3-D Conformance Analysis of Manufacturing Plans Using M-Maps,

by Explicating Formal GD&T Schema from the Process Plan

by

Payam Haghighi

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

Approved May 2015 by the
Graduate Supervisory Committee:

Jami Shah, Chair
Joseph Davidson
Yi Ren

ARIZONA STATE UNIVERSITY

August 2015

ABSTRACT

A process plan is an instruction set for the manufacture of parts generated from detailed design drawings or CAD models. While these plans are highly detailed about machines, tools, fixtures and operation parameters; tolerances typically show up in less formal manner in such plans, if at all. It is not uncommon to see only dimensional plus/minus values on rough sketches accompanying the instructions. On the other hand, design drawings use standard GD&T (Geometrical Dimensioning and tolerancing) symbols with datums and DRFs (Datum Reference Frames) clearly specified. This is not to say that process planners do not consider tolerances; they are implied by way of choices of fixtures, tools, machines, and operations. When converting design tolerances to the manufacturing datum flow, process planners do tolerance charting, that is based on operation sequence but the resulting plans cannot be audited for conformance to design specification.

In this thesis, I will present a framework for explicating the GD&T schema implied by machining process plans. The first step is to derive the DRFs from the fixturing method in each set-up. Then basic dimensions for the features to be machined in each set up are determined with respect to the extracted DRF. Using shop data for the machines and operations involved, the range of possible geometric variations are estimated for each type of tolerances (form, size, orientation, and position). The sequence of manufacturing operations determines the datum flow chain. Once we have a formal manufacturing GD&T schema, we can analyze and compare it to tolerance specifications from design using the T-map math model. Since the model is based on the manufacturing process plan, it is called resulting T-map or m-map. Then the process plan can be validated by adjusting parameters

so that the m-map lies within the T-map created for the design drawing. How the m-map is created to be compared with the T-map is the focus of this research.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Jami Shah for his supervision and support throughout this work. All of the progress made and things I learned would not have been possible without his valuable guidance. I would also like to thank Dr. Davidson and Dr. Sugar for taking out time to serve on my committee.

I am deeply thankful to my colleagues at DAL lab for their friendship, support and guidance specially Prashant Mohan.

I also want to thank my mom and dad for their endless support and my dear Maryam for her heartwarming attention.

Finally, I would like to acknowledge the financial support from NSF (national Science Foundation), Grant No. CMMI-0969821.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

CHAPTER 1: INTRODUCTION

1.1 Background

Machining process plan is a sequential instruction of machining processes and tooling to meet the detailed design specification for a given part. Process planning is a labor-intensive and time-consuming activity that requires the specification of detailed step by step instructions to personnel on the shop floor about all operations and manufacturing resources to be used in production. There are many factors to be considered in process planning, such as tool selection, machine selection, machine-error considerations (e.g. feed rate, tool approach direction, etc.), fixturing methods, time scheduling, and cost modeling.

The manufacturing process plan should satisfy the design specifications communicated in the form of design drawings or CAD models. Thus, process plans must satisfy the allowable range of dimensional variations permitted by design GD&T specs. Process planners do this based on personal experience, or rules of thumb. Process planning is variant, generative or the combination of them. Variant is when the planner retrieves and modifies an existing plan for a similar part, and Generative is when the planner generates a new plan from scratch. Thus, preparing a process plan included retrieval and manipulation of a great deal of information from many sources, including established standards for manufacturing, machinability data, machine capabilities, tooling inventories, stock availability and existing practice.

Nowadays, many automated or semi-automated CAPP (Computer Aided Process Planning) software are developed to help the process planner create manufacturing plans, but still an automated system that can perform the complete task, from creating the plan to doing iterations and coming up with the optimal process plan do not exist. CAPP software are developed to help the process planner in three main areas. The first area is to help the planner develop the process plan interactively and doing tasks such as: selecting the tool, creating the tool path, deciding on the number of passes required, creating the G code for CNC machines etc. The second area is related to tolerance and manufacturing errors, such as automatically translating GD&T specification to +/- values for the plan and doing tolerance analysis such as Tolerance charting. The third area is to determine how to fixture the part for each machining setup, which in literature is addressed as Computer Aided Fixture Design (CAFD)

The goals of this research do not really lie in any of the three areas mentioned above. We neither want to create the process plan in CAD/CAM nor doing tolerance transformation. We want to audit the generated process plan from the GD&T point of view, by extracting the necessary information from the Process plan, present it in a formal schema that is familiar for everyone in the Design and Manufacturing industry, and check to see if the process plan is in conformance with the design tolerance limits

1.2 Problem definition

Process planners generate plans based on design specifications. Their choices of fixturing, tools, machines and finishing operations are based on achieving the desired level of precision. Their decisions are also based on the tools available in the machine shop, thus they sometimes follow the process plans for similar parts in their company archive. Company-specific codes for machines, tools, operations are used along with textual instructions and informal sketches in creating process plans. Standard symbols, like GD&T symbols of design drawings, are typically not used in process plan documents to represent different types of permissible errors; instead less formal +/- annotations are used on selected sketches or in the textual instructions. All of these result in non-standard process plans that are for human operators only and are not computer machines readable.

In order to clearly show the differences in standard representation of design drawings and non-standard presentation of process plans, a sample of each one in Figure 1 and 2 are compared. Figure 1 shows a design drawing which contains GD&T specifications using datums and tolerance frame symbols in conformance with ASME and ISO standards [1, 2]. This stands in contrast to a page taken from a process plan; the first stage of machining a casted connecting rod (Figure 2). In this figure, Datum reference frames (DRF) are not explicitly specified and standard symbols are not used.

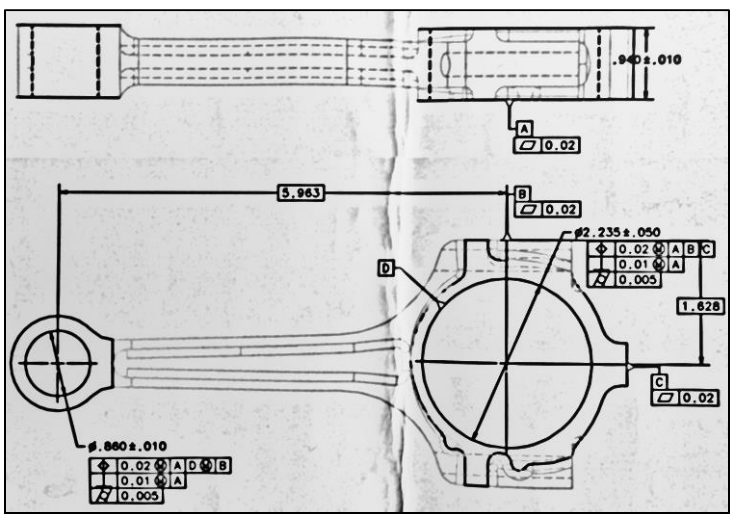
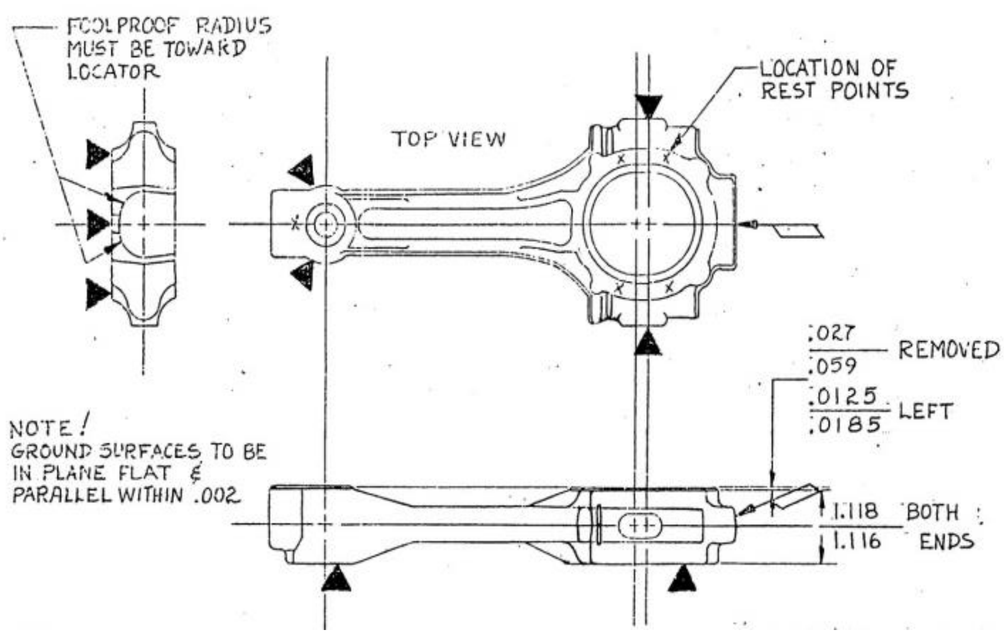


Figure 1: *A Design drawing of a part with formal GD&T Specifications*

OPERATION	DEPT. NO.	MACHINE NAME AND NO.	FIXTURES AND DIES	TOOLS
Rough grind both sides to	811	Blanchard #18-A2	(24) Station Work	Tool Layout
1.095/1.092 Dim.		Grinder #11704-8	Holding Fixture	3172338 KL-1
			3172338-KF-1	Sheet #1
MACHINE DATA		Machine Layout		
		3172338-KM-1	Clamp Lever	Spindle #1 Grinding
Spindle Speed 1200 RPM			3172338 KF-1	Wheel 18" x 5" x 12"
Wheel Speed 5650 SFPM		Floor Plan	Detail #9	89A46 E11 B31GRDCS
Table Speed (1) Rev. = .93 Min.		3172338-KM-1		C.N. #02-200-5934
Load & unload during machine		Sheet #1	(24) Clamp Jaws	
cycle.			3172341 KH-36	Spindle #2 Grinding
12 Stations for 1st Side		Wiring Diagram	C.N. #92-275-3600	Wheel 18" x 5" x 12"
12 Stations for 2nd Side		3172338 KM-1		A89A60 G +11 BEXDCS
		Sheet #2	(24) Locating Block	C.N. #02-200-5937
M.C.T./Piece = .080 Min.			3172341 KH-37	
Gross - 750 pcs./hr.		Worm Gear D-48437	C.N. #92-275-3700	
		C.N. #55-383-7837		
NOTE: Manually dress both			(24) Locating Block	
grinding wheels		Pinion Gear D-48436	Pin End	
every shift		C.N. #55-383-7836	3172341 KH-39	
Reset wheel height			C.N. #92-275-3900	
		Drive Gear D-48432		
		C.N. #55-383-7832		
		Adaptor Plate 18"		
		Gardner Machine Co		
		#D22743		
		C.N. #55-387-6250		

a) Process plan sample sheet



b) Process plan sample sketch

Figure 2: *First step in the process plan of machining the connecting rod.*

Table 1 gives a summary of differences in design and manufacturing tolerance specification. Machining is typically done in multiple setups, requiring parts to be fixtured and oriented in ways that provide proper cutting tool approach and unobstructed access to the features being machined in that setup. The setups used by process planners often result in datums and datum reference frames (DRF) that are different from datum features and DRFs specified by designers. This may be to simplify the fixturing and machining steps or to achieve desired accuracy based on the tools and machines and their capabilities and precision in the shop floor. Features created in one setup may be used as datums in a different setup. These datums may be transition features, which are neither on the stock work-piece nor on the finished part. The foregoing implies that the manufacturing datum flow chain is quite often different than the design datum flow.

Table 1: *Comparing of design drawing and conventional process plan*

Designs (formal GD&T)	**Process plan (implied GD&T)**
DRFs explicitly shown	DRFs are implicit in setups, fixtures
Formal GD&T frames	At most, +/- for dimensions, No GD&T
Datum flow chain directly extracted	Datum, and flow chain implicit , distributed
Consolidated info, in single Drawing	Distributed info (in multiple steps/pages)
Drawings represent final parts	Plans represent many transitions
Many tolerance analysis methods used (1D/2D/3D)	Mostly 1-D tolerance charts used by process planners

This begs the question: how can a party, other than the planner himself, verify if the process plan will meet the desired design tolerances? For this purpose, one would have to make explicit the tolerancing implied by a machining process plan. In order to trace the errors accumulation to verify, he needs to construct a manufacturing variability chain. This is due to the fact that machined features on the final part result from a sequence of operations, some even done in multiple setups. Also the fixturing features may or may not be the same as datums used in design GD&T.

Besides, in current industry practice, there seems to be little time to objectively determine the goodness of a plan. In automated CAPP systems, minimization of production time appears to be the only measure of goodness used, and even that is applied to a few alternatives that have been generated in an ad-hoc manner. Inefficiencies in creating a process plan come from the fact that it is a trial-and-error process (Figure3). First, design tolerances are translated into manufacturing tolerances for individual machining operations. Then a tentative process plan is made based on personal experiences and knowledge or based on company- specific practices for similar parts available in the library of the shop floor. Then a few parts are machined following the instructions in the trial (initial) process plan. These sample parts are inspected, and based on their evaluated quality the trial process plan is modified. Possible modifications include changing the machining processes, the production equipment, the locating schemes, and/or the sequence of operations. This new process plan is also tested and modified until satisfactory results are obtained. It seems that we can decrease the time to get to an acceptable final process plan; if we audit the process plan itself and make sure that the errors in the resultant product based on the process plan info will not exceed the design requirements before the trial-and-error.

In order to verify a process plan and check if design tolerance requirements are met, traditionally planners used to do 1D analysis. But the 1D approach does not account for

DRFs, datum precedence, or tolerance-zone. Also since some error contributors are not aligned with the direction of analysis, they are ignored, which may yield incorrect results. Later process planners started to employ other approaches, in some cases supported by computer software.

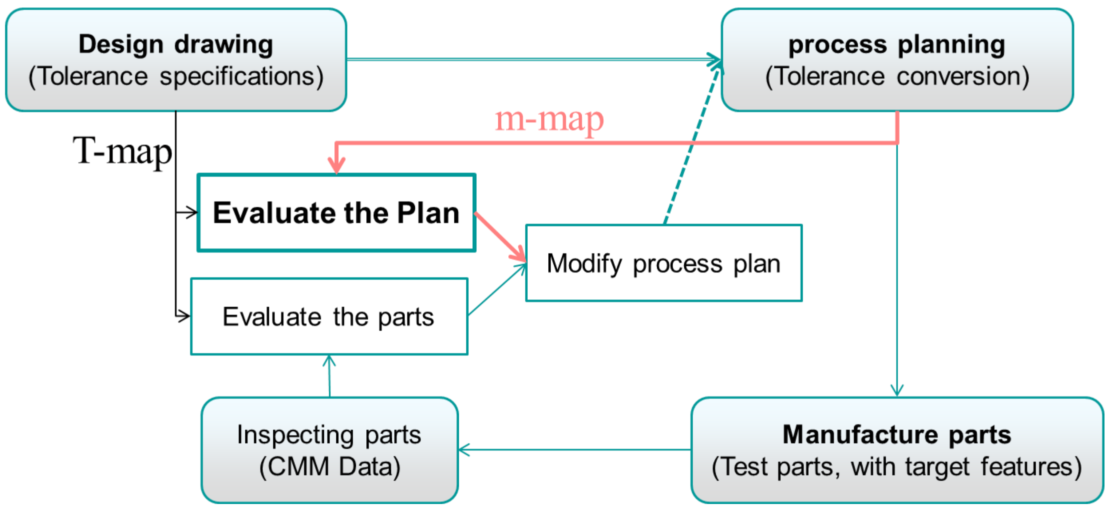


Figure 3: *Overview of Process Plan evaluation*

These include the 1D manual charting method, which does not have the shortcomings of the traditional 1D approach mentioned above. Tolerance charting is often taught in ASME professional development classes and in both design and process planning versions, it is practiced primarily as manual procedures. Later there have been a few attempts at automating the charting method and then trying to automate process planning or making interactive computer tools for process planning. Although, unlike the process planner's 1D analysis method, such 1D chart take into account all rules and tolerance types as indicated in Y14.5 standard, still it cannot do statistical analysis and treat contributors in different directions that are not coupled. In order to achieve proactive tolerance control in process planning, and to save the effort of trial-and-error, we propose an innovative method to verify a process plan in terms of machining error prediction with respect to tolerance specification. Error prediction is based on actual machining error synthesis and error propagation tracking through the multiple processes.

We suggest extracting the process plan variations and to use the same representation schema as used on the design drawing. Then we can easily evaluate the error propagation in the process plan, and check to see if it exceeds the allowable limit.

1.3 Approach overview

In order to make it possible to comprehensively "audit" the error accumulation from a process plan, and be able to do 3D tolerance analysis of manufacturing flaws, we need to perform three tasks:

1. Tolerance Explication: make explicit, the error accumulation implied by the combination of setups, fixtures, operations, and machining allowances in the process plan, using GD&T specifications for tolerances recommended by the standards [1, 2],
2. Manufacturing error Accumulation (m-map): construct manufacturing feature variability chains
3. Tolerance Conformance Checking: Perform 3D stack analysis on transformed accumulated error (m-maps) in the explicit scheme to determine conformance with design specifications (T-maps).

Once process plan errors can be represented with GD&T symbols in the same way as design, we can use the same 3D analysis tools to automatically extract tolerance chains and perform both worst case and statistical analysis.

Figure 4 shows a flow chart that outlines the major steps that need to be carried out to achieve the goals. The input to the system includes the geometry of the part in each step, as it evolves from the raw stock to the finished item, in CAD formats along with the machines and tools used. Also the fixturing methods along with the faces of the part that contact the fixture at each setup are needed. For tolerance explication, we need to extract and collect the information for geometrical and dimensional errors information that are implied in each stage of the manufacturing process. This starts with the identification of implied DRFs for each setup based on the method of fixturing. Then we determine the sizes and location dimensions with respect to the found DRFs by analyzing the geometry model (CAD file) of the part at each stage. Then, using error models for each machining procedure, we find the range of expected variations. As a result, dimensional and geometrical errors will be derived and represented in a standard format that we label Process-Plan Constraint Tolerance Feature Graph (PCTF), a data structure developed previously [13]. It stores the temporary geometry information of the part in process, the exposed faces (machined faces in each step), and the errors related to each face and feature. This data structure makes it possible to create the manufacturing map (m-map) of each type of error corresponding to each feature, in the same way that we create the T-maps for design tolerances [16].

