Feature Cluster Algebra and Its Application for Geometric Tolerancing

by

Yadong Shen

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

Approved November 2012 by the
Graduate Supervisory Committee:

Jami Shah, Chair
Joseph Davidson
Kenneth Huebner

ARIZONA STATE UNIVERSITY

August 2013

ABSTRACT

The goal of this research project is to develop a DOF (degree of freedom) algebra for entity clusters to support tolerance specification, validation, and tolerance automation. This representation is required to capture the relation between geometric entities, metric constraints and tolerance specification. This research project is a part of an on-going project on creating a bi-level model of GD&T (Geometric Dimensioning and Tolerancing).

This thesis presents the systematic derivation of degree of freedoms of entity clusters corresponding to tolerance classes. The clusters can be datum reference frames (DRFs) or targets. A binary vector representation of degree of freedom and operations for combining them are proposed. An algebraic method is developed by using DOF representation. The ASME Y14.5.1 companion to the Geometric Dimensioning and Tolerancing (GD&T) standard gives an exhaustive tabulation of active and invariant degrees of freedom (DOF) for Datum Reference Frames (DRF). This algebra is validated by checking it against all cases in the Y14.5.1 tabulation. This algebra allows the derivation of the general rules for tolerance specification and validation. A computer tool is implemented to support GD&T specification and validation. The computer implementation outputs the geometric and tolerance information in the form of a CTF (Constraint-Tolerance-Feature) file which can be used for tolerance stack analysis.

DEDICATION

My research achievement is dedicated to my beloved parents, Kaihua Jiao and Tianfu Shen for their selflessness and hardship in supporting me. I also dedicate this to my dearest wife, Jie Zhu, for her encouragement and affection.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my committee chair Dr. Jami Shah for his inspiring guidance and continuous support. His diligence and ingeniousness in conducting research have influenced both my academic experience and professional preparation. I am deeply delighted to work in this renowned research lab and to be given such precious platform to participate in this interesting research project.

I would like to thank Dr. Joseph Davidson for his patient instruction in academic courses and valuable advice to my research. I am also very thankful to Dr. Kenneth Huebner for consenting to serve on my committee.

My thanks are sending to my colleagues in the Design Automation Lab, Yifei He, Zihan Zhao, Samir Savaliya, Prashant Mohan, Prabath Vemulapalli who has been working with me in tolerance research group. Thanks also go to Ying Chen, Xiang Ke for their assistance and sharing of knowledge.

I gratefully acknowledge the financial support of the National Science Foundation (grant CMMI-0969821).

Finally I would like to give my special thanks to my family and all my friends for their love and support.

TALBE OF CONENTS

Page

CHAPTER Page

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

ABBREVIATIONS, NOMENLATURE & SYMBOLS

× — Cross Product

CAD — Computer Aided Design

CAT — Computer Aided Tolerance (analysis system)

CS — Coordinate System

CTF — Constraint Tolerance and Feature Graph

TC — Target Cluster

DRF — Datum Reference Frame

DOF — Degrees of Freedom

GD&T — Geometric Dimensioning and Tolerancing

MMC — Maximum Material Condition

LMC — Least Material Condition

RFS — Regardless of Feature of Size

TDOF — Translational Degree of Freedom

RDOF — Rotational Degree of Freedom

TTRS — Technological and Topologically Related Surface

Y14.5 — Tolerance Specification Standard (ASME Y14.5-2009)

CHAPTER 1 INTRODUCATION

1.1 Background and motivation

Dimensioning and geometric tolerancing describe the range of deviation of the nominal size and shape of manufactured parts. Tolerances are important in both design and manufacturing stages. It governs the required procedure and accuracy for manufacturing and quality control. Designers specify and allocate tolerances to meet desired functional requirements; manufacturing engineers use specified tolerances to direct process planning and manufacturing methods; inspectors follow the tolerance specifications to examine if the results are allowable in terms of variations. The importance of GD&T is a key driver in manufacturing industry and is tasked with improving productivity and the quality of their products.

The current tolerance standard ASME Y14.5-2009 provides designers with rules, symbols, and interpretations on different tolerance classes and cases. It is literally a technical language to be communicated between designers and manufacturers to ensure that design requirements are interpreted unequivocally. However, ASME Y14.5-2009 does not have a rigorous mathematical representation of tolerance specifications including target, datum reference feature (DRF) and tolerance classes. This produces some difficulties for designers to specify proper tolerance classes and DRF to correct target entity; in addition, it also brings about ambiguity or misinterpretation between designers, machinists, and inspectors.

Even though many commercial CAD/CAT (Computer Aided Design/Computer Aided Tolerance) systems can be found in the market to specify tolerance information, none of them check for full conformance to ASME Y14.5-2009. Some of the commercial CAD/CAT systems which claim to integrate tolerance elements into their systems actually just display tolerance symbols on the screen. In these systems the end-users still need to find the feasible tolerance elements without any help from the systems. Computer modules are not "intelligent" enough to perform valid reasoning and decision making on tolerance specification and scheme. An example of a system that is adequate but not "intelligent" is SolidWorks' DimXpert module. The module is able to perform tolerance automation and specification; however, it does not provide a verification process to check the validity of DRFs. Furthermore, DimXpert does not incorporate an advising module to provide designers valuable tolerancing suggestions based on tolerance practice rules. Therefore, a robust mathematical model for geometric dimensioning and tolerancing (GD&T) is a prerequisite prior to attempting to build an automated system and develop a computer data structure for supporting GD&T applications (tolerance specification, tolerance validation, tolerance analysis, tolerance optimization). Mathematical model contrives a mathematical or semantics interpretation for a tolerance specification. A computer data structure for GD&T uses computer algorithms and data structure to represent tolerance specification.

A successful computer model should satisfy several desired requirements: (1) Completeness, (2) Compatibility, (3) Computability, and (4) Validity.

Completeness requires a model to be capable of representing all the geometric and tolerancing information, i.e., tolerance classes, DRF and material condition, etc. Compatibility highlights the ability to be consistent with engineering practice, particularly with national and international standards, such as ASME Y14.5-2009. Computability indicates that the model must be understandable to CAD/CAT and geometric reasoning systems without human interactions. The model must integrate the semantics of geometric tolerance into CAD/CAT systems. Validity means that incorrect or illegal GD&T specifications should be detected and resolved. A self-validating model has a data structure that prevents incorrect GD&T specifications. A proper model thus will not only be in accordance with hard rules in the standards, but also helps to implement good practice rules for design, manufacturing, and inspection.

Figure 1.1 shows some examples of invalid GD&T specifications. In Figure 1.1 (a), a simple syntax error occurred when a cylindricity tolerance is applied to a rectangular plane. This error can be avoided if the computer model has feature recognition before tolerance verification. In Figure 1.1 (b), a parallelism tolerance is improperly specified on target plane perpendicular to its datum reference. This error can be fixed if the computer model has metric relations verification which is capable of detecting the implied metric constraint in tolerance specification. In Figure 1.1 (c), the secondary datum is apparently redundant because it controls the same DOFs of the target entity that primary datum does. It is can be found if the computer model has DOFs checking module to perform DOF validation.

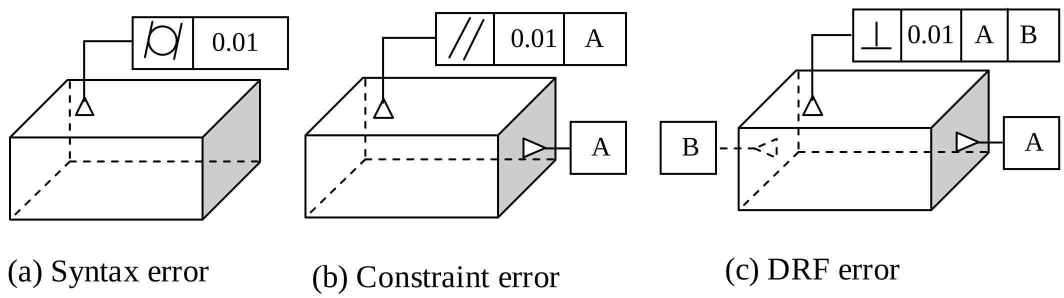


Figure 1.1 Invalid GD&T Specifications

In order to create an effective and powerful computer tool for GD&T representation, a bi-level GD&T model has been proposed and developed [Davidson & Shah 1998, 2001B, 2003; Davidson, Mujezinović & Shah 2000, 2002; Mujezinović, Davidson & Shah 2001, 2004; Wu 2002; Wu, Shah & Davidson 2003A]. As the name suggests, this bi-level GD&T consists of two levels: local and global. The local level is used to model and place tolerance specifications applied on one feature. In contrast, the global level is used to interrelate all tolerance specifications on a part or an assembly. The global level model aims to achieve functions such as tolerance specification, validation and advisor, etc.

The local model is based on the concept of Tolerance Map® (T-Map®), which is a finite set of multivariate regional models to represent all possible geometrical variations controlled by each tolerance class [Davidson, Mujezinović & Shah 2000; Davidson, Shah 2001]. A Tolerance Map® (T-Map®) is a hypothetical volume of points corresponding to all possible locations and variations of a segment of a plane (or an axis) which can arise from tolerance on size, form, position and orientation [Davidson, Mujezinović & Shah 2000].

The global model has been studied and researched for many years. A graph-based model was first developed by Zhang, Yan, and Shah. This model is originated from the Degrees of Freedom (DOFs) for three primitive geometric elements (point, line and plane) and features of size (parallel faces, cylinder, sphere) [Zhang 1992; Yan 1995; Shah, Yan & Zhang 1998]. It is followed by a study on validation aspects of GD&T specifications [Kandikjan & Shah 1998; Kandikjan, Shah & Davidson 2001]. Next, a GD&T advising system is applied to the global model. The system is a comprehensive platform to perform tolerance specification and validation of dimensioning and tolerancing schemes consistent with ASME Y14.5-2009 and good practice rules. The most recent global model studies the metric relations of the local model and describes the geometric variations of all tolerance classes with the exception of free form profiles. This thesis will further explore the application of degrees of freedom and devise DOF algebra to substantiate the tolerancing specification, and to optimize the selection of datum references for a particular tolerance specification. In the end, this research attempts to implement a computer framework for validation.

1.2 Problem statement

The first objective of this work is to investigate the entity clusters composed of primitive features and features of size, and apply DOF algebra to identify or categorize the entity clusters. The second objective of this work is to develop control rules and mathematical tools to assist tolerance validation and analysis. Major tasks in this research include:

(1) Represent entities (Point, Line, and Plane) in terms of variant and invariant DOFs;

(2) Define Boolean operations to calculate the DOFs of combinations or clusters, such as point – point, line – line, plane – plane, plane – line, etc;

(3) Express tolerance classes by DOF; associate target clusters with datum reference frames;

(4) Implement a computational tool to specify, validate, and advise tolerance specification.

CHAPTER 2 LITERATURE REVIEW

In this chapter, a literature review will be presented on a number of topics related to this research. Relevant topics include GD&T mathematical models and GD&T advisor.

2.1 GD&T representation models

Over the past years, researchers have devoted a significant amount of time to exploit various approaches and mathematical definitions to represent appropriately GD&T in computer systems. In the very beginning of GD&T research, plus/minus +/- tolerance in dimension can be easily represented by arithmetic expression. After DRF is related to tolerance, geometric zone is then defined in the representation. However, some representation models, such as attribute and offset models, were simply used for the tolerance representation in computers. They did not have any discussion on tolerance validation. This situation did not change until the TTRS (Technologically and Topologically Related Surface) model and DOF concept were proposed in 1990s.

2.2.1 Mathematical Model

Several groups of researchers have developed various generic approaches to represent geometric tolerances. The five major GD&T representation models are: parametric models, offset zone models, attribute models, kinematic models, and DOF models.

Parametric models: In parametric CAD, tolerances are modeled by allowing +/- variations in dimensional or shape parameters. The parameter values can be represented by a set of simultaneous equations presenting the dimensions and constraints. This method has been successful with 2D profiles, but it is not applicable in 3-D problems mainly because equations are written and solved for vertex positions, which limits the application to a polyhedral shape. [Hillyard and Braid 1978] devised the representation of dimensional tolerance for point position. [Krishna et al, 1997] then improved this method by representing a form tolerance and the derivation of DRF. [Turner et al. 1993] also formulated a parametric model in 2-D for size, form, and orientation.

Offset models: In this method, the maximal and minimal object volumes are obtained by offsetting the object by amounts equal to the tolerances on either side, and then a tolerance zone is represented by Boolean operation of the two volumes. [Requicha et al. 1983] was the first to introduce an offset model. Offset models are not only effective in distinguishing between effects in different tolerances types, but also to study their interaction.

Attribute models: In attribute models, the tolerance specification is stored as the basic characteristic of either geometric entities or metric relations in CAD systems [Shah and Miller, D., 1990]. The disadvantage of the attribute approach is its incapability to perform tolerance validation and analysis since the GD&T semantics is not stored in the model structure.

Kinematic Models: Features in parts are represented by kinematic links and joints. The link connects a tolerance zone and its datum reference features. [Chase et al. 2000] developed a kinematic model by using vector loop to represent the dimensional distance between joints in the assembly. The magnitude of the dimension is the length (L) of the individual vector in the loop. The kinematic joints describe the motion constraints at the points of contact between mating parts.

2.2.2 DOF Model

The DOF model proposed that geometric entities (point, line, and plane) are rigid bodies with degrees of freedom. Geometric tolerancing and DRF are the constraints on DOFs. Tolerance classes and datum features incorporate to determine how each DOF of geometric entity is controlled.

[Shah, Yan & Zhang 1998] developed a Dimension and Tolerance Graph (DTG) model for GD&T representation based on [Zhang 1992]. The nodes of the graph symbolized geometric primitives of the part. The arcs indicated the type of control frame and the DOF controlled. [Kandikjian, Shah, Davidson 1999] validated GD&T specifications by grouping geometric entities in clusters. Entities that are mutually and completely constrained are organized into clusters. The clusters are progressively combined into bigger clusters and the dimensioning is complete when all the nodes of the part in the graph are in the same cluster. The dimension graph was complete if all the DOFs of all the nodes in the DTG of a part were constrained by dimensions. Tolerance information was related to the geometric entities as their attributes. The datum reference frame (DRF) and datum precedence were stored. Tolerance validations were relatively complete, including checking the type of entities, the type of dimensions, the validity of a DRF, the constrained DOF of the tolerance zone, the hierarchy of tolerance classes, and over-tolerancing. However, the check for over-tolerancing was not correct since it was based on an incorrect definition of over-tolerancing. The hierarchy of tolerance classes was not fully explored.

Both models have employed DOF concept into tolerance representation. The computability of DOF brings the possibility for the tolerance validation and DRF verification.

2.3 GD&T Advisor

The Y14.5-2009 standard is complex and it has many tolerancing rules. Many engineers are not familiar with all its intricacies. A few tolerancing computer tools have been developed for GD&T support but they simply focus on expressing tolerance specifications as textual attributes. In addition, these tools never check the tolerancing specifications against good practice rules. In the tolerancing stage, a good advising system is highly desirable to facilitate engineers to check the tolerancing specifications against manufacturability and inspectability criteria.

[Bley et al 1999] justified the need for a system based on rules and heuristic knowledge to guide a designer during the design process. They identified four decision criteria from a manufacturing viewpoint that are important for a designer during tolerance specification: feasibility, cost, machine availability, and production time. This work included the implementation of a tolerancing advisory system in features based design environment. [Wittmann 1999] also proposed a similar concept. In their system, the knowledge base contains information about the technical feasibility of manufacturing a part to specified tolerances and its cost. The prototype database has information regarding available machine tools and their respective precision capabilities. There are no specific rules or practices mentioned in either of the systems described above and the approach is reactive, rather than proactive; there is no advice given to the designer about tolerances.

[Manivannan et al 1989] developed a knowledge-based system for the specification of ISO fits for the manufacture of rotational mating parts with interference, clearance, and transition fit types. The Rule Oriented System for Computer-Aided Tolerancing(ROSCAT) has a rule base consisting of various tolerance information, fit types, rules, and procedures. Rulesets within the rule base invoke the procedural knowledge base, select a set of constraints and perform a search to obtain the best fit for either a shaft basis or hole basis system. This system also does not provide any recommendations to the designers. An expert system to guide users in machine and workpiece fixturing was developed by [Darvishi and Gill 1988]. “IF/THEN” rules were used to guide the interaction between features.

[Ramaswamy, Shah, Davidson, 2001] presents a compilation of good practice GD&T rules collected from various sources and also their implementation in an advisory system. The advisory system purports to detect and avoid invalid tolerancing specifications and also to help designers make the good decisions during the specification stages. The system involves the implementation of an architecture or a seamless integration of solid modeling, dimension and tolerance specification and validation, and tolerance allocation modules.

CHAPTER 3 ASU GD&T GLOBAL MODEL

The motivation for creating and enhancing GD&T model is to represent geometric features, metric relations, and tolerance information in an independent and uniform format. This format should be able to be usable by tolerance analysis software. In order to achieve such a model, the ASU GD&T Global Model [Wu 2002; Wu, Shah & Davidson 2003A] was developed. This research will add DOF computations to the model and implement a GD&T specification and validation software.

3.1 GD&T elements

Once a target entity is placed with a tolerance specification, there are three major elements involved in GD&T: geometric entities, metric relations between entities, and tolerance information (allowed variation and tolerance class). Figure 3.1 shows three elements in GD&T. Geometric entities could be vertices, edges, faces, feature of size (FOS), and pattern of feature of size. FOS includes cylindrical faces, slots, and spheres. A face refers to a plane or a free form surface.

Metric relations are used to control the size and shape of entities and they also indicate the dimensional relations between entities. In this research, metric relations consist of dimension, orientation, location, and shape. Dimension describes the size and geometric information of an entity. Shape metric relations define the intrinsic shape of a geometric entity. Orientation metric relations describe the angular constraint on entities and could be an angle, a perpendicular relation or a parallel relation. The location relation usually involves distance and orientation relations. Coincident and concentric relation reveals that two features of size should have zero distance and zero angles.

A tolerance class is semantic notation for an allowable variation of the nominal size and shape of a targeted entity. Basically the categorization of a tolerance class is consistent with that of the corresponding metric relation, because tolerance specification depends on the geometric entities and metric relations. There must be compatibility between these three GD&T elements. For instance, an orientation tolerance cannot be specified as a dimensional metric relation. The compatibility between the three elements will be discussed later in Chapters 4 and 5.

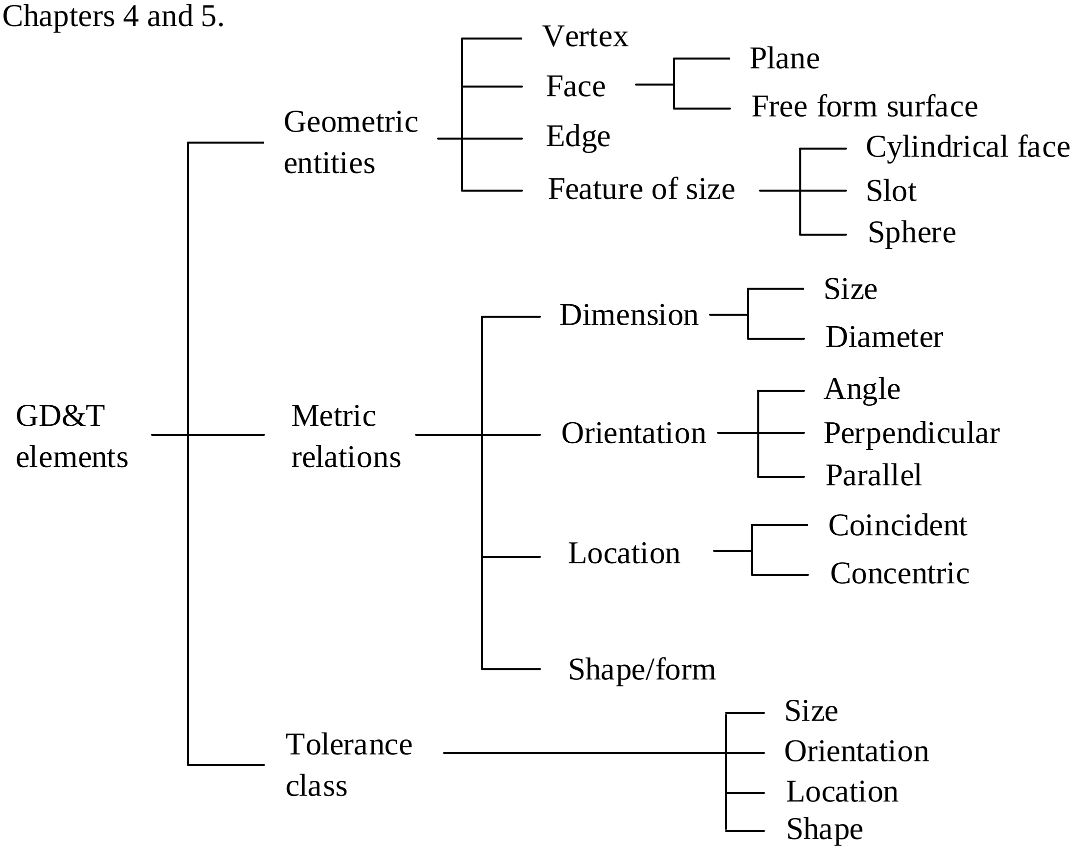


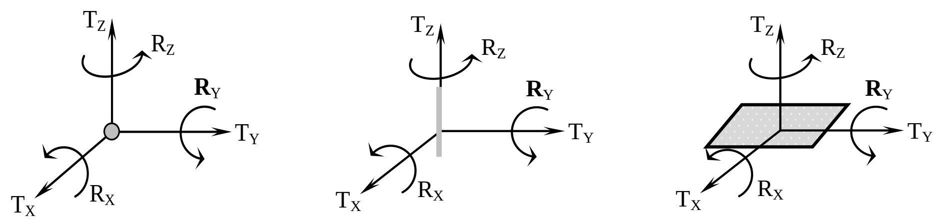
Figure 3.1 GD&T Elements

3.2 Geometric primitives and their DOFs

Many past studies have shown that a vast majority of geometric features on parts can be represented by a point, line, plane or the combination of all three. A point, line, and plane in 3-D space are defined as primitive entities. An unconstrained rigid object in 3-D space can be thought of as having six possible motion components: three translational DOFs (x, y, z) (TDOFs) and three rotational DOFs (α, β, γ) (RDOFs). An active DOF is an independent displacement quantity that needs to be specified or controlled. An invariant DOF is a quantity (distance or angle) that does not change under free transformations. Since rotations of a point about itself will result in no change in position of the point, it is said to have all rotations invariant, leaving three translations as active DOFs. Similarly, an infinite line does not change position if translated along its length or rotated about itself. Therefore, it has 2 TDOFs and 2 RDOFs free. On the other hand, an infinite plane will not change if rotated about its normal or translated along the plane itself, so it has 1 TDOF and 2 RDOFs remaining. For the purpose of this development, I consider a line and plane to be infinite.

To represent the active or invariant DOF of any primitive entity in a uniform way, we propose a six dimensional binary vector. We will write the TDOFs and then RDOFs, separated by a comma thus: (T_x T_y T_z , R_x R_y R_z), where T_i and R_i are the translational and rotational degree of freedom for axis ‘i’. We will use 1 to indicate existence and 0 for non-existence. For instance, (0 1 1, 1 0 0) means that (T_y T_z R_x) exist as active DOFs and the rest are invariant in the

DOF vector X_{DOF} while the inverse is expressed as X_{INV} vector. Figure 3-2 shows the active and invariant DOF of the primitives and binary vector representation.



	Point	Line	Plane
X_{DOF}	$(T_x\ T_y\ T_z\ ,\ ---) \rightarrow (111{,}000)$	$(T_x\ T_y\ -,\ R_x\ R_y\ -) \rightarrow (110{,}110)$	$(-\ -\ T_z\ ,\ R_x\ R_y\ -) \rightarrow (001{,}110)$
X_{INV}	$(---,\ R_x\ R_y\ R_z) \rightarrow (000{,}111)$	$(-\ -\ T_z\ ,\ -\ -\ R_z) \rightarrow (001{,}001)$	$(T_x\ T_y\ -\ ,\ -\ -\ R_z) \rightarrow (110{,}001)$

Figure 3.2 Active & invariant DOF of the primitives and binary

By looking at the active and invariant DOFs any geometric entity may have, all the geometric entities in 3-D space can be boiled down to three primitive entities (i.e. point, infinite line and infinite plane) and three combinations [Kandikjan, Shah & Davidson 2001]. In this thesis these six cases will be referred to as control frames. Each control frame can constrain a certain combination of translational and rotational DOFs. Figure 3-3 shows the six control frames and their combined DOFs. Features of size have size or shape control. If their size or shape control is represented as a special independent DOF [Wu, Shah & Davidson 2003], the features of size in geometry can also be rendered as special cases of primitives or their combinations. For instance, a sphere is a center point with a diameter. The change in diameter would not have any impact on the location and motion of the center point. A cylinder is a center line with a diameter and height parameter and the location of center line is independent from these parameters. A slot/tab is extracted as a center plane with three parameters of thickness, length and depth. It is treated as a plane primitive.

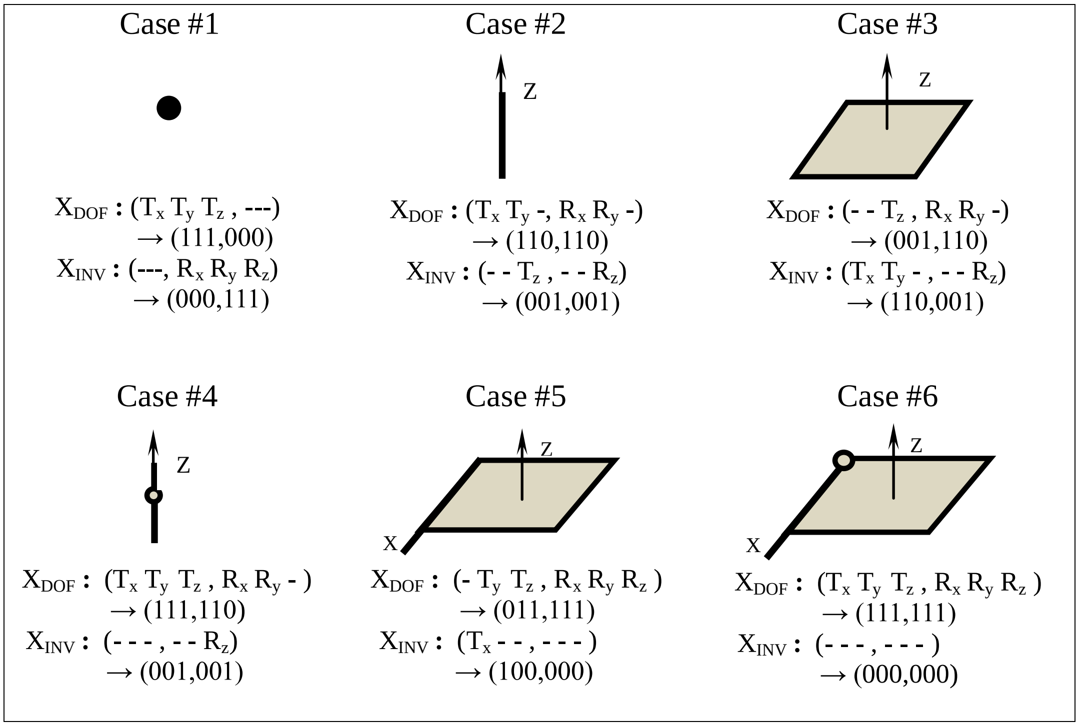


Figure 3.3 Control frames and their combined DOFs

3.3 Metric relations in GD&T

Metric relations are geometric constraints that define the nominal shape, size and location of any geometric feature with respect to other geometric entities on a part. We can think of it as different metric relations constraining the DOFs of geometric entities [Wu, Shah & Davidson 2003]. In this research, we would like to explore the DOF representation applied on metric relations.

Metric relations are classified into four groups based on the types of DOFs controlled: size, location (distance), orientation (angular), and shape. A size constraint controls the size DOF of features of size. A distance constraint controls TDOF of a geometric feature. An orientation metric relation controls RDOFs of geometric features.

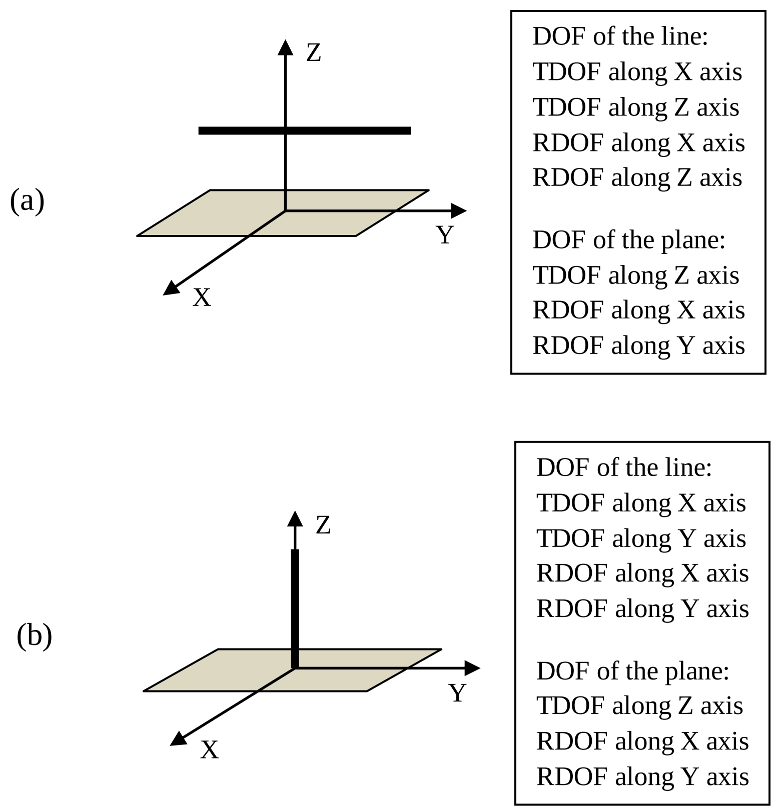


Figure 3.4 Common DOFs between features

A line and a plane are parallel, as shown in Figure 3.4(a). The plane has a TDOF along its normal direction. The translational motion of the line can be identified if and only if the line moves along the normal of plane. Therefore, the metric relation indicates that two geometric features must have DOFs in common.

In Figure 3.4 (a), the line has two TDOFs and two RDOFs and the plane has one TDOF and two RDOFs. The metric relation could be either distance or parallel. However, in Figure 3.4 (b), the metric relation would be nothing but perpendicular. The distance metric relation cannot be applied because the pair of features does not share any common translational DOFs. The maximum constrained DOFs of a pair of features depend on the tolerance specification, which is discussed in Chapter 5.

3.4 Global model representation

The GD&T global model is a data structure used to store GD&T elements for further validation and analysis. Since tolerance specifications should be consistent with feature types and metric relations, the global model should be able to represent the interrelations among three GD&T elements. The ASU GD&T global model is a geometric constraint based model [Shen Z. 2005; Zhang 1992; Shah et al 1998; Kandikjian, Shah 1999]. It encapsulates geometric entities, metric relation between entities, and tolerance specifications as attributes of the corresponding metric relation. The tolerance specifications include tolerance class, tolerance value, DRF, and material modifiers if they are applicable.

[Shen Z. 2005] developed a global model is to represent geometric features, GD&T, and constraints in a neutral format, independent of any particular tolerance analysis model. Driving all tolerance analysis tools with the same independent model provides a uniform basis for comparison of all the approaches. [Wu 2002] has attached all the GD&T data to the geometric entities as attributes.

This attributes CAD model is not suitable for direct use in developing computer-aided tolerance analysis tools. It does not facilitate tolerance verifications. The CAD global model [Shen Z. 2005] contains all the information that is needed for tolerance analysis: nominal geometry (features), constraints, tolerances, degrees of freedom (DOFs) to be controlled, and assembly hierarchy. These different types of information are inter-related to each other. The representation form of global model is a CTF-Graph. [Shen Z. 2005] shows the derivation of DOFs of a primitive feature, and explained how the tolerance specifications control the DOFs of a certain feature. But it does not develop a mathematical representation and application of DOFs of geometric features in tolerance verification and advising.

A CTF graph is referred to as constraint-tolerance-feature-graph. It is a useful tool in GD&T modeling. It is a directed graph structure where geometric features are represented as the nodes. Metric relations and tolerance specifications are attached to the graph as the arcs. All the constraints in the global model form a constraint-graph called C-Graph. The geometric constraint could be a dimension or derived geometric constraints such as perpendicular, parallelism, etc. Figure 3-5 shows the GD&T specifications for a part with two through slots. Figure 3-6 shows the C-Graph for this part. The tolerance specification allows the range of deviation in form, orientation, and location from its nominal size and shape with respect to other features. The tolerance information forms the tolerance graph called T-Graph. Figure 3-7 shows the T-Graph for the part. The T-Graph is a directed graph because the datum reference frame is involved in the tolerance specification. Tolerance specifications without a DRF point to the geometric feature itself. Since a tolerance specification cannot exist without its corresponding geometric constraint and the nodes in both of graphs are completely in common, C-Graph and T-Graph can be combined together to form a directed constraint-tolerance-feature-graph. Therefore, from the structure of CTF, I can conclude that two steps are needed to construct a global GD&T model for a part: (1) identify the metric relations of the dimension scheme, (2) attach tolerance specifications to the corresponding metric relations.

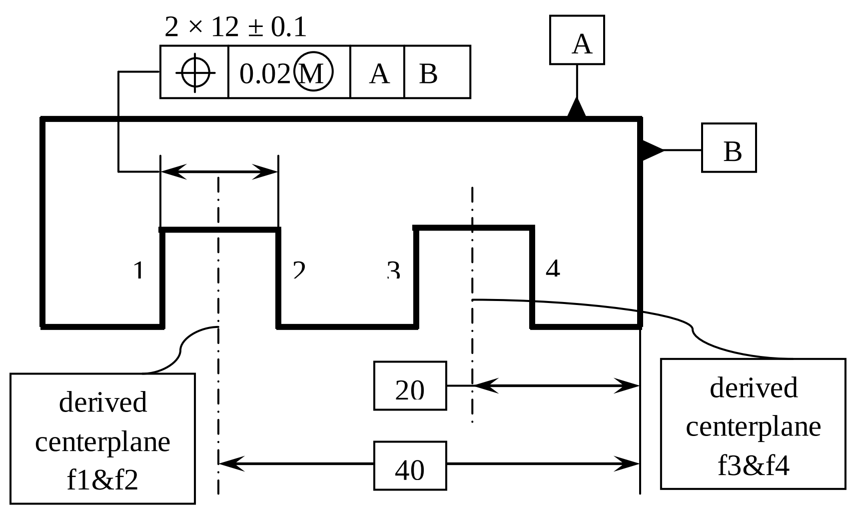


Figure 3.5 GD&T specifications for a part with two through slots [Shen Z. 2005]

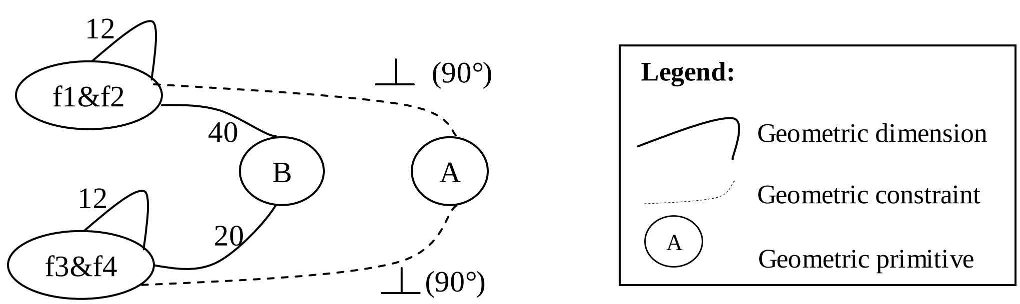


Figure 3.6 The C-Graph for this part [Shen Z. 2005]

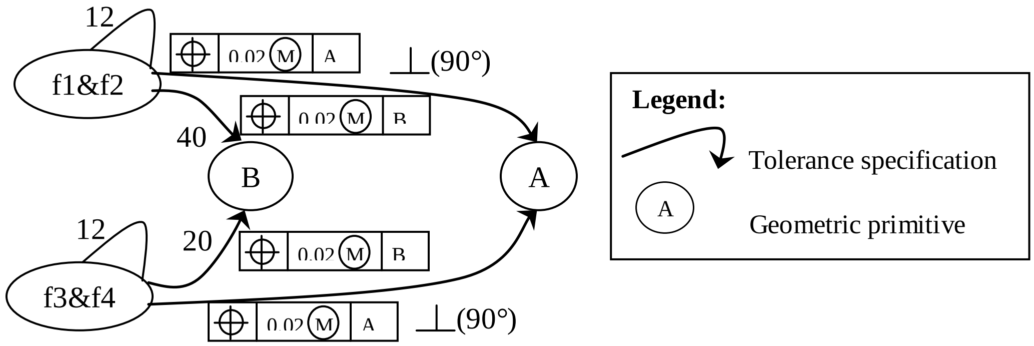


Figure 3.7 The T-Graph for this part [Shen Z. 2005]

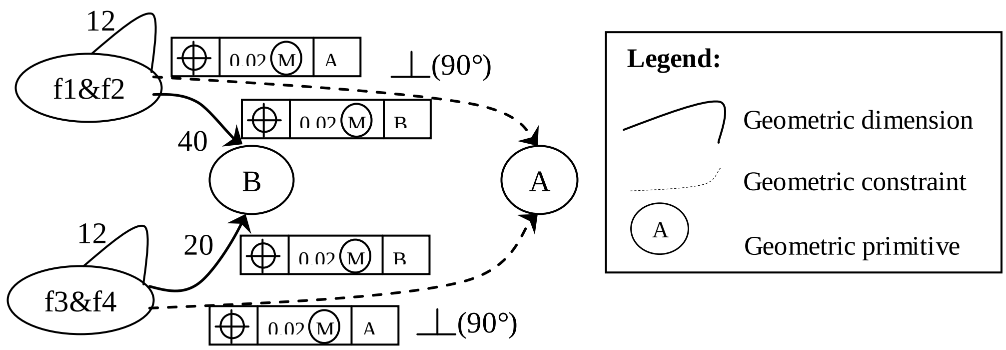


Figure 3.8 The combined CTF-Graph for this part [Shen Z. 2005]

CHAPTER 4 DOF ALGEBRA FOR GEOMETRIC TOLERANCING

This chapter will introduce entity clusters composed of point, line, and plane primitives with different geometric relations to each other (coincident, parallel, perpendicular etc.). The ASME Y14.5.1 companion to the Geometric Dimensioning and Tolerancing (GD&T) standard gives an exhaustive tabulation of active and invariant degrees of freedom (DOF) for Datum Reference Frames (DRF). A systematic DOF algebra is proposed and validated by applying it to all cases in the Y14.5.1 tabulation; the results are consistent with rules in the standard. Of course, only a handful of DRF cases in the standard have practical use. This thesis investigates the systematic derivation of DRF and target clusters' DOFs and associates them with tolerance classes, which will be discussed in chapter 5.

4.1 Development of DOF algebra

A datum provides a reference for measurements. The type and number of datums needed depends on the tolerance class. The concept of the datum reference frame is given by ASME Y14.5M-2009. A datum is theoretically a point, axis, or plane. A Datum Reference Frame (DRF) may consist of up to three interrelated datums to establish a coordinate system (CS) for measurement. Figure 4-1 shows a typical datum reference frame formed by orthogonal planes. A datum is simulated by inspection equipment such as machine tables and surface plates in contact with designated features of parts to be inspected. The specification of GD&T is initially dependent on the establishment of DRF which is closely associated with the function of the part in the assembly, the shape and quality of the real part faces. A datum can be a feature (face, edge) or resolved geometry of a feature of size (axis of cylinder, middle of a slot, etc.). Three datums in precedence are labeled as primary, secondary, and tertiary datum respectively.

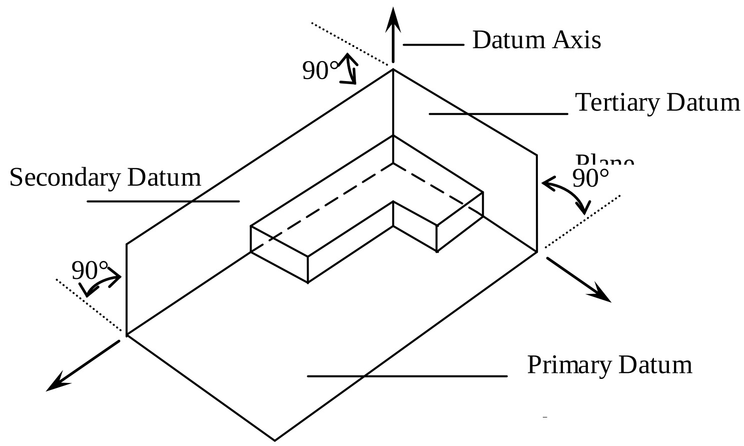


Figure 4.1 Datum reference frame formed by orthogonal planes

Many past studies [A. Desrochers and A. Clement 1994; Kramer 1992] have shown that a vast majority of features on parts can be represented by points, lines (or axes), planes and combinations of those. ASME Y14.5.1 enumerates all 52 cases of valid DRFs composed of various combinations of points, lines, and planes which are classified into six groups based on invariant and free DOFs. The purpose of this section is to present systematic derivation of active/invariant DOFs. Since DRF and target DOFs determine the type of tolerance classes that can be applied, this methodology can be used in tolerance validation.

When combining entities into clusters it is necessary to use a common coordinate system (CS). If the entity CS axes are not identical to those of the cluster, a coordinate transformation is required. In most DRF cases, datum features are orthogonal or collinear. We therefore consider only orthogonal transformations. Such transformations can be done simply by rearranging the elements of the DOF vector. We define the CS transformation operator OP i→j as the swapping of i and j element values for both TDOF and RDOF. Thus OP z→y {[110,110]} = [101,101]. Figure 4-2 shows coordinate system transformation operation applied to a line. This is valid for any entity or cluster and for active or invariant DOF vector. Also note that the direction of translation or rotation does not matter.

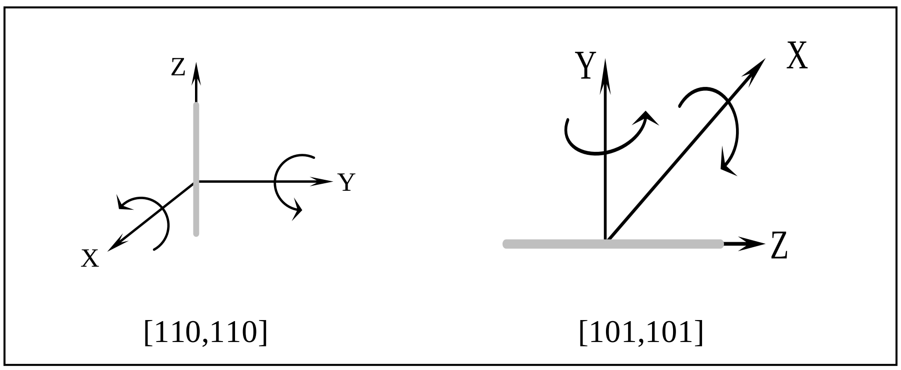


Figure 4.2 CS transformation operation applied to a line

In GD&T, a datum reference frame is composed of up to three geometric features. It is imperative to express the DRF in mathematical representation. In order to represent mathematically the combination of a number of geometric primitives, two Boolean operations are employed in this algebra: Union and Intersection. This is done by performing the union or intersection operation on two DOF vectors. These operators are analogous to Binary Logic OR/AND gates. Table 4-1 shows Boolean operations for two DOF vectors. For example, there are two DOF vectors, [110,110] and [001,110]. The result of union operation on these two vectors is [111,110]; the result of intersection operation on two vectors is [000,110]. In addition, some common logical relations are also valid in DOF algebraic computation. Standard associative, distributive, and idempotence relations apply to Boolean operations as well. Table 4-2 shows four typical algebraic relations. A or B represents an individual DOF vectors in operations. "⋃" is union operation. "⋂" means intersection operation. "∅" represents null set in which each position is equal to zero in the vector. "**I**" is an identity vector in which every position is equal to one. "RCP" is the operation to set vector to reciprocal.

Table 4.1 Boolean operations for two DOF vectors [110,110] and [001,110]

Union Operation ⋃			Intersection Operation ⋂		
a_i	b_i	$a_i \cup b_i$	a_i	b_i	$a_i \cap b_i$
1	0	1	1	0	0
1	0	1	1	0	0
0	1	0	0	1	0
1	1	1	1	1	1
1	1	1	1	1	1
0	0	0	0	0	0

Table 4.2 Algebraic relations

Commutative	$[A] \cup [B] = [B] \cup [A]$
Null Set	$[A_{DOF}] \cap [A_{INV}] = [\emptyset] = [000,000]$
Identity Vector	$[A_{DOF}] \cup [A_{INV}] = [\mathbf{I}] = [111,111]$
Reciprocal	$[A_{INV}] = RCP\ \{[A_{DOF}]\}$

The above DOF vector definitions, CS transformation, and Boolean operations can now be used to compute the cluster DOF vector from its composing entity vectors. For example, suppose a cluster X is formed by entity/cluster A and B. Then, the rule for computing the degree of freedom of each DRF cluster can be written as: (1). $[X_{DOF}] = [A_{DOF}] \cup [B_{DOF}]$; (2). $[X_{INV}] = [A_{INV}] \cap [B_{INV}]$. Figure 4-3 shows a few more examples of cluster DOFs derivation to demonstrate the computation rules.

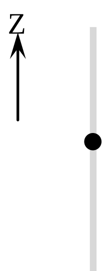
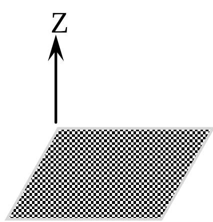
Case 1. Point-Line (coincident)	
	Entity DOFs: Point A: A_{DOF} = [111,000] A_{INV} = [000,111] Line B: B_{DOF} = [110,110] B_{INV} = [001,001] Combined DOFs: $[(AB)_{DOF}] = [A_{DOF}] \cup [B_{DOF}]$ = [111,110] $[(AB)_{INV}] = [A_{INV}] \cap [B_{INV}]$ = [000,001]
Case 2. Point-Plane (coincident)	
	Entity DOFs: Point A: A_{DOF} = [111,000] A_{INV} = [000,111] Plane C: C_{DOF} = [001,110] B_{INV} = [110,001] Combined DOFs: $[(AC)_{DOF}] = [A_{DOF}] \cup [C_{DOF}]$ = [111,110] $[(AC)_{INV}] = [A_{INV}] \cap [C_{INV}]$ = [000,001]

Figure 4.3 Examples of cluster DOFs derivation

Case 1 and 2 do not involve any CS transformation or indirect definition of primitives. We use A, B, C to imply point, line, and plane respectively. In Case 1, only one RDOF (about the line) is invariant, while in Case 2 only the RDOF about the normal remains. Figure 4-4 shows more complex examples. Case 3 involves CS transformation prior to Boolean operations. In case 4, the primitive entities are indirectly generated; two orthogonal lines define a plane and a point. Thus, the above result should be identical to that obtained for point A, line B, and C coincident cluster, as shown: $[(ABC)_{DOF}] = [(AC)_{DOF}] \cup [OP_{z\rightarrow x}\{B_{DOF}\}] = [111,110] \cup [011,011] = [111,111] = [I]$.

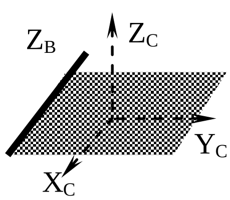
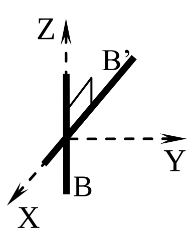
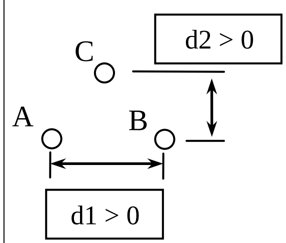
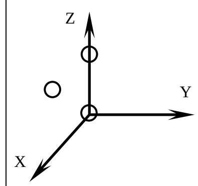
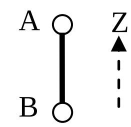
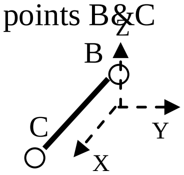
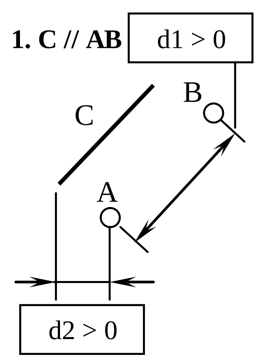
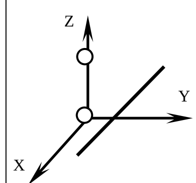
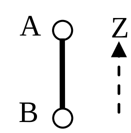
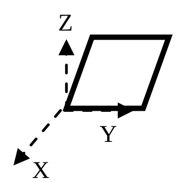
Case 3. Line-Plane (co-planar) Line CS needs to be transformed to plane CS	
	Entity DOFs: Plane C: $C_{DOF} = [001,110]$ $C_{INV} = [110,001]$ Line B: $B_{DOF} = [110,110]$ $B_{INV} = [001,001]$ Combined DOFs: $[(BC)_{DOF}] = [\mathbf{OP}_{z\rightarrow x}\{B_{DOF}\}] \cup [C_{DOF}]$ $= [011,011] \cup [001,110] = [011,111]$ $[(BC)_{INV}] = [\mathbf{OP}_{z\rightarrow x}\{B_{DOF}\}] \cap [C_{INV}]$ $= [100,100] \cap [110,001] = [100,000]$
Case 4. Line-Line (orthogonal) B' CS needs to be transformed	
	Entity DOFs: Line B: $B_{DOF} = [110,110]$ $B_{INV} = [001,001]$ Line B': $B'_{DOF} = [011,011]$ $B'_{INV} = [100,100]$ Combined DOFs: $[(BB')_{DOF}] = [\mathbf{OP}_{z\rightarrow x}\{B'_{DOF}\}] \cup [B_{DOF}]$ $= [011,011] \cup [110,110] = [111,111] = [\mathbf{I}]$ $[(BB')_{INV}] = [\mathbf{OP}_{z\rightarrow x}\{B'_{DOF}\}] \cap [B_{INV}]$ $= [100,100] \cap [001,001] = [000,000]$

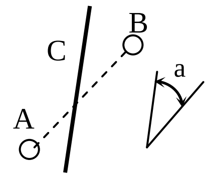
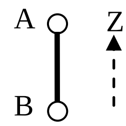
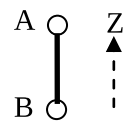
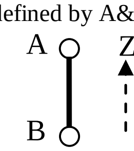
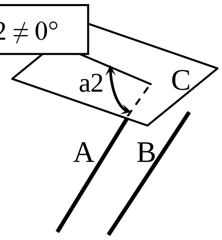
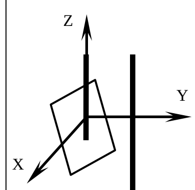
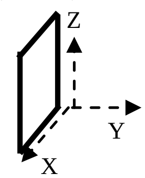
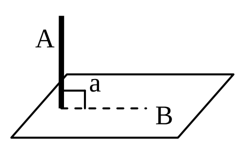
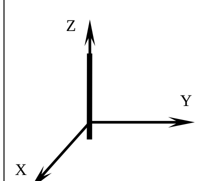
Figure 4.4 Complex examples of clusters DOFs derivation

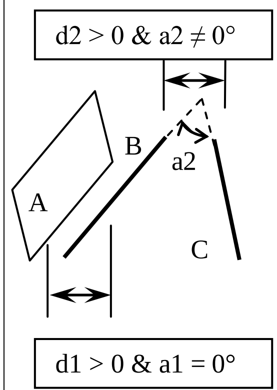
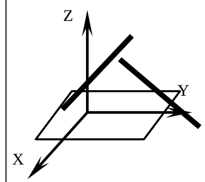
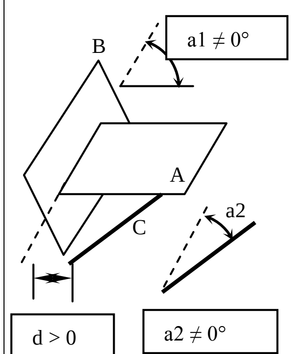
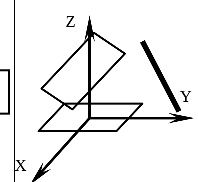
4.2 Combining entities and their DOFs in a DRF

Until now, the DOF algebra has been presented. By using this algebra, we are able to derive the active and invariant DOFs for all cases tabulated in ASME Y14.5.1 standard. Points, lines, and planes are not always directly specified as such. There are indirect ways in which a line can be defined. For example, an implicit line is defined by two non-coincident points or by the intersection of two planes. Similarly, a plane can be defined by two intersecting lines or a line and non-coincident point. Therefore, we also require algebra rules for including implicit entity definitions. All cases can be classified into six cases based on the derived DOF vectors. This is also in agreement with the six control frames discussed in Chapter 3. Table 4-3 shows a few examples derivation of DRF and combined DOF vectors.

Table 4.3 Examples derivation of DRF and combined DOF vectors

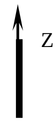
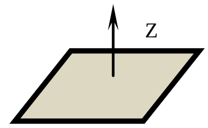
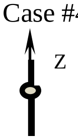
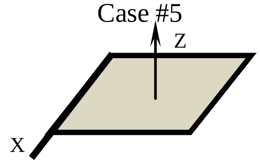
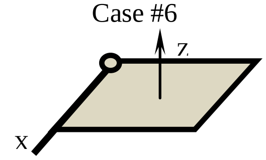
Case	Datums			Validity Conditions (Corresponding to ASME Y14.5.1 – 1994 standard)	Condition in CS	Indirect Entity Type
	A	B	C			
1.3	PT	PT	PT	(A ≠ B) ^ (C ⊄ {L1 A-B}) ($X_A = X_B$; $Y_A = Y_B$; $Z_A \neq Z_B$; C ⊄ AB) 		1. A line defined by points A&B  2. A line defined by points B&C 
1.4	PT	PT	AX	(A ≠ B) ^ (C ≠ {L1 A-B}) ($X_A = X_B$; $Y_A = Y_B$; $Z_A \neq Z_B$; C ≠ AB) 		1. A line defined by A&B  2. Case 1: a plane defined by A&C (PL⊥X) 

1.4	PT	PT	AX	**2. C ∦ AB**  d2 ≥ 0 & a≠ 0°		3. Case 2: a line defined by A&B  4. Case 3: a line defined by A&B  5. Case 4: a line defined by A&B 
2.11	AX	AX	PL	(A≠B) ^ (A//B) ^ ¬ (A//C) a2 ≠ 0°  d1 > 0 & a1 = 0°		1. Case 1: a plane defined by A&B (PL ⊥ Y)  2. Case 2&3: the combination of A&B&C has restrained the all Dofs in this DRF, thus we don't need to consider the indirect entity
2.14	AX	PL	-	A ⊥ B  a = 90°		

3.12	PL	AX	AX	(A//B)^ ¬ (B//C) 		
3.17	PL	PL	AX	¬ (A//B) ^ ¬ (C//{L1(A∩B)}) 		

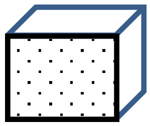
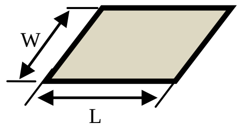
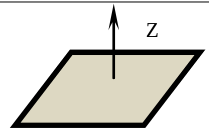
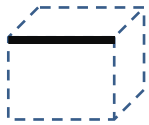
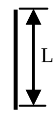
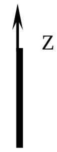
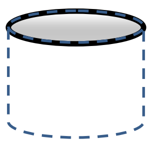
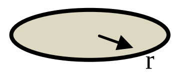
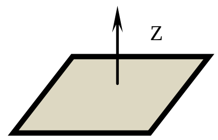
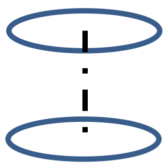
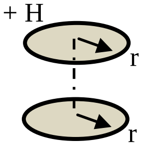
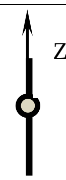
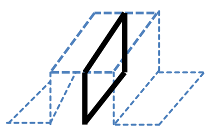
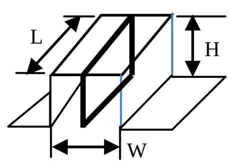
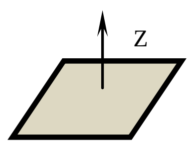
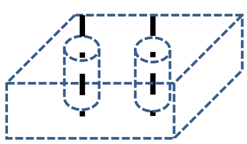
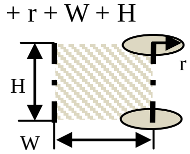
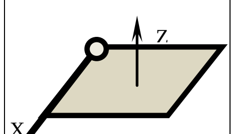
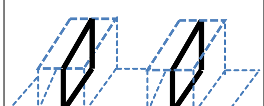
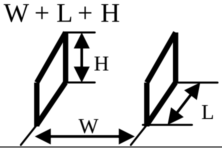
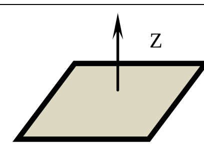
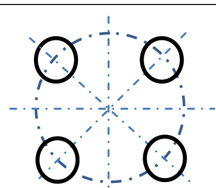
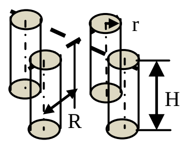
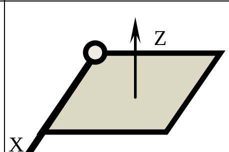
By looking at all of the active and invariant DOFs an geometric entity may have, all the geometric entities in 3-D space can be boiled down to three primitive entities (i.e. point, infinite line and infinite plane) and three independent combinations[Turner, J.U., 1993]. Table 4-4 shows the active and invariant DOFs of the primitives and their combinations. In this paper these six cases will be referred to as control frames (CF). Each control frame can constrain a certain combination of translational and rotational DOFs.

Table 4.4 Primitive control frames

Case #1  X_{DOF} : (111,000) X_{INV} : (000,111)	Case #2  X_{DOF} : (110,110) X_{INV} : (001,001)	Case #3  X_{DOF} : (001,110) X_{INV} : (110,001)
Case #4  X_{DOF} : (111,110) X_{INV} : (001,001)	Case #5  X_{DOF} : (011,111) X_{INV} : (100,000)	Case #6  X_{DOF} : (111,111) X_{INV} : (000,000)

Now I can extend the application of the algebra to tolerance entities and classes. The target entity is the geometric feature whose variations are controlled by a tolerance specification. Similar to datum features, most target entities can be represented as point (center of sphere), line (boundary of geometry, axis of cylinder, etc.), and plane (face, mid-plane). Note that we are not considering freeform profiles in this thesis. Target entities can also be considered to have six degrees of freedom: three translations and three rotations. The range of allowed nominal shape, size, and location of every geometric target entity must be specified with respect to other geometric entities on a part. While datum features are considered as infinite, target entities are considered trimmed within the dimensions of the feature. The active and invariant DOFs of target entities and clusters can be computed in the same way as for DRFs. Table 4-5 demonstrates the target entity clusters and their corresponding control frame. Besides the active DOFs that define a primitive feature and their combinations, additional geometrical parameters are used to define or locate the target features. For instance , a pin feature and a conical feature are both of a primitive line feature. A pin has one diameter, a height, and a type attribute, while a conical has two heights, one cone angle, and a type attribute. The target feature's active DOFs and possible parameters are also listed in table 4.5.

Table 4.5 Target entity clusters and their DOF vectors

Target Feature	Target Cluster	DOFs & possible parameters	Control frame	DOF Vector
rectangular face		TDOF + 2RDOFs + L + W 		(001,110)
edge		2TDOFs + 2RDOFs + L 		(110,110)
circular face		TDOF + 2RDOFs + r 		(001,110)
hole/pin		3TDOFs + 2RDOFs + r + H 		(111,110)
tab/slot		TDOF + 2RDOFs + W + L + H 		(001,110)
linear holes		3TDOF + 3RDOFs + r + W + H 		(111,111)
linear slots		TDOF + 2RDOFs + W + L + H 		(001,110)
radial holes		3TDOFs + 3RDOFs + R + r + H 		(111,111)

CHAPTER 5 TOLERANCE SEMANTICS

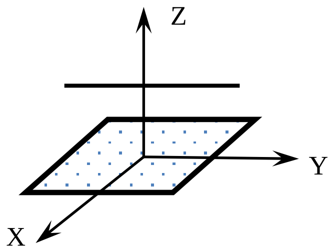
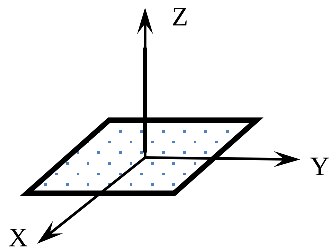
Now that we know how to characterize DRFs and TCs (target cluster) in terms of active and invariant DOFs, we would like to investigate their role in tolerance modeling and validation. Specifically, can we answer the following types of question: **(1)** Given a tolerance frame (TC, DRF, tolerance class), can we verify if the intended DOFs will be controlled? **(2)** Given a TC and DOF control desired, can we determine candidate DRF clusters that would be needed? **(3)** Given all tolerances on a TC, can we identify redundant or conflicting specifications? We believe these questions can be answered by juxtaposing DOFs related to tolerance classes, DRFs, and TCs.

5.1 Controllable DOFs by tolerance classes

Each tolerance class is available to certain types of target features. Each tolerance class actually controls the variations of certain DOFs of the target cluster. Meanwhile, the DOFs of tolerance zones are constrained by the common freedoms between the target cluster and assigned DRF. Position tolerance involves the distance relation between the target cluster and DRF cluster. A distance relation controls TDOFs of an entity. A distance relation sometimes also implies a parallel relation, such as a distance between two lines or planes. In this case, RDOFs are also constrained by the distance relation. This is to say, a distance relation basically controls the location of an entity. Orientation tolerance implies angular control between the target cluster and DRF cluster. Angular relations control RDOFs of geometric entities; essentially, constraining the orientation of geometric entities. Runout tolerance is applied on cylinder face which is trimmed down as an axis. Similar to orientation tolerance, runout places constraints on RDOFs of geometric entities. Since the tolerance specification includes three elements: target feature, datum reference frame, and tolerance classes, we need to represent the tolerance classes into DOF vectors. For example, orientation tolerance has constraints on RDOFs of geometric entities; therefore we represent it as (000,111). Consequently we represent it as (000,111).

Table 5-1 shows an example of calculation of constrained DOFs by tolerance classes. Going back to the example in Chapter 3 that demonstrates the metric relations, there is a plane and a line with parallel metric relation. The plane is referenced as datum and is coincident with the XY coordinate plane in the Figure 3.4. The line has two TDOFs and two RDOFs and the plane has one TDOF and two RDOFs. The maximum common DOFs between the line and the plane are the TDOF along the Z axis and the RDOFs around the X axis. If the position tolerance is applied to the line with respect to the plane and it is assumed that the plane is fixed, the TDOF along the Z axis and the RDOF around the X axis of the line are constrained. That means, the line cannot translate along the Z axis and cannot rotate along the X axis. If the parallelism tolerance is placed on the line with respect to plane, only one RDOF around the X axis of the line is constrained, the line is still free to move along the Z axis.

Table 5.1 Calculation of constrained DOFs

Case / **DOF Class**		
Target Line X_{DOF}	(101,101)	(101,101)
Datum Plane X_{DOF}	(001,110)	(001,110)
Tolerance Type X_{DOF}	(111,111)	(000,111)
Controlled DOFs X_{DOF}	(001,100)	(000,100)

The metric relations between the target entity and DRF can be extracted from the DOF vector of the target cluster and DRF cluster. In second row of Table 5.1, the target feature is a line (110,110) which is parallel with the datum line (110,110). If we transfer the DOF of target line by coordinate transformation to (101,101), the target is perpendicular to the datum line. Although the position of target feature with respect to DRF is varied, the Boolean operation is still valid for calculating the constrained DOFs. The above example demonstrates that the constrained DOFs are the intersection of the DOFs of the three tolerance elements. We introduce the Boolean operator " $\cap$ " for intersecting the constrained DOFs of tolerance specifications. The target, DRF, and tolerance class are completely represented in terms of DOF vector. From table 4-5 we know that target clusters can also be classified into six control frames. Table 5-2 presents a list of the constrained DOFs by associating the target feature with DRF clusters and tolerance types.

A number of tolerancing rules can be concluded based on the table 5.2. In the first row of the table, a point feature could represent a sphere. If a point feature is toleranced, the tolerance type could be position tolerance or concentric tolerance. Other tolerance types such as parallism and perpendicularity tolerance do not make any sense in this case. Runout tolerance is represented with vector (111,111) as well, but it is not applicable to target entity represented with vector (111,000). It is applicable to axis feature exclusively. In the second row, a line target feature is represented with vector (110,110). The orientation tolerance types are represented with vector (000,111). The location tolerance types are represented with vector (111,111). The location tolerance types include position tolerance, runout tolerance and coincidence tolerance. The eligible DRF candidates associated with orientation tolerance should be represented with vector (110,110) and (001,110). The candidates can be chosen from ASME Y14.5 based on the DOF representation. In same way, the row 3-6 can be analyzed with DOF vector representation. This table can provide designers with guidelines on how to choose tolerance types and DRF combinations according to target entity.

5.2 Tolerance validation based on the DOF algebra

Based on the observation and calculation of the DOF algebra and metric relations among tolerance elements, a set of control rules can be concluded. These rules contribute to validate the tolerance specifications. The validation of an

Table 5.2 Constrained DOFs for different combination of target, DRF candidates and tolerance classes

Control frame No.		1	2		3		4		5	6
Target		(111,000)	(110,110)		(001,110)		(111,110)		(110,111)	(111,111)
DRFs candidate (Control frame CF)	CF#1	(111,000)	-	-	-	-	-	-	-	-
	CF#2	-	(110,110)	-	(110,110)	-	(110,110)	-	-	-
	CF#3	-	(001,110)	-	(001,110)	-	(001,110)	-	-	-
	CF#4	(111,110)	-	(111,110)	-	(111,110)	(111,110)	(111,110)	-	-
	CF #5	-	-	(110,111)	-	-	-	-	(110,111)	-
	CF#6	(111,111)	-	(111,111)	-	(111,111)	-	(111,111)	(111,111)	(111,111)
Tolerance type	Orientation tol.	-	(000,111)	-	(000,111)	-	(000,111)	-	-	-
	Runout tol.	-	(000,111)	-	(000,111)	-	(000,111)	-	-	-
	Position tol.	(111,111)	-	(111,111)	-	(111,111)	-	(111,111)	(111,111)	(111,111)
Constrained DOFs		(111,000)	(000,110)	(110,110)	(000,110)	(001,110)	(000,110)	(111,110)	(110,111)	(111,111)

individual tolerance specification is to check all the components inside a tolerance specification.

(1) Tolerance type validation: The tolerance specified should be applicable to the target entity type.

(2) Tolerance specification has to be in association with the datum features and implied metric relations of datum features with target. For example, a concentricity tolerance should include a datum feature with an axis parallel with target feature or feature of size.

(3) The datum and target should have at least a common DOF that is constrained by tolerance class. For orientation tolerance, target cluster and DRF cluster must have common rotational DOFs. Similarly for position and runout tolerance, target cluster and DRF cluster must have common translational DOFs.

(4) Different tolerances in different tolerance classes control varying types of the target and the variations of different DOFs of the target. A tolerance specification can be used to control the variation of metric relations.

(5) Each datum feature contributes its controllable DOFs to its DRF. The controllable DOFs of a DRF are a union of all DOFs of its datum members. The advantage of DOF algebra is to validate the individual datum members. Previous work [Kandikjian, Shah, Davidson 1999] used control frame to validate DRFs. However, the control frame only contains the combined DOFs instead of individual DOF of each datum member. In this case, a control frame is not able to

tell designers if any redundant datum members exist because the datum member does not control more DOFs than previous datum members in the DRF.

(6) A DRF cluster for geometric tolerances is composed of one to three datum features. A datum feature is redundant in the DRF cluster if the datum feature does not control more DOFs than primary and secondary datum. In other words, the constrained DOF vector does not change after we taking into account the effect of the additional datum feature.

(7) Tolerance specification should be consistent with the target entity type. For instance, a cylindrical face would never be placed with a flatness tolerance specification.

These validation rules are summarized by the study of GD&T elements and the application of DOF algebra. In addition, there are heuristic based practice rules for the validation of tolerance specification. A summary of these rules are listed below [Ramaswamy S., 2001].

(8) When applying a straightness tolerance to the axis of a cylindrical feature with a large length to diameter ratio (> 4), it is better to apply a combined overall and unit length tolerance.

(9) It is beneficial to apply orientation tolerances at Maximum Material Condition (MMC) on features of size. Virtual condition boundaries are generated by the collective effect of the MMC of a feature of size and the geometric tolerance applicable at that size. The surface of the feature must not violate the virtual condition boundary. Hence, a single inspection gage (at the virtual condition of

the feature of size) will suffice to inspect orientation tolerances of any feature of size produced within the specified size tolerance, rather than requiring a series of gages or adjustable gages.

(10) When applying an angularity tolerance, it is advisable to use two datums to establish the desired level of control. This prevents the rotation of the toleranced feature around an axis perpendicular to the primary datum. Figure 5-1 shows a situation requiring the specification of an angularity tolerance. In the figure, if the face marked ‘X’ is to be assigned an angular tolerance with ‘A’ as reference, specifying ‘B’ as the secondary datum will prevent the rotation of ‘X’ around an axis perpendicular to ‘A’.

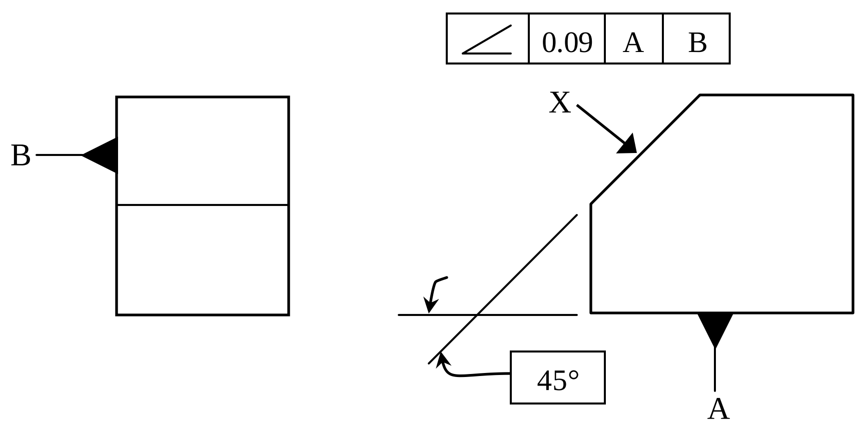


Figure 5.1 Datum requirement for Angularity tolerance specification

(11) While specifying a position tolerance for a cylindrical feature of size, it is best to select the face on which the feature “sits” as the primary datum. This ensures that the axis of the feature is perpendicular to the surface or at a basic angle to the surface.

(12) Concentricity tolerance must be avoided as much as possible. Position or runout tolerance can provide the same level of control and the inspection of these two types of tolerances are much easier in comparison to concentricity. During inspection, an acceptance test for concentricity tolerance involves the use of a simple full indicator movement (FIM) method. However, the rejection of a part requires differential measurements to be made since FIM also includes the out-of-roundness error, which is not a part of concentricity control. Runout control includes both concentricity control as well as form characteristics of the surface.

(13) When tolerancing cylindrical features that are relatively large in diameter and short in length, a planar face must be specified as the secondary datum.

(14) Consider a perpendicularity tolerance requirement between two faces, one significantly larger than the other; for instance, 20 x 5 and 5 x 5 (Figure 5-2). In such a situation, it is better to tolerance the face ‘A’, with ‘B’ as the datum, rather than vice versa. If ‘A’ were to be chosen as the datum, any slight surface variation in ‘A’, would cause a large angular deviation of ‘B’ from the theoretical datum plane formed by ‘A’. This also stresses the need for specifying a flatness tolerance for such a datum.

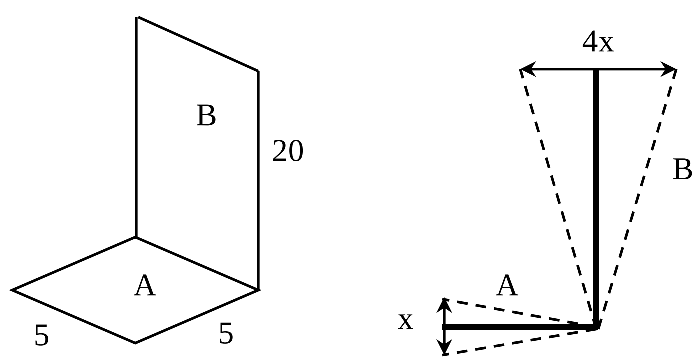


Figure 5.2 Datum selection depending on size [Ramaswamy, 2001]

(15) When an orientation tolerance is applied to a surface, with a plane as a primary datum, it is a good practice to apply a flatness tolerance to that plane. This ensures that the amount of deviation of the datum is minimized while forming the datum plane. (Note that three points are required on the primary datum for forming such a plane). An alternative to this is to specify datum targets (lines or points) on the surface so that poor surface form does not affect measurements. Datum targets are also used when the manufacturing process employed is casting or forging and machining is not cost effective in the situation.

(16) When a cylindrical feature of size is chosen as the primary datum, a cylindricity tolerance should be applied to the cylindrical face to minimize errors during inspection.

The GD&T validation rules assist in maintaining the syntax of the GD&T representation model. These rules are implemented to validate a tolerance specification. The detailed implementation will be introduced and discussed in Chapter 6 and 7.

CHAPTER 6 IMPLEMENTATION OF GD&T GLOBAL MODEL

This chapter will present computer implementation that incorporates GD&T specification and validation system. Section 6.1 will introduce the general architecture of the integrated GD&T system. Section 6.2 – 6.6 will cover each module in detail.

6.1 Global model system architecture

The integrated GD&T system is to implement the mathematical model discussed in chapter 4 and 5. The mathematical model defines three basic elements: target entity, metric relation, and tolerance specification in GD&T. Furthermore, the mathematical model provides an intelligent and robust DOF representation and algebra of three basic elements. A number of valuable and useful validation rules are found in the calculation to help perform tolerance validation.

The computer implementation developed in this thesis consists of GD&T specification module, metric condition verification module, DOF analysis module, and tolerance scheme advisor. As it is demonstrated in previous chapters, tolerance specification and metric relation is interrelated. The metric condition verification module is actually a sub-function of tolerance specification module. The whole system is implemented with computer language C++. Figure 6-1 shows an overview of the GD&T integrated system. The commercial geometric kernel, ACIS, provides the API and DI to retrieve the geometric data and computation of geometry.

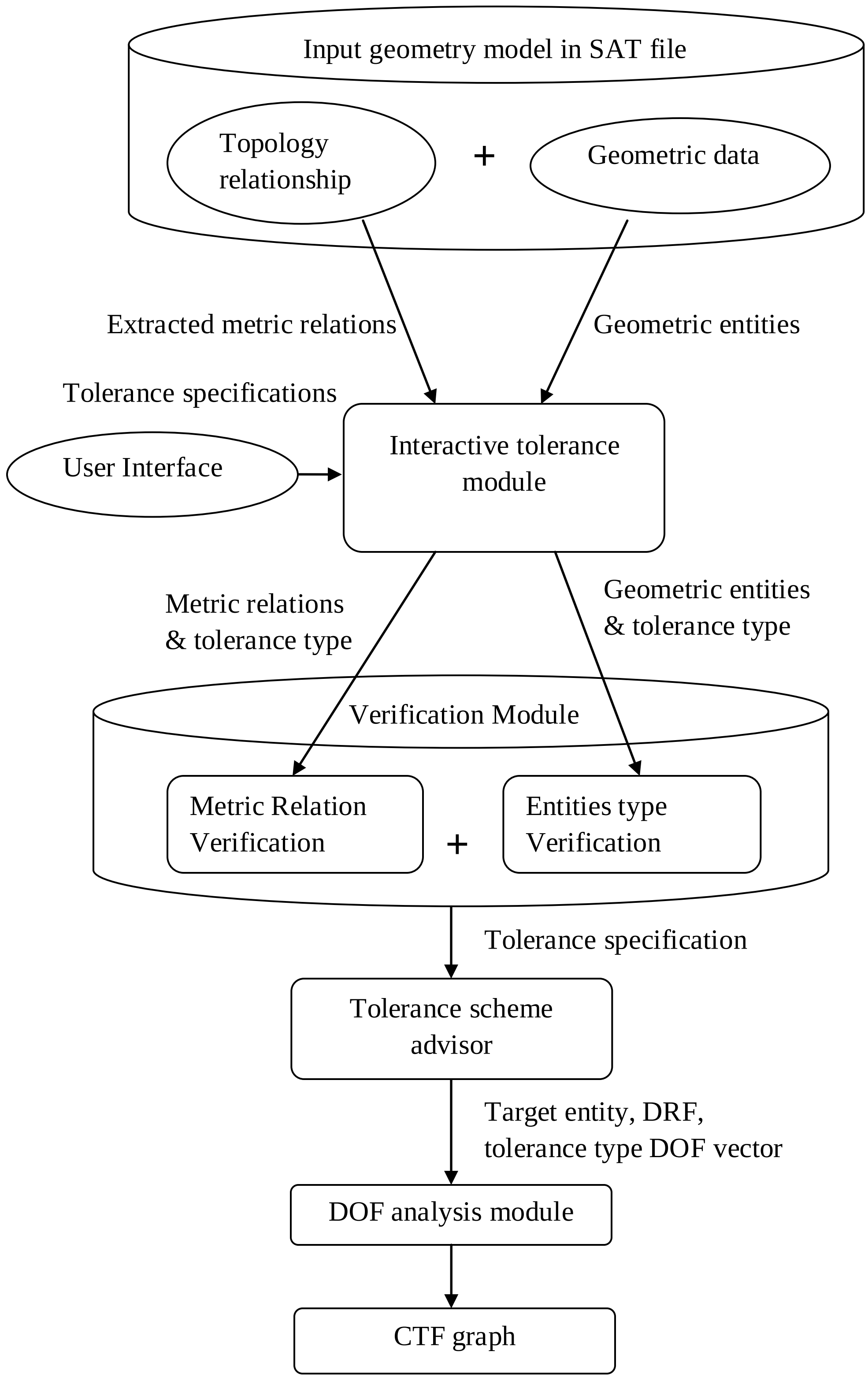


Figure 6.1 Integrated GD&T system overview

The output of the integrated GD&T is a CTF tree data structure for the input part or assembly. The input geometries are represented with a B-Rep CAD model and read in as SAT format. The GD&T data is attached as attributes to the entities. Most of them are faces. Once the geometry is read in, then tolerance specifications are instantiated interactively and validated according to control rules introduced in Chapter 5.

6.2 Macro data structure of GD&T global model

In the integrated GD&T system, solid models of geometric entities are the foundation for metric constraint and tolerance specification. Three data structures are created to store the information of the three basic elements in GD&T. The information of the geometric entity includes the fundamental geometric features such as dimensions and radius etc. The metric relations are attached as attributes to geometric entities. All metric relations include entities, metric relation type, and constraint value. The tolerance specification data structure represents the tolerance control frame. It encompasses the information of tolerance class, tolerance value, datum references, and material modifier.

As pointed out in Section 3.2, a point, line, or plane can correspond to many different geometric features. A line can represent a pin, a hole or a cone. It is rare to have tolerance specification on a pure line or an edge. A point represents a sphere. In the design and tolerancing stage, designers place tolerance specifications on faces and planes in a majority of time. In this implementation the faces and plane features are concentrated on due to their heavy use. Table 6-1 shows the classification of face features and feature characteristics. These faces and planes have common data to be used in CTF graph: root point, normal vector or axis direction. However, they do have different geometric characteristics. Pins and holes have radius and height. Planes have width and length. Slots and tabs have width, height and length. Circular planes have radius.

Table 6.1 Face classification and feature characteristics

Geometry	Feature Characteristics					
	Root Point	Normal/Axis	Radius	Height	Length	Width
Plane	●	●			●	●
Circle	●	●	●			
Pin/Hole	●	●	●	●		
Sphere	●	●	●			
Tab/Slot	●	●		●	●	●

The FeatureInfo class is the data structure of geometric features. Figure 6-2 shows the definition of the geometric feature class. The member variables in this class are FeatureAdd, FeatureType, FeatureID, RootPoint, Normal, Radius, Height, length, width, FeatureLineNumber, next. FeatureAdd is an ACIS entity pointer to store the geometric entities which are involved in tolerance specification. FeatureType indicates the type of a specific geometric entity type (such as circular plane, cylindrical face). FeatureID is a number attached to the geometric entity. It is useful when we need to track the geometric entity in the tree data structure. It can be used to identify the geometric entity. RootPoint and Normal are all essential geometric data to define the geometric entity. Radius and Height are used to define the cylinder or circular plane. When Height equals to zero the geometry is a circle. The member variables length and width are used to define a plane. They can be extracted or calculate by API and DI in ACIS. Since all geometric feature instances in this class need to be linked in the CTF graph, a class pointer next is generated to do this job.

```
//Restore the selected feature information

#ifndef FEATUREINFO_H
#define FEATUREINFO_H

struct FeatureInfo
{
    ENTITY *FeatureAdd;
    int FeatureType;
    int FeatureID;
    double RootPoint[3];
    double Normal[3];
    double Radius;
    double Height;
    double length;
    double width;
    FeatureInfo *next;
};
#endif
```

Figure 6.2 Definition of the geometric feature class

The ConstraintInfo class begins to work when the implied metric relation in tolerance specification needs to be recorded and verified. Its member variables contain all common data of geometric constraints. Figure 6-3 shows the definition of the geometric constraint class. These variables are ConstType, ConstValue, ConstEntity_1 & ConstEntity_2, ConstEntity_1_ID & ConstEntity_2_ID, ConstEntity_1_Type & ConstEntity_2_Type, mating_flag, next. ConstType is a pointer of integer. It indicates the type of a specific geometric constraint with a number. For example, the parallel constraint is labeled with 1 and the distance constraint is labeled with 2. The constraint type should be consistent with the appropriate tolerance class. ConstValue is the number used to record the value of the constraint. ConstValue could be angle, diameter or radius. The rest member variables in this class represent and record the geometric entities involved in the constraint. The member variable, mating_flag, indicates if the geometric constraint (metric relation) is a mating condition. If it is, mating_flag is set to 1; otherwise, it is set to 0. The class point next works to link all constraint together and display in the CTF graph.

The integrated GD&T program includes parallel, perpendicular, angle, concentric, coincident, diameter constraint. Since most metric relations are binary and include two geometric entities, we have two ACIS ENTITY pointer member variables: ConstEntity_1 and ConstEntity_2. For the class of a unary metric relation (such as diameter constraint), its member variables contain one ACIS ENTITY pointer ConstEntity_1 representing the geometric entity constrained by the geometric constraint. Meanwhile, ConstEntity_2 is set to NULL pointer. The instances of constraint class are stored in a linked list. A tolerance specification might involve target entities and DRF. The DRF would have up to three datum entities. Target entities have individual metric relation with each datum entity. There are accordingly three different instances of constraint classes in the list.

```
//Restore the selected constraint information

#ifndef CONSTRAINTINFO_H
#define CONSTRAINTINFO_H

struct ConstraintInfo
{
    int ConstType;
    CString ConstValue;
    ENTITY *ConstEntity_1;
    ENTITY *ConstEntity_2;
    int ConstEntity_1_ID;
    int ConstEntity_2_ID;
    int ConstEntity_1_Type;
    int ConstEntity_2_Type;
    ConstraintInfo *next;
};
#endif
```

Figure 6.3 Definition of the geometric constraint class

The ToleranceInfo class is the data structure that records all essential components involved in tolerance specification. The tolerance class hierarchy is shown in Figure 6-4. The ToleranceInfo class represents the feature control frame. Figure 6-5 shows the definition of the tolerance specification class. The member variable, tol_selection, is an int type variable to keep track of the tolerance selection in the tolerance specification user interface. The dia_symbol member variable indicates a cylindrical or spherical tolerance zone and is placed before the tolerance value. The tolerance value is represented by the variable tol_value. The target, primary, secondary, and tertiary datums are represented by the ACIS ENTITY pointer target, primary, secondary, and tertiary respectively. The default values of these variables are set to NULL. The material modifiers are recorded by four CString variables, target_modifier,

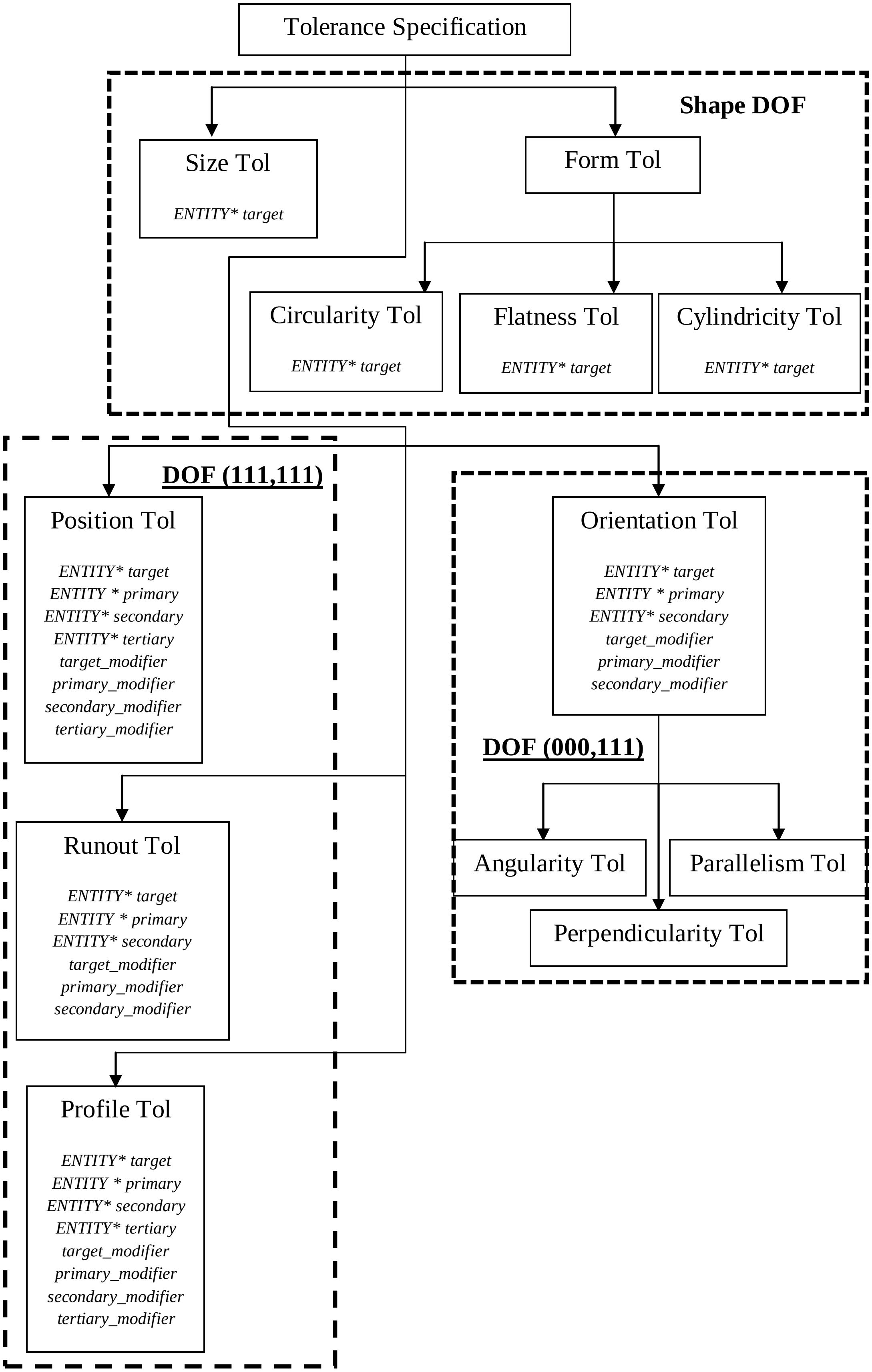


Figure 6.4 Representation of the tolerance information

primary_modifier, secondary_modifier, and tertiary_modifier. If no material modifier is applied, the RFS condition is activated. An instance of the ToleranceInfo class is instantiated by putting the tolerance information into the dialog box.

```
//each node represents a entity which has following information
//The nodes in a linked list is listed as the tolerance info in tolerancetreelist

#ifndef TOLERANCEINFO_H
#define TOLERANCEINFO_H

struct ToleranceInfo
    {
    //public:
        int tol_selection;
        int dia_symbol;
        int LineNumber;
        CString tol_value;
        CString target_modifier;
        CString primary_modifier;
        CString secondary_modifier;
        CString tertiary_modifier;
        ENTITY* target;
        ENTITY* primary;
        ENTITY* secondary;
        ENTITY* tertiary;
        ToleranceInfo * next;
    };
#endif
```

Figure 6.5 Definition of the tolerance specification class

6.3 User specified tolerances module

The tolerance specification module provides the user an interactive interface to specify tolerances on a part. This module plays a key role in this implementation because the input parameters in this module are also used by other modules. Figure 6-6 shows the interface of the module. The tolerances module consists of three stages. The first stage is the pre-checking stage to verify the compatibility between the tolerance and target entity type, the compatibility between tolerance type, and the implied metric relations. A set of validations discussed in Chapter 5 is carried out to perform the pre-checking function. The second stage is instantiating the tolerance specification. In this stage, the module lists all parameters in the tolerance specification which are selected or inputted by the end-user. The third stage is to output a CTF graph. This CTF graph will be validated by the DOF analysis module. Figure 6.7 shows the path to produce the CTF graph from the user specified tolerance dialog box.

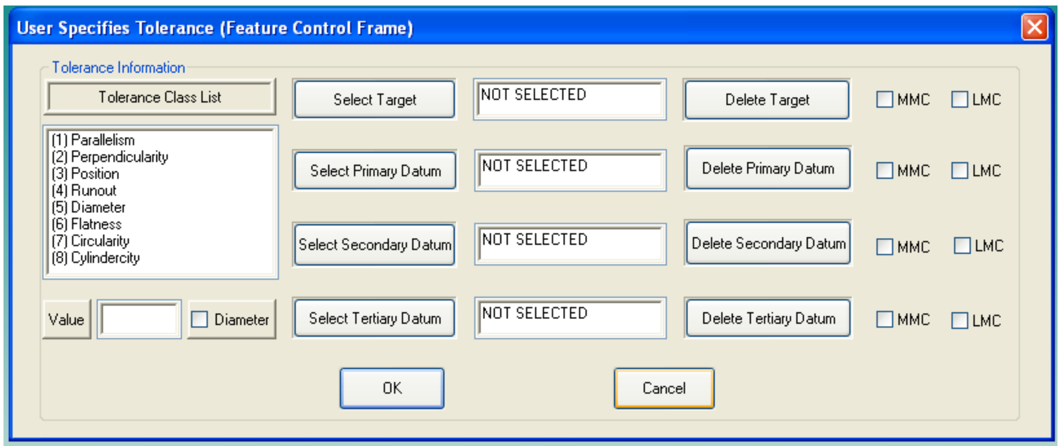


Figure 6.6 Interface of the tolerance specification module

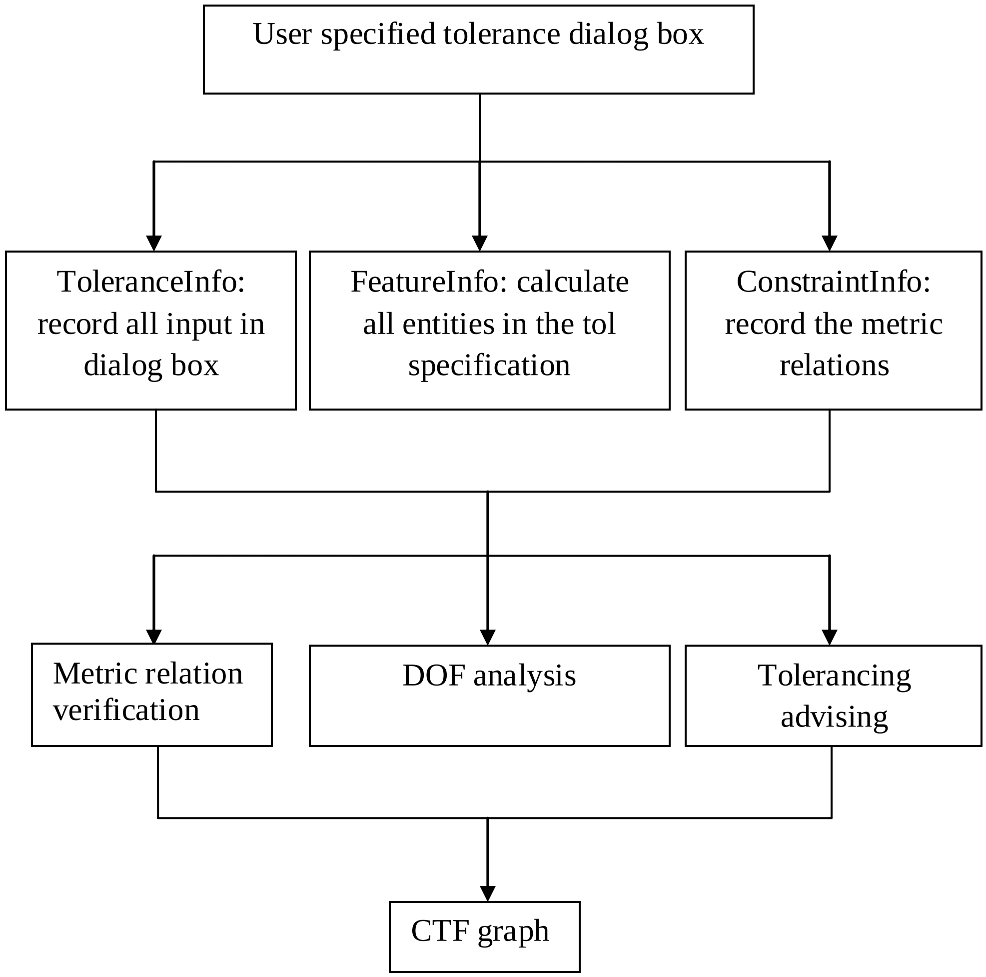


Figure 6.7 Path to produce CTF graph from dialog box

Although there are different tolerance classes involved in this module, the pre-checking stage for all the tolerance classes is the same. The first step is to validate the target entity type and the datum entity type according to the tolerance class (user specified tolerance module). The second step is to check if there is a corresponding metric relation to be toleranced between target and datums. This step will be performed by the metric relation verification module (see section 6.4) which is embedded in user specified tolerances module. Some metric constraints such as distance could be extracted in this step. If the implied corresponding metric relation does not exist and is not consistent with the target entities, the intended tolerance cannot be specified and an error would be reported to the user. The third stage is to record all validated tolerance information into a tolerance data structure instance. The module would deliver selected target and datum references to two sequential modules: metric condition verification module and DOF validation module. Figure 6.8 shows flow-chart of pre-checking for tolerance specification. Figure 6.9 to Figure 6.11 show the pre-checking stage for different tolerance type.

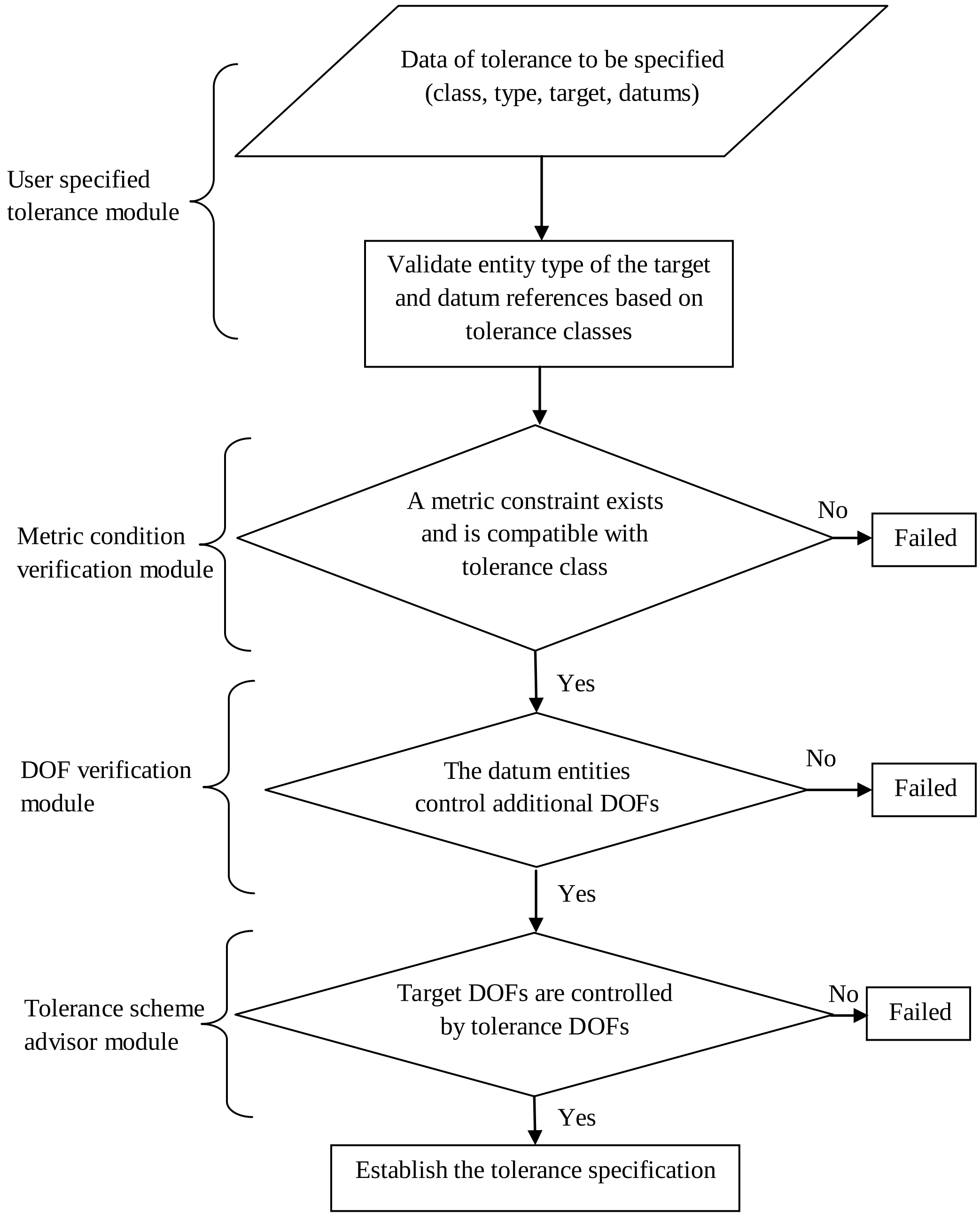


Figure 6.8 Flow-chart of pre-checking for tolerance specification

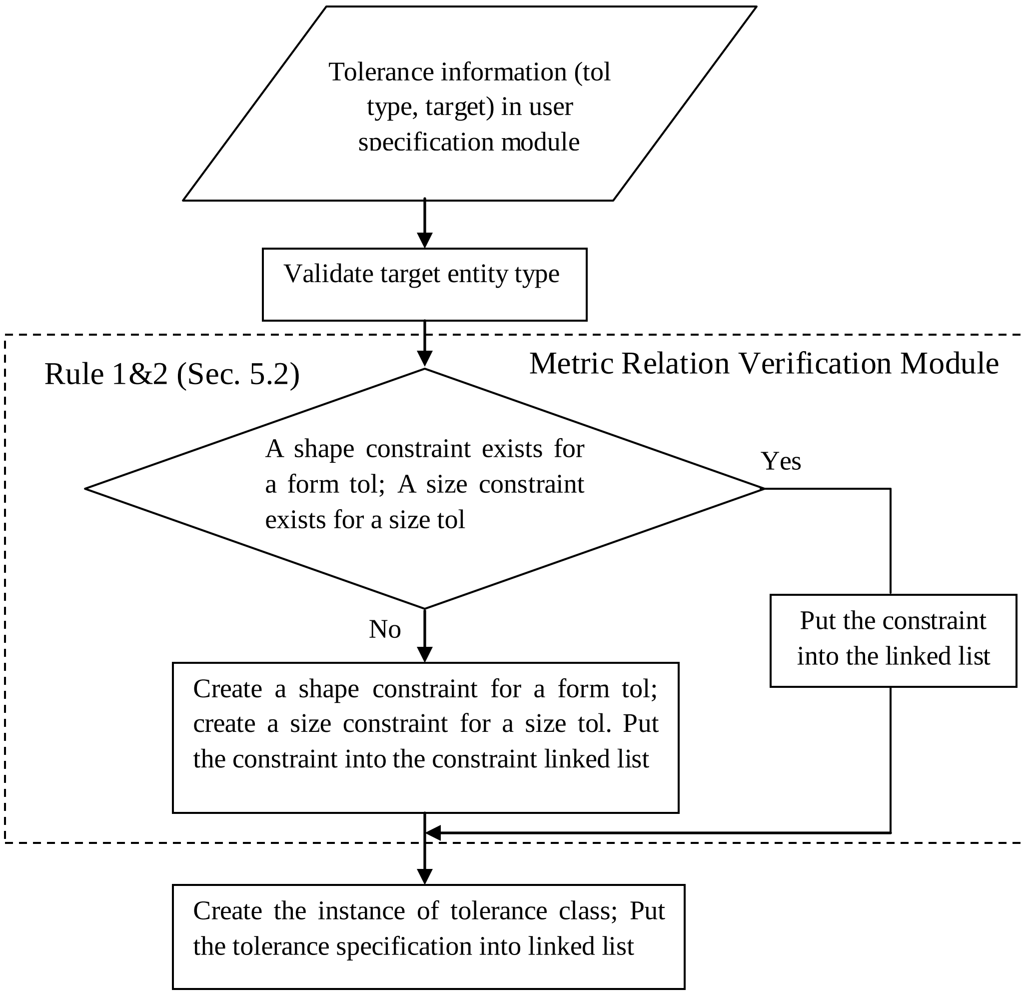


Figure 6.9 Pre-checking for size and form tolerance

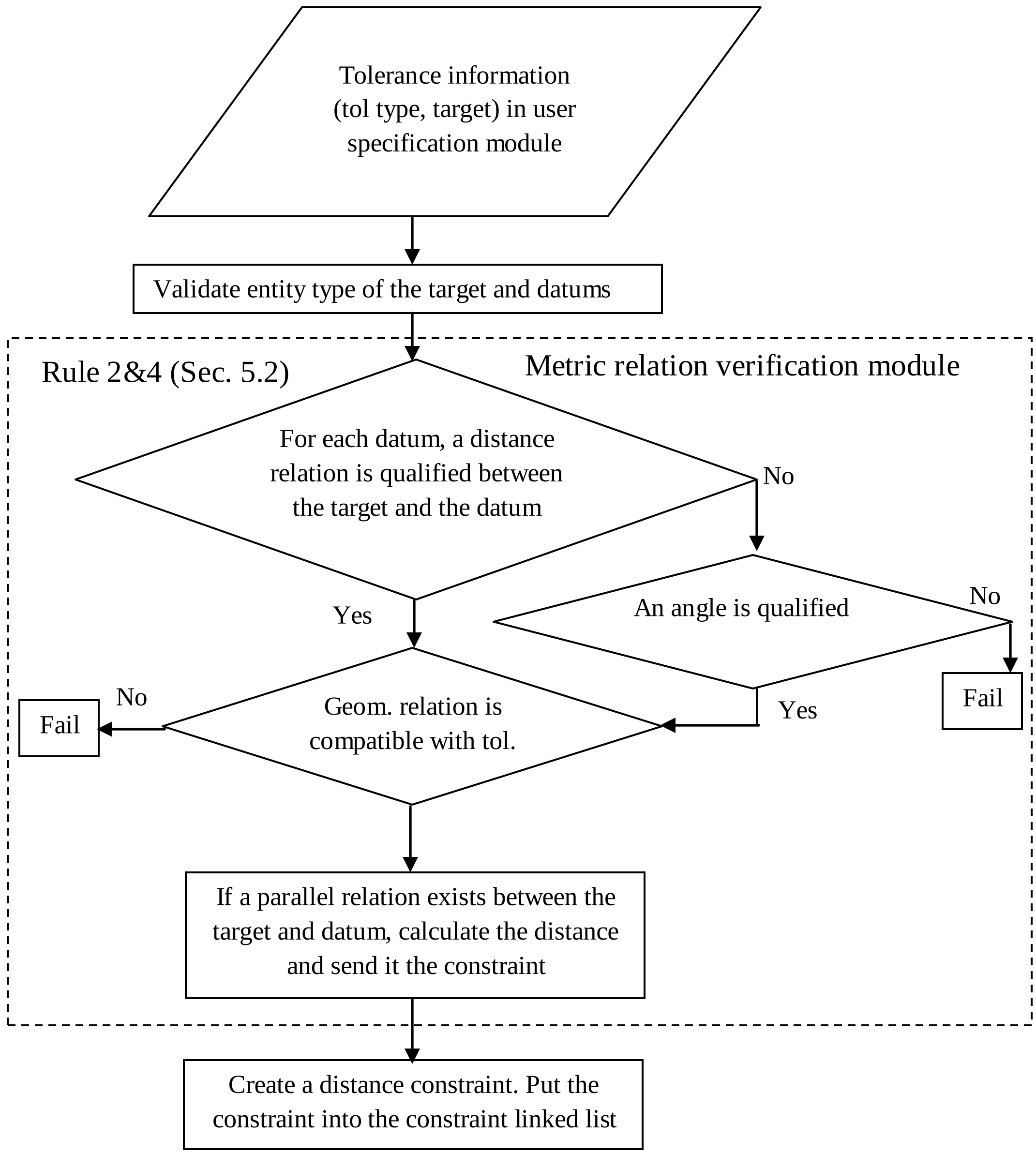


Figure 6.10 Pre-checking for position tolerance

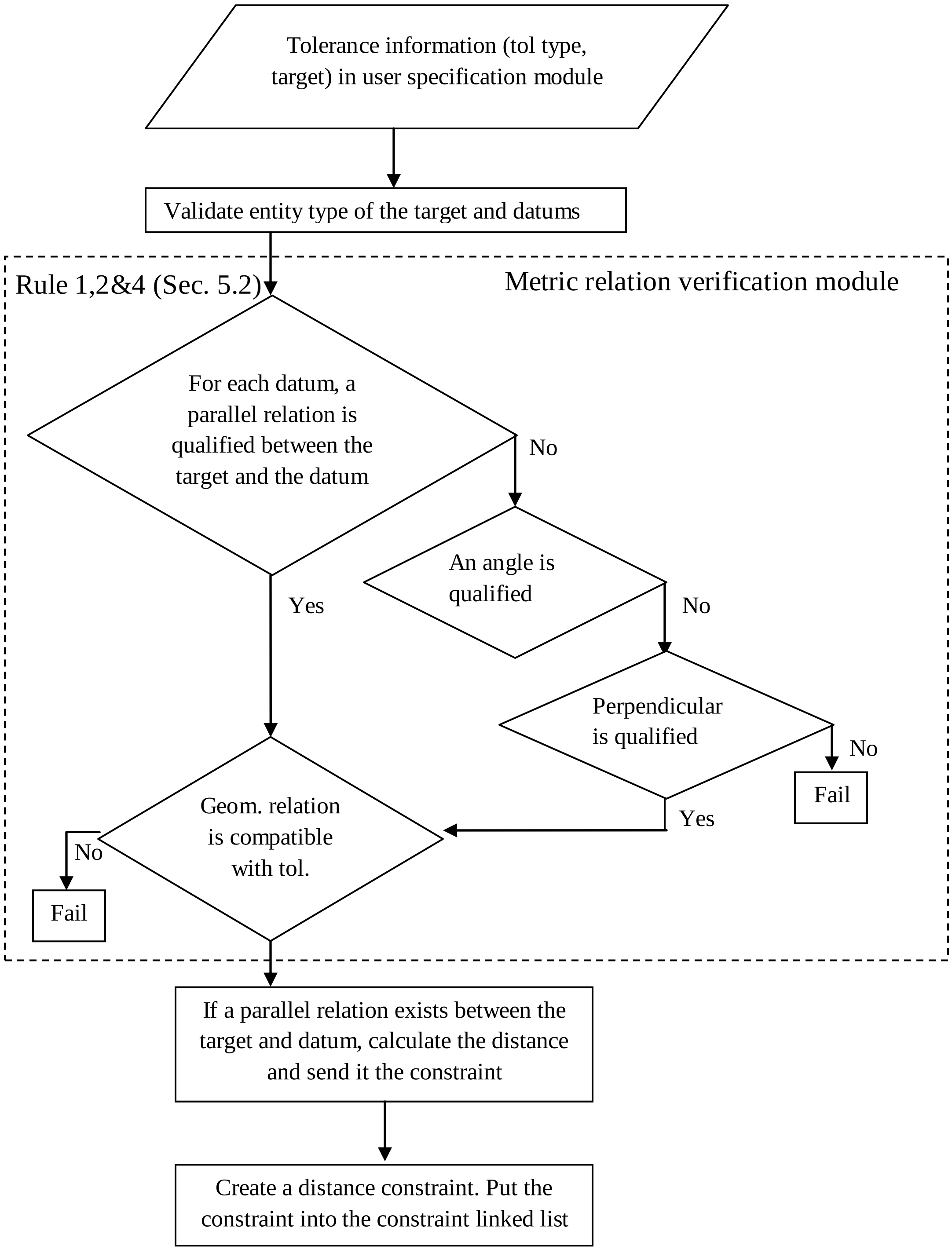


Figure 6.11 Pre-checking for orientation tolerance

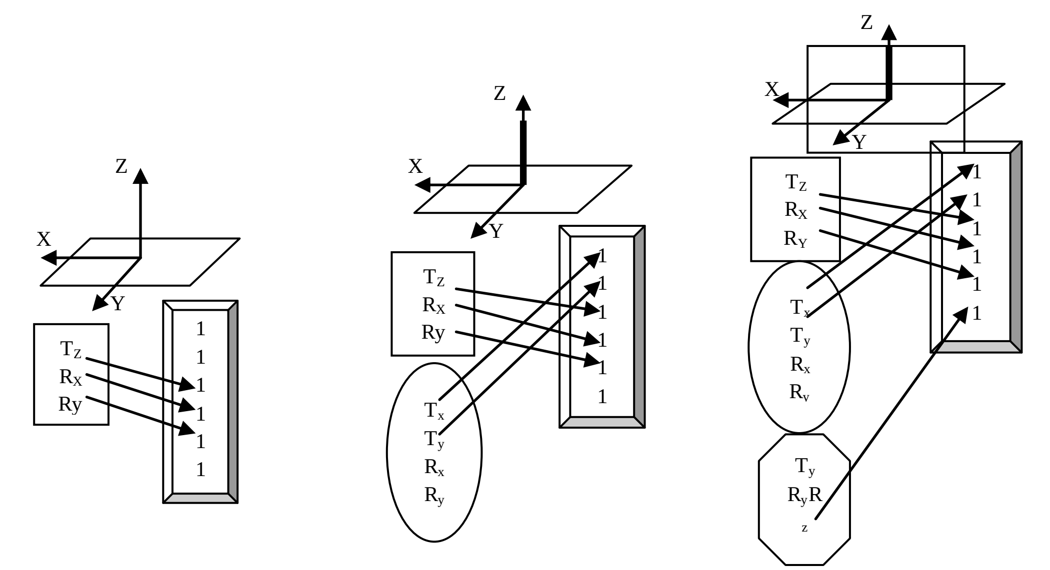
6.4 Metric condition verification module

Metric condition verification is an independent module to validate the compatibility of metric constraint and tolerance class (see Figure 6.8). It receives the entities and tolerance data from the tolerance specification module. This module consists of three steps. The first step is to automatically get essential data from a tolerance specification instance and store in a geometric constraint data structure. The second step is the validation process to provide metric condition verification. As stated before, this step is designed to check the compatibility of implied metric constraint and its tolerance specification. Any incompatibility would terminate the tolerance specification and will be notified to users as an error. If the compatibility is approved, the metric condition would be detected and recorded. The metric relation verification module is used to ensure that addition of the new metric relation will not introduce an over-constrained problem. If the corresponding metric relation does not exist and is not consistent with the part's geometry (such as applying a perpendicular relation required by a perpendicularity tolerance to a pair of parallel faces), the intended tolerance cannot be specified. When specifying a position tolerance, if a parallel constraint already exists between the same pair of geometric entities, the module would calculate the distance between them and record it as a distance constraint. A parallelism tolerance cannot only be attached to parallel constraint; it can also be attached to a distance constraint. When a parallelism tolerance is created, the system will check if there already exists a parallel constraint in the linked list. If the constraint does not exist, the parallelism tolerance will be attached to the parallel relations, which needs to be created if it does not exist and is applicable.

6.5 DOF analysis module

The DOF analysis module uses DOF algebra to validate the development of datum reference frame. The existence of a corresponding DRF is the pre-requisite for the existence of a tolerance type. DOF analysis translates the control rules into a mathematical operation which is easier to be detected and verified. The DOF module is able to check the tolerances specification and datum selection by comparing the constrained DOFs and the target DOFs or checking the DRF DOFs.

The work of determination and validation of DOFs within a tolerance specification using the DOF algebra method consists of three parts as well. Figure 6.12 shows the procedure of DOF analysis in the system. The purpose of the first part is to obtain datum features and set-up a coordinate system (CS) based on a given DRF cluster. The CS is not visible to the user and rather is recorded on the program's backend for future use. The second part determines the composite DOFs of combined entities in a DRF cluster and the combined target entities. The objective of the third part is to track the controlled DOFs of a target cluster by its corresponding DRFs and tolerance classes. Tolerance classes have their respective aimed DOFs. DOF vector representation implies the metric relations between target entities and DRF. If both of a target line and a datum line have the DOF vector (110,110), this case indicates that they are parallel with each other; If a target line has the DOF vector (110,110) and a datum line has the DOF vector (101,101), this case reveals the perpendicular metric relation between them; If a target line has the DOF vector (110,110) and a datum plane has the DOF (001,110), there is perpendicular metric relation between them; If a target line has the DOF vector (101,101) or (011,011) and a datum plane has the DOF vector (110,110), the metric relation between them is parallel. Boolean operator is enabled to calculate the constrained DOFs. The control rules discussed in Chapter 5 would instruct the program in order to perform the validation of DRF and tolerance classes. The datums in a DRF are supposed to control more DOFs than pervious datum. This rule avoids the redundancy of datums. Figure 6.13 shows the controlled DOF by each datum in DRF. The example of validation with DOF analysis module will be given in chapter 8.



(a) Controllable DOFs of the DRF after Datum A (plane) is added

(b) Controllable DOFs of the DRF after Datum B (line) is added

(c) Controllable DOFs of the DRF with Datum A, Datum B, and Datum C(plane)

Figure 6.13 Controlled DOF by each datum in DRF

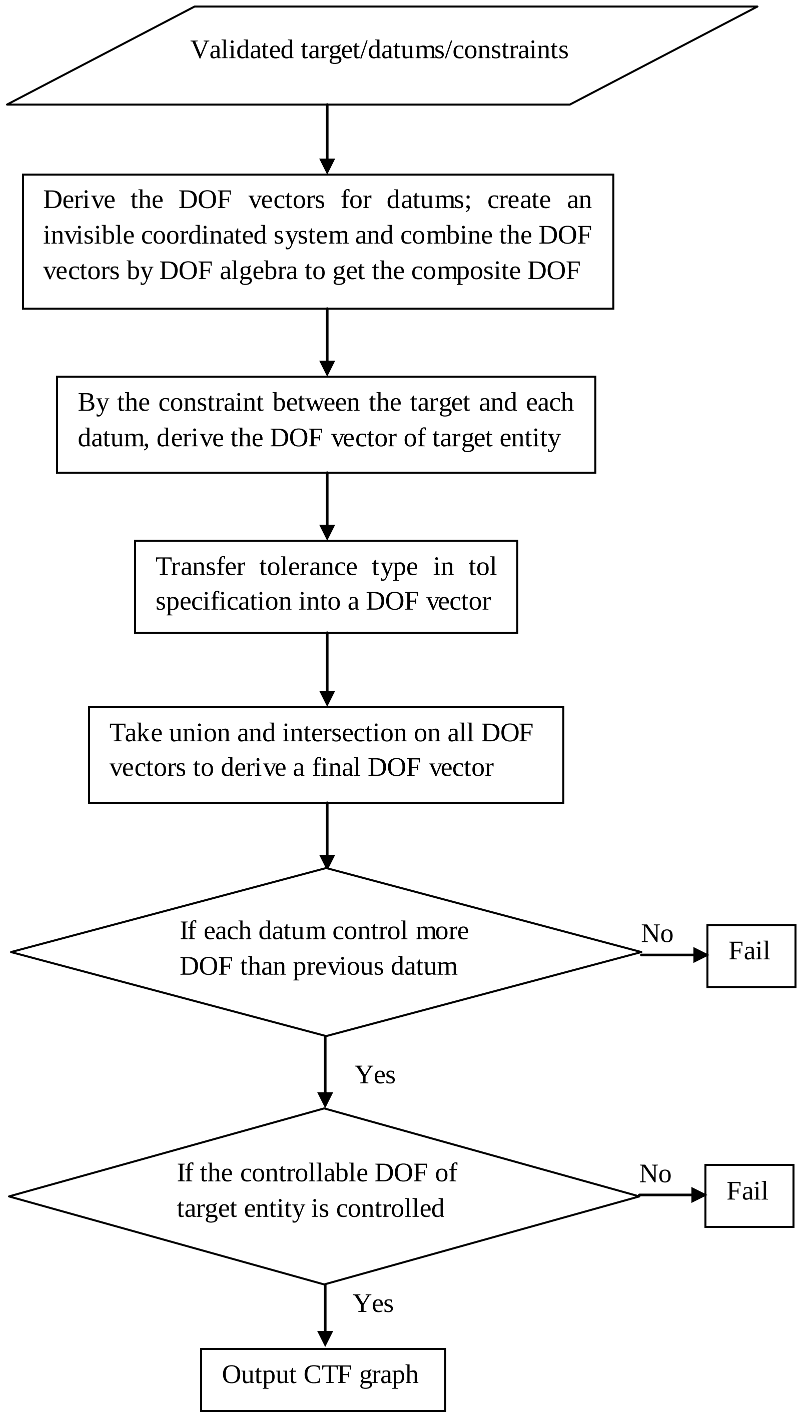


Figure 6.12 Procedure of DOF analyses

6.6 Tolerance scheme advisor

After the accuracy of the GD&T scheme is guaranteed, a further step should be taken to develop a better and more rigorous GD&T scheme to assure the manufacturability of parts and to reduce the manufacturing and inspection cost. Therefore, tolerance scheme advisor is implemented to assist designers to perfect tolerance specification and balance the tradeoffs between the cost and functionality. A good tolerance specification is supposed to take good GD&T practice rules into account. The former work in DAL at Arizona State University [Ramaswamy, Shah, Davidson 2001] studied the good practice rules in GD&T and implemented them. These practice rules can be classified into two categories, validation rules and recommendation rules. The validation rules have been carried out through the GD&T representation model. The recommendation rules are independently fulfilled in this GD&T scheme and designed under designer's permission in this module. Whenever a violation is detected, the system will provide an alternative tolerance specification consistent with the good practice rules. The recommendation rules are listed following based on the work of [Ramaswamy, Shah, Davidson 2001].

(1) The shape of a datum needs to be controlled by a form tolerance. Since each valid tolerance specification is recorded in the tolerance linked list, it is easy to check the form tolerance specification of the datum. If the form tolerance specification does not exist, the tolerance specification would be created, in which the datum is the target entity.

(2) While choosing multiple datums, it is advisable to control the orientation of the datums with respect to each other, since the datums establish the coordinate frame composed of three mutually perpendicular planes. The following controls are suggested [Meadows 1998]: an orientation control for the secondary datum with primary datum as reference and an orientation control for the tertiary datum with the primary and/or secondary datum(s) as reference. This rule is implemented in the metric relation verification module. The module not only checks and creates the constraints between target and datums; but also check and create the constraints between each datum in the DRF. These constraints between each datum would also be added into constraint linked list.

(3) Large measuring uncertainties are associated with smaller datums. Such smaller datums must be avoided where possible. Swapping the datum and target may be considered when such a situation is encountered. This is done in the target and datum entity validation step. The geometric characteristics of target entity and datum entities can be derived by ACIS functions. We need to compare the areas of target entity and datum entity. If one is significantly larger than another one, the one with larger area should be chosen as the datum. Consider a perpendicularity tolerance requirement between two faces, one significantly larger than the other; for instance, 20 x 5 and 5 x 5 (Figure 6-1). In such a situation, it is better to tolerance the face ‘A’, with ‘B’ as the datum, rather than vice versa. If ‘A’ were to be chosen as the datum, any slight surface variation in ‘A’, would cause a large angular deviation of ‘B’ from the theoretical datum plane formed by

‘A’. This also stresses on the need for specifying a flatness tolerance for such a datum. Figure 6.14 shows the datum selection depending on size.

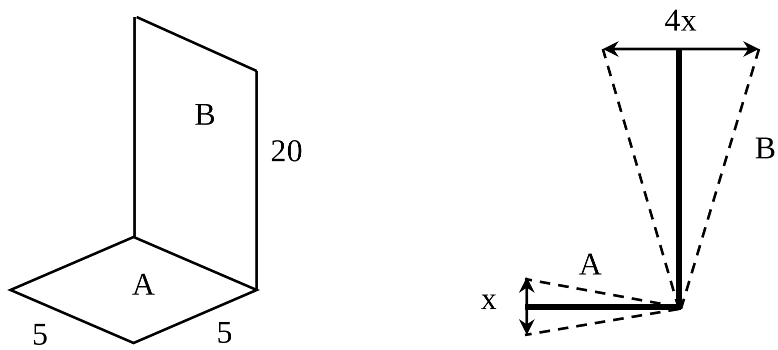


Figure 6.14 Datum selection depending on size

(4) Assigning tight tolerances to holes with a large length to diameter ratio (greater than 4) and small diameter should be avoided since the inspection of such holes is very difficult and expensive.

(5) For a runout tolerance, if it is relatively larger in diameter and relatively short in length, use a planar face as primary datum. The way to implement the rule (4) and (5) is basically same. Both of the rules work when the tolerance specification is placed on conical faces. The diameter and length of cylinders need to be extracted from solid model. If the ratio between two parameters is greater than 4, the error is activated and the tolerance specification is terminated.

CHAPTER 7 CASE STUDIES

This chapter introduces a number of cases to get users and readers familiar with the main functions that can be achieved by the modules in the integrated GD&T system. These functions are placing a tolerance specification, metric relation checking, diagnosing tolerances against the good practice rules and DOF analysis. The functions are operated by the tolerance specification module, the verification module, tolerance advisor module and DOF analysis module in the integrated GD&T system.

7.1 Placing a tolerance specification

The users pick tolerance elements through tolerance specification dialog. The tolerance specification dialog is designed as a class which has target entities, datum entities, tolerance class, tolerance value and modification condition as the members. After the geometry is imported into the system, target entities and datum entities are picked in display window. Once they are picked, the address of them are delivered back to dialog and stored in an instance of the dialog class. The edit box tells you whether the entity is selected and stored or not. The picked entities can be deleted by click on the individual "Delete" button in the dialog box. Figure 7.1 shows the process of picking geometric entity. Next, the tolerance class, tolerance value and material condition should be picked by users. Figure 7.2 shows the inputting of tolerance class and value. If the tolerance specification goes through all verification modules, the entities involved in the tolerance specification would be listed and labeled with a number. This number is an attribute of entity. The number can be used to identify the entity in subsequent tolerance specifications. Figure 7.3 shows the display of entity geometric information, such as shape, label number, root point and etc.

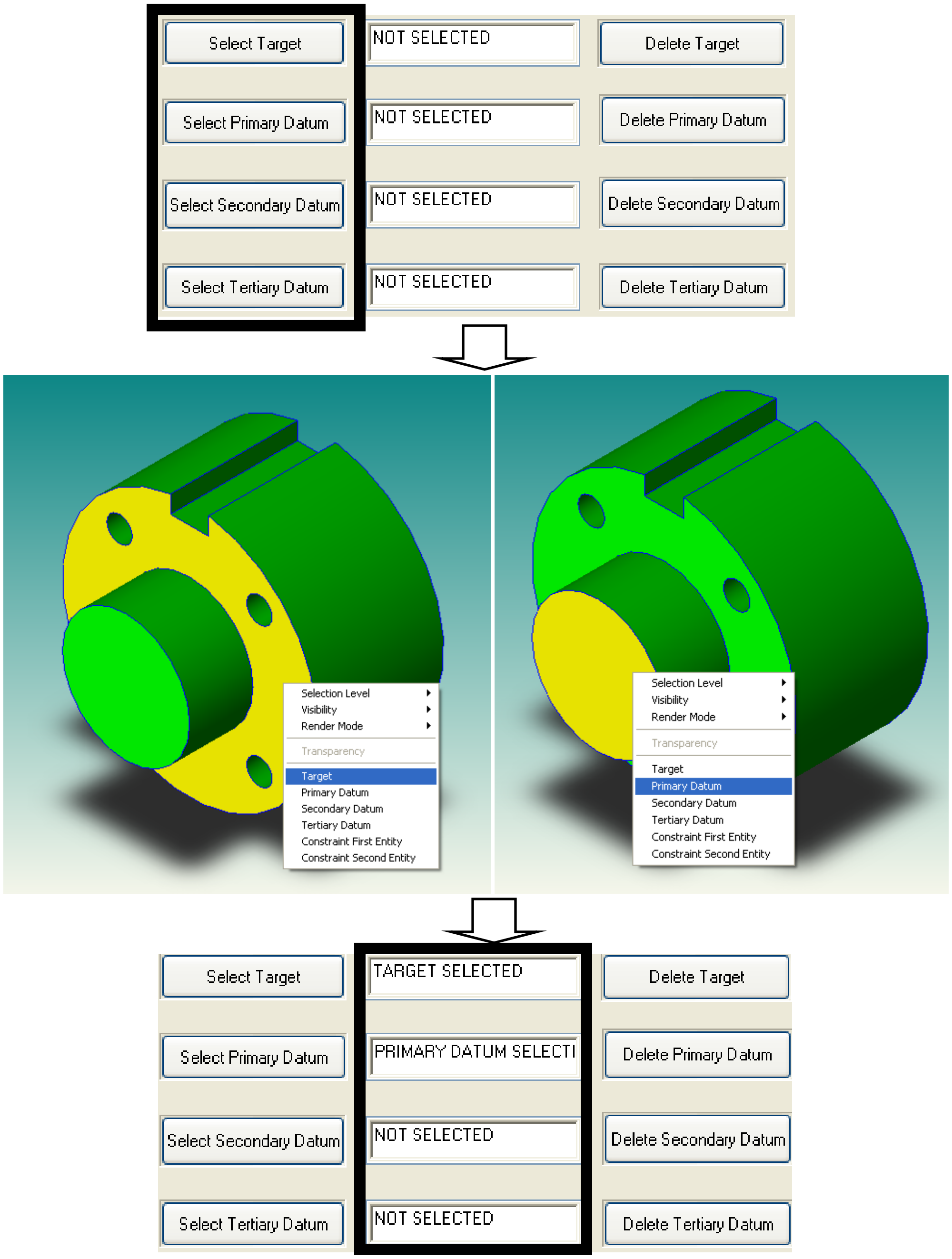


Figure 7.1 Process of picking entities

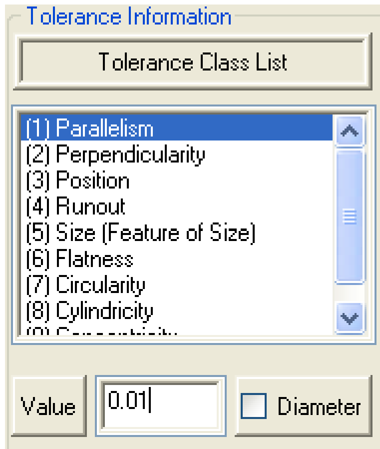


Figure 7.2 Inputting of tolerance class and value

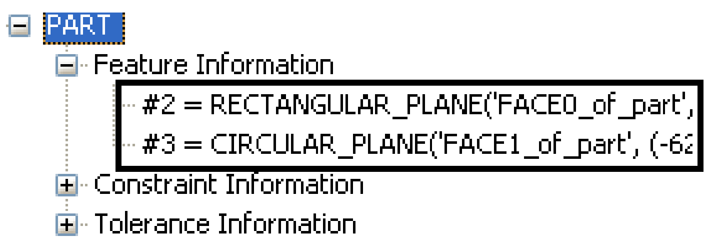


Figure 7.3 Display of entity geometric information

7.2 Verification module

When the user specifies a tolerance, the verification module in the system performs a set of validations in both the pre-checking stage and the instantiating stage (see Section 6.3). These validations include checking the entity type based on tolerance class, checking the metric relation involved in the tolerance specification, finding out the location constraint in tolerance specification, and so on. These validations help to maintain the self-validation of a GD&T representation model.

The entity feature type should be consistent with tolerance class. The verification module checks if the selection of either target entity or datum entity is allowed by tolerance class. Some tolerance classes only allow the cylinder face to be the target. Size and shape tolerance never need a DRF in tolerance specification. In Figure 7.4, a planar face is picked for a cylindricity tolerance, which is a failure reported by the system.

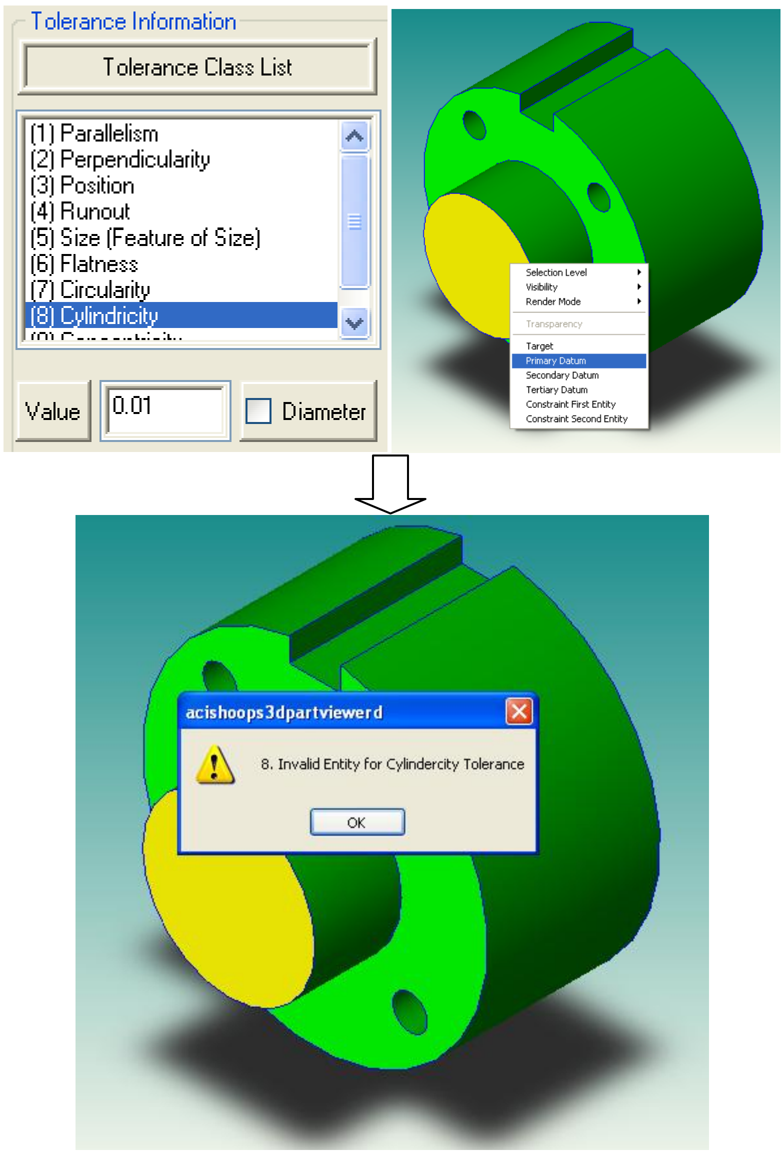


Figure 7.4 Invalid entity pick for cylindricity tolerance

In Figure 7.5, a primary datum is picked in a flatness tolerance. The system reports a failure message instantly.

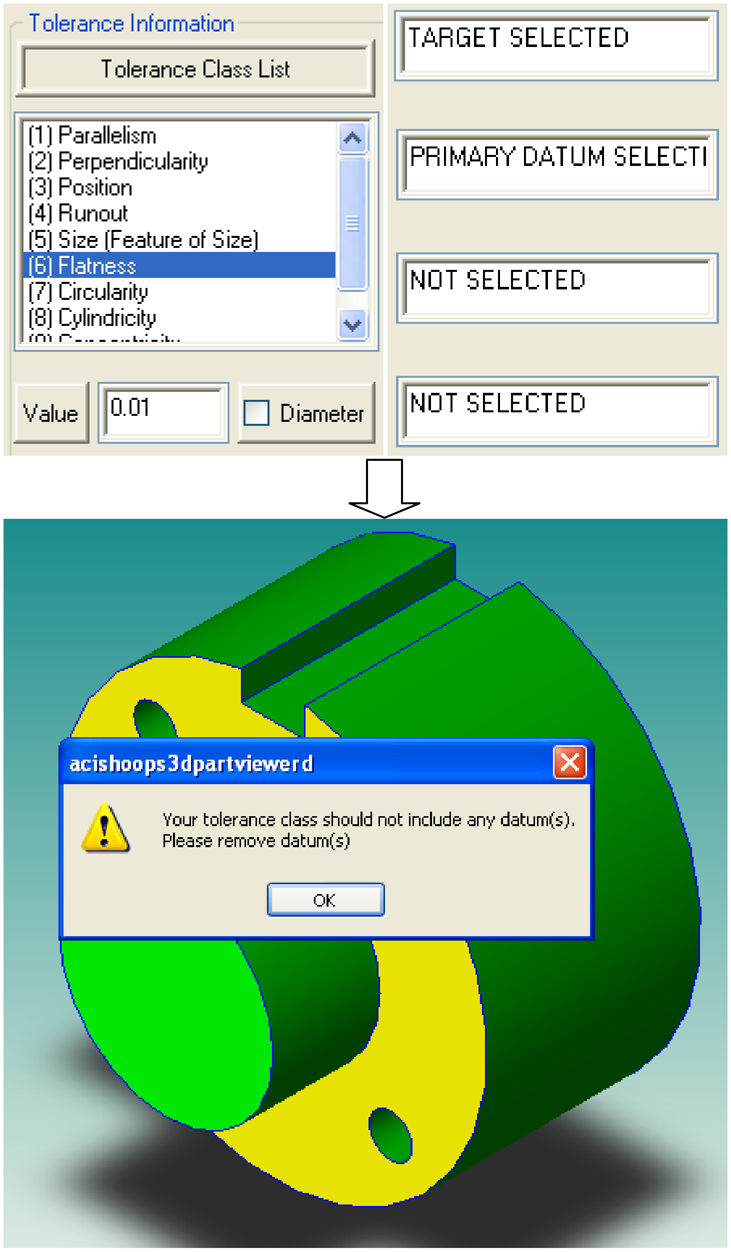


Figure 7.5 Invalid pick of datum entity in flatness tolerance

Metric relations need to be checked before a tolerance specification is developed. If the required metric relation is not consistent with the pick geometry entities, the tolerance specification aborts and the system informs the user the reason for the failure. In Figure 7.6, the pair of highlighted faces is parallel to each other. To specify perpendicularity tolerance to any one of these two faces with respect to the other causes a failure.

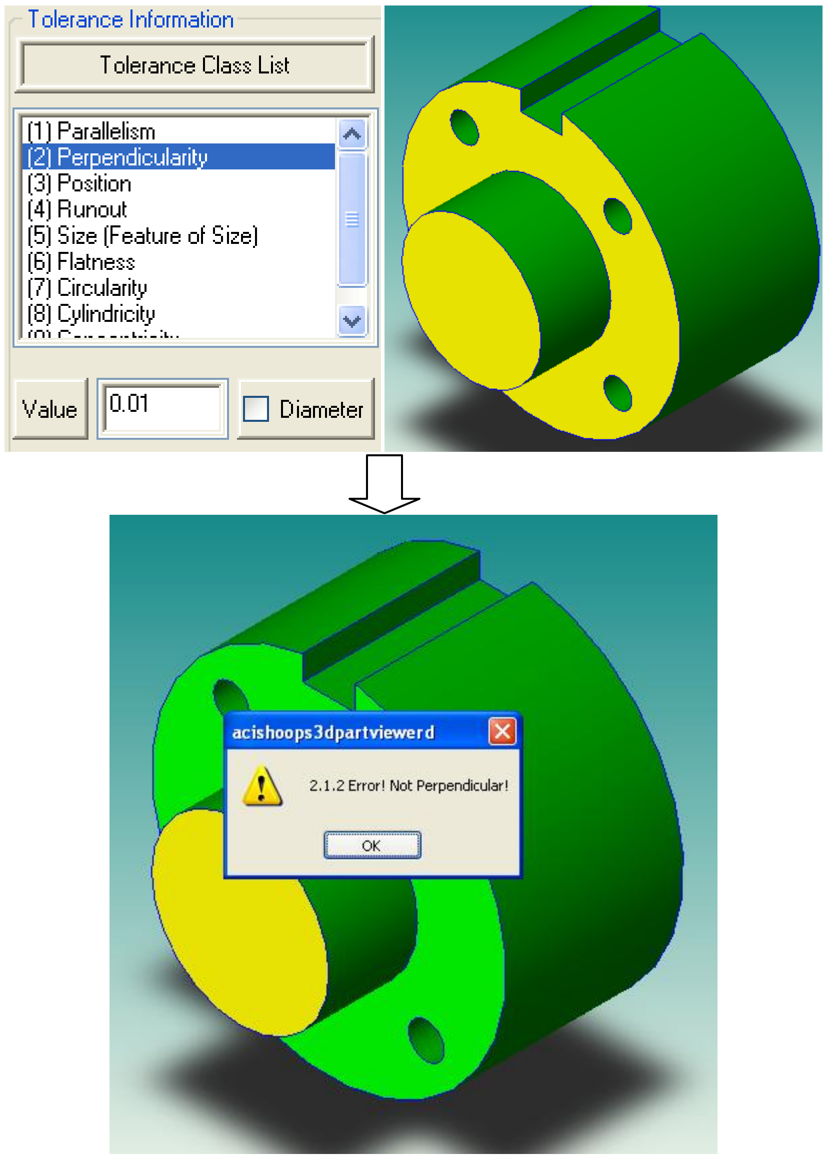


Figure 7.6 Invalid metric relation for perpendicularity tolerance

Material condition needs to be checked before the instantiation of tolerance specification. According to ASME standard Y14.5, material condition modifiers are not used with form tolerance class except straightness tolerance. This validation rule should be carried out in the verification module as well. In Figure 7.7, the system reports an error because the material condition modifier for target entity is selected in a form tolerance.

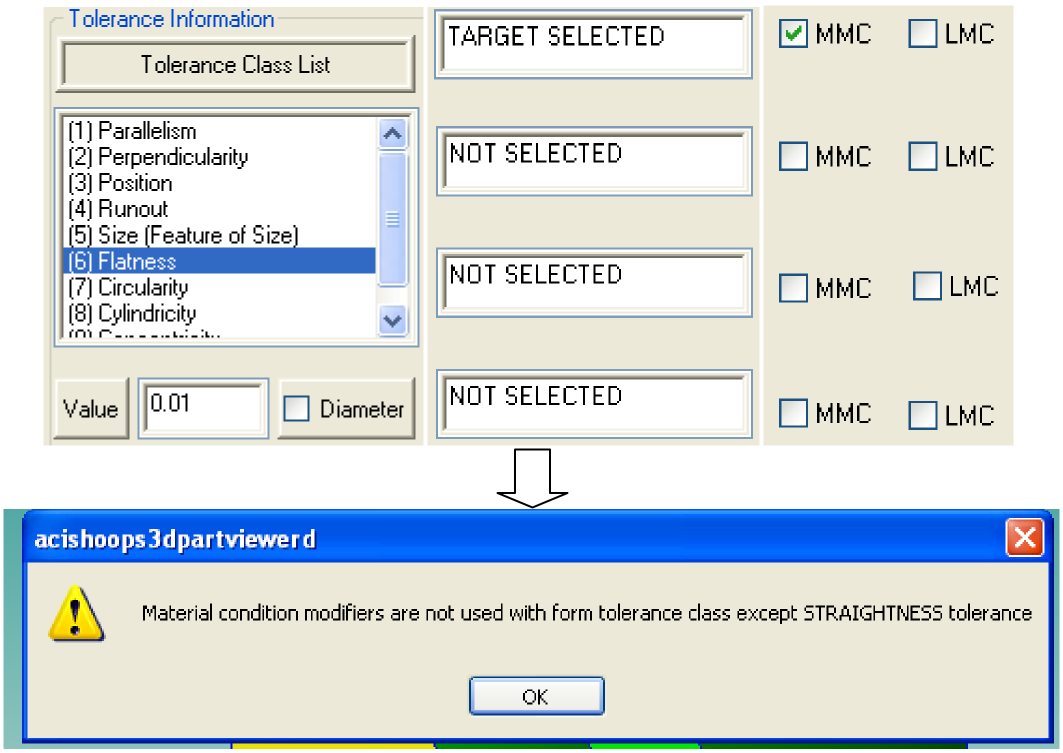


Figure 7.7 A failure caused by material condition modifier

7.3 DOF Checking

The purpose of development of DOF algebra is to validate the tolerance specification by the mathematical calculation of DOFs of tolerance elements. In Chapter 4 and 5, a set of control rules regarding DOF algebra and DOF validation are discussed. In Figure 7.8, an example of DOF validation is shown. The DOF analysis module is designed to check the capability of controlling additional DOF of each datum member of DRF in precedence. The user wants to specify a perpendicularity tolerance on cylinder FACE0. The primary datum (Datum A) is the top face FACE1. The secondary datum (Datum B) is the bottom face FACE2 where the FACE0 sits. Since the FACE1 (datum A) is parallel to the bottom face FACE2, it cannot control additional DOF other than those controlled by the previous datum in the DRF. This datum combination specified is not a valid DRF. The reason for the failure is displayed by the system.

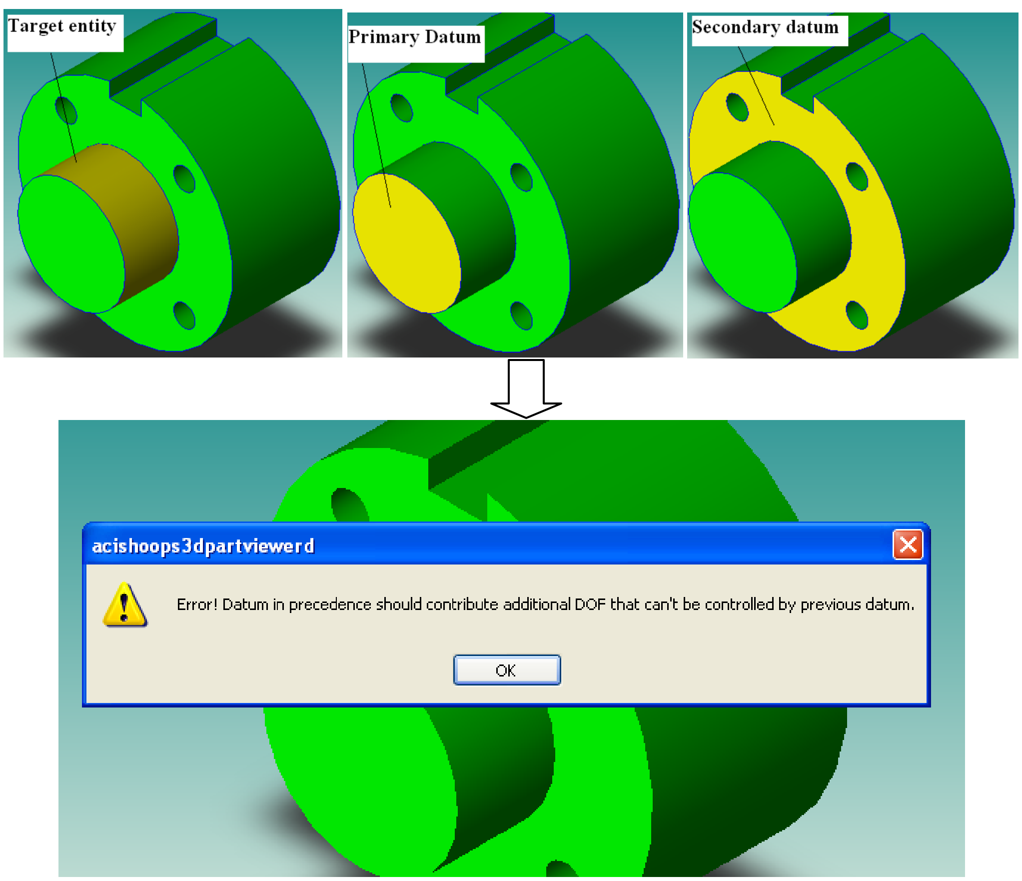


Figure 7.8 An example of checking the capability of controlling

Figure 7.9 shows an example of under-toleranced DOFs of target entities in a positional tolerance. The user wants to specify a positional tolerance on the cylinder FACE0 in the previous case. The primary datum (Datum A) is the bottom face FACE2 where the FACE0 sits. The secondary datum (Datum B) is the center plane of the slot. The datum A and B form a DRF for the target FACE0. The DOF vector of the datum A is [001,110]. The DOF vector of the datum B is [010,101]. By DOF algebra, the DOF vector of the DRF is the Boolean union of DOF vector of the datum A and B. It is [011,111]. The DOF vector of the target in DRF is [110,110]. The DOF vector of the positional tolerance is [111,111]. The target entity is under-toleranced because the DOF vector of the DRF misses a constraint on T_x. The system reports an under-toleranced warning to user.

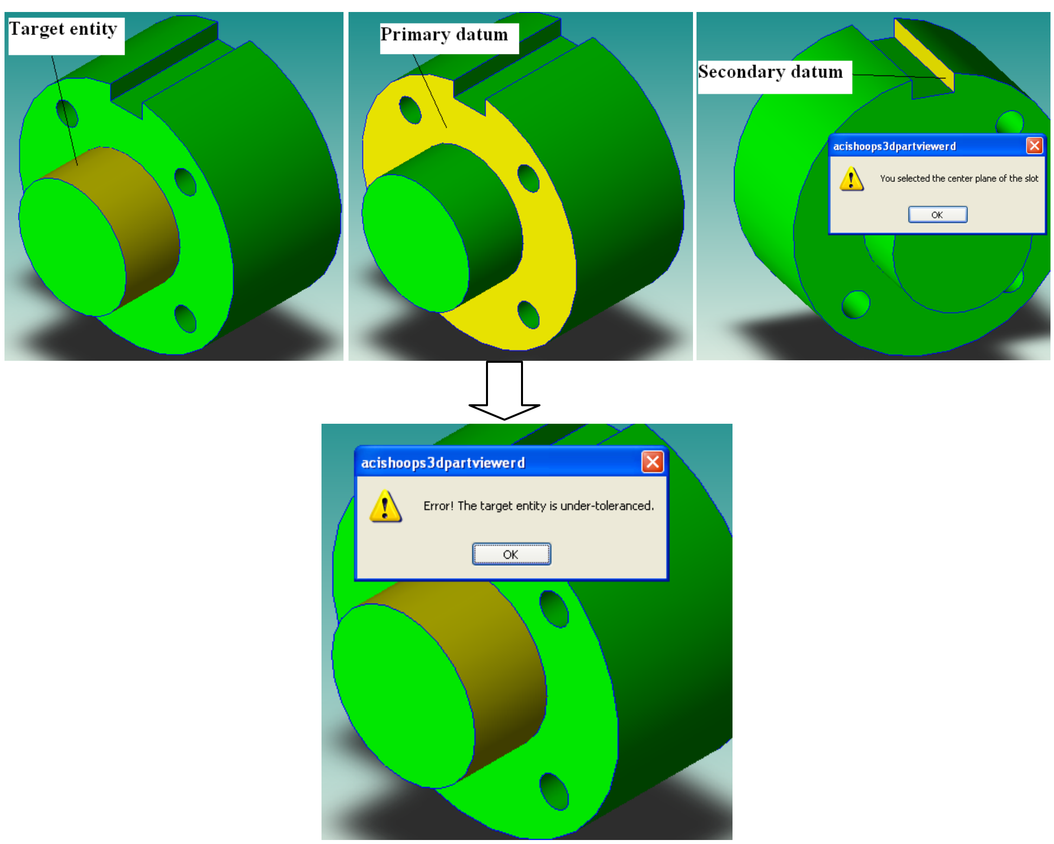


Figure 7.9 An example of under-toleranced DOF

7.4 Evaluating tolerances against good practice rules

The tolerance advisor in the integrated GD&T system is able to check the tolerance against good practice rules, it also provides the alternative ways to modify the tolerance so that the tolerance could be consistent with the good practice rules. These good practice rules have been discussed in Section 5.2. In Figure 7.10, the bottom face FACE2 has a perpendicularity tolerance with respect to the slot base face. Now the tolerance scheme advisor checks the tolerance against the good practice rules. The advisor finds that the datum is much smaller than the target. It provides an alternative solution of swapping the target and the datum. If the user accepts this suggestion, the advisor then finds that there is no form tolerance controlled on the new datum (the bottom face), the system asks the user if a form control can be added. If the user agrees, a flatness tolerance is added on the new datum.

In Figure 7.11, the cylinder face has a runout tolerance with respect to the bottom face. The ration of diameter and height is greater than 4. Now the tolerance scheme advisor checks the tolerance against the good rules. The advisor gives a suggestion to loose the tolerance a bit when it finds the length to diameter ratio of this feature is considerably large.

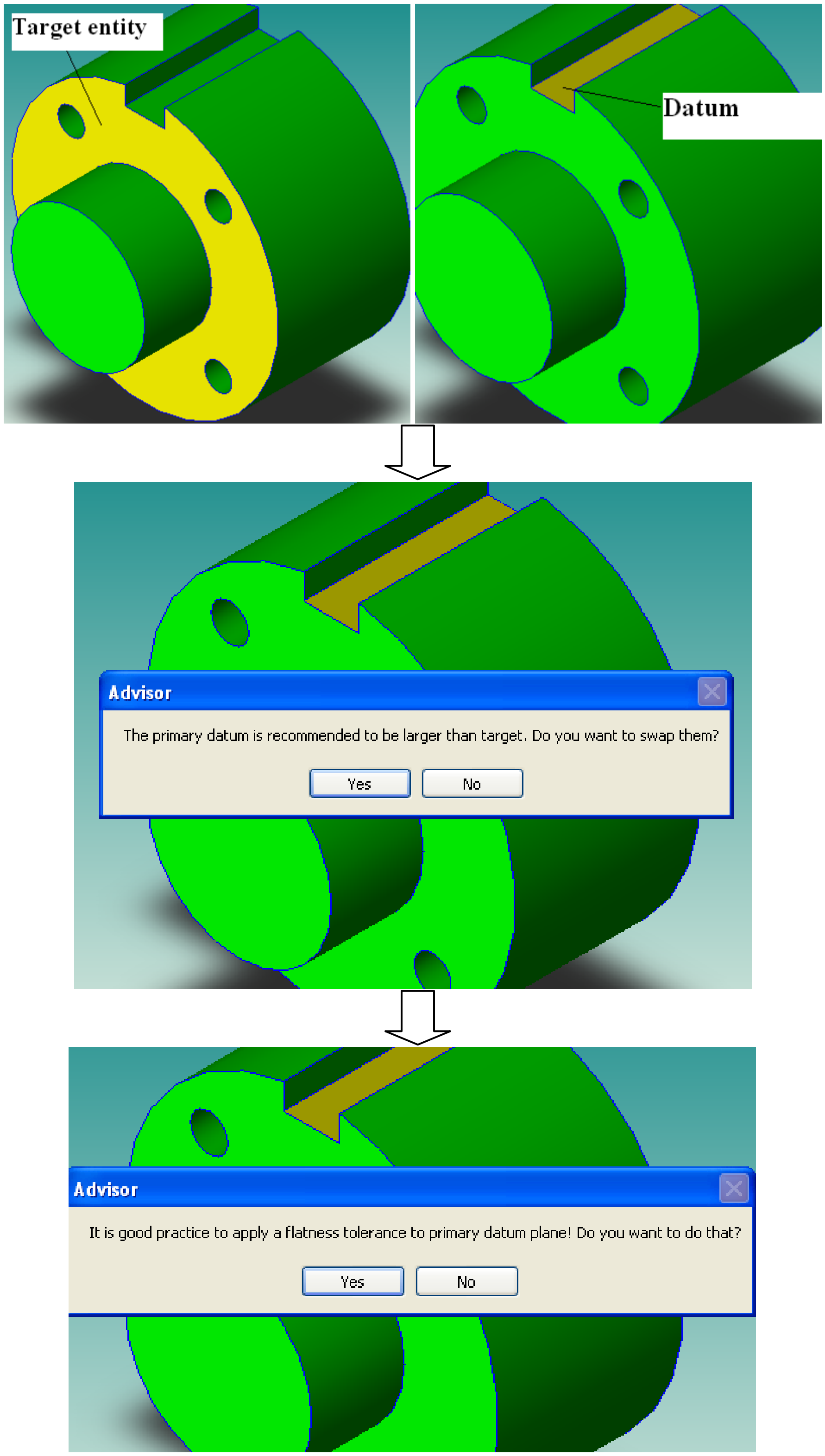


Figure 7.10 Advisor suggests swapping the target and datum

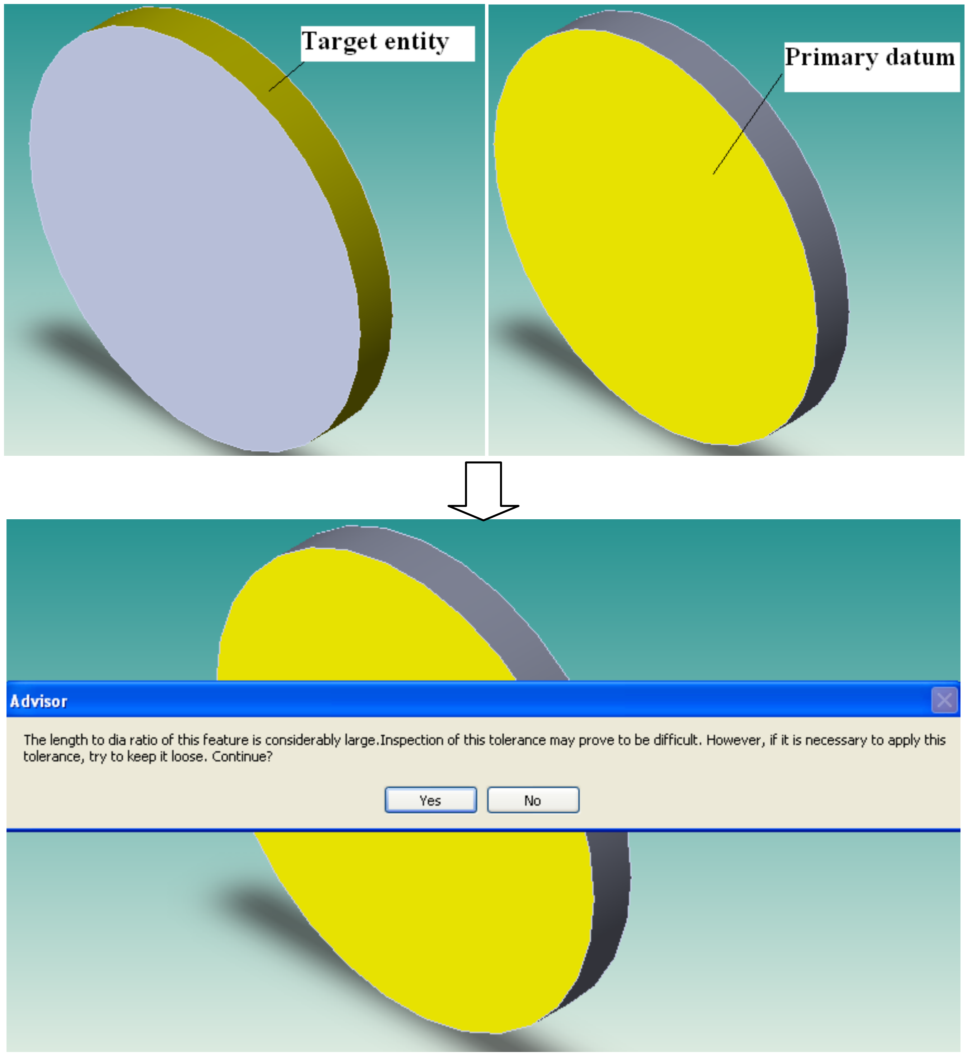


Figure 7.11 Advisor suggests to loose the tolerance value for a flat cylinder

7.5 CTF display

CTF graph contains all the information about a part i.e. What type of features it have , what type of entities, the constraints on features, tolerance information and the degrees of freedom. It is actually a textual representation of all the things mentioned above. In the integrated GD&T system, the CTF graph is displayed as a list. The top face of small cylinder has perpendicularity tolerance with respect to the bottom face where the small cylinder sits. There is no form tolerance on both of faces. Figure 7.12 shows the CTF graph list for a parallelism tolerance specification. The feature information includes the face type, face number, root point, and other geometric characteristics. The constraint information includes the constraint type involved in the tolerance specification, constraint value and distance. The tolerance information includes the tolerance class, the tolerance value, DRF and DOF information.

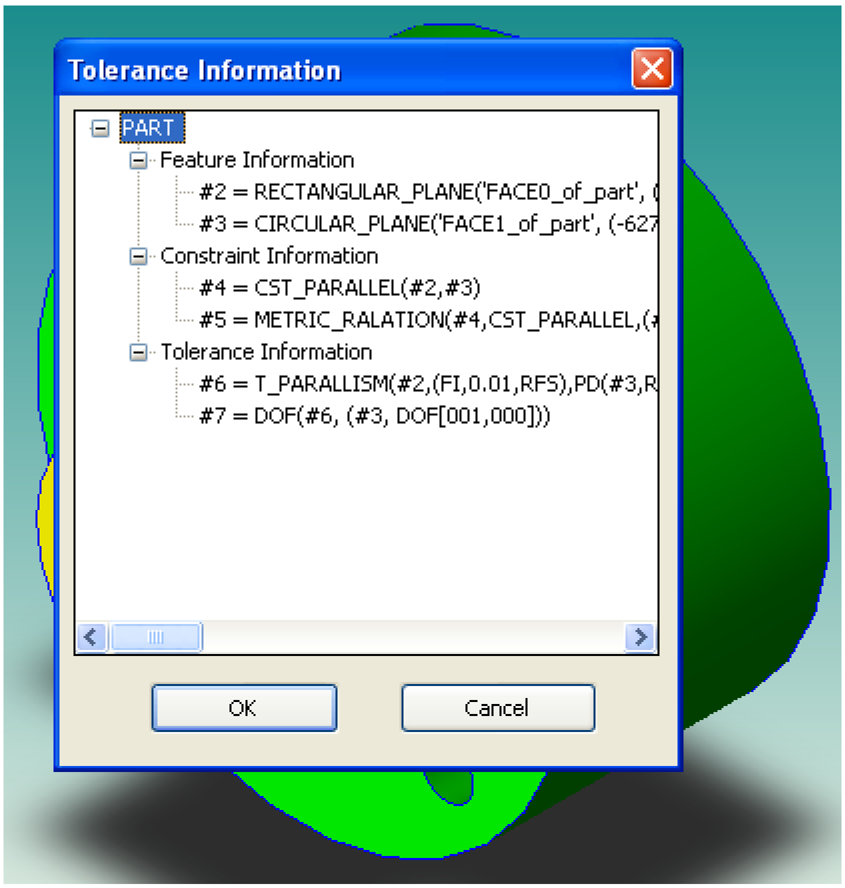


Figure 7.12 The CTF graph list for a parallelism tolerance specification

CHAPTER 8 CONCLUSION AND FUTURE WORK

This thesis develops a computational tool package for a GD&T mathematical model. This package is able to support basic GD&T tasks: tolerance specification, mathematical representation, and various validations. The contributions of this research and the future work are discussed in following sections.

8.1 Original contributions

DOF algebra is developed in a formal mechanism for modeling and validating tolerancing schemes in CAD systems. The mechanism is able to store the tolerancing specification including datum reference frame and tolerance class as attributes to the corresponding target entities like face, edge, and vertices. The introduction of Boolean operation in DOF algebra enforces the tolerance specification into a pure mathematical calculation. This addition makes a computer implementation of specification validation more realizable and reliable.

Rigid entities are simulated by the variation space of ideal primitive features (line or plane). All information of active and invariant DOFs is stored as a DOF binary vector. DOF algebra displays what DOFs of DRF and the target have and what DOFs of target entity are constrained. This helps to validate a DRF efficiently. The work of the determination of DOF within a tolerance specification using the DOF algebra method consists of three steps. The first step is to set up a coordinate system (CS) based on a given DRF cluster. The next step is to determine the composite DOFs of combined entities in a DRF cluster and that of the combined target entities. The third step is to track the controlled DOFs of a target entity by its DRF and tolerance class in a tolerance specification. The tolerance class has its respective aimed DOFs. DOF representation implies the metric relations between target entities and DRF.

Validity is a weighty effect in a mathematical GD&T model. To accomplish the validity requirement, the relations among tolerance specification elements in GD&T should be fully studied. This research addressed a fundamental and applicable model needed in validation. The existence of a corresponding DRF is the prerequisite for the existence of a tolerance type. A tolerance class is only allowed to be associated with certain types of target entity. The DOF algebra derives a set of control rules into mathematical operation. DOF algebra is able to come to the judgment by comparing the constrained DOFs and targets DOFs or checking the DRF DOFs.

In the implementation of the GD&T DOF representation model, a user-friendly tool package incorporates several self-validated and automated functional modules. The tolerance specification module takes care of all inputted tolerance components needed for a tolerance specification. It is an interface for users to select or specify necessary components and quantity in a tolerance specification. The metric condition verification module is responsible for verifying that the metric constraint suggested by target entity and DRF cluster is consistent with the tolerance class. The DOF verification module fulfills the DOF algebra into implementation. It transforms the tolerance elements into individual DOF binary vectors and performs logical operation on these vectors. It will finally produce a validation judgment based on the calculation. A CTF graph is constructed as a tree-viewer to supply the information of GD&T specification. A tolerance advisor scheme is developed to offer suggestion on improving the tolerance specification in terms of useful practice rules.

8.2 Future works

Further studies of good practice rules will be needed. Such studies require the corporation of the engineering industry to accumulate a complete list of good dimensioning and tolerancing rules. The GD&T practice rules used for evaluating toleracing in this work have not been systematically well organized. In addition, the criteria for judging a good practice rule need to be studied depending on various products, functionality, and manufacturing procedure.

So far the research on tolerance specification has been confined to validating a tolerance specification. Further research needs to be conducted in order to apply DOF and CTF graphs to the tolerance statistical analysis and allocation.

REFERENCES

ASME Standard, 2009, “Dimensioning and Tolerancing”, ASME Y14.5M-2009, American Society of Mechanical Engineers, New York

ASME Standard, 1994b, “Mathematical definition of Dimensioning and Tolerancing Principles”, ASME Y14.5.1-1994b, American Society of Mechanical Engineers, New York

Ameta, Davidson, Shah.J., 2007, “Tolerance-Maps Applied to a Point-Line Cluster of Features”, Journal of Mechanical Design, August 2007, V129, pp782-792

Bettig, B.P. and Shah, J.S., 2001, “Deerivation Of A Standard Set Of Geometric Constraints For Parametric Modeling and Data Exchange”, Computer-Aided Design, Volume 33, Issue 1, January 2001, Pages 17-33

Bettig, B.P., 1999, “A Graph-Based Geometric Problem Solving System For Mechanical Design And Manufacturing”, Ph.D. thesis, Department of Mechanical & Aerospace Engineering, Arizona State University, Tempe, Arizona

Bhide, S. S., 2002, “A new mathematical model for geometric tolerances applied to cylindrical features”, M.S. thesis, Department of Mechanical & Aerospace Engineering, Arizona State University, Tempe, Arizona

Coxeter H.S., 1969, “Introduction to Geometry”, 2nd Ed., Wiley

Davidson J. and Shah J., 1998, “Project Description: Mathematical Modeling of Geometric Variations to Integrate Parametric CAD with Tolerance Analysis and Optimization”, NSF proposal, Arizona State University, 1998

Davidson J. and Shah J., 2001B, “Geometric Tolerances: A New Application for line Geometry and Screws”, IMechE J. of Mechanical Eng. Sci., Cambridge University, Dec. 2001

Davidson J. and Shah J., 2003, “Using Tolerance-Maps to represent material condition on both a feature and a datum”, Proceedings on the 8th CIRP International Seminar on Computer Aided Tolerancing-Managing Geometric Uncertainty in the Product Lifycycle, April 28-29, Charlotte, North Carolina, USA

Davidson J., Mujezinovic A. and Shah J., 2000, “A New Mathematical Model for Geometric Tolerances as Applied to Round Faces”, ASME Proceedings, DETC00/DAC-14249, September 10-13, 2000, Baltimore, Maryland, USA

Gao, J., Chase, K.W. and Magleby, S.P., 1998, "Generalized 3-D Tolerance Analysis of Mechanical Assemblies with Small Kinematic Adjustments", IIE Transactions, v30(4), 98, p367-377

Giordano, M., Pairel, E., and Samper, S., 1999, "Mathematical Representation of Tolerance Zones", In Global Consistency of Tolerances, Proc., 6th CIRP Int'l Seminar on Computer-Aided Tolerancing, Univ. of Twente, Enschede, Netherlands, March 22-24, F. van Houten and H. Kals, eds., p177-186

Kandikjan T. and Shah J., 1998, "A Computational Model for Geometric Dimensions and Tolerances Consistent with Engineering Practice", Proceeding of DETC98: 1998 ASME Design Engineering Technical Conference, September 13-16, 1998, Atlanta, GA

Kandikjan T., Shah J. and Davidson J., 2001, "A mechanism for validating dimensioning and tolerancing schemes in CAD system", Computer-Aided Design Vol 33(2001), pp721-737

Kramer, G. A., 1992, Solving Geometric Constraint Systems: A Case Study in Kinematics, MIT Press

Liu, Y.S. and Gao, S. M., 2004, "Variational Geometry Based Pre-Inspection of A Pattern of Holes", Intl J. of Production Research, V42(8), pp1659-1675

Moroni, G. and Polini, W., 2003, "Tolerance-based Variations in Solid Modeling", Journal of Computing and Information Science in Engineering December 2003, Vol.3, pp345-352

Mujezinovic A., 1999, "A new mathematical model for representing geometric tolerances", M.S. thesis, Department of Mechanical & Aerospace Engineering, Arizona State University, Tempe, Arizona

Mujezinovic A., Davidson J. and Shah J., 2000, "A New Mathematical Model for Geometric Tolerances as Applied to Polygonal Faces", ASME Transactions, J. of Mechanical Design, v126(3), p504-518

Prisco, U. and Giorleo, G., 2002, "Overview of Current CAT systems", J.of Integrated Computer-Aided Eng., n9, pp373-387

Ramaswamy S., 2000, "A Design Support System for Tolerance specification and Verification", Source: MS Thesis, Arizona State University, Tempe, Arizona

Ramaswamy S., Shah J.J. and Davidson J., 2001, "Computer aided GD&T advisor based on Y14.5 conformance and good practice", ASME Design for Manufacturing conference, Sep. 2001, Pittsburgh, USA

Roy, U., and Li, B., 1998, “Representation and Interpretation of Geometric Tolerances for Polyhedral Objects: I. Form Tolerances,” Computer-Aided Design, v31, pp273-285

Roy U. and Liu C.R., 1993, “Integrated CAD frameworks: Tolerance Representation Scheme in A Solid Model”, Computers & Industrial Engineering, 1993, Vol.24, No.3, pp495-509

Shah J., Yan Y. and Zhang B-C, 1998, “Dimension and tolerance modeling and transformations in feature based design and manufacturing”, Journal of Intelligent Manufacturing, V9(5), pp475-488, 1998

Shah J, Rogers M, Bhatnagar A, Hsiao D. “The ASU feature testbed – an overview”, Proceedings of ASME Computers in Engineering Conference, 1990.

Shen. Z., 2003, “Software Review-Tolerance analysis with EDS/VisVSA”, ASME Transactions on the Journal of Computing and Information Science in Engineering, special issue on GD&T, volume 3, number 1, Mar. 2003, pp95-99

Shen, Z., Ameta, G., Shah, J.J. and Davidson, J.K., 2004, “A Comparative Study Of Tolerance Analysis Methods”, CD-ROM Proc. of ASME DETC’04, 24^{th} CIE Conference, Paper Number: DETC2004/CIE-57699, Sept. 28-Oct.2, 2004, Salt Lake City, Utah, USA

Shen, Z., 2005, “Development of A Framework for A Set of Computer-Aided Tools for Tolerance Analysis”, Ph.D. thesis, Department of Mechanical & Aerospace Engineering, Arizona State University, Tempe, Arizona

Shen Y, Shah J, J.K.Davidson, 2010, “DOF Algebra for Entity Cluster in GD&T”, Techanical Report, August 2010, Arizona State University, Tempe, AZ 85287

Shen, Y., Shah, J.J. and Davidson, J.K., 2011, “Feature Cluster Algebra in Geometric Dimensioning and Tolerancing”, CD-ROM Proceeding of ASME DETC’2011, Paper Number: DETC2011/CIE47937, Design Engineering Techanical Conference and Computers & Information in Engineering Conference, August 29-31, 2011, Washington D.C.

Wade O.R., 1967, “Tolerance Control in Design and Manufacturing”, Industrial Press, New York, 1967

Wu Y., 2002, “Deveolopment of Mathematical Tools for Modeling Geometric Dimensioning and Tolerancing”, Ph.D. thesis, Department of Mechanical & Aerospace Engineering, Arizona State University, Tempe, Arizona

Wu, Y., Shah J. and Davidson J., 2003A, "Computer Modeling of Geometric Variations in Mechanical Parts and Assemblies", ASME Transactions on the Journal of Computing and Information Science in Engineering, special issue on GD&T, volume 3, number 1, Mar. 2003, pp54-63

Yan, Y., 1995, "Dimensions and Tolerances Mapping", MS Thesis, Arizona State University, Tempe, Arizona

Zhang, B.C., 1992, "Geometric Modeling of Dimensioning and Tolerancing", Ph.D., thesis, Arizona State University, Tempe, Arizona