Automating GD&T Schema for Mechanical Assemblies

by

Sayed Mohammad Hejazi

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

Approved July 2016 by the
Graduate Supervisory Committee:

Jami Shah, Chair
Joseph Davidson
Dianne Hansford

ARIZONA STATE UNIVERSITY

August 2016

ABSTRACT

Parts are always manufactured with deviations from their nominal geometry due to many reasons such as inherent inaccuracies in the machine tools and environmental conditions. It is a designer job to devise a proper tolerance scheme to allow reasonable freedom to a manufacturer for imperfections without compromising performance. It takes years of experience and strong practical knowledge of the device function, manufacturing process and GD&T standards for a designer to create a good tolerance scheme. There is almost no theoretical resource to help designers in GD&T synthesis. As a result, designers often create inconsistent and incomplete tolerance schemes that lead to high assembly scrap rates. Auto-Tolerancing project was started in the Design Automation Lab (DAL) to investigate the degree to which tolerance synthesis can be automated. Tolerance synthesis includes tolerance schema generation (sans tolerance values) and tolerance value allocation. This thesis aims to address the tolerance schema generation. To develop an automated tolerance schema synthesis toolset, to-be-toleranced features need to be identified, required tolerance types should be determined, a scheme for computer representation of the GD&T information need to be developed, sequence of control should be identified, and a procedure for creating datum reference frames (DRFs) should be developed. The first three steps define the architecture of the tolerance schema generation module while the last two steps setup a base to create a proper tolerance scheme with the help of GD&T good practice rules obtained from experts. The GD&T scheme recommended by this module is used by the tolerance value allocation/analysis module to complete the process of automated tolerance synthesis. Various test cases are studied to verify the suitability of this module. The results show that software-generated schemas are

proper enough to address the assemblability issues (first order tolerancing). Since this novel technology is at its initial stage of development, performing further researches and case studies will definitely help to improve the software for making more comprehensive tolerance schemas that cover design intent (second order tolerancing) and cost optimization (third order tolerancing).

ACKNOWLEDGMENTS

I would like to thank my advisor Dr. Jami Shah, for his valuable guidance and support throughout this work. All of the things I learned would not have been possible without his supervision. I would also like to express sincere gratitude to Dr. Davidson and Dr. Hansford for serving on my committee.

I am deeply thankful to all my colleagues at DAL especially my project team mates for their friendship, support and guidance.

Finally, I would like to acknowledge the financial support from DMDII (Digital Manufacturing and Design Innovation Institute), Grant No. 14-0-05.

TABLE OF CONTENTS

Page

Page

Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

ABBREVIATIONS, NOMENCLATURE AND SYMBOLS

<u>1-D:</u> One Dimensional, alternatively written as 1D.

<u>2-D:</u> Two Dimensional, alternatively written as 2D.

<u>3-D</u>: Three Dimensional, alternatively written as 3D.

<u>Assemblability</u>: Term to describe that manufactured component parts successfully fit together.

<u>Assembly Feature</u>: Mating feature pairs on distinct parts in an assembly.

<u>Auto-Tolerancing</u>: Software to automate tolerance synthesis.

<u>Basic Dimension</u>: Nominal dimensions defined to identify the controlled dimensions and help locate the tolerance zones.

<u>Datum</u>: A theoretically exact point, axis, or plane derived from the true geometric counterpart of a specified datum feature. A datum is the origin from which the location or geometric characteristics of feature of a part are established.

<u>Datum Feature</u>: An actual feature of a part that is used to establish a datum.

<u>Datum Reference Frame (DRF)</u>: Coordinate systems used to locate and orient a part feature. All measurements for the features, which have geometric tolerances related to the datums will originates from the established datum reference frame, and not a part. Tolerance zones are located or oriented to the datum reference frame and no the datum features. Geometric tolerances related to a DRF are tolerances of location, orientation run-out and sometimes profile. Datum reference frame usually consists of one to three entities; it also could be established by pattern of holes. Theoretically, secondary and tertiary datums should be perpendicular or align with the primary datum. (1) when actual feature of secondary datum is not aligned with the primary datum as it is supposed to, secondary

datum needs to be translated or projected to the primary datum; (2) Rotation of the inclined secondary or tertiary datums to be perpendicular to primary datum.

DOF: Degree of freedom are the position, orientation and size of geometric entities when treated as rigid bodies in 3D.

Feature: A feature refers to a physical portion of a part, such as a surface, hole or slot. Dimension is a numerical value expressed in appropriate units of measure and indicated on a drawing and in other documents along with lines, symbols and notes to define the size or geometric characteristic, or both, of a part or part feature.

Feature of Size (FOS): Any three dimensional feature with a size dimension is a feature of size.

First Order Tolerancing: GD&T based only on geometric conditions for assemblability.

GD&T: Geometric Dimensioning and Tolerancing.

Local DRF: Feature that is datum for another feature and itself has position and orientation tolerances.

Master DRF: Feature that is datum to another feature but does not have any position or orientation tolerance.

Second Order Tolerancing: GD&T based on both assemblability and design intent/function.

Third Order Tolerancing: GD&T based on all of the above, while optimizing manufacturing cost.

Tolerance Allocation: Given the acceptable range of variation on an assembly stack (e.g., clearance), tolerance allocation involves the distribution of tolerance (values) between all contributors to that stack.

Tolerance Analysis: Checking the extent of variation of a dependent dimension or clearance for a given GD&T scheme. Analysis may be done at 1D, 2D, or 3D level; analysis may be for worst case (100% interchangeability) or statistical (typically 6σ).

Tolerance Schema: the tolerance types needed on each feature, datum reference frames for each, the sequence in which to apply dimensional and geometric controls (datum flow chain).

Tolerance Synthesis: Determination of allowable geometric and dimensional variations to meet design function; consists of schema development and tolerance allocation.

CHAPTER 1

INTRODUCTION

1.1 Background

1.1.1 Tolerancing Background

It is impossible to manufacture a part without imperfections and deviations from the nominal geometry [1].These deviations occur because of many reasons such as inherent inaccuracies in the machine tool and its tooling, environmental conditions, jigs, and fixtures. Tolerances define the degree to which the nominal design can vary without compromising functional requirements [2]. The need for organized tolerancing framework appeared after mass production started with the Industrial Revolution in 1800s. Prior to that, mechanical assemblies were made in small quantities, so each part was designed and manufactured in a way that it could assemble with prior parts. Hence, there was no specific tolerance framework for making assemblies in huge quantities. After the Industrial Revolution, manufacturers started to mass produce in order to reduce costs. As a result, they needed a proper way to define the allowable tolerances and the traditional plus-minus tolerance framework was developed. In this framework, allowable deviations of size, location and orientation are represented by plus-minus sign (Figure 1 - 1). This standard is still used in some companies.

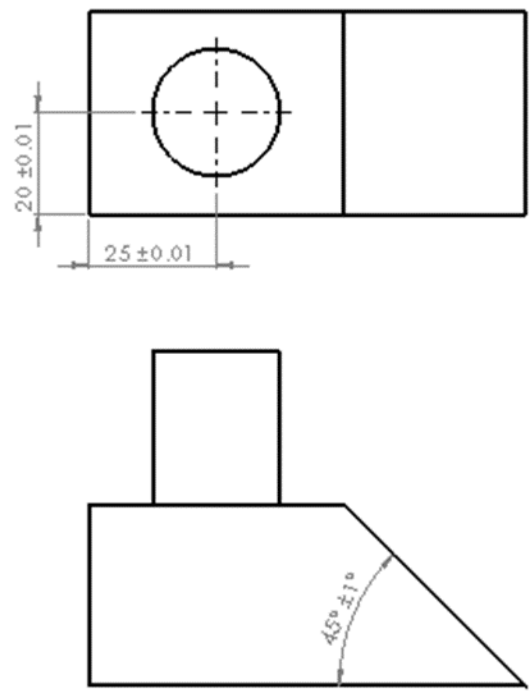


Figure 1 - 1: Traditional Plus-Minus Tolerancing Framework

Generally, it is hard for a designer to devise proper tolerance values since it requires strong practical knowledge and years of experience [3]. If the tolerances are too large, the manufactured parts cannot be assembled and the assembly scrap rate will increase. If tolerances are too low, it will result in exorbitant manufacturing costs. Hence, the designer's job is to allow reasonable freedom to a manufacturer for imperfections and inherent variability without compromising performance. In other words, parts should be manufactured as inaccurate as possible while functional requirements are satisfied. The designer can achieve this goal by applying only general tolerances. The allocated values of general tolerances shall be equal or greater that the customary workshop accuracy [1]. Customary workshop accuracy is defined as with tolerances that can be achieved by normal effort and using normal machinery. Therefore, applying proper tolerance is a tradeoff between functional requirements, manufacturing limits, complications in inspection and cost of the products.

1.1.2 GD&T Background

As mentioned earlier, the traditional tolerance standard defines the allowable deviations from nominal geometry by plus/minus sign. Although this standard is still in use in some companies, there are two major problems associated with it.

The first problem refers to the true position tolerance zone. In this method, the allowable tolerance zone is square, while some investigations on plus-minus framework during World War II revealed that features out of the square zone are still acceptable as long as they are within a circle that encompasses the corners of the square zone (Figure 1 - 2). Second, the interpretation of the defined tolerances usually differs from designer and machinist points of view.

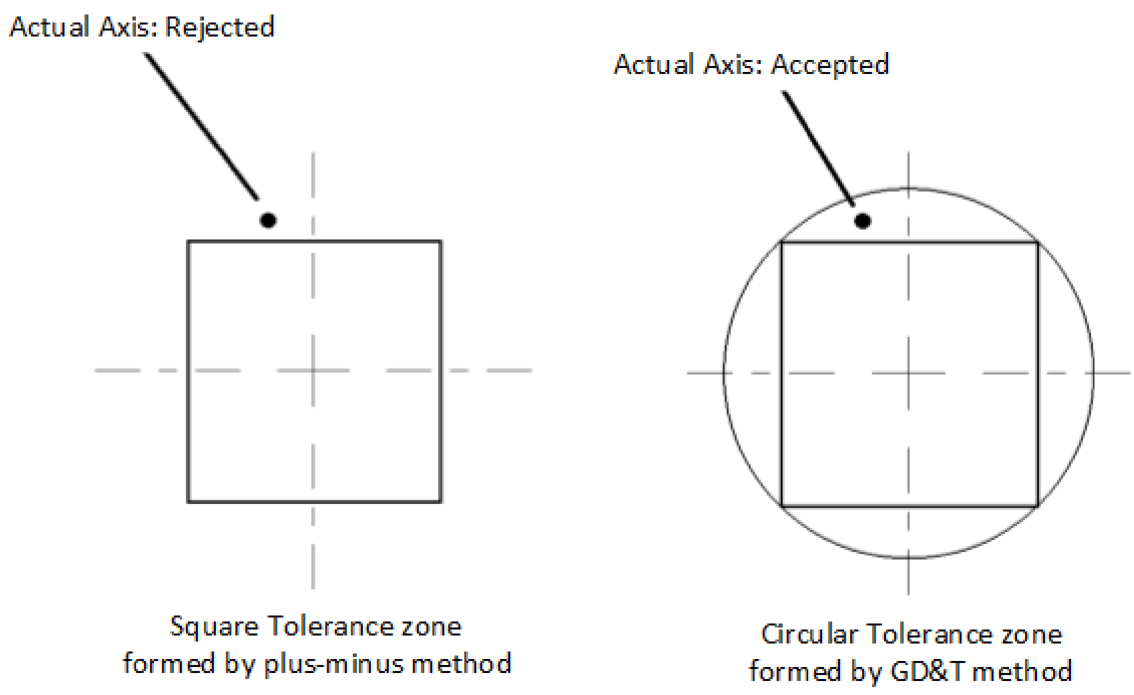


Figure 1 - 2: True Position Tolerance Zone

Due to the problems mentioned above, the efforts for having a comprehensive tolerance framework led to initial Geometric Dimensioning and Tolerancing (GD&T) standards both in Britain and the United States during the World War II.

In Britain, high assembly scrap rates hindered wartime production seriously. The British Military assessed that high scrap rates are the results of the weakness of the plus-minus system and the absence of the complete information on technical drawings. British published set of pioneering drawings' standards in 1944 and "Dimensional Analysis of Engineering Design" standard in 1948 due to demands of war. This was the first comprehensive standard that used the fundamental concept of true position tolerancing (cylindrical tolerance zone rather than square one) [4].

On the other hand, in the United States, the US Army followed the British Military and published the first dimensioning and tolerancing "MIL-STD-8" in 1949. This standard was followed up by "MIL-STD-8A" in 1953 and "MIL-STD-8B" in 1957. At the same time, the American Standard Association (ASA) and the Society of Automotive Engineers (SAE) had their own dimensioning and tolerancing standards which were close to "MIL-STD-8B" standard. After years of debate, the American National Standard Institute (ANSI) - successor to the ASA- published first united GD&T standard in 1966 known as ANSI Y14.5. This standard was updated by ANSI in 1973 and 1984 [4]. The American Society of Mechanical Engineers (ASME) published the next update of Y14.5 standard as the ASME Y14.5 in 1994. The current Y14.5 standard was published in 2009 [5]. This standard is extensively in use in many companies. International Standard Organization (ISO) also publishes another widely in use GD&T standard known as ISO 1101[6]. Both ASME

Y14.5 and ISO 1101 standards address all the issues and ambiguities associated with traditional plus-minus tolerancing.

1.1.3 GD&T Standards

As mentioned above, there are two GD&T standards in use: ASME Y14.5 and ISO 1101 [5,6]. These standards define two different tolerance classes: a class for dimensional variations (Size), and a class for geometric variations (form, orientation, profile, location, and runout). This categorization is made because the types of variation that need to be controlled depend on assembly and functional requirements. For example, form needs to be controlled for surface variations; perpendicularity is important for insertion of long features and location must be controlled to satisfy assemblability condition. Figure 1 - 3 shows the tolerance classes according to the ASME Y14.5 2009 standard.

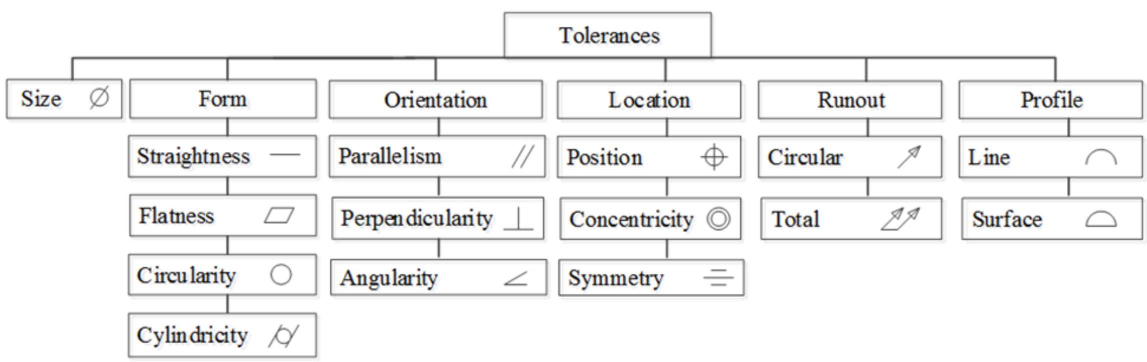


Figure 1 - 3: Tolerance Classes Defined in ASME Y14.5 Standard

GD&T plays a very important role in connecting the different stages of the Product Life Cycle from process planning to inspection planning [7]. Desired tolerances by designers, manufacturers and inspectors are different in each stage of the Product Life Cycle (Figure 1 - 4). Designers usually prefer tight tolerances due to fact that they want functional requirements to be satisfied. Manufacturers tend to use loose tolerances because they are concerned with the ease of manufacturing and the cost. Inspectors also prefer loose tolerance since they are eager to keep inspections cost as low as possible [7].

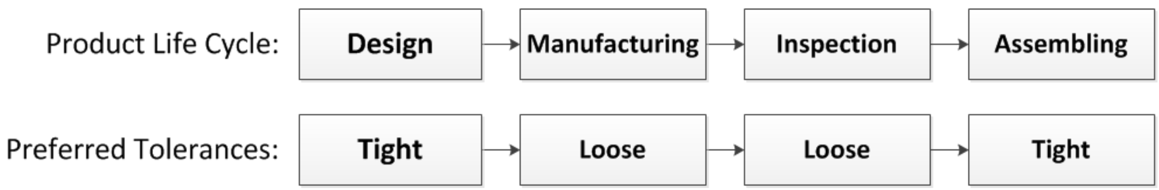


Figure 1 - 4: Desired Tolerance During Different Stages of the Product Life Cycle

1.2 Problem Statement and the Scope of the Research

Designing an assembly includes two major objectives: nominal design and tolerance design [8]. Tolerance design includes two main tasks: tolerance synthesis and tolerance analysis. Tolerance synthesis determines the allowable geometric and dimensional variations to meet assemblability conditions and design function and it consists of tolerance schema development and tolerance values allocation. Tolerance analysis checks the extent of variation of a dependent dimension or clearance for a given GD&T scheme. Tolerance analysis may be done at 1D, 2D, or 3D level; analysis may be for worst case (100% interchangeability) or statistical (typically 6σ). Tolerancing of the design is done typically by the designers towards the end of the design process. Application of a proper tolerance scheme requires in-depth knowledge and years of experience. A designer manually applies tolerances to a model of a part or assembly depending upon the type of functionality and performance. Since the number of tolerances and relationships increase as the number of parts in an assembly increases, a usual tolerancing task by a designer becomes cumbersome and more subject to error. As a result, designers often create inconsistent and incomplete tolerance schemes. Commercial interactive computer aided tolerance software (CATS) tools are available today for worst case and statistical tolerance stack analysis. However, these tools do not aid synthesis; they require the user to input a complete and consistent trial GD&T scheme. Automating GD&T synthesis from nominal models of assemblies was started in the Design Automation Lab (DAL) in order to investigate the degree to which tolerancing can be automated. In the DAL, we classified automated tolerance synthesis into three levels: first order tolerancing in which GD&T is based only on the geometric conditions for satisfying assemblability, second order tolerancing where GD&T is based on both assemblability and design intent, and third order tolerancing in which developed GD&T scheme addresses all of the above, as well as manufacturing cost optimization. Currently in the DAL, a first order automated tolerancing software is being developed. Developing such a toolset requires investigation of the following areas:

1. Identification of the features that need tolerance (to-be-toleranced features)
2. Determination of the required tolerance types
3. Computer representation of features, constraints, and tolerances
4. Identification of the sequence of control (datum flow chain)
5. Determination of the datums and datums reference frames (DRFs) for feature controls
6. Iterative allocation and analysis of the tolerance values
7. Translation of the recommended GD&T to a standard format

Steps 1 to 5 refers to the automated tolerance schema synthesis which is the focus of my research.

1.3 Organization of the Thesis

This thesis consists of seven chapters. Chapter 2 is the literature review. In this chapter the previous works and topics related to this research will be studied in detail. Chapter 3 gives a review of the past work done in the DAL for auto-tolerancing project. In this chapter, all preprocessing modules and interaction between them are explained. Chapter 4 covers the changes made on preprocessing modules. The conceptual design of the first order GD&T schema generation module is discussed in chapter 5. In this chapter, the steps that need to be taken in order to generate automated GD&T schemas are explained. The current implementation of the tolerance schema generation module is explained in chapter 6. The implementation of the module includes the data structures and function pseudo codes. Chapter 7 contains the case studies and comparisons of the software- generated and manually-created GD&T schemas. Finally, chapter 8 summaries the research, addresses the limitations of the current work and proposes future works.

CHAPTER 2

LITERATURE REVIEW

2.1 Identification of the To-be-toleranced Features

In order to ensure that assemblability condition is satisfied in an assembly, critical assembly-level properties of the product should be identified and controlled. Assemblability is defined as "the ability to assemble/fit a set of parts in a specified configuration given a nominal geometry and its corresponding tolerances" [9]. Whitney [10,11] refers to the critical assembly-level properties as key characteristics (KCs) that relate a datum or feature on one part to the one on another part in the assembly. Thornton [12] defines KCs as product, subassembly, part and features that their variations from nominal geometry affect the assemblability, performance, and the final cost of the assembly. Many companies use similar definitions to determine KCs. The technical drawings created by these companies usually contain hundreds or thousands of dimensions and tolerances while many of these dimensions and tolerances are not required [10]. Whitney uses the ability of complete nominal geometry representation in the current CAD systems and adopts KCs to focus attention on those dimensions that are critical, affected a variation-sensitive characteristic, and are worth controlling. By his definition, a feature is not considered as KC if it is not critical, if it doesn't affect a variation-sensitive characteristic, or the cost of controlling that feature is not rewarded.

Haghighi et al [13] suggest the extraction of tolerance loops (stacks) to find critical assembly conditions (KC) that need to be controlled. A tolerance loop or stack is a sequence of feature variations that contributes to the dimensional variation of a feature of interest (e.g., clearance) [9]. Various researches have been conducted on automatic extraction of tolerance loops. Lai and Yuen [14] suggest a vector-based datum transformation schema for extracting tolerance stacks. Anselmetti [15] proposes a method for defining the tolerances based on the assembly process. These two studies extract tolerance loops from models that already have GD&T. Hence, the critical features and dimensions have been already defined by the designer. Haghighi et al [13] suggest and develop the automated tolerance stack extraction scheme without the benefit of GD&T as a part of Auto-Tolerancing project in the DAL. Although their aim is to extract tolerance stacks, the actual output of their software is assembly feature loops since datums are not participating in these loops. Their method requires the identification of the assembly features and extraction of the directions of control (DoC). An assembly features (Figure 2 - 1) is defined as a stereotypical association between two part features, which are on different parts [16]. Similarly, a part feature is defined as a stereotypical shape with certain topological and geometric properties. Mohan et al develop the Assembly Feature Recognition (AFR) toolset that extracts assembly features from the nominal CAD geometry in a neutral B-Rep format (STEP AP203) automatically.

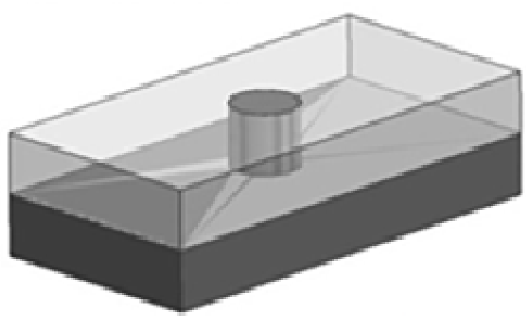


Figure 2 - 1: A pin-hole assembly feature

DoC is referred to the finite number of directions in a mechanical part in which dimensional variations and tolerances of the features are controlled. For example, a simple axisymmetric part has two directions: axial and radial (Figure 2 - 2). Mohan et al develop a toolset that extracts all DoC from the nominal geometry provided in a CAD model.

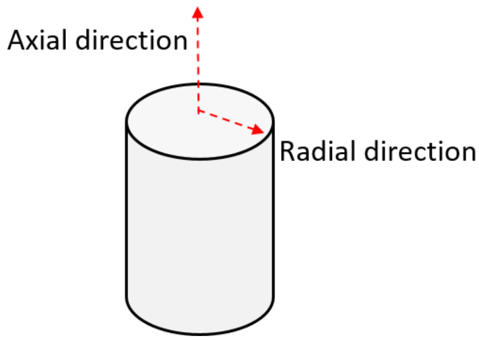


Figure 2 - 2: An Axisymmetric Part with Two Directions of Control

2.2 Determination of the required tolerance types

ASME Y14.5 GD&T standard [5] defines six tolerance types: size, location, orientation, form, profile, and run-out. Haghighi et al [8] use GD&T good practice rules extracted by GD&T experts in RECON SERVICES and reported by Rao [17] to define that only size, location, orientation, and form tolerances are essential for satisfying the assemblability condition (first order tolerancing). Profile and run-out tolerances are usually used to cover functional (second order) tolerancing issues. Haghighi et al refer to size and location tolerances as primary tolerances. These tolerances are only applied to features of size (FOSs). A FOS is any three dimensional feature with a size dimension (Figure 2 - 3). Orientation and form tolerances are defined as secondary tolerances and can be applied to any feature (FOS or non-FOS). The reason that orientation and form tolerances are called secondary tolerances is that according the ASME Y14.5 rule #1, variations in the orientation and form of a feature can be controlled by the limits defined by size tolerance and location tolerances.

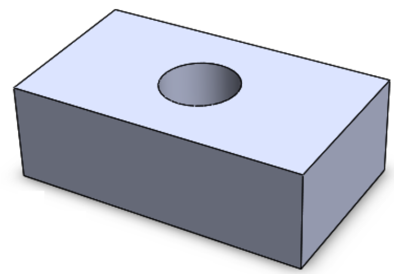


Figure 2 - 3: A Cylindrical FOS (Hole) Inside the Cuboid

2.3 Computer representation of features, constraints, and tolerances

Regarding ASME Y14.5 standard [5], dimension is a numerical value that defines the size and geometric characteristic of a part or part feature. This standard also defines tolerance as total amount by which a specific dimension is permitted to vary. In order to represents these definitions in computers, they need to be translated in mathematical forms.

Yu et al [18] reviewed several schemes for computer presentation, representation[1] and analysis of the dimensions and geometric tolerances. Requicha [19–21] proposed the use of nominal surface normal offset in order to represent tolerance zone. On the other hand, Ingham [22] tried to address tolerance analysis problems by representing tolerances with kinematic mechanism rather than defining the meaning of tolerance.

Wu et al [23] develop a model for GD&T representation in computer systems to support tolerance specification, validation and tolerance analysis. Shen [24] develop the ASU GD&T Global Model based on the model proposed by Wu for the assemblies containing sub-assemblies. ASU GD&T Global Model serves as a constraint, tolerance, feature (CTF) graph that contains all the information needed for delivering a complete GD&T information (Figure 2 - 4).

[1] Tolerance presentation and representation refer to the computer-based graphical and semantic definition of tolerances respectively

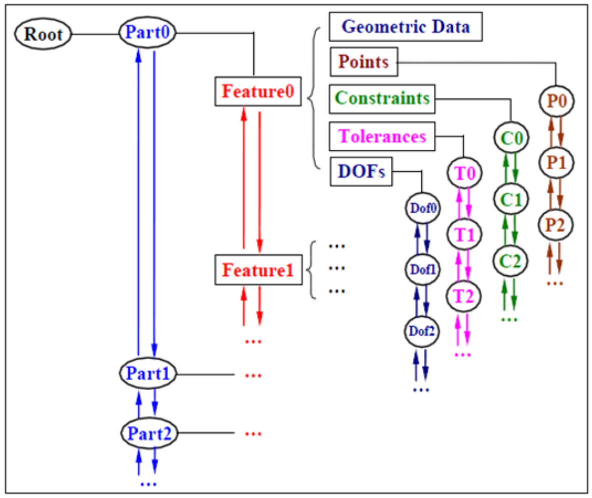


Figure 2 - 4: ASU GD&T Global Model

CTF-Graph data structure is constructed by five main sections (Figure 2 - 5).

1- Section A includes the name and directory of the input geometry.

2- Section B contains the features and their geometric parameters in each part.

3- Section C encompasses constraints' data and their metric relations.

4- Section D gives the tolerance information and their corresponding DOFs

5- Section E covers assembly hierarchy

```
#0=FILE('C:\ASU_GDTtestbed\sat\A4_slot_tab_all_wrt_AC_2_floating.sat');
```

Section A: File directory

```
#1=PART('part0', #2, #3, #4, #5, #6);
#2=RECTANGULAR_PLANE('FACE0_of_part0', (10.8833, 2.98359, 17.7), [0, 0, -1], 76.2, 25.4, [-1.16559e-017, 1, 0]);
#3=RECTANGULAR_PLANE('FACE7_of_part0', (10.8833, 2.98359, 43.1), [0, 0, 1], 76.2, 76.2, [0, 1, 0]);
#4=SLOT('CTP(FACE2&FACE5)_part0', (29.9333, 2.98359, 30.4), [0, 0, -1], 12.7, 12.7, 76.2, [0, 1, 0]);
```

Section B: Part and features

```
#13=CST_DISTANCE(25.4, #2, #3);
#14=METRIC_RELATIONSHIP(#13, CST_DISTANCE, (25.4, #2[PLANE], #3[PLANE]));
#15=CST_DISTANCE(19.05, #4, #5);
#16=METRIC_RELATIONSHIP(#15, CST_DISTANCE, (19.05, #4[PLANE(center plane of SLOT)], #5[PLANE]));
#27=CST_M_FLOAT(#4, #8);
#28=METRIC_RELATIONSHIP(#27, CST_M_FLOAT, (#4[PLANE(center plane of SLOT)], #8[PLANE(center plane of TAB)]));
```

Section C: Assembly constraints and basic dimensions

```
#31=T_FLATNESS(#2, (nFI, 0, RFS));
#32=DOF(#31, (SHAPE_DOF));
#33=T_PARRALLELISM(#2, (FI, 0, RFS), PD(#3, RFS));
#34=DOF(#33, (#3, RDOF[-1.16559e-017,1,0], RDOF[1,1.16559e-017,0]));
#35=T_DIMENSION(#2, (nFI, 0, RFS), PD(#3, RFS));
#36=DOF(#35, (#3, TDOF[0, 0, -1], RDOF[-1.16559e-017, 1, 0], RDOF[1, 1.16559e-017, 0]));
#37=T_SIZE(#4, (nFI, 0, RFS));
#38=DOF(#37, (SIZE_DOF, SHAPE_DOF));
#39=T_POSITION(#4, (FI, 0, MMC), PD(#3, RFS), SD(#5, RFS));
#40=DOF(#39, (#3, RDOF[0, 1, 0], RDOF[0, 0, 1]), (#5, TDOF[1, 0, 0]));
```

Section D: Tolerances and corresponding DoF

```
#59=ASSEMBLY('assembly_0', #1, #7)
#60=MODEL(#59)
```

Section E: Assembly hierarchy

Figure 2 - 5: A Sample CTF File and Its Sections

2.4 Identification of the sequence of control (datum flow chain)

Current CAD systems are part-oriented rather than assembly-oriented; they don't capture design intent of an assembly in conceptual design level. In part-oriented assembly design, the dimensional relationships explicitly defined between parts of an assembly are the ones convenient to construct CAD models and not necessarily those that need to be constrained for proper functionality [11]. Assembly must be designed in way that its KCs get achieved (delivered) once parts are manufactured and assembly together. Whitney [11] presents the concept of datum flow chain (DFC) for achieving this goal. There are two types of joint that connect parts in the assembly: the joints that stablish dimensional relationship between parts (mates) and the joints that only support and fasten the parts once they are located (contacts). Therefore, mates are directly associated with the KCs since they define spatial relationships between parts. The DFC can be graphically represented as a directed acyclic graph for dimensional transfer with node representing the parts and arcs representing mates between them. The DFC is a concept that must be regarded prior to the assembly design since it provides the location strategy before performing any kind of analysis. Once DFC designed for an assembly, assembly design and planning can benefit from it.

2.5 Determination of the datums reference frames (DRFs) for feature controls

DRF is coordinate systems used to locate and orient a part feature. All measurements for the features, which have geometric tolerances related to the datums will originates from the established datum reference frame, and not a part. Geometric tolerances related to a DRF are tolerances of location, orientation run-out and sometimes profile. Datum reference frame usually consists of one to three entities. Theoretically, secondary and tertiary datums should be perpendicular or align with the primary datum. (1) when actual feature of secondary datum is not aligned with the primary datum as it is supposed to, secondary datum needs to be translated or projected to the primary datum; (2) Rotation of the inclined secondary or tertiary datums to be perpendicular to primary datum.

Haghighi et al [8] propose the process of datum selection and DRF creation based on the good GD&T practice rulesets reported by Rao [17] and extraction of the DoC [25]. These rulesets set criteria for the features that can be potentially selected as datums based on their geometric properties such as area, length, and aspect ratio. These rules also control the relationship between datums in a DRF. The main problem associated with the datum selection rulesets is that number of datums required for controlling each type of target feature has not been defined. The combination of the datums in a DRF should constrain all rotational and translation degrees of freedom (DOFs) of the target feature. As it shown in Figure 2 - 6, each rigid body in 3D space has three translational (along x, y, and z axes) and three rotational DOFs (around x, y, and z axes).

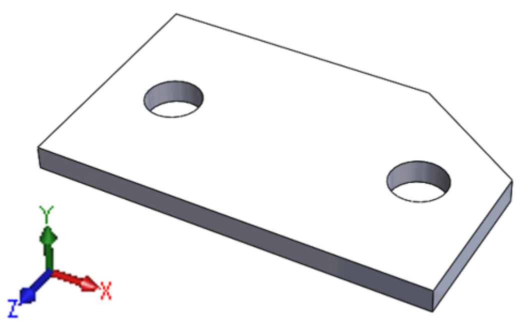


Figure 2 - 6: A Rigid Body in 3D space

Shen [7] uses the concept of DOF algebra to explain how many DOFs of a target feature are controlled by each datum in the datum reference frame. Each primitive geometric elements (point, line, and plane) has some active and some invariant DOFs.

Invariant DOFs are the ones that do not influence entity's shape, location and orientation. In the variation based tolerance analysis, it is not necessary to constraint the entity's invariant DOFs. Figure 2 - 7 shows the active and invariant DOFs of different geometric primitives.

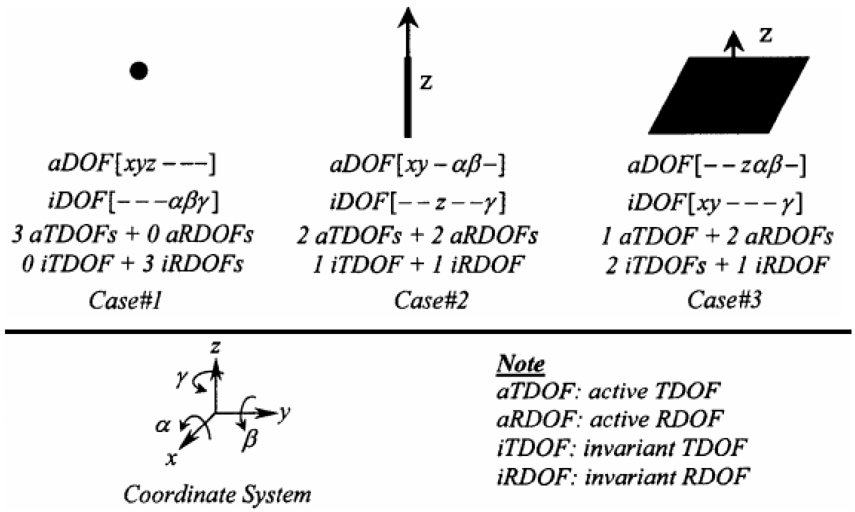


Figure 2 - 7: Active and Invariant DOFs of the Primitives

Shen discusses that other primitives such as cylinder, tab/slot, can be considered as the special cases of the geometric primitive elements. For example, a cylindrical pin is a line that has radius and height. Sphere, cylinder and are all FOS. If their size control is treated as a special DOF [23], they will have size DOF(s), in addition to appropriate translational and rotational DOFs.

2.6 Automated Tolerance Synthesis

Most of the previous computer-based approaches on tolerance synthesis automation have been done in the tolerance representation, value allocation and analysis area [2]. Only a few researchers have tried to automate the process of tolerance schema synthesis.

One of the first attempts for automating tolerance synthesis was done by Lu et al [2]. Although they did not implement an actual software, they developed a framework that addressed all the validity, consistency, sufficiency issues of the tolerances and the functional requirements of the assembly for tolerance synthesis. Their framework incorporates seven computing and data representation tasks to support computer-based tolerance synthesis. In their proposed model (CASCADE-T), geometric tolerances are synthesized based on the functional requirement and shape information of the part in a manner that complex geometric solids are constructed from combination of the primitive solids in the Constructive Solid Geometry (CSG) method; each complex geometric tolerance is created from combining primitive tolerances. Then, the final tolerance scheme is synthesized according to the available manufacturing processes, vendor specifications, cost and reliability. They report that the implementation of the CASCADE-T relied heavily on the several key contributors in tolerance theory and Artificial Intelligence (AI). Besides, the functional requirements such as ability to assemble, performance requirements, and so on should be either included in conditional tolerance relations or added with additional equations.

The attempt for developing an automatic tolerance synthesis software started in Design Automation Lab (DAL). Mohan et al [9] proposed and developed four preprocessing analysis modules toward collecting part and assembly characteristics in support of automating GD&T synthesis for mechanical assemblies represented as neutral B-Rep. The first module is Assembly Feature Recognition (AFR) which extracted assembly features and mating constraints. The second module is Pattern Feature Recognition (PFR) that provides pattern features information in the assembly. The third module is Directions of

Control (DoC) where the directions that control dimensions and tolerances of critical features are extracted. Last module is Assembly Analysis (Assembly Loop Detection) which extracts all tolerance loops in the assembly. These modules will be explained in detail in next chapter. Haghighi et al [8] proposed a conceptual design of an automated first order tolerance schema generation and value allocation toolset. They showed how a tolerance schema can be synthesized automatically using the assembly information and experiential GD&T rules reported by Rao [17].

CHAPTER 3

REVIEW OF THE PREVIOUS WORK DONE IN DESIGN AUTOMATION LAB

Auto-Tolerancing project was initiated in the DAL more than ten years ago to aid designers generate proper tolerance schemas. During this period, many tools that were required for developing the Auto-Tolerancing software were prepared. The actual step for developing and implementing Auto-Tolerancing software was taken as a part of DARPA Adaptive Vehicle Make (AVM) three years ago. Mohan et al [9] proposed and developed four preprocessing modules that provide the required assembly information for GD&T synthesis. These modules are Assembly Feature Recognition (AFR), Pattern Feature Recognition (PFR), Directions of Control Recognizer (DoC) and Assembly Analysis Module (Assembly Loop Recognizer). Figure 3 - 1 shows the preprocessing modules and the interactions between them.

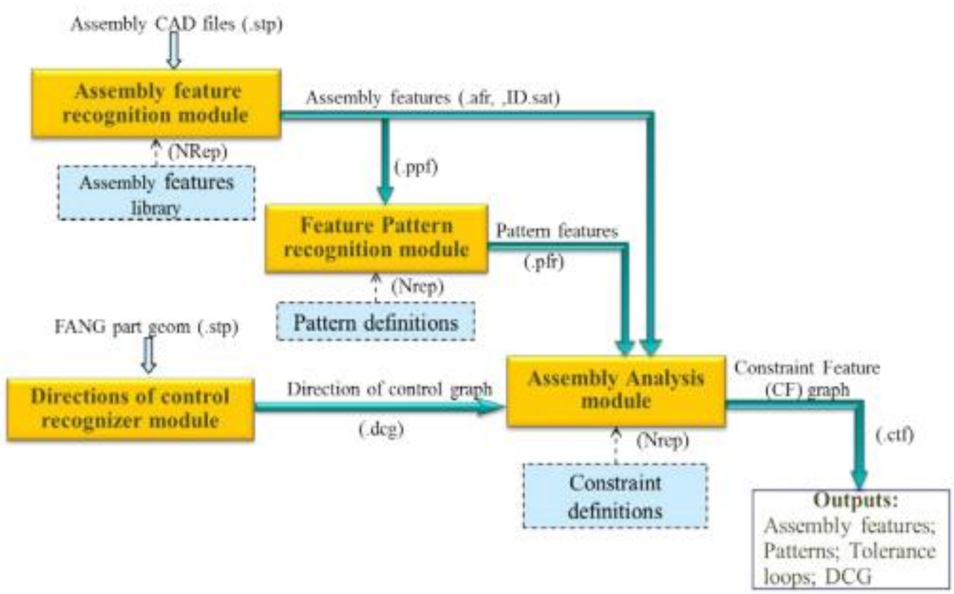


Figure 3 - 1: Assembly Preprocessing Modules and Data Flow

3.1 Assembly Feature Recognition (AFR)

Recognition of assembly features facilitates understanding of an assembly's function and assemblability conditions that must be satisfied. Mating part features derived from assembly features are the critical (KCs) features which their deviations from nominal geometry should be controlled (to-be-toleranced features). Assembly features are typically recognized by finding contacting pairs of faces. It must be mentioned that CAD models constructed by different designers may use different construction methods.

The question which arises here is that what criterion should be used to consider if two faces are close enough to be in contact? We call this the proximity value, the gap between two faces on different parts in order to be considered in nominal contact. Since recognizing assembly features is very sensitive to the proximity value, different ways of determining it must be implemented to make AFR more robust. AFR supported two ways of determining proximity value: adaptive method and hard-coded default value method.

In the adaptive method, the proximity value is calculated as a specific percent of the radius of the largest cylinder in the assembly. Hard-coded default value method uses the value of 1e-3 in logarithmic scale. The absolute proximity value is calculated based on the assembly bounding box. Once proximity value is determined, the assembly feature type should be recognized. AFR reads the input assembly CAD file in STEP AP203 format and uses predefined assembly features library to find assembly features. Currently, assembly features library contains nine types of assembly features (Figure 3 - 2). Vemulapalli et al [26] also developed and implemented a module with graphical user interface (GUI) in which user can define new assembly features by picking contact faces on a CAD model. When the user defines a new assembly feature, it is added to the assembly features library.

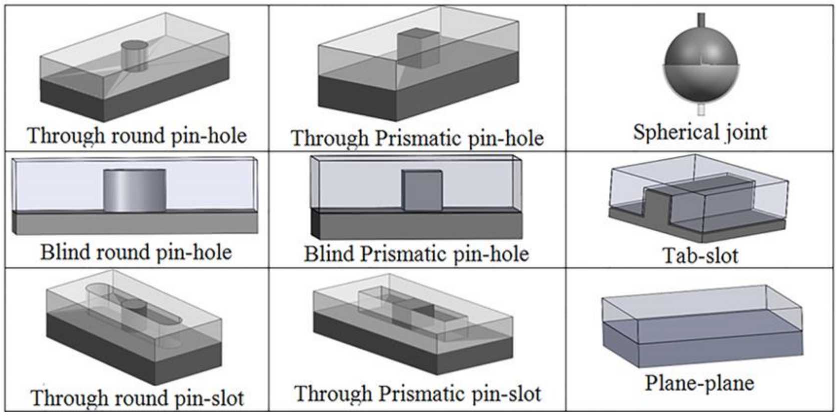


Figure 3 - 2: Assembly Features Library

To facilitate the process of finding assembly features, an input text file that determines the sequence of assembly can be provided by user.

The process of finding assembly features can be summarized as follow:

1. Read assembly CAD file in SAT format
2. Read assembly features library
3. Read the assembly sequence (Optional)
4. Read the liaison graph (If the assembly sequence is provided)
5. Sub-divide parts into groups
6. Calculate the proximity value (By different methods mentioned earlier)
7. Find the contact faces within proximity value
8. Create face adjacency graph with all contact faces
9. Group adjacent contact pairs into sub-graphs
10. Filter assembly feature (AF) definitions
11. Find matching sub-graphs corresponding to AF definition
12. Check the constraints for the recognized features
13. Evaluate the parameters for recognized features

14. Output the feature information to output files

AFR produces two output files (.afr and .ppf) which are consumed by PFR and Loop Detection.

3.2 Pattern Feature Recognition (PFR)

Pattern features (PFs) are helpful for facilitating determination of relative constraints applicable to groups of features, such as bolt holes. They are also helpful for generating a proper GD&T that is compatible with ASME Y14.5 standard. This standard suggests tolerancing pattern features together. There are two main tasks for pattern recognition to detect pattern features: 1. Pattern feature matching which is checking conditions for pattern existence between features. Features that are going to be considered as pattern features must have same type (pin, hole, etc.) and same geometric parameters (diameter, width, etc.). They also must lie on a same plane and mate with same counterpart. 2. Pattern type matching which is matching the pattern type with pre-defined pattern shapes. Figure 3 - 3 shows the library of predefined pattern types. The output of PFR module is a text file (.pfr) containing all patterns in the assembly and pattern parameters.

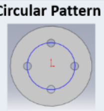
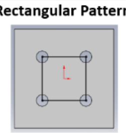
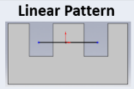
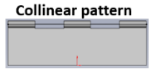
Patterns	Parameters	Extracted Constraints
Circular Pattern 	-Center of pattern Defined as the center of the circle -R_{pin}: Radius of pin pattern circle -R_{hole}: Radius of hole pattern circle -Θ: The angular distance of pins/holes	Global constraint: Center of pin-pattern concentric with center of hole-pattern Position of pins (in the local pattern Co. Sys.) position of hole: $R_{pin} = R_{hole}$, $\theta_{i,pin} = \theta_{i, hole}$
Rectangular Pattern 	-Center of pattern: Defined as the rectangle center - X_{pin} = length of hole pattern/2 - X_{hole} = length of pattern rectangle/2 - Y_{pin} = width of hole pattern/2 - Y_{hole} = width of hole pattern/2	Global constraint: Center of pin-pattern concentric with center of hole-pattern Position of pins match with position of hole: $X_{pin} = X_{hole}$, $Y_{pin} = Y_{hole}$ Length = the longer side of rectangle width = the longer side of rectangle
Linear Pattern 	-distance of the pins/holes from the first pin/hole (x) along the pattern line	Global constraint: Position of pins match with position of hole: $x_{i, pin} = x_{i,hole}$
Collinear pattern 	-Holes center line -Hinges length -Hinges distance	Global constraint: Hinges length be less than the other parts hinges distance Global constraint: Holes centerline to be concentric

Figure 3 - 3: Library of the Patterns

3.3 Directions of Control Graph

In GD&T practices, the dimensional controls are applied in only a small number of particular directions. Regardless of using chain dimensioning, reference dimensioning or geometric tolerancing, all size, and basic dimensions of location line up in a finite number of directions [9]. Extracting potential DoCs in a part is fundamental to GD&T synthesis since it helps finding directions in which variations of the location of the to-be-tolerance features need to constrained. The DoC algorithm processes parts individually. Only planar and cylindrical part faces are used in DoC extraction. All planar faces of the part whose normals are parallel are put into the same DoC. For cylindrical features the axes are used for DoC. This is done after all planar feature DoCs have been determined. Cylindrical features can be put into any DoC whose planes are parallel to the axis (i.e., axis is perpendicular to DoC direction). Typically, axes would appear in at least two DoCs generated from planes, which would be candidate directions for locating the feature [9]. Figure 3 - 4 presents seven DoCs found in the standard part created at ASU DAL.

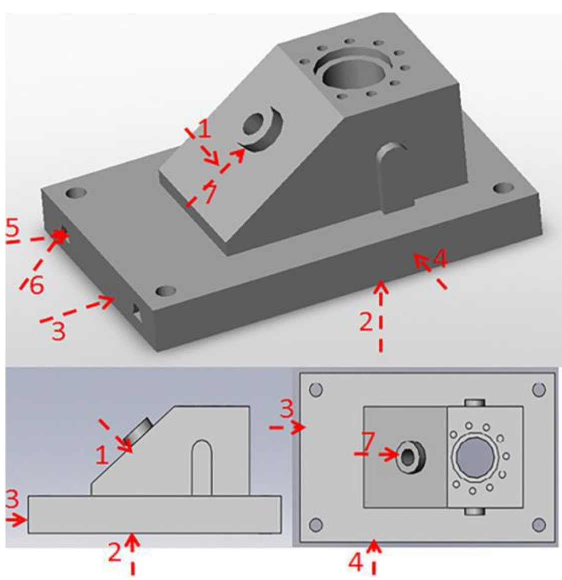


Figure 3 - 4: Directions of Control for a Standard Part Created in DAL

3.4 Assembly Analysis Module (Loop Detection)

A loop is a collection or stack of dimensions and tolerances that contribute toward the accumulation of variation for a target feature. Loop detection involves finding all of the features that contribute to the variation of interest. The purpose of the loop detection task is to locate the shortest, continuous chain of known constraints (i.e., dimensions, mating conditions and assembly features), from the start entity until the end entity [13]. Faces of assembly features (or pattern features) are considered as nodes and the link between the nodes are either: in a part, that comes from the DoC output; or between two parts, that comes from AFR output. The output of Assembly Analysis module is text file containing all unique loops in an assembly. Figure 3 - 5 illustrates a sample loop found in a cam follower assembly.

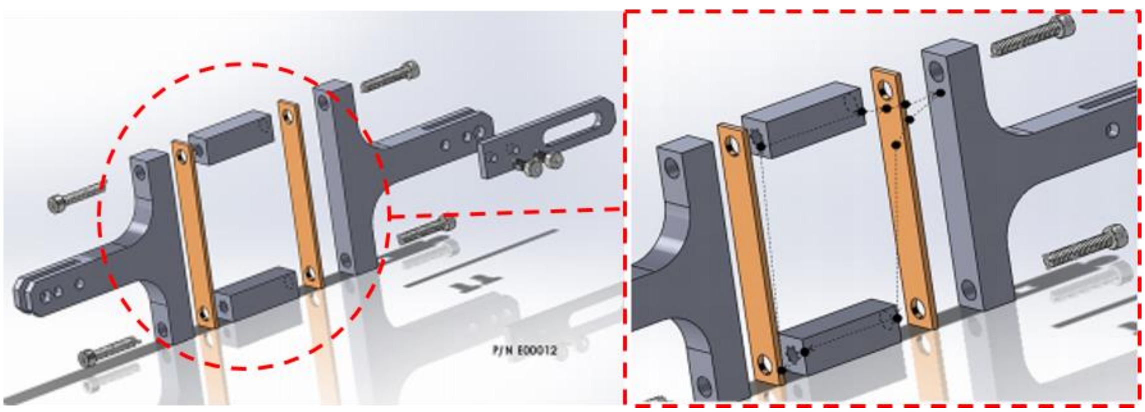


Figure 3 - 5 Assembly Constraint Loop on Cam Follower Assembly

CHAPTER 4

MODIFICATIONS MADE ON PREPROCESSING MODULES

As part of the work for this thesis, the preprocessing modules were updated while work on this research was in progress in order for the second phase to complete correctly. These changes include modifying AFR and PFR

4.1 Modifications Made on the AFR Module

There are two main changes made on AFR: modifying the ways of defining proximity value and adding a new output for the schema generation module consumption.

To make AFR robust in terms finding assembly features, two changes made on the methods of providing proximity value. First, we changed the way of computing adaptive proximity value. Although the previous adaptive method mentioned in chapter 3 worked well for most assemblies, it fails when an assembly doesn’t contain any cylindrical feature at all. In the new implementation, proximity value is calculated by a formula that takes equivalent length of the smallest part’s volume and the ratio of largest part to the smallest part into account. This formula is presented in equation 4. 1.

$$P = \frac{1}{\left[\frac{1^{\sqrt[3]{r}}}{1.5} + 1\right] * 5} * L_{eq} \qquad 4.\ 1$$

Where P = Proximity value, r = $\frac{V_{max}}{V_{min}}$, L_{eq} = $\sqrt[3]{V_{min}}$, V_{min} = Bounding box volume of the smallest part in the assembly, and V_{max}= Bounding box volume of the biggest part in the assembly.

A new way of determining proximity value has also been implemented in AFR. In this method user defines the proximity value in millimeter.

Beside the changes were made for determining proximity value, another change was made in AFR for producing new output file. Early version of AFR discussed in previous chapter produces two output files. None of these outputs provides the complete mating part features and assembly features information for the schema generation purpose. One of these outputs (.afr) contains all features without their geometric parameters (i.e. radius, width, etc.) and the other one (.ppf) only contains prismatic and cylindrical pins and holes with their parameters. Tolerance schema generation module requires a file that contains all features (planar faces, tabs, etc.) along with their geometric parameters. For this reason, AFR was modified in a way that it produces the third output file for tolerance schema synthesis module.

4.2 Modifications Made on the PFR Module

There are two main changes made on the PFR module: the algorithm for finding potential pattern features is improved, a new pattern type defined in the library of patterns.

One the most important conditions to check whether two features form a pattern is that they must mate with the features on the same counterpart. In many assemblies, the features that form a pattern do not mate with opposite features on the same part directly; usually there are chains of mating features on different parts in between. These chains of mating features on different parts are called mating chains. In the early version of PFR developed in the DAL lab, mating chains are constructed for only one level; if features do not mate opposite features on the same part, nor pattern is detected. Figure 4 - 1 illustrates an assembly that holes on the chassis mate with pins on different parts.

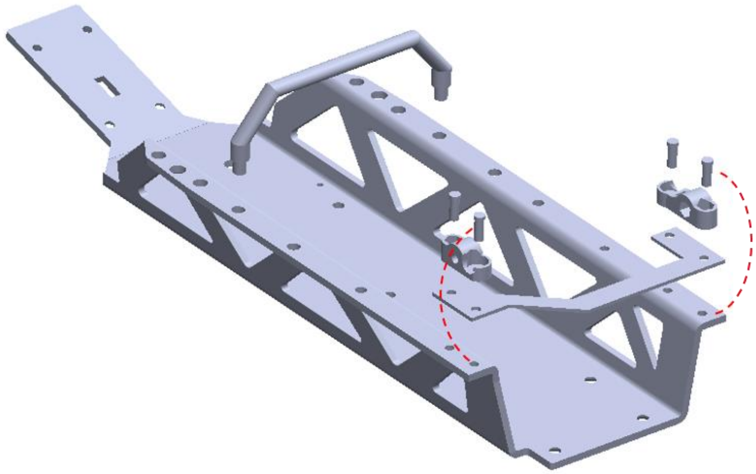


Figure 4 - 1: Holes on the Chassis Mate Pins on Different Part

If we run this assembly though PFR module, no pattern gets detected between the holes on the chassis. To solve this deficiency, I modified the pattern recognition algorithm in a way that a mating chain for each potential pattern feature is constructed step by step. At each step program checks if the potential pattern feature mate the same part. If they do, features are grouped together as pattern features, otherwise, the construction of mating chains is continued until they get exhausted. Figure 4 - 2 shows the construction of mating chains for holes on the chassis and how they end up visiting holes on the same part (rear beam).

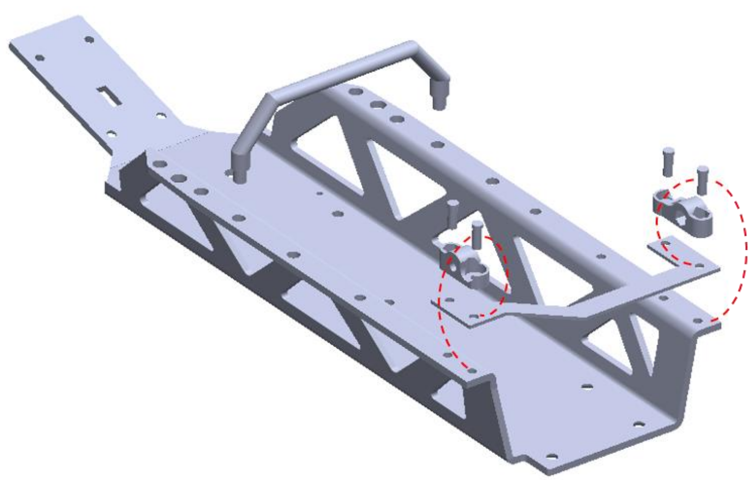


Figure 4 - 2: Creating Mating Chains Leads to Finding Patterns

Beside the change made on the PFR algorithm, a new type of pattern is added to the patterns library. This pattern is a combination of the collinear and linear pattern types and is called collinear-linear pattern (Figure 4 - 3).

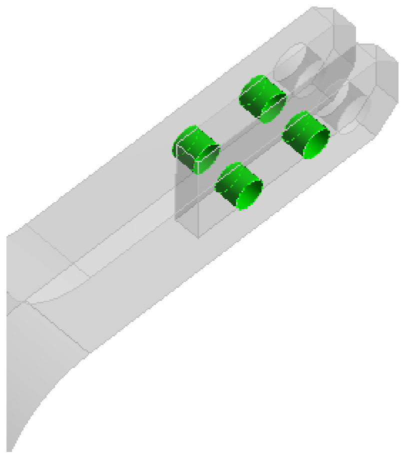


Figure 4 - 3: Collinear-Linear Pattern

CHAPTER 5

FIRST ORDER GD&T SCHEMA GENERATION: CONCEPTUAL DESIGN

To design a knowledge-based system for automating GD&T schema synthesis, the following issues mentioned in chapter 1 should be addressed:

1. Identification and extraction of the to-be-toleranced features
2. Determination of the required tolerance types
3. Computer representation of features, constraints, and tolerances
4. Identification of the sequence of control (datum flow chain)
5. Determination of the datums and datums reference frames (DRFs) for feature controls

This chapter starts with covering the first three issues since they define the architecture of the software. Last two steps will be addressed during the discussion of the schema development process.

5.1 Identification and Extraction of the To-be-toleranced Features

As described in chapter 2, the variations of the critical features from nominal geometry that affect the assemblability of an assembly should be controlled. These critical features are the ones that participate in the tolerance stacks. Two groups of features participate in the tolerance stacks: target and datum features. Regarding first order tolerancing, target features are the mating part features derived from assembly features. The information of these features can be obtained AFR module’s output. However, the information of the datum features in not available since these feature will be identified during the process of DRFs creation.

5.2 Determination of the Required Tolerance Types

Based on GD&T good practice reported by Rao [17], only size, location, orientation, and form tolerances are required to address first order tolerancing issues [8]. Profile and run-out tolerances are usually used when the design intent of an assembly is under consideration. The GD&T rulesets define the conditions for creating each of these for tolerances and will be discussed in detail in section 5.4.1.

5.3 Features, Constraints, and Tolerances Representation

A GD&T scheme contains three major parts: features, constraints and tolerances. As mentioned in chapter 2, several methods are developed for computer representation of these elements. However, most of the methods are only cover one or two of these elements not the whole GD&T scheme. ASU Global GD&T Model that has been developed and updated during the past decade in the DAL is a scheme that capture all the required GD&T information in a neutral representation. In this research, the GD&T information is represented using this model and the recommended GD&T is transmitted to the tolerance value allocation and verification module through the CTF file format.

5.3.1 Features Representation

There are three major types of features currently defined in ASU GD&T Global Model: planar, prismatic and cylindrical. Planar features are in two types: single plane and co-planes. There are four types of prismatic features: tab, slot, pin, hole. Cylindrical features are in two types: pin and hole. There are also two pattern features defined for prismatic and cylindrical features. Figure 5 - 1 presents some of the geometric features defined in a ASU GD&T Global Model.

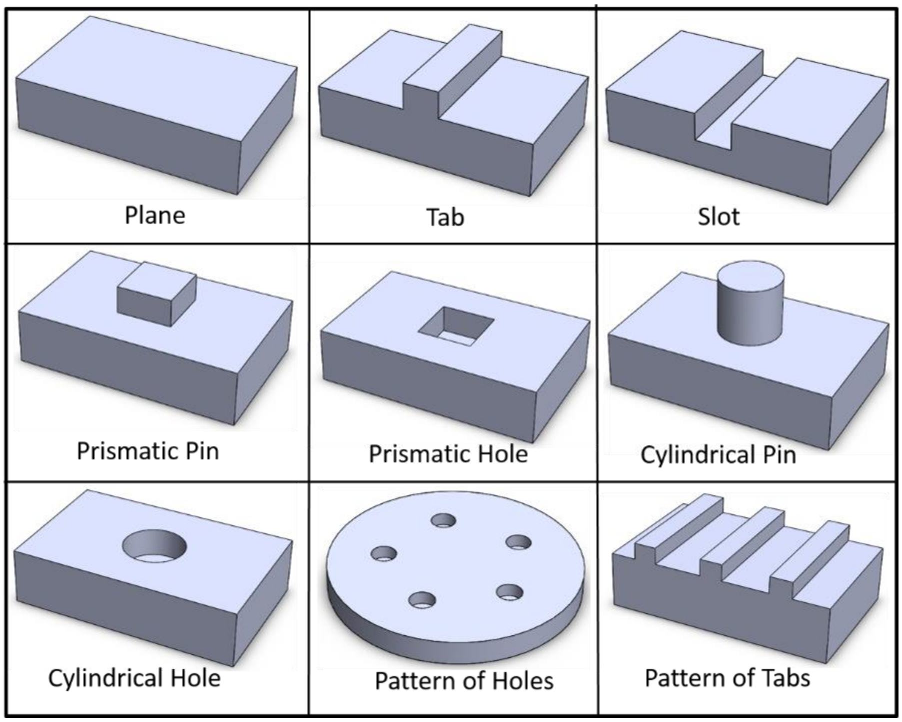


Figure 5 - 1: Some of the Features Defined in ASU GD&T Global Model

5.3.2 Constraint Representation

Currently there are two major types of constraints defined in ASU GD&T Global Model: part constraints and assembly constraints. Part constraint includes distance, coincidence, concentricity, angularity, parallelism, and perpendicularity. Assembly constraint contains float, press fit and against.

5.3.3 Tolerance Representation

There are four major types of tolerances defined in ASU GD&T Global Model currently: location, size, orientation, and form. Location tolerance includes symmetry and position. Size tolerance contains size dimension and location dimension. Orientation tolerances are in three types: angularity, parallelism, and perpendicularity. Finally, form tolerance includes flatness and straightness.

5.4 GD&T Rulesets

Once the to-be-toleranced features, required tolerance types and a way for representing GD&T information is defined, the tolerance schema generation module can be architecture. Schema generation module is a knowledge-based system that requires GD&T rules as its building blocks. Since there is almost no documented theoretical background about generating an efficient tolerance schema, GD&T rules must be extracted from technical drawings containing GD&T information created by experts.

As mentioned earlier, good practice GD&T rulesets are provided by experts of RECON SERVICES, developed in the DAL and reported by Rao [17]. These rules are classified as D, M, L, P and X rulesets and are as follows:

Rules for master DRF selection (M Rule set)

M1. Consider all outermost faces of a part and assembly feature-faces as potential primary datums. Give preference to assembly features if their area is not less than the 60% of the area of non- assembly face area for planar surfaces. Potential rectangular planar face datums should not have an aspect ratio more than 8.

M2. Planar feature with largest area or a cylindrical feature of largest length should be primary datum candidates.

M3. Among two candidate cylindrical features choose the one that has longer length.

M4. Among different planar features, choose them as primary, secondary and tertiary datums based on area.

If there are planar and cylindrical features, then for choosing the primary datum, check the functionality of the part. If the part has free rotational DOF w.r.t the axis of the cylindrical face, then choose the cylindrical face as primary datum. This rule is not applicable to the first part in the critical loop.

M6. Prefer planar face as the higher precedence datum over the cylindrical face. But if the cylinder length is larger than the dimensions of plane, give preference to the cylindrical face as higher precedence datum.

M7. Among two planes, take the one that has good accessibility (accessibility is defined by concavity of surrounding faces) as the higher precedence datum subject to condition that its surface area is not smaller than 60% of largest surface.

M8. Among two parallel features (planar), give precedence to the face that is continuous as a datum rather than the one that has partitions.

M9. For choosing the lower precedence datums, give preference to those faces that are adjacent to higher precedence datums.

M10. In the case of planar features lower precedence datums should be perpendicular to the higher datums.

M11. If the secondary datum is an axis (axis of a cylinder) then the tertiary should be a plane or axis (not concentric) which is parallel to the secondary datum and perpendicular to the primary datum. Or it can be a point that is not on the secondary datum.

M12. If the primary datum is a plane, and, for secondary datum, if an axis (axis of a cylindrical feature) is considered, then the axis should be perpendicular to the plane.

M13. The combination of primary, secondary and tertiary datums should constrain all the freedoms. If not, other combinations of features should be checked which fully constrain the part.

M14. If there are no mating features that can be taken as the secondary or tertiary datums of the DRF, then check if the DOF's of remaining interfacing features need the unavailable datums.

M15. A pattern has to be used as the datum only when the above rules are not satisfied with the features among the mating features.

Rules for additional DRF selection (L Rule Set)

L1. For better control of the orientation of features of size (blind holes, pins, slots and tabs) choose the feature that they intersect with as primary datum.

L2. For tolerancing holes that are parallel to one another, intersecting the same feature, and if the corresponding mating features belong to a single part, tolerance one hole with respect to reference (first) DRF and tolerance the remaining holes with the previous hole as datum. If the corresponding mating features belong to different parts, they should be tolerance with respect to the primary datum.

L3. While assigning positional tolerance to a pattern of holes, choose a hole that is parallel to the pattern and also different in diameter, as the secondary datum.

Rules for Dimensioning (D Rule set)

D1. Features of sizes (FOS) should be directly dimensioned (size)

D2. Dimension the position of all features/faces of a part if they are distant from datums.

D3. Give size tolerance of 0.5% of the nominal size to all features of Size (FOS)

D4. Baseline dimensioning should be used instead of chain dimensioning.

D5. A functional feature should be used as a baseline subject to ease in manufacturing and inspection.

D6. If a feature of size is offset from an edge or surface, then the latter should be used as a datum.

Rules for Position Tolerances (P Rule Set)

P1. Give positon tolerances to every FOS and patterns, unless the FOS is a Datum in a DRF without a higher precedence datum.

P2. While specifying a position tolerance for a cylindrical feature of size, it is best to use the face on which the feature "sits", as the primary datum for the control frame of the position tolerance, unless it is not applicable

P3. In tolerancing a pattern of holes, a finer control of the holes than provided by a single position tolerance is often required. Care should be taken while deciding which of the two -composite or two single segment position tolerancing - is used to introduce the second level of control. Composite tolerancing should be used only when the orientation control (with respect to the datums) needs to be. Two single segment position tolerancing should be used when a refinement of the location is also required.

P4. If the features are concentric with the datums, assign positional tolerance (avoid concentricity tolerance).

P5. When two patterns reference the same datums and in the same order, it is good practice to consider them as one pattern and eventually inspect them as one pattern. Also, the material condition modifiers for the two patterns need to be the same. This does not apply to the second level position control.

P6. For counter-bored holes: For an individual hole, the through-hole is assigned a datum identifier and the hole is positioned to the hole datum at MMC.

Miscellaneous Other Rules (X Rule Set)

X1. Assign flatness to the primary datum if a planar face with largest area is chosen.

X2. For cylindrical features, if the fit is clearance type, assign straightness to the axis; if the fit is interference assign a profile tolerance to the surface for form control.

X3. When applying form tolerances to axis of a cylindrical feature with large length to diameter ratio (>4), it is better to apply a combined overall and unit length tolerance. The same applies for flatness on a large area.

X4. If there is only one interfacing feature, assign form tolerance and do not check any other rules. If there are two, tolerance them with respect to each other based on the rules of DRF; do not check any further rules.

X5. If the features are parallel to the datum features, apply parallelism tolerance to the feature.

X6. If the features are perpendicular to the datum features apply perpendicularity tolerance.

X7. When applying an angularity tolerance, use two datums to establish the desired level of control. It is beneficial to apply orientation tolerances at Maximum Material Condition (MMC) on features of size.

5.4.1 Newly Developed GD&T Rulesets

The initial attempt for conceptual design and implementation of the schema generation module started with using GD&T rulesets discussed above. During the initial implementation process, it was observed that there are three types of issues associated with these rulesets.

First, there is a group of rules that can be interpreted in different ways and there is group of rules that are not related to schema generation. Rule D5 that presents abstract ideas for basic dimensioning is the example of the former group while rule D3 that addresses tolerance value allocation issue is the example of the latter. For the most of the rules in group one, I rewrote the rules based on the most accurate interpretation that was made after several discussions with my colleagues in the DAL. I ignored the rest of the rules of group one that were hard to interpret and all of the rules in second group.

Second, some of the rules address 2nd order tolerancing issues. Rule P3 that reviews the benefits of using composite and multiple single-segment positional tolerancing and rule X2 that addresses functional tolerancing issue by suggesting profile tolerance are two examples of the rules that are not related to the 1nd order tolerancing topic.

Third, there are missing rules that need to be considered. There is no explicit ruleset for extracting basic dimensions. Basic dimensions are essential for tolerance value allocation/analysis since this module hasn't built upon ACIS and it doesn't have access to underlying nominal geometry. There is also no specific rule for selecting secondary and tertiary datums for different types of target features. There must be rules that define how far datum selection needs to be continued for each type of target feature.

Because of all the reasons mentioned above, new rulesets are developed upon previous ones to satisfy 1st order tolerancing issues. The new rulesets are as follow.

Rules for controlling dimensional variations (S rule set)

As it shown in Figure 1 - 3, dimensional variations (size DOFs) of the target features are controlled by assigning size tolerance. Size tolerance is divided into two groups: size dimension and location dimension. Size dimension refers to the size of a FOS while location dimension refers to the dimension between two planes that do not have the same area (i.e. a step feature, the height of a cylindrical pin). This categorization is compatible with the ASME Y14.5 standard and STEP A242 implementation practices. The rules for creating size tolerance frame are as follow:

S1. All mating FOS must have size dimension tolerance frames.

S2. Pattern FOS should have only one size dimension tolerance frame.

S3. All non-mating FOS that participate in datum flow chain must have size dimension tolerance frame.

S4. All size DOFs of a target feature should be controlled by appropriate number of size tolerance frames.

Rules for controlling location variations (L rule set)

Location tolerance is applied to FOS in order to control their active DOFs. Based on the ASME Y14.5 standard [5], location tolerance has three subgroups: position, concentricity, and symmetry. The rules for generating location tolerance frame are as follow.

L1. All mating FOS must have location tolerance frames unless the target feature doesn't have enough datums to constrain its active DOFs.

L2. Pattern features should be toleranced together. For the 1^{st} order tolerancing, assigning composite or multiple single-segment positional tolerancing to patterns can be avoided.

L3. Avoid concentricity tolerance as much as possible since it is difficult to inspect.

L4. If the target feature is a prismatic FOS, it has only one datum feature in its reference frame which is a prismatic FOS datum feature and mid-planes of the target and datum features are coincidence, assign symmetry tolerance frame.

L5. If tolerance is not either concentricity or symmetry, assign position tolerance frame to the target feature.

L6. It is beneficial to apply location tolerances at Maximum Material Condition (MMC) on FOS.

L7. If the local controls between some FOS are important, first feature can be located with respect to the master DRF and the rest of features can be located with respect to first feature (local DRF).

Rules for DRF selection (D rule set)

D1. If two planar faces with the same mating condition (both mating or non-mating) are candidates to be a datum feature, give preference to the one with larger area. It's preferable if the face with larger area has the aspect ratio[2] of less than eight.

D2. If a mating and a non-mating planar faces are both candidates to be a datum feature, give preference to the mating one unless its area is less than the 60% of the non-mating's area.

D3. Among two planar features with the same mating condition, same area and same aspect ratio, give the preference to the feature that is more accessible.

D4. Among two planar features with the same mating condition, same area, same aspect ratio and same accessibility, give the preference to the feature that is continuous as datum rather than the feature that has portions.

D5. Avoid selecting a cylindrical FOS as datum feature reference if it is not participating in any assembly loop.

D6. Give preference to the lower precedence datums that are perpendicular to the higher datums.

---

[2] Aspect ratio for a planar face is defined as ratio of the length to the width

D7. Between two cylindrical datum features candidates, give preference to the one with longer length. It's preferable that the cylindrical FOS with larger length has the aspect ratio less than four.

D8. The process of datum selection is continued until all active DOFs of the target feature are restricted.

D9. For choosing the lower precedence datums, give preference to those faces that are adjacent to higher precedence datums.

D10. Between a cylindrical and a non-cylindrical datum features candidates, give preference to cylindrical feature if its length is larger than 40% of the plane's area.

Rules for orientation and form tolerances (X ruleset)

As it shown in Figure 1 - 3, orientation tolerance is divided into three groups: perpendicularity, parallelism and angularity. Form tolerance is also categorized into four groups: straightness, flatness, cylindricity and circularity. In this project, only straightness and flatness are considered.

X1. Create orientation tolerance frames for all datum features in the part.

X2. Each datum feature must be oriented with respect to its preceding datums. Number of referenced datums should not be more than two.

X3. Orientation tolerance should assign to all mating FOS. Mating FOS should be oriented with respect to two of their datums.X4. If the target feature is parallel to the datum feature, apply parallelism tolerance.

X5. If the target feature is perpendicular to the datum feature, apply perpendicularity tolerance.

X6. If the target feature neither is perpendicular nor parallel to the datum feature, apply angularity tolerance.

X7. Setup form tolerance frames for all datum features in the part.

X8. Setup form tolerance frames for all mating FOS in the part.

X9. If the target feature is a planar face, assign flatness tolerance frame to it.

X10. If the target feature is a prismatic FOS, apply flatness tolerance frame to its mid-plane.

X11. If the target feature is a cylindrical FOS, assign straightness tolerance frame to its axis.

X12. Pattern features must be toleranced together.

X13. It is beneficial to apply orientation and form tolerances at Maximum Material Condition (MMC) on FOS.

Rules for Extracting Basic Dimensions (B rule set)

Basic dimensions are the nominal dimensional values. Basic dimensions are reported in section C of the CTF as constraints. Basic dimensions are divided into 6 groups: distance, coincidence, concentricity, perpendicularity, parallelism and angular constraints. The rules for extracting basic dimensions are as follow.

B1. If a location dimension tolerance has been defined between two features, extract the basic dimension between those features.

B2. Extract the basic dimensions between target features and its datums for location and orientation tolerance frames.

B3. If the target and datum features are in the same direction of control, if the distance is not zero, assign distance constraint, if distance is zero and both target and datum features are cylindrical FOS, assign concentricity constraint, and if distance is zero and both target and datum features are planar or prismatic FOS, assign coincidence constraint.

B4. If the target and datum features are not in the same direction of control, if the target and datum features are perpendicular, assign a perpendicularity constraint, if the target and datum features are parallel, assign parallelism constraint, and if the target and datum features are not either parallel or perpendicular, assign angular constraint.

5.5 Schema Generation

Figure 5 - 2 shows the overview of the tolerance schema generation module's architecture.

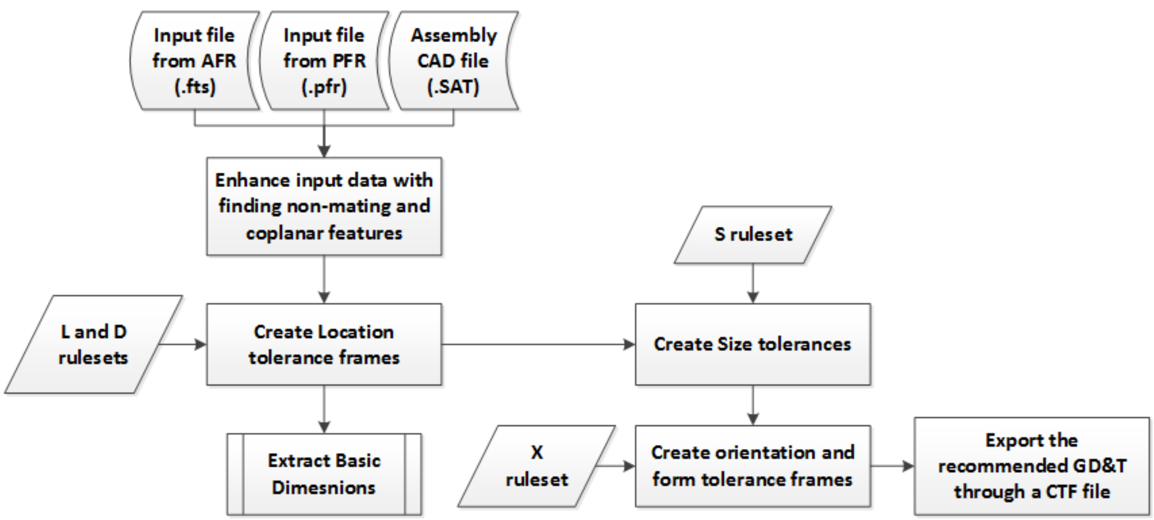


Figure 5 - 2: The Flowchart of Schema Generation Module

5.5.1 Input Information

Tolerance schema generation module reads three input files, a text file containing mating part features and assembly features from AFR (.fts), a text file containing pattern features from PFR (.pfr) and the assembly CAD file from translation module(.SAT). These three inputs help schema generation module to get the initial information about the assembly.

5.5.2 Enhancing the Input Information

Beside the assembly information provided by preprocessing modules, two extra pieces of geometric information should be extracted from the CAD file: non-mating features and coplanar features. This information that is required for generating an efficient tolerance schema enhances the input data and makes the schema more compatible with ASME Y14.5M standard. Recognition of both non-mating and coplanar features requires extraction of the DoCs in advance.

5.5.2.1 Extracting Modified Directions of Control

The main aim of extracting DoCs was discussed in chapters 2 and 3. In this section, the benefit of extracting these directions for finding non-mating and coplanar feature is explained. The automated detection of the DoC method proposed by Mohan et al [25], extracts all possible directions in a part while the GD&T schema generation module only requires specific ones to create a GD&T scheme. These directions are called active directions and are the ones that control the active DOFs of the to-be-toleranced features, and in order to extract them, a modification to the original DoC function is required. This modification takes two steps: first, the list of mating part features features in the part are queried and all unique direction vectors (normals, axes) are obtained; second, during the process of DoC detection, only the directions that are either perpendicular or parallel to the unique direction vectors obtained in first stage, are extracted. This modification to the original DoC function helps datum selection function to go through the list of features that could be potential datums; hence it reduces the computational time and memory consumption. Figure 5 - 3, illustrates this idea for a part that has two to-be-toleranced features (a pin and a slot). Left figure shows the DoCs extracted by the original DoC function while the right figure presents the ones obtained by the modified DoC function.

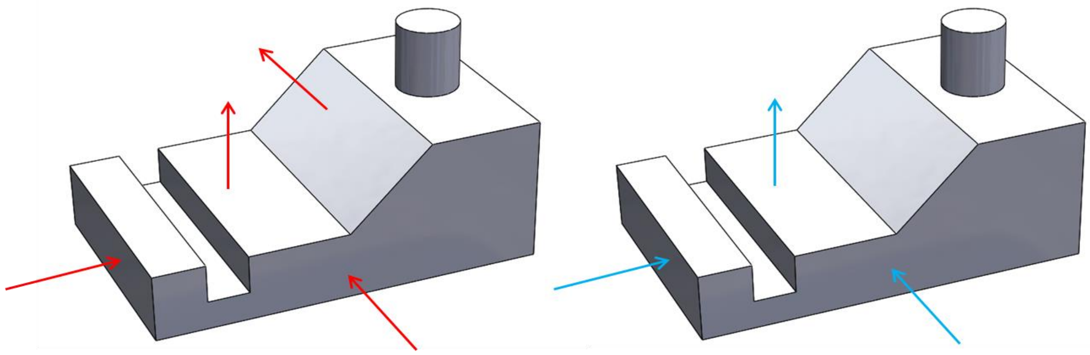


Figure 5 - 3: DoCs in a Part with Two Critical Features

5.5.2.2 Finding Non-Mating Features

The information of non-mating features is required for schema generation since they can be selected as datum features during the datum selection process. Selecting a non-mating cylindrical features as datum feature is usually avoided in GD&T practices, therefore, non-mating feature recognizer only detects prismatic and planar features. The process of finding non-mating planar features is almost straightforward since they only contain one face. On the other hand, recognizing tabs/slots needs extra steps considering this fact that they are defined by two parallel faces. A planar feature is represented by a planar face. If there is a planar face in any DoC that is not mating or hasn't been already detected, a planar feature can be created for that face. Three topological characteristics must be satisfied when two planar faces create a tab/slot feature. First, these two planes must have the same shape (rectangular, circular, and general). Second, they must have the same area and finally their normal vectors must be antiparallel. Once all these three topological characteristics are held for two planar faces, a prismatic FOS can be created.

5.5.2.3 Finding Coplanar Features

In CAD systems, when a face is split into different segments through a boolean operation, each resulting segment is considered as a separate face. However, in GD&T practices, coplanar faces are usually treated as a single feature. In order to generate a tolerance schema that is compatible with ASME Y14.5, a preprocessing function is required to find coplanar faces. Figure 5 - 4 shows a coplanar feature selected as primary datum in a technical drawing.

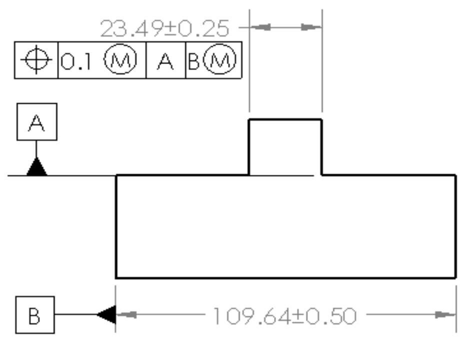


Figure 5 - 4: An Example When a Coplanar Feature is a Datum for a Tab

Coplanar faces have a distinct geometric property: they have zero relative distance with respect to each other. Using this property facilitates detection of coplanar faces. A function for finding coplanar faces needs two stages of operations. In the first stage (preprocessing stage), potential coplanar faces are grouped together based on zero relative distance property. In the second stage (post-processing stage), a coplanar feature is created from grouped faces.

5.5.3 Creating Location Tolerance Frames

Initially, the process of creating location tolerance frames was designed to take place after generating size tolerance frames. In the initial design, size tolerance frames were assigned to all FOS (both mating and non-mating ones) and then the non-mating FOSs that were not participating in tolerance stacks got filtered. Assigning size tolerance frame to all FOS and filtering some of them increased the module's runtime. Hence, the schema development process redesigned in a way that size tolerance creation took place after constructing location tolerance frames.

Generating location tolerance frames require both L and D rulesets. L ruleset covers the criteria for generating different location tolerance frames. On the other hand, D ruleset covers the process of finding potential datums and creating DRFs. To set up location tolerance frames, potential datums need to be found and DRFs should be created first. In the following subsections the feature scoring system, use of DOF algebra for datum selection and the process of creating DRFs and location tolerance frames are discussed.

5.5.3.1 Finding Potential Datums

To create a DRF and generate a location tolerance frame for a target feature, potential datums must be found first. For each target feature, potential datum features must lie in the target's DoCs. The initial design of the function for finding potential datums included all features in the part no matter if they lied in the DoC's of the target features. Then, the ranked features with the highest score were selected as datums if they were in the target feature's DoC. This design was computationally expensive for some test cases where many of the features that scored were not in the DoC's of any target feature. To solve this issue, only features that can potentially control the active DOFs of the target features are extracted using modified DoC function and then scored and ranked using D ruleset. Beside the rulesets for datum selection, identification of the sequence of control helps choosing proper datums.

5.5.3.2 Determining the Sequence of Control (Datum Flow Chain)

It was discussed in chapter 2 that identification of the DFC is procedure that should take place prior to assembly design. A DFC defines what Key Characteristics (KCs) must be delivered by an assembly. Delivering KCs includes designing which joins must be designed as mates and which must be designed as contacts. However, in the Auto-Tolerancing project we are dealing with the situation in which the nominal design of the assembly has been already made by the designer. We can substitute the use of designing DFCs with using extracted assembly loops from nominal geometry (by Assembly Analysis module) in order to find critical loops that need to be controlled. Finding critical loops can help in datum selection process to choose a datum feature that provides a better control for target feature. In the current design of the tolerance schema generation module, the assembly loops information is not used. However, using this information is suggested in chapter 8 as a future work.

5.5.3.3 Feature Scoring System

The datum selection GD&T rules (D rules) explained earlier in this chapter are relative. In other words, these rules define a comparison between two potential datum features in order to find the best candidate. It is better to avoid implementing relative rules directly since at each step of comparison, one feature is qualified over the other one and the unqualified feature is thrown away from the pool of the datum feature candidates. Thus, these rules are better to be represented by some metrics that rank potential datum features by their level of suitability in the pool of potential candidates. Feature scoring system is a procedure to translate the GD&T datum selection rules to the quantitative metrics that are machine understandable. These metrics are the scores that are assigned to the features based on their topological and geometric characteristics. In this design each type of feature has its own specific type of score. It should be noted that all of these scores are designed and developed based on the experiential GD&T rules and observed test cases. Performing more case studies in the future can lead to defining new metrics and/or redefining current scores. The scoring system is designed in a way that these metrics can be tweaked and tuned easily in the future.

Scoring System for Planar Features

In the current design, three metrics are considered for the planar features. These metrics are the area ratio score, accessibility score and aspect ratio score. In the initial design of the scoring system, no weight was assigned to these scores. While studying test cases, it was realized that choosing a planar feature as a datum is more influenced by its area rather than its accessibility or aspect ratio. Therefore, a weight of ten was assigned to the area ratio score after doing calibrations for different case studies.

The planes with larger area stablish a proper datum for manufacturing and inspection (Figure 5 - 5). Therefore, they should have a better chance to be selected as datum features. Area ratio score is defined as the ratio of the feature's area to the area of the largest feature in the same part. Taking the ratio makes the score normalized in the range of zero to one-hundred. To have a comprehensive score that addresses all the datum selection rules associated with area, it should be checked that if the to be scored feature is mating or not. Based on the rule D3 the mating planar feature should be chosen over the non-mating one if its area is more than 60% of the non-mating feature's area. To cover this rule, the mating multiplier is introduced for the area ratio score. This multiplier enlarges the area ratio score by the scale of 1.67 if the to be score feature is mating. To prevent having scores more than one hundred due to multiplying mating feature's area by1.67, the largest area in the part is also multiplied by 1.67.

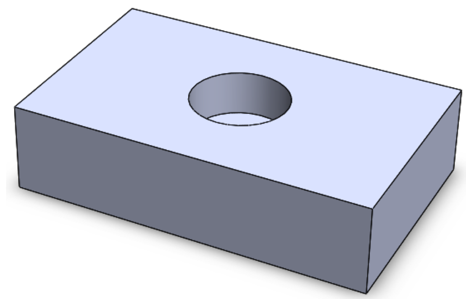


Figure 5 - 5: Top Face Is the Best Candidate for Controlling Hole' Orientation

Aspect ratio score is defined as the ratio of the plane's length to its width. Planes with aspect ratio more than eight are not good candidates to be datums since they don't stablish a good reference for target features [17]. Figure 5 - 6 shows two faces with the same area while one of them (face #2) has a large aspect ratio. Between these two candidates, the face with small aspect ratio (face #1) is selected as datum feature reference. It should be noted that if a part is 2D (one of the dimensions is small in comparison with the other two dimensions), aspect ratio can be neglected.

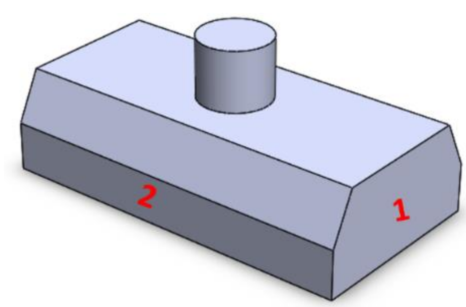


Figure 5 - 6: Face #1 Stablishes a Better Datum to Control Position of the Pin

There are three types of planes' shapes defined in this research: rectangular, circular and general. Calculating aspect ratio for rectangular and circular planes is straightforward. The aspect ratio of a circular plane is always one, therefore the aspect ratio score is one hundred since the aspect ratio is less than eight. The aspect ratio of a rectangular plane is obtained by dividing its length by width. If the aspect ratio of a rectangular plane is less than or equal to eight, aspect ratio score is considered as one hundred, otherwise, it is computed by equation 5. 1 which is developed heuristically.

$$S_1 = \left(\frac{8}{AR}\right) * 100 \qquad 5.\ 1$$

Where S_1 is the aspect ratio score for a rectangular plane and AR is the aspect ratio.

For the general planes, calculating aspect ratio score is a little complicated because their geometric parameters are defined by their bounding box due to tolerance analysis module requirements. A general plane might have a bounding box with aspect ratio less than eight while the plane itself occupies a small percentage of the bounding box. To address this issue, equation 5. 2 is developed after several trial and errors made on the critical value for the ratio of theactual area to the bounding box area. It should be noted that when the aspect ratio of the bounding box is less than or equal to eight and the ratio of the actual area to the bounding box area is greater than or equal to 0.5, the aspect ratio score of the feature is hundred, otherwise, the aspect ratio score of the feature is calculated by this equation.

$$S_2 = \left(\frac{8}{AR}\right) * \left(\frac{\frac{A_P}{A_B}}{0.5}\right) * 100 \qquad 5.\ 2$$

Where S_2 is the aspect ratio score for a general plane, and AR is the aspect ratio of the bounding box, A_P is the area of the plane and A_B is the area of the bounding box.

Accessibility defined in the GD&T rulesets is a general concept that needs more clarification in order to be used as a metric. One of the topological characteristics of the face that can represent the idea of the accessibility is the convexity condition of the edges. When a face has several concave edges, it means that face is not a good candidate to be chosen as datum regarding its low accessibility. Figure 5 - 7 illustrates two faces with the same area while one of them (face #2) has one concave edge and as a result low accessibility. Between these two faces, its better two choose face #1 to control translational variations of the hole.

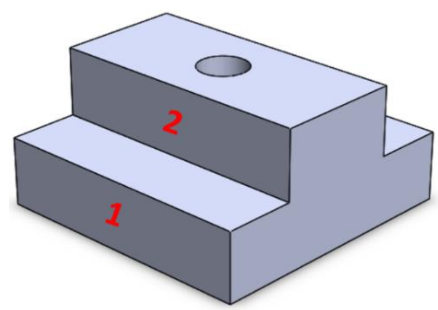


Figure 5 - 7: Face #1 Stablishes a Better Datum to Control Position of the Hole

To represents this idea, equation 5. 3 is developed for computing accessibility score. It should be noted that when all the edges of the face are concave, this equation is not used and the accessibility score is zero.

$$S_3 = 100 - \left(\frac{N_C}{N_T - N_C}\right) * 10 \qquad 5.3$$

Where S_3 is the accessibility score, N_C is number of concave and N_T is total number of edges.

Scoring System for Prismatic FOS

In the GD&T standards, the prismatic FOSs are represented by their mid-planes when they are treated as target features or datums. Hence, in datum selection process, prismatic FOSs and planar features are ranked and grouped in one category since the mid-planes of the prismatic FOSs and planar features share the same geometric and topological characteristics. A prismatic FOS has the same types of scores as a planar feature although the method of calculating accessibility score is different.

The area ratio score for a prismatic FOS is calculated by taking the ratio of the mid-plane's area to the largest area in the same part. If the prismatic FOS is mating, the mating condition multiplier enlarges the area ratio score of the FOS.

The aspect ratio of a prismatic FOS is computed as the same as it was explained for a planar feature; the ratio of the mid-plane's length to the mid-plane's width.

The accessibility score for a prismatic FOS is calculated based on feature type. If the feature is external, the accessibility score is hundred. If the feature is a tab, slot or pin, the accessibility score is ninety. Finally, if the feature is a hole, the accessibility score is eighty. As mentioned above, these metrics are developed based on heuristics and might be modified or removed by performing further case studies.

Scoring System for Cylindrical FOS

There are two types of scores defined for a cylindrical FOS: length ratio and aspect ratio. The length ratio score is a normalized score in the range of zero to hundred that is computed by taking the ratio of the feature's height/depth to the height/depth of the largest cylindrical feature in the same part. Cylindrical features with larger length stablish datums that are good for manufacturing and inspection. Figure 5 - 8 presents two cylindrical pin candidates for positioning a hole. The pin with larger length (pin #1) is a better choice over the pin with smaller length (pin #2).

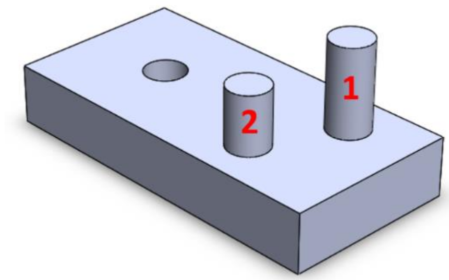


Figure 5 - 8: Pin #1 Stablishes a Better Datum due to Its Length

The aspect ratio is defined by the ratio of the height/depth to the diameter. An aspect ratio less than or equal to four is considered as good ratio. Cylindrical features with aspect ratio more than four do not establish proper datums for manufacturing and inspection. Figure 5 - 9 shows two cylindrical candidates with same length for controlling the position of a hole. One of the pins (pin #2) has a large aspect ratio which makes it a bad candidate for being a datum feature.

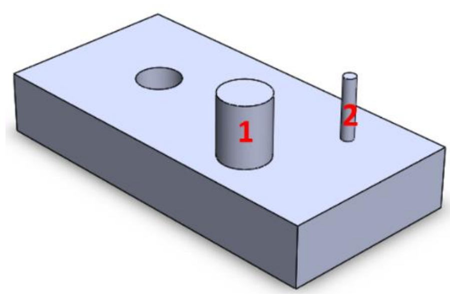


Figure 5 - 9: Pin #1 Stablishes a Better Datum due to Its Small Aspect Ratio

If the aspect ratio is more than four, the score is computed by the equation 5. 4.

$$S_4 = \left(\frac{4}{AR}\right) * 100 \qquad 5.\ 4$$

Where S_4 is the aspect ratio score for a cylindrical feature and AR is the aspect ratio.

5.5.3.4 Using the Concept of DOF Algebra for DRF Creation

DOF algebra is an important concept for developing a tolerance schema since it defines how far the datum selection process should go. Based on this concept, the datum selection process continues until all of the active DOFs of the to-be-toleranced feature are constrained by the datum feature references in the control frame. The background on the use of DOF algebra for DRF creation was discussed in chapter 2. In that chapter the active and invariant DOFs of the geometric primitive elements are discussed and illustrated. In this section the active and invariant DOFs of the real features such as cylindrical and prismatic features are presented.

Active and Invariant DOFs of a Cylindrical FOS

A cylindrical feature, has two active RDOFs around and two active TDOFs along x and y axes as shown in Figure 5 - 10.

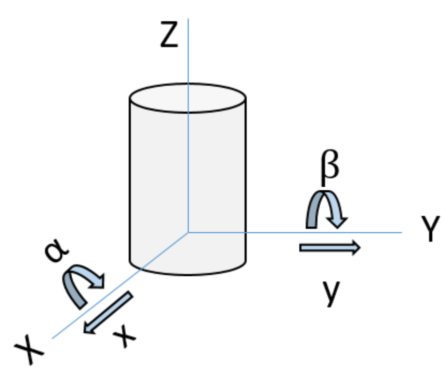


Figure 5 - 10: Active DOFs of a Cylindrical FOS

If the active DOFs of a cylindrical feature need to constrained by a datum feature reference, four different possibilities can happen depends on the type of the reference feature and its orientation with respect to the target feature. Table 5 - 1 lists these possibilities and the corresponding DOFs that each one can control. Before going to this table, here are the common notations and symbols to simplify the explanation: Let CY, PL and PR stand for cylindrical, planar and prismatic features respectively, AR and AT for the active RDOFs and TDOFs respectively, T and D for the target and datum (reference) features respectively, CR and CT for the number of constrained active RDOFs and TDOFs respectively, ⫽ for the parallelism and ⊥ for the perpendicularity relationships between datum and target features respectively.

T		D							
CY		CY				PL / PR			
		⊥		⫽		⊥		⫽	
AR	AT	CR	CT	CR	CT	CR	CT	CR	CT
2	2	1	1	2	2	2	0	1	1

Table 5 - 1: DOFs of a Cylindrical FOS Controlled by Different Datum Features

Active and Invariant DOFs of a Prismatic FOS

As it will be discussed in next chapter, prismatic FOSs are in two main types: tab/slot and pin/hole. The pin/hole feature is treated as two perpendicular tabs/slots in this research, therefore, the active DOFs are only discussed for the tab/slot feature. The tab/slot feature is a special case of plane primitive. It has two active RDOFs around x and z axes and one active TDOF along y axis as it illustrated in Figure 5 - 11.

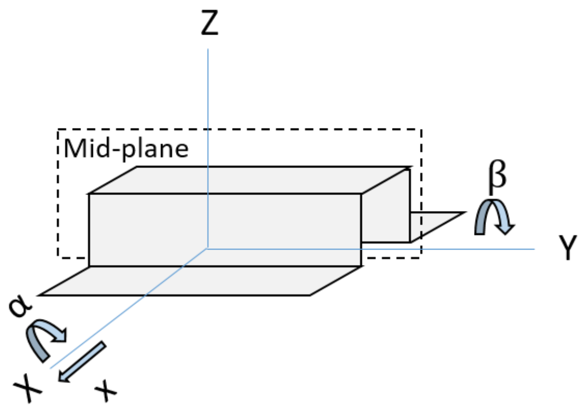


Figure 5 - 11: Active DOFs of a Prismatic FOS

The active DOFs of a prismatic FOS and the DOFs that are controlled by each type of datum feature is presented in Table 5 - 2.

T		D							
PR		CY				PL / PR			
		⊥		//		⊥		//	
AR	AT	NR	NT	NR	NT	NR	NT	NR	NT
2	1	1	0	1	1	1	0	2	1

Table 5 - 2: DOFs of a Prismatic FOS Controlled by Different Datum Features

5.3.3.5 Datum Selection and DRF Creation

As discussed in earlier, the combination of the datum feature references in location control frame should constrains all active DOFs of the target feature. In order to decide when to stop datum selection process, restricted DOFs of the target should be calculated after choosing each datum and be compared to its total active DOFs. Once the DRF is created for the target feature, location tolerance frame is generated. This procedure is illustrated through the following example.

Positioning a Hole: An Example

Figure 5 - 12 shows a simple cubic body with a hole in it. To position this hole, the potential datum features should be scored and ranked first.

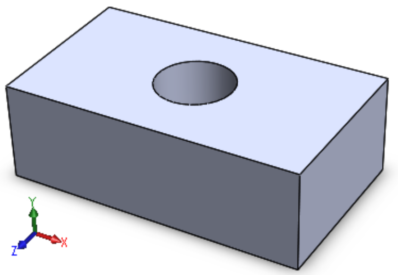


Figure 5 - 12: A Cuboid with a Blind Hole in It

Suppose in this part, top plane where the hole seats, bottom plane, the external FOS in z direction and external feature in x direction are ranked from one to four. The top plane is chosen as the primary datum for the pin since it has the highest rank. Regarding Table 5 - 1, this plane controls two RDOFs of the hole around x and z axes. Based on the GD&T rule D9, the secondary datum should be perpendicular to the primary datum. Hence, external FOS in z direction is selected as secondary datum. Although this feature can constrain one RDOF and one TDOF of the pin based on Table 5 - 1, since both RDOFs of the pin has been already restricted by the primary datum, the secondary datum constrains only one TDOF in z direction. The tertiary datum is the external FOS in x direction which is perpendicular to both primary and secondary datums. The tertiary datum constrains the remaining TDOF in x direction. With the combination of these datum features, all active DOFs of the pin are controlled and a position tolerance can be assigned to it. This example was only for understanding the DOF algebra concept and its role in datum selection without going into the details of feature scoring process. In chapter 7, more real world test cases will be studied in detail.

5.5.4 Creating Size Tolerance Frames

Size tolerance is applied to a FOS ant it controls FOS's size and shape variations (size and shape DOFs). Although form is independent of size, its limits are defined size tolerance under the rule #1 of the ASME Y14.5 standard[5] . Therefore, size tolerance only constrains the zone where size and form can vary. In order to satisfy 1st order tolerancing, size tolerance should only be assigned to the mating FOSs and to the non-mating FOSs which are chosen as datum features. Each FOS has different number of size DOFs to be controlled. Table 5 - 3 lists number of size DOFs for each type of feature. Beside Following notations are used in this table to simplify the explanation: SDOFs stands for the size DOFs, P for a pin, H for a hole, BH for a blind hole, TH for through hole, T for a tab, S for a slot, HT for the height DOF, DR for the diameter DOF, DH for the depth DOF, WH for the width DOF, and LH for the length DOF.

Feature Type	CY			PR			
	P	H		T/S	P/H		
					P	H	
		TH	BH			TH	BH
Size DOFs	1 DR 1 HT	1 DR	1 DR 1 DH	1 HT 1 WH	1 WH 1 LH 1 HT	1 WH 1 LH	1 WH 1 LH 1 DH

Table 5 - 3: Size DOFs of Different Types of Features

5.5.5 Creating Orientation and Form Tolerance Frames

According the ASME Y14.5 rule #1, variations in the orientation and form of a FOS can be controlled by the limits defined by size tolerance. On the other hand, the tolerance zone created by the location tolerance can determine the allowable orientation and form variations and the tolerance zone defined by the orientation tolerance controls the form variations of the feature. Therefore, orientation and form tolerances are applied to the target feature when a finer tolerance zone other than the zones defined by the preceding tolerances is required or when there is no preceding tolerance assigned to the feature. In the current scope of the tolerance schema generation module, orientation and form tolerances are applied to datum features and mating FOSs. Although in many cases refining the tolerance zone for a feature that has low sensitivity in a tolerance loop is not needed, assigning tolerance refinements (orientation, form) cannot be avoided since the sensitivity information is not available during the schema generation process. The extraction of the sensitivity information requires initial tolerance value allocation and is done in a downstream module. Therefore, unrequired orientation and form tolerances can be filtered in the tolerance value allocation and analysis module regarding the sensitivity data.

5.6 Communicating The Recommended GD&T Scheme

The idea of having ASU Global Model as a neutral platform for representing and communicating the GD&T data was discussed earlier in this chapter. Once the GD&T schema is produced by the schema generation module, it should be exported through a CTF file to the downstream modules. Except the sections A and E of the CTF file that contain directory of the file and the assembly tree information, remaining sections contain recommended GD&T information. In the section B of the CTF file, the information of the features that are participating in the GD&T schema is communicated part by part. Features are presented by their geometric and topological properties such as direction vector, base point, underlying face, and so on. Figure 5 - 13 presents the feature section of a CTF file.

```
#3=RECTANGULAR_PLANE('(FACE8)_part1', (48.1243, 5.06365, 10.2133), [-1, 0, 0], 66.67
#4=SLOT('CTP(FACE17&FACE19)_part1', (88.7643, 5.06365, 10.2133), [-1, 0, 0], 3.302,
#5=GENERAL_MIDPLANE('(FACE12&FACE4)_part1', (88.572, 5.06365, 10.2133), [0, -1, 0],
#6=RECTANGULAR_PLANE('(FACE18)_part1', (88.7643, 5.06365, 10.2133), [1, 0, 0], 12.7,
#7=PATTERN_OF_HOLES(2'(FACE23&FACE22)_part1', [-1, 5.20417e-017, 0], 2.6543, 9.525);
#8=HOLE('(FACE23)_part1', (48.1243, -23.5114, 10.2133), [-1, 0, 0], 2.6543, 9.525);
#9=HOLE('(FACE22)_part1', (48.1243, 33.6386, 10.2133), [-1, 0, 0], 2.6543, 9.525);
```

Figure 5 - 13: Feature Section (B) of a sample CTF file

Section C serves a platform for transferring constraints in the GD&T schema. All constraints point to the target features that the constraint has been defined between. If the constraint is distance or angularity, the basic dimension is also printed as constraint value. The representation of constraint in a sample CTF file is shown in Figure 5 - 14.

```
#68=CST_COINCIDENT(0, #8 ,#2);
#69=METRIC_RELATIONSHIP(#68, CST_COINCIDENT, (0, #8[LINE(axis of HOLE)], #2[MIDPLANE]))
#70=CST_COINCIDENT(0, #9 ,#2);
#71=METRIC_RELATIONSHIP(#70, CST_COINCIDENT, (0, #9[LINE(axis of HOLE)], #2[MIDPLANE]))
#72=CST_DISTANCE(28.575, #8 ,#5);
#73=METRIC_RELATIONSHIP(#72, CST_DISTANCE, (28.575, #8[LINE(axis of HOLE)], #5[MIDPLANE]))
#74=CST_DISTANCE(28.575, #9 ,#5);
#75=METRIC_RELATIONSHIP(#74, CST_DISTANCE, (28.575, #9[LINE(axis of HOLE)], #5[MIDPLANE]))
```

Figure 5 - 14: Constraint Section (C) of a sample CTF file

Finally, section D contains all the tolerances information in the assembly. Each tolerance type is represented by the target feature, target feature material modifier, tolerance zone modifier, DRF and datum features material modifiers is they are applicable. It should be noted that all tolerance values are zero and it is the task of tolerance value allocation/verification module to allocate tolerance values. Figure 5 - 15 illustrates the tolerance information in a sample CTF file.

```
#252=T_SIZE(#5, (nFI, 0, RFS));
#253=DOF(#252, (SIZE_DOF, SHAPE_DOF));
#254=T_POSITION(#7, (FI, 0, MMC)), PD(#2, MMC), SD(#3, NONE), TD(#5, MMC));
#255=DOF;
#256=T_POSITION(#10, (FI, 0, MMC)), PD(#2, MMC), SD(#3, NONE), TD(#5, MMC));
#257=DOF;
#258=T_POSITION(#4, (FI, 0, MMC)), PD(#2, MMC));
#259=DOF;
#260=T_PERPENDICULARITY(#3, (FI, 0,NONE)), PD(#2, MMC));
```

Figure 5 - 15: Tolerance Section (D) of a sample CTF file

CHAPTER 6

FIRST ORDER GD&T SCHEMA GENERATION: IMPLEMENTATION

This chapter explains the process and challenges of implementing the schema generation module. In section 6.1, the development environment, required external packages and libraries and programming language are introduced. Section 6.2 covers the data structures used for feature representation, constraint representation and tolerance representation. In this section, the issues and limitations associated with current data structures implementation are discussed. Section 6.3 illustrates the implementation of the tolerance frames generation by the help of relevant pseudo codes. Finally, the implementation of CTF creation is presented in section 6.4.

6.1 Environment: Software and Hardware

The GD&T schema generation module has been developed on 64-bit Microsoft Windows 7®[3] operating system on a PC. The PC hardware that this module has been developed on has 4GB of RAM and 3.0 GHz CPU. The programming language and IDE used are C++ and Microsoft Visual Studio®[3] 2012 respectively. The external library used for this project is commercial geometric kernel ACIS®[4] R25.

6.2 Data Structures

In order to maintain the GD&T information that are compatible with ASU GD&T Global Model and ASME Y14.5 standards [5], different classes are designed for representing features, constraints and tolerances with the help of object-oriented capability of C++.

---

[3] Windows 7, Visual Studio are all registered trademarks of the Microsoft Corporation.
[4] ACIS and SAT are registered trademark of the Spatial Technologies Corporation.

6.2.1 Feature Representation

In the current implementation, all geometric features (planar, prismatic, and cylindrical) are represented by a single class structure. Although current implementation hold all the characteristics required by ASU GD&T Global Model, it makes the module hard to develop and maintain. Designing a base geometric feature class and deriving all types of feature from the base class makes the code neat and reusable for further researches. Due to this requirement, a new data structure and hierarchy for geometric features is proposed in chapter under future works. The proposed data structure holds a semantic relationship between different types of feature and it is easy to implement in near future.

The current class structure contains the topological and geometric properties of the features. The topological properties include underlying faces and features types. On the other hand, geometric properties contain geomatics shape of the feature, radius, width, direction vector, etc. Beside the data members that represents these topological and geometric properties, features class have other data members that represent scoring system information for datum selection. Figure 6 - 1 shows the feature class data structure and its data members and functions. Some of data members have different meaning for each type of feature. For example, first member in the geometric parameters vector holds area for planar feature or radius for cylindrical FOS. This is one of deficiencies of the current implementation that requires the developer to be aware of data members that have different meaning for each type of feature.

```
// This is feature class.

Class Feature
{
        private:
        // data members
        vector<int> faceIDs;
        ENTITY_LIST faceList;
        geom_enum_type geometryType;
        feat_enum_type featureType;
        SPAunitvector direction;
        SPAposition basePoint;
        vector<double> geometricParameters;
        int overllScore;
        ...

        public:
        // set functions
        void set_face_IDs (vector<int> FaceIDs);
        void set_overall_score (int OverallScore);
        ....
        // get functions
        SPAunitvector get_direction_vector();
        ....
};
```

Figure 6 - 1: Feature Class

6.2.2 Constraint Representation

Constraints are presented in the section C of the CTF file. Constraint is divided into two main types: basic dimensions and mating constraint. There are nine subtypes of constraints used in this projects: distance, concentricity, coincidence, angularity, parallelism, perpendicularity, against, float and press fit. First six constraints belong to basic dimension category while last three belong to mating constraint. In the current implementation, two main classes for these nine constraints: class of basic dimension and class of mating constraints. This data structure also needs to be redesigned. In the new design, two main types of constraints are inherited from the constraint parent abstract class and subtypes are inherited from main types. This design will be explained in next chapter for future improvements. Figure 6 - 2 and Figure 6 - 3 shows the current data structures for basic dimension and mating constraint classes respectively.

```
// This is basic dimension constraint class.

Class BasicDimensionConstraint
{
        private:
        // data members
        Feature * feature1;
        Feature * feature2;
        constr_enum_type constraintType;
        float basicDimensionValue;

        public:
        // set functions
        void set_feature2 (Feature * Feature2);
        ....
        // get functions
        float get_value();
        ....
};
```

Figure 6 - 2: Basic Dimension Constraint Class

```
// This is mating constraint class.

Class MatingConstraint
{
        private:
        // data members
        Feature * feature1;
        Feature * feature2;
        constr_enum_type constraintType;

        public:
        // set functions
        void set_feature1 (Feature * Feature1);
        ....
        // get functions
        constr_enum_type get_type();
        ....
};
```

Figure 6 - 3: Mating Constraint Class

6.2.3 Tolerance Representation

The tolerance information is represented in section D of the CTF file format. Tolerance information completes the recommended GD&T for downstream modules. There are four types of tolerances used in this project: size, location, orientation and form.

In the current implementation, each major tolerance type has its own class. Since there are some data members and functions that are common between all tolerance types, it is better to redesign the tolerance data structure in a way that four major tolerances (size, location, orientation, and form) get inherited from parent tolerance class. Also tolerance subclasses such as concentricity, flatness, etc. get derived from four major tolerance classes. A new data structure and hierarchy for representing tolerances is suggested in the next chapter. Figure 6 - 4 to Figure 6 - 7 show the class design of four major tolerance classes used in this project.

```
// This is location tolerance class.

Class LocationTolerance
{
        private:
        // data members
        feature * target;
        vector<Feature*> datums;
        tol_enum_type tolType;
        mat_mod_enum_type targetModifier;
        ....

        public:
        // set functions
        void set_target (Feature * Target);
        ....
        // get functions
        tol_enum_type get_tol_type();
        ....
};
```

Figure 6 - 4: Location Tolerance Class

```
// This is size tolerance class.

Class SizeTolerance
{
        private:
        // data members
        feature * feature1;
        feature * feature2;
        tol_enum_type tolType;
        ....

        public:
        // set functions
        void set_feature1(Feature * Feature1);
        ....
        // get functions
        tol_enum_type get_tol_type();
        ....
};
```

Figure 6 - 5: Size Tolerance Class

```
// This is orientation tolerance class.

Class OrientationTolerance
{
        private:
        // data members
        feature * target;
        tol_enum_type tolType;
        vector<features*> datums
        Vector<mat_mod_enum_type> datumModifiers
        ....

        public:
        // set functions
        void set_datums(vector<feature*> datums);
        ....
        // get functions
        tol_enum_type get_tol_type ();
        ....
};
```

Figure 6 - 6: Orientation Tolerance Class

```
// This is form tolerance class.

Class FormTolerance
{
        private:
        // data members
        feature * target;
        tol_enum_type tolType;
        mat_mod_enum_type targetModifier
        ....

        public:
        // set functions
        void set_target(feature * Target);
        ....
        // get functions
        mat_mod_enum_type get_modifier ();
        ....
};
```

Figure 6 - 7: Form Tolerance Class

6.3 Tolerance Schema Generation

In this section, the implementation of the 1st GD&T schema generation functions is presented. This section starts with illustrating the functions used for reading input files information. Then, the process of creation location tolerance frames is shown through relevant data structures and pseudo codes. Next, the implementation of the functions for generating size tolerance frames is presented. Finally, pseudo codes and data structures for setting up orientation and form tolerance frames are shown respectively.

6.3.1 Reading Input Files and Populating Data Structures

Input files from AFR and PFR provide the assembly and pattern features information as well as mating conditions to the schema generation module. Input CAD file provides the underlying topology and geometry of the given assembly in the SAT[4] file format. The order of reading input file is as follow: first read the CAD file, then read ".pfr" and ".fts" files respectively. Figure 6 - 8 shows the pseudo code for reading input files and populating feature and mating constraint data structures.

```
// Start
Read assembly CAD file (.SAT)
Read pattern features information (.pfr)
Store pattern information in a temporary data structure
Read part features information (.fts)
For each part in the assembly
        Store corresponding features in the feature data structure
        if feature is in pattern
                Label it as pattern feature
For each assembly feature
        Store mating constraint
// End
```

Figure 6 - 8: Pseudo Code for Reading Input Files

6.3.2 Enhancing the Input Data

To enhance the input data, modified DoCs should be extracted, non-mating features should be recognized and coplanar features need to be identified.

6.3.2.1 Extracting Modified Directions of Control

It should be noted that DoC works in part level so for given assembly, extraction of DoC is performed part by part. The implementation of DoC has been explained in details by Mohan et al [25]. Figure 6 - 9 illustrates a high level implementation of the modified DoC pseudo code.

```
// Start
Read mating features list
Extract unique features' direction vectors
Read face list of current body
If the face is planar
        If face's normal is parallel or perpendicular to any of feature's direction vectors
                If face's  normal direction vector is already in DoC list
                        Add the face at the end of that direction's face list
                Else
                        Create a new direction and put that face as first member of that list
                End if
        End if
Read face list of current body
If face is cylindrical
        If face's axis is parallel or perpendicular to any of mating features' direction vector
                If face's axis direction is perpendicular to any of DoC list members
                        Add the face at the end of that direction's face list
                End if
        End if
End if
Read DoC list
Read each DoC face list
Sort faces based on their relative distance to extreme face in that direction
// End
```

Figure 6 - 9: Pseudo Code for the Modified DoC Function

6.3.2.2 Finding Non-Mating Features

The implementation of non-mating feature recognition function requires three nested loops. First loop traverses all the directions in the DoC list. For each direction in the DoC list, the second loop traverses all faces associated with that direction. The third nested loop is designed for finding tabs/slots. When a planar face is reached in the second loop its characteristics such as shape, normal vector and area are extracted. Then, third loop traverses the remaining faces in the same direction. If a planar face is reached in the third loop, its shape, normal and area are extracted. If the characteristics of first and second planes satisfy the tab/slot conditions, a prismatic FOS will be created. The pseudo code for finding non-mating features is shown in Figure 6 - 10.

```
// Start
For each direction in DoC list
        For each face in the direction face list
                if a planar face is reached
                        Get shape, area and normal vector of the face
                        For the rest of the faces in the same direction
                                If a planar face is reached
                                        Get shape, area and normal vector of the face
                                        If both faces have same shape, area and they are antiparallel
                                                Create a prismatic FOS and add it to features list
                                        Else
                                                Continue the third loop
                                        End if
                                End if
                        End for
                End if
        End for
        For each face in the direction face list
                If a planar face is reached
                        If it is not part of a prismatic FOS or hasn't already been detected
                                Create planar feature and add it to features list
                        End if
                End if
        End for
// End
```

Figure 6 - 10: Pseudo Code for the Non-Mating Features Recognition Function

6.3.2.3 Finding Coplanar Features

Recognizing the coplanar feature is a two-stage function. Each direction in DoC list contains a list of faces and their distances from extreme face in the same direction. To find potential coplanar faces, relative distances between faces need to be extracted. Then planar faces that have zero relative distance with respect to each other are clustered together. During the second stage - where a coplanar feature is created from the group of coplanar faces- if corresponding planar features are already stored separately in the features list, they group together as a new feature and geometric parameters of the new feature such as centroid and length are updated. If individual planar features are not in the features list, a new feature is created and added to the features list.

Preprocessing function includes two nested "*for*" loops for finding potential co-planes. On the other hand, post-processing function has only one "*for*" loop. Figure 6 - 11 presents the pseudo code of finding coplanar features function.

```
// Start of the preprocessing stage
For each direction in the DoC list
Calculate relative distances from absolute distance list
For each face in the direction face list
        If face is planar
                Add this face to a list for potential co-planes
                For the rest of the faces in the same direction
                        If face is planar, relative distance is 0 and it doesn't belong to any prismatic FOS
                                Add this face to potential co-planes list
                        Else
                                Clear the potential co-planes list
                        End if
                End for
        End if
End for
// End of the preprocessing stage
// Start of the post-processing stage
For each set of potential co-planes in the co-planes list
        If corresponding feature is already in features list
                Combine features as on feature and update the geometric parameters
        Else
                Create a new coplanar feature and add it to features list
        End if
End for
// End of the post-processing stage
```

Figure 6 - 11: Pseudo Code for Coplanar Features Recognition Function

6.3.3 Creating Location Tolerance Frames

As discussed in previous chapter, creating location tolerance frames includes three steps: Scoring potential datum features and then ranking them by their overall scores, selecting DRF for each mating FOS from highest ranked features and finally setting up the location tolerance frames.

Once location tolerance frames are generated, basic dimension between target features and their datums are extracted. These basic dimensions are stored in the constraint class data structure. The pseudo code for constructing location tolerance frames is presented in Figure 6 - 12.

```
// Start Scoring Feature
For each feature in the part
        If feature is planar or prismatic
                Score its area ratio, aspect ratio and accessibility
        Else
                Score its length ratio and aspect ratio
        Calculate overall score for each feature
End For
Rank features based on their types and overall scores
Sort features based on their overall score using merge sorting
// End Scoring Features

// Start Selecting DRFs and Creating Tolerance Frames
For each mating FOS in the part
        Select highest ranked non-cylindrical feature as potential primary datum
        Select highest ranked cylindrical feature as potential primary datum
        Compare two candidates and choose the best one as primary datum
        If all active DOF of the target feature has been restricted by primary datum
                Create location tolerance frame and extract basic dimension

        Select second highest ranked non-cylindrical feature perpendicular to primary datum as potential secondary datum
        Select second highest ranked cylindrical feature perpendicular to primary datum as potential secondary datum
        Compare two candidates and choose the best one as primary datum
        If all active DOF of the target feature has been restricted by primary and secondary datums
                Create location tolerance frame and extract basic dimensions
        Select third highest ranked non-cylindrical feature perpendicular primary and secondary datums as potential tertiary datum
        Select third highest ranked cylindrical feature perpendicular primary and secondary datums as potential tertiary datum
        Compare two candidates and choose the best one as primary datum
        If all active DOF of the target feature has been restricted by primary, secondary and tertiary datums
                Create location tolerance frame and extract basic dimensions
        Else
                Go to the next target feature from the list (Don't create location tolerance frame)
End For
// End Selecting DRFs and Creating Tolerance Frames
```

Figure 6 - 12: Pseudo code for Creating Location Tolerance Frames

6.3.4 Creating Size Tolerance Frames

As it explained in chapter 5, there are two types of size tolerance: size dimension and location dimension. Size dimension is assigned to control size variations of a FOS and location dimension is applied between two planes to control the location variations. Each FOS has different number of size dimension(s) and location dimension(s) to be controlled. For example, a pin needs one size control for its diameter and one location control for its length (location dimension between pin's top plane and the plane where pin is seated). On the other hand, a prismatic blind hole needs two size control for its width and length and one location control for its depth. Hence, based on the feature type, different number of tolerance frames is generated for different types of features. Figure 6 - 13 shows the process of creating size tolerance frames through a pseudo code.

```
// Start
For each feature in the part
    If feature is mating FOS or non-mating FOS that has been selected as datum
        Create size dimension tolerance frame(s) as many as size(s) need to be control for target feature
        Create location dimension frame if feature has depth/height to be controlled
End for
```

Figure 6 - 13: Pseudo Code for Creating Size Tolerance Frames

6.3.5 Creating Orientation Tolerance Frames

As it discussed in previous chapter, orientation tolerance frames are created for datums and mating FOSs. As it presented in Figure 6 - 14, orientation tolerance is generated for datums and then for mating FOSs.

```
// Start Assigning Orientation Tolerance to Datums
For each datum in the part
    Create orientation tolerance frame(s) with respect to up to two preceding datums
End For
// End Assigning Orientation Tolerance to Datums

// Start Assigning Orientation Tolerance to Mating FOS
For each mating FOS in the part
    For each datum in the target feature DRF list (up to two datums)
    If target is parallel to its datum
        Create parallelism orientation tolerance frame
    Else if target is perpendicular to its datum
        Create perpendicularity orientation tolerance frame
    Else
        Create angularity orientation tolerance frame and extract basic dimension
End For
// End Assigning Orientation Tolerance to Mating FOS
```

Figure 6 - 14: Pseudo Code for Creating Orientation Tolerance Frames

6.3.6 Creating Form Tolerance Frames

Form tolerance frames are also created for datums and mating FOS. It should be noted that form tolerance frames like size tolerance frames do not have DRF. Hence, material modifier is only applied to the target feature. If the target feature is FOS, MMC material modifier is specified for target feature, otherwise no material modifier is considered. Two types of form tolerances are used in this project: flatness for planes and mid-planes and straightness for axes. Figure 6 - 15 shows the pseudo code for this function.

```
// Start Assigning Form Tolerance to Datums
For each datum in the part
        If feature is planar
                Create flatness tolerance frame
        Else if feature is prismatic
                Create flatness tolerance frame for its mid-plane
        Else
                Create straightness tolerance frame for its axis
End For
// End Assigning Form Tolerance to Datums

// Start Assigning Form Tolerance to Mating FOS
For each mating FOS in the part
        If target is prismatic
                Create flatness tolerance frame for its mid-plane
        Else
                Create straightness tolerance frame for its axis
End For
// End Assigning Form Tolerance to Mating FOS
```

Figure 6 - 15: Pseudo Code for Creating Form Tolerance Frames

6.4 Exporting The Recommended GD&T Scheme: Creating The CTF File

Generating output CTF file is started with outputting the input geometry name and directory. Then, features information is printed for each part with a heading line identifying corresponding part and number of features in that part. While features information is being printed, their line number are updated to be a reference for sections C and D where constraints and tolerances point to features. Constraints are printed in two steps: first mating constraints are printed for the whole assembly and then basic dimensions are printed part by part. In section D, tolerance information is outputted part by part. For each part, size, location, orientation, and form tolerances are printed respectively. Once the whole recommended GD&T information is printed, the assembly hierarchy is printed as the last line of the CTF file. Figure 6 - 16 presents the process of creating output CTF file through a pseudo code.

```
// Start
Print input geometry name and directory
For each part in the assembly
        Print part number and line numbers of its features
        For each feature in that part
                Print feature information
        End For
End For
For each mating constraints in the assembly
        Print constraint information with pointers to the corresponding features
End For
For each part in the assembly
        For each basic dimension constraint in that part
                Print constraint information with pointers to the to the corresponding features

        End For
End For
For each part in the assembly
        For each size tolerance in that part
                Print tolerance information with pointers to the to the corresponding features
        End For
        For each location tolerance in that part
                Print tolerance information with pointers to the to the corresponding features
        End For
        For each orientation tolerance in that part
                Print tolerance information with pointers to the to the corresponding features
        End For
        For each form form in that part
                Print tolerance information with pointers to the to the corresponding features
        End For
End For
Print assembly hierarchy information
// End
```

Figure 6 - 16: Pseudo Code for Printing the CTF File

CHAPTER 7

CASE STUDIES AND SOFTWARE VERIFICATION

This chapter focuses on presenting the results of tolerance schema generation module for two different assemblies. For each assembly, the process of developing GD&T scheme for a sample part is shown step by step along with describing the rules behind developing this schema[5]. For other parts in each assembly, the final GD&T schema is presented since they obey almost the same logic. Finally, for each assembly the recommended GD&T created by this software is compared with the GD&T made by an expert and a novice.

7.1 Case Study 1: Cam Follower

The first test case is a Cam Follower assembly which has thirteen parts (Figure 7 - 1). This assembly was provided by our industry partner RECON SERVISES. For this assembly, the Right Support is taken as an example. The same procedure is applicable to the rest of the parts.

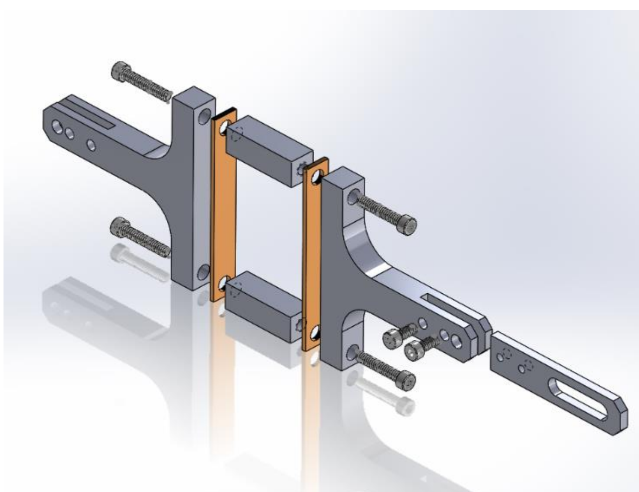


Figure 7 - 1: The Cam Follower Assembly

[5] The visualization of assembly and pattern features is done by ANSYS® Design Modeler. The visualization of the tolerances is done by SolidWorks® DimXpert.

7.1.1 Recommended GD&T by Schema Generation Module

Assembly Analysis Results

First, the assembly need to be analyzed by preprocessing modules. Running the assembly CAD file in STEP AP203 format through AFR and PFR modules, lead to the following results. There total number of 35 part-level and 23 assembly features detected by AFR in the assembly. Out of total part-level features in the assembly, there are eight features (six through holes, one slot and one plane) in the Right Support that are shown in Figure 7 - 2.

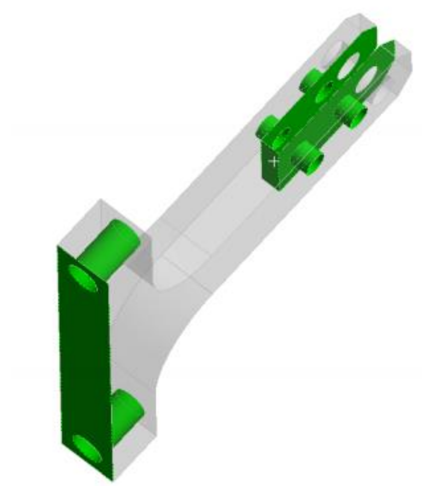


Figure 7 - 2: Part-Level AFR Results for the Right Support

The whole number of patterns detected by PFR in the assembly are five. Out of these six patterns, there are two patterns in this part: one linear pattern of two holes and one collinear-linear pattern of four holes. Both constructing mating chain and defining the new type of pattern (collinear-linear) that were made as modifications to the PFR module led to detecting the second pattern. The PFR results is visualized in Figure 7 - 3.

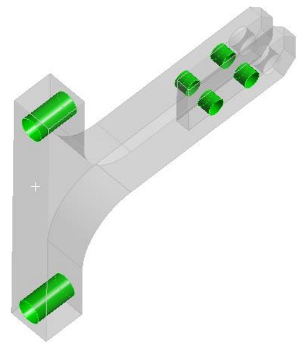


Figure 7 - 3: Patterns Detected by PFR

Finding Non-Mating and Coplanar Features

Once preprocessing analysis are done, the AFR and PFR outputs plus the CAD geometry in the SAT format is fed to GD&T schema generation module. These files populate the module's data structure with initial assembly information. Next step is enhancing the input data by running preprocessing functions (finding non-mating and coplanar features). Prior to that, part's DoC need to be extracted. Running this part through modified DoC function shows that there are three active DoC in this part. Figure 7 - 4 presents the extracted active DoC and their corresponding features for the Right Support.

```
#1(1i,0j,0k)---#2(0i,1j,0k)---#3(0i,0j,1k)
     |               |              |
   Plane           Plane          Plane
     |               |              |
   Plane           Axis           Plane
     |               |              |
   Plane           Plane          Axis
     |               |              |
   Plane           Axis           Axis
     |               |              |
   Axis            Axis           Plane
     |               |              |
   Axis            Axis           Plane
     |               |
   Axis            Axis
     |               |
   Axis            Axis
     |               |
   Axis            Axis
     |               |
   Axis            Plane
     |               |
   Plane           Axis
     |               |
   Plane           Plane
```

Figure 7 - 4: Extracted Active DoC for the Right Support

Non-mating features recognizer finds three external prismatic FOS in this part by traversing extracted DoCs (Figure 7 - 5). Coplanar feature recognizer also finds two coplanar features in this part (Figure 7 - 6).

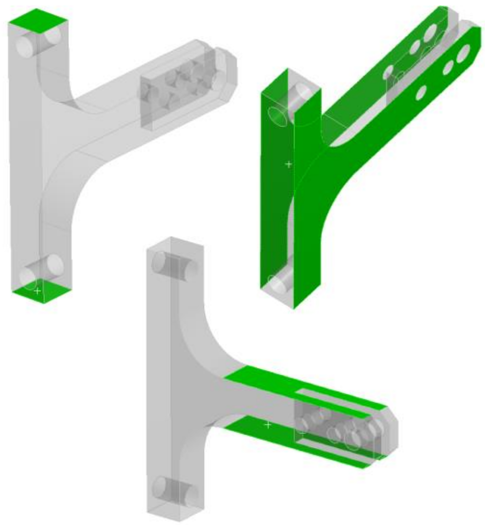


Figure 7 - 5: Three External Tab Features Detected in the Right Support

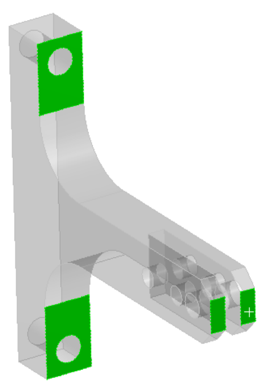


Figure 7 - 6: Two Coplanar Features Detected in the Right Support

Location Tolerance Frames

At this stage, all required information is gathered and the process of generating GD&T schema can start. Creating GD&T scheme starts with generating location tolerance frames for mating FOS. As mentioned above, there are seven FOS in this part in which six of them belong to two patterns. Due to rules L1 and L2, there will be three location tolerance frames: one for slot, one for two holes in the linear pattern and one for four holes in the collinear-linear pattern.

Before creating the location tolerance frame, all features are scored and ranked. As discussed in chapter 5, the scoring system assigns different scores to the different feature types, puts them in non-cylindrical and cylindrical features groups and then ranks them. Note that pattern features will not be scored and ranked since it is better not to choose them as datums (rule D13). Hence, there is no cylindrical candidate feature for this part. Table 7 - 1 shows the ranked features based on their category and overall score.

Type	Non-Cylindrical	Score	Cylindrical feature	Score
1	The external T-shape FOS	65		
2	The mating face where at the left end	52		
3	The mating slot	31		
4	The external U-shape FOS	27		
5	The coplanar faces where two holes seat	24		
6	The external square-shape FOS	19		
7	The coplanar faces at the right end	17		

Table 7 - 1: Features and Their Scores for the Right Support

To create a location tolerance frame for a FOS, a DRF should be created first. At each level of datum selection, two features are chosen if they available: one from cylindrical category and one from non-cylindrical category (either planar or prismatic). Then, the best candidate will be selected among these two features by applying the rule D8. As discussed in previous chapters, the process of datum selection continues till all the active DOFs of the target feature is constrained.

The mating slot has three active DOFs, two rotational and one translational. Best non-cylindrical candidate based the overall score is the external T-Shape FOS. There is no cylindrical candidate in this part therefore the external T-shape FOS is selected as primary datum. Since this datum feature is parallel to the slot, it constrains all three active DOFs of the target feature. As a result, the datum selection stops here and a symmetry tolerance frame is created for the slot. The basic dimension is also extracted between the datum and the target. Since the mid-planes of the both target and datum features are coincident, a coincidence constraint is created. Figure 7 - 7 illustrates the symmetry tolerance frame created for the slot.

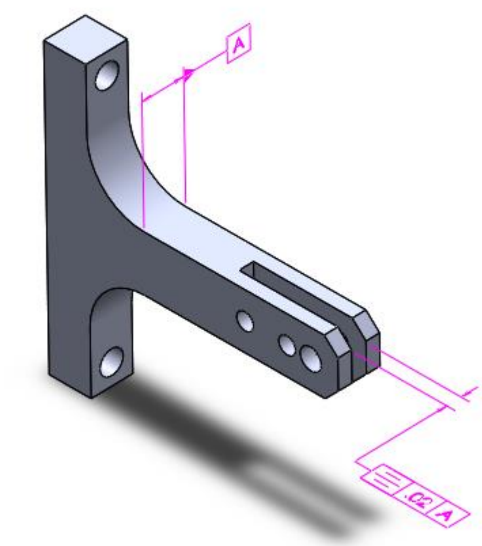


Figure 7 - 7: Symmetry Tolerance for the Slot

The procedure of DRF and position tolerance frame creation is the same for both patterns. Hence, this process is explained only for the pattern of two holes. The pattern of two holes has four active DOFs. Two translational and two rotational. The process of selecting primary datum is the same as explained for slot. Thus, the primary datum for this pattern is the external T-shape FOS. This datum only constrains one translational DOF and one rotational DOF of the target feature. Thus, the process of datum selection continues. To choose candidates for the secondary datum, the mating rectangular plane is selected as non-cylindrical candidate since it has the highest rank among non-cylindrical features that are perpendicular to the primary datum. Again there is no cylindrical feature to be selected as potential secondary datum and as a result, the planar feature is selected as secondary datum. The secondary datum constrains one rotational DOF of the target feature. The datum selection continues until the last translational DOF of the target feature is restricted. For tertiary datum, the non-cylindrical candidate is the external U-shape tab that is perpendicular to both primary and secondary datums. Since there is no cylindrical candidate, the external U-shape tab is chosen as tertiary datum. This datum constrains the last translational DOF. At this stage the position tolerance frame can be generated for pattern of two holes. Figure 7 - 8 illustrates the position tolerances assigned to both patterns. Once tolerance frames are constructed, basic dimensions are extracted.

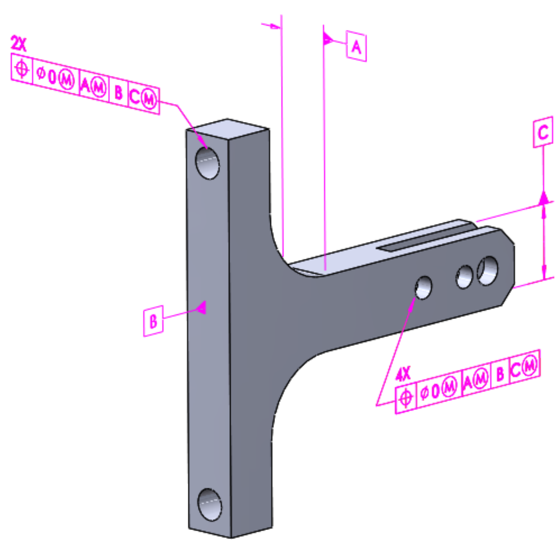


Figure 7 - 8: Position Tolerance Frames for Patterns

Size Tolerance Frames

As mentioned earlier, there are seven mating FOS in this part. The slot needs to size tolerance frames: one for controlling the variations of its width's size dimension and the other one for controlling the variations of its depth's location dimension. The pattern of two holes and pattern of four holes also require one size dimension tolerance each (rules S1 and S2). Beside these features, there are two external non-mating FOS that are used as datums. Thus, each of them needs one size dimension tolerance (rule S3). Therefore, finally there are six size tolerance frames in this part. (Figure 7 - 9).

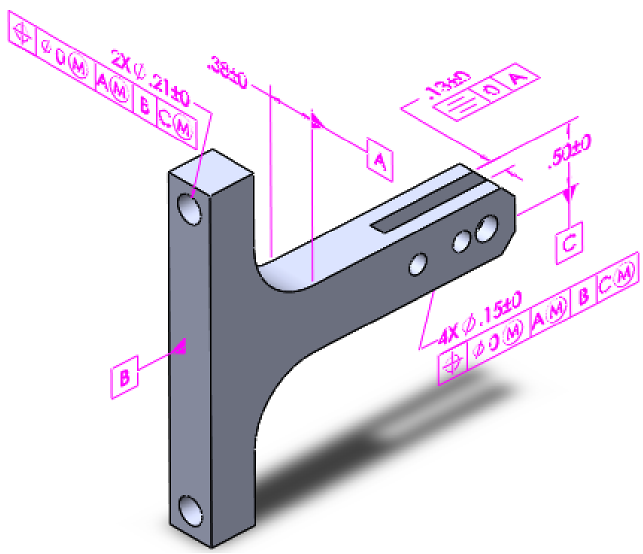


Figure 7 - 9: Right Support with Size Tolerance Frames Added

Orientation Tolerance Frames

Based on the rules X1 and X2, datums need orientation tolerance with respect to their preceding datum(s). Hence, the secondary datum (datum B) requires a perpendicularity tolerance with respect to the primary datum (datum A) and the tertiary datum needs a perpendicularity tolerance with respect to the primary and secondary datums. Beside orientation tolerance frames for datums, mating FOS also require orientation tolerance (rule X3). A parallelism tolerance frame is created for the slot with respect to the primary datum. Pattern of two holes have two tolerance frames: one parallelism frame with respect to the primary datum and one perpendicularity with respect to the secondary datum. Pattern of four holes also have two tolerance frames: one perpendicularity with respect to the primary datum and one parallelism with respect to the secondary datum. All seven orientation tolerance frames are shown in Figure 7 - 10.

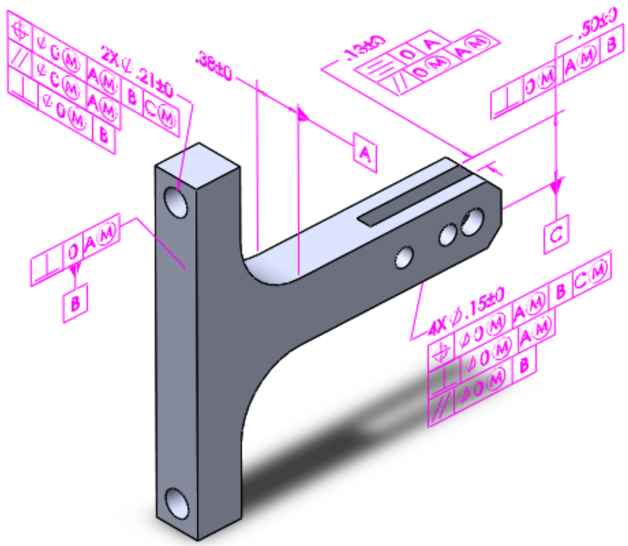


Figure 7 - 10: Right Support with Orientation Tolerance Frames Added

Form Tolerance Frames

Based on rules X7 and X8, total number of six form tolerance frames are required for this part. Three flatness tolerance frames for datums, one flatness tolerance frame for the slot and two straightness tolerance frames for the pattern of two holes and the pattern of four holes. The results for form tolerance frames is presented Figure 7 - 11.

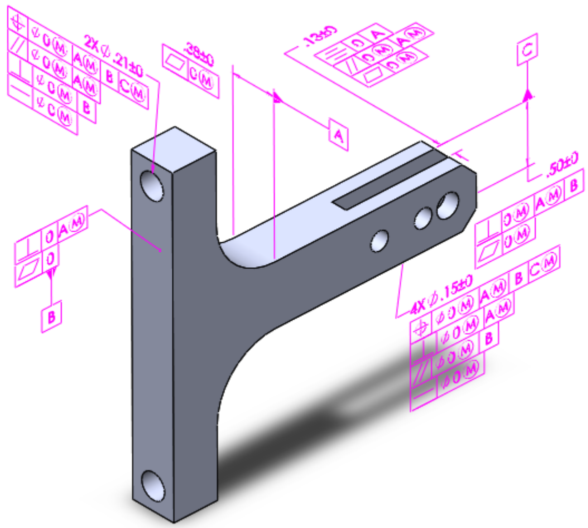


Figure 7 - 11: Right Support with Form Tolerance Frames Added

Creating GD&T Schema for The Remaining Parts in The Assembly

The same procedure mentioned above is applied for other parts in order to complete the GD&T schema for the whole assembly. In this assembly six parts are standard bolts which do not require tolerance scheme. Out of the remaining six parts, four parts are in pairs. Hence, Figure 7 - 12 presents the final GD&T for the remaining four distinct parts in the assembly. Figure 7 - 13 also shows a section of the skeleton CTF generated for Cam Follower.

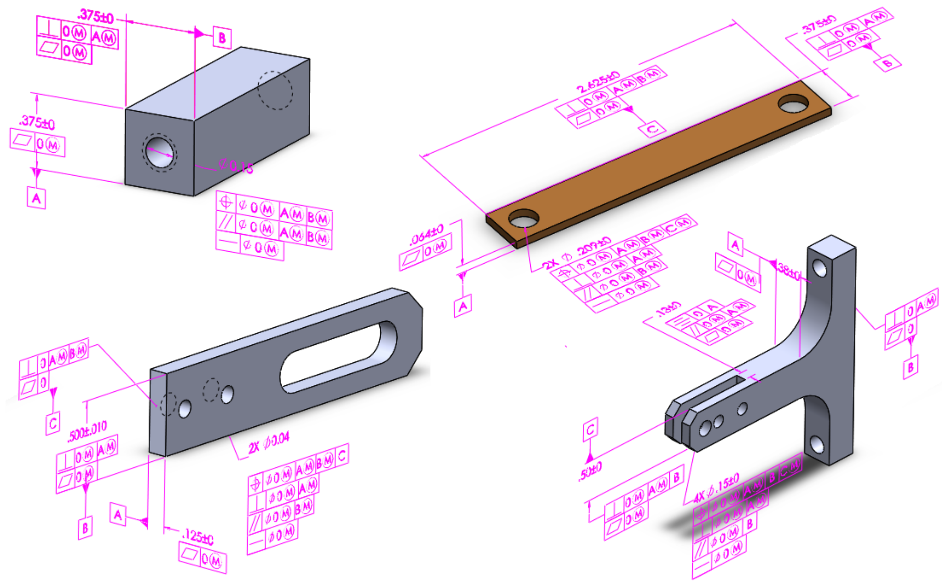


Figure 7 - 12: Complete GD&T Scheme for the Remaining Distinct Parts

```
#0=CamFollower.SAT
#1=PART('part1', #2, #3, #4, #5, #6, #7, #8, #9, #10, #11, #12, #13, #14, #15);
#2=GENERAL_MIDPLANE('(FACE1&FACE21)_part1', (68.6725, 5.06365, 10.2133), [0, 0, 1],
#3=RECTANGULAR_PLANE('(FACE8)_part1', (48.1243, 5.06365, 10.2133), [-1, 0, 0], 66.67
#4=SLOT('CTP(FACE17&FACE19)_part1', (88.7643, 5.06365, 10.2133), [-1, 0, 0], 3.302,
#5=GENERAL_MIDPLANE('(FACE12&FACE4)_part1', (88.572, 5.06365, 10.2133), [0, -1, 0],
#6=RECTANGULAR_PLANE('(FACE18)_part1', (88.7643, 5.06365, 10.2133), [1, 0, 0], 12.7,
#7=PATTERN_OF_HOLES(2'(FACE23&FACE22)_part1', [-1, 5.20417e-017, 0], 2.6543, 9.525);
#8=HOLE('(FACE23)_part1', (48.1243, -23.5114, 10.2133), [-1, 0, 0], 2.6543, 9.525);
#9=HOLE('(FACE22)_part1', (48.1243, 33.6386, 10.2133), [-1, 0, 0], 2.6543, 9.525);

#68=CST_COINCIDENT(0, #8 ,#2);
#69=METRIC_RELATIONSHIP(#68, CST_COINCIDENT, (0, #8[LINE(axis of HOLE)], #2[MIDPLANE]))
#70=CST_COINCIDENT(0, #9 ,#2);
#71=METRIC_RELATIONSHIP(#70, CST_COINCIDENT, (0, #9[LINE(axis of HOLE)], #2[MIDPLANE]))
#72=CST_DISTANCE(28.575, #8 ,#5);
#73=METRIC_RELATIONSHIP(#72, CST_DISTANCE, (28.575, #8[LINE(axis of HOLE)], #5[MIDPLANE]))
#74=CST_DISTANCE(28.575, #9 ,#5);
#75=METRIC_RELATIONSHIP(#74, CST_DISTANCE, (28.575, #9[LINE(axis of HOLE)], #5[MIDPLANE]))

#252=T_SIZE(#5, (nFI, 0, RFS));
#253=DOF(#252, (SIZE_DOF, SHAPE_DOF));
#254=T_POSITION(#7, (FI, 0, MMC)), PD(#2, MMC), SD(#3, NONE), TD(#5, MMC));
#255=DOF;
#256=T_POSITION(#10, (FI, 0, MMC)), PD(#2, MMC), SD(#3, NONE), TD(#5, MMC));
#257=DOF;
#258=T_POSITION(#4, (FI, 0, MMC)), PD(#2, MMC));
#259=DOF;
#260=T_PERPENDICULARITY(#3, (FI, 0,NONE)), PD(#2, MMC));
```

Figure 7 - 13: A Section of the Generated CTF for the Cam Follower Assembly

7.1.2 Recommended GD&T Manually Applied by an Expert

In this subsection, the manually applied GD&T on the Cam Follower assembly is presented. This GD&T was made by an expert from RECON SERVICES and it is shown in Figure 7 - 14.

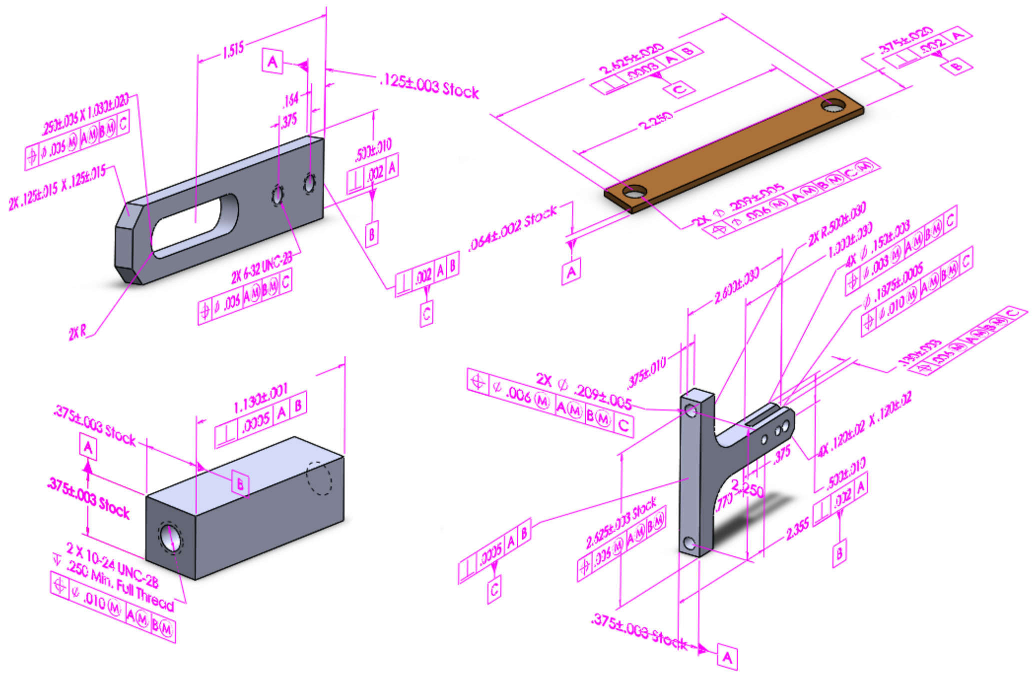


Figure 7 - 14: The Manually Applied GD&T by an Expert

7.1.3 Comparing GD&T Results

In this subsection, the GD&T schema generated by the schema generation module and the one that is applied manually are compared to find similarities and differences in location tolerances and datum selection. This comparison is made only for two parts the comparison results are presented in the corresponding tables.

The Tab

Figure 7 - 15 shows the size and location tolerance frames generated for the Tab: left presents the frames created by schema generation module and right illustrates the ones that applied manually.

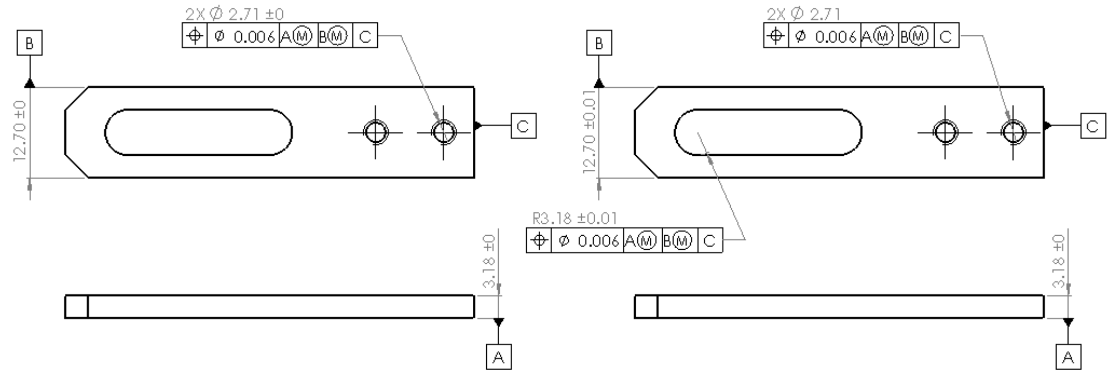


Figure 7 - 15: Size and Location Tolerance Frames Created for the Tab

Table 7 - 2 presents the similarities and differences between these two schemes for the Tab in choosing datums and creating DRFs.

Similarities	The datums for both GD&T schemas created manually and the one created by schema generation module are the same and also in the same order
Differences	There is an extra position tolerance frame for the rounded slot in the GD&T schema created manually. Since rounded slot is not a mating feature, there is no position tolerance frame in the GD&T schema created by schema generation module
Logic behind datum selection done by software	Datum A: This feature is mating and has largest are Datum B: This feature has the second largest length. Although this feature has a big aspect ratio (16), since this part is labeled as 2D, aspect ratio is not considered Datum C: This feature has the third largest area

Table 7 - 2: Similarities and Differences Between GD&T Schemes for the Tab

The Bar

Figure 7 - 16 shows the GD&T schema generated for the Bar: top is the schema created by generation module and bottom is the manually applied schema.

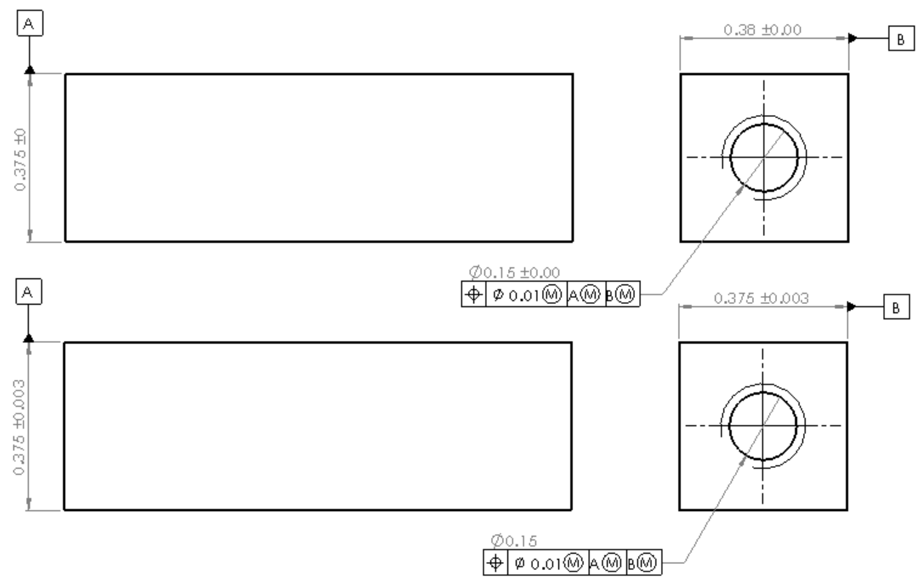


Figure 7 - 16: Size and Location Tolerance Frames Created for the Bar

Table 7 - 3 presents the similarities and differences between these two schemes for the Thin Plate in choosing datums and creating DRFs.

Similarities	The datums for both GD&T schemas created manually and the one created by schema generation module are the same and also in the same order
Differences	There is no difference between these two schemas
Logic behind datum selection done by software	Datum A: Although this feature is not mating, its area is large enough to be chosen over the external mating FOS Datum B: This feature has the same area as datum A and is large enough to be chosen over the external mating FOS Datum C is not required since all active DOFs of the holes are constrained by datums A and B

Table 7 - 3: Similarities and Differences Between GD&T Schemes for the Bar

7.2 Case Study 2: The Radio Car

Second case study is the Radio Car that includes nine parts (Figure 7 - 17). Out of these nine parts, four of them are standard pins. Hence, the GD&T is applied to five remaining parts. For this assembly, the Chassis is taken as an example. For the other four parts, the same procedure is applicable.

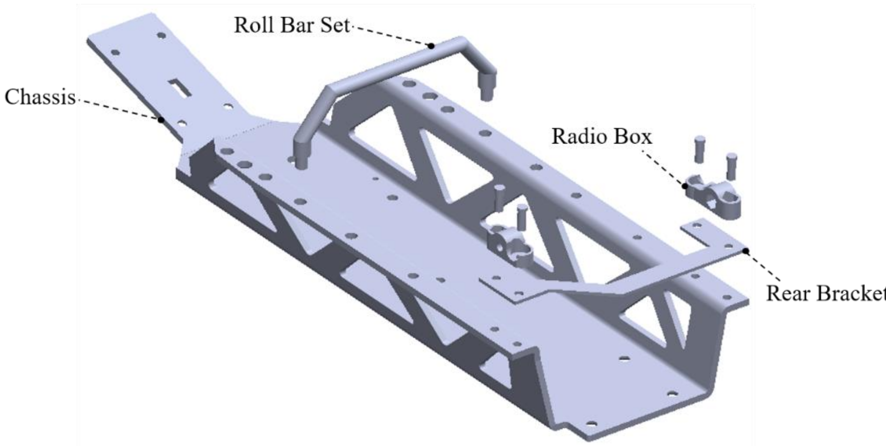


Figure 7 - 17: The Radio Car Assembly

7.2.1 Recommended GD&T by Schema Generation Module

Assembly Analysis Results

The preprocessing analysis by AFR and PFR modules on the assembly CAD model of the Radio Car in STEP AP203 format have the following results. There are thirty four part level mating and twenty four assembly features in this assembly. Eight mating features (six holes, two planes) belong to the Chassis (Figure 7 - 18).

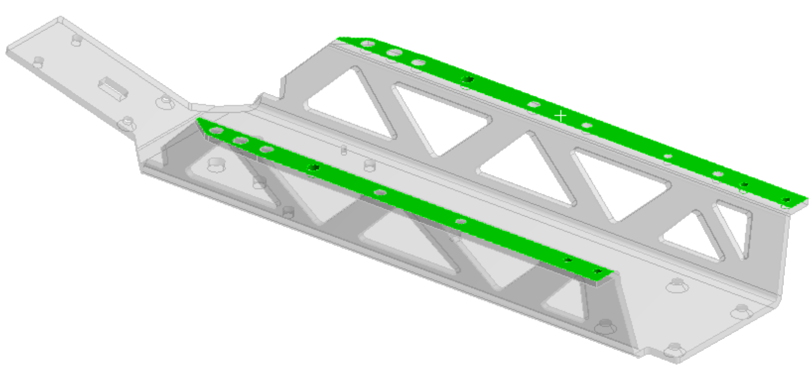


Figure 7 - 18: Mating Features Detected by AFR on the Chassis

PFR also recognizes three patterns in the assembly. Out of these three patterns, two patterns are in the Chassis: a rectangular pattern of four holes and a linear pattern of two holes (Figure 7 - 19).

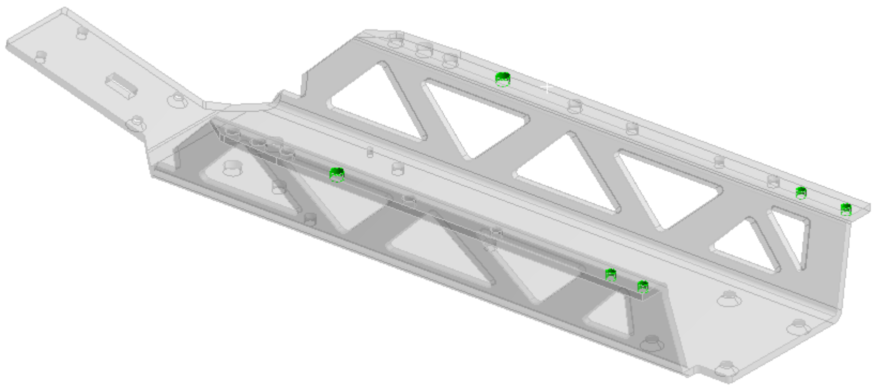


Figure 7 - 19: Pattern Features detected by PFR on the Chassis

Finding Non-Mating and Coplanar Features

Modified DoC function extracts three active directions in this part. These directions are lined up in x, y and z directions. Since there are lots of cylindrical faces in this part, Figure 7 - 20 illustrates the directions with only some of their corresponding features.

```
#1(1i,0j,0k)---#2(0i,1j,0k)---#3(0i,0j,1k)
      |             |             |
    Plane         Plane         Plane
      |             |             |
    Plane         Axis          Axis
      |             |             |
    Plane         Plane         Axis
      |             |             |
    Plane         Axis          Axis
      |             |             |
    Plane         Plane         Axis
      |             |             |
    Plane         Axis          Axis
      |             |             |
    Plane         Plane         Axis
      |             |             |
    Plane         Axis          Axis
      |             |             |
    Plane         Axis          Axis
      |             |             |
    Plane         Axis          Axis
      |             |             |
      |             |             |
```

Figure 7 - 20: Extracted Active Directions for the Chassis

The non-mating features recognizer finds two external prismatic FOS, two slots and three planar feature. The results are shown in Figure 7 - 21.

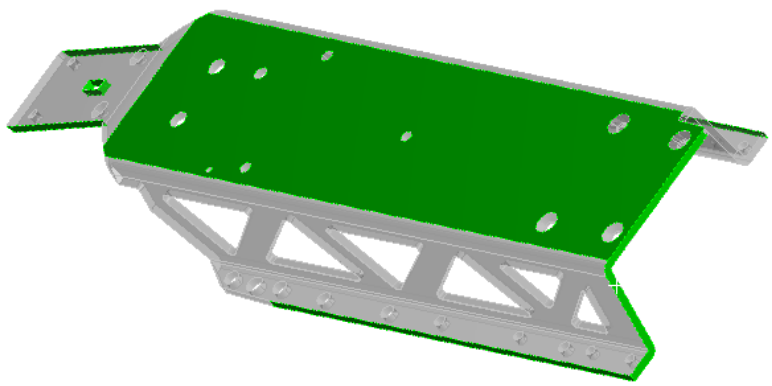


Figure 7 - 21: Non-Mating Features Detected in the Chassis

The coplanar feature recognizer detects two coplanar features in this part by traversing the extracted DoCs. The first one includes two top mating planes and the second one consists two planes under mating ones. These features are presented in the Figure 7 - 22.

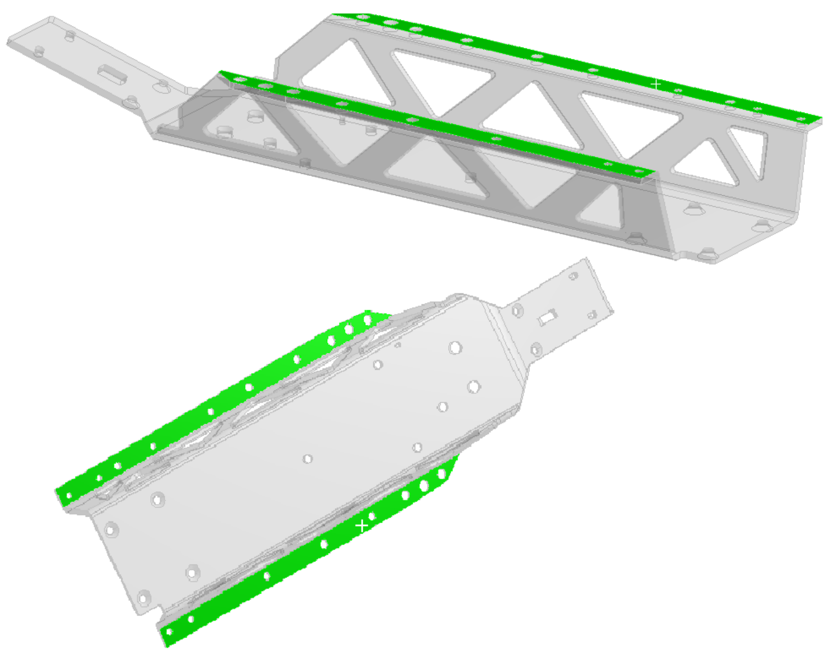


Figure 7 - 22: The Coplanar Feature Detected in the Right Support

Location Tolerance Frames

In this stage, all features are scored and ranked in their groups. Table 7 - 4 shows some of the features and their scores.

Type	Non-Cylindrical	Score	Cylindrical feature	Score
1	Bottom face of the Chassis	65		
2	Coplanar faces where two patterns seat	43		
3	Rear face of the Chassis	17		
4	The external rectangular FOS	16		
5	Non-mating slot	14		
6	Non-mating slot	13		

Table 7 - 4: Some of the Features in the Chassis and Their Scores

Two location tolerances are required for this part: one for the pattern of four holes and the other one for the pattern of two holes. These two patterns have the same position tolerance and DRF. Thus, the process of selecting datums is described for one of the patterns and the same procedure is applicable to the other one.

To choose the primary datum, bottom face of the Chassis is the non-cylindrical candidate. As it shown in Table 7 - 4, there is no cylindrical candidate so as a result, bottom face is selected as primary datum. This datum controls two rotational DOFs of the target feature. For the secondary datum, rear face of the Chassis is selected since it has the highest rank among features that are perpendicular to the primary datum. The secondary datum constrains one of the translational DOFs of the target feature. There is still one translational DOF left and so, datum selection continues. For the tertiary datum, the external rectangular tab is chosen. At this stage, all active DOFs of the target features are restricted. So, the position tolerance frame is generated and basic dimensions are extracted. Figure 7 - 23 illustrates the position tolerance frames created for both patterns.

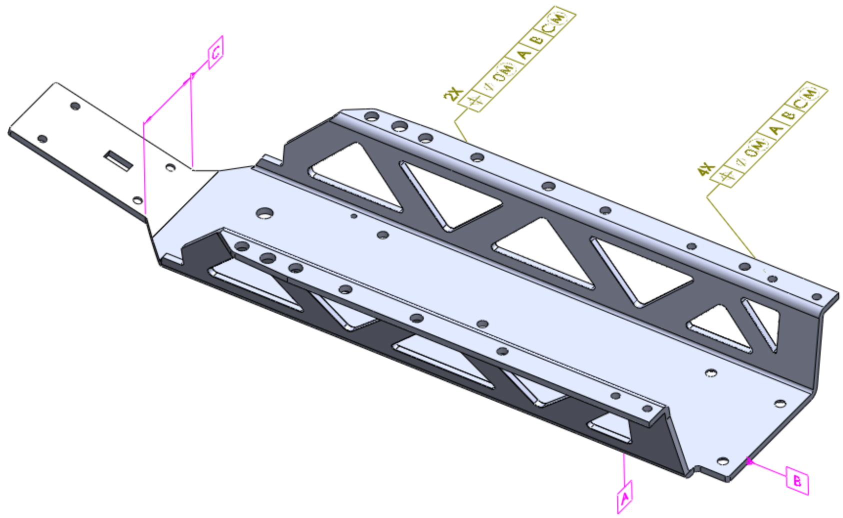


Figure 7 - 23: Position Tolerance Frames for Both Patterns

Size Tolerance Frames

Two pattern of features and one external FOS that is used as tertiary datum (datum C) require size tolerances. There is one size dimension tolerance frame for each of them to control size variations. Figure 7 - 24 shows the generated size tolerances on this part.

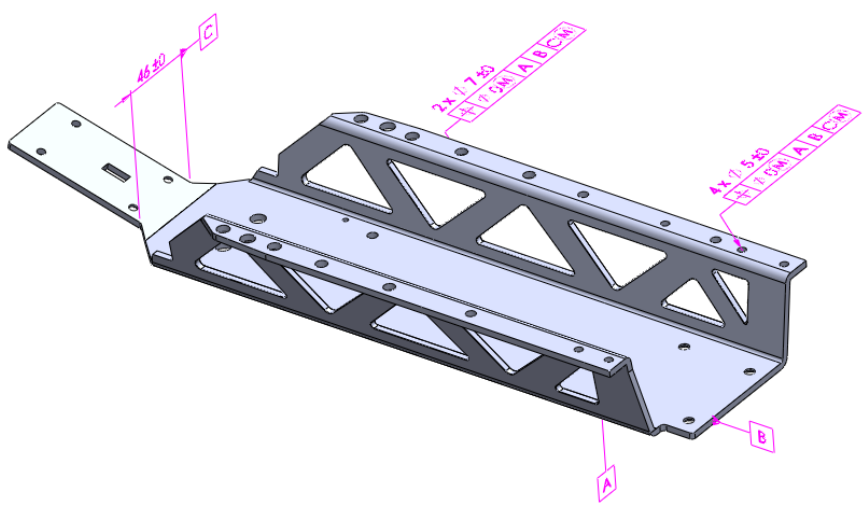


Figure 7 - 24: Size Tolerance Frames for the Chassis

Orientation Tolerance Frames

Secondary and tertiary datums need the orientation tolerance with respect to their preceding datums. Therefore, a perpendicularity tolerance is assigned to each of them. Patterns also require orientation tolerance. Each of the patterns has a parallelism tolerance with respect to its primary datum and a perpendicularity tolerance with respect to its secondary datum (Figure 7 - 25).

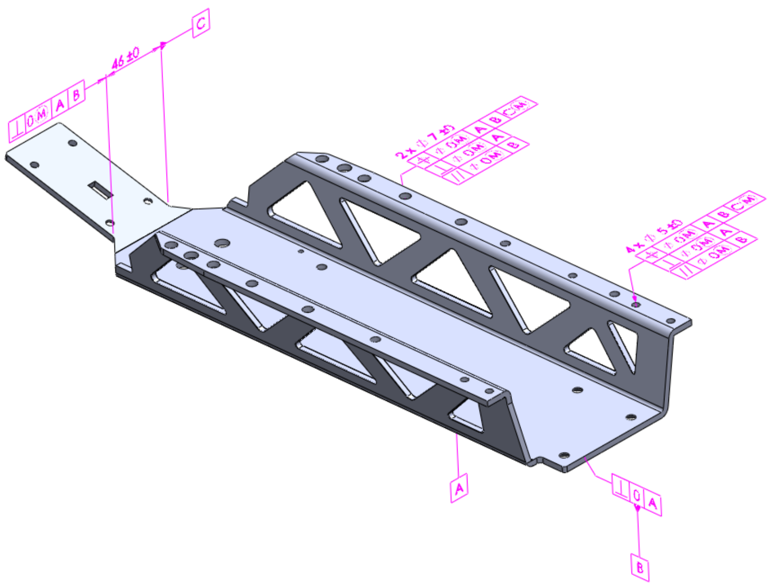


Figure 7 - 25: Orientation Tolerance Frames for the Chassis

Form Tolerance Frames

A total number of five tolerance frames are required for this part. A flatness tolerance must be assigned to each datum. A straightness tolerance frame is also must constructed for each pattern. Figure 7 - 26 presents the form tolerance frames created for this part.

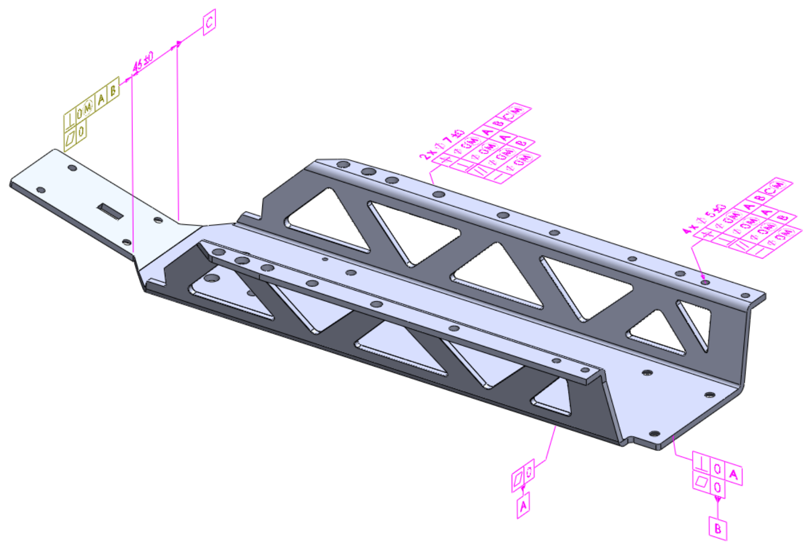


Figure 7 - 26: Form Tolerance Frames for the Chassis

Creating GD&T Schema for Remaining Parts in the Assembly

As mentioned earlier, except standard pins, other four parts need GD&T scheme. The same procedure of creating GD&T scheme that is used for the Chassis is applicable to the remaining parts. Since two Brackets in the Radio Car are the same, Figure 7 - 27 presents the final GD&T for the remaining three parts. Figure 7 - 28 also shows a section of the skeleton CTF generated for this assembly.

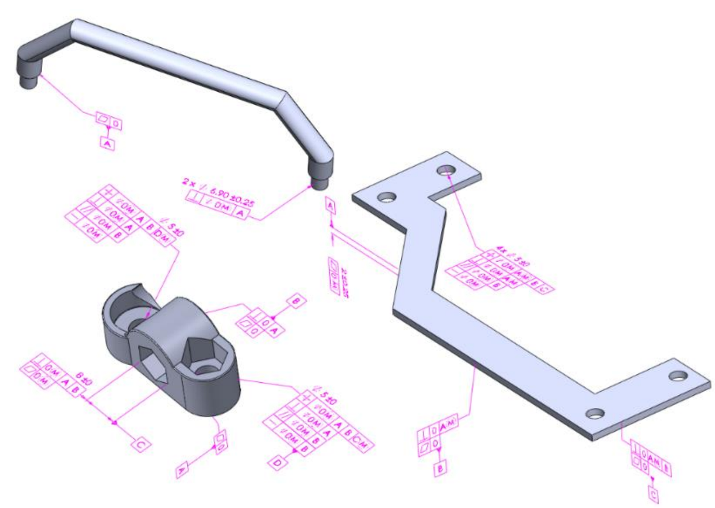


Figure 7 - 27: Complete GD&T Scheme for the Remaining Distinct Parts

```
#0=rd.SAT
#1=PART('part1', #2, #3, #4, #5, #6, #7, #8, #9, #10, #11, #12);
#2=GENERAL_PLANE('(FACE54)_part1', (-54, -62.3405, 175.516), [-1, 0, 0], 370.739,
#3=GENERAL_PLANE('(FACE49)_part1', (-38.047, -35.5883, -7), [0, -0, -1], 120.038,
#4=GENERAL_MIDPLANE('(FACE160&FACE164)_part1', (-37.7863, -62.5, 412.799), [0, 1,
#5=PATTERN_OF_HOLES(4'(FACE42&FACE43&FACE19&FACE20)_part1', [-1, 0, 0], 2.5, 4);
#6=HOLE('(FACE42)_part1', (-0.1, -125, 27), [-1, 0, 0], 2.5, 4);
#7=HOLE('(FACE43)_part1', (-0.1, -125, 47), [-1, 0, 0], 2.5, 4);
#8=HOLE('(FACE19)_part1', (-0.1, 0, 1), [-1, 0, 0], 2.5, 4);
#9=HOLE('(FACE20)_part1', (-0.1, 0, 29), [-1, 0, 0], 2.5, 4);

#54=CST_DISTANCE(62.5, #6 ,#4);
#55=METRIC_RELATIONSHIP(#54, CST_DISTANCE, (62.5, #6[LINE(axis of HOLE)], #4[MIDPLANE]))
#56=CST_DISTANCE(62.5, #7 ,#4);
#57=METRIC_RELATIONSHIP(#56, CST_DISTANCE, (62.5, #7[LINE(axis of HOLE)], #4[MIDPLANE]))
#58=CST_DISTANCE(62.5, #8 ,#4);
#59=METRIC_RELATIONSHIP(#58, CST_DISTANCE, (62.5, #8[LINE(axis of HOLE)], #4[MIDPLANE]))
#60=CST_DISTANCE(62.5, #9 ,#4);
#61=METRIC_RELATIONSHIP(#60, CST_DISTANCE, (62.5, #9[LINE(axis of HOLE)], #4[MIDPLANE]))
#62=CST_DISTANCE(223.5, #11 ,#3);
#63=METRIC_RELATIONSHIP(#62, CST_DISTANCE, (223.5, #11[LINE(axis of HOLE)], #3[PLANE]))

#182=T_SIZE(#5, (nFI, 0, RFS));
#183=DOF(#182, (SIZE_DOF, SHAPE_DOF));
#184=T_SIZE(#10, (nFI, 0, RFS));
#185=DOF(#184, (SIZE_DOF, SHAPE_DOF));
#186=T_SIZE(#4, (nFI, 0, RFS));
#187=DOF(#186, (SIZE_DOF, SHAPE_DOF));
#188=T_POSITION(#5, (FI, 0, MMC)), PD(#2, NONE), SD(#3, NONE), TD(#4, MMC));
#189=DOF;
#190=T_POSITION(#10, (FI, 0, MMC)), PD(#2, NONE), SD(#3, NONE), TD(#4, MMC));
#191=DOF;
#192=T_PERPENDICULARITY(#3, (FI, 0,NONE)), PD(#2, NONE));
#193=DOF;
```

Figure 7 - 28: A Section of the Generated CTF for the Radio Car Assembly

7.2.2 Recommended GD&T Manually Applied

The manual GD&T scheme on the Radio Car assembly was initially made in the University of Vanderbilt. Figure 7 - 29 shows the GD&T scheme for chassis and roll bar sub- assembly.

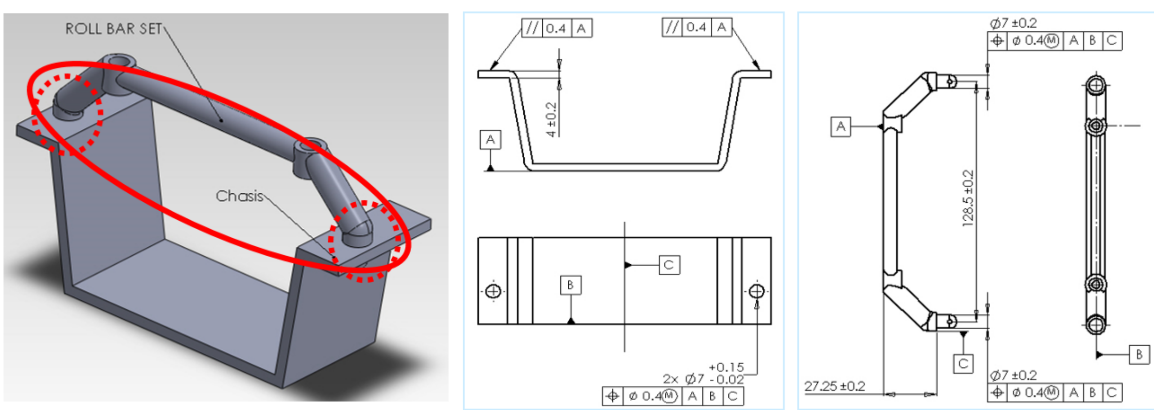


Figure 7 - 29: GD&T Created in the University of Vanderbilt for the chassis and roll bar sub-assembly

The tolerance analysis on this GD&T showed that it was an incomplete scheme. Therefore, the GD&T was improved in the DAL without altering tolerance values (Figure 7 - 30).

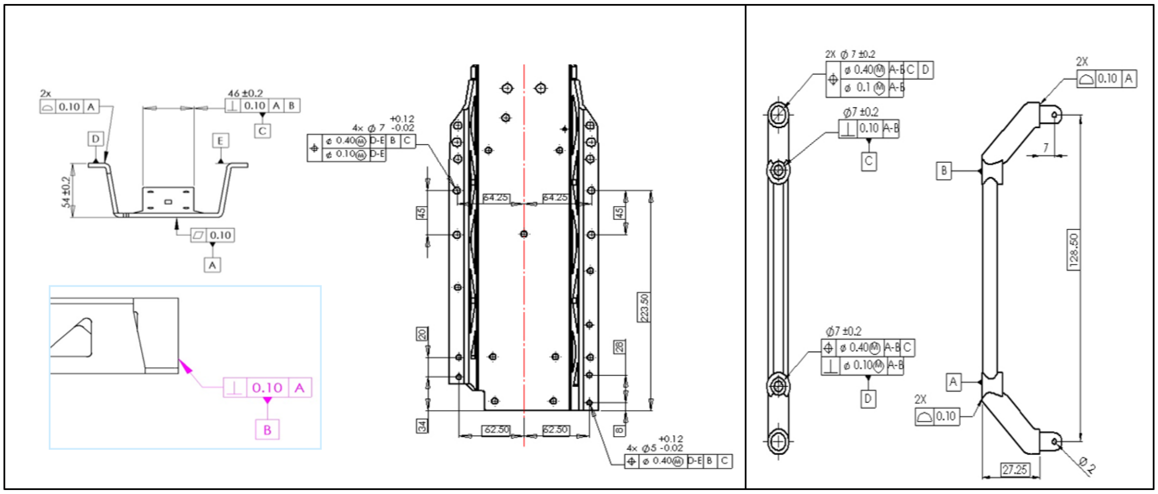


Figure 7 - 30: Improved GD&T by DAL for the chassis and roll bar sub-assembly

7.2.3 Comparing GD&T Results

The manually applied GD&T scheme and the one created by the schema generation module for one parts of the assembly are compared together in this section. The comparison is only made to find the similarities and differences in selecting datums.

The Chassis

Figure 7 - 31 GD&T schemas for the Chassis. Left figure presents the tolerance frames made by this module and right illustrates the manually applied ones. Table 7 - 5 also addresses the similarities and differences between these two schemes in choosing datums.

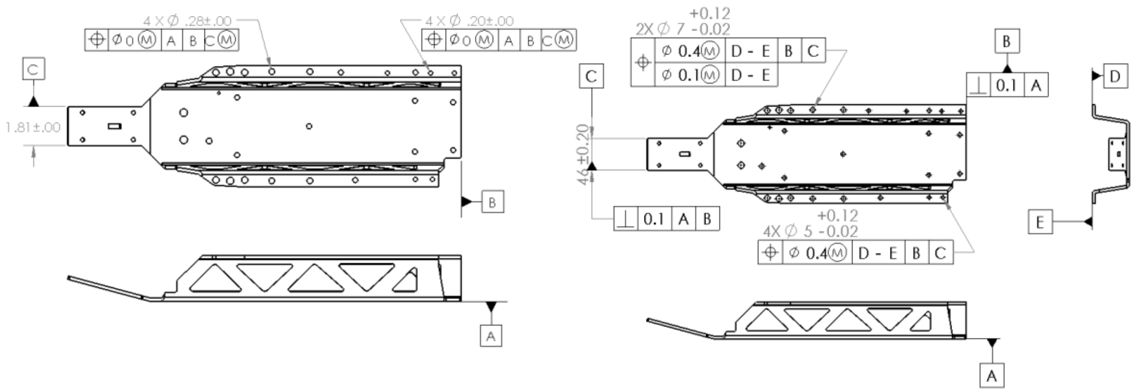


Figure 7 - 31: GD&T Schemas Created Manually and by Schema Generation Module

Similarities	Datums A, B and C are the same for both schemas Secondary and tertiary datums for two pattern are the same in both schemas
Differences	Two new datums are defined in the manually applied GD&T: datums D and E Primary datum for two patterns in the manually applied GD&T is D-E Primary datum for two pattern in the software-generated GD&T is A
Logic behind datum selection done manually	Two top faces are selected as primary datum because it can provide better control for locating and orienting pattern of holes
Logic behind datum selection done by software	Although top faces are mating, bottom face is large enough to choose over them as primary datum

Table 7 - 5: Similarities and Differences Between GD&T Schemes for the Chassis

7.3 Case Study 3: The Body Cap

The third test case is a Body Cap assembly from the ASME Y14.5 standard book [5] which has eight parts (Figure 7 - 32). Out of these eight parts, five of them are standard pins. For this assembly, the Cap is taken as an example. The same procedure is applicable to the rest of the parts.

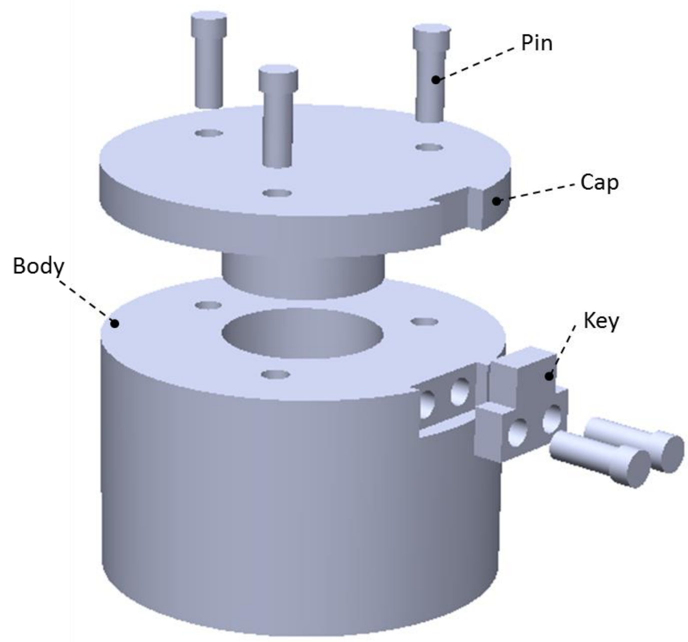


Figure 7 - 32: The Body Cap Assembly

7.3.1 Recommended GD&T by Schema Generation Module

Assembly Analysis Results

Running the assembly CAD file through preprocessing modules lead to the following results. There total number of 23 part-level and 14 assembly features detected by AFR in the assembly. Out of 23 part-level features in the assembly, there are six features (three through holes, one pin, one slot and one plane) in the Cap (Figure 7 - 33).

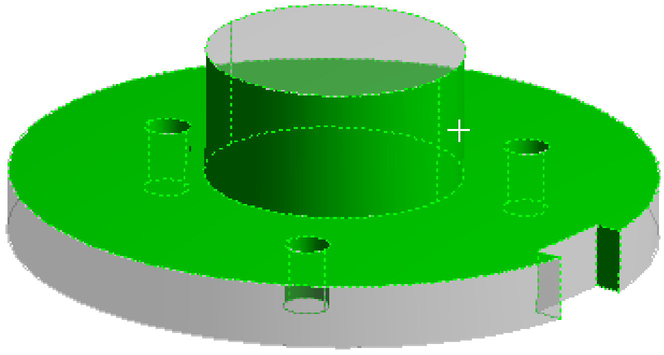


Figure 7 - 33: Part-Level AFR Results for the Cap

There are four patterns detected by PFR in this assembly. Out of these four patterns, there is one circular pattern of holes in this part. The PFR results is visualized in Figure 7 - 34.

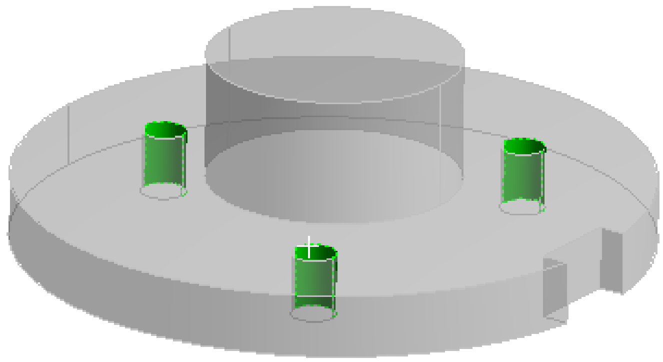


Figure 7 - 34: PFR Results for the Cap

Finding Non-Mating and Coplanar Features

Running the Cap through the modified DoC function shows that there are three active DoCs in this part. Figure 7 - 35 presents the extracted active DoCs and their corresponding features.

```
#1(1i,0j,0k)---#2(0i,1j,0k)---#3(0i,0j,1k)
      |              |              |
     Axis          Plane           Axis
      |              |              |
     Axis          Plane           Axis
      |              |              |
     Axis          Plane           Axis
      |                             |
     Plane                         Axis
      |                             |
     Axis                          Axis
      |                             |
     Plane                         Plane
      |
     Axis
```

Figure 7 - 35: Extracted Active DoCs for the Cap

Non-mating features recognizer finds two planar features in this part by traversing extracted DoCs (Figure 7 - 36). There is no coplanar feature recognized in this part.

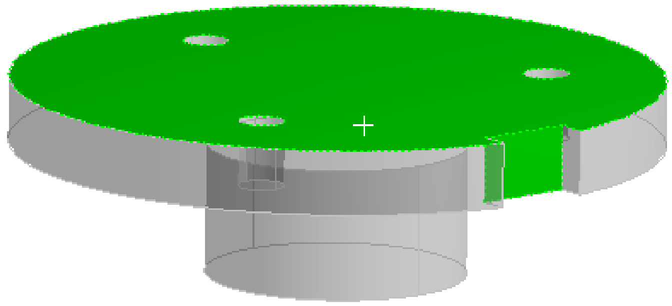


Figure 7 - 36: Two Planar Features Detected in the Cap

Location Tolerance Frames

To create location tolerance frames, all features are scored and ranked. Table 7 - 6 shows the ranked features based on their category and overall score.

Type	Non-Cylindrical	Score	Cylindrical feature	Score
1	The mating semi-circular face	85	The central pin	100
2	The non-mating semi-circular face	17		
3	The mating slot	15		
4	The rectangular face at the end of the slot	15		

Table 7 - 6: Features and Their Scores for the Cap

Three tolerance frames are required to locate mating part features in this part: one for the central pin, one for the slot, and one for the pattern of holes. To create a DRF for the central pin, all potential features in both cylindrical and non-cylindrical groups should be queried. For primary datum, the non-cylindrical candidate is the mating semi-circular plane where the central pin seats. The cylindrical candidate is the central pin itself. Since a feature cannot be used as datum to locate itself, the mating semi-circular plane is the only option for primarily datum. The primary datum constrains only two rotational DOFs of the central pin while two translational DOFs of the central pin is left to be controlled. For secondary datum, the non-cylindrical candidates that are perpendicular to the primary datum are the mating slot and the rectangular plane at the end of slot. Due to the rule D10, when we want to create location tolerance frame for the slot, the central pin is selected as datum. Hence, the slot cannot be used as a datum feature reference for positioning the central pin and rectangular plane is selected as secondary datum. Secondary datum controls one of the translational DOFs of the central pin and as a result, datum selection process should continue. There is no potential datum feature that is perpendicular to primary and secondary datum features to be selected as tertiary feature. Because all of the active DOFs of the central pin are not constrained by the combination of the primary and secondary datums, the location tolerance frame cannot be created for the central pin.

To locate the slot, there are two candidates for choosing primary datum. The mating semi-circular plane and the central pin. Using the rule D10, the plane is selected as primary datum. This plane restricts one rotational DOF of the slot. The rectangular plane at the end of the slot and the central pin are the candidates for secondary datum that are perpendicular to the primary datum. Among these two candidates, the central pin is selected as secondary datum using the rule D10. The central pin controls the remaining translational and rotational DOFs of the slot, hence, position tolerance frame is created for the slot. The same procedure is applicable to the pattern of holes. Figure 7 - 37 illustrates the location tolerance frames for both slot and pattern of holes.

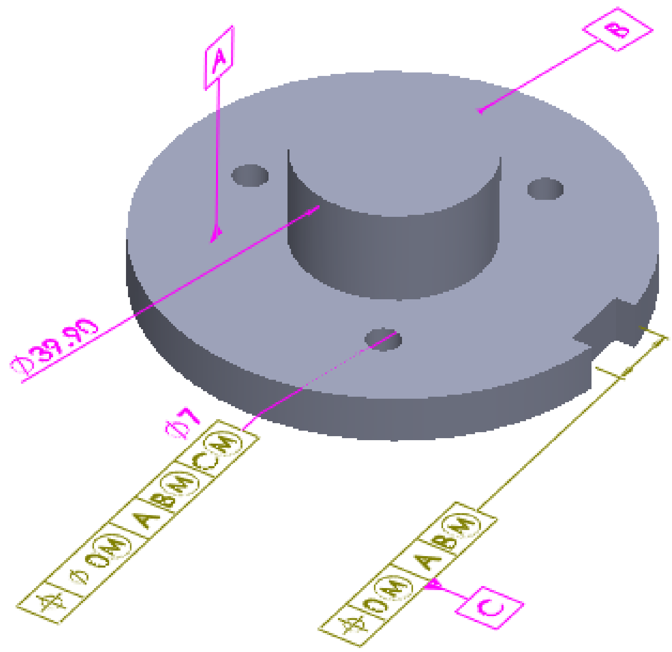


Figure 7 - 37: Location Tolerances for Both Slot and Pattern of Holes

Size Tolerance Frames

There are total number of five size tolerances required for this part: two size tolerances for the slot, one for the pattern of holes and two for the central pin (Figure 7 - 38).

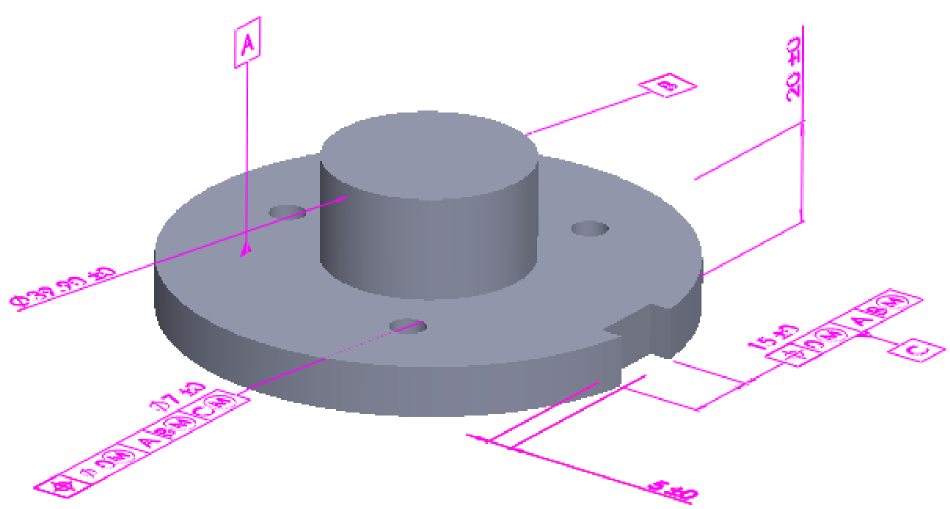


Figure 7 - 38: The Cap with Size Tolerance Frames Added

Orientation Tolerance Frames

Secondary and tertiary datums need the orientation tolerance with respect to their preceding datums. Therefore, a perpendicularity tolerance is assigned to each of them. The central pin, the pattern of holes and the slot also require orientation tolerance frames. Figure 7 - 39 shows the added orientation tolerance frames.

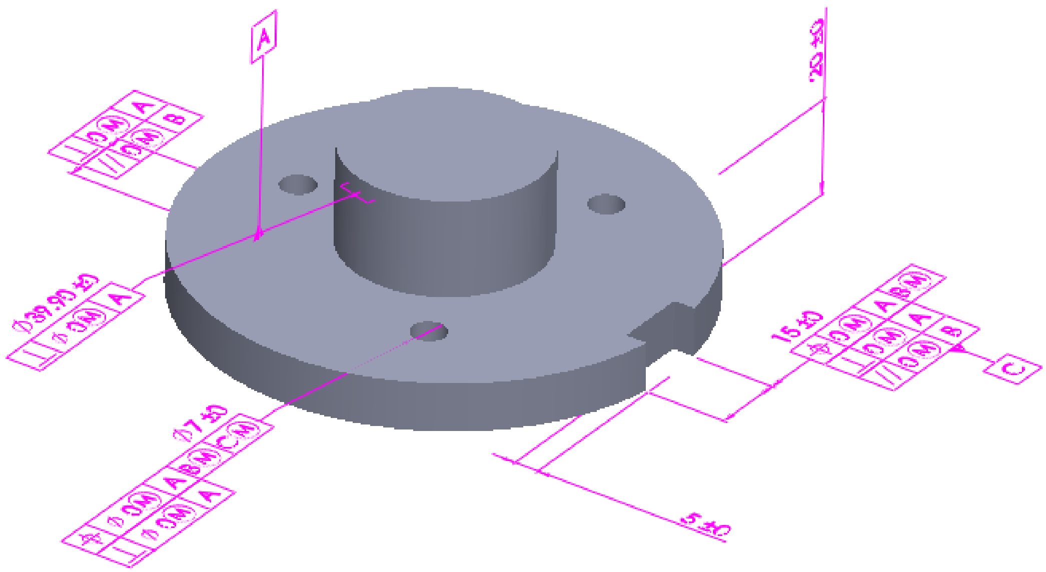


Figure 7 - 39: The Cap with Orientation Tolerance Frames Added

Form Tolerance Frames

A Total number of four tolerance frames are required for this part. Two flatness tolerances for primary and tertiary datum features, two straightness tolerances for secondary datum feature and holes in the pattern. Figure 7 - 40 presents the form tolerance frames created for this part.

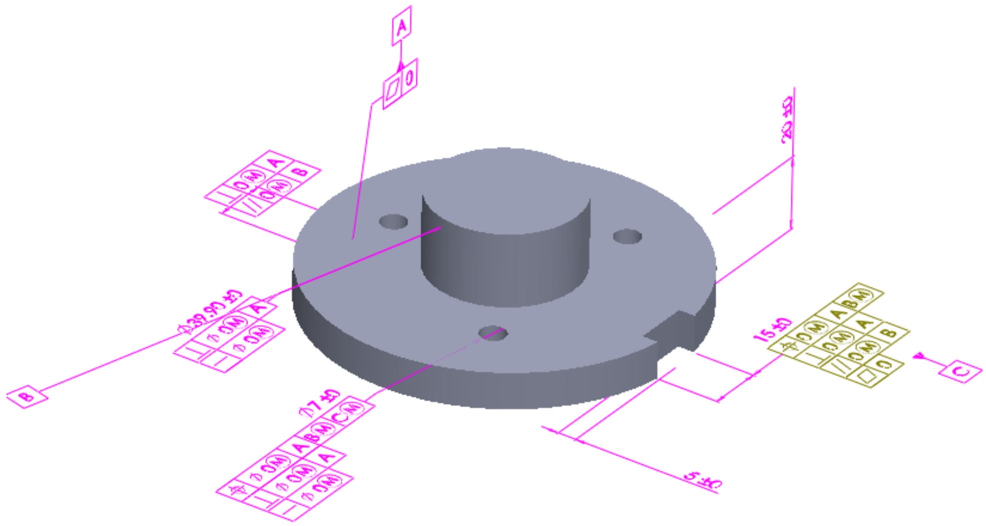


Figure 7 - 40: Right Support with Form Tolerance Frames Added

Creating GD&T Schema for The Remaining Parts in The Assembly

The same procedure mentioned above is applied for other parts in order to complete the GD&T schema for the whole assembly. Figure 7 - 41 presents a section of the skeleton CTF generated for the Body Cap assembly.

```
#14=PART('part2', #15, #16, #17, #18, #19, #20, #21, #22, #23);
#15=PIN('(FACE20)_part2', (30.6655, 51.3691, 67.2408), [0, 1, -0], 19.95, 20);
#16=GENERAL_PLANE('(FACE25)_part2', (30.6655, 71.3691, 67.7742), [0, -1, 0], 100
#17=SLOT('CTP(FACE29&FACE28)_part2', (30.6655, 76.3691, 22.2408), [0, 0, 1], 15,
#18=RECTANGULAR_PLANE('(FACE30)_part2', (30.6655, 76.3691, 22.2408), [0, 0, -1],
#19=PIN('(FACE24)_part2', (30.6655, 71.3691, 67.2408), [0, 1, -0], 50, 10);
#20=PATTERN_OF_HOLES(3'(FACE22&FACE23&FACE21)_part2', [0, 1, -0], 3.5, 10);
#21=HOLE('(FACE22)_part2', (2.9527, 71.3691, 51.2408), [0, 1, -0], 3.5, 10);
#22=HOLE('(FACE23)_part2', (30.6655, 71.3691, 99.2408), [0, 1, -0], 3.5, 10);
#23=HOLE('(FACE21)_part2', (58.3783, 71.3691, 51.2408), [0, 1, -0], 3.5, 10);
#24=PART('part3', #25);
#25=PIN('(FACE31)_part3', (58.3783, 63.3691, 51.2408), [-0, 1, 0], 3.45, 18);
#26=PART('part4', #27);
#27=PIN('(FACE36)_part4', (30.6655, 63.3691, 99.2408), [0, 1, -0], 3.45, 18);

#85=METRIC_RELATIONSHIP(#84, CST_PERPENDICULAR, (90, #11[LINE(axes of HOLES in F
#86=CST_DISTANCE(45, #19 ,#18);
#87=METRIC_RELATIONSHIP(#86, CST_DISTANCE, (45, #19[LINE(axis of PIN)], #18[PLAN
#88=CST_DISTANCE(16, #21 ,#15);
#89=METRIC_RELATIONSHIP(#88, CST_DISTANCE, (16, #21[LINE(axis of HOLE)], #15[LIN
#90=CST_DISTANCE(32, #22 ,#15);
#91=METRIC_RELATIONSHIP(#90, CST_DISTANCE, (32, #22[LINE(axis of HOLE)], #15[LIN
#92=CST_DISTANCE(16, #23 ,#15);
#93=METRIC_RELATIONSHIP(#92, CST_DISTANCE, (16, #23[LINE(axis of HOLE)], #15[LIN
#94=CST_DISTANCE(27.7128, #21 ,#17);
#95=METRIC_RELATIONSHIP(#94, CST_DISTANCE, (27.7128, #21[LINE(axis of HOLE)], #1

#218=T_SIZE(#20, (nFI, 0, RFS));
#219=DOF(#218, (SIZE_DOF, SHAPE_DOF));
#220=T_POSITION(#20, (FI, 0, MMC)), PD(#16, NONE), SD(#15, MMC), TD(#17, MMC));
#221=DOF;
#222=T_POSITION(#17, (FI, 0, MMC)), PD(#16, NONE), SD(#15, MMC));
#223=DOF;
#224=T_PERPENDICULARITY(#15, (FI, 0,MMC)), PD(#16, NONE));
#225=DOF;
#226=T_PERPENDICULARITY(#17, (FI, 0,MMC)), PD(#16, NONE));
#227=DOF;
#228=T_PARALLELISM(#17, (FI, 0,MMC)), PD(#15, MMC));
#229=DOF;
#230=T_PERPENDICULARITY(#20, (FI, 0,MMC)), PD(#16, NONE));
#231=DOF;
#232=T_PARALLELISM(#20, (FI, 0,MMC)), PD(#15, MMC));
#233=DOF;
```

Figure 7 - 41: A Section of the Generated CTF for the Body Cap Assembly

7.3.2 Comparing GD&T Results

In this subsection, the GD&T schema generated by the schema generation module and the one that is provided through ASME Y14.5 standard book [5] are compared to find similarities and differences in location tolerances and datum selection. This comparison is made only for the Cap. Comparing the software-generated schema and the one provided in the ASME Y14.5 standard shows that they have the same location tolerance frames and same DRFs (Figure 7 - 42).

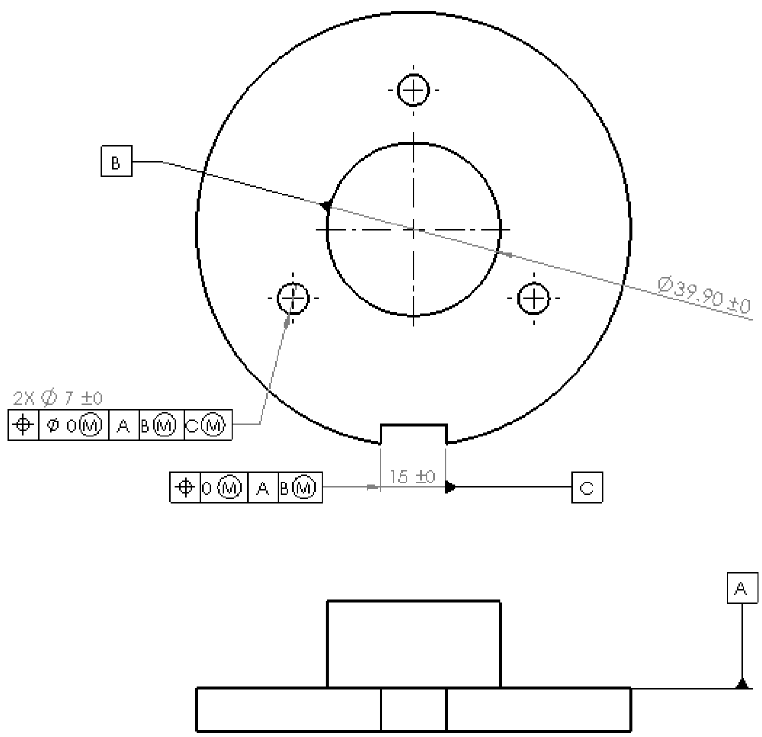


Figure 7 - 42: Size and Location Tolerance Frames Created for the Cap

CHAPTER 8

CLOSURE

This chapter summarizes the research described in previous chapters and concludes its limitations. Possible future works that can extend the scope of this research are also discussed in this chapter.

8.1 Summary and Conclusion

A proper tolerance scheme can cover all the imperfections and variations due to errors associated with design and manufacturing. Generating the appropriate GD&T scheme needs an in-depth understanding of two GD&T standards[5,6] and years of experience and in the field of tolerance synthesis and analysis. GD&T schema as a base point for process planning, serves the role of a common language for designer, manufacturer and inspector. Observations shows that most of the problems associated with low acceptance rate of the manufactured assemblies are due to lack of proper tolerance scheme. Most of the researches on automating the tolerance synthesis have been performed in the tolerance value allocation and analysis area. Only a few number of them have been done in tolerance schema generation. To address this issue, the Auto-Tolerancing software was proposed and developed in DAL. Auto- Tolerancing software automates the process of tolerance schema generation as well as the tolerance value allocation and analysis. This research starts with giving an overview of the Auto-Tolerancing project. In this overview, all of the modules and the relationship between them are explained. This research focuses on the conceptual design and implementation of the 1st order tolerance schema generation module as a part of Auto-Tolerancing project. Tolerance schema development starts with extracting the geometric properties that help module to produce an effective schema. These geometric includes finding non-mating and coplanar features and defining the shape of planes. Since schema generation module is a knowledge-based system, it requires some sets of rules to develop a GD&T schema. The strengths and drawbacks of the experiential GD&T rules extracted by experts are discussed and then these rules are developed and improved. As the first tolerance type in the GD&T schema, location tolerance frames are created for mating FOSs using the proposed feature scoring system and DOF algebra. Second tolerance that is generated in the GD&T schema is size tolerance. Size tolerance controls the size and shape DOFs of the target features. Orientation and form tolerances are assigned to target features to refine tolerance zones defined by other types of tolerances or to stablishes a new control for orientation and form variations. Once the GD&T schema is completed, its communicated to the downstream modules for tolerance value allocation/analysis and file conversion.

8.2 Limitations

As an academic piece of software that is developed based on heuristic GD&T rules, the tolerance schema generation module cannot produce a GD&T schema that is compatible with the one made by an expert. The module developed in this research has the following major limitations:

1- This module is limited to the 1st order tolerancing. It means the functional requirements of the assembly or cost limitations have not been taken into account while module creates a GD&T schema.

2- The tolerance types that are assigned to target features are limited to four major types: size, location, orientation, and form. The use of profile and run-out tolerances haven't been considered in the current scope of the research.

3- Only few test cases have been studied so far. To verify the comprehensiveness of the recommended tolerance schemas and the scalability of the toolset, more complex case studies with more number of parts are required.

8.3 Future Work

This section suggests the future improvements that can be made on both conceptual design and implementation of the software in the future. These modifications can enhance the efficiency of the software-generated GD&T schemas and extend the scope of the module.

8.3.1 Use of Assembly Loops for Local Control

The assembly loop detection is a module developed in preprocessing phase in order to extract assembly feature loops. When the framework of the auto-tolerancing project was developed initially, the loop detection module was suggested to help building the tolerance schema. During the time that the schema generation module was designing and developing, it was assumed that loop detection information is not necessary. The logic behind this assumption was that the assembly information from AFR and PFR and the extracted modified DoCs plus GD&T rulesets are enough for making and effective tolerance schema. After studying test cases it was realized that using assembly loops information can help making more efficient schemas where a local control between two features is required.

The local control refers to the situation where it is preferable to locate a FOS with a local rather than master DRF. This happens when the target feature and its potential datum are in the same assembly loop. In this situation, controlling the location of the target feature with the potential datum that is in the same loop as the target is, eliminate the stack-up tolerance between two features therefore, a wider tolerance can be used for locating the target feature. The GD&T scheme shown in the Figure 8 - 1 presents the idea of the local control. In this figure, the pin is positioned with respect to the master DRF. Suppose that the pin and the hole next to the pin are in the same assembly loop. Positioning the hole with the master DRF increases the variations between pin and hole. Since the pin and hole are in the same loop, the more variations between them means the less chance of assemblability. Hence, it is better to use the pin as a datum (local DRF) for the hole.

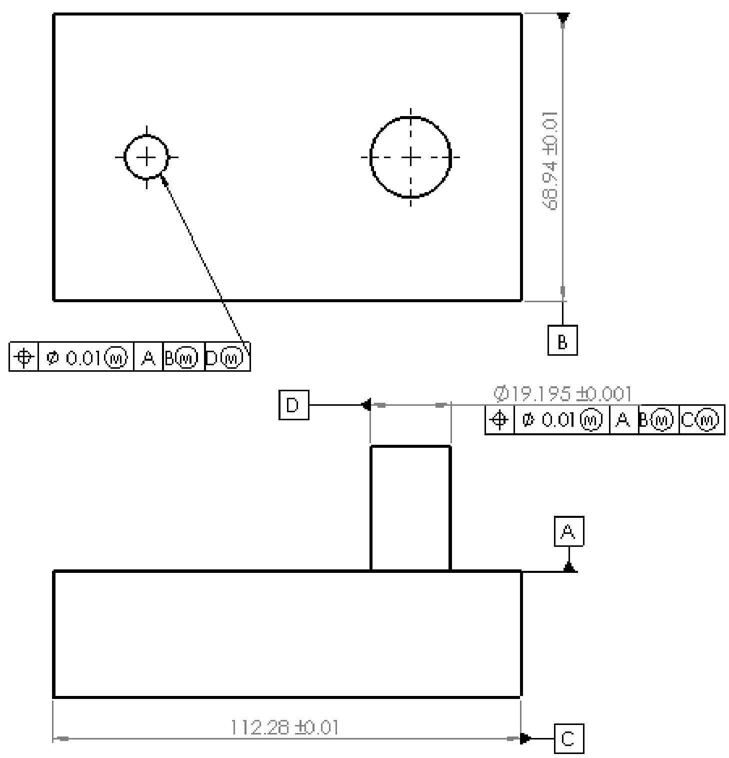


Figure 8 - 1: The hole is positioned using local DRF

It should be noted that the assembly loop information cannot be used as the only criterion for selecting datums. Although the local control is important, if the local DRF is not appropriate to use due to geometric inappropriateness of the potential datum, the master DRF must be used for positioning the target feature. To elaborate, suppose the pin in Figure

8 - 1 has very short length and small diameter. Although it can provide a local control for the hole, its unqualified geometric properties (short length and small diameter) prevents it from to be chosen as datum for the hole. Hence, there should be criterion that identifies when to use loop information over rulesets. This could be done by assigning scores to the assembly loops. In this situation, if a loop has good enough score, it dominates the rule of selecting datums. This idea can be implemented in the near future since the assembly loop information is already available.

8.3.2 Redesigning the data structures

As discussed in previous chapter, the current design of the data structures for representing features, constraints and tolerances is not efficient in a sense that this design makes the module difficult to maintain and reuse by other developers. This section proposes the new data structure design for representing features, constraints and tolerances. These new designs not only addresses issues mentioned above, but also makes the data structure more compatible with ASME Y14.5 standard. It should be noted that these changes are easy to implement and can be done in near future.

New Data Structure for Representing Features

In the new design, different feature and their subtypes get inherited from the base geometric feature class. The base feature class contains all data and function members that are common between all features. Then, each feature will have its own specific data and function members. Another change that is proposed here for representing features is that features can point to the tolerances associated with them. In the current implementation, there is no such a data member defined in the class structure. Figure 8 - 2 shows the hierarchy of the proposed data structure.

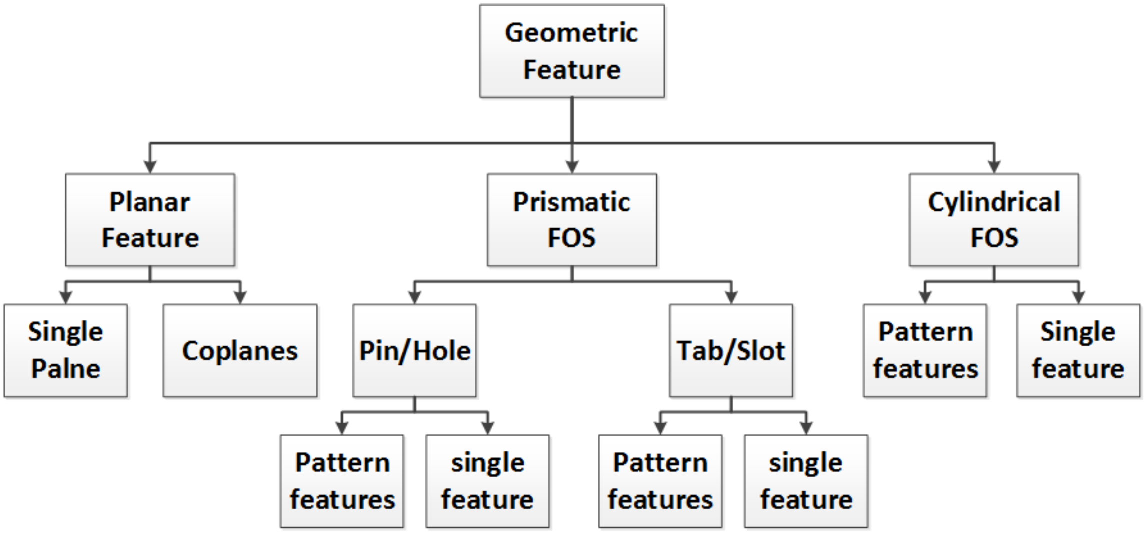


Figure 8 - 2: The Hierarchy of the Proposed Features Classes

In this new design, geometric feature (the parent) class has the structure as it shown in Figure 8 - 3. The class members are the ones that are common between all types of features.

```
// This is Geometric Feature class.

Class GeometricFeature
{
        private:
        // data members
        geom_enum_type geometryType;
        SPAunitvector direction;
        SPAposition basePoint;
        int overllScore;
        vector<Constraints*> constraints
        vector<Tolerances*> tolerances
        …

        public:
        // set functions
        void set_overall_score (int OverallScore);
        void set_base_point (SPAposition BasePoint);
        ….
        // get functions
        vector<Tolerances*> get_tolerances();
        ….
};
```

Figure 8 - 3: Geometric Feature Class Structure

The planar class is inherited from geometric class and has two subclasses: coplanar feature and planar feature. These two subclasses have the same data members other than that the coplanar feature class points to multiple underlying topological entities (faces) and stores multiple face IDs while the planar feature class only points to the one face and stores one face ID. The structures of the planar class and its two subclasses are presented in Figure 8 - 4.

```
// This is Planar Feature class.

Class PlanarFeature: public GeometricFeature
{
        private:
        // data members
        feat_enum_type featureType;
        float area;
        float  length;
        float width;
        SPAunitvector lengthDirection;
        int areaRatioScore
        int aspectRatioScore
        …

        public:
        // set functions
        void set_area (float Area);
        void set_width (float width);
        void set_aspect_ratio_score (int AspectRatioScore);
        ….
        // get functions
        Float get_length();
        int get_accessibility_score ();
        SPAunitvector get_length_direction
        ….
};
```

```
// This is Single Plane class.

Class SinglePlane: public PlanarFeature
{
        private:
        int faceID;
        FACE * face;
        …

        public:
        // set functions
        void set_face (FACE * Face);
        ….
};
```

```
// This is Coplanes class.

Class Coplanes: public PlanarFeature
{
        private:
        vector<int> faceIDs;
        …

        public:
        // get functions
        ENTITY_LIST get_faces ();
        ….
};
```

Figure 8 - 4: Planar Feature Class and Its Subclasses

The prismatic class is also inherited from the geometric feature class. It has two subclasses of the pin/hole and tab/slot. The reason for defining two subclasses for prismatic class is that the pin/hole class points to the four side faces and it contains the centroids and the normal vectors of its two mid-planes while the tab/slot class only points to two side faces and it contains the geometric information of the one mid-plane. The pin/hole and tab/slot classes have also two subclasses each. Class of pattern features and class of single feature. Since both tab/slot and pin/hole could be in patterns, it is required to have subclasses in order to distinguish between them. As it presented in Figure 8 - 5 single feature class only points to the faces while pattern features class points to the features. The data structures of all these classes are shown in Figure 8 - 5.

```
// This is Prismatic FOS class.

Class PrismaticFOS: public GeometricFeature
{
        private:
        // data members
        float area;
        float width;
        float height;
        int areaRatioScore
        SPAunitvector heightDirection;
        feat_enum_type featureType;

        ...

        public:
        // set functions
        void set_area (float height);
        void set_height_dir(SPAunitvector heightDirection);
        void set_accessibility_score (int AccessibilityScore);
        ....
        // get functions
        Float get_height();
        int get_area();
        feat_enum_type get_feature_type();
        ....
};
```

```
// This is Tab Slot class.

Class TabSlot: public PrismaticFOS
{
};
```

```
// This is Single Tab Slot class.

Class SingleFeatureT: public TabSlot
{
        private:
        vector<int> faceIDs
        ....

        public:
        // set functions
        void set_face_list(ENTITY_LIST FaceList)
        ....
};
```

```
// This is Pattern Tab Slot class.

Class PatternFeaturesT: public TabSlot
{
        private:
        vector <SingleFeatureT* > Features

};
```

```
// This is Pin Hole class.

Class PinHole: public PrismaticFOS
{
        private:
        SPAposition basePoint2
        ....

        public:
        ....
};
```

```
// This is Single Pin Hole class.

Class SingleFeatureP: public PinHole
{
        private:
        ENTITY_LIST faceList
        ....

        public
        ....
};
```

```
// This is Pattern Pin Hole class.

Class PattrenFeatureP: public PinHole
{
        private:
        vector <SingleFeatureP* > Features
        ...

        Public
        ....
};
```

Figure 8 - 5: Prismatic FOS Class and its Subclasses

The cylindrical class also has two subclasses: a class for single features and a class for pattern features. As mentioned in previous paragraph, it is required to identify whether a cylindrical feature is in a pattern or not. Figure 8 - 6 illustrates the data structures of the cylindrical class and its two subclasses.

```
// This is Cylindrical FOS class.

Class CylindricalFOS: public GeometricFeature
{
        private:
        // data members
        feat_enum_type featureType;
        float  length;
        float diameter;
        int lengthRatioScore
        int aspectRatioScore
        ...

        public:
        // set functions
        void set_length (float Area);
        void set_diameter (float Diameter);
        void set_aspect_ratio_score (int AspectRatioScore);
        ....
        // get functions
        float get_length();
        int get_length_ratio_score();
        feat_enum_type get_feature_type();
        ....
};
```

```
// This is Single Pin Hole class.

Class SCPinHole: public CylindricalFOS
{
        private:
        ENTITY_LIST faceList;
        ...

        public:
        // set functions
        void set_face (FACE * Face);
        ....
};
```

```
// This is Pattern Pin Hole class.

Class PCPinHole : public CylindricalFOS
{
        private:
        vector<SCPinHole*> features;
        ...

        public:
        // get functions
        enum_type get_pattern_type ();
        ....
};
```

Figure 8 - 6: Cylindrical FOS Class and its Subclasses

New Data Structure for Representing Constraints

Constraints should be redesigned in a way that are more compatible to GD&T standards and reusable for future schema generation module's developments. In the proposed design, a parent constraint class is defined including all the common data members that its subclasses have. The constraint class has seven subclasses: mating, distance, coincident, concentric, perpendicular, parallel and angle constraints. Figure 8 - 7 presents constraints classes hierarchy and their data structures.

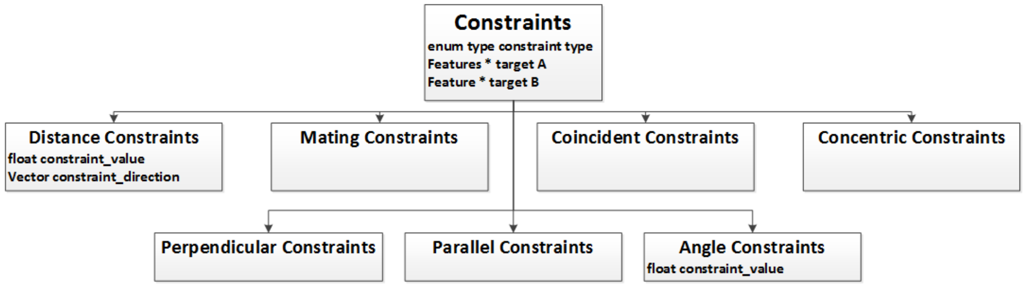


Figure 8 - 7: The Hierarchy and Data Structures of the proposed Constraint Classes

New Data Structure for Representing Tolerances

The proposed new design of the data structure for representing tolerances includes one parent class (tolerance) and four children classes (size, location, orientation and form). Each of these children also has their own children classes. Figure 8 - 8 shows the hierarchy and data structures of the tolerance class and its subclasses.

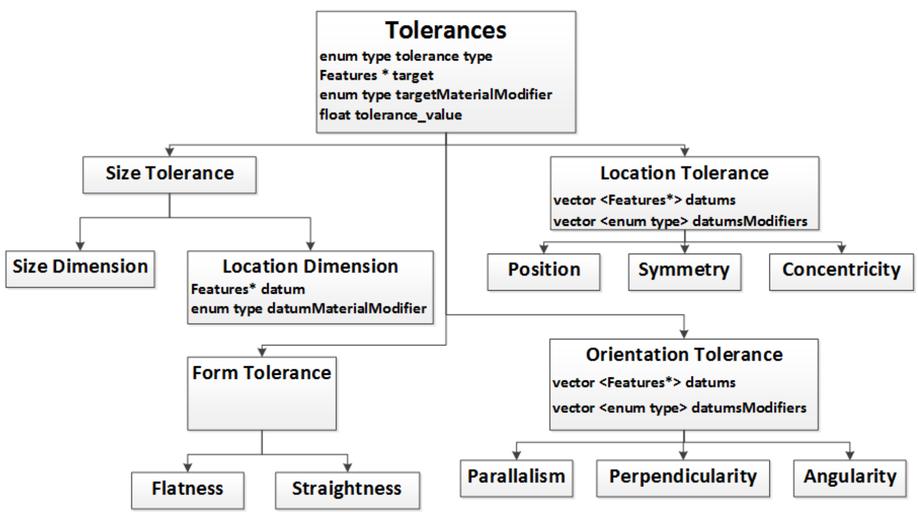


Figure 8 - 8: The Hierarchy and Data Structures of the Proposed Tolerance Classes

8.3.3 Incorporating Second and Third Order Tolerancing

In order to enable schema generation module makes more efficient and realistic GD&T schemes, it is a good idea to incorporate 2nd and 3rd order tolerancing into the current implementation.

8.3.4 Incorporating Other Tolerance Types

Some types of tolerances that are not included in this research can be implemented in the future. These tolerances include: composite and multiple single-segment pattern tolerances, profile tolerance, run-out tolerance, and so on.

8.3.5 Iterative Schema Improvement System

In the current implementation of the Auto-Tolerancing software, once the tolerance schema is generated, the recommended GD&T is transmitted to the tolerance value allocation module. In the tolerance value allocation module, tolerance values are allocated and analyzed iteratively till the satisfactory results are obtained. If the recommended GD&T is not efficient, tolerance value allocation might consume huge amount of time and memory to achieve desired results. Therefore, a feedback system can be designed and implemented between tolerance value allocation and schema generation modules. This system can provide feedbacks to the schema generation module to change the tolerance schema if the it is not efficient. Figure 8 - 9 shows the tentative design of this system.

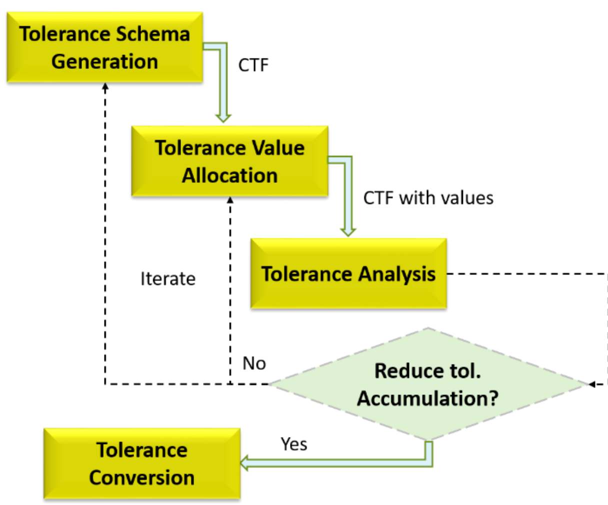


Figure 8 - 9: Proposed Iterative Schema Improvement System

REFERENCES

[1] Henzold, G., 2006, Geometrical dimensioning and tolerancing for design, manufacturing and inspection: a handbook for geometrical product specification using ISO and ASME standards, Butterworth-Heinemann.

[2] LU, S. C. Y., and Wilhelm, R. G., 1991, “Automating tolerance synthesis: A framework and tools,” J. Manuf. Syst., **10**(4), pp. 279–296.

[3] Hejazi, S. M., Biswas, D., Venkiteswaran, A., Shah, J. J., and Davidson, J. K., 2016, “Automated 1st Order Tolerancing: Schema Generation,” ASME 2016 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE 2016, ASME, Charlotte, NC, p. 14.

[4] Krulikowski, A., 1998, Geometric Dimensioning and Tolerancing, Cengage Learning.

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

[6] Standard, I. S. O., 2004, “1101: 2004 Geometrical Product Specifications (GPS)--geometrical tolerancing--tolerances of form, orientation, location and run-out,” Int. Organ. Standarization, Geneva.

[7] Shen, Z., 2005, “Development of a framework for a set of computer-aided tools for tolerance analysis.”

[8] Haghighi, P., Mohan, P., Kalish, N., Vemulapalli, P., Shah, J. J., and Davidson, J. K., 2015, “Toward automatic tolerancing of mechanical assemblies: first-order GD&T schema development and tolerance allocation,” J. Comput. Inf. Sci. Eng., **15**(4), p. 41003.

[9] Mohan, P., Haghighi, P., Vemulapalli, P., Kalish, N., Shah, J. J., and Davidson, J. K., 2014, “Toward Automatic Tolerancing of Mechanical Assemblies: Assembly Analyses,” J. Comput. Inf. Sci. Eng., **14**(4), p. 041009.

[10] Whitney, D. E., 2004, “Mechanical Assemblies: Their Design, Manufacture, and Role in Product Development,” Adv. Manuf., p. 517.

[11] Mantripragada, R., and Whitney, D. E., 1998, “The Datum Flow Chain: A systematic approach to assembly design and modeling,” Res. Eng. Des., **10**(3), pp. 150–165.

[12] Thornton, A. C., 1999, “Mathematical framework for the key characteristic process,” Res. Eng. Des. - Theory, Appl. Concurr. Eng., **11**(3), pp. 145–157.

[13] Haghighi, P., Mohan, P., Shah, J. J., and Davidson, J. K., 2014, “Automatic Detection and Extraction of Tolerance Stacks in Mechanical Assemblies,” ASME 2014 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, pp. V02BT03A018–V02BT03A018.

[14] Lai, D. K. L., and Yuen, M. M. F., 2012, “Vector based datum transformation scheme for computer aided measurement,” Comput. Aided. Des. Appl., **9**(3), pp. 289–305.

[15] Anselmetti, B., 2006, “Generation of functional tolerancing based on positioning features,” CAD Comput. Aided Des., **38**(8), pp. 902–919.

[16] Shah, J. J., and Rogers, M. T., 1993, “Assembly modeling as an extension of feature-based design,” Res. Eng. Des., **5**(3-4), pp. 218–237.

[17] Rao, S., 2013, Reconciling GD&T Rules: RECON vs ASU rule set.

[18] Yu, K. M., Tan, S. T., and Yuen, M. F., 1994, “A Review of Automatic Dimensioning and Tolerancing Schemes,” Eng. Comput., (10), pp. 63–80.

[19] Requicha, A. A. G., 1983, Representation of Tolerances in Solid Modelling: Issues and Alternative Approaches.

[20] Journal, I., Robotics, O. F., Requicha, a., and Chan, S., 1986, “Representation of geometric features, tolerances, and attributes in solid modelers based on constructive geometry,” IEEE J. Robot. Autom., **2**(3), pp. 156–166.

[21] Requicha, a. a. G., 1983, “Toward a Theory of Geometric Tolerancing,” Int. J. Rob. Res., **2**(4), pp. 45–60.

[22] Ingham, P. C., 1977, “Computer-aided analysis of engineering tolerances,” Aston University.

[23] Wu, Y., Shah, J. J., and Davidson, J. K., 2003, “Computer Modeling of Geometric Variations in Mechanical Parts and Assemblies,” J. Comput. Inf. Sci. Eng., **3**(1), p. 54.

[24] Shen, Z., Shah, J. J., and Davidson, J. K., 2008, “Analysis neutral data structure for GD&T,” J. Intell. Manuf., **19**(4), pp. 455–472.

[25] Mohan, P., Haghighi, P., Shah, J. J., and Davidson, J. K., 2014, “Automatic detection of directions of dimensional control in mechanical parts,” ASME 2014 International Manufacturing Science and Engineering Conference collocated with the JSME 2014 International Conference on Materials and Processing and the 42nd North American Manufacturing Research Conference, pp. V001T04A005–V001T04A005.

[26] Vemulapalli, P., Mohan, P., Shah, J. J., and Davidson, J. K., 2014, “User Defined Assembly Features and Pattern Recognition From STEP AP203,” ASME 2014 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, pp. V01AT02A067–V01AT02A067.