However, as will be shown, the S-Map will represent a convex space. As shown in [13], a convex space can be represented as the intersection of a set of half-spaces, known as a ‘polytope’. This simplifies the process and allows the use of standard algorithms for intersection as defined in [13].

In implementation, the S-Maps will use collections of half-spaces for representing the allowable adjustment at each feature to create a polyhedron. Intersection of multiple polytopes is then performed by collecting all half-spaces of each polytope and computing the new intersection polytope. This process will be described in more detail in chapter 4. The essential operation for S-Maps is the transformation between coordinate systems. Like T-Maps, S-Maps will be created in canonical coordinate systems, and then transformed to be represented in an alternate coordinate system. Points in S-Map space are transformed from the j-frame to the i-frame using the screw transformation described in equation 4.52 of [1], repeated here with small displacement coordinates:

$$^{i}\begin{bmatrix} \phi \\ \psi \\ \theta \\ \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = {}^{i}\begin{bmatrix} [R_{ij}] & \emptyset_{3\times 3} \\ [X][R_{ij}] & [R_{ij}] \end{bmatrix} {}^{j}\begin{bmatrix} \phi \\ \psi \\ \theta \\ \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} \text{ where } [X] = {}^{i}\begin{bmatrix} 0 & -z_j & y_j \\ z_j & 0 & -x_j \\ -y_j & x_j & 0 \end{bmatrix} \qquad (2.10)$$

and [R_{ij}] is the 3x3 rotation matrix whose three rows are the (*x*, *y*, *z*) components of unit vectors corresponding to the *x*-, *y*- and *z*- coordinate axes of the *j*-frame represented in the *i*-frame. Each unique small displacement of the cast part is represented as a point in a *continuous* six-dimensional space. In order to represent *the* continuum of points, the constraints that define valid points are imposed as inequalities. To simplify computer implementation, linear inequalities will be used—the equivalent geometric interpretations of these entities are called 'half-spaces'. In a higher-dimension Euclidean space, linear half-spaces (*H*) are represented as all of the points $P(x_1, x_2, \dots, x_n)$ to one side of a 'hyper'plane and those in the hyperplane itself. I have chosen to use half-spaces that represent points "below" the hyper-plane, denoted the "left half-space" in [13]. Algebraically, this is denoted as:

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n + b \leq 0 \tag{2.11}$$

Geometrically, the vector $(a_1, a_2, \dots, a_n)$ will represent the *outward* normal of the hyperplane, and $b\big/\sqrt{\sum a_i^2}$ is the signed distance *from* the hyperplane *to* the origin of coordinates along the normal. For half-spaces that contain the origin of coordinates, b is *negative*. The equation for the hyperplane is normalized so that $\sum a_i^2 = 1$ and b is the distance from the plane to the origin which will be useful for the algorithms discussed later in chapter 5.

Small displacement coordinates represented in the 3-D spatial j-frame ${}^{j}(\phi, \psi, \theta, \Delta x, \Delta y, \Delta z)$ describe points in a 6-D space. To be conformable with a metric space, and for subsequent algorithmic operations, the angular coordinates (ϕ, ψ, θ) have an implicit characteristic length of 1 unit (in., mm, etc.). This value will not interfere with the screw transformation mathematics where the angular displacement value in radians will be equal to the metric value in the consistent 6-D space. Care must be taken when using the screw transformation—the matrix applies to angular values in radians, so scaled angular values must be divided by their characteristic length before transformation. The homogeneous representation of a point includes an extra coordinate that denotes the scale of the coordinate system. The homogeneous representation of a point in small displacement coordinates is: ${}^{j}(\phi, \psi, \theta, \Delta x, \Delta y, \Delta z, 1)$. The equation of the half-space for small displacements is:

$$a_{\phi}\phi + a_{\psi}\psi + a_{\theta}\theta + a_{\Delta x}\Delta x + a_{\Delta y}\Delta y + a_{\Delta z}\Delta z + b \leq 0 \tag{2.12}$$

or:

$$[a_{\phi}\ a_{\psi}\ a_{\theta}\ a_{\Delta x}\ a_{\Delta y}\ a_{\Delta z}\ b][\phi\ \ \psi\ \ \theta\ \ \Delta x\ \ \Delta y\ \ \Delta z\ \ 1]^{T} \leq 0 \tag{2.13}$$

where $\left(a_{\phi}, a_{\psi,}\, a_{\theta}, a_{\Delta x}, a_{\Delta y}, a_{\Delta z}, b\right)$ are the coefficients of the hyperplane in small displacement space. As derived in Appendix B, the transformation to represent the half-space constraints for small displacements in a new 3-D spatial coordinate system is:

$$\begin{aligned} &{}^{i}[a_{\phi}\ \ a_{\psi}\ \ a_{\theta}\ \ a_{\Delta x}\ \ a_{\Delta y}\ \ a_{\Delta z}\ \ b] \\ &= {}^{j}[a_{\phi}\ \ a_{\psi}\ \ a_{\theta}\ \ a_{\Delta x}\ \ a_{\Delta y}\ \ a_{\Delta z}\ \ b] \begin{bmatrix} \multicolumn{3}{c}{[R]^{T}} & \multicolumn{3}{c}{\emptyset_{3\times 3}} & \begin{matrix}0\\0\\0\end{matrix} \\ \multicolumn{3}{c}{[R]^{T}[X]^{T}} & \multicolumn{3}{c}{[R]^{T}} & \begin{matrix}0\\0\\0\end{matrix} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{2.14}$$

Note that because this corresponds to a shearing operation it does not preserve the normalization of the coordinates representing each half-space, and therefore the

${}^{i}(a_{\phi}, a_{\psi,}\, a_{\theta}, a_{\Delta x}, a_{\Delta y}, a_{\Delta z}, b)$ coordinates must be re-normalized after applying this transformation for the algorithms described in Chapter 5.

Now that the S-Map coordinates representing small body displacements and feature sizes have been chosen, and the fundamental transformations have been described, the next step is to describe which small displacements are considered acceptable. The next chapter will illustrate the principles for deciding how constraints are chosen. The following will describe the process for building the constraints that ultimately intersect to create the S-Map.

CHAPTER 3 PRINCIPIA SETUP-MAPICA

The Setup-Map (S-Map) for a part is a description of all the allowable or necessary small displacements of the cast part that achieve two basic objectives:

1. All surfaces are to be machined completely, and
2. Any specified tolerances will be satisfied by the machining operations when performed in the chosen setup.

The S-Map for a part is generated by intersecting S-Map primitives each of which represents the allowable adjustments of the casting to obtain complete machining of the feature. For surfaces with geometric tolerances in addition to size boundaries, multiple S-Maps may be used to describe the allowable adjustment. It is useful to maintain the separation of the multiple S-Map primitives for a single feature to be able to discern which specific S-Map primitives limit the amount of acceptable adjustment the most in the final S-Map. In this section, I will explain the principles for creating S-Map primitives that describe the limits on adjusting the entire part imposed by a single feature.

3.1 S-Maps for Machinability

The first criterion for qualifying and positioning castings answers the question: Is there enough material in the right spot to complete the machining operations? When a casting is disqualified, it is due to a lack of material where the intended machining operations will take place. The trivial solution is to always include an excessive amount of machining stock on all features that are to be machined. So, why not start with a large block of material with a size greater than the bounding box on the part, machine the entire part from it, and forget the work in this thesis entirely? Well, before jumping to such extremes, there are valuable reasons for casting a part that is as close to the final product as possible. Including excessive machining stock is not cost effective for manufacturing—more material will be turned to scrap, it will drive up the time taken to machine the parts, reduce the working life of cutting tools, and possibly adversely affect the dimensional accuracy of the as-cast part during the cooling process via a buildup of residual stresses. This does, however raise a valuable principle for creating the S-Map primitive of a feature: when more material is present at a feature, there is a larger amount of acceptable position variation when considering the machinability for the individual feature.

3.1.1 **Casting and Machining Process Design Expectations** The first step taken by the process designer is to determine where machining stock will be added, and where cleanup machining will take place. For the purposes of S-Maps, the extra information that must be included in addition to the CAD model and GD&T is this specification of which features are marked for finishing machining. As shown in Figure 3.1, a hypothetical nominal part is used to determine a machining plan and the intended as-cast part with machining stock. It is assumed that if a perfect part were placed on the machining tool, the cutting tool would trace the surfaces—not cutting material, nor losing contact.

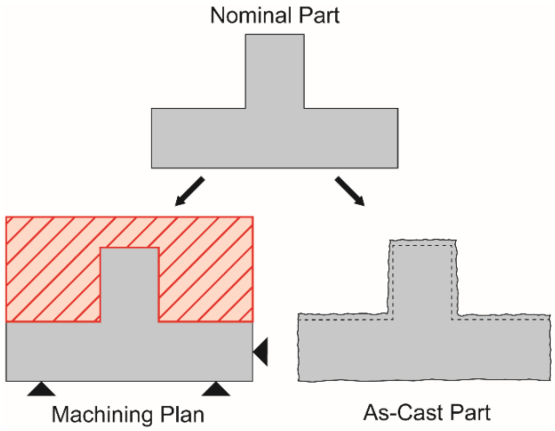


Figure 3.1 Deriving the Machining and Casting Plan from the Nominal Part

3.1.2 **S-Maps for Machinability and the Analogy to Assemblability** As demonstrated in Figure 3.2(a), if a cast part with some deviation were to be placed on the fixture from Figure 3.1 with no adjustment, it is uncertain that the pre-specified machining operations would yield acceptable cleaned up features. If the fixture is adjusted, it may be possible to align the part with respect to the machining operations to create acceptable finished features, as shown in Figure 3.2(b).

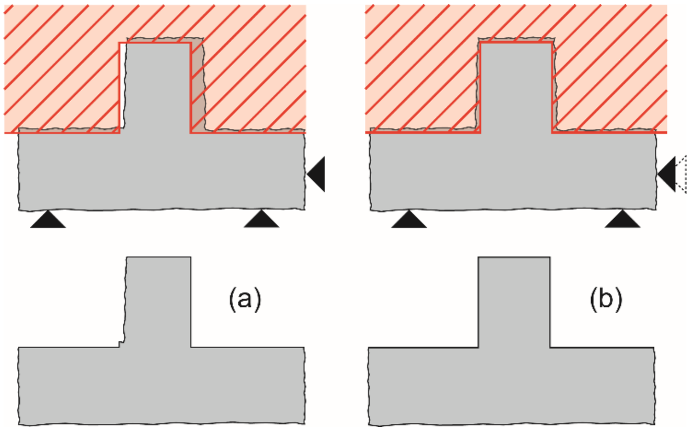


Figure 3.2 Adjusting a Part into the Machining Region

It is made apparent by Figure 3.2 that the goal of machinability is to *fit* the machining operation boundary, represented as the red boundary line of the cross-hatched zone, within the material that is present. We can think of the problem of machinability as one of trying to assemble an analog part corresponding to the machining operation boundaries within another analog part corresponding to the material boundaries. Reversing the boundaries as shown in Figure 3.3, we can reframe the problem in terms of clearance in an assembly, which is discussed in both [7] and [8]. As demonstrated in Figure 3.3(a), when there is enough machining stock on the part on the left, the analog assembly to the right has clearance between the parts, indicating that the two can be assembled. This is in contrast with Figure 3.3(b) where not enough material is present on the left part for the prescribed machining operations to complete successfully. In this case, the analog assembly shows part interference, indicating an infeasible assembly, and therefore an infeasible casting.

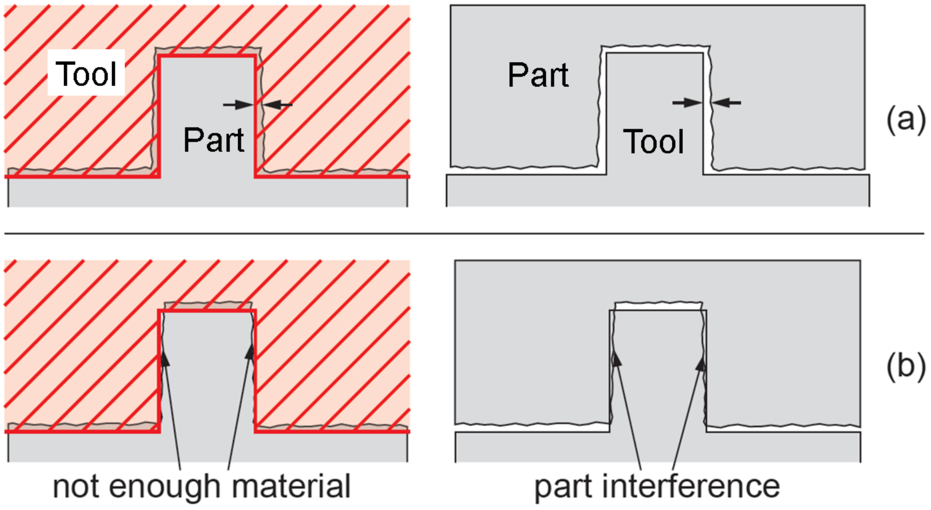


Figure 3.3 Equivocating the Machinability Problem with Assemblablity

3.1.3 **Minimum Material Envelope** When determining the actual material boundaries of a surface, points are measured on the surface of the TBM features. For real parts, the actual points will not lie on perfect geometrical features. Rather, they will lie on the imperfect surfaces generated from the casting process. In order to compare with the theoretically perfect nominal features used to create the tool paths, the point cloud is reduced to a geometric feature. While there exist a handful of fit types (least squares fit, one sided, constrained or unconstrained, the proper fit for TBM features should describe the “unrelated actual minimum material envelope, described in [4] as “a similar perfect feature counterpart contracted about an internal feature or expanded within an external feature, and not constrained to any datum reference frame.” If we perform a machining operation on one of the features in Figure 3.4, the operation that leaves the most material but completely removes any surface defects will trace the minimum material envelope. Therefore, any material outside of the minimum material envelope is ignored. The point cloud is then reduced to a single geometric feature, the minimum material envelope, that will be used for S-Map calculations.

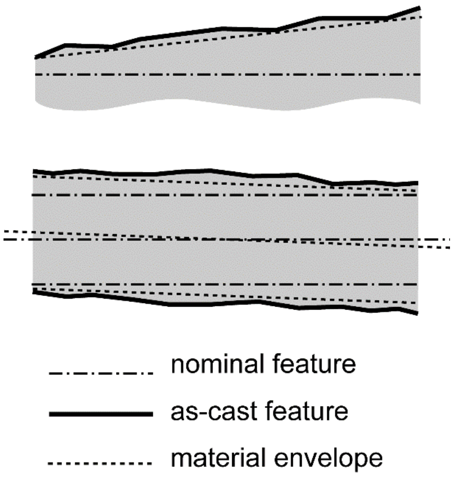


Figure 3.4 The Unrelated Actual Minimum Material Envelope for a Plane, and Feature of Size

3.1.4 **Building in Allowance for Fixturing and Machining** In Figure 3.1, I assumed that the machining operations and the setup by the machine operator would be performed perfectly. In real scenarios, it is not likely that either of these assumptions will remain valid, and it may be important to account for some amount of error in both setup and machining. If the measured minimum material envelope of a feature of size, such as a cylinder, were at its smallest limit, known as the least material condition (LMC), the machinist would be required to set up and execute the machining operation perfectly to create cleaned up features—any deviation would yield an unacceptable feature with defects. At the LMC size, the corresponding S-Map primitive is a single feasible point indicating that machining can be performed, but there is no allowance for errors. However, since at least some material must be removed, it is next to impossible to perform the operation when the feature is at its LMC size, it may be necessary to use a boundary further into the material for creating the S-Map. Depending on one's tolerance to accept some levels of risk in the machining process, one may choose to assume that the maximum material condition (MMC) size corresponds to the size at which no deviation is allowable. This gives the machinist some allowance for misaligning the feature with respect to the machine tool while still being able to create an acceptable feature; however, this may falsely disqualify some castings that might have been salvageable. Some other size may be used—perhaps one might set a fixed amount of allowance representing an offset from the LMC size for all features. This is entirely up to the designer's willingness to accept risk for accidentally removing too much material, or not fully machining a surface.

Figure 3.5 shows the tradeoff for assuming that the machining will happen at either (a) the LMC boundary of the tolerance zone, (b) the nominal feature, or (c) the MMC boundary of the tolerance zone. Three parts with different amounts of machine stock at the two opposite surfaces, whose minimum material envelopes of the planar features fit exactly at (1) the LMC tolerance-zone boundary, (2) the nominal features, (3) the MMC tolerance-zone boundary. Figure 3.5(a) shows the case of assuming that the machining operation will be performed at the LMC boundary for all features. Parts made at the LMC size, case (1), are accepted, though perfect, read as *impossible*, alignment and tool operation is required to completely the two surfaces on the left and right sides. This case accepts the most number of castings, but it will likely yield bad parts. Figure 3.5(b) is the case that assumes that the machining operation is performed at the nominal feature. This is a valid assumption, given that tool paths are intended to create features as close to the nominal as possible. For as-cast features measured as being equivalent to the nominal size, given that the tool path will trace the nominal boundaries, the part must also be setup perfectly to avoid defects. Therefore, I will assume that a feature must be at its toleranced MMC size or greater for it to be accepted, the case shown in Figure 3.5(c). This gives the machinist some allowance when trying to create nominal features, with the trade-off that it rejects as-cast parts with features that are larger than the LMC boundary.

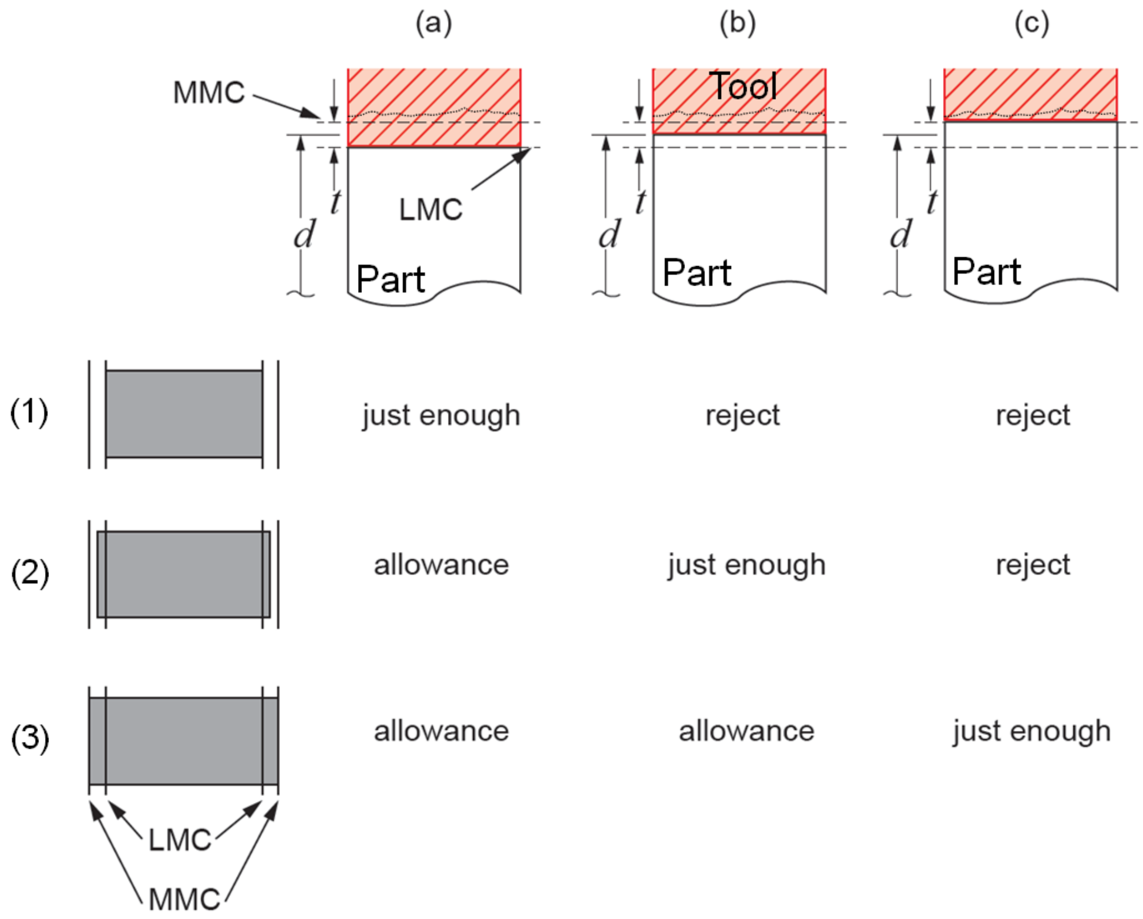


Figure 3.5 Trade-Off for Choosing an Assumed Machining Size

3.2 S-Maps for Satisfying Tolerances

In qualifying castings, it would also be helpful to ensure that the finished surface can be made in locations such that they satisfy any tolerances specified by the designer. While tolerance stack-ups usually involve tedious algorithms, for S-Maps, a few assumptions greatly simplify the task.

Consider a datum feature that is to be machined, and another TBM feature that has tolerances that relate its position or orientation to the datum feature. Because the machining operation boundaries are based on the nominal features, it is assumed that the machining operations will satisfy the tolerances between two machined surfaces when only one setup is used. By positioning each feature for machinability, the tolerances are automatically satisfied. Figure 3.6(a) shows that, because the machining plan is derived from nominal CAD, tolerances between machined features are automatically satisfied. It is up to the designer and machinist to decide which operations to use in order to achieve the specified tolerances. In a similar sense, tolerances of form are not considered because they do not represent relative positioning of a TBM feature to a datum. The assumption is that form tolerances are satisfied by choice of machining operation as well.

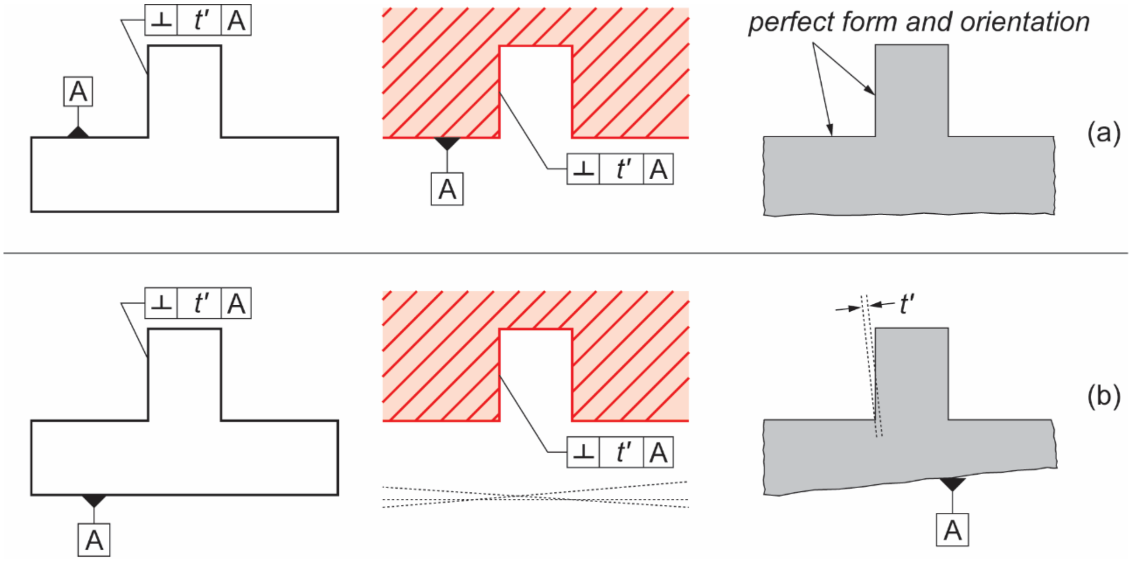


Figure 3.6 Tolerances With Respect to Machined or Un-Machined Features

What about the case when a datum is not machined? When the target feature is machined, it is created in a nominal position with respect to the machine tool coordinate system. Because tolerances are relative, the datum must be positioned in a location that would satisfy the tolerance. Figure 3.6(b) shows the restriction imposed on the adjustment of the part based on the location of the new finished feature and the as-cast datum. Tolerances are directional in that a tolerance value is applied to the target feature with respect to the datum, and not the other way around. For this reason, care must be taken in the representation of allowable adjustment of the datum feature. The size and shape of the T-Map will be derived from the shape and type (plane segment, cylinder, etc.) of the target feature, but the shift amount will be based on the actual location of the datum reference frame instead of the location of the minimum material envelope of the feature. The details of implementation will be discussed in the next chapter.

CHAPTER 4 S-MAP FORMULATION

The approach I've taken for creating T-Maps and S-Maps seeks intuition through construction. The goal is to create and intersect S-Map primitives for multiple features in a single 'global' coordinate system. Because the global coordinate system can be generally placed on the part, it is not an easy choice for constructing the initial constraints at the features. Instead, the S-Map constraints are built in a coordinate system local to the TBM feature. A similar process is used for first representing the half-spaces that make up the local feature T-Map whereby the half-spaces are generated in coordinate systems related to the vertices of a feature, then transformed to the local coordinate system. The final defining characteristic of the S-Map is that each primitive is "shifted" or translated in S-Map space by amounts corresponding to the deviation of its corresponding measured TBM feature on the cast part from its nominal location and size. The final shifted S-Map primitive represents allowable displacements of the part based on the constraints imposed by the actual feature with respect to its corresponding machining boundary or tolerance zone.

4.1 Coordinate Systems of the Part

The various coordinate systems used in building up the S-Map primitive will be denoted as such:

Global Coordinate System (GCS): The reference frame in which features on the part are nominally dimensioned and positioned and in which as-cast features are measured. In implementation, this could be the implicit coordinate system defined in the CAD model. For simplicity the GCS will often coincide with the Fixture Coordinate System.

Fixture Coordinate System (FxCS): This reference frame represents the coordinate axes and ‘zero’-position of the machining tool. This will likely coincide with the GCS, though it is possible to identify an alternate coordinate system in which the fixture is represented. The FxCS is located nominally with respect to the GCS.

Local/Feature Coordinate System (LCS): Each feature will have a coordinate system aligned with its defining geometry for determining allowable small displacements at the feature. The LCS is located *nominally* with respect to the GCS, and does not correspond to the measured feature.

Vertex Coordinate System (VCS): points or vertices that define the geometry of a locally represented feature will have their own coordinate systems. The axes of these coordinate systems are parallel to the LCS and located by vectors that describe their relative positions in the LCS. Each VCS is located nominally with respect to the LCS, and thus they are located nominally to the GCS.

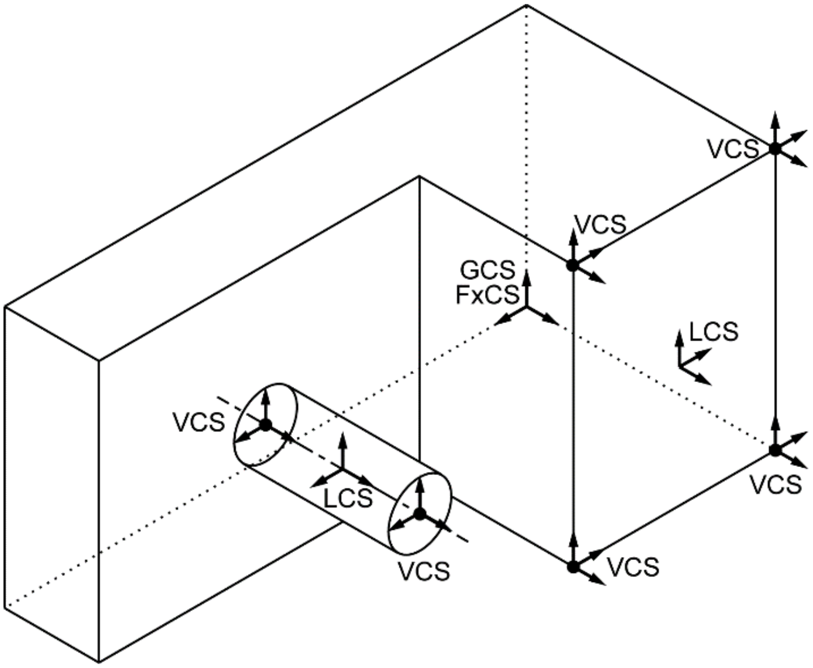


Figure 4.1 A Part with the Various Coordinate Systems for Constructing S-Maps shown.

4.2 Creating T-Maps and S-Maps for Planar Surfaces

T-Maps and S-Maps describe the allowable locations for perfect form features based on a set of boundaries that no point on the feature must cross. For T-Maps, the boundaries are the tolerance zone boundaries of the feature, and for S-Maps, as discussed in §3.1.4, the feature, or some offset to it, traced by the machining operation is the one-sided boundary. Here, the scheme for creating the T-Map or S-Map in small displacement coordinates for a general planar shape is presented.

4.2.1 **Representing allowable deviations for generally shaped plane segments**. The acceptable zone for a planar feature is represented as a single plane for S-Maps or pair of perfect planes for T-Maps which no point of the feature must deviate beyond. In order to determine which combinations of rotations and translations of a plane are acceptable, the geometry of the planar segment must be taken into account. While [5] and [6] describe the method for creating T-Maps for circular and rectangular planar faces using basis points, the described method below will allow creation of a T-Map for a generally shaped plane.

First, a few definitions: the planar feature is referred to as a *planar segment*, signifying that it is only a portion of the plane; a *plane* is a flat geometric entity that extends across Euclidean space indefinitely. The planar segment exists in the plane and is bounded by the outlines of the feature. The planar segment may have internal boundaries, or it may be a collection of disjointed planar segments. Small displacement coordinates are used to describe the displacement of the plane in which the planar segment lies. The *tolerance zone* boundary is given as a pair of planes parallel to the nominal plane, each displaced by half of the tolerance amount in the case of symmetric tolerancing. These planes bound a zone that acceptable planar segments will lie within. The local coordinate system (LCS) of the plane is set up so that the *x*- and *y*-axes lie in the nominal plane, and the *z*-axis is the outward-pointing normal. No one location for locating the LCS is better than the other, but the result will exhibit more symmetry by choosing the barycenter. I will show how the T-Map is made to represent the tolerance zone, and then by extension I will show the formulation of the S-Map.

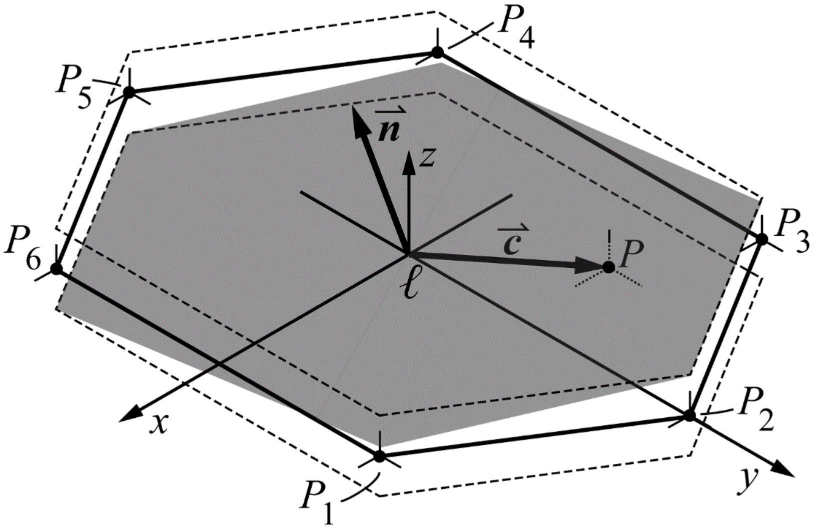
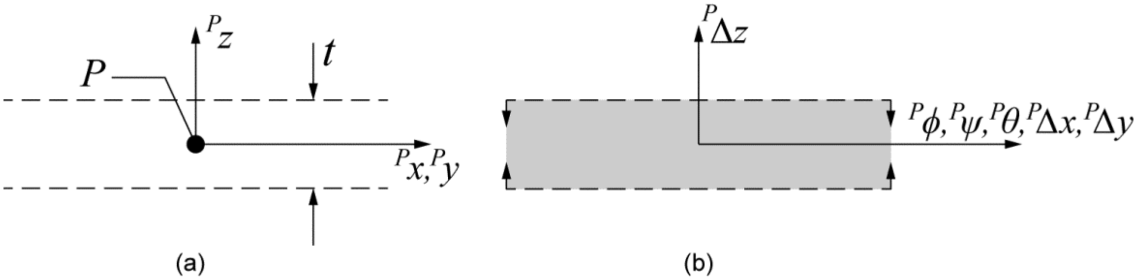


Figure 4.2 The Tolerance Zone for a Hexagonal Planar Feature

Since no point of the planar segment may lie outside of the tolerance zone, represented by the dashed lines in Figure 4.2, I will start by describing the constraint at a general point in the planar segment, *P*(*x*,*y*). While a general point is not necessarily a vertex of the planar segment, for sake of convention I will call the coordinate system established at the point the 'VCS' (vertex coordinate system). Vector $\vec{\boldsymbol{c}}$ describes the location of *P*(x,y) relative to the LCS, *ℓxyz*. In the VCS, the displacement of the point is only limited in the *z*-direction by the tolerance zone boundaries. As shown in Figure

4.3(a), no rotation or displacement in the *x*- and *y*-directions will move the point *P* beyond the tolerance zone, but point P provides no constraint to rotation about either the ${}^{P}x$- or ${}^{P}y$-axes. The T-Map "primitive" is a space bounded by two parallel planes perpendicular to the ${}^{P}\Delta z$ coordinate of small displacement space, as depicted by the dashed lines in Figure 4.3(b). The two bounding half-spaces in 6-D spaces may then be represented by their coefficients of the 6-D hyperplane, in the notation discussed at the end of §2.4.2, as the vectors:

$$ {}^{P}\left(a_{\phi}, a_{\psi}, a_{\theta}, a_{\Delta x}, a_{\Delta y}, a_{\Delta z}, b\right) = \begin{cases} \left(0,0,0,0,0,1,-\frac{t}{2}\right) \ (bottom) \\ \left(0,0,0,0,0,-1,-\frac{t}{2}\right) \ (top) \end{cases}. \tag{4.1} $$



${}^{P}z$
t
P
${}^{P}x, {}^{P}y$
(a)
${}^{P}\Delta z$
${}^{P}\phi, {}^{P}\psi, {}^{P}\theta, {}^{P}\Delta x, {}^{P}\Delta y$
(b)

Figure 4.3 The Tolerance Zone and Primitive T-Map for a point P in the planar segment.

The next step is to transform the T-Map primitive from the VCS to the LCS. For this, we can use the specialized small displacement transformation and the corresponding half-space transformation, Equation 4.2 derived in Appendix B.3, to transform the half-spaces of the primitive T-Map represented in the VCS in Equations 4.1 to the LCS, in Equations 4.3:

$$ {}^{\ell}[a_{\phi} \ \ a_{\psi} \ \ a_{\theta} \ \ a_{\Delta x} \ \ a_{\Delta y} \ \ a_{\Delta z} \ \ b] = \begin{bmatrix} \pm 1 & -\frac{t}{2} \end{bmatrix} \begin{bmatrix} c_y & -c_x & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.2} $$

$$^{\ell}\left(a_{\phi}, a_{\psi}, a_{\theta}, a_{\Delta x}, a_{\Delta y}, a_{\Delta z}, b\right) = \begin{cases} \left(c_y, -c_x, 0, 0, 0, 1, -\frac{t}{2}\right) \ (bottom) \\ \left(-c_y, c_x, 0, 0, 0, -1, -\frac{t}{2}\right) \ (top) \end{cases} \tag{4.3}$$

This transformation between coordinates shears the T-Map space. The resulting half-spaces tilt, each according to the position vector $\vec{\boldsymbol{c}}$ of the VCS, as shown in the 3-D cross-sections of Table 4.1

It is apparent from the transformation matrix in Equation B.3.4 that the amount of shearing corresponds to the magnitude of vector $\vec{\boldsymbol{c}}$. In any given direction, from an interior point of the plane segment, the magnitude is always the biggest for points on the convex hull. For this reason, only the points that correspond to vertices of the convex hull of the planar segment are needed to construct the local T-Map for the planar segment. Points that are on the convex hull of the planar segment are the limiting points because a rotation of the planar segment will intersect the tolerance zone at these points before any point further in the interior. Intersecting the six T-Map primitives for the hexagonal plane in Figure 4.2 gives the resulting local T-Map in Figure 4.4. For comparison, the feature coordinates for the plane (*p*, *q*, *s*) are overlain along with the small displacement coordinates ($^{\ell}\phi$, $^{\ell}\psi$, $^{\ell}\Delta z$). The lightly shaded planes show that the primitive generated using the depicted location of point *P* in Figure 4.2 do not contribute to the final boundaries of the T-Map.

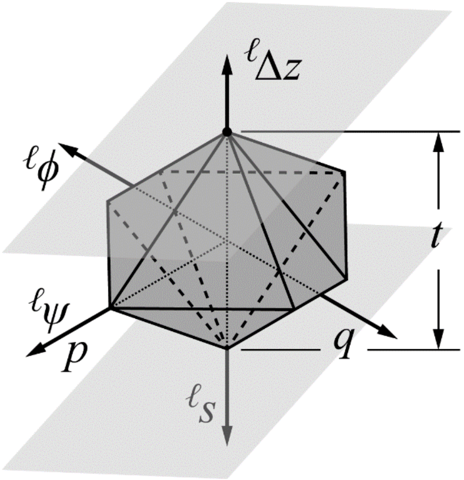
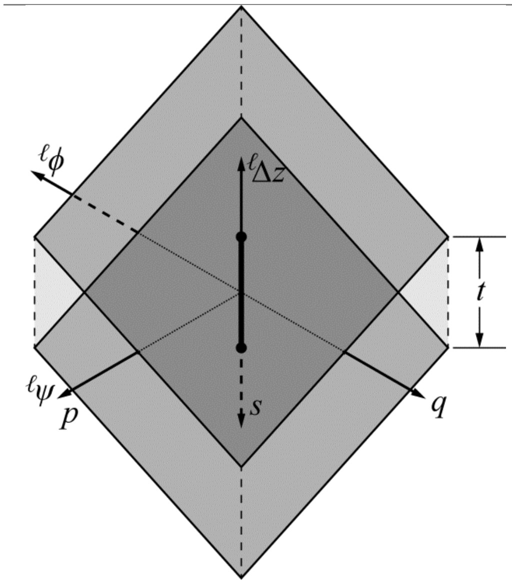
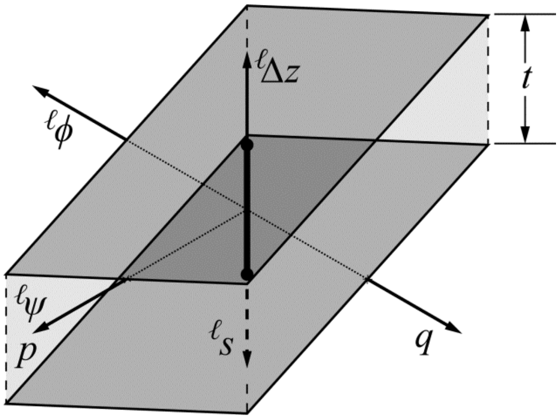
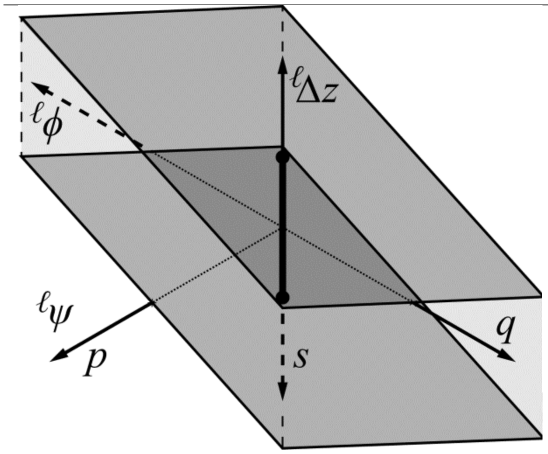
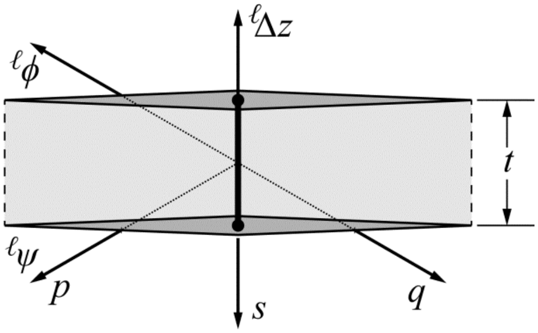


Figure 4.4 The 3-D cross-section of the T-Map for the Hexagonal Planar Segment

Table 4.1 Sheared T-Map Primitives

 $^{\ell}\phi$ $^{\ell}\Delta z$ t $^{\ell}\psi$ p s q	 $^{\ell}\Delta z$ t $^{\ell}\phi$ $^{\ell}\psi$ p $^{\ell}s$ q
$\vec{\boldsymbol{c}}(\boldsymbol{c_x}, \boldsymbol{c_y}) = (+,+)$ (P_1 in Figure. 4.2)	$\vec{c} = (+,-)$ (P_6)
 $^{\ell}\Delta z$ $^{\ell}\phi$ $^{\ell}\psi$ p s q	 $^{\ell}\Delta z$ $^{\ell}\phi$ t $^{\ell}\psi$ p s q
$\vec{\boldsymbol{c}} = (-,+)$ (P_3)	$\vec{c} = (-,-)$ (P_4)

The 3-D shape in Figure 4.4 is a cross-section of the 6-D T-Map representing all of the viable small displacement coordinates for the plane-segment in Figure 4.2. To get an idea of what the full 6-D T-Map is like, one can imagine extruding the 3-D T-Map along the coordinate axes of three additional higher dimensions, just as one would extrude a circle out of its 2-D space to get a 3-D circular cylinder. It is triply open in higher dimensions, corresponding to the lack of constraint in rotation about the z-axis and translations along the *x*- and *y*-axes of the LCS (Figure 4.2). Finally, in order to arrive at the T-Map in its traditional feature coordinates, the coordinate transformation, Equation A.1 described in Appendix A may be applied.

The resulting T-Map represents the allowable small displacements of the *nominal* planar segment within the boundaries of a fixed *tolerance zone.* However, as mentioned in Chapter 3, S-Maps for machinability should describe the condition that the minimum material envelope of the measured part should lie in the region where the machine tool will remove material—a one-sided boundary. Assuming that the machining operation is set to cut the material along the nominal feature as measured from the fixturing frame, the machinist is given the rest of the tolerance zone as allowance for setup and machining error. Thus the S-Map is created with the intention that material must lie beyond the MMC boundary of the tolerance zone, as shown in Figure 4.5. The procedure for creating the T-Map for a planar segment is followed, but with only one half-space representing the allowable adjustment at each vertex of the convex hull:

$$^{\ell}\left(a_{\phi}, a_{\psi}, a_{\theta}, a_{\Delta x}, a_{\Delta y}, a_{\Delta z}, b\right) = \left(c_y, -c_x, 0, 0, 0, 1, \frac{t}{2}\right) \tag{4.4}$$

The resulting intersection has half the number of half-spaces of the T-Map, and is open, as shown in Figure 4.6. This is because any adjustment that moves the measured planar feature further into the machining zone will still produce a completely machined feature.

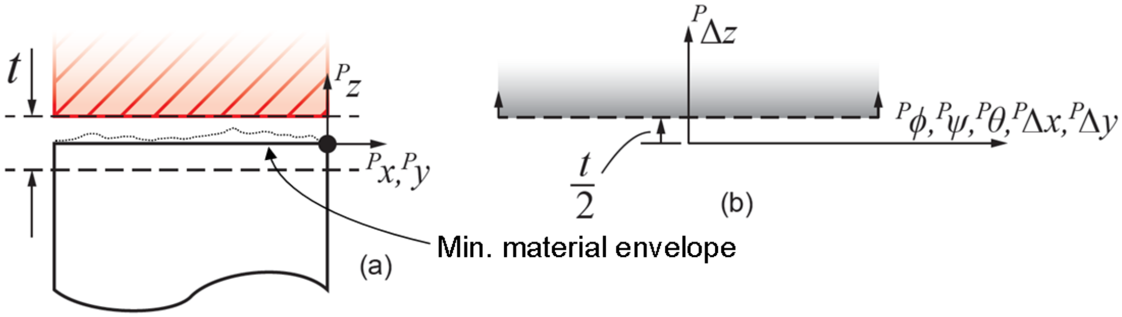


Figure 4.5 The S-Map Half-Space in the VCS Corresponding to the Machining Boundary

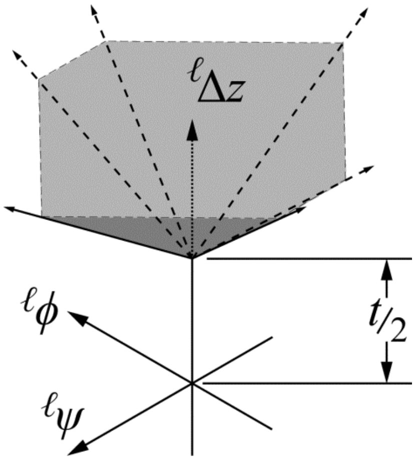


Figure 4.6 S-Map for Machinability for the Hexagonal Planar Face Before Shifting

4.2.2 **Shifting the S-Map Primitive** The local S-Map primitive is initially generated based on the location of the nominal feature in the CAD model, but the goal is to represent the allowable adjustments of the actual measured feature. In its unshifted location in small displacement space, the S-Map primitive represents the adjustments for a feature whose minimum material envelope corresponds with the nominal feature from the CAD model. In this case, the S-Map primitive does not contain the origin of the S-Map space, as shown in Figure 4.7(a), because in order to give the machinist some allowance one would have to move the TBM planar feature (and the rest of the part) further into the machining region so that it passes the MMC tolerance zone boundary (the new assumed machining boundary). The location of the minimum material boundary of the as-cast part is determined from point-cloud reduction, and then it is compared with the location of nominal feature. From this, we can determine the deviation in displacement coordinates that relate the position of the actual feature with respect to the nominal feature. This deviation corresponds to a shifting (translation) transformation in the S-Map space. In order to represent which displacements of the part will *return* the feature to an acceptable configuration for machining, the S-Map is then shifted in the *opposite* direction. For another part with a different deviation, as in Figure 4.7(b), notice that the corresponding S-Map primitive now includes the origin of the S-Map space because this instance of the feature lies in an acceptable configuration and does not require any further adjustment for complete machining. For a third instance of the part, such as in Figure 4.7(c), the feature of interest has an induced rotational deviation in a negative sense (clockwise) in addition to the translational deviation exhibited in the part from Figure 4.7(b). The S-Map is shifted in the positive sense because a positive rotation (counter-clockwise) would return the feature to an acceptable configuration.

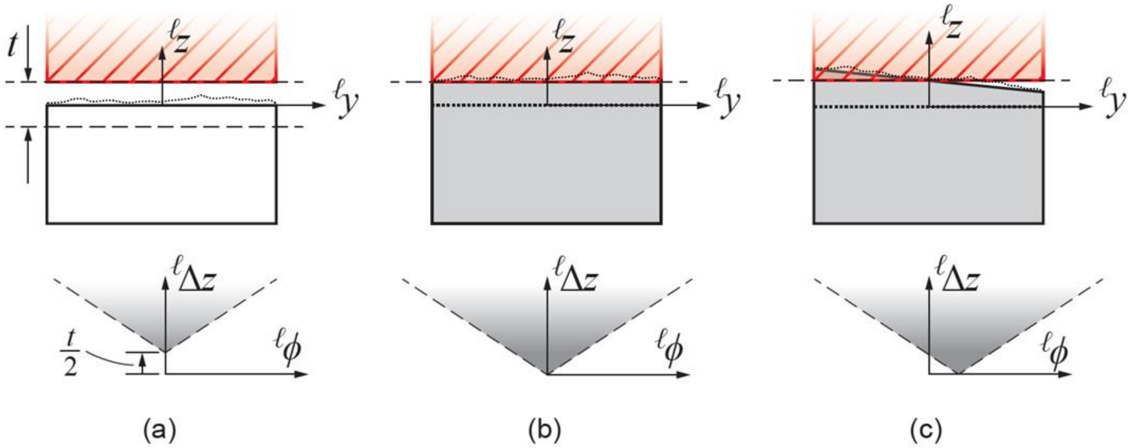


Figure 4.7 Shifting the S-Map Primitive for Different Material Boundaries on Separate Parts

To shift an S-Map primitive, the calculated shift (to account for the deviation of the minimum material boundary from the nominal feature) is added to each point in the space. In shifting half-spaces, the hyperplane normal vector does not change, only the distance value to the origin changed. To shift the half-spaces, Appendix B.5 demonstrates that for a shift vector in small displacement space, δ, the new b value is calculated as:

$$ {}^{i}b = {}^{j}b - \delta_{\phi}a_{\phi} - \delta_{\psi}a_{\psi} - \delta_{\theta}a_{\theta} - \delta_{\Delta x}a_{\Delta x} - \delta_{\Delta y}a_{\Delta y} - \delta_{\Delta z}a_{\Delta z} \qquad (4.5) $$

4.3 Generating T-Maps and S-Map Primitives for Cylindrical Surfaces

The process for creating T-Maps and S-Maps for a cylindrical feature follows the methods used in the case of the planar feature. However, unlike the planar feature, a cylinder is a feature of size. The variations in size of the as-cast cylindrical features will permit greater or lesser amounts of allowable adjustment of the casting in its fixture.

4.3.1 **The T-Map and the S-Map Primitive for an Axis** I will start by creating the T-Map for an axis by the new method of intersecting T-Map primitives; this method is an alternative to the intuitive method described in [14]. The tolerance zone for an axis is a narrow bounding cylinder with its diameter equal the tolerance value, as depicted in Figure 4.8(a). The LCS ℓxyz is oriented so that the z-axis is along the axis of the cylinder. The x- and y-axes point in orthogonal directions to the cylinder axis. For convention, the point in the center of the cylinder segment is chosen as the origin. To satisfy the tolerances applied, no point of the axis may lie beyond its cylindrical tolerance zone—Figure 4.8(b) shows an acceptable location for the axis. Like the case of the plane, the points furthest from the origin will limit the feature the most. For the cylinder, the two endpoints are the only two vertices for which a VCS will be created because they make up the 1-D convex hull. The VCS is parallel to the LCS and located at an endpoint of the axis of the cylindrical feature, e.g. as shown in Figure 4.8(a).

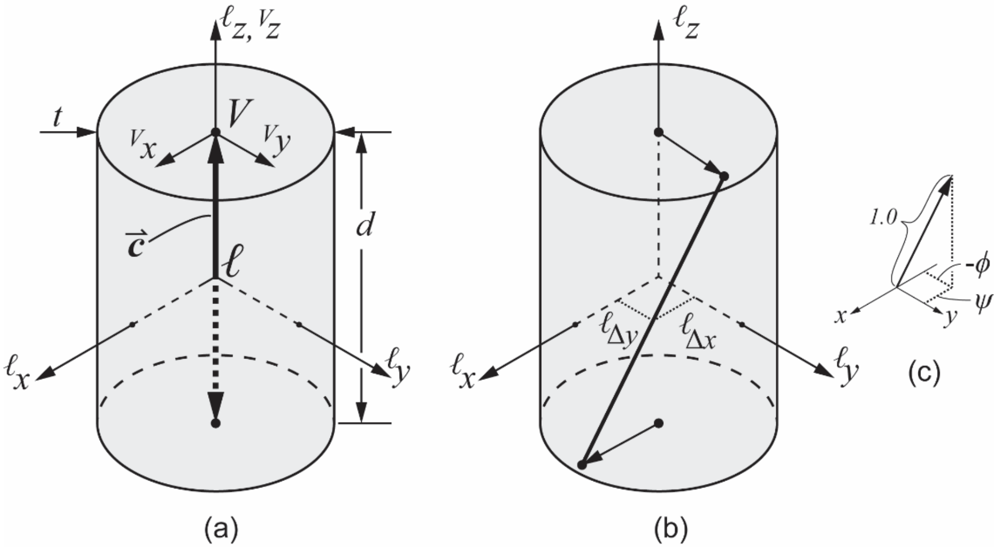


Figure 4.8 The Tolerance Zone of the Cylinder Axis

An endpoint of the axis, labelled V in Figure 4.8(a) and then isolated in Figure 4.9(a), represented in the VCS, is limited in translation in the $^V x$- and $^V y$- directions by the axis tolerance zone, but it is unlimited in all rotations and in translation along the z-axis. Therefore, the primitive for the single vertex is the open-ended 6-D shape, shown in Figure 4.9(b) where the four dimensions that lack constraint (${}^{V}\phi$, ${}^{V}\psi$, ${}^{V}\theta$, ${}^{V}\Delta z$) are shown on a single axis, but are actually four separate mutually perpendicular axes in 6-D. In order to represent this T-Map primitive with linear half-spaces, several are encircled tangent to the circular boundary of the S-Map primitive as an approximation, so that their normal vectors only have non-zero components ${}^{V}(a_{\phi}, \dots, a_{\Delta z})$ in the ${}^{V}\Delta x$ and ${}^{V}\Delta y$ directions, as shown in Figure 4.9(c).

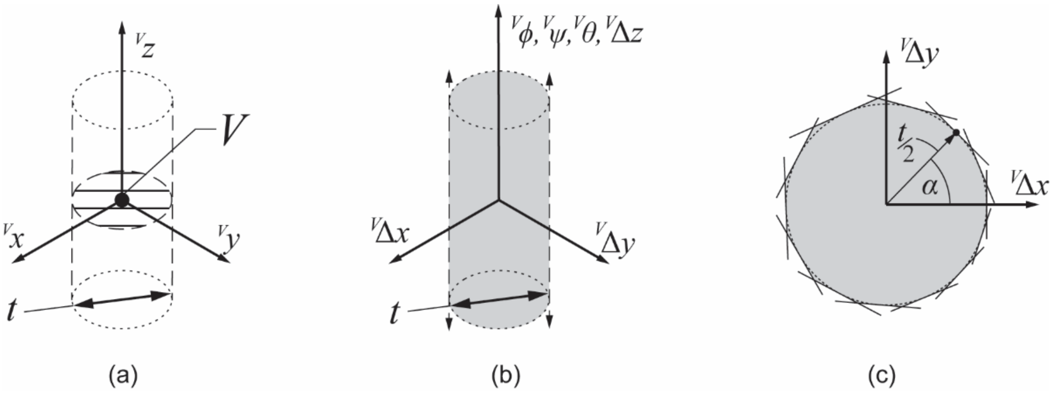


Figure 4.9 Creating the T-Map Primitive for a Point on the Axis of a Cylinder.

While the method of using tangent half-spaces slightly over-estimates the region, the numerical representation is simple. For a half-space that contacts the circle at a point along a radial line at angle α from the Δx coordinate axis as shown in Figure 4.9(c), the vector representation is:

$$ {}^{V}\left(a_{\phi}, a_{\psi}, a_{\theta}, a_{\Delta x}, a_{\Delta y}, a_{\Delta z}, b\right) = \left(0, 0, 0, \cos(\alpha), \sin(\alpha), 0, -\frac{t}{2}\right) \tag{4.6} $$

The simplified transformation back to the LCS derived in Appendix B is:

$$
\begin{aligned}
&{}^{\ell}\begin{bmatrix} a_{\phi} & a_{\psi} & a_{\theta} & a_{\Delta x} & a_{\Delta y} & a_{\Delta z} & b \end{bmatrix} \\
&\quad = {}^{V}\begin{bmatrix} \cos(\alpha) & \sin(\alpha) & \dfrac{t}{2} \end{bmatrix} \begin{bmatrix} 0 & c_z & 0 & 1 & 0 & 0 & 0 \\ -c_z & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned} \tag{4.7}
$$

and gives the local half-spaces defined as:

$$
\begin{aligned}
&{}^{\ell}\left(a_{\phi}, a_{\psi}, a_{\theta}, a_{\Delta x}, a_{\Delta y}, a_{\Delta z}, b\right) \\
&\quad = \left(-c_z \sin(\alpha), c_z \cos(\alpha), 0, \cos(\alpha), \sin(\alpha), 0, -\frac{t}{2}\right)
\end{aligned} \tag{4.8}
$$

This transformation shears the T-Map primitives so that the two intersect as shown in Figure 4.10(a)—in this case, the characteristic length is 1.0 length units. However, if a different characteristic length, such as for comparison with [14] where the angular dimensions are scaled by half the length of the cylinder, we arrive at the T-Map in Figure 4.10(b).

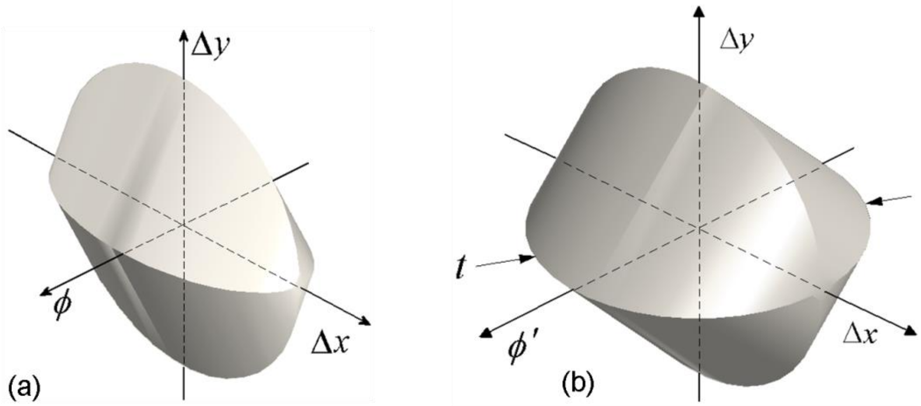


Figure 4.10 3-D Cross Section of the T-Map for an Axis

Now that a method is established for representing the allowable deviations of an axis within a cylindrical zone, the size of which corresponds to the assigned tolerance, an extension is made to creating the S-Map primitive shape. The S-Map primitive will describe the allowable positions of the axis of the cylindrical feature within a zone defined by the amount of material present for machining. For an external cylinder, such as a pin, the amount of allowance depends on the size of the as-cast feature. As shown in Figure 4.11(a), a larger as-cast feature allows the fixed machining boundary to be positioned within the material. Once again, depending on one's tolerance for risk, the machining boundary size, denoted as 'F_m', may be chosen as the size of the tolerance zone LMC or MMC boundary, or some other intermediate size—I will use the MMC size to give the most setup deviation allowance to the machinist. For parts with a larger measured feature size, denoted as 'F', there is a greater amount of positional variation allowed. We can establish an equivalent 'tolerance' that describes the size of the zone in which the axis of the cylinder may move about. For external features, the zone size is the equivalent tolerance, t_{eq}, calculated as:

$$t_{eq} = F - F_m. \tag{4.9}$$

Then, the local S-Map primitive for the machinability of a cylindrical feature is constructed in the same manner as the T-Map. Note that t_{eq} is negative for features that are smaller than the machining size. When half-spaces are intersected to create the S-Map primitive, no change to the computational process is necessary. The half-spaces will outline a circle of diameter $|t_{eq}|$, but all half-spaces will point outward, yielding no intersection. However, for the techniques described in chapter 5, it may still be valuable to represent this "negative" S-Map primitive.

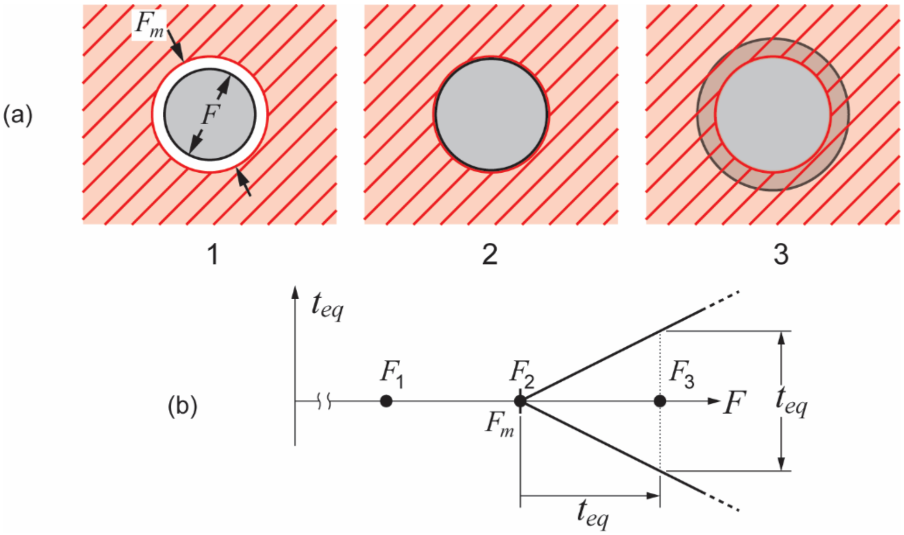


Figure 4.11 The Equivalent Tolerance for the S-Map of an External Cylinder

For an internal cylinder, such as any cylindrical hole, the amount of allowable displacement depends on the opposite relationship between the machining boundary size and the actual measured size of the feature. For parts cast with a larger hole, there is less available adjustment at the as-cast feature, as shown in Figure 4.12(a). However, a noticeable difference is that when no hole is cast into the part, there is no constraint on where the part must be positioned with respect to the non-existent feature—feature size is necessarily positive, or zero. The relationship between feature size '*F*' and equivalent tolerance for creating the S-Map primitive shape can be calculated as:

$$t_{eq} = \begin{cases} F_m - F & \text{if } \; F > 0 \\ \quad \infty & \text{if } \; F = 0 \end{cases}. \tag{4.10}$$

Again, I will let t_{eq} take a negative value for creating the S-Map primitive shape when *F* exceeds the machined feature size, F_m.

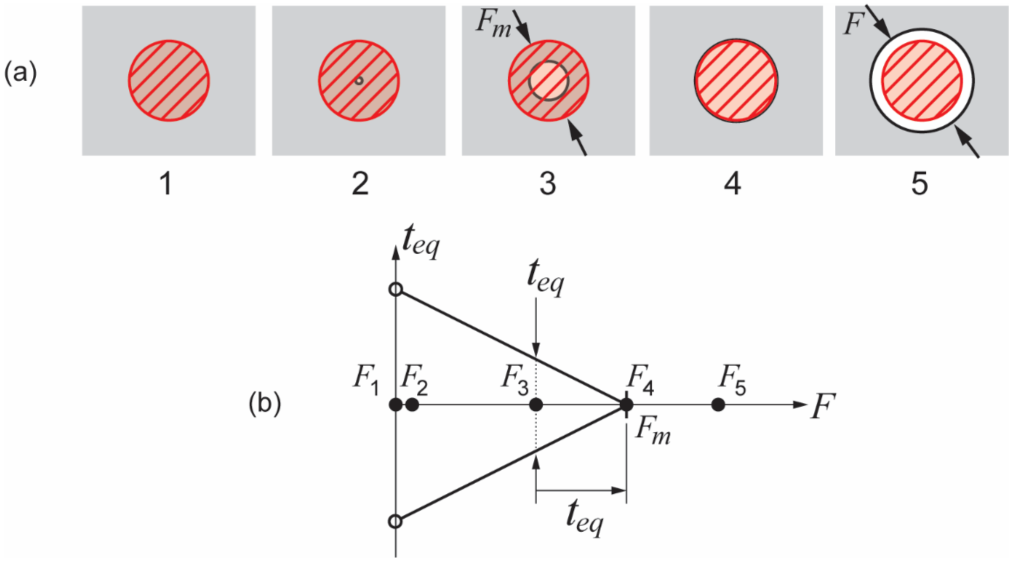


Figure 4.12 The Equivalent Tolerance for the S-Map of an Internal Cylinder

4.3.2 **Shifting the S-Map Primitive** Like the S-Map primitive for the plane, the S-Map primitive shape generated is then shifted by the opposite of the deviation measured from its nominal position. This opposite value represents the displacement of the entire casting that *returns* the minimum material envelope that represents the actual feature to the nominal location.

4.4 S-Map Primitives for Tolerances

The S-Map primitive for achieving geometric tolerances starts as a T-Map that describes allowable positions of the target feature within the tolerance zone. Like the S-Map for machinability, the T-Map is then shifted. However, instead of using the small displacement of the measured target feature, the T-Map is shifted by the displacement of the measured datum reference frame from its nominal position and orientation, again in the opposite direction. If the datums are not in their nominal locations, neither are the tolerance zones that are defined with respect to the datum reference frame. Therefore, the

S-Map describes adjustments that reposition the datums so that the machining operation will create a feature that lies within the tolerance zones established by the datum reference frame.

The measured datum reference frame is constructed based on point cloud data. During machining and inspection, datums are typically simulated by placing the part in contact with a gage surface. To simulate this from the point cloud data, the datum reference frame is created using the *maximum* material envelope of the points for each datum separately. The final datum reference frame is constructed using the new datums, respecting datum precedence as described in Section 4 of ASME Y14.5-2009 [3].

4.5 **Conclusion**

This chapter has described the standard method that will be used to create S-Map primitives (S-Maps for individual features) for planar and cylindrical features. The methods described deconstruct the process for generating the Tolerance-Map of any feature based on the available displacements of points in the tolerance zone. This process was then extended to S-Map generation by using the principles outlined in the previous chapter. The process described in this chapter should be extensible to features other than cylinders and planar faces. In this chapter, all S-Map primitives were created and shifted in their local coordinate systems (LCS) under the assumption that the feature data would be present for timely calculation. What was not explained is the method by which the data were brought to the LCS and how the interrelationship between features is maintained. The next step is to transform the S-Map primitives to be represented in the global coordinate system (GCS), and intersect them to find the common small displacements that satisfy the conditions imposed by each S-Map primitive. In the next chapter, I will suggest an order of operations for creating the final S-Map for an actual part. Based on how the measurements are made, I will explain the necessary transformations between local and global coordinate systems, and avoid unnecessary calculation steps.

CHAPTER 5 S-MAP OPERATIONS

In the previous chapter, I demonstrated the process for creating the S-Map primitive to describe the allowable deviation of a single feature from its nominal size and position when represented in its local coordinate system (LCS). Then, based on the displacement of the actual measured feature from the nominal, the S-Map was shifted in the *opposite* direction to represent small displacements that *return* the feature to its nominal position. At this state, the S-Map is a set of half-spaces represented in a Local Coordinate System (LCS) that have been shifted by a small displacement corresponding to the deviation of the feature represented in the LCS. However, this process may not correspond with the available information at the time of creating the S-Maps. In this chapter, I will suggest that there are equivalent methods for performing the calculations through to the final intersection in the Global Coordinate System (GCS). This allows the order of operations to be flexible so that operations are performed at the time when information is present. The hope is to avoid unnecessary transformations to and from coordinate systems.

5.1 **Equivalent Transformation and Shift Operations:**

The LCS is useful for creating the S-Map primitive shape in the fewest necessary dimensions, however it was not necessary to choose the coordinate system over any other. The coordinate system is an artificial construct that is used to establish a center of rotation, a position reference frame and directions from which the values of allowable rotation and translation are represented. When the S-Map primitive is transformed to a new coordinate system, it describes the same allowable displacements with a new center of rotation, and different coordinate axis directions. For this reason, the S-Map shape can be created in one coordinate system, transformed to a new coordinate system, and then shifted there. This is useful for performing operations where the data is present without unnecessary manipulation.

To illustrate this concept, it is useful to think of a model where no coordinate system is defined, where an actual part and nominal part are locked in relative position with their fixturing surfaces aligned. The S-Map primitive for a given feature is the set of small displacements, $\{s\}$, that represents the allowable adjustments of the part for complete machining of the nominal feature, $\{s_n\}$, plus the shift that represents displacement of the part that returns the measured minimum material envelope to the nominal, δ. This can be represented as the formula:

$$s = s_n + \delta \tag{5.1}$$

The displacement (shift) for the machinability condition is calculated as the *opposite* of the displacement of measured feature, f, from its nominal or theoretical location, f_n:

$$\delta_m = -(f - f_n) = (f_n - f) \tag{5.2}$$

where the minus sign between two features is meant to represent the calculation of the relative small displacement from the subtrahend feature to the minuend feature.

For tolerances with respect to un-machined datums, the nominal tolerance zone of a feature is displaced by the small displacement that would return the datum reference frame created by the measured features, D, to its nominal location, D_n. D is constructed with the maximum material envelopes of the point clouds that represent the as-cast datums, and by following the datum precedence specified in the tolerance control frame, as explained in [4]. The displacement of the actual datum reference from its nominal location, δ_t, is shown notionally in Equation 5.3.

$$\delta_t = -(D - D_n) = (D_n - D) \tag{5.3}$$

The only restriction for performing the three operations listed above is that the features and displacements must be represented in the same coordinate system at calculation time. In order to transform between 3-D coordinate frames, small displacements are transformed with the screw transformation matrix, $\$\$_{ij}$, defined in equation 4.52 of [1], and features are transformed with the appropriate transformation that I will generally call T_{ij}, which may transform position vectors, direction vectors, or any other feature coordinates. For a list of the transformations used, see Appendix B. These two transformations, in the given notation, take a small displacement or feature represented in the *j*-frame, and represents it in the *i*-frame.

5.2 The S-Map Generation Process for Implementation

It would be beneficial to perform the minimum amount of calculations that will represent an S-Map in the GCS in a robust manner. I mention robustness only to aid implementation of the shift (δ) calculations. For this calculation, it is preferred to represent the nominal and measured features in their LCS, where the number of invariant coordinates is a maximum. For our hypothetical part, I will make a temporary assumption that the unadjusted fixture is the body that establishes the GCS, and this is used to make measurements of the TBM features, and datum features on the part, giving ${}^{G}f$ and ${}^{G}D$. In addition, the CAD model communicates the nominal features so that they are represented in the GCS, giving ${}^{G}f_n$.

The goal is to calculate the S-Map primitive represented in the GCS in the fewest steps. The full expansion of the formulae can be found in Appendix C. With the above notation, and using the pre-superscripts G, ℓ, and d to denote the GCS, TBM feature LCS and datum reference frame respectively, the formulae can be written as follows.

For the machinability S-Map primitives:

$${}^{G}s = \$\$_{G\ell}\left({}^{\ell}s_n + \left({}^{\ell}f_n - T_{\ell G}\,{}^{G}f \right)\right), \tag{5.4}$$

and for the tolerance zone S-Map primitives:

$${}^{G}s = \$\$_{G\ell}\,{}^{\ell}s_n + \$\$_{Gd}\left({}^{d}D_n - T_{dG}\,{}^{G}D \right) \tag{5.5}$$

For machinability, the measured TBM feature in the GCS is transformed to the LCS where it is compared with the nominal feature to get the shift value. The S-Map primitive shape for the machinability is created in the LCS, the shift is applied, and then it is transformed to the GCS.

For satisfying tolerances, the T-Map for the tolerance zone of the nominal target feature is created in the LCS and transformed to the GCS. The datum features represented in the GCS are transformed to the nominal datum reference frame where they are compared with the nominal DRF to get the local shift value. This shift is transformed back to the GCS where it is applied to the T-Map, giving the S-Map primitive for the specified tolerance.

The pseudo-code for performing the operations on a single part with the results of each step is as follows:

```
START
Measure the part in GCS, or transform the measured features to the GCS: ${}^{G}f$, ${}^{G}D$
For each TBM feature:
    Calculate the feature transformation from the GCS to LCS: $T_{\ell G}$
    Calculate the screw transformation from the LCS to GCS: $\$\$_{G\ell}$
    Calculate the S-Map primitive for the TBM feature in the LCS: ${}^{\ell}s_n$
    Transform feature measured in GCS to LCS: ${}^{\ell}f = T_{\ell G}\,{}^{G}f$
    Calculate the shift in the LCS: ${}^{\ell}\delta =- \left( {}^{\ell}f - {}^{\ell}f_n \right)$
    Shift the S-Map primitive in the LCS: ${}^{\ell}s = {}^{\ell}s_n + {}^{\ell}\delta$
    Transform the shifted S-Map primitive to the GCS: ${}^{G}s = \$\$_{G\ell}\,{}^{\ell}s$
For each geometric tolerance applied to a TBM feature from an un-machined datum:
    Calculate the transformation from the GCS to the datum LCS: $T_{dG}$
    Calculate the screw transformation from the datum LCS to the GCS: $\$\$_{Gd}$
    Calculate the screw transformation from the target feature LCS to GCS: $\$\$_{G\ell}$
    Calculate the S-Map primitive shape for the target feature in the feature LCS: ${}^{\ell}s_n$
    Transform datums measured in GCS to LCS: ${}^{d}D = T_{dG}\,{}^{G}D$
    Calculate the shift in the datum LCS: ${}^{d}\delta =- \left( {}^{d}D - {}^{d}D_n \right)$
    Transform the shift vector from the datum LCS to the GCS: ${}^{G}\delta = \$\$_{Gd}\,{}^{d}\delta$
    Transform the local S-Map primitive to the GCS: ${}^{G}s_n = \$\$_{G\ell}\,{}^{\ell}s_n$
    Shift the S-Map primitive in the GCS: ${}^{G}s = {}^{G}s_n + {}^{G}\delta$
Intersect all S-Map primitives, and find the 'setup point'
END
```

Of course this is the representation of the transformations of points intended to show the logical order of operations. Where necessary, the half-space transformations may be substituted. With all S-Map primitives created and represented in the GCS, we may use the term “small body displacement”, or its foreshortened, SBD, to signify that the S-Map primitives truly describe the displacement of the part and all of its features as one entity. The set of common SBDs that exist in all of the S-Map primitives make up the S-Map. To calculate the common set of SBDs, the intersection operation is performed over all S-Map primitives, and from this, a single point is chosen as the ‘setup point’.

5.3 S-Map Primitive Intersection and Setup Point Selection

The intersection is performed through a half-space intersection scheme implemented in the open-source software, Qhull [15], and its methods are described in chapter 7 of [13]. At this point, the S-Map is represented as a set of half-spaces, with likely many redundant half-space hyperplanes that will not contribute to the final intersection shape. Given a single point that exists in all half-spaces in the set, the intersection process and filtering of redundant half-spaces are performed with Qhull. However, finding this point is not trivial—had the point been known *a priori*, there would have been no reason to create the S-Map as this point corresponds to a viable machining setup. As the equations of half-spaces are formulated as feasibility regions of points in 6-D space, I will solve for the setup-point as an optimization problem using linear programming where the objective is to find a point that is furthest into the feasible interior of the 6-D S-Map shape.

5.3.1 Linear Programming to Find the Setup Point The setup point is the single 6-D SBD point that is ultimately chosen from the S-Map shape. Because the half-spaces that limit the final intersected shape will represent setups that position a casting with at least one feature in a maximally displaced location, an algorithm to find the setup point should pick one that is distant from all the boundary faces of the S-Map—one that is well within the interior. As a diagnostic for cast parts with excessive deviation, where no S-Map intersection exists, there will not exist a point that lies within all half-spaces. The method for finding the setup-point may be framed as an optimization problem. For the set of points in the S-Map, I will choose a point whose minimum distance to any half-space is maximal. In *n* fully bounded dimensions, this point will be located a distance d_O away from a minimum of *n*+1 boundaries that define a simplex for non-degenerate cases. A 2-D example is depicted in Figure 5.1(a).

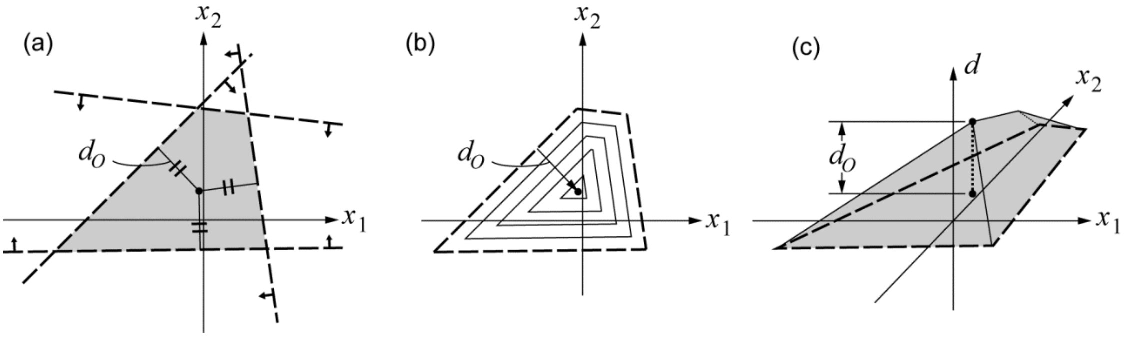


Figure 5.1 Method for Calculating the Setup Point

There are a few interpretations of the method used to find the setup point. When it was devised, the goal was to find the point that is furthest from the hyperplanes that define the shell of the S-Map. In order to find this point, as depicted in Figure 5.1(b), all the boundaries are moved inward at equal normal rates. As the shape decreases in size, in general it also decreases in number of faces. In the *n*-dimensional case, the shape becomes an *n*+1-sided simplex (in Figure 5.1, a triangle) before collapsing into a single point. The mathematics and support for the method are described as follows:

A feasible region is the set of points that exist within all of the constraining half-spaces. A point *x*(*x*$_i$) is feasible if it satisfies the linear inequalities for all half-spaces that define the region. For an initially feasible region, to find the optimal point, equally shift the half-spaces in the opposite direction of the outward normal until there exists only one feasible point. The half-space is defined as the points that satisfy the inequality:

$$\sum_{i=1}^{n} a_i x_i + b \leq 0. \tag{5.6}$$

The absolute value of *b* is the distance between the origin and the bounding hyperplane of a half-space when $\sqrt{\sum_{i=1}^{n} a_i^2} = 1$. For each half-space that contains the origin, b is *negative*. Therefore, the hyperplane shifts a positive distance d towards the origin as b *increases*. The shifted hyperplane has the form:

$$\sum_{i=1}^{n} a_i x_i + b' \leq 0 \text{ where } b' = b + d. \tag{5.7}$$

Consider a feasible region defined by a set of *N* half-spaces that bound the space. The point *x*(*x*$_i$) is feasible if it satisfies each of the *j* = 1, …, *N* half-space inequalities:

$$\sum_{i=1}^{n} a_{ij} x_i + b_j' \leq 0, \text{ or } \sum_{i=1}^{n} a_{ij} x_i + b_j + d \leq 0. \tag{5.8}$$

Thus the optimization problem is as follows:

$$\text{maximize} \qquad f(x, d) = d \tag{5.9}$$

$$\text{subject to:} \qquad a_j^{\mathbf{T}} x + b_j + d \leq 0, j = 1, \ldots, N \tag{5.10}$$

In the case of setup point selection, the variable $x \in \mathbf{R}^n$ is the 6-D vector of small displacement variables, $(\phi, \psi, \theta, \Delta x, \Delta y, \Delta z)$ and $d \in \mathbf{R}$ is the hyperplane shift distance. Each a_j is a vector of the normalized half-space hyperplane outward normal vector components ${}^{G}(a_\phi, a_{\psi,} a_\theta, a_{\Delta x}, a_{\Delta y}, a_{\Delta z})$ with a corresponding scalar b_j that is the initial distance from the hyperplane to the origin.

One can imagine the problem in the (n+1)-dimension space, as depicted in the 3-D space of Figure 5.1(c). The point that is chosen corresponds to the apex of the pyramidal shape that is created. The slanted constraint surfaces all rise at an angle of 45° from the base because as d increases, the corresponding n-D half-spaces move inward by distance d along their normal vector direction. As final confirmation of their similarity, we can equate Figure 5.1(b) and (c). Just as a cartographer depicts mountainous regions in a topographic map, Figure 5.1(b) can be interpreted to show the equal-elevation contours of the n-dimensional mountain, shown in 3-D in Figure 5.1(c). All imaginations of scaling higher-dimension mountains aside, this diagram shows the promise of the technique.

There are other benefits of representing the problem as the linear programming model. If no feasible region exists for the initial set of half-spaces, the linear programming method will return a negative value for d, indicating that the half-spaces must be moved *outward* for a feasible point to emerge. The half-spaces that define the simplex of half-spaces that ultimately limit the optimal point may be retraced to their corresponding S-Map primitives to identify which features and constraints limit the possible SBDs most. The distance value d, may also be interpreted as a metric for the volume of the S-Map, and therefore the amount of displacement available.

However, there are a few limitations where unfavorable, though correct, results may arise. If the initial n-dimension space is not entirely bounded, an unusable point at infinity may arise. To handle this, I include artificial constraints that limit the maximum and minimum of the x_i to values that are an order of magnitude or two greater than the tolerance values. In a similar issue, for a fully bounded feasible region with a pair of parallel constraints that are significantly closer together than the rest, a degenerate case appears where there exists more than one point with the same optimal value. An easy to imagine example is a rectangular region in which a line segment of points are equally optimal. This degenerate case can take on the form of a 2-flat (a line) segment up to an (n-1)-flat (a hyperplane) segment of points whose minimum distance to a constraint is equally maximal, and therefore equally optimal. In these cases, the simplex method will end at a vertex of the degenerate shape. To avoid this, 'interior point' methods based on partial derivatives of the constraints are used to estimate a point that is furthest between all constraints. This method is beyond the scope of explanation for this thesis, but the open source code CLP (COIN-OR Linear Programming) [16], provides the required functionality.

Thus, the setup point is found by equalizing the distance to the closest, or most critical, S-Map half-spaces.

5.3.2 **Half-Space Intersection** The methods for performing half-space intersection are fully described in detail in chapter 7 of [13] and implemented in the open source code Qhull [15]. Once a feasible point is identified, half-spaces are transformed into their dual representation as homogeneous points. The convex hull of the dual points is created which eliminates interior points that correspond to redundant half-spaces. Dual points of the convex hull are transformed back to their primal half-spaces, and the generated dual facets are transformed back to vertices in the primal model. This establishes the HV-representation (half-spaces and vertices) of the S-Map.

5.4 Using the Small Body Displacement for Setting up the Part

It is handy to assume that the measurements were all made in the GCS with the fixturing features aligned with their respective feature simulators. However, this may not necessarily be a practical assumption for companies that would like to measure the part in one geographical location and machine it in another. First, I will describe the *in-situ* methods that may be used for making adjustments when the part is measured in the exact location that it will also be machined. Then, I will describe the necessary steps to simulate the fixturing coordinate system (FxCS) so that parts may be measured offsite, then placed on the machining tool bed with the adjustments already made.

5.4.1 ***In-situ*** **Measurement** For production facilities that have an integrated tooling platform and coordinate measurement machine, it may be possible to have the part sitting on an adaptive fixture that can provide small adjustments. Calibration should first synchronize the CMM to the GCS of the machining tool. When the part is roughly aligned on the fixture and then measured, the S-Map will generate the SBD that will align the part to the GCS of the machine tool. The benefit of this method is that successive measurements may be made to verify part alignment. The downside is that the measurement time is extra time that the machining tool is sitting idle.

5.4.2 **Offsite Measurement** In order to perform offsite measurement, the Fixturing Coordinate System (FxCS) must be recreated from the specific measured features that correspond to the surfaces in which the fixture will make contact with the part. The transformation that aligns the FxCS features is applied to all features of the part. Then, the features are transformed to be represented in the GCS by the nominal difference between the FxCS and the GCS if the two coordinate systems do not coincide. This is because the nominal feature information from CAD will relate all features in the

GCS. The S-Map is then used to calculate the setup point in the GCS. The fixture data must also be communicated to calculate the adjustments that will create the displacement of the part. For this process, there must be a standard calibration of the fixture that ‘zeros’ the adjustors before the adjustments are made. A gage part with special alignment features might be used to accomplish this. The benefit of this method is that the adjustments can be calculated while the part is not sitting on the tool bed. The downside is that no validation of part alignment can be performed before executing the machining operation. In addition, the accuracy of alignment depends on the accuracy of the gage part and the ability of the operator to calibrate the fixture.

5.5 Adjusting the As-Cast Part

The final step is to convert the SBD of the setup point into adjustments of the fixture. In order to automate this, one must know the geometry of the fixture, the directions of action of each adjustor, and the types of displacements possible with the types of adjustors specified. The full scope of such an endeavor is beyond that of this thesis, but I will highlight the important aspects for implementation and further study.

The trivial case is the fixture where no adjustment can be made to the fixing features. This corresponds with a fixture that is locked in place with respect to the machining tool. The cast part is set down in a fixed location so that its fixing features are forced into contact with the surfaces, and then the machining operation is executed. This type of fixture is very efficient when the casting process is well-studied with most variations accounted for at other stages of the process. In this case, the S-Map is only useful as a diagnostic for feasibility. If the S-Map contains the zero-displacement SBD, machining can be performed successfully. However, even the simplest fixtures that are commonly used in machine shops can do better.

For example, adding the ability to translate the tool bed in three perpendicular directions with respect to the machine tool affords the use a subset of the S-Map corresponding to the condition that the values of the angular coordinates of the SBD must be equal to 0. The S-Map space is collapsed to a 3-D cross-section in Δx, Δy, Δz coordinates. For example, a typical standup milling machine with a rigid fixture attached to its tool bed will allow x-, y- and z- adjustment of the part with respect to the mill axis. By adding shims to the fixing surfaces, even small rotations are producible, and the restriction on the S-Map dimension may be lifted. It may even be possible to create an adaptive fixture for lathe turning operations. The process for this will involve aligning the displaced axis with the lathe’s axis of rotation, and choosing a new zero relative location for the axial coordinate. There are fixtures that will depend on the sizes of fixing features such as V-blocks for positioning cylindrical features or positioning pins for locating cylindrical holes. Thus, the process for automating such a process of determining the adjustments is very involved.

It is worth mentioning that for every adjustment of the part with respect to the machine tool, there is an equivalent adjustment of the tool with respect to the part that will requires the opposite direction of adjustment. For example, a multi-axis CNC mill with a fixed tool bed may be re-zeroed to correspond with the new displaced coordinate system by displacing its coordinate system by the opposite of the small *body* displacement of the chosen setup point. In this case, the equivalent of the standup milling machine tool bed is a 3-axis gantry-arm type CNC milling machine whose coordinate axes may be redefined to that of the translationally displaced coordinate system.

5.5.1 **An Adaptive Fixture for Castings** I would like to highlight an example of an adaptive fixture that affords the use of all six displacement components of the SBD and that incorporates principles of standard casting practice. The assumption for such a fixture is that the part is moved with respect to the fixed machining operations. As described in [17], castings are sometimes made with location points that are used to align a part with a fixed coordinate system. The location points follow the “3-2-1” principle where the first fixturing datum plane, F, is created with three points, the second datum plane , G, is defined with the perpendicular to the first plane and two points in the plane, and the third datum, H, is perpendicular to both the first and second datums and located with one point. At each of these location points, we can assign a ball-ended adjustor that will move in a perpendicular direction to the datum plane it helps define.

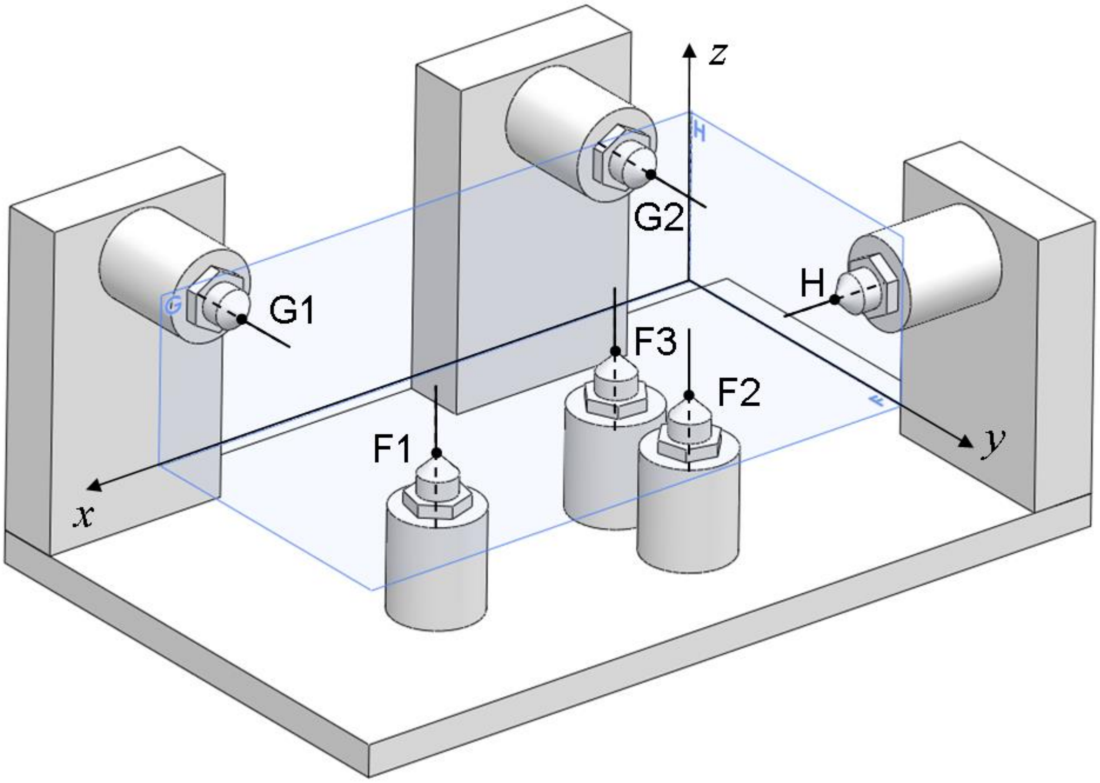


Figure 5.2 An Adaptive Fixture for Manipulating TBM Parts

At setup, the casting is forced into contact with all six adjustors. As the adjustors are moved, the ball end in contact with a planar feature allows sliding, but for modeling, the assumption is that the points of contact on the adjustor and the part will remain fixed together. Because the part will only be displaced by a small amount, I will only consider the instantaneous motion of each contact point when a SBD is applied. This may be modeled using the robotics notation for a robot with parallel actuators, as described in chapter 8 of [1] where the input is a twist corresponding to displacements instead of velocities, and the outputs are the linear actuator displacements instead of speeds. The part will rest on location points F1, F2, F3, G1, G2, and H with nominal locations in the FxCS:

$$\begin{aligned} F1(x,y,z) &= (f_{1x}, f_{1y}, 0) \\ F2(x,y,z) &= (f_{2x}, f_{2y}, 0) \\ F3(x,y,z) &= (f_{3x}, f_{3y}, 0) \\ G1(x,y,z) &= (g_{1x}, 0, g_{1z}) \\ G2(x,y,z) &= (g_{2x}, 0, g_{2z}) \\ H(x,y,z) &= \left(0, h_y, h_z\right) \end{aligned} \tag{5.11}$$

From their locations and directions of action, the screws ($\$_\mathrm{i}$) that define the actuators can be expressed in normalized Plücker line coordinates ($L,M,N;P,Q,R$) such that $L^2 + M^2 + N^2 = 1$.

$$\begin{aligned} \$_{F1} &= (0,0,1; f_{1y}, -f_{1x}, 0) \\ \$_{F2} &= (0,0,1; f_{2y}, -f_{2x}, 0) \\ \$_{F3} &= (0,0,1; f_{3y}, -f_{3x}, 0) \\ \$_{G1} &= (0,1,0; -g_{1z}, 0, g_{1x}) \\ \$_{G2} &= (0,1,0; -g_{2z}, 0, g_{2x}) \\ \$_{H} &= \left(1,0,0; 0, h_z, -h_y\right) \end{aligned} \tag{5.12}$$

Then, in order to calculate the adjustments, we build the Jacobian matrix J represented in the FxCS with the screws ($\$_i$) as row vectors, re-ordered as ($P,Q,R;L,M,N$).

$$[J] = \begin{bmatrix} f_{1y} & -f_{1x} & 0 & 0 & 0 & 1 \\ f_{2y} & -f_{2y} & 0 & 0 & 0 & 1 \\ f_{3y} & -f_{3y} & 0 & 0 & 0 & 1 \\ -g_{1z} & 0 & g_{1x} & 0 & 1 & 0 \\ -g_{2z} & 0 & g_{2x} & 0 & 1 & 0 \\ 0 & h_z & -h_y & 1 & 0 & 0 \end{bmatrix} \tag{5.13}$$

Then, the calculation is:

$$\begin{bmatrix} \Delta F1 \\ \Delta F2 \\ \Delta F3 \\ \Delta G1 \\ \Delta G2 \\ \Delta H \end{bmatrix} = \begin{bmatrix} f_{1y} & -f_{1x} & 0 & 0 & 0 & 1 \\ f_{2y} & -f_{2y} & 0 & 0 & 0 & 1 \\ f_{3y} & -f_{3y} & 0 & 0 & 0 & 1 \\ -g_{1z} & 0 & g_{1x} & 0 & 1 & 0 \\ -g_{2z} & 0 & g_{2x} & 0 & 1 & 0 \\ 0 & h_z & -h_y & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \phi \\ \psi \\ \theta \\ \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} \tag{5.14}$$

The matrix multiplication of the line coordinates as row vectors with the small body displacement in Equation 5.13 gives the component of the displacement of 3-D space along each actuator line, which is equivalent to the adjustment at each actuator.

Thus, given a small body displacement represented in the GCS, it can be transformed to the FxCS and then converted into adjustment values. The benefit of performing the operation as such is that the fixture is defined in its local coordinate system, which simplifies its representation.

CHAPTER 6 SOFTWARE IMPLEMENTATION

The processes described in this thesis have been implemented in C++ class structures that may be used for calculation and 3D visualization. The benefit of using C++ is its object-oriented nature which simplifies operation through abstraction of data types. Two main classes were implemented in order to support operations up through small body displacement calculation: local geometry ('*LocalGeom*' base class), and T-Maps ('*TMap*' base class). The optimization and intersection schemes were implemented with the use of open source C++ code from both Qhull [15], and COIN-OR Linear Programming (CLP) [16].

6.1 **Class Structures:**

Because intersection is the only necessary operation, only the half-space description of the S-Maps and T-Maps is stored in the *TMap* class. Half-spaces are represented as an array of 7 characters representing the ordered set for a 6-D half-space in the '*halfspace6*' class. This class also includes a method for normalization.

Table 6.1 The halfspace6 Class Structure

Class: *halfspace6* (6-d half-space)	
Variables	
coeff	The ordered coefficients for halfspaces in 6-D. $(a_\phi \quad a_\psi \quad a_\theta \quad a_{\Delta x} \quad a_{\Delta y} \quad a_{\Delta z} \quad b)$
Methods	
normalize	Divides each coefficient by the length of the normal vector

The structure of the data was designed to try to group common functionality into base classes from which the specific derived classes are created. The *LocalGeom* class is the base class for representing geometry in its local coordinate system (LCS) it contains information about the relative position vector and rotation matrix from the GCS. The *TMap* base class encapsulates the main T-Map functionality, and allows for the storage of half-spaces. The *LocalPlnTMap* class creates either open or closed S-Maps for a plane, represented in its LCS. It inherits traits from both the *TMap* class and the *PlnSeg* (plane segment) class (which inherits the *LocalGeom* class) to create an object that has the T-Map structure combined with the structure of the local geometry to perform the generation of a local T-Map and to transform it to the GCS. Table 6.2 and Table 6.3 show the relevant variables and methods in for the *LocalGeom* and *TMap* classes. The class structures of the derived *LocalGeom* and *TMap* classes, shown in Figure 6.1, can be found in Appendix D. Note that arrows in Figure 6.1 can be thought of as "inherits" or "is a" e.g. "*LocalPlnTMap* is a *TMap* and inherits *PlnSeg*."

Table 6.2 The Variables and Methods of the *LocalGeom* Base Class

Base Class: *LocalGeom*	
Variables	
R	Container for the rotation matrix from the GCS to the LCS
Cx, Cy, *Cz*	Containers for the relative position vector from the GCS to the LCS
X	The cross product matrix for screw transformations
Methods	
setCvect	Sets the values of Cx, Cy, Cz, and populates the *X* matrix
get/setRmat	Gets/sets the *R* matrix

Table 6.3 The Variables and Methods of the *TMap* Base Class

Base Class: *TMap*	
Variables	
hspacelist	List of *halfspace6* objects that make up the boundaries of the T-Map
TMapQhull	A Qhull object for computing and storing the intersection
optpoint	The central point of the intersection shape
Methods	
addHalfspace	Adds a *halfspace6* to *hspacelist*
addTMap	Adds the *hspacelist* of another *TMap* object to the stored *hspacelist*
calcInteralPoint	Calculates the central point of the *TMap* and stores it in *optpoint*
scaleAngular	Scales the angular dimensions by a value (characteristic length)
createOpt3Dsection	Creates a 3-D cross section of the 6-D T-Map with 3 coordinates set to their optimal value
createZero3Dsection	Creates a 3-D cross section of the 6-D T-Map with 3 coordinates set to zero

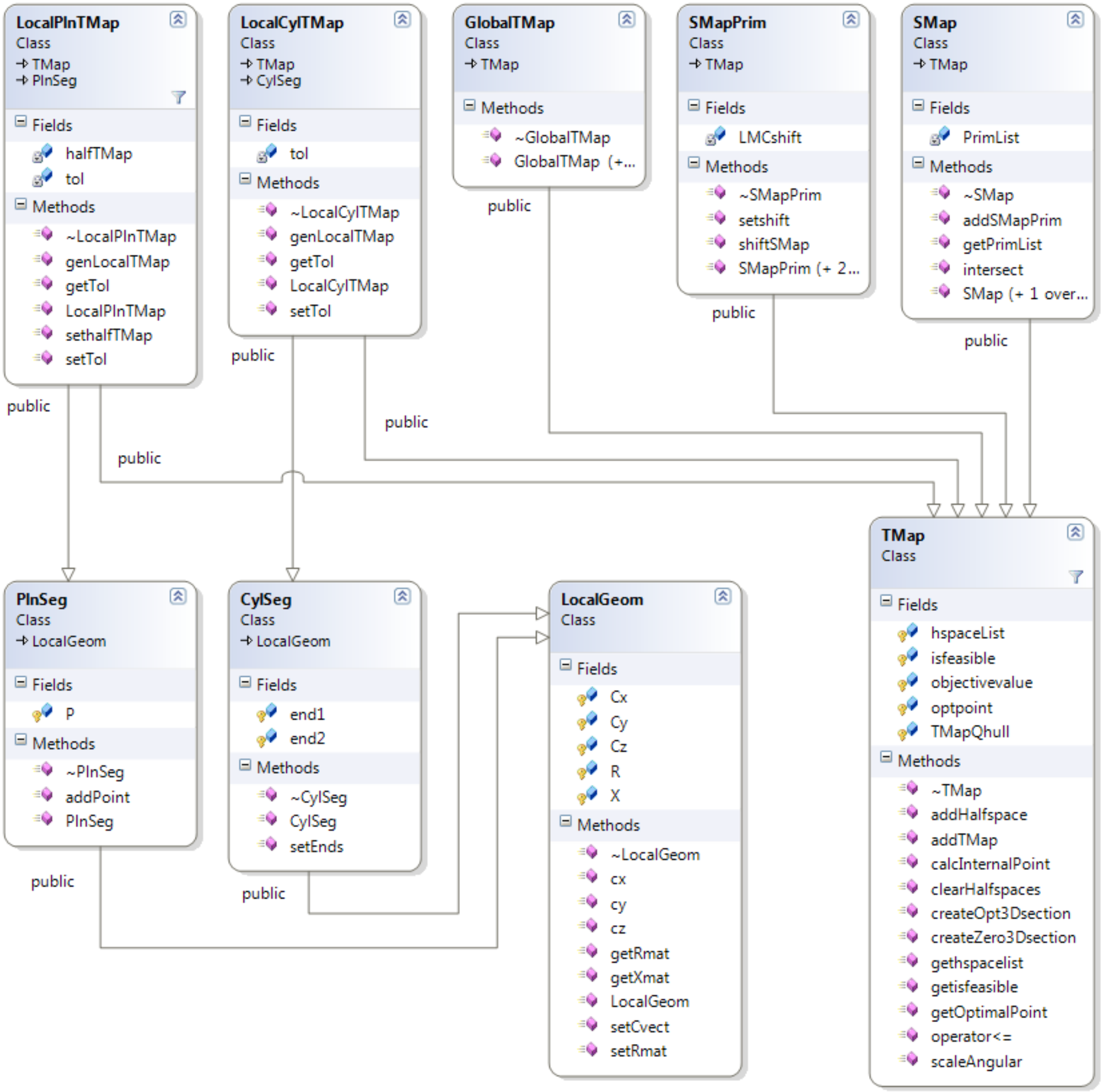


Figure 6.1 The Class Structures for 'TMap' and 'LocalGeom' objects

6.2 Computational Geometry Algorithms Implementation

The following open source programs were used in creating the intersection of S-Maps:

6.2.1 **Setup Point selection with CLP:** The Computational Infrastructure for Operations Research (COIN-OR) is an open-source software bank that provides open-source code for the operations research community [16]. The goal of the project is to provide version control to maintain working distributions of optimization code. The COIN-OR Linear Programming (CLP) open-source code includes a simplex algorithm and a barrier method interior point method for solving linear programming. The interior point method was chosen for its ability to handle the degenerate cases discussed in chapter 5. It works by traversing the interior of the "feasible space" [18], rather than traversing the facets of the convex boundary shape as the simplex algorithm does. The algorithms implemented for setup point selection work for n-dimensional problems, as described in chapter 5.

6.2.2 **Intersection with Qhull** Qhull[15] is the open-source code that can create convex hulls, half-space intersections, and other geometric algorithms with an n-dimensional algorithm. Its developers do not recommend its use above 8-D because the size of the output for the convex hull for number of points n in dimension d is estimated at $n^{\lfloor d/2 \rfloor}$ facets [15]. In order to perform half-space intersection, the Qhull algorithm dictates that an internal point must first be specified. Therefore, output from the CLP code feeds into the Qhull algorithm for intersection. The main use of Qhull is to trim redundant half-spaces—ones that do not contribute to the final intersection—from the final S-Map. It is also useful for visualizing 3-D cross-sections. By taking a 3-D cross section of the 6-D space, Qhull can then find the 3-D vertices and facets which can be plotted in other software, as shown in Figure 6.2.

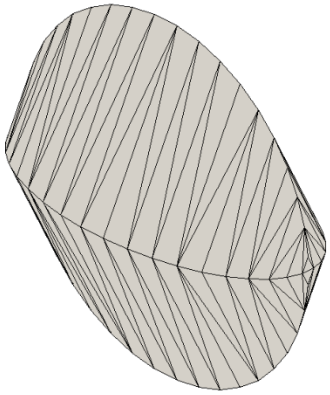


Figure 6.2 The Qhull-Generated T-Map for an Axis in Small Displacement Coordinates

CHAPTER 7 TESTING THE SETUP-MAP

To show the workings of the S-Map, the model of a hypothetical engine block with three in-line cylinders, shown in Figure 7.1, will be used for demonstration. Red surfaces in the depiction are the to-be-machined features. Seven geometric tolerances are specified and listed in Table 7.1 since the tolerance applied to the cylinders is replicated twice. The part would be placed on an adaptive fixture, such as the one shown in Figure 5.2 from Chapter 5, and then machined. Two hypothetical parts will demonstrate the ability of the software to choose a setup point that will align each as necessary. In a limited demonstration, I will then show that when the displacement is applied to the part, there is a significant reduction in the amount of additional adjustment necessary.

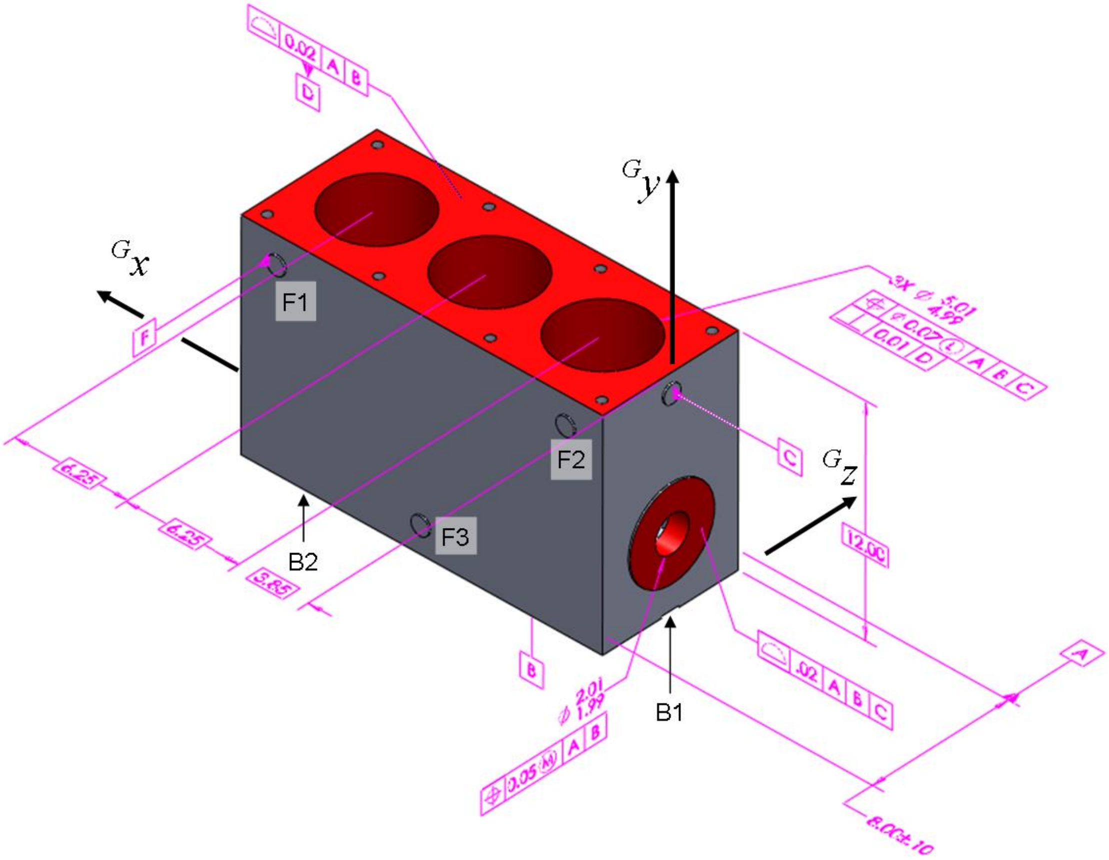


Figure 7.1 The Model and GD&T of the Engine Block Example

7.1 **Setting up the Problem**

The six machined faces (four holes, two circular faces) will each be the source for an S-Map primitive for machinability. The tolerances will contribute another six S-Map primitives. Though there are seven specified, the perpendicularity tolerance is referenced to datum ‘D’, the top-face TBM feature, so per discussion in §3.2 it does not contribute to the final S-Map and can be excluded. Therefore, six T-Maps for the six contributing tolerances and six S-Maps for the six TBM faces are to be created. The tolerance data is shown in Table 7.1, as extracted from the CAD model, shown in Figure 7.1.

Table 7.1

Relevant Tolerances Specified on the In-Line Engine Example

Tolerance Type	Value	Target Feature	Primary Datum	Secondary Datum	Tertiary Datum
Profile	0.02	Top Face	A	B	--
Profile	0.02	Shaft Face	A	B	C
Position	0.07 Ⓛ	3x Cylinder Bores	A	B	C
Position	0.05 Ⓜ	Shaft Hole	A	B	C
Perpendicularity	0.01	3× Cylinder Bores	D	--	--

The machinability S-Map primitives are created based on the amount of material at each feature. The machining stock and sizes used for the example are listed in Table 7.2. The t_{eq} for the cylinders are calculated as the MMC size minus the actual size.

Table 7.2

Machining Stock for TBM Features on the Engine Block Example

TBM Feature	As-Cast Size/Machining Stock
Top Face	+0.015” stock → added to shift
Shaft Face	+0.015” stock → added to shift
3x Cylinder Bores	⌀4.90” → t_{eq} = 0.09”
Shaft Hole	⌀1.90” → t_{eq} = 0.09”

The GCS is defined by datums A (the mid-plane of the width of the part), B (the bottom face) and C (the single location point on the shaft side). The FxCS axes are nominally parallel to those of the GCS, with the fixing points of datum F replacing A as the primary datum. The location points F1, F2, F3 and C are raised by 0.1” from their respective rectangular planar surfaces on which they are positioned, while the B1 and B2 locations are recessed by 0.1”.

The LCS locations are obtained in the GCS, and summarized in Table 7.3. The vectors from the GCS to the LCS are calculated using the recessed locations of datum targets B1 and B2, and the raised location of datum target C. As an example, for the three parallel piston cylinders, each have a depth of 5” so their LCS locations at the center of the cylinders are 2.5” below the top face; with the recessed datum targets B1 and B2, the actual distance from datum B to the top face is 11.9”; therefore, the *y*-coordinates of the cylinder bore LCSs are all 11.9” - 2.5” = 9.4” from the GCS, as shown in Table 7.3.

There are two necessary rotation matrices: one for the LCS of the top face and the three cylinder bores, and one for the LCS of the side shaft face and shaft hole.

$$R_{G\ell,top} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \ R_{G\ell,side} = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

Table 7.3

LCS to GCS Information for the Engine Block Example

TBM Feature	$\vec{\boldsymbol{c}}(\boldsymbol{c_x}, \boldsymbol{c_y}, \boldsymbol{c_z})$ Global to Local Vector	Rotation Matrix
Top Face	(10.1, 11.9, 0)	R_{top}
Cylinder Bore 1	(3.85, 9.4, 0)	R_{top}
Cylinder Bore 2	(10.1, 9.4, 0)	R_{top}
Cylinder Bore 3	(16.35, 9.4, 0)	R_{top}
Shaft Face	(0, 3.9, 0)	R_{side}
Shaft Hole	(0.55, 3.9, 0)	R_{side}
FxCS*	(0, 0, -4.1)	*Identity*

*defined for conversion of the setup point to fixture adjustments

The fixture is defined as in Chapter 5, with location datums F, B, and C. The points where the fixture will make contact, as represented in the FxCS are listed in Table 7.4.

Table 7.4

Definition of the Location Points in the FxCS

F1(*x*,*y*,*z*)	F2(*x*,*y*,*z*)	F3(*x*,*y*,*z*)	B1(*x*,*y*,*z*)	B2(*x*,*y*,*z*)	C(*x*,*y*,*z*)
(18, 10, 0)	(2, 10, 0)	(10, 1.5, 0)	(0.5, 0, 4.1)	(19.5, 0, 4.1)	(0, 11, 4.1)

7.2 The Six T-Maps for the Specified Tolerances

For the example part in Figure 7.1, there is only one critical datum reference frame specified from which all tolerances are referenced. As an assumption for simplification, for all test cases, the unmachined datum features are considered to have been made with minimal deviation from the nominal geometry, while the to-be-machined features will have considerable deviation. This means that the FxCS (datums F, B, and C) and the GCS (datums A, B, and C) are located nominally relative to one another. Because there is no deviation of the datums, there will be no shift applied to the T-Maps for the six tolerances for any of the test cases. Therefore the T-Maps intersection can be made once and used for all test cases.

Tolerance values were chosen assuming that the features would be made to their MMC sizes. For the 0.05” position tolerance on shaft hole, with the MMC material modifier, no bonus tolerance is assigned, and therefore the tolerance value for creating the T-Map is 0.05”. For the three cylindrical bores, the 0.07” position tolerance is specified with the LMC material modifier. Assuming that the holes are made at their MMC condition (4.99”), the bonus tolerance is 5.01” – 4.99” = 0.02”, so the combined tolerance value for creating the T-Map is 0.09”.

The resulting intersection in Figure 7.2 describes the allowable positions of the casting that are permitted by the locations of the datums. Adjustment beyond the boundaries of the solid red shape will yield features that do not meet tolerances with respect to the datums after machining. Figure 7.2 shows the Δ*x*, Δ*y*, Δ*z* cross section (*ϕ*, *ψ*, *θ* = 0) being limited by the S-Map Primitives from the two profile tolerances and the position tolerance of the shaft hole.

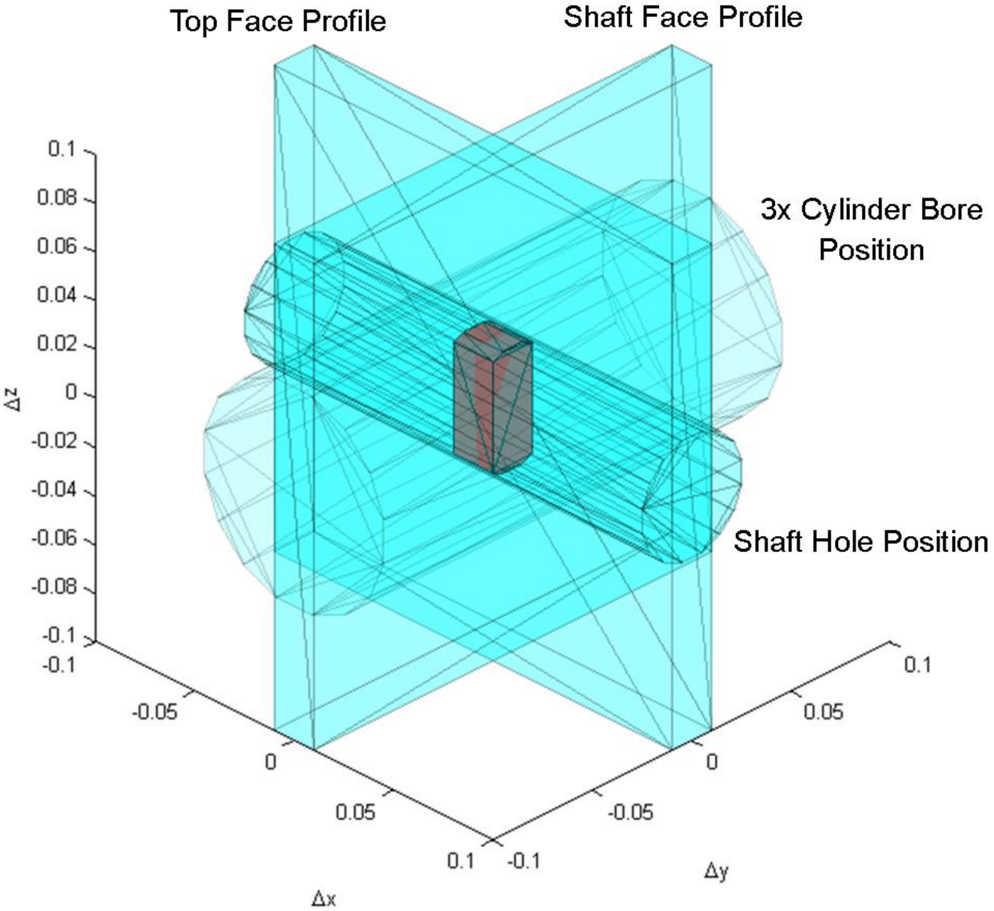


Figure 7.2 A 3-D Cross Section of the Intersection of the Tolerance Constraints for the Engine Block Example

7.3 **Test Cases**

The following test cases will be demonstrated: (A) A part with no translation or rotation deviation of any feature, (b) simple translational deviations of features that predictably cause infeasibility. (c) A general case to test convergence of the algorithms.

7.3.1 **Test Case (A)** Test Case (A) is the part with no translation or rotation deviations of the TBM features from their nominal locations. For this part, there should be little to no suggested adjustment on the order of the tolerances specified. As shown in Figure 7.3, in the translation dimensions, the resulting S-Map is centered on the origin. The constraints for machinability do not limit the final intersection shape. The shape of the S-Map intersection for machinability is two right cylinders intersecting one another. This constraint arises is from the cylinder bores and the shaft hole. Each cylinder has the same size due to the choice of values (t_{eq} is the same).

The values for adjustment from the first case are listed in Table 7.5. The values are an order of magnitude smaller than the smallest tolerance value (0.02”), though not as small as first expected. The selected optimal point is verifiably inside the intersection shape and is therefore a good choice for positioning the part for machining.

Table 7.5 Results from Test Case (A)

Small Displacement Coordinates					
ϕ (rad/deg)	ψ (rad/deg)	θ (rad/deg)	Δx (inches)	Δy (inches)	Δz (inches)
-3.3E-9	-0.0004/0.02°	-0.0001/0.005°	0.0006	0.0023	-0.0025
Adjustments (inches)					
F1	F2	F3	B1	B2	C
0.0045	-0.0017	0.0013	0.0022	0.0045	0.0003

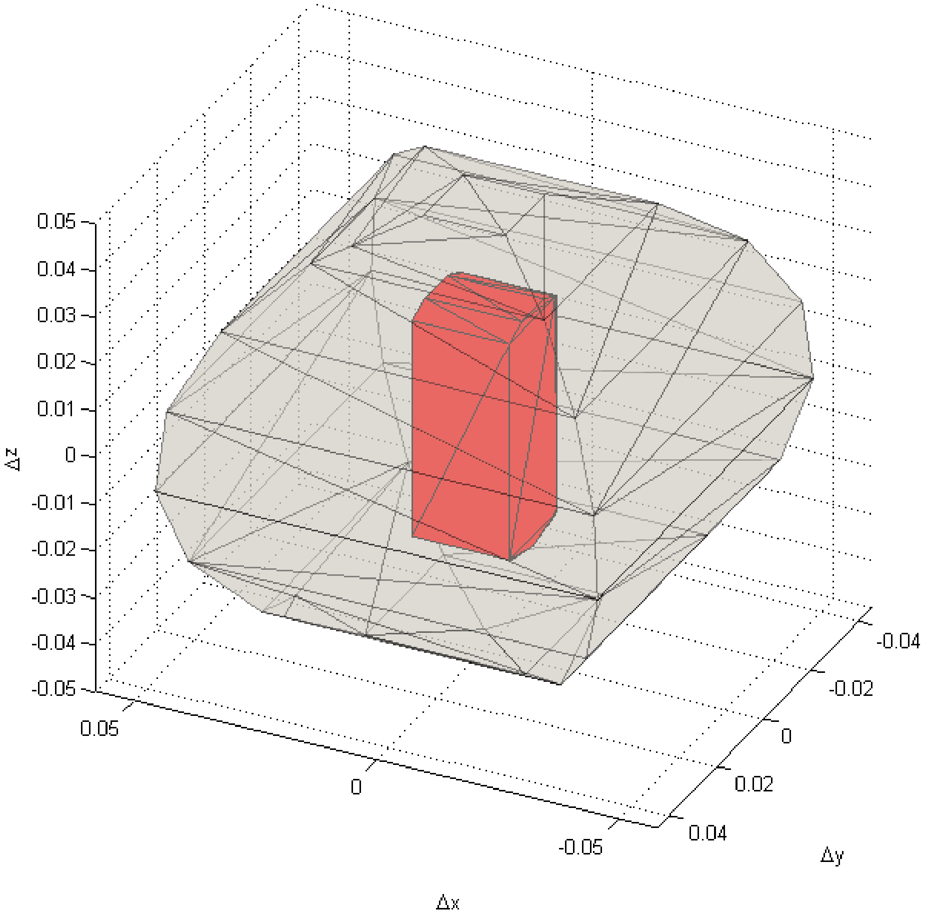


Figure 7.3 The Intersection of the Tolerance Constraints with the Machining Constraints for Test Case (A)

7.3.2 **Test Case (B)** The second test will try to get a close to infeasible result by first guessing the deviation of an object that will create a predictable value. The shaft hole has a 0.05" position tolerance (assigned at MMC, which is the assumed machining size). In the red intersection shape shown in Figure 7.2, the constraint on the curved ends of the long, rectangular prism shape are due to the shaft hole tolerance. The equivalent tolerance of the S-Map primitive for the shaft hole is 0.09" from Table 7.2. The maximum deviation of the minimum material envelope in the Δz direction is obtained by combining these to tolerances and dividing by two to get the allowable radial displacement of (0.05"+0.09")/2 = 0.07 inches. Because small displacements are already limited by the T-Map intersection, the predicted value for Δz should be no greater than 0.05"/2 = 0.025" if no angular deviation occurs. The schematic of the test is shown in Figure 7.4 where the small red arrow shows the direction of adjustment that would bring the center of the 4.90" as-cast red, dashed hole to the intended center of the 4.99" black hole where it could be completely machined and fit within is tolerance zone with respect to the datum reference frame.

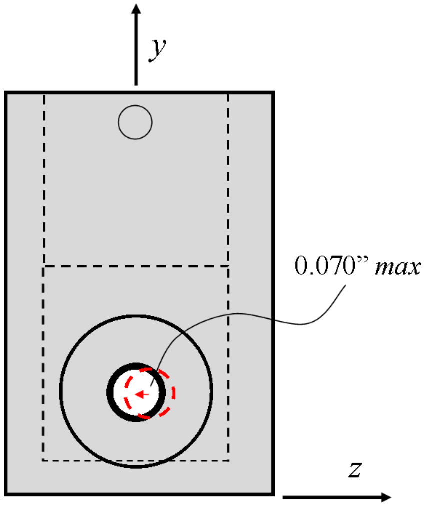


Figure 7.4 The Schematic of Test Case B

The part was *infeasible* with an input shift of -0.070”. The shift was reduced to -0.069”, and rerun. The part was *feasible* at this value. As shown in Table 7.6, the Δz value exceeded the predicted value; however a small positive ϕ rotation has occurred in addition. Figure 7.5 shows that the intersection that occurred is quite small, as predicted.

Table 7.6

Results from Test Case (B)

Small Displacement Coordinates					
$\boldsymbol{\phi}$ (rad/deg)	ψ (rad/deg)	θ (rad/deg)	Δx (inches)	Δy (inches)	Δz (inches)
0.00067/0.038°	9.6E-6	0.00017/0.01°	0.0025	-0.0013	-0.027
Adjustments (inches)					
F1	F2	F3	B1	B2	C
-0.0206	-0.0205	-0.0262	-0.0039	-0.0074	0.0006

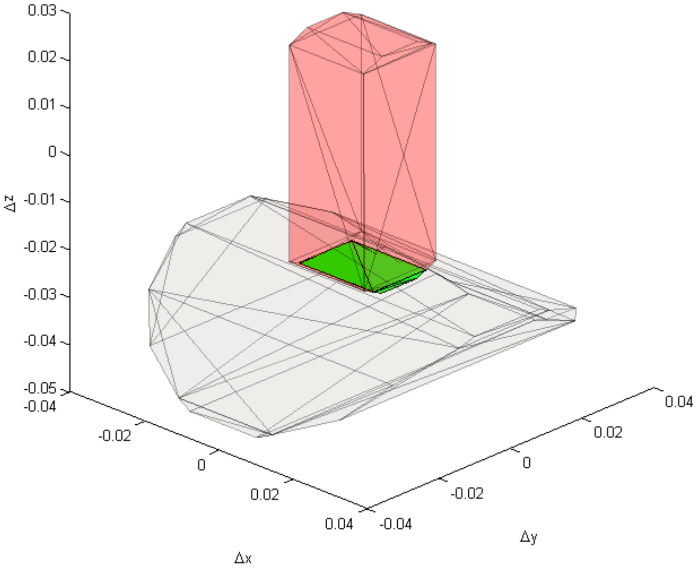


Figure 7.5 The Intersection from Test Case (B)

7.3.3 **Test Case (C)** The final test case attempts to show convergence of the optimization algorithm, described in §5.3.1. For the test case C, the S-Map primitives were each given an initial arbitrary 6-D shift not in excess of 0.02”—the shift values are reported in Table E.1 in Appendix E. The setup point was then calculated. The entire S-Map was then shifted in the opposite direction to locate the calculated setup point on the origin of the coordinate system to simulate an adjustment of the part to the desired location for machining. This process of calculating the setup point using the optimization algorithm and shifting was then repeated for a total of five iterations. The goal was to see if the S-Map point selection algorithm returned the same point each time within an acceptable amount of computational error. For graphing the results, the angular variables were multiplied by a characteristic length of 10 inches, or half the longest dimension on the part; however, a characteristic length of 1.0 inch was used during computation and intersection. Figure 7.6 shows that small displacement values that were initially larger on the first setup point calculation (ϕ, ψ, and Δz) seemed to converge to zero and require no further adjustment on subsequent calculations. The three other variables seemed to oscillate about the zero mark. This may indicate that the part represented by this S-Map is loosely constrained in the θ'-, Δx-, and Δy-directions. The successive setup point coordinates are shown in Table E.2 in Appendix E. Tvo views of the 3-D ϕ, ψ, and Δz cross section of the first S-Map intersection with coordinate values θ' = -0.000012E-5”, Δx = 0.00061”, Δy = 0.0000243” are shown in two views in Figure 7.7.

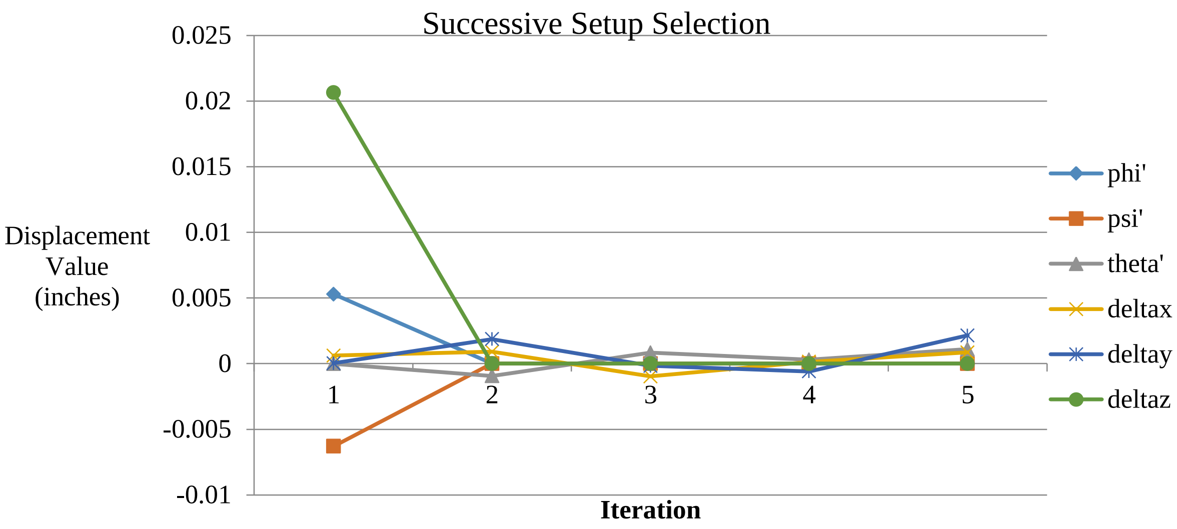


Figure 7.6 Shifting the S-Map Successively to Analyze Convergence of the Optimization Algorithm

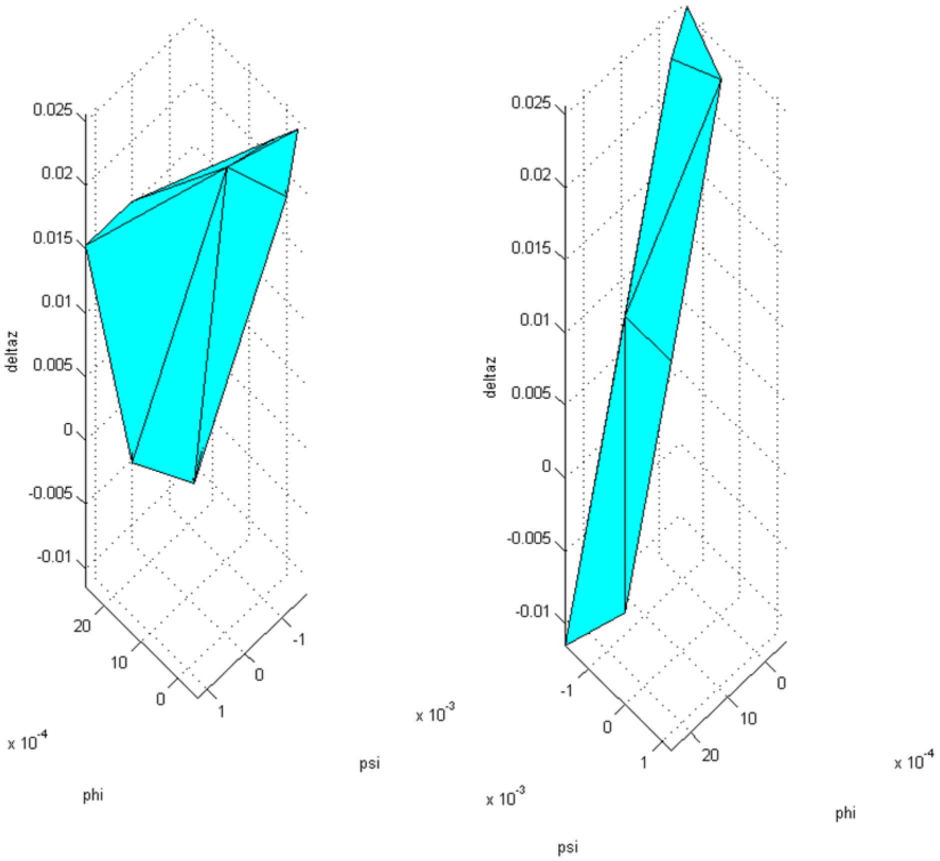


Figure 7.7 Cross Sections of the S-Map for Test Case (C)

CHAPTER 8 CONCLUSIONS AND FUTURE WORK

This thesis discusses a computational tool for use by manufacturers to aid in the process of setting up large and complex parts for machining. The hope is that such a tool will allow manufacturers of smaller production volume parts to save money by reducing setup times and the amount of parts scrapped. The contributions of this research and future works are discussed in the following sections.

8.1 Original Contributions

The theories for making S-Maps in this thesis formalize the definitions of the constraints for positioning each as-fabricated part with respect to its intended location. Ultimately, S-Maps solve a non-trivial extension of the problem of assemblability. This was done through an amalgamation of vector-space models for tolerances (T-Maps and Deviation Spaces), computer-aided geometry tools, optimization techniques, and the kinematics of parallel-actuated robotics. By outlining the process, and creating the algorithmic and computational foundation for solving this type of problem, I have demonstrated the generalized method for approaching this problem.

The basis of the solution is to find a space of allowable adjustments, and choose one that splits the difference between the constraints. This is not so far from what a skilled operator does in practice; however, instead of only being able to balance the constraints of a few features at a time, a computer can consider them all at once. The process for S-Map generation seeks an intuitive approach by considering the allowance of each feature, one at a time, and then combining them to represent the allowance of the entire part.

Initial tests of the computational tool seem to satisfy intuition about how it should behave in simple cases. It is foreseeable that the tool will be able to exceed the abilities of a skilled person in both time and accuracy if further developed.

8.2 **Future Work**

While the framework for representing an S-Map of any type has been developed, there are many implementation steps that must be done to make a seamless tool. Simple future tasks involve creating the algorithms for specifying the S-Maps of other types of features and tolerances that may arise. This will involve finding the equivalence of the plentiful T-Map types in small displacement coordinates, and then implementing the methods as functions for generating the set of half-spaces. A slightly more challenging variation to this task is to implement the constraints of tolerances with respect to incomplete sets of datums, where more degrees of freedom may exist, but this too falls under the simple task of determining which half-spaces to create.

More challenging issues such as data-transfer must be solved to create an automated tool. While there are data formats that are intended to communicate standard part and tolerance information, there are very few standard formats. Incorporating more explicit models such as ASU’s Constraint Tolerance and Feature (CTF) Graph would be useful for the task of automating the S-Map generation process.

And finally, an open algorithmic problem is how to model the S-Maps for a part that is to be machined in multiple stages with multiple setups. It may be that the solution is not far from what has already been done, but it will require additional thinkering.

REFERENCES

[1] Davidson, J. K., and Hunt, K. H., 2004, Robots and Screw Theory, Oxford University Press, Oxford.

[2] Ameta, G., Serge, S., and Giordano, M., 2011, “Comparison of Spatial Math Models for Tolerance Analysis: Tolerance-Maps, Deviation Domain, and TTRS,” J. Comput. Inf. Sci. Eng., **11**(2), p. 021004.

[3] Mansuy, M., Giordano, M., and Davidson, J. K., 2013, “Comparison of Two Similar Mathematical Models for Tolerance Analysis: T-Map and Deviation Domain,” J. Mech. Des., **135**(10), p. 101008.

[4] American Society of Mechanical Engineers, 2009, ASME Y14.5-2009 - Dimensioning and Tolerancing.

[5] Davidson, J. K., Mujezinović, A., and Shah, J. J., 2002, “A New Mathematical Model for Geometric Tolerances as Applied to Round Faces,” J. Mech. Des., **124**(4), p. 609.

[6] Mujezinović, A., Davidson, J. K., and Shah, J. J., 2004, “A New Mathematical Model for Geometric Tolerances as Applied to Polygonal Faces,” J. Mech. Des., **126**(3), p. 504.

[7] Ameta, G., Davidson, J. K., and Shah, J. J., 2007, “Using Tolerance-Maps to Generate Frequency Distributions of Clearance and Allocate Tolerances for Pin-Hole Assemblies,” J. Comput. Inf. Sci. Eng., **7**(4), pp. 347–359.

[8] Giordano, M., and Duret, D., 1993, “Clearance Space and Deviation Space,” Proceedings of 3rd CIRP Seminars on Computer Aided Tolerancing, ENS Cachan, France, pp. 179–196.

[9] Houten, F., Kals, H., and Giordano, M., 1999, “Mathematical Representation of Tolerance Zones.,” Glob. consistency Toler. Proc. 6th CIRP Int. Semin. Comput. Toler. Univ. Twente, Enschede, Netherlands, 22-24 March, 1999, pp. 177–186.

[10] Teissandier, D., Couétard, Y., and Gérard, A., 1999, “Computer aided tolerancing model: Proportioned assembly clearance volume,” CAD Comput. Aided Des., **31**(13), pp. 805–817.

[11] Homri, L., Teissandier, D., and Ballu, A., 2015, “Tolerance analysis by polytopes: Taking into account degrees of freedom with cap half-spaces,” CAD Comput. Aided Des., **62**, pp. 112–130.

[12] Requicha, A. G., 1980, “Representations for Rigid Solids: Theory, Methods, and

Systems," ACM Comput. Surv., **12**(4), pp. 437–464.

[13] Preparata, F. P., and Shamos, M. I., 1985, Computational Geometry: An Introduction, Springer-Varleg.

[14] Bhide, S., Ameta, G., and Davidson, J., 2007, "Tolerance-maps applied to the straightness and orientation of an axis," Model. Comput. Aided, (6963824), pp. 45–54.

[15] Barber, C. B., Dobkin, D. P., and Huhdanpaa, H., 1996, "The quickhull algorithm for convex hulls," ACM Trans. Math. Softw., **22**(4), pp. 469–483.

[16] Lougee-Heimer, R., 2003, "The Common Optimization INterface for Operations Research: Promoting open-source software in the operations research community," IBM J. Res. Dev., **47**(1), pp. 57–66.

[17] Campbell, J., 2004, Castings Practice: The Ten Rules of Castings, Elsevier Butterworth-Heinemann, Oxford.

[18] Mehrotra, S., 1992, "On the Implementation of a Primal-Dual Interior Point Method," SIAM J. Optim., **2**(4), pp. 575–601.

APPENDIX A

NUANCES IN CONVERTING T-MAPS TO SMALL DISPLACEMENT COORDINATES

The transformation of T-Map coordinates to small displacement coordinates follows the process: perform a small displacement, and note the effects on the T-Map coordinates. As explained in chapter 2, in Figure A.1, by rotating the plane about the *y*-axis in the positive-counterclockwise convention, the value of *p* increases, and conversely, rotating about the *x*-axis results in a decrease in *q*. Also, Δz is the opposite of *s*. There are three kinematic degrees of freedom that will not change the equation of the plane geometry, and will therefore not change the values of *p*, *q*, *r,* and *s.* In the canonical coordinate system, these correspond to the Δ*x*, Δ*y*, and *θ* small displacements. These directions are also unconstrained by the tolerance-zone in Figure A.1.

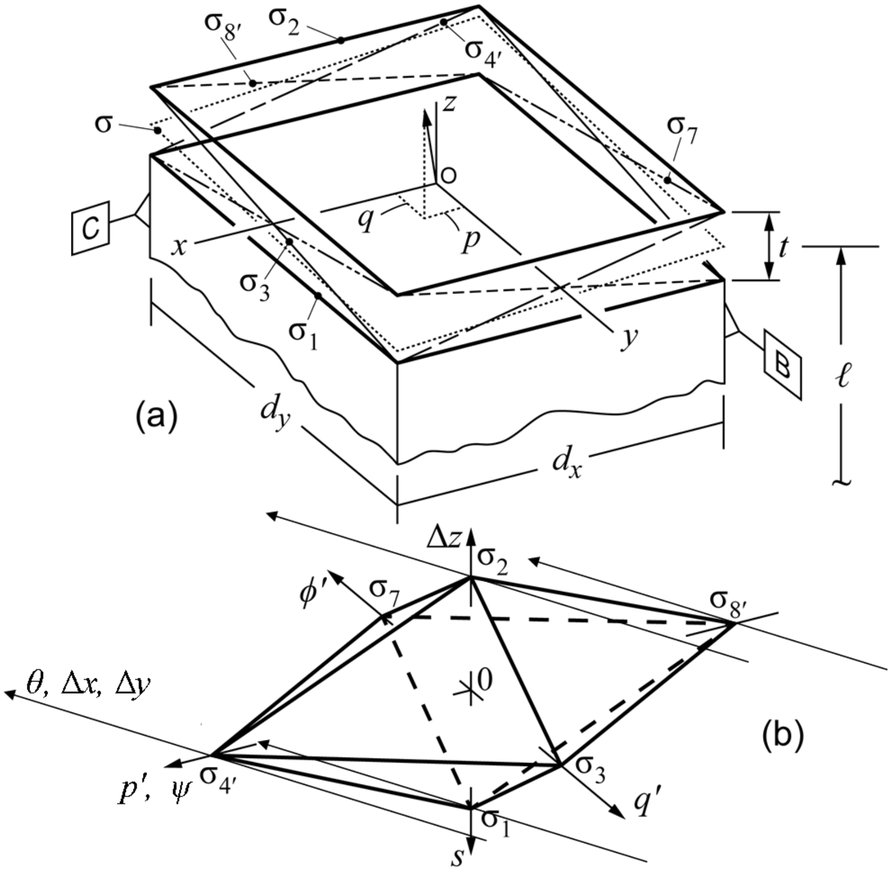


Figure A.1 Projections of the Unit Normal of a Rotated Plane

Therefore, there is a one way mapping of small-displacements to T-Map coordinates. The matrix multiplication in Equation A.1 assumes that the plane is oriented with the z-axis as its normal, and the nominal plane is at the center of the tolerance zone:

$$\begin{bmatrix} p \\ q \\ s \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} \phi \\ \psi \\ \theta \\ \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} \tag{A.1}$$

The small displacement coordinate to T-Map (feature) coordinate transformation is therefore a surjective function, but not an injective one—there is a many-to-one relationship. The reverse is true of the other operation. The following transformation in Equation A.2 is *injective* but not *surjective* because a one-to-many operation is needed in order to return the small displacements back to their original plentiful state.

$$\begin{bmatrix} \phi \\ \psi \\ \theta \\ \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} p \\ q \\ s \end{bmatrix} \tag{A.2}$$

In order to compensate for this, the unconstrained degrees of freedom that are represented in small displacements must be accounted for. When the T-Map coordinates are transformed into small displacement coordinates, they must be "extruded" in the unconstrained dimensions to values that are larger than the tolerances. The extrusion reverses the collapsing of a 6-D space to a 3-D one that occurs in Equation A.1.

This is true for the axis as well. The unconstrained displacements for an axis representing a cylinder in its "canonical" coordinate system are Δz and *θ*, as can be observed in Figure A.2(b). An axis for a cylinder is unlimited in z-translations and *z*-

rotations within its cylindrical tolerance zone. T-Maps use the Plücker line coordinates [1] to represent displaced axes. As shown in Figure A.2(c), *L* and *M* represent the direction of the axis. They behave like *p* and *q* from the T-Map for the plane. *P* and *Q* represent the components of the moment created about the origin when a unit force acts along the line. For the intents of this paper, *P* represents the displacement of the line along the *y*-direction, and *Q* is the opposite of the displacement in the *x*-direction, as shown in Figure A.2(b).

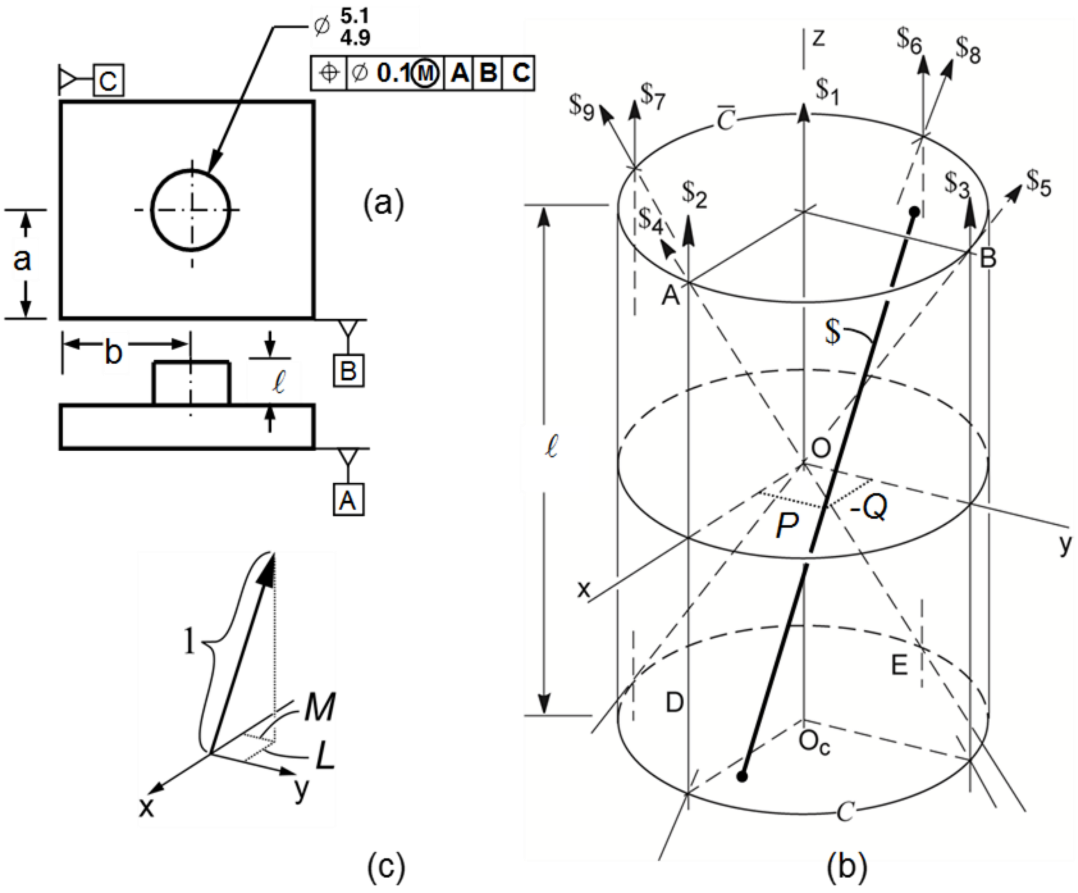


Figure A.2 The Tolerance Zone for an Axis

The matrix representation of the transformation is similar to that of the plane case. For the T-Maps of lines, the θ and Δz small displacements are collapsed by the transformation in Equation A.3, reducing the dimension of the space from six to four.

$$\begin{bmatrix} L \\ M \\ P \\ Q \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \phi \\ \psi \\ \theta \\ \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} \tag{A.3}$$

Because of this, the inverse transformation (Equation A.4) is not complete.

$$\begin{bmatrix} \phi \\ \psi \\ \theta \\ \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} L \\ M \\ P \\ Q \end{bmatrix} \tag{A.4}$$

In order to restore the small displacements model, one can perform the "extrusion" operation in the θ and Δz directions to reverse the collapsing that occurs in Equation A.3.

APPENDIX B

TRANSFORMATIONS

B.1 Transforming Geometric Entities

Given the 3-by-3 rotation matrix R_{ij} and translation vector $\vec{c}(c_x, c_y, c_z)$ that represents the location and orientation of the *j*-frame in the *i*-frame, using the passive description from [1], the transformations of various entities are:

Free Vector $\mathbf{v}(v_x, v_y, v_z)$:

$$ {}^{i}\mathbf{v} = \left[R_{ij}\right] \; {}^{j}\!\begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \tag{B.1.1} $$

Homogeneous Point $\mathbf{p}(x, y, z, 1)$:

$$ {}^{i}\mathbf{p} = \begin{bmatrix} & & & c_x \\ & [R] & & c_y \\ & & & c_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \; {}^{j}\!\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \Rightarrow \; {}^{i}\mathbf{p} = \left[T_{ij}\right] \; {}^{j}\mathbf{p} \tag{B.1.2} $$

Lines, Small Displacements, Screws $\$(L, M, N; P^*, Q^*, R^*)$ (Eq. 4.52 in [1])

$$ {}^{i}\!\begin{bmatrix} L \\ M \\ N \\ P^* \\ Q^* \\ R^* \end{bmatrix} = {}^{i}\!\begin{bmatrix} \left[R_{ij}\right] & [0_3] \\ [X]\left[R_{ij}\right] & \left[R_{ij}\right] \end{bmatrix} \; {}^{j}\!\begin{bmatrix} L \\ M \\ N \\ P^* \\ Q^* \\ R^* \end{bmatrix} \; where \; [X] = \begin{bmatrix} 0 & -c_z & c_y \\ c_z & 0 & -c_x \\ -c_y & c_x & 0 \end{bmatrix} \tag{B.1.3} $$

$$ {}^{i}\$ = \$\$_{ij} \; {}^{j}\$ $$

Its inverse is a vector in the *i*-frame that locates origin O$_j$ relative to O$_i$:

$$ {}^{j}\!\begin{bmatrix} L \\ M \\ N \\ P^* \\ Q^* \\ R^* \end{bmatrix} = \begin{bmatrix} [R]^{\mathrm{T}} & [0_3] \\ \left[[X][R]\right]^{\mathrm{T}} & [R]^{\mathrm{T}} \end{bmatrix} \; {}^{i}\!\begin{bmatrix} L \\ M \\ N \\ P^* \\ Q^* \\ R^* \end{bmatrix} \Longrightarrow \; {}^{j}\$ = \left[\$\$_{ij}\right]^{-1} \; {}^{i}\$ \tag{B.1.4} $$

B.2 Deriving the Transformation for Half-Spaces

While the small displacement transformation in Appendix B.1 is useful for transforming points in the S-Map, It is necessary to transform the *half-spaces*, or the coefficients of the underlying hyperplane. The transformation follows from the transformation of planes and points in 3D. The equation of a 3-D plane is the dot product of the coefficients with a homogenous point. If T transforms a point, then T^{-1} transforms a plane when pre-multiplied by the coefficients of the plane. In 3-D, this is verifiable from Eq. 4.49 from [1].

$$[p \quad q \quad r \quad s][T]^{-1}[T][x \quad y \quad z \quad 1]^{\mathrm{T}} = d \qquad \text{(B.2.1)}$$

Given the normalized homogeneous coordinates of a hyperplane and a homogenous point in the 6-D small displacement space, the equation for the hyperplane is:

$$[a_\phi \quad a_\psi \quad a_\theta \quad a_{\Delta x} \quad a_{\Delta y} \quad a_{\Delta z} \quad b][\phi \quad \psi \quad \theta \quad \Delta x \quad \Delta y \quad \Delta z \quad 1]^T = 0 \qquad \text{(B.2.2)}$$

where $b/\sqrt{\sum a_i^2}$ is the signed distance along the hyperplane normal vector to the origin and corresponds to s in 3-D (Equation B.2.1).

Considering a small displacement as a point in 6-D space and making the appropriate changes to the homogeneous representation, as discussed in Chapter 3.5 of [1], gives:

$$[\$\$] = \left[\begin{array}{ccc} [R] & [0_3] & \begin{matrix}0\\0\\0\end{matrix} \\ [X][R] & [R] & \begin{matrix}0\\0\\0\end{matrix} \\ 0 \quad 0 \quad 0 & 0 \quad 0 \quad 0 & 1 \end{array}\right] \tag{B.2.3}$$

Then, we apply the transformation to points, and the inverse for the hyperplane:

$$[a_\phi \;\; a_\psi \;\; a_\theta \;\; a_{\Delta x} \;\; a_{\Delta y} \;\; a_{\Delta z} \;\; b]\left[\$\$_{ij}\right]^{-1}\left[\$\$_{ij}\right][\phi \;\; \psi \;\; \theta \;\; \Delta x \;\; \Delta y \;\; \Delta z \;\; 1]^T = 0 \tag{B.2.4}$$

Giving the transformation for the hyperplane of a half-space as:

$$\begin{aligned} &{}^{i}[a_\phi \;\; a_\psi \;\; a_\theta \;\; a_{\Delta x} \;\; a_{\Delta y} \;\; a_{\Delta z} \;\; b] = \\ &{}^{j}[a_\phi \;\; a_\psi \;\; a_\theta \;\; a_{\Delta x} \;\; a_{\Delta y} \;\; a_{\Delta z} \;\; b] \left[\begin{array}{ccc} [R]^{\mathrm{T}} & [0_3] & \begin{matrix}0\\0\\0\end{matrix} \\ \left[[X][R]\right]^{\mathrm{T}} & [R]^{\mathrm{T}} & \begin{matrix}0\\0\\0\end{matrix} \\ 0 \quad 0 \quad 0 & 0 \quad 0 \quad 0 & 1 \end{array}\right]. \end{aligned} \tag{B.2.5}$$

B.3 Specialization of the Screw Transformation for Plane Primitives

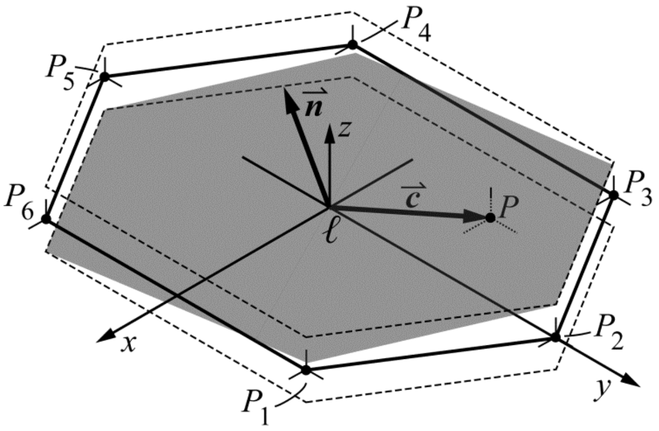


Figure B.1 The Tolerance Zone for a General Plane (repeated from Chapter 4.2)

For T-Map primitives of the plane, such as the hexagonal plane in Figure B.1, the coordinate axes of the LCS and VCS are parallel, and the VCS is displaced by vector $\vec{c}$ with non-zero *x*- and *y*-components. The transformation for small displacement points in the T-Map primitive is specialized to:

$$^{\ell}\begin{bmatrix} \phi \\ \psi \\ \theta \\ \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & c_y & 1 & 0 & 0 \\ 0 & 0 & -c_x & 0 & 1 & 0 \\ -c_y & c_x & 0 & 0 & 0 & 1 \end{bmatrix} {}^{P}\begin{bmatrix} \phi \\ \psi \\ \theta \\ \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} \qquad \text{(B.3.1)}$$

And, for half-spaces, using equation B.2.3, the transformation is:

$$
\begin{aligned}
&{}^{\ell}[a_{\phi}\ \ a_{\psi}\ \ a_{\theta}\ \ a_{\Delta x}\ \ a_{\Delta y}\ \ a_{\Delta z}\ \ b] \\
&= {}^{P}[a_{\phi}\ \ a_{\psi}\ \ a_{\theta}\ \ a_{\Delta x}\ \ a_{\Delta y}\ \ a_{\Delta z}\ \ b]\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -c_y & 1 & 0 & 0 & 0 \\ 0 & 0 & c_x & 0 & 1 & 0 & 0 \\ c_y & -c_x & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
\tag{B.3.2}
$$

All half-spaces in the T-Map primitives for points in the plane have their sole vector component in the Δz-direction:

$$
{}^{P}[a_{\phi}\ \ a_{\psi}\ \ a_{\theta}\ \ a_{\Delta x}\ \ a_{\Delta y}\ \ a_{\Delta z}\ \ b] = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \pm 1 & \frac{t}{2} \end{bmatrix}
\tag{B.3.3}
$$

Multiplying through by the zeros in the hyperplane row vector, the operation reduces to:

$$
{}^{\ell}[a_{\phi}\ \ a_{\psi}\ \ a_{\theta}\ \ a_{\Delta x}\ \ a_{\Delta y}\ \ a_{\Delta z}\ \ b] = \begin{bmatrix} \pm 1 & \frac{t}{2} \end{bmatrix}\begin{bmatrix} c_y & -c_x & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}
\tag{B.3.4}
$$

And finally, the half-spaces for the primitive T-Map for the in the local feature frame, ℓ, are:

$$
{}^{\ell}[a_{\phi}\ \ a_{\psi}\ \ a_{\theta}\ \ a_{\Delta x}\ \ a_{\Delta y}\ \ a_{\Delta z}\ \ b] = \begin{bmatrix} \pm c_y & \mp c_x & 0 & 0 & 0 & \pm 1 & \frac{t}{2} \end{bmatrix}
\tag{B.3.4}
$$

B.4 Specialization of the Transformation For Cylinder Primitives

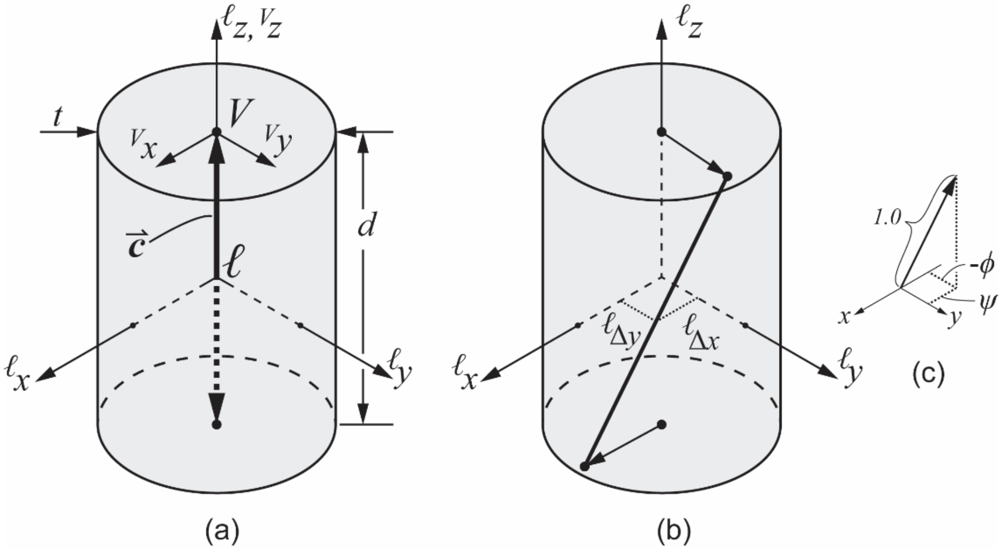


Figure B.3 The Tolerance Zone for the Axis of a Cylinder (repeated from Figure 4.8)

For T-Map primitives of the cylinder, the coordinate axes of the LCS and VCS are parallel, and the VCS is displaced by vector $\vec{c}$ with only a non-zero z-component. The transformation for small displacement points in the T-Map for each of the limiting circles of the tolerance zone is thus specialized to:

$$ {}^{\ell}\begin{bmatrix} \phi \\ \psi \\ \theta \\ \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -c_z & 0 & 1 & 0 & 0 \\ c_z & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} {}^{P}\begin{bmatrix} \phi \\ \psi \\ \theta \\ \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} \qquad \text{(B.4.1)} $$

For the linear half-spaces that encircle the T-Map primitive boundaries, using equation B.2.3, the transformation is:

$$
\begin{aligned}
&{}^{\ell}[a_{\phi}\ \ a_{\psi}\ \ a_{\theta}\ \ a_{\Delta x}\ \ a_{\Delta y}\ \ a_{\Delta z}\ \ b] \\
&= {}^{V}[a_{\phi}\ \ a_{\psi}\ \ a_{\theta}\ \ a_{\Delta x}\ \ a_{\Delta y}\ \ a_{\Delta z}\ \ b]\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & c_z & 0 & 1 & 0 & 0 & 0 \\ -c_z & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
\qquad \text{(B.4.2)}
$$

The linear half-spaces that encircle the boundaries of the T-Map primitives for the ends, each with a unique angle α that positions the tangent hyperplane around the boundary of the primitive (Figure 4.9(c)), will have the form:

$$
{}^{V}[a_{\phi}\ \ a_{\psi}\ \ a_{\theta}\ \ a_{\Delta x}\ \ a_{\Delta y}\ \ a_{\Delta z}\ \ b] = \left[0 \quad 0 \quad 0 \quad \cos(\alpha) \quad \sin(\alpha) \quad 0 \quad \frac{t}{2}\right] \qquad \text{(B.4.3)}
$$

Multiplying through by the zeros in the hyperplane row vector, the operation reduces to:

$$
\begin{aligned}
&{}^{\ell}[a_{\phi}\ \ a_{\psi}\ \ a_{\theta}\ \ a_{\Delta x}\ \ a_{\Delta y}\ \ a_{\Delta z}\ \ b] \\
&= {}^{V}\left[\cos(\alpha) \quad \sin(\alpha) \quad \frac{t}{2}\right]\begin{bmatrix} 0 & c_z & 0 & 1 & 0 & 0 & 0 \\ -c_z & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
\qquad \text{(B.4.5)}
$$

And finally, the half-spaces for the primitive T-Map in the LCS, ℓxyz, are:

$$
\begin{aligned}
&{}^{\ell}[a_{\phi}\ \ a_{\psi}\ \ a_{\theta}\ \ a_{\Delta x}\ \ a_{\Delta y}\ \ a_{\Delta z}\ \ b] \\
&= [-\mathrm{c_z}\sin(\alpha) \quad c_z\cos(\alpha) \quad 0 \quad \cos(\alpha) \quad \sin(\alpha) \quad 0 \quad b]
\end{aligned}
\qquad \text{(B.4.6)}
$$

B.5 Shifting Half-Spaces

Shifting the S-Map is performed by affine transformation of points:

$$ {}^{i}\begin{bmatrix} \phi \\ \psi \\ \theta \\ \Delta x \\ \Delta y \\ \Delta z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & \delta\phi \\ 0 & 1 & 0 & 0 & 0 & 0 & \delta\psi \\ 0 & 0 & 1 & 0 & 0 & 0 & \delta\theta \\ 0 & 0 & 0 & 1 & 0 & 0 & \delta\Delta x \\ 0 & 0 & 0 & 0 & 1 & 0 & \delta\Delta y \\ 0 & 0 & 0 & 0 & 0 & 1 & \delta\Delta z \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} {}^{j}\begin{bmatrix} \phi \\ \psi \\ \theta \\ \Delta x \\ \Delta y \\ \Delta z \\ 1 \end{bmatrix} \tag{B.5.1} $$

Or for half-spaces:

$$ {}^{i}[a_\phi \;\; a_\psi \;\; a_\theta \;\; a_{\Delta x} \;\; a_{\Delta y} \;\; a_{\Delta z} \;\; b] $$

$$ = {}^{j}[a_\phi \;\; a_\psi \;\; a_\theta \;\; a_{\Delta x} \;\; a_{\Delta y} \;\; a_{\Delta z} \;\; b] \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & -\delta_\phi \\ 0 & 1 & 0 & 0 & 0 & 0 & -\delta_\psi \\ 0 & 0 & 1 & 0 & 0 & 0 & -\delta_\theta \\ 0 & 0 & 0 & 1 & 0 & 0 & -\delta_{\Delta x} \\ 0 & 0 & 0 & 0 & 1 & 0 & -\delta_{\Delta y} \\ 0 & 0 & 0 & 0 & 0 & 1 & -\delta_{\Delta z} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{B.5.2} $$

Multiplying through, and simplifying:

$$ \begin{cases} {}^{i}b = {}^{j}b - \delta_\phi a_\phi - \delta_\psi a_\psi - \delta_\theta a_\theta - \delta_{\Delta x} a_{\Delta x} - \delta_{\Delta y} a_{\Delta y} - \delta_{\Delta z} a_{\Delta z} \\ {}^{i}a_x = {}^{j}a_x \end{cases} \tag{B.5.3} $$

APPENDIX C

ORDER OF S-MAP OPERATIONS

Using the notation from Chapter 5, the equations generated for representing the process of creating the S-Maps are verified.

<u>For Machinability:</u>

$$ {}^{G}s = {}^{G}s_n + {}^{G}\delta $$

Transforming to LCS:

$$ {}^{G}s = \$\$_{G\ell}({}^{\ell}s_n) + \$\$_{G\ell}\left({}^{\ell}\delta\right) $$

Collecting Terms:

$$ {}^{G}s = \$\$_{G\ell}\left({}^{\ell}s_n + {}^{\ell}\delta\right) $$

Expanding

$$ {}^{G}s = \$\$_{G\ell}\left({}^{\ell}s_n + \left({}^{\ell}f_n - {}^{\ell}f\right)\right) $$

$$ \boxed{{}^{G}s = \$\$_{G\ell}\left({}^{\ell}s_n + \left({}^{\ell}f_n - T_{\ell G}\,{}^{G}f\right)\right)} $$

<u>For Tolerances with Respect to Datums:</u>

$$ {}^{G}s = {}^{G}s_n + {}^{G}\delta $$

Transforming to LCS:

$$ {}^{G}s = \$\$_{G\ell}({}^{\ell}s_n) + \$\$_{Gd}\left({}^{d}\delta\right) $$

Expanding:

$$ {}^{G}s = \$\$_{G\ell}({}^{\ell}s_n) + \$\$_{Gd}\left({}^{d}D_n - {}^{d}D\right) $$

$$ \boxed{{}^{G}s = \$\$_{G\ell}({}^{\ell}s_n) + \$\$_{Gd}\left({}^{d}D_n - T_{dG}\,{}^{G}D\right)} $$

Thus, the final transformed and shifted S-Map primitive points (${}^{G}s$) are made from: the local unshifted S-Map points (${}^{\ell}s_n$), measured part data (${}^{G}f$, ${}^{G}D$), and transformations that are calculable from CAD geometry ($T_{\ell G}$, T_{dG}, $\$\$_{G\ell}$, $\$\$_{Gd}$)

APPENDIX D

RELEVANT DATA STRUCTURES IMPLEMENTED IN C++

Table D.1

LocalGeom Base Class Structure

Base Class: *LocalGeom*	
Variables	
R	Container for the rotation matrix from the GCS to the LCS
Cx, *Cy*, *Cz*	Containers for the relative position vector from the GCS to the LCS
X	The cross product matrix for screw transformations
Methods	
setCvect	Sets the values of Cx, Cy, Cz, and populates the *X* matrix
get/setRmat	Gets/sets the *R* matrix

Table D.2

PlnSeg Derived Class Structure

Derived Class: *PlnSeg* (Plane Segment)	
Inherits: *LocalGeom*	
Variables	
P	List of 2-D points on the boundary of the plane segment
Methods	
addPoint	Adds a 2-D point to the list of points on the boundaries

Table D.3

CylSeg Derived Class Structure

Derived Class: *CylSeg* (Cylinder Segment)	
Inherits: *LocalGeom*	
Variables	
end1	The distance to the first end point of the cylinder from the LCS
end2	The distance to the second end point of the cylinder from the LCS
Methods	
addPoint	Adds a 2-D point to the list of points on the boundaries

Table D.4

TMap Base Class Structure

Base Class: *TMap*	
Variables	
hspacelist	List of *halfspace6* objects that make up the boundaries of the T-Map
TMapQhull	A Qhull object for computing and storing the intersection
optpoint	The central point of the intersection shape
Methods	
addHalfspace	Adds a *halfspace6* to *hspacelist*
addTMap	Adds the *hspacelist* of another *TMap* object to the stored *hspacelist*
calcInteralPoint	Calculates the central point of the *TMap* and stores it in *optpoint*
scaleAngular	Scales the angular dimensions by a value (characteristic length)
createOpt3Dsection	Creates a 3-D cross section of the 6-D T-Map with 3 coordinates set to their optimal value
createZero3Dsection	Creates a 3-D cross section of the 6-D T-Map with 3 coordinates set to zero

Table D.5

LocalPlnTMap Derived Class Structure

Derived Class: *LocalPlnTMap* (Local Plane T-Map)	
Inherits: *TMap* and *PlnSeg*	
Variables	
halfTMap	Logical value. Set to true to represent the open ended S-Map
tol	Tolerance value for the size of the tolerance zone
Methods	
sethalfTMap	Set the *logical* value of *halfTMap*
get/setTol	Gets/Sets the tolerance value
genLocalTMap	Generates the half-spaces for the plane segment with the specified tolerance

Table D.6

LocalCylTMap Derived Class Structure

Derived Class: *LocalCylTMap* (Local Cylinder T-Map)	
Inherits: *TMap* and *CylSeg*	
Variables	
tol	Tolerance value for the size of the tolerance zone
Methods	
get/setTol	Gets/Sets the tolerance value
genLocalTMap	Generates the half-spaces for the cylinder segment with the specified tolerance

Table D.7

GlobalTMap Derived Class Structure

Derived Class: *GlobalTMap* (Global T-Map)	
Inherits: *TMap*	
Methods	
Overloaded Constructors	Inputs either a *LocalPlnTMap* or *LocalCylTMap* and stores the transformed T-Map

Table D.8

SMapPrim Derived Class Structure

Derived Class: *SMapPrim* (Global S-Map Primitive)	
Inherits: *TMap*	
Variables	
LMCshift	Stores the 6-D shift vector for shifting the S-Map primitive
Methods	
setshift	Sets the *LMCshift* vector
shiftSMap	Shifts the halfspaces by the *LMCshift* vector. Sets *LMCshift* to zero vector

Table D.9

SMap Derived Class Structure

Derived Class: *SMap* (Global S-Map)	
Inherits: *TMap*	
Variables	
PrimList	A list of pointers to *SMapPrim* objects for intersection
Methods	
addSMapPrim	Adds the half-spaces of an *SMapPrim* and adds the pointer to the *SMapPrim* to the *PrimList*
intersect	Performs the intersection of all *SMapPrim* objects and trims the redundant half-spaces

APPENDIX E

SUPPLEMENTARY DATA FOR CHAPTER 6 TEST CASE (C)

Table E.1

The Initial Shift Values of the S-Map Primitives Represented in the GCS

TBM Feature	$\delta\phi$	$\delta\psi$	$\delta\theta$	$\delta\Delta x$	$\delta\Delta y$	$\delta\Delta z$
Top Face	0.005	0	0	0	0.01	0
Shaft Face	0	0.001	0	0.1	0	0
Shaft Hole	0	0	0	0	0.02	0
3x Cyl Bore (together)	0.001	0	0	0	0	0.01

Table E.2

Coordinates for Each Successively Calculated Setup Point

Iteration	$\phi \times 10''$	$\psi \times 10''$	$\theta \times 10''$	Δx	Δy	Δz
1	0.0053	-0.0063	-1.2E-05	0.00061	2.43E-5	0.02064
2	2.47E-07	-2.1E-07	-0.00093	0.00091	0.00187	-4.46E-7
3	-3E-07	7.56E-08	0.000837	-0.00097	-0.00017	4.39E-7
4	3.69E-08	6.99E-08	0.000296	0.00014	-0.00059	-7.03E-9
5	2.27E-07	-1.5E-07	0.0011	0.00086	0.00215	-3.87e-7