In the second task, geometrical errors that are just local such as size and profile will be omitted, because they are not depending on any reference frame. Then we track the datums for other types of error on each feature in all stages looking for their transformation and its order. If there is a datum change, it means that there is a difference between the DRF for design and process plan which needs to be resolved. Thus, we have to do datum transformation and create the m-map for the corresponding errors. To sum up the effect of different errors in multiple steps, they will be stored and superimposed into a virtual part that has all the final and intermediate faces of the part through machining steps with errors assigned to them.

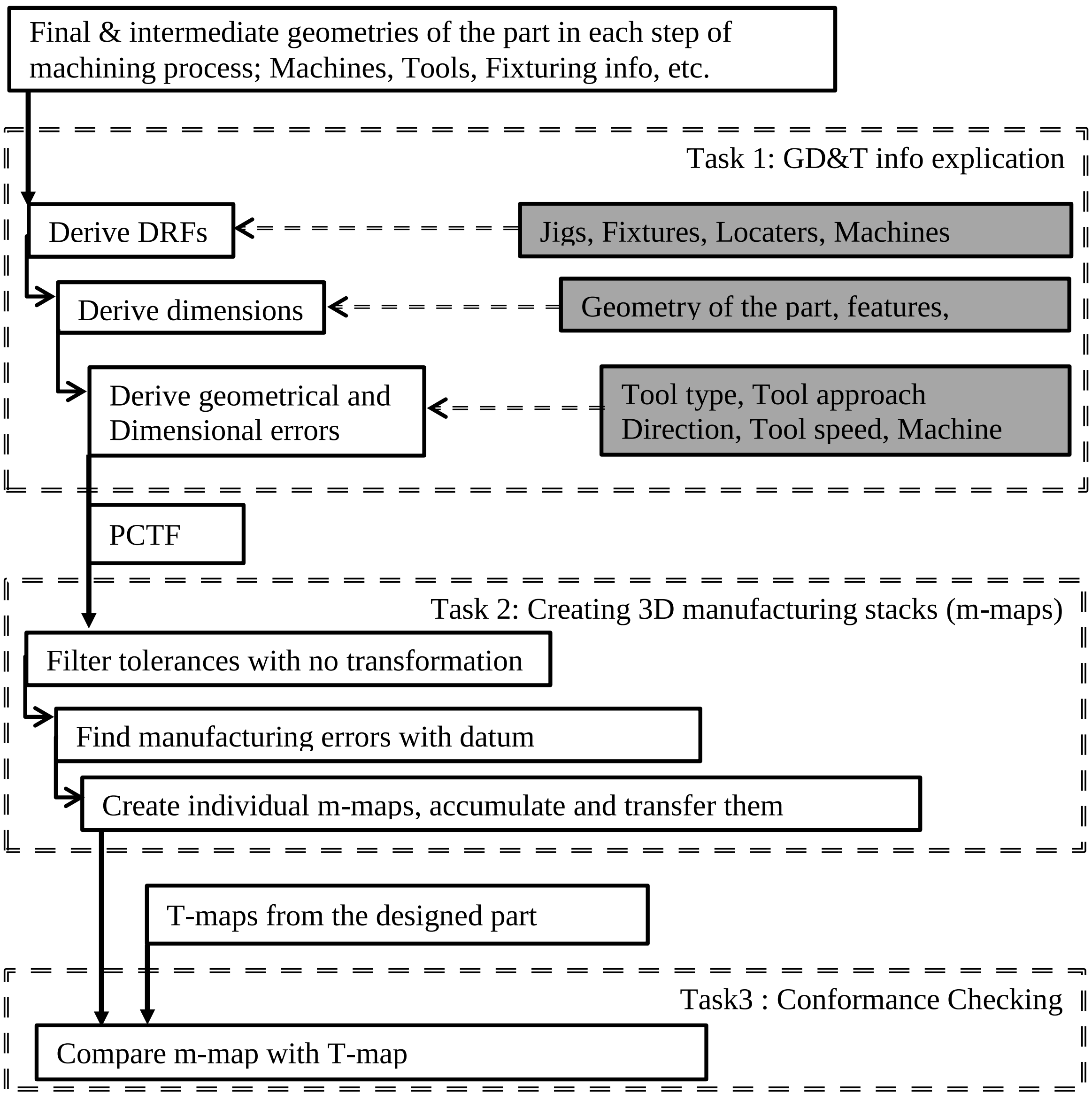


Figure 4: *Overview of computational procedure*

In task 3, in order to do 3D conformance analysis, we want to model the errors with T-map math model and do accumulations to derive m-maps. While, T-map is a metric model that represents the extent of allowable variation specified by each tolerance class, m-map represents the variations of errors in machining processes.

In order to do accumulation with m-maps, we have adapted Minkwoski sum which was initially proposed for summation of T-maps but the mathematical method is only developed for planar faces to date. The same method can be executed, since both maps are topological math models in an n-dimensional point space representing geometrical variations.

Also, the transformation of m-maps is limited to cylindrical and planar faces. It is noteworthy that with spit of the fact that size errors would not accumulate (has no m-maps because has no datum), but the ones that are in datum transformation will be taken into account. Finally, in Task 3, we compare the m-maps developed in previous tasks to compare them with the T-maps that are created based on the design drawing of that part with full GD&T schema. Here we compare the m-map of a feature and its corresponding T-map, to see if the m-map fits inside it. If it does, it means that the design specification is met; if not, then we have to see which contributor is causing the problem, and trace it back in the process of m-map creation to see where that error have had the biggest contribution and how can it be reduced.

1.4 Thesis Organization

In this thesis, the main focus is on GD&T explication and m-map construction tasks. In the current chapter the reason of this study will be discussed and in the second chapter a Literature review of this topic will be summarized. Later, in chapter 3, the process of extracting the GD&T info is discussed. Then the inputs that are needed, the libraries developed, details of each step, and methods and algorithms developed will be presented.

In chapter 4, the data structures and the information that are handled to be used in each step will be discussed. Finally, each step of the information extraction and explication phase with a sample part process plan will be demonstrated in chapter 5. At the end, in chapter 6, I will do a summary and suggestions for future work.

1.5 Scope of the work

The developed software can read any CAD geometry for different steps of machining of a part, in the form of ACIS ".sat" file. After solving the unique ID problem automatically, all types of errors derived from the process plan can be assigned to geometry in the testbed interactively. All geometry and geometrical errors information will be investigated and the ones that are needed for conformance checking of error accumulation will be stored in a data format that can be analyzed with the same methods and software developed for tolerance analysis of parts.

Since the work in this thesis covers many areas in the field of CAD/CAM research, some details around the main focus are limited but they serve their purpose to prove the concept of the work. Some of these limitations are discussed in following paragraphs.

Although extracting the GD&T errors are not automated in this research but it is discussed in detail in a way that it can be automated. The GD&T extraction task is an interactive task in the current state of the work which needs to be done by the user based on the two tables provided to them. The first table shows different fixturing schemes, but the fixturing faces are limited to planar and cylindrical faces. Also the fixuring tools are Vise, Clamp and Machine Tables for planar faces, V-Blocks for outer cylindrical faces, and Long Pins and

Diamond Pins for short and long inner cylindrical holes respectively and finally pin locaters for point locating

The T-map technology for modeling Geometrical and Dimensional tolerances are not fully developed for all types of tolerances. The transformation the Tolerance datums, which will change the T-map, is dependent on the math models developed for this purpose which is limited to planar and cylindrical faces based on Jiang work [22]. For summation of these point spaces, it is proposed to use the Minkowski sum method, which is mathematically well defined, but adaption and implementation of it is limited

CHAPTER 2: LITERATURE REVIEW

To determine a suitable tolerance scheme and allocate values, designers conduct tolerance analysis, which may range from 1D (one dimensional) manual tolerance charts for worst case to statistical analyses based on Monte Carlo simulation using state of the art CATS (Computer aided tolerance software) packages[3,4]. Traditionally, process planners only do worst case analysis with 1D dimensional stacks (position converted to +/- dimensions and size tolerances); they do not include geometric tolerances in stack analysis [5]. Turned parts are analyzed in radial and axial directions separately; prismatic parts in three orthogonal directions, or more, if there are angled faces. The 1D approach does not account for DRFs, datum precedence and zone tolerances. Also, since some contributors are not aligned with the direction of analysis, they are ignored, which may yield incorrect results. There have been a few attempts at automating 1D tolerance charts, in both design and process planning versions. This includes research by Ahluwalia and Karolin [6], Li and Zhang [7], Whybrew et al. [8], and Shen [9]. Besides, researchers have been trying to automate process planning and making interactive computer tools for it. Y. Zhang et al [10] proposed a computerized graph based setup/fixture planning using GD&T. Shah et al [11] have developed a dimensional model that facilitates the conversion of dimensions and tolerances from design models to machining features which are extracted automatically by a feature recognition system. For geometric tolerances, Thimm [12] explores a system that derives and rewrites alternative geometric and size design specifications with the aim of improving the manufacturability of a design. Polini and Giovanni [13] have proposed a model to incorporate different types of tolerances in manufacturing to do tolerance analysis. The objective of tolerance analysis is to check the extent of variation of a dependent dimension or clearance for a given GD&T scheme. Analysis approaches can be classified as 1D, 2D, and 3D, according to dimensionality; as worst-case or statistical according to the analysis objective. Popular analysis methods are manual 1D charts [14], linearized 2D/3D analysis, and Monte Carlo simulation [15]. For 3D tolerance analysis, Shah and Davidson introduced a math model that can describe all possible variations constrained by design or machining tolerances. T-Map is a hypothetical Euclidian point space model which the size and shape of it reflects all variation possibilities for a target feature. T-maps have been created for all types of geometric tolerances with Primitive T-Map elements. The accumulation the T-map in machining is called m-map. [16]

CHAPTER 3: CONCEPTUAL DESIGN OF SOFTWARE – PCTF

The ultimate goal in this research is to use the outputs of the Tolerance Explication module to do tolerance conformance analysis and verify the manufacturing process plans automatically. Therefore, the information extracted from process plans need to be organized in a formal machine readable format. For this purpose, a data structure is designed that can store and represents the explicated GD&T information of process plans in conjunction with the CAD model of the part to be used for analysis. Since the required data structure needs to store the Geometrical and Dimensional errors of a process plan and the geometry of the part in different steps, the data would be very similar to tolerances of a designed part. Thus the data structures developed previously in Design Automation Lab at Arizona State University, CTF and SCTF [21] graphs, are modified to create a new data structure called PCTF (ProcessPlan-Constraint-Tolerance-Feature) graph. This Data structure is very similar to its ancestors, so the details of the similar parts of them can be found in literature, but the main points and the differences are presented in this chapter.

Here the basic concepts and specifications of CTF is reviewed first. Then it is shown how the CTF is adjusted so that it can include and store the information needed for creating m-maps.

In order to lay the groundwork for the development of the GD&T data model, we consider the types of information, entities and relations needed to express GD&T in accordance with the standards. Size tolerances as applied to linear, radial or angular dimensions corresponding to parameters are related to features of size (i.e. holes, pins, slots, tabs, pockets, bosses, etc.). Therefore, a definition of FOS (feature of size) is needed along with its parametrization (radius, diameter, depth, etc.). Size tolerance specifies max/min values and can be expressed in a variety of ways: max/min parameter limits, nominal value and +/- variation which may be equal on both sides (i.e. equal bilateral size tolerance), unequal or unilateral. Any of these can be calculated for any of the other representations. Size tolerances are directly related to the corresponding size parameter, which are defined by the distance/angle between 2 lines and 2 planes or between the center and the boundary of a radial feature. Geometric tolerances are applied to given entities (edge, surface) or to features of size. There is a tolerance type, value and up to three datums if applicable. Modifiers may be applied to the tolerance value or tolerance zone shape. The geometric tolerances will control the orientation, location, shape (form), and profile of the tolerance entity, relative to datum reference(s) of frame.

This leads to the following requirements for entities that need to be supported: face (planar or freeform surface), line, point, and features of size (cylindrical, spherical, tab/slot, etc.) and relations that must be supported: size, orientation, locations, and shape, where a shape relation control the intrinsic form of the feature, and it could be one or several linear, and orientation relations.

3.1 Overview of CTF model

In this section, I will go over the definitions of the features, constraints and the degree of freedom, and basic concepts used in *Constraint-Tolerance-Feature-Graph-Based Model* (or *the CTF Model* for short). The model content includes nominal geometry (features), constraints (including dimensions, mating conditions, assembly constraints), tolerances (including datum reference frames), and degrees of freedom (DoFs). The nominal geometric information of the model is composed of the geometric primitives, and their combinations. Each geometric entity has certain inherent DoFs to be controlled.

3.1.1 Real features and trimmed features

A feature is a stereotypical shape defined by specific topology, geometry, and constraints. As a matter of fact, real features (i.e. toleranced surfaces on a part, not features with nominal parameters) can be of any type and shape. There are no abstracted primitives like a pure point feature, an infinite line feature, or an infinite plane feature. Instead, what we see are trimmed features, or approximated features that are idealized from real ones. In tolerance analysis, it is necessary to approximate the real features by trimmed features, which are defined as the features simplified or abstracted from the real ones with minor cutouts and protrusions suppressed.

For example, all the planar features (assuming the toleranced planar surfaces are involved in a tolerance analysis, rather than otherwise like the tolerance slot, or hole, or hole pattern) in Figure 5, can be approximated by an ideal rectangular planar feature. Indeed, these real surfaces, with their cutouts and/or protrusions, would most likely to be manufactured (e.g. milled) at one setup. The presence of these minor cutouts and/or protrusions would not affect, in most cases, the choice of the manufacturing process. Therefore, this abstraction from the real feature to the ideal ones is not only necessary but also reasonable, because its effect on the simulation result is negligible.

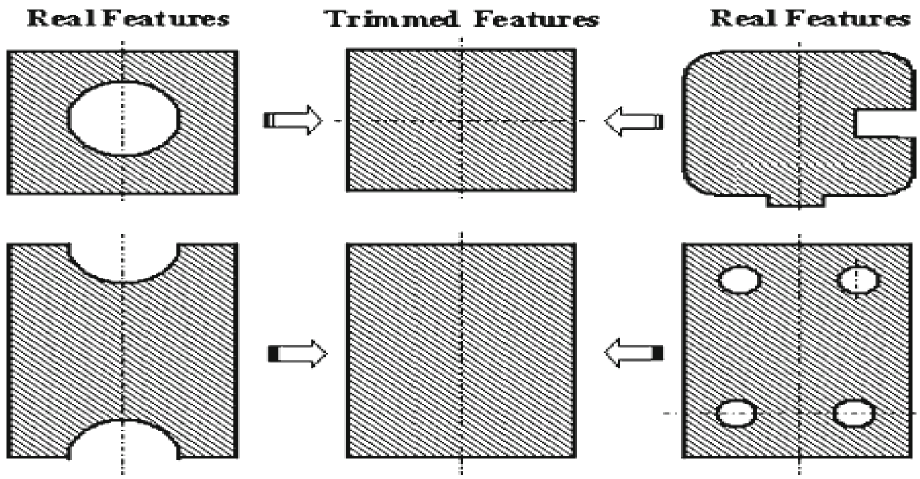


Figure 5: *Trimmed feature vs. Real features.*

To view the CTF-Graph Based Model, the user would like to see the real features or at least the trimmed features, but not the primitives or their combinations.

3.1.2. Constraints and metric relationships:

A geometric constraint in GD&T corresponds to a basic metric relationship between the primitives. Each metric relationship may be expressed in one or more analytical equations from the analytical geometry.

Different metric relationships exist between the primitives, and constrain the DoFs of geometric entities w.r.t. each other (Wu et al. 2003). We use the same representation, i.e. (X_i, Y_i, Z_i) for points, $A_i X + B_i Y + C_i Z + D_i = 0$ for planes, and $(X-X_i)/p_i = (Y-Y_i)/q_i = (Z-Z_i)/r_i$ for lines. But for example, when a feature of size is involved, the feature's size must be taken into account; or in order to compute the distance between a point and a circle (i.e. a special plane), the coincident relationship between a point and a plane can be used to check if the point is coincident with the circle defined plane.

Geometric constraints may be specified dimensions, mating conditions, or geometric relations, such as perpendicularity, parallelism. For size features, size constraints can be directly attached to the features themselves. A geometric constraint may have a measurement direction associated with it, but it is not always the case. For instance, if a constraint involves a *plane*, its measurement direction is the plane normal. If it involves two parallel but non-coincident *lines*, its measurement direction is the direction that passes through and perpendicular to the lines. If a constraint involves two coincident *lines* only, its measurement direction is the *line direction*. A size constraint of a sphere (i.e. a *point* feature) will not have a fixed measurement direction. Other cases are not enumerated here. The key point is that a constraint, especially a dimension, requires a measurement direction, and that direction will depend on how this constraint is actually measured in manufacturing and inspection. The user can specify the measurement direction for a geometric constraint by specifying the datum of this measurement.

3.1.3 Entity degrees of freedom

The geometric primitives or the trimmed features have their respective active DoFs and invariant DoFs. To limit the variation of a particular feature, its active DoFs should be fully controlled within certain ranges, i.e. tolerances. In other words, the tolerance specification should control the feature's variations along its active DoFs with respect to its datum reference frame. Indeed, a GD&T specification will control the corresponding active DoFs of the toleranced feature, and each datum, if any, will control some of them.

Since it is not in the scope of this work, there is no need to list all possible tolerance classes and how they control the active DoFs of the primitive features, i.e. point, line, and plane features in most scenarios. Instead, we will use the position tolerance of three datums on a line for demonstration. But interested readers can refer to (Shen 2005) for a complete coverage.

3.2. PCTF graph, Data structure

PCTF, similar to its ancestors (CTF & PCTF, Shen 2005 & 2008), can represent all the tolerance types in the standards, and can contain all the information that is needed for error analysis; but this time it is modified so that it can contain the geometry of the part in process, and the errors of the process plan derived assigned to the geometry. This means that we can store work-holding information and tolerances along with the geometry for each operation (step) in the process plan to able to do manufacturing variability analysis on the whole process plan. Here we will review the main characteristics of PCTF, how it is represented and implemented.

3.2.1 PCTF-Graph Implementation

The PCTF model has been implemented using the C++ language, and commercial geometric kernel *ACIS*. Written in C++, ACIS provides an open architecture framework for wireframe, surface, and solid modeling from a common, unified data structure.

With the attributed CAD model the PCTF-Graph Based Model can be automatically created, as explained below:

(1) Traverse the attributed CAD model to retrieve all the GD&T information, i.e. geometric constraints and the associated tolerances, mating conditions.

(2) Check all the GD&T data to find out all the geometric entities (real physical features) involved, and group the entities according to their owning parts.

(3) Create the general tree to capture the setup sequence information, and populate this tree in the order of part, constraint, tolerance, and DoF.

(4) Create the real physical features for the visualization purpose.

(5) Abstract the real physical features to the trimmed features, which correspond to the geometric primitives and their combinations. This is where the feature recognition technique can play and important role.

(6) Create all the trimmed features and populate all the GD&T data (geometric constraints and their dependent tolerances) to generate the PCTF Graph Based Model.

It is important to note that the same model is always created for the same attributed CAD model, regardless of what independent parameter is being analyzed. With the PCTF Graph Based Model automatically created, it is possible to conduct different types of tolerance analyses right on top of this model. See (Shen 2005) for how different types of tolerance analyses are performed driven from the same CTF model.

The PCTF-Graph, at the highest level, is a general tree. The data structures for this *general tree* are the *tree node* class and the *general tree* class. A general tree node has a data member variable *data* and three pointers to link current node to its *parent*, its *child* and its *sibling* nodes. It is also a template class, since the tree node is a template class. To traverse, modify and retrieve data from the general tree, various *access*, *utility* and *modifier* functions are defined as well.

Using the template general tree representation, general trees of different data types can be created, depending on the user-defined data type. PCTF has a nested *doubly-linked list* data structure. The general tree node contains the node's pointers, name and a CTF-Graph, thus or example the data type T is defined in a *CTF struct*. PCTF expands as shown in Figure 6.

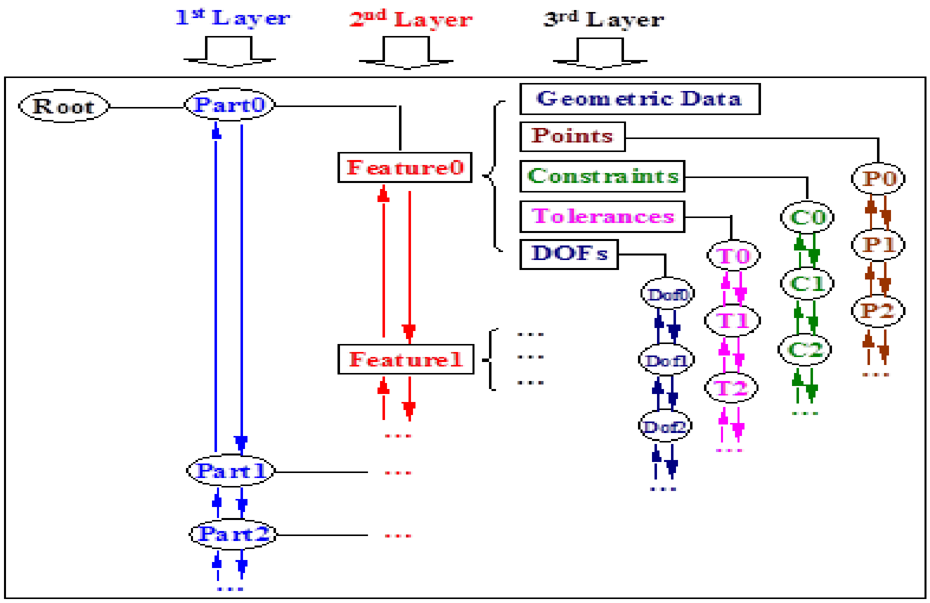


Figure 6: *CTF hierarchy*

At the very first layer, it is a list of "parts" which are the geometries of the part under process in different stages. At the second layer, each "part" is composed of a list of geometric "features" (of *CGeometry* type). At the third layer, each geometric "feature" contains its basic geometric data, a list of geometric constraints, a list of tolerances, a list of DoFs, and a list of associated points (i.e. a special *CGeometry*).

The whole model is created from top down, and lower level data is gradually populated when the higher-level data is available. The order is:

"*The general tree → parts → features → constraints → tolerances → DoFs*"

The lower-level data representations, such as those for *CGeometry* (or feature), constraint (C-graph), tolerance (T-graph), and DoF are discussed in the following sub-sections.

3.2.1.1 PCTF-Graph, Constraint structure

The constraints are represented by a C-Graph, an undirected graph with the involved trimmed features at the nodes and the geometric constraints as the arcs. This measurement direction is useful for traversing the C-Graph to detect the tolerance chain (Shen 2005; Shen et al. 2008). Note that the face ID numbers are automatically assigned when the model is created, and uniqueness of the face ID is guaranteed.

The geometric constraints in the Model are represented in different classes derived from the constraint base class *DAL_Geom_Cst*. Note that constraint type *eType* is an *enum* type data. Involved geometric entities are saved in the pointers (i.e. *gTarget_A*, *gTarget_B*) of the type *CGeometry*. A geometric constraint can have direction vector, defined as a *CCoordinate3D* structure.

3.2.1.2 PCTF-Graph, Feature data structure

Geometric information of the trimmed features needs to be encoded along with the GD&T information. Representation of the trimmed features is designed in such a way that it can link to the GD&T data and be supported by the geometric constraint solver, and can accommodate the requirements from the analysis processes. As pointed out in section "Real features and trimmed features", all geometric entities are resolved to *point*, *line* or *plane* from the DoF point of view. A point, line or plane can correspond to many different trimmed features. During the tolerance analysis, it is necessary to distinguish one from the other within the group of trimmed features corresponding to the same primitive entity. For instance, a *line* can represent a *pin*, a *hole*, a *cone*, and other revolved surfaces; but they each have their special attributes that a pure line does not have. In tolerance analysis involving a line, it is necessary to distinguish a *pin* or *hole* feature from a *cone* or *helix* feature. Therefore, the feature representation will recognize this difference, instead of just three types of geometry, i.e. *point*, *line*, and *plane*. Like the CTF graph, in the hierarchy of the feature classes the *CGeometry* is used as the base class, and other geometric types are all derived from it. This class is fully discussed in previous researches done by Shen et al. (2005)

3.2.1.3 PCTF-Graph, Tolerance data structure

The tolerance information in a CTF-Graph Based Model forms a tolerance-graph, called T-Graph. Unlike C-Graph, T-Graph is a directed graph with the toleranced geometric features at the nodes and tolerance specification as the arcs. For those tolerances that have no datum reference frame (DRF, i.e. the coordinate systems used to locate and orient a part feature (ASME 1994), the tolerance is attached to the geometric features itself.

Since a tolerance is used to control the variation of a certain geometric constraint, it depends on the corresponding geometric constraint. In other words, a tolerance cannot exist without its corresponding geometric constraint. For instance, a dimensional plus/minus tolerance has no meaning if the corresponding dimension does not exist. Therefore, C-Graph and T-Graph can be combined together to form a directed constraint-tolerance-graph, i.e. CT-Graph. A CT-Graph is also referred to as constraint-tolerance-feature-graph (CTF-Graph), since it has the geometric features (or trimmed features) at its nodes. Indeed, the geometric features are indispensable components for a C-Graph and a T-Graph. The CTF-Graph does not include the machining process information, which is encoded in the PCTFGraph.

The tolerance information in the Model is represented in different classes derived from the tolerance base class *DAL_Geom_Tol*. Note that tolerance type *eType* is an *enum* type data. Involved geometric entities are saved in the pointers (i.e.*g Target*, *datum_A*, *datum_B*, *datum_C*) of the type *CGeometry*. Material condition modifiers (e.g. MMC, LMC, and RFS) are also saved as *enum* type.

3.2.1.4 PCTF-Graph, DoF representation

A DoF (in the kinematic sense) of a feature is represented in a *struct* using *CCoordinate3D* definition. All active DoFs of a feature is saved in a *doubly-linked list* in the feature itself. These activeDoFs are controlled by the corresponding tolerance(s) specified on this feature; therefore, for each tolerance object, there is *doubly-linked list* that contains all the DoFs this tolerance actually controls. The contents of the Tolerance class are: tolerance type, tolerance value (fValue), Diameter modifier (Diam_Symbol), Target feature pointer (eTarget), Target type (eTargetType), and a Linked List representing the DoFs. A union of all the DoFs associated with the tolerances specified on a feature should be equal to the set of DoFs contained in the list held by the feature itself.

3.2.2 PCTF-Graph Example

A sample PCTF is shown in Figure 8. It belongs to the machining process of the cap of the assembly shown in Figure 7, and it is created with the ASU_M-MAP_TESTBED by manually inputting the extracted information of the process plan. The information in the PCTF can be categorized in four sections: A, B, C, and D.

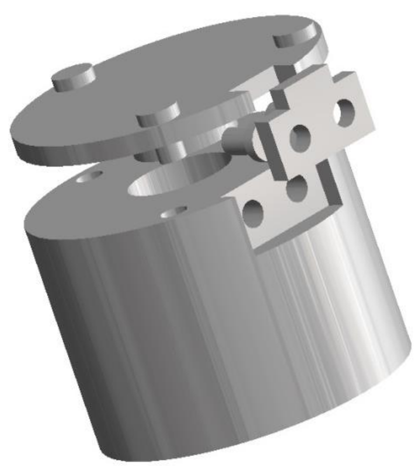


Figure 7: *Cap cylinder assembly CAD model*

Section A of this graph (line 1) contains the name and address of the B-Rep model file of the geometry of machined part after an specific step and the process plan errors for the exposed faces in GD&T format for the very step. So for each machining step, a B-Rep model needs to be associated with GD&T information derived in Task 1.

Section B of this graph contains information about the features in a Part and their data such as the type of the face, its axis or normal direction and position. Section B of this Graph extends from line #1 to #8 and gives us all the features which a part is made of. First line of section B says the part number followed by the line numbers which contain the information about the features of that part. If there is more than once occurrence of a feature they are named differently for example line #4, #5 and #6 shows that there are three rectangular planes in this part and they are named as *Face4, Face5* and *Face 6*. Also, Line #4 tells us that part 1 hole feature on one of its face which is named as face 3 (by the software internal modeling scheme which can be changed accordingly and has root point [-35.866, 4.026, 5], then the point [0, 0, 1] is the axial direction of the center of the hole, 3.3 is the radius of the hole, 6 is the height of the hole.

Numbering of features can be in any random order. The very important alteration from CTF to PCTF is that the exposed faces in each step will have a new ID and unaltered faces (fixed faces) will have the same ID as its previous steps among the PCTF of all steps. The algorithm to track the fixed faces and identify exposed faces for a part under process is given somewhere. Also since we are not dealing with assemblies, this section will always have one part.

```
#0= FILE('C:\ASU_M-MAP_testbed\sat\Cylinder Cap.sat');                                                    A

#1= PART('part2', #2, #3, #4, #5, #6, #7, #8);
#2= PIN('FACE1_of_part2', (-4.06647, 4.02592, 5), [0, 0, -1], 12.75, 6);
#3= RECTANGULAR_PLANE('FACE10_of_part2', (29.5174, 4.02592, 5), [0, 0, -1], 18, 6.66986, [0, 1, 0]);
#4= HOLE('FACE3_of_part2', (-35.8665, 4.02592, 5), [0, 0, 1], 3.3, 6);
#5= HOLE('FACE4_of_part2', (17.9335, 27.0259, 5), [0, 0, 1], 3.3, 6);                                     B
#6= HOLE('FACE2_of_part2', (17.9335, -18.9741, 5), [0, 0, 1], 3.3, 6);
#7= RECTANGULAR_PLANE('FACE5_of_part2', (29.5174, -4.97408, 8), [0, 1, 0], 6.66986, 6, [1, 0, 0]);
#8= RECTANGULAR_PLANE('FACE7_of_part2', (29.5174, 13.0259, 8), [0, -1, 0], 6.66986, 6, [1, 0, 0]);

#9= CST_ANGLE(90, #2, #3);
#10= METRIC_RELATIONSHIP(#9, CST_ANGLE, (90, #2[LINE(axis of PIN)], #3[PLANE]));
#11= CST_DISTANCE(31.8, #2, #4);
#12= METRIC_RELATIONSHIP(#11, CST_DISTANCE, (31.8, #2[LINE(axis of PIN)], #4[LINE(axis of HOLE)]));
#13= CST_DISTANCE(31.8277, #2, #5);
#14= METRIC_RELATIONSHIP(#13, CST_DISTANCE, (31.8277, #2[LINE(axis of PIN)], #5[LINE(axis of HOLE)]));
#15= CST_DISTANCE(31.8277, #2, #6);
#16= METRIC_RELATIONSHIP(#15, CST_DISTANCE, (31.8277, #2[LINE(axis of PIN)], #6[LINE(axis of HOLE)]));
#17= CST_DISTANCE(58.5102, #6, #4);                                                                       C
#18= METRIC_RELATIONSHIP(#17, CST_DISTANCE, (58.5102, #6[LINE(axis of HOLE)], #4[LINE(axis of HOLE)]));
#19= CST_DISTANCE(58.5102, #4, #5);
#20= METRIC_RELATIONSHIP(#19, CST_DISTANCE, (58.5102, #4[LINE(axis of HOLE)], #5[LINE(axis of HOLE)]));
#21= CST_DISTANCE(18, #7, #8);
#22= METRIC_RELATIONSHIP(#21, CST_DISTANCE, (18, #7[PLANE], #8[PLANE]));

#23=T_SIZE(#2, (nFI, 0.05, RFS));
#24= DOF(#23, (SIZE_DOF, SHAPE_DOF));
#25= T_ PERPENDICULARITY(#2, (FI, 0.2, MMC), PD(#3, RFS));
#26= DOF(#25, (SHAPE_DOF, #3, RDOF[1,0,0], RDOF[0,1,0]));
#27= T_FLATNESS(#3, (nFI, 0.12, RFS));                                                                    D
#28= DOF(#27, (SHAPE_DOF));
#29= T_DIMENSION(#7, (nFI, 0.1, RFS), PD(#8, RFS));
#30= DOF(#29, (#8, TDOF[0, 1, 0], RDOF[1, -9.28146e-014, 0], RDOF[0, 0, -1]));

#31= SETUP('setup_0', $)
#32= SETUP('setup _1', #4,#5,#6)
#33= SETUP('setup _2', #7,#8)                                                                             E
#34= PP_MODEL(#31, #32, #33)
```

Figure 8: *Example PCTF Graph for part in figure 7*

<u>Section C</u> contains all information about the constraints and metric relations, including the mating conditions (mating conditions in case of assembly). In this case section extends from line #9 to #22.

For instance, line #11 tells us that there is a distance of 31.8 between the features defined in line #2 and #4. Line #12 of this section tells us that the metric relationship is of type *CST_Distance* defined in line #11 which is between the axis of feature defined in line #2 and line #4. Similar structure is followed throughout this section. First there is a type of constraint its value and the involved features are defined. Then the metric relationship about the constrained is defined and then in next line the next constraint is defined. This process goes on till we have listed all the constraints and relationships for every part.

Section D of this PCTF graph contains all the error tolerances and DoFs (degrees of freedom) information of a part. Section D occupies the lines from #23 to #30 in this case. This section first defines a tolerance associated with a feature and its values and material condition.

The first line of this section (# 23) defines a tolerance of size on feature defined in line #2 which is of value ϕ (± 0.05) and is at RFS (Regardless of feature size).

Finally in Section E, we can see the list of the setups the part has gone through. This section is contain the exposing faces machine din each setup. Thus all the exposing faces that has gone through datum transformation will be stored and can be tracked in this section easily.

3.3. PCTF & GDT Testbed

Using the CTF-Graph Based Model, a tolerance analysis testbed has been previously developed in DAL. With the input of the hybrid attributed CAD model, the G&T info is stored in a Neutral Representation model (like PCTF-Graph) which can serve as the common data model for all types of tolerance analyses, such as automatic charting, simulation-based analysis and T-Maps based analysis.

I have modified this Testbed to be able to take in the CAD models of a machining part under process to interactively assign and allocate the GD&T info in each stage extracted from the process. With this Testbed, we can map the GD&T info to create m-maps like the T-maps we could create for design tolerances. This Testbed store the machining info for each step in multiple CTFs which will be compared at the end to create the PCTF. If a user interactively input all the extracted errors, it will create a textual representation of the PCTF. For this purpose, the user needs to input the machining variability (error limits/tolerances) of each machining operation. In our algorithm, each machining process is considered one step and the geometry of the part before and after the machining step needs to be available. So we need to have the CAD model of the part in process from the raw material or stack, till the final shape. The user then needs to select the datums, type of the error and its values in each step correspondent to each machine process. In this thesis, the application of the testbed, is shown with a help of an example in chapter 6.

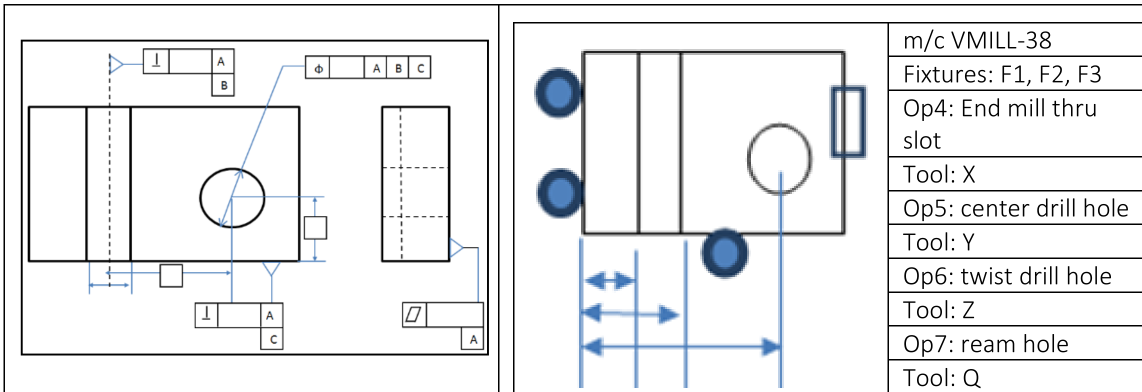
Recall the CTF-Graph Based Model contains the T-Graph, which is a directed graph representing the datum target relationship between different features in the Model. This directed graph facilitates downstream GD&T processing like tolerance analysis. Note that in reality, the arcs just contain the pointers to the corresponding objects (e.g. constraints, tolerances, mating conditions), which in turn, hold pointers to the target features and datum features. This way, the travel can be two-way. With such a model, it is trivial to find the target-datum relationship of all the features involved. If we want to perform a tolerance analysis, say create an m-map for a part, the simulation can start from the primary datum of the part (not a specific tolerance), i.e. the datum feature that does not reference any other features as datums. Then the simulation continues upwards along the target-datum relationship, in one or more paths, until all the involved features are varied.

Conclusion

In this chapter we have explained the representation model, the PCTF-Graph, which overcomes the shortcomings of the other models to handle all different types of machining error. This lean model holds just enough information that is needed for the representation and use of the data. To be specific, this neutral model contains all the geometric, constraint, assembling, and tolerance information needed for different types of tolerance analyses. This neutral model can be used to create m-maps which has the same representation with t-maps. Thus we can perform conformance analysis by comparing these two.

CHAPTER 4: TOLERANCE EXPLICATION

Tolerance explication requires the following details of the process plan in each setup: the before/after geometry of the part in process; the faces and features created; the machines and operations used; the method used to fix and locate the part in the machine; and part fixturing features/faces. Because the machine tools used affect the accuracy of the parts created, we need to supplement process planning information with machine error maps, as well. To illustrate these ideas, let us look at the simple example shown in Figure 9.



a) Design GDT spec of a part | b) Partial Plan

Figure 9: *Part design and process plan*

Figure 9a shows the design specification of a part with standard GD&T symbols (tolerance values and modifiers have been omitted on purpose in order to focus on the control schema only). The hole position is controlled by the DRF A-B-C, where A is the bottom face, B is the slot feature and C is the side face. However, both the slot and the hole are machined in the same setup according to the process plan shown partially in Figure 9b. This can be thought of as a macro-plan, i.e. pre NC code generation.

The machining DRF uses the left side face as the secondary datum instead of the slot used in the design DRF. Center drilling controls the position of the twist drill in the following operation. We need to rely on machine and operations accuracy to know exactly the variation to expect, as the process plan gives only nominal position and contains no explicit values for the position tolerance. This example illustrates the following issues for investigation:

- How to determine the implied DRF from setup and fixture specification
- How to convert +/- feature tolerances to geometric FOS (feature of size) position or orientation tolerance (e.g. the slot's position)
- How to transfer tolerances of Design DRF and Process DRFs in order to verify conformance (tolerance transfer problem)
- How to extract operation tolerances from machine, cutting tool and operation type

As stated before, the aim is to extract and represent the implied manufacturing tolerances in the same syntax as ASME Y14.5 standard tolerance frames. Apart from having clearly defined semantics, the rationale for this choice is that the same 3D tolerance analysis tools can now be used on manufacturing GD&T. This includes our tolerance T-map math model, T-maps [16]. In the next three sections we will discuss our approach in more detail. As shown in Figure 4 these steps include extracting datums, implied DRFs, dimensions and GD (Geometrical & Dimensional) errors.

4.1 Datum Extraction

Datums and datum reference frames (DRF) are used as references for measurements; they specify the direction of measurement and set up coordinate systems. Datums are theoretically exact points, axes, lines and planes. They are neither on the measured part nor on the gage blocks or inspection tooling; they are simulated by contact between the two. A DRF is a set of two or three mutually perpendicular features (planes, mid-planes, axes) that are derived from sufficient datum features or portions of them. Figure 10 illustrates the so-called 3-2-1 principle for simulating a DRF with three mutually perpendicular datums.

Since some types of errors (e.g., orientation, position) need a datum to be measured from we need to derive datums first. And since, datums correspond to the way the part is located and fixed in each machine; it can be derived from the fixturing features and fixturing method in each setup.

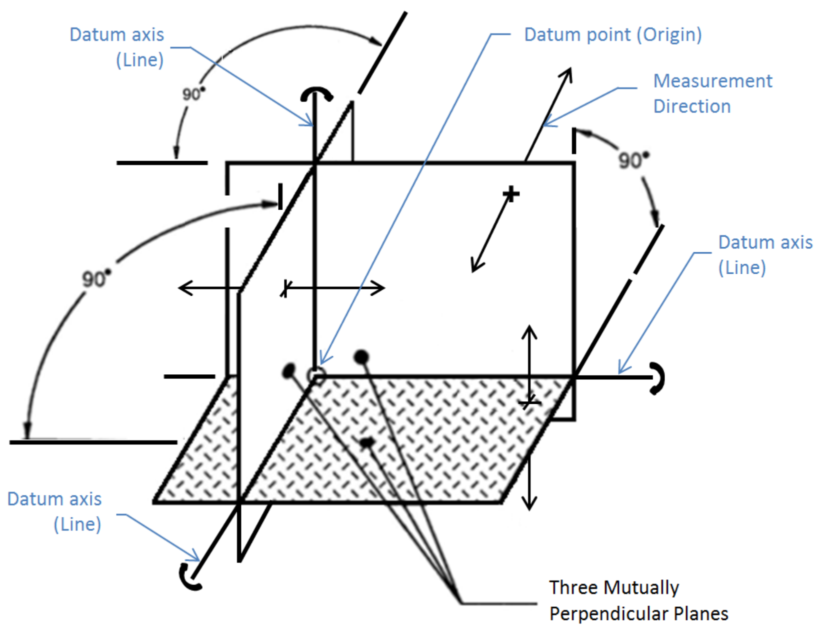
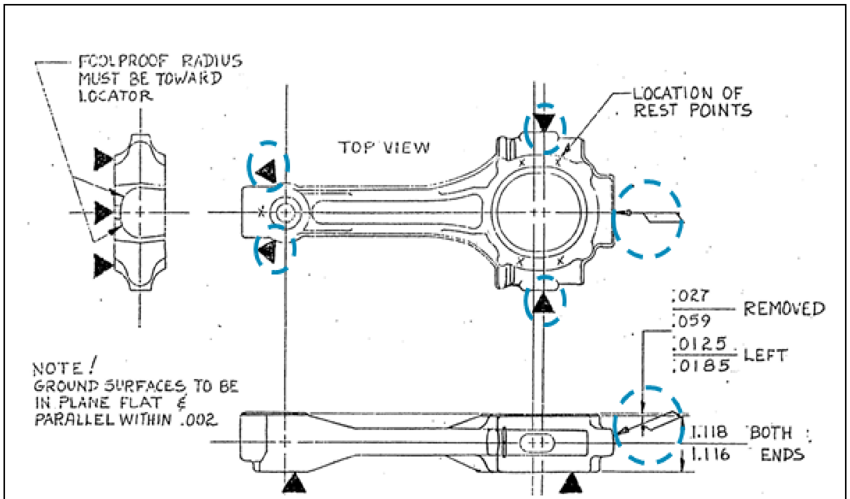
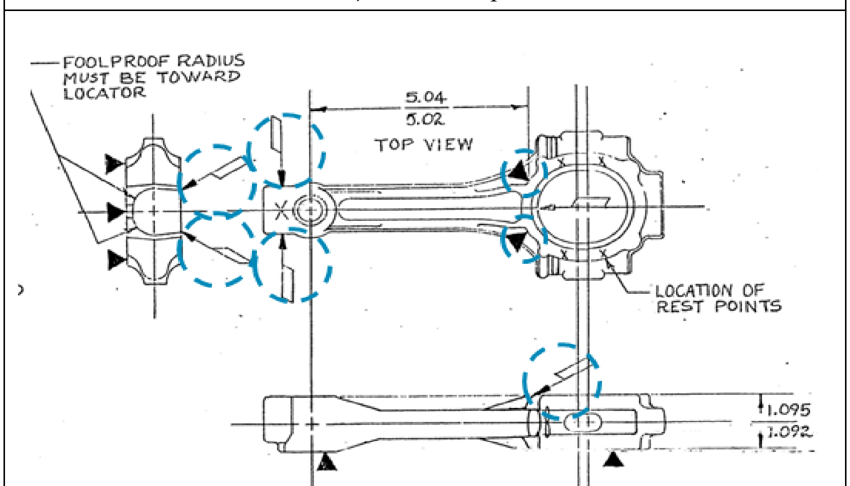


Figure 10: *Illustration of 3-2-1 principle*

Figure 11shows the first two setups for machining a connecting rod from a forged work piece. In this example the locators and clamps which fix the part are shown. So by analyzing at the position of the clamps (or the locators), or the feature of the part that are clamped (or used to locate the part), we can determine the equivalent DRFs corresponding to the setup used for manufacturing features machined in this configuration. The first datum can simply be the face that the connecting rod is sitting on, as shown in the side views of both setups. The secondary and tertiary datums have to be extracted from the locaters and clamps shown in the top view of each setup separately.



a) First setup



b) Second Setup

Figure 11: *First two stages of machining a connecting rod*

For the first setup, the two locators on the side planes of the crank hole can be translated as the mid plane between those to be the second datum. And the other two locators pointing at the cylindrical face of the hole for the gudgeon pin can be interpreted as the tertiary datum to be the axis of the smaller hole. One locator at the end is also constraining the part from moving in longitudinal direction so another datum can be the top plate of the rod.

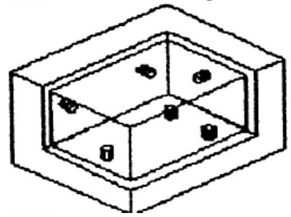
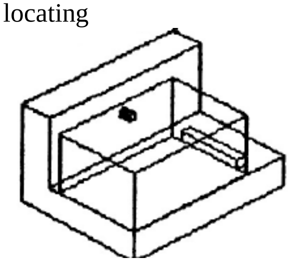
With the same logic, for the second setup, a look at the clamps at the end of the rod and the locators below the crank hole help us determine the second and third datums: the second datum being the mid plane between two end faces at the smaller hole end; the tertiary datum is the cylindrical face of the crank hole.

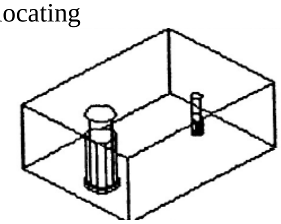
In order to develop a software to extract implied datums and DRFs, the procedure provided above should be organized in the form of a flow chart or pseudo code. This requires classification of fixturing methods and rules for datum extraction associated with each. Although many different fixture configurations are used in industry, the most common methods of fixturing are limited. The scope of this thesis does not include custom jigs and fixtures. Thus, in order to make the task of datum extraction easier, we have created a library of fixturing types. This library (shown in Table) illustrates the main methods of fixturing along with different information that is related with the datums geometry that are essential for later use in analysis.

Table 2: *Library of Fixturing configurations*

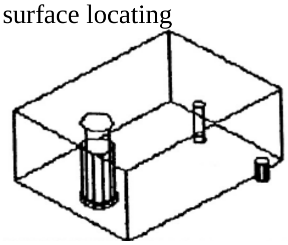
<table>
<tr><th>Category</th><th>Sub-Category</th><th>Method of Fixturing</th><th>Corrs. Datums</th></tr>
<tr>
<td rowspan="5">Locating

model</td>
<td rowspan="2">Plane Locating

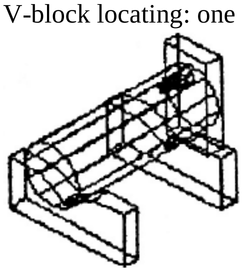
<table><tr><th>Type of datum</th><th>limiting DOF</th></tr><tr><td>First plane, three points</td><td>3</td></tr><tr><td>Second plane, two points</td><td>2</td></tr><tr><td>Third plane, one point</td><td>1</td></tr></table></td>
<td>3-2-1 Point locating
</td>
<td rowspan="2">Three planar faces</td>
</tr>
<tr>
<td>Plane , Small surface, and Pin locating
</td>
</tr>
<tr>
<td rowspan="2">Pin-Hole locating

<table><tr><th>Type of datum</th><th>limiting DOF</th></tr><tr><td>Long cyl. pin</td><td>4</td></tr><tr><td>Short cyl. pin</td><td>2</td></tr><tr><td>diamond pin</td><td>1</td></tr></table></td>
<td>Round & Diamond pin locating
</td>
<td>One cylindrical and

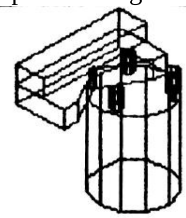
one prismatic hole

with tow mid planes</td>
</tr>
<tr>
<td>Short shaft , Pin, and small surface locating
</td>
<td>Two cylindrical hole

and one planar face</td>
</tr>
<tr>
<td>External profile locating

<table><tr><th>Type of datum</th><th>limiting DOF</th></tr><tr><td>long V block</td><td>4</td></tr><tr><td>short v block</td><td>2</td></tr><tr><td>locating pad</td><td>1</td></tr></table></td>
<td>V-block locating: one
</td>
<td>One cylindrical face,

one planar face, one</td>
</tr>
</table>

		V-pad locating 	Two planar faces
Clamping	Simple Clamp	Vertical model	Mid-Plane (vertical)
		Horizontal model	Mid plane (Horizontal)

For instance, as shown in Table 2, the 3-2-1 locating method can be translated as the most common DRF systems (6 point DRF, fully constrained), with the surface of three points as the primary datum and the planes with 2 and 1 points as the second and tertiary datums respectively. Also, we should save the association of the imaginary planes considered for DRFs extracted with the faces of the part geometry in each stage. In other words, we have to know which faces in the nominal geometry of the part in each machining stage are considered as the datums. As discussed earlier, all the faces in the whole manufacturing process, fixed or exposed ones, have unique ID numbers that will be used later to track the error propagation in subsequent stages of machining.

In order to illustrate the application of this procedure, we will discuss a case in which a part can be located and fixed in two alternative ways, involving different fixturing faces and features (Figure 12). The two options are shown in Figure 12 a, c. In order to drill the holes in the part which has a slot in the middle-top, we can fix the part to constrain it in x direction either with its outer right plane as shown in (a) or by using the slot as in (c). Therefore, for DRFs we derive for each case, the third datum is different, as shown in (b) and (d), respectively.

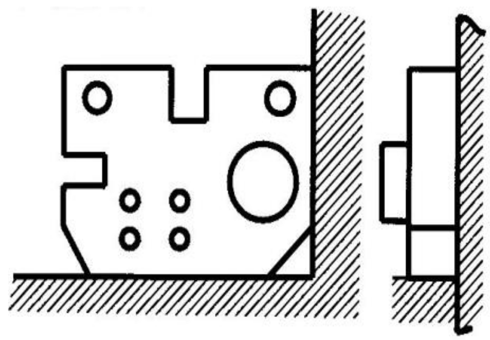
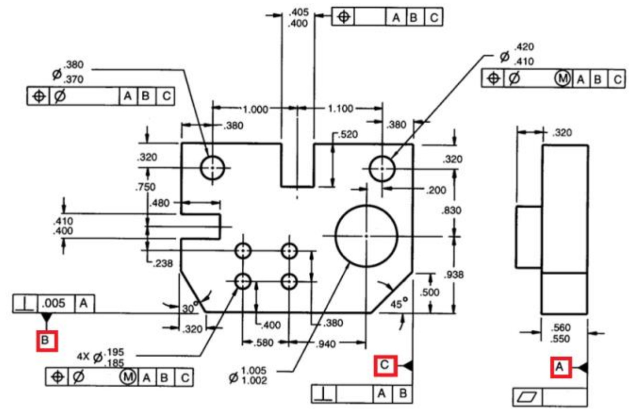
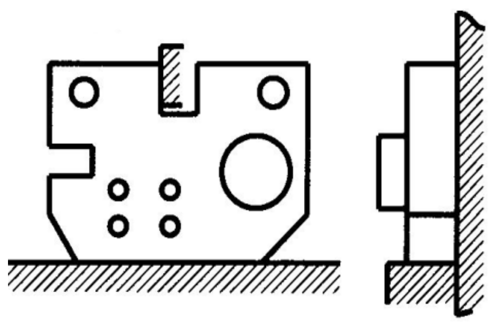
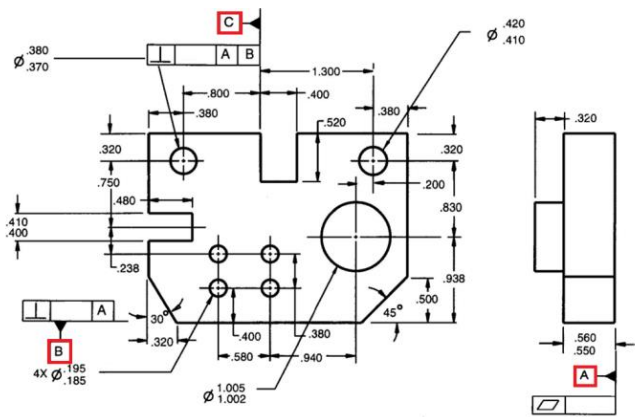
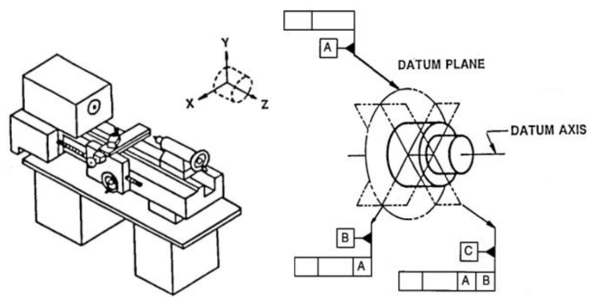
	
a) First fixturing method	b) DRF extracted for the first fixturing method
	
c) Second fixturing method	d) DRF extracted for the second fixturing method

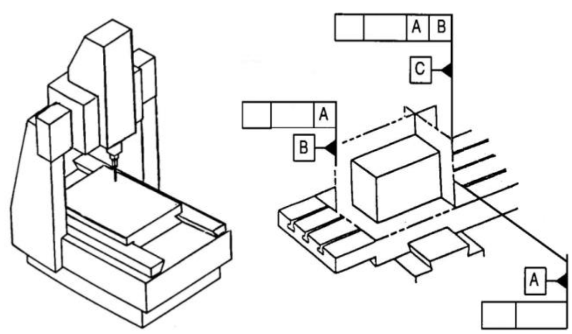
Figure 12: *Different Fixturing methods for machining the same part*

In order to make clear the consequence of this datum difference, we have added dimensions and position tolerances to all of the holes and slots. In these Figures the extracted sizes of all Features of Size (or a pattern of them) are also shown. Thus it is essential to know the datum precedence and the face used as a datum since the errors in creating that face (which is now used as the datum) will affect the features created later.

Now in the case that in a machining process, we use machine tables and surface plates to fix the part, then the planes and axis are derived from the machine coordinate system; this establishes the simulated datums from which the measurements need to be done and dimensions to be verified. For the most common machines used in industry (turning, milling, drilling, etc.) we can derive the machine coordinate system. Thus based on the machine used in each step, which is noted in the process plan, the corresponding DRF can be derived. For instance, it is shown in Figure 13a that if a part is fixed for a turning process, a datum axis and datum plane can be derived.



a) Turning machine



b) Milling machine

Figure 13: *Different machines and the DRF derived with respect to their coordinate systems*

And for a milling machine shown in Figure 13b the bottom plate can be considered as a datum plane, and the other datums need to be derived from the way the part is fixtured. It is noteworthy that the accuracy of the machine surface accuracy with respect to nominal

Machine Coordinate system will affect the accuracy of the part feature created in that setup. The foregoing provides a systematic method for automating DRF extraction.

4.2 Dimensions and dimensional errors extraction

In this step it will be shown how to extract the dimensions and dimensional errors in machining. This task is facilitated by a software module developed previously for extracting so-called "Directions of Dimensional Control" [15]. We can think of a part's geometry as consisting of a number of planes, mid-planes and cylinder axes. We can think of planes as belonging to the same direction of control if their normals point in the same direction, i.e. the planes are parallel. Parallel axes also line up in particular directions of control. Features in the same direction of control chain are displaced from each other by linear distance. This module can find all the direction of a part where we have faces and feature, and also the distance of each face in each direction from the first node is output. The distance between the features can be calculated by subtracting their absolute distance (distance from the first node). Some of these distances correspond to size dimension and some to position. Essentially, these directional chains represent linear stacks for tolerance accumulation.

We need also tolerances values with each of these dimensions and positions in the directional chains. These will be dependent on machine and operation precision that produce the finished features. In order for this happen, the process plan must be read to its lowest level with all the notes and details. For example, if the last cut producing the final size has a relatively deep depth of cut and a relatively high feed rate it will yield the least precise finished product; if the last cut has a relatively shallow depth of cut and a significantly slower feed rate than the preceding cuts a more smooth surface and less errors in product will result. Also, if the last cut with the first tool leaves a relatively shallow depth of cut which is finish cut with a different tool at a relatively slow feed rate the best precision will be achieved. This is because taking a relatively shallow finish cut at a slower feed rate produces less tool pressure which yields more uniform results over the life of the tool by minimizing the variability due to increased tool pressure as the tool wears. As well, using a separate cutter to finish will result in longer accurate production as it will not be worn by the heavy roughing cuts. Beyond this if the same surface is subsequently ground and/or lapped increasingly close tolerance control can be achieved. Therefore, finishing operation machining analysis would be necessary to determine the expected resulting tolerance control.

Therefore, to get tolerance values from a process plan we have to look at the information provided about the tools and machines that are used in each stage of producing the part. To have an automated system we have classified the sources of error as follows:

I. Locating/positioning errors sources:

- Fixture errors
- Datum errors
- Raw material errors

II. Machining errors sources:

- Machine tool errors
- Cutting tool errors

We might not be able to consider all detail of the process plan with this classification as a general method, but these notes can get implemented as rules to the systems. For instance, roughing operations do not affect machining tolerances in most of the cases, but they affect position tolerances as in the case of twist drilling. Finishing operations affect the size and form tolerances, e.g. flatness. Such information is not directly available in process plans. We need to supplement the process plan with data about the machines and tools.

As an example let's look again at the information available in the machining process plan for the casted connecting rod we discussed earlier. As shown in Figure 14, some dimensional values are specified with +/- range. These values can be translated to a position and size error. The position tolerance is a geometrical tolerance that we will discuss in the next section, but the size which is a dimension also will have some errors that we need to extract first. In Figure 15, we have underlined the information about fixtures, clamps, machines, tools, feeds, and speeds are underlined from which geometrical errors and their values can get extracted with some heuristics. These heuristics will be used to identify the fixture type and fixturing method used. For instance, usually when a part is fixed with a clamp it will result in more errors than a part located with a pin with tight clearance. And then the accuracy of the pin location will affect the dimensional size tolerance of the exposing face.

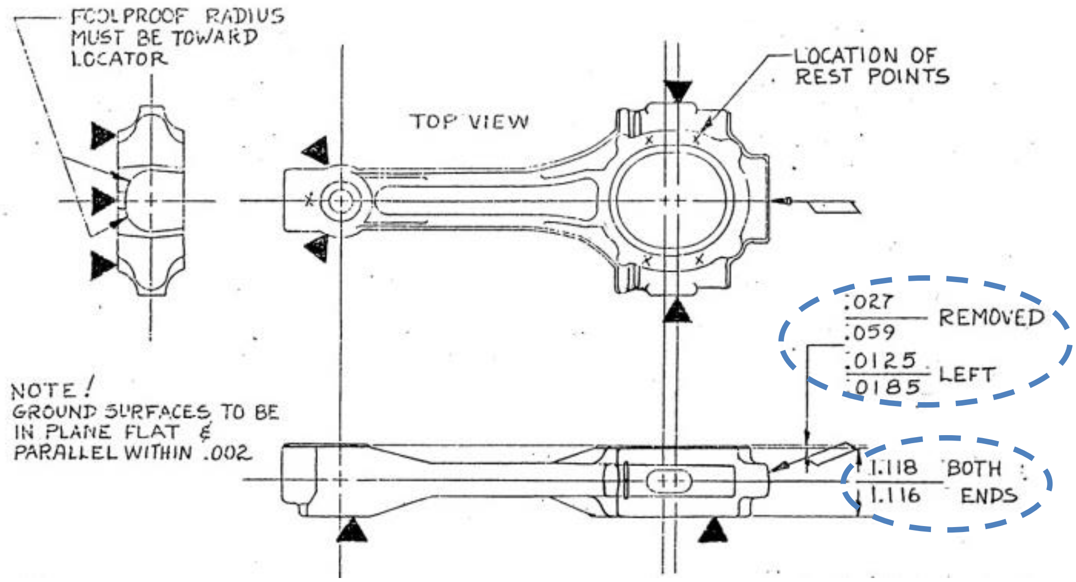


Figure 14: *Annotated figures accompanying the first stage of connecting rod process plan*

OPERATION	DEPT. NO.	MACHINE NAME AND NO.	FIXTURES AND DIES	TOOLS
Rough grind both sides to	811	Blanchard #18-A2	(24) Station Work	Tool Layout
1.095/1.092 Dim.		Grinder #11704-8	Holding Fixture	3172338 KL-1
			3172338-KF-1	Sheet #1
MACHINE DATA		Machine Layout		
		3172338-KM-1	Clamp Lever	Spindle #1 Grinding
Spindle Speed 1200 RPM			3172338 KF-1	Wheel 18" x 5" x 12"
Wheel Speed 5650 SFPM		Floor Plan	Detail #9	89A46 E11 B31GRDCS
Table Speed (1) Rev. = .93 Min.		3172338-KM-1		C.N. #02-200-5934
Load & unload during machine		Sheet #1	(24) Clamp Jaws	
cycle.			3172341 KH-36	Spindle #2 Grinding
12 Stations for 1st Side		Wiring Diagram	C.N. #92-275-3600	Wheel 18" x 5" x 12"
12 Stations for 2nd Side		3172338 KM-1		A89A60 G +11 BEXDCS
		Sheet #2	(24) Locating Block	C.N. #02-200-5937
M.C.T./Piece = .080 Min.			3172341 KH-37	
Gross - 750 pcs./hr.		Worm Gear D-48437	C.N. #92-275-3700	
		C.N. #55-383-7837		
NOTE: Manually dress both			(24) Locating Block	
grinding wheels		Pinion Gear D-48436	Pin End	
every shift		C.N. #55-383-7836	3172341 KH-39	
Reset wheel height			C.N. #92-275-3900	
		Drive Gear D-48432		
		C.N. #55-383-7832		
		Adaptor Plate 18"		
		Gardner Machine Co		
		#D22743		
		C.N. #55-387-6250		

Figure 15: *Process data for extracting errors in machining the connecting rod*

4. 3 Geometrical errors extraction

To be able to create a complete GD&T scheme we also need other geometric tolerances, not just size and position. Without that, a full tolerance analysis cannot be done. However, process plans have no geometrical tolerances (form, orientation, profile, runout) explicitly specified. We assume that the process planner did consider them in choosing the machines, operations, tooling, fixturing. So here we consider if we can "reverse engineer" his thought process to derive geometric tolerances implicit in his plan.

Since we can recognize the effect of different errors in machines and tools on target features, we need to put them in a formal format to be able to use them in our system. For example while machine precision affects the size and the position of features, tool type and speed of cut affect the form of features. We can code the factors on a selected scale and populate the values for each machine in a database. Table 3 summarizes the type of tolerances affected by the combination of the machine type, tool type and tool direction in a tabular format. This information can be stored in a database for different combinations to be used to extract the geometrical errors. In Table , "FN" stands for Face Normal vector, "TA" is the Tool Axis, and "CA" is the Cylindrical face Axis. Also, "FN ⊥ TA" means face normal vector is perpendicular to the tool axis; and "FN || TA" means face normal vector is parallel to the tool axis.

The terms "fixed" and "exposed" has been used throughout this document but the technical definition of them are provided here to better understand the Table 3.

Exposed features are the features or faces that are machined in a setup; we use this term here because they are exposed on the work-piece after material removal in the machining process; tolerances are applied to exposed faces. Several factors influence the machining tolerances on exposed features.

Fixed faces are the faces that are not machined in a setup. Since the faces that we can use to fix the part are among fixed faces, the DRF that we extract will be the nominal geometry of the chosen fixed faces. For instance, Orientation tolerance on exposed planar face should be given with respect to the fixed faces that are perpendicular to it. Similarly position tolerance on exposed planar face should be given with respect to the fixed face that is parallel to it.

The requirement of the number of fixed faces to associate errors extracted with, depends on type of exposed face. Below is description for requirement of types of tolerance on three types of target features and their faces.

Planar face: While machining a planar face, three fixture faces are required to control orientation and position of the exposed face. Let’s name the three fixed faces as primary, secondary and tertiary fixed face. The three fixed faces should be perpendicular to each other; and it is practical to assume that the face lying on the table to consider as primary fixed face. Three types of tolerance are relevant for exposed planar face, Form (flatness), Orientation (Parallelism or Perpendicularity), and Position. The position tolerance is ± tolerance on distance between exposed face and fixed face parallel to it.

There are two conditions to apply tolerance on the exposed planar face.

When the face is parallel to primary fixed face: In this condition the position and orientation (parallelism) tolerance can be applied with respect to primary fixed face. Other two fixed faces do not contribute to any variations.

When the face is not parallel to primary fixed face: In this condition the orientation (perpendicularity) tolerance with respect to two fixed faces that are perpendicular to the exposed face and position tolerance applied with respect to a fixed face parallel to the exposed face.

Note that form tolerance does not require a datum, i.e. no fixed face needed. Hence it can be applied the same way regardless of exposed face and fixed face relation.

Cylindrical face (Hole/Pin): Machining cylindrical faces also requires three fixed faces. Two fixed faces that are parallel to axis of cylindrical surface control position; and the fixed face perpendicular to cylindrical axis controls orientation of the axis. As per definition of position tolerance in ASME standard perpendicularity tolerance is included in the position tolerance itself. Hence we can control the perpendicularity of the hole by just applying position tolerance on it with respect to all three fixed faces. Also, form tolerance (straightness) and size tolerance are also applicable to cylindrical faces regardless of fixed face.

Slot: Slot is group of planar faces of which two faces are antiparallel. Size tolerance is applied between these two faces and position tolerance is applied on the mid-plan (formed by the two faces) with respect to all three faces (similar to position tolerance applied to hole).

Table 3: Geometric errors on exposed features in milling operations

Machine Type	**Tool**	**Face type exposed after the process[4]**	**Face-Tool relation[1] TA = Tool axis, FN = face normal vector CA = Cyl. face axis**	**Fixed faces required[2] P = Primary, S = Secondary, T = Tertiary**	**Required Tolerances on the face[3]**
VMC	End Mill / Face Mill	Planar face	FN ‖ TA	P	- Form, - Orientation w.r.t. P, - Position w.r.t. P,
		Planar face	FN ⊥ TA	P, S, T	- Form, - Orientation w.r.t. P and S or T, - Position w.r.t. S or T
	End Mill	Cylindrical face (Hole / Pin)	CA ‖ TA	P, S, T	- Form, - Size, - Position w.r.t. P,S,T
	Drill	Cylindrical face (Hole)	CA ‖ TA	P, S, T	- Form, - Size, - Position w.r.t. P,S,T
HMC	Plain Mill	Planar face	FN ‖ TA	P	- Form, - Orientation w.r.t. P, - Position w.r.t. P,
	Side Mill	Planar face	FN ⊥ TA	P, S, T	- Form, - Orientation w.r.t. P and S or T, - Position w.r.t. S or T

Some of the variations on exposed faces occur due to these fixed faces. So basically if a fixed face that is previously machined in a different setup, is used to fix the part for a new setup, the errors in the fixed faces initiated in the previous machining process will be transferred to the exposing faces of the new setup. Note that for some of the exposed faces that are removed in subsequent processes, but used as fixed faces during a machining process, it is necessary to sum up their resultant errors. Some shapes or features are more complex to machine and some are not accessible directly by the tool when there is a lack of appropriate fixture. Therefore, a process planner utilizes several methods to create those features with the existing capability of the machine shop. One of those techniques involves creating features just to serve as a fixture or a datum to create other intended features. We call these features, transient features and they are temporary. They will disappear down the line of subsequent operations. Sometimes transient features are created intentionally, and sometimes they are just created in one step then removed in subsequent steps.This can be due to several reasons, for instance a machine may create a certain part in multiple steps or a hole of a large diameter cannot be created in a single operation etc. Creation of transient features whether intentionally or unintentionally is a common phenomenon occurring at the shop floor. Often the importance of these transient features is neglected in dimensional and geometrical accuracy of the final part due to the fact that they are transient in nature and do not appear on the final part but they may have affected the precision if for instance they have been used as a datum down the line in machining processes.

A simple example of intentional transient features is shown in Figure 16. Let’s assume that we want to create a circular part with a rectangular hole in the middle from a raw square plate. A procedure that a process planner may suggest is to make 4 holes in the middle of the plate, which fit in the square hole that we want to cut from the plate, as the fixture. Thus, the steps to make this part can be as shown in the Figure 17. In the first step, we make the four holes that will serve as the fixture faces for the second and third steps. In the second step, we machine the outer cylindrical surface. And finally, in the third step we cut the rectangular hole, containing the four holes, from the plate. There may be better solutions for manufacturing this part; it is greatly dependent on the type of machines available in the shop. But this example is just to highlight the fact that some features can be temporary which we have called transient features. We can see that these four holes are neither in the beginning, nor in the end. We create them in order to serve as the fixture for the next steps; therefore the errors in creating the holes will be accumulated with the errors that occur in the steps where these holes serve as the point of fixturing for. The accumulation of errors in these distinct steps of manufacturing is the key point here.

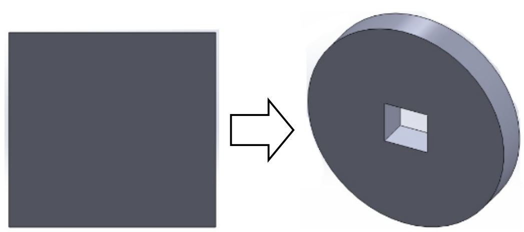
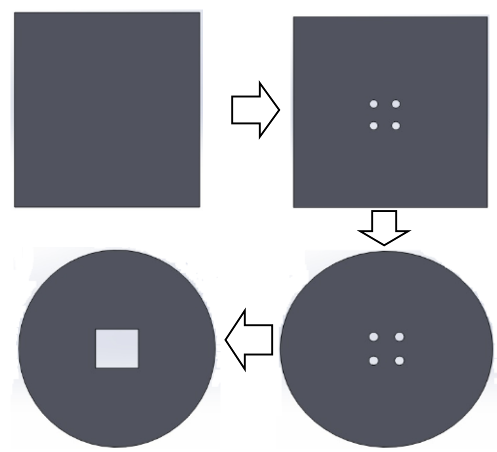
	

Figure 16: *Process plan intent*	***Figure 17:*** *Process plan steps*

Thus, transient features that are created in a manufacturing process play an important role in the tolerance control of the feature for which they are acting as a reference (datum or a fixture). Since these transient features contribute to the tolerance accumulation, their error should be taken into account and here we do it by adding necessary tolerances to the exposed faces in each setup which can be a part a transient feature. This information, including the transient features, needs to be collected in a form that facilitates determining their effect on exposed features.

Table 4: *Sources and factors affecting geometric tolerances*

Variation Type	**Source Of Error**	**Factors**
Form	Machine static error	Tool travel w.r.t origin
	Dynamic error	Length / Diameter ratio (for hole)
Position	Machine static error	Tool travel w.r.t origin (for hole)
	Fixture error	Height of fixture locators
	Dynamic error	Feed rate, Speed, Depth of cut, length of tool (for plane)
Size	Tool (for holes)	Tool wear (Tool material vs work piece material)
Orientation	Fixture error	Difference in height of fixture locators

Back to the problem of geometrical errors, there are many parameters that create variations on faces. It is very difficult to give explicit information about source of error and types of error. Here we tried to predict them based on major sources of error. Note that the variations modeled here may be different than practical machining variations. This model may be used temporarily to produce some sensible tolerance values. One can use more sophisticated variation model to obtain tolerance values. The variation type and associated source of error is described in table 4. So with having the Geometric tolerances and their connection with the geometry, we can establish our GD&T schema based on errors that will occur in the process plan.

4.4 Conclusion

In this chapter, we discussed our new approach of deriving the information needed for verification of machining process plan from GD&T point of view. We showed that this information are in the process plan but they are not explicitly represented. To have Geometrical and Dimensional machining errors in the standard representation of design tolerances, we need to extract three basic information: 1. Datums, 2. Variation types, and 3. Variation tolerance values. Therefore, we first discussed how the information in the process plan such as machine working tables and coordinate systems, method of fixturing and etc. that convey the datum features can be extracted. Then we discussed how to identify the type of errors that happen during milling and drilling process from the machining operation, machine types and the tools used, tool path and etc. And finally we discussed how to estimate the amount of variations for each tolerance class. The discussions in this chapter and libraries developed, with limited scope in this work, will help the m-map testbed user to assign the errors to the geometry of the part in each step for further analysis. It is noteworthy that although the process of GD&T information extraction is assumed to be done interactively by the user in this research, but the groundworks are based in such a way that an automated system can be developed on top of the same methodologies and classifications.

CHAPTER 5: m-map CONSTRUCTION FROM EXPLICIT GD&T

5.1Algorithms and Implementation

In this chapter the method for creating m-maps is discussed. This approach is summarized in the flowchart of Figure 18. As shown, the inputs to the systems are the process plan data files including the geometry of the part in each step (Task 0). The geometry needs to be in ACIS (.sat) file format, which can be output from most major CAD/CAM packages, and other information such as tool listing or operation lists are usually in .txt file format.

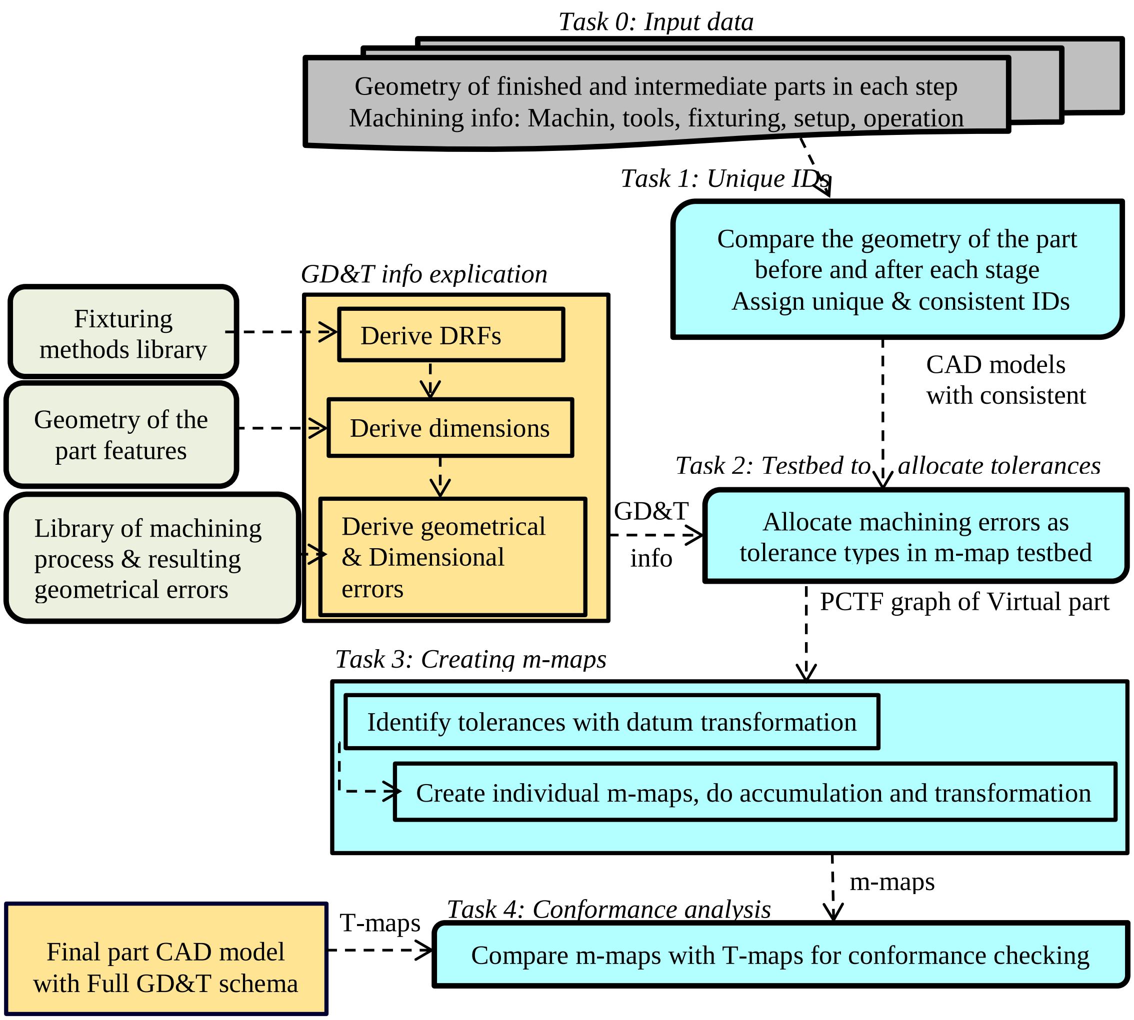


Figure 18: *flowchart of the tasks to creat m-maps*

The system will take the geometry for the part in each step to compare and assign unique IDs to faces that are similar (not changed through machining process) in different machining steps (Task1). Then the user needs to input the dimensional and geometrical machining errors, based on the process plan information, in the form of design GD&T for every machined face (exposed face) in each setup. This means that based on machining information provided in the process plan, the user needs to identify the corresponding error type, its value and its respective Datum from the look up tables. Then the user can allocate the errors interactively, the same way tolerances are allocated, to the geometry using the software. This information will first get stored for every step in CTFs and then they will be compared to create the final PCTF. The PCTF includes the superposition of all the errors that has datum transformation. The information in the PCTF can be interpreted as the tolerance information for a virtual part that has all intermitted faces with assigned errors of all machining steps.

5.2: Algorithm for the consistent IDs in PCTF:

In our system each step corresponds to one machining operation and we need to track the changes of the fixed and exposing faces through all the steps. For this purpose, the geometry of the parts before and after each machining process is considered; the geometry before machining in each step is called “stock part”, and the one after machining is called “machined part”. The method used for identifying machined faces is named “Clashing Faces” method. In this method, the geometry of the machined part is compared with the stock part in each step and in the same coordinate system to find all the faces that are clashing (faces that are coincident, per ACIS kernel API definition). The faces that are clashing are fixed faces and other faces are new machined faces. Although fixed faces may be trimmed and have a different size and topology in the machined part of each step, they have the same reference geometry, thus they need to have the same ID to prevent an unnecessary datum transformation later. Accordingly, all fixed faces of the machined part will get the same ID as the stock part and exposed faces will get new face IDs. Face IDs are unique textual attributes attached to each face of a part. The algorithm for Unique IDs can be summarized in the flow chart shown in Figure 19:

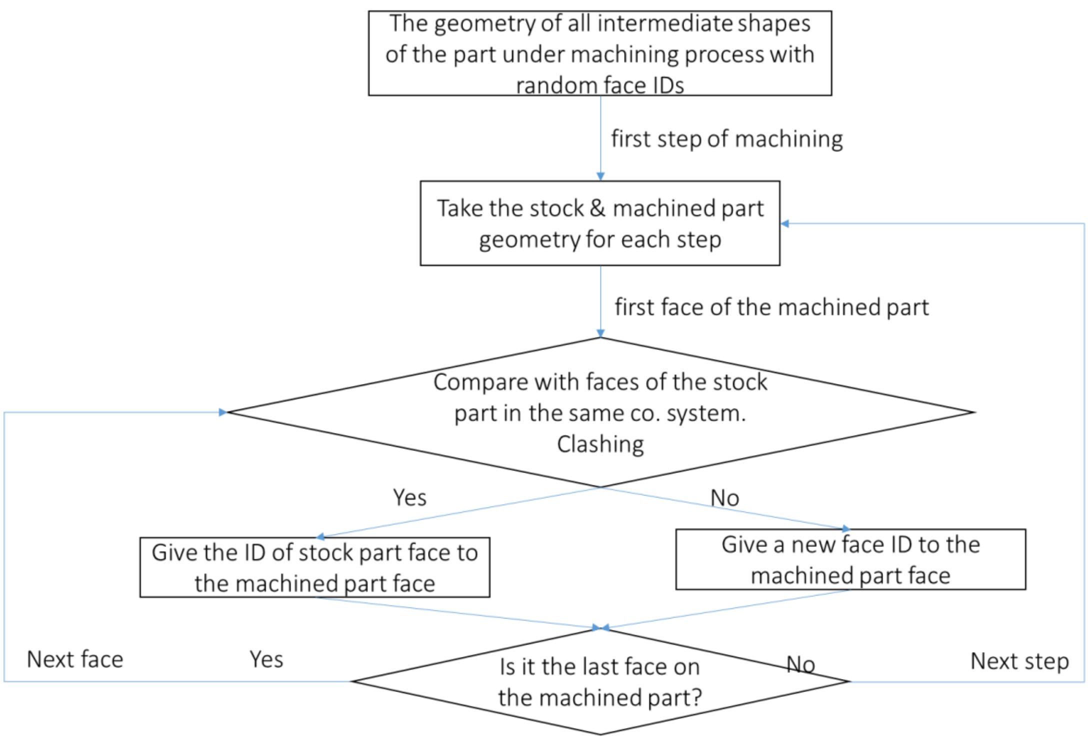


Figure 19: *Unique ID algorithm for the CAD geometry of the part under machining process*

To better illustrate the concept of unique IDs for a part under machining process, I will walk you through an example that starts with drilling a hole in a stock part. The stock part is a cube with 6 faces, thus it has 6 unique IDs attached to each face as shown in Figure 20:

#1 planar face

#2 planar face
#3 planar face
#4 planar face
#5 planar face
#6 planar face

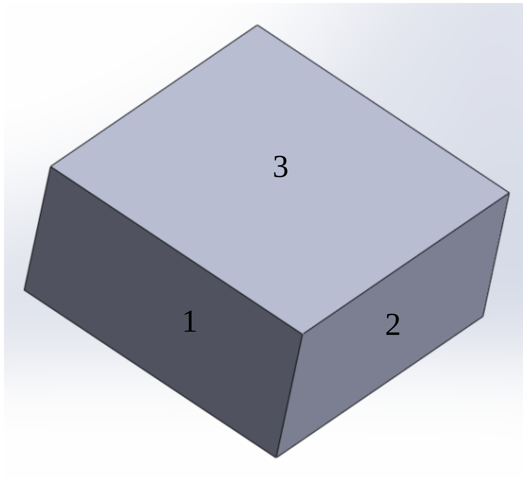


Figure 20: *Stock part (a cube)*

In the first step (Figure 21), when we drill the hole, a new cylindrical face will be created and the two top and bottom faces will also change. We want these faces to keep their IDs and have the same number as the ones assigned to them in the stock part. Thus we will bring this machine part with the stock part and compare them in the same coordinate system to find the clashing faces. The four side faces of this part are not changed, thus they are fixed faces. Since face 3 (Top face) and face 6 (bottom face) will clash with their original face, they will also be recognized as fixed faces (although they are altered and their size, shape and topology is changed) and the unique IDs from the Stock part will be transferred to the machined part in the first step. The only non-clashing face is the new cylindrical face created in drilling operation which will be recognized as an exposed face (machined face). This Cylindrical face will be assigned a new face ID that is added to the face list in Figure 21.

#1 planar face
#2 planar face

#3 planar face
#4 planar face
#5 planar face
#6 planar face
#7 cylindrical face

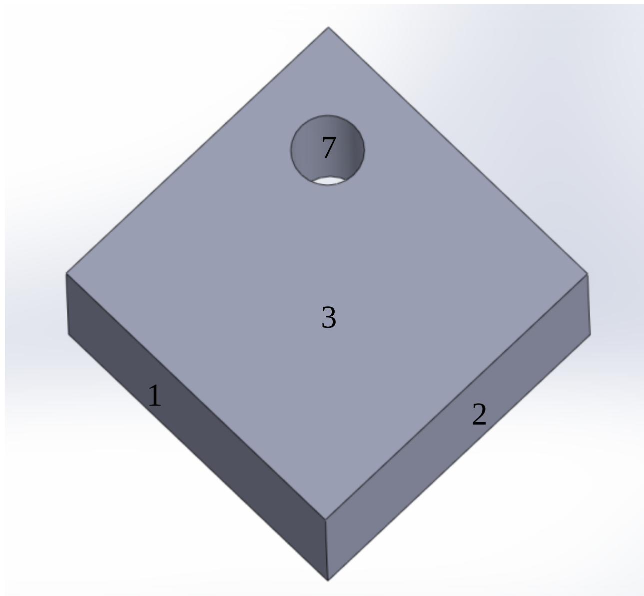


Figure 21: *First Machining Process, drilling a hole*

Two scenarios are considered for machining in the next step. In the first scenario, the part is machined in a milling operation to create a step on two edges of the part (Figure 22). In the second scenario, the part is machined to create an "L" shaped slot as shown in Figure 23. The reason for bringing a second scenario is to point out the details of the unique ID generating algorithm. In the first scenario, by comparing the CAD model of the machined part with the previous step, we find three non-clashing (exposing) faces which will be given new face IDs (faces 8, face 9 & face10). Face8 and face 9 are parallel to face1 and face 2 but they are distant, so they are exposing faces. It is the same for face # 10, which is parallel to face 3 but distant from it.

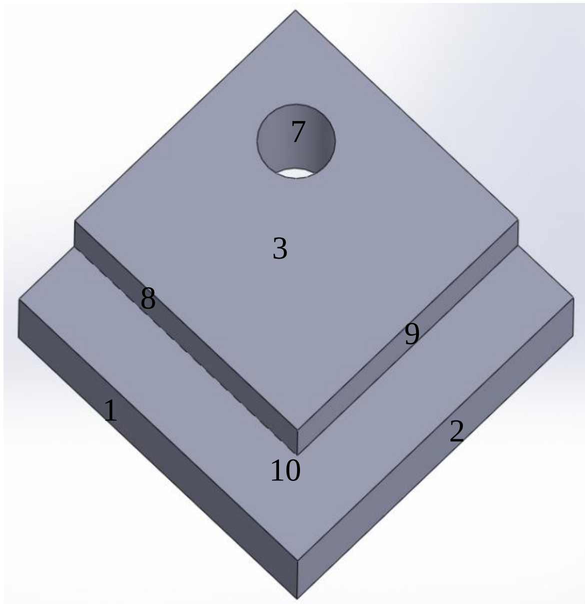


Figure 22: *First scenario for the second machining process, milling a step*

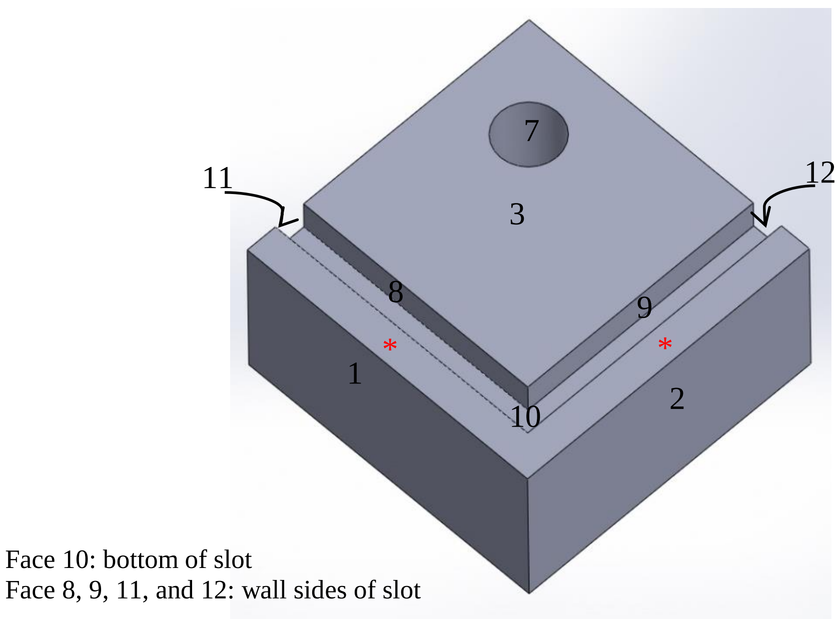


Figure 23: *Second scenario for the second machining process, milling a slot*

In the second scenario, following the same algorithm, 5 machined faces (exposed faces) will be identified. The main point is that in machining the slot, the top plane (face 3) is split into two pieces and when checking the clashing faces, both of them will coincide with the face 3 of the previous steps. Although from computational geometry and constraints point of view the two top faces are different, but they will be assigned the same ID. As shown in Figure 23, there are 4 exposing faces (face8, face9, face10, and face11) as the walls of the slot and one exposing face (face12) as bottom of the slot. Also, the face marked with asterisks in Figure 23 which will be recognized as a new face in geometry of the part, is not an exposing face. Thus, by applying the same algorithm all exposed faces will be identified and the face attributes saved for later use. This unique ID generating algorithm makes sure that all of the faces of the part are uniquely identified throughout the machining steps.

5.3 Deriving and assigning Geometrical and Dimensional errors (Creating PCTF)

When faces of the CAD models for the part in different steps have same face IDs, any information attached to them can be tracked throughout the machining steps. At this point, the user can interactively assign the error information derived from the process plan, based on the libraries that were discussed in chapter 4. The system is capable of storing the Geometrical and Dimensional errors information along with the geometry in CTF as discussed in chapter 3. Initially, the info for each step is stored in separate CTFs, and then they will be combined and summarized in one PCTF file for creating m-maps. The GD&T information of each step is stored in a CTF which will be compared to the CTF of other steps. The system will look for faces (of any feature of the part) that has been machined in one step and later used as the datum (for fixtruing the part in next steps) —these are the cases where datum transformation occurs for which the m-map needs to be generated. Thus the information of those faces, the errors types and value for each step they are involved, and the order of the error accumulation will be stored in the PCTF.

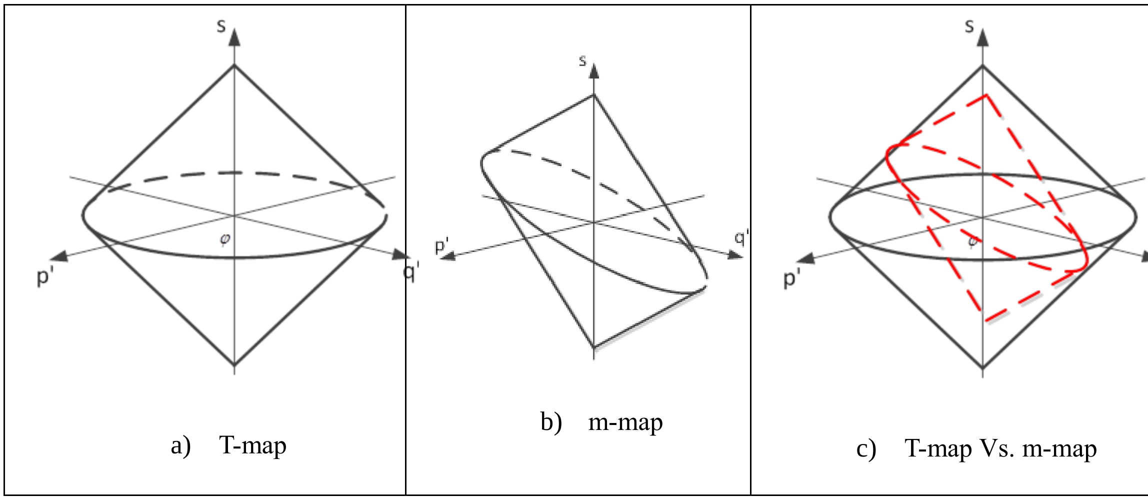
Basically, the PCTF represents a virtual part that include all intermediate faces of the part under machining processes, with all the errors superposed. This virtual part does not really exist, and it cannot be modeled in conventional CAD software because the constraint solver of CAD/CAM packages cannot solve the constraints. However, this virtual part can be analyzed with conventional tolerance analysis methods such as Tolerance charts, Monte Carlo simulation, or T-maps. In this thesis, we will use the T-map, or “Tolerance Map”, method developed by the Design Automation Lab (DAL) at Arizona State University (ASU). When the T-map method is used to model the accumulation of machining errors through multiple steps of manufacturing based on a process plan, it is called m-maps where “T” for tolerances is substituted with “m” for manufacturing.

5.4 Creating accumulation maps (m-maps)

In this step we have the CAD models of all intermediate shapes, with unique IDs for the faces. Also the GD&T errors are populated in the testbed and stored in CTF graphs. As we know, CTF is structured in a way that stores the errors with respect to a datum which is marked with its unique ID. Thus by parsing through the CTFs, the tolerances/errors which have datum transformation, if any, will be selected to create the m-map. In order to do this, we start with the datums of the final step, and trace them with their ID though previous CTFs, if they have machining tolerance error with respect to another datum in an earlier step, they have datum transformation, and their info needs to be stored. All the geometrical errors with datum transformation are superimposed into one “virtual” part and their information are stored in a PCTF. Finally, based on spatial and geometric parameters along with accumulated errors, m-maps are created and transformed to be checked versus T-maps. Detail of T-maps can be found in appendix B. How m-maps are compared with T-maps is discussed in the next section.

5.5 Conformance and checking

In the final phase, when all the accumulated m-maps are created, the m-maps can be compared with the T-maps. In order to do this, first we need to make sure that the two maps are in the same coordinate system. This does not happen unless, the final machining datums of the accumulated error (in m-map) coincide with datums specified in the design drawing (in T-map); thus in most cases we need to do transformation. Ke Jiang [22] has developed the mathematical model to describe transfer of cylindrical and planar datum comprehensively. His model can also obtain machining tolerances accurately when taking material condition on datums. Now, basically a worst case analysis checks if the m-map fit in the T-map. If it does, we can realize that in each direction, corresponding to one type of tolerance/error, the process plan can meet the design specification. If not, we may need to generate alternative plans to reduce machine errors exceeding the allowed tolerances in design. To illustrate this, we have shown a simple 3-D T-map, and a sample transformed accumulated m-map in Figure 24. We can see in Figure 24-c that the m-map in red dashed lines fit in the T-map when they are shown in the same coordinate system. To analyze each type of error we have to check each direction, and for this reason different cross sections of the maps can be compared. This shows us the directions in which the m-map has exceeded its tolerance limits represented by the T-map (Figure 25).

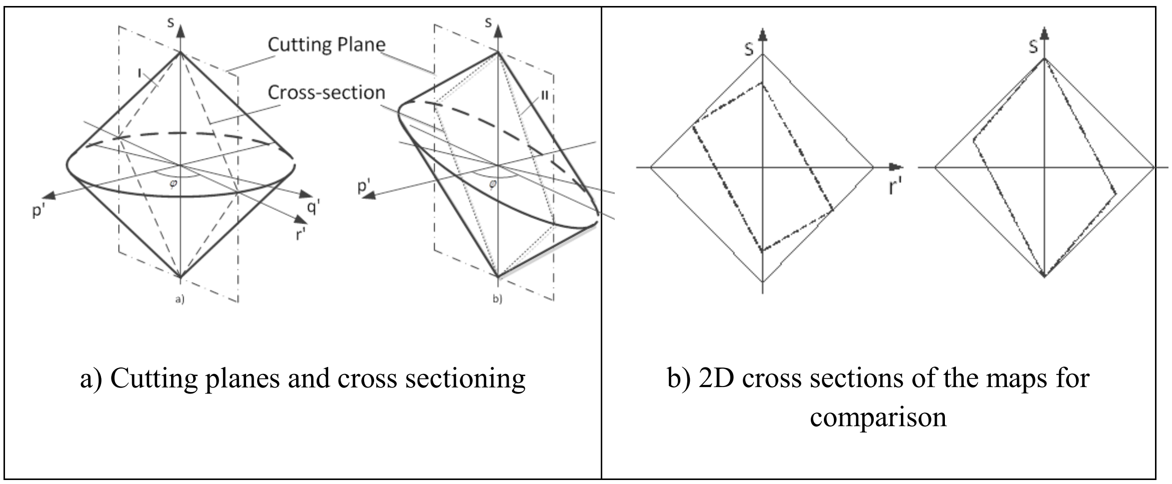


a) T-map

b) m-map

c) T-map Vs. m-map

Figure 24: *Example of a T-map Vs. m-map [23]*



a) Cutting planes and cross sectioning

b) 2D cross sections of the maps for comparison

Figure 25: *Cross-sections of the T-map and M-maps for comaparison*

Note that m-map and T-map are both n-dimensional point spaces, thus their volume-area are comparable. It is also possible to see what stage of the manufacturing process has the most contribution in the resulting error. This can happen by tracking the m-map accumulation contributors in the PCTF. Thus the ideal system can optimize the process plan versus design drawing e.g. for minimum manufacturing cost in an iterative process.

All of the steps, shown in Figure 18, which were discussed in this chapter, will be illustrated with an example in the next chapter

CHAPTER 6: CASE STUDY AND VERIFICATION

In this chapter, I will go through each step of the algorithm with the help of an example. The example part is a simplified model of an eccentric shaft. The Design drawing and the CAD model of this part is shown in Figure 26.

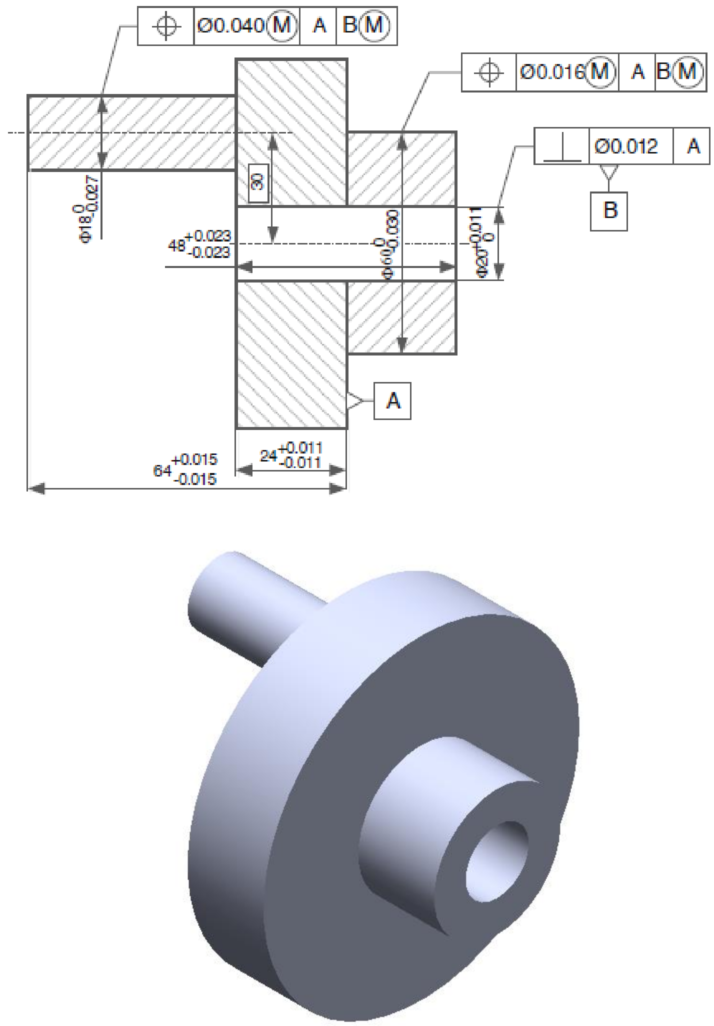


Figure 26: *Design Drawing and CAD model of a simple eccentric shaft*

The design drawing of the part specifies allowable tolerance limits, and the datums. The manufacturing scheme might not follow the same datums for fixturing in all the steps, which is decided by the process planner and manufacturer at the machine shop floor. On possible way to machine this part from stock is shown in Figure 27. This figure shows the geometry of the part in each step, and the blue faces are machined faces of each one. Also the the faces that are sued to fic the part are marked with locating signs. The stock is first fixed to mill the cylindrical boss on top of the part, and then the middle hole is drilled to the needed depth. Then, the part is turned to be fixed with the boss and the hole, to machine the outer cylindrical surface and then the outcentric boss (shaft) on the other side. In the final two steps, the part can be machined either with a turning or a milling machine, which depend on the availability and precision of the machines on the shop floor. Since in the facility the experiment is conducted, the milling machine has more precision, milling is considered for the last two machining operations.

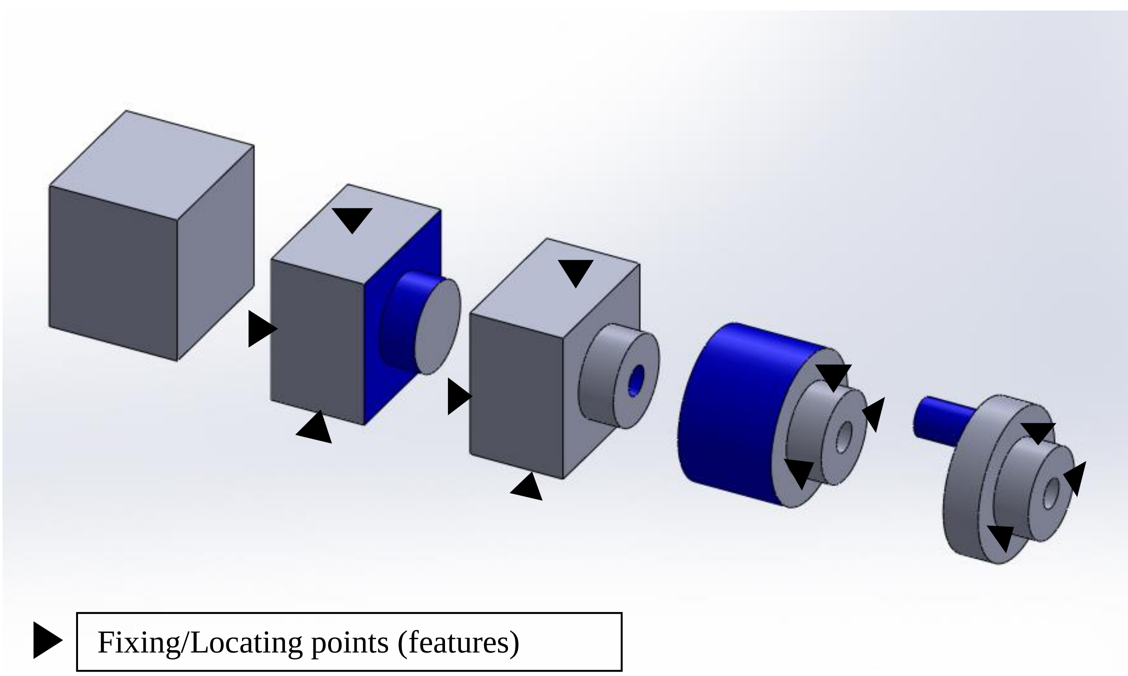


Figure 27: *Manufacturing steps of machining the eccentric shaft*

Details of manufacturing, such as the machine and tools to be used in each step, are available in the process plan table (Table 5). This table presents the machining information of each step, e.g. in the first step (step10), milling machine M1 with tool F662 is used to machine and remove material to create the boss on the top of the part from the block stock part. We have created the process plan for this example part in NX CAM software. The same can be understood from the next three steps. Figures 28 and 29 are also output reports of the NX CAM package being the "tool list" and the "operation list" respectively. These lists provide more detailed information about the machines and tools used. Also the number of passes the tool path and other information are available in the process plan output data files which are not provided here. One last piece of information that is usually provided to the manufacturing process is informal sketches along with notes on design requirements.

Table 5: *Machining process plan of the cap of the cap-cylinder assembly*

Part No: 001		Part name: Cylinder Cap	planner: P. Haghighi	checked by: P. Mohan, N. Kalish		Date: 8/28/14	Page: 1/1
Material: Acrylic glass (PMMA)		Stock size: 80x80x50	Comments:				
Setup No.	Operation Description:		Dept	Machine	Tooling	setup	Std.
10	Mill cyl.l surface: 60mm Dia. x 24mm Depth		Mill	M1	F662	1 hr.	5 min
20	Drill hole: 30mm Dia. x 50mm Depth		Drill	D1	J555	0.5 hr.	3 min
30	Mill cyl. surface: 100mm Dia. x 70mm Depth		Lathe	M1	G0810	0.7 hr.	5 min
40	Mill cyl. surface: 25mm Dia. x 40mm Depth		Mill	M1	F630	1 hr.	7 min
50							

```
============================================================
Information listing created by :  phaghigh
Date                           :  9/10/2014 2:23:06 PM
Current work part              :
Node name                      :  en4074087
============================================================

                          SHOP FLOOR DOCUMENTATION
                          ************************

-----------------------------------------------------------------------
CREATED BY : phaghigh          DATE : Wed Sep 10 14:23:08 2014
PART  NAME :
-----------------------------------------------------------------------

                     OPERATION LIST BY PROGRAM
                     *************************

-----------------------------------------------------------------------

PROGRAM NAME : PROGRAM
-----------------------------------------------------------------------
OPERATION NAME          OPERATION DESCRIPTION             TOOL NAME
-----------------------------------------------------------------------
PECK_DRILLING           drill/PECK_DRILLING               DRILLING_TOOL

ZLEVEL_PROFILE          mill_contour/ZLEVEL_PROFILE       MILL

FACE_MILL               mill_contour/FACE_MILL            MILL
```

***Figure 28:** Tool list*

```
Information listing created by :  phaghigh
Date                           :  9/10/2014 2:29:12 PM
Current work part              :
prt.prt
Node name                      :  en4074087
============================================================

                          SHOP FLOOR DOCUMENTATION
                          ************************

-----------------------------------------------------------------------
CREATED BY : phaghigh                   DATE : Wed Sep 10 14:29:13 2014
PART  NAME :
-----------------------------------------------------------------------

                                TOOL LIST
                                *********
DRILLING TOOLS
-------------------------------------------------------------------------
TOOL NAME         DESCRIPTION                DIAMETER  TIP ANG   FLUTE LEN  ADJ RE
-------------------------------------------------------------------------
MILL              Milling Tool-5 Parameters  5.0000    0.0000    0.0000     0
-------------------------------------------------------------------------

MILLING TOOLS
-------------------------------------------------------------------------
TOOL NAME        DESCRIPTION                 DIAMETER  COR RAD   FLUTE LEN  ADJ RE
-------------------------------------------------------------------------
MILL             Milling Tool-5 Parameters   20.0000   0.0000    50.0000    0
-------------------------------------------------------------------------

DRILLING TOOLS
-------------------------------------------------------------------------
TOOL NAME        DESCRIPTION                 DIAMETER  TIP ANG   FLUTE LEN  ADJ RE
-------------------------------------------------------------------------
DRILLING_TOOL    Drilling Tool               30.0000   118.0000  35.0000    0
-------------------------------------------------------------------------
```

***Figure 29:** Operation list*

Now we want to go over each step of our flowchart in Figure18 for creating the m-maps. We have all the CAD models and some detail of machining for each step. In the first step, by choosing the “ID” icon in the testbed, all the CAD models can be input to the software, and the system will create “.sat” files with unique IDs automatically. The input parts should follow the naming convention and the .sat files created will have the same name with “-ID” added to their name, in order to not overwrite the original files. Then the user can open the CAD models of each step and based on machining information of the process plan, assign geometrical and dimensional errors in to the part. The user has to first select the type of error he/she want to add to the geometry, then select the face or faces to assign the error to. For example to assign dimensional error, the user has to select the size tolerance (+/-) icon, and then select two entity , one at a time, between which he wants to add size tolerance.

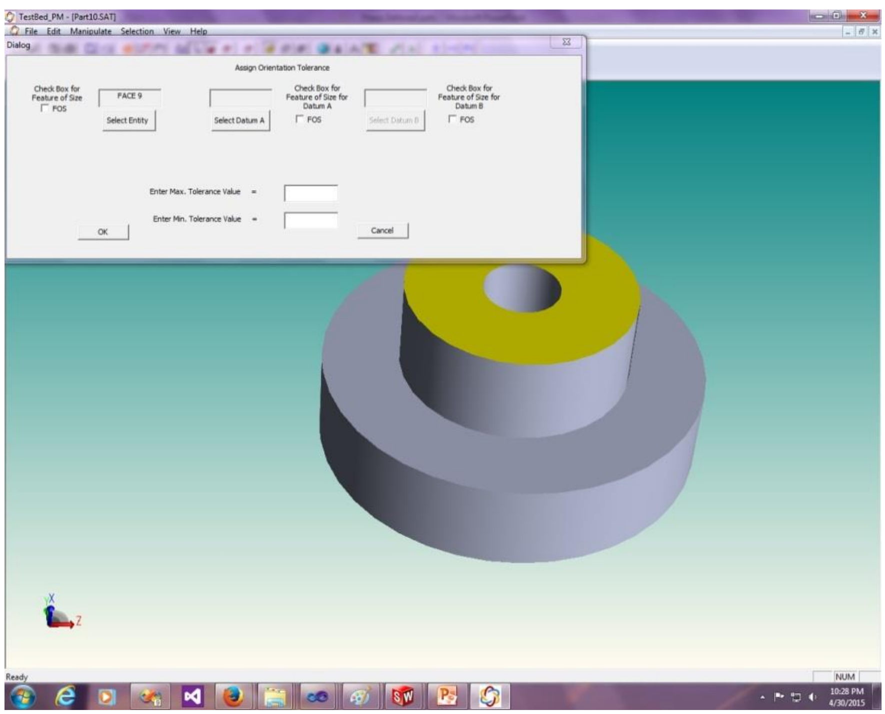
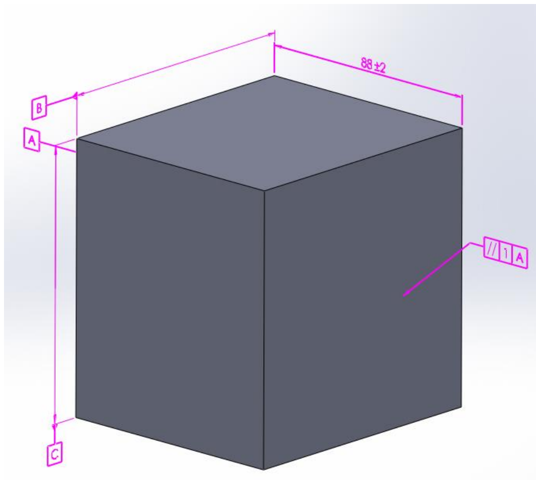


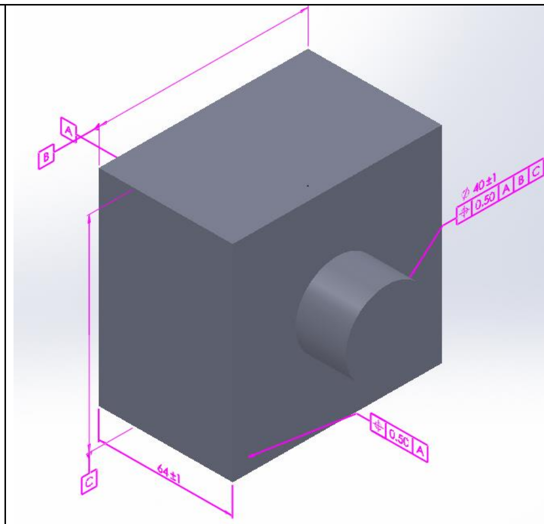
Figure 30: *Screen shot of the Testbed while assigning orientation error to the top surface*

For geometrical errors, based on the machining operation for that setup, the user can select the type of errors needed by looking at Table 3. Then he can select the corresponding icon from the menu, and then select the face (or faces for features of size) that needs to be toleranced. Also for geometrical tolerance, based on the type, from1 to 3 datums may be asked by the testbed. Figure 30 is a screenshot of the testbed, while assigning orientation error for the top surface with respect to the bottom face in the third step of machining. The entity to tolerance is face No. 6 which is highlighted and since it's a planar face, the system only asks for one datum.

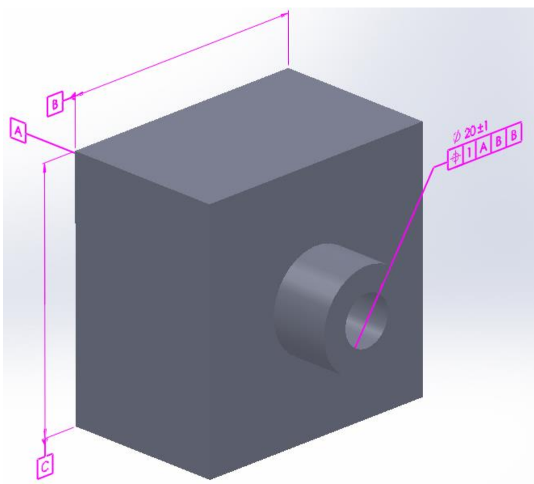
For each step, when machining errors are assigned to the geometry, the information will be stored in a separate CTF. The information in each CTF can be red back into the software to visualize or edit, if needed. In Figure 31, the Geometrical and Dimensional errors of the five machining steps in the example are represented in the same way as design tolerances in five 3D-CAD models. In steps 1 to 3 of machining, shown in figures A through c of Figure 31, the bottom plate is datum A as it is the face it is sitting on, and the side perpendicular faces are datums B and C (for each one two opposite faces are chosen as feature of size) since they are used to fix the part in this setup. Then the setup is changed for the 4^{th} and 5^{th} step and the cylindrical boss and the middle hole machined in earlier steps are used to fix the part to create the outer cylindrical face and the out-centric shaft.



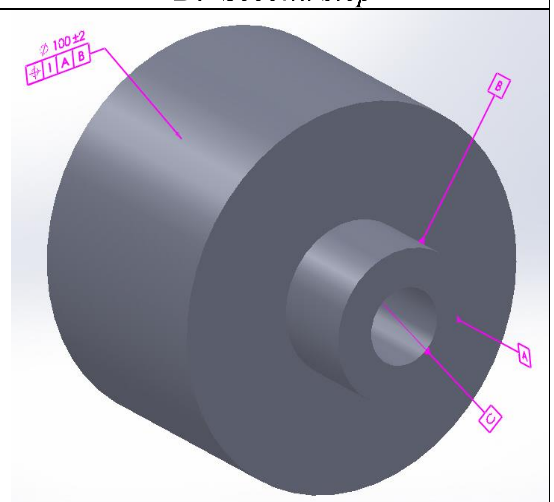
A: *Frist step*



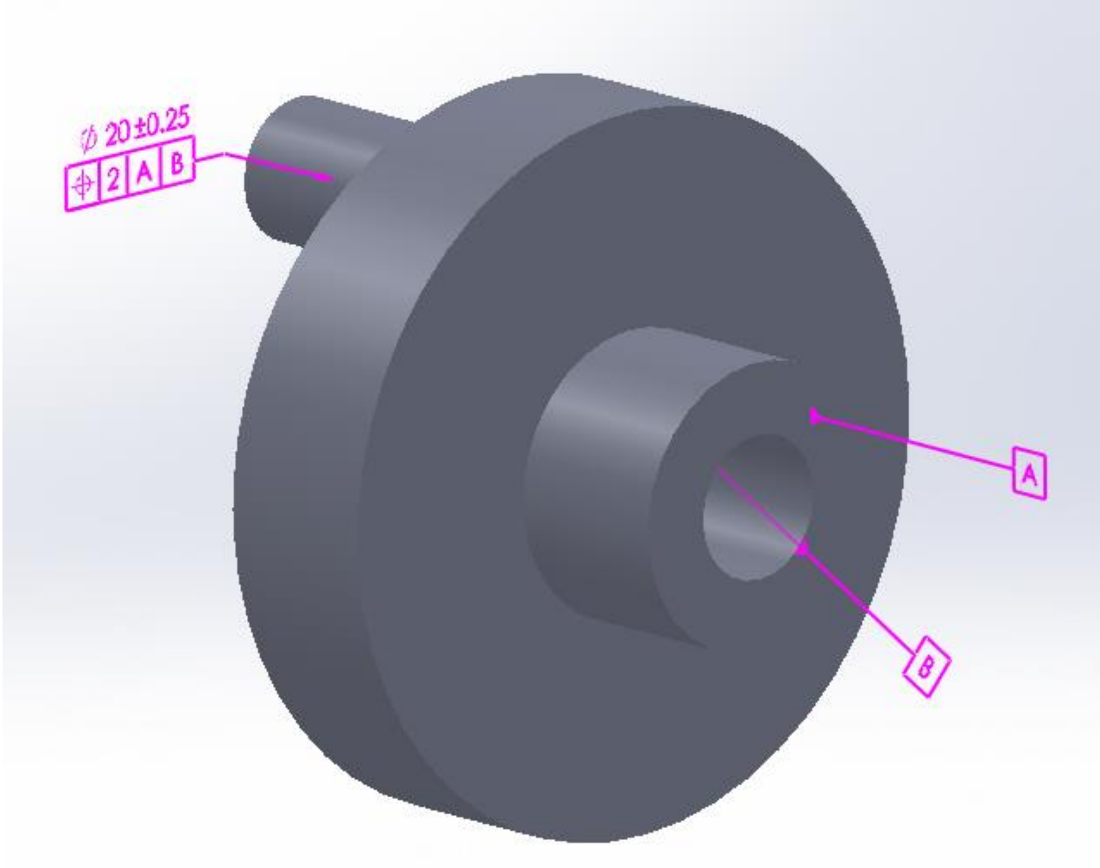
B: *Second step*



C: *Third step*



D: *Forth step*



E: *Fifth step*

Figure 31: *The CAD model of the five steps of machining the sample part with tolerances*

When all the errors for each step are interactively assigned to the geometry, all the information will be stored in CTF format, and will be saved in the computer with the same name (step number) and with .ctf extension. The CTF for the five steps of machining the example part are provided here. The information in each one are organized as discussed in chapter 3 of this thesis: 1. the name of the file, 2. part number (step number), 3. Constraints, and 4. Tolerances.

CTF for Step1:
ASU - Constraint - Feature - Graph
#1('part1', #2, #3);
#2=PLANE('FACE4_of_part0', (88, -50, 50), [-1, 0, 0]);
#3=PLANE('FACE5_of_part0', (0, -50, 50), [-1, 0, 0]);
#5=T_DIMENSION(#3, (nFI, 2, RFS), PD(#2, RFS));
#7=T_PARALLELISM(#2, (FI, 1, RFS), PD(#3, RFS));

CTF for Step2:
ASU - Constraint - Feature - Graph
#1('part2', #2, #3, #4, #5);
#2=PIN('(FACE8&FACE6)_of_part0', (64, 0, 0), [1, 0, 0], 20, 24);
#3=PLANE('FACE5_of_part0', (0, -50, 50), [-1, 0, 0]);
#4=PLANE('FACE2_of_part0', (88, 50, 50), [0, 1, 0]);
#5=PLANE('FACE1_of_part0', (88, 50, -50), [0, 0, -1]);
#6=CST_ANGLE(90, #2, #3);
#7=METRIC_RELATIONSHIP(#6, CST_ANGLE, (90, #2[LINE(axis of PIN)], #3[PLANE]));
#7=CST_ANGLE(90, #3, #4);
#8=METRIC_RELATIONSHIP(#7, CST_ANGLE, (90, #3[PLANE], #4[PLANE]));
#8=CST_DISTANCE(50, #2, #4);
#9=METRIC_RELATIONSHIP(#8, CST_DISTANCE, (50, #2[LINE(axis of PIN)], #4[PLANE]));
#9=CST_ANGLE(90, #3, #5);
#10=METRIC_RELATIONSHIP(#9, CST_ANGLE, (90, #3[PLANE], #5[PLANE]));
#10=CST_ANGLE(90, #4, #5);
#11=METRIC_RELATIONSHIP(#10, CST_ANGLE, (90, #4[PLANE], #5[PLANE]));
#11=CST_DISTANCE(50, #2, #5);
#12=METRIC_RELATIONSHIP(#11, CST_DISTANCE, (50, #2[LINE(axis of PIN)], #5[PLANE]));
#13=T_DIMENSION(#3, (nFI, 1, RFS), PD(#0, RFS));
#15=T_SIZE(#2, (nFI, 1, RFS));
#16=DOF(#15, (SIZE_DOF, SHAPE_DOF));
#17=T_POSITION(#2, (FI, 0.5, RFS), PD(#3, RFS), SD(#4, RFS), TD(#5, RFS));

CTF for step3:
ASU - Constraint - Feature - Graph
#1('part3', #2, #3, #4, #5, #6);
#2=PIN('(FACE6&FACE8)_of_part0', (64, 0, 0), [1, 0, 0], 20, 24);
#3=HOLE('(FACE10&FACE9)_of_part0', (0, 0, 0), [1, 0, 0], 10, 88);
#4=PLANE('FACE5_of_part0', (0, -50, 50), [-1, 0, 0]);
#5=PLANE('FACE2_of_part0', (88, 50, 50), [0, 1, 0]);
#6=PLANE('FACE1_of_part0', (88, 50, -50), [0, 0, -1]);
#7=CST_ANGLE(90, #3, #4);
#8=METRIC_RELATIONSHIP(#7, CST_ANGLE, (90, #3[LINE(axis of HOLE)], #4[PLANE]));
#8=CST_ANGLE(90, #4, #5);
#9=METRIC_RELATIONSHIP(#8, CST_ANGLE, (90, #4[PLANE], #5[PLANE]));
#9=CST_DISTANCE(50, #3, #5);
#10=METRIC_RELATIONSHIP(#9, CST_DISTANCE, (50, #3[LINE(axis of HOLE)], #5[PLANE]));
#10=CST_ANGLE(90, #4, #6);
#11=METRIC_RELATIONSHIP(#10, CST_ANGLE, (90, #4[PLANE], #6[PLANE]));
#11=CST_ANGLE(90, #5, #6);

```
#12=METRIC_RELATIONSHIP(#11, CST_ANGLE, (90, #5[PLANE], #6[PLANE]));
#12=CST_DISTANCE(50, #3, #6);
#13=METRIC_RELATIONSHIP(#12, CST_DISTANCE, (50, #3[LINE(axis of HOLE)], #6[PLANE]));
#14=T_SIZE(#3, (nFI, 1, RFS));
#15=DOF(#14, (SIZE_DOF, SHAPE_DOF));
#16=T_POSITION(#3, (FI, 1, RFS), PD(#4, RFS), SD(#5, RFS), TD(#6, RFS));
------------------------------------------------------------------------------------------------------------------------------------------
CTF for Step4:
ASU - Constraint - Feature - Graph
#1('part4', #2, #3, #4);
#2=PIN('(FACE11&FACE12)_of_part0', (0, 0, 0), [1, 0, 0], 50, 64);
#3=PLANE('FACE4_of_part0', (88, -50, 50), [-1, 0, 0]);
#4=PIN('(FACE6&FACE8)_of_part0', (64, 0, 0), [1, 0, 0], 20, 24);
#5=CST_ANGLE(90, #2, #3);
#6=METRIC_RELATIONSHIP(#5, CST_ANGLE, (90, #2[LINE(axis of PIN)], #3[PLANE]));
#6=CST_ANGLE(90, #3, #4);
#7=METRIC_RELATIONSHIP(#6, CST_ANGLE, (90, #3[PLANE], #4[LINE(axis of PIN)]));
#7=CST_DISTANCE(0, #2, #4);
#8=METRIC_RELATIONSHIP(#7, CST_DISTANCE, (0, #2[LINE(axis of PIN)], #4[LINE(axis of PIN)]));
#9=T_SIZE(#2, (nFI, 2, RFS));
#10=DOF(#9, (SIZE_DOF, SHAPE_DOF));
#11=T_POSITION(#2, (FI, 1, RFS), PD(#3, RFS), SD(#4, RFS));
------------------------------------------------------------------------------------------------------------------------------------------
CTF for Step5:
 ASU - Constraint - Feature - Graph
#1('part5', #2, #3, #4);
#2=PIN('(FACE13&FACE15)_of_part0', (40, 30, 0), [-1, -0, -0], 10, 40);
#3=PLANE('FACE4_of_part0', (88, -50, 50), [-1, 0, 0]);
#4=HOLE('(FACE9&FACE10)_of_part0', (0, 0, 0), [1, 0, 0], 10, 48);
#5=CST_ANGLE(90, #2, #3);
#6=METRIC_RELATIONSHIP(#5, CST_ANGLE, (90, #2[LINE(axis of PIN)], #3[PLANE]));
#6=CST_ANGLE(90, #3, #4);
#7=METRIC_RELATIONSHIP(#6, CST_ANGLE, (90, #3[PLANE], #4[LINE(axis of HOLE)]));
#7=CST_DISTANCE(30, #2, #4);
#8=METRIC_RELATIONSHIP(#7, CST_DISTANCE, (30, #2[LINE(axis of PIN)], #4[LINE(axis of HOLE)]));
#9=T_SIZE(#2, (nFI, 1, RFS));
#10=DOF(#9, (SIZE_DOF, SHAPE_DOF));
#11=T_POSITION(#2, (FI, 0.5, RFS), PD(#3, RFS), SD(#4, RFS));
```

After that all CTFs are created, by choosing the PCTF Icon in the testbed, the system allows the user to select multiple CTF files. By choosing the CTFs of all the machining steps in order of machining sequence, they will be compared by the program to create the PCTF. The software looks at the final step first; if the ID of any of the datums in the final step is found as the machined face in prior CTFs, it will store it in the PCTF. Then the system will do the same check for the datums used in the step before the last and continue till the second step. For example, in Step 5, there is a position tolerance with respect to face 4 and Face 9 &10 (highlighted). When these faces are tracked in previous steps, it can be seen that in step 3, Face 9&10 have position tolerance with respect to Face5 and Face 2 and Face1 (highlighted). Thus we have a datum transformation and error accumulation, therefore the information will be stored in PCTF. The following is the PCTF for this example:

```
ASU - Constraint - Feature - Graph
#1('part1', #2, #3);
------------------------------------------------------------------------------------------------------------------------------
#2=PLANE('FACE4_of_part0', (88, -50, 50), [-1, 0, 0]);          From first setup
#3=PLANE('FACE5_of_part0', (0, -50, 50), [-1, 0, 0]);

#4=PIN('(FACE8&FACE6)_of_part0', (64, 0, 0), [1, 0, 0], 20, 24);
#5=PLANE('FACE2_of_part0', (88, 50, 50), [0, 1, 0]);            From second
#6=PLANE('FACE1_of_part0', (88, 50, -50), [0, 0, -1]);

#7=HOLE('(FACE10&FACE9)_of_part0', (0, 0, 0), [1, 0, 0], 10, 88);   From third setup
------------------------------------------------------------------------------------------------------------------------------
#8=CST_ANGLE(90, #2, #3);
#8=METRIC_RELATIONSHIP(#6, CST_ANGLE, (90, #2[LINE(axis of PIN)], #3[PLANE]));
#9=CST_ANGLE(90, #3, #4);
#9=METRIC_RELATIONSHIP(#7, CST_ANGLE, (90, #3[PLANE], #4[PLANE]));
#10=CST_DISTANCE(50, #2, #4);
#10=METRIC_RELATIONSHIP(#8, CST_DISTANCE, (50, #2[LINE(axis of PIN)], #4[PLANE]));
#11=CST_ANGLE(90, #3, #5);
#11=METRIC_RELATIONSHIP(#9, CST_ANGLE, (90, #3[PLANE], #5[PLANE]));
#12=CST_ANGLE(90, #4, #5);
#12=METRIC_RELATIONSHIP(#10, CST_ANGLE, (90, #4[PLANE], #5[PLANE]));
#13=CST_DISTANCE(50, #2, #5);
#13=METRIC_RELATIONSHIP(#11, CST_DISTANCE, (50, #2[LINE(axis of PIN)], #5[PLANE]));
#14=CST_ANGLE(90, #4, #5);
#14=METRIC_RELATIONSHIP(#8, CST_ANGLE, (90, #4[PLANE], #5[PLANE]));
#15=CST_DISTANCE(50, #3, #5);
#15=METRIC_RELATIONSHIP(#9, CST_DISTANCE, (50, #3[LINE(axis of HOLE)], #5[PLANE]));
#16=CST_ANGLE(90, #4, #6);
#16=METRIC_RELATIONSHIP(#10, CST_ANGLE, (90, #4[PLANE], #6[PLANE]));
#17=CST_ANGLE(90, #5, #6);
#17=METRIC_RELATIONSHIP(#11, CST_ANGLE, (90, #5[PLANE], #6[PLANE]));
#18=CST_DISTANCE(50, #3, #6);
#18=METRIC_RELATIONSHIP(#12, CST_DISTANCE, (50, #3[LINE(axis of HOLE)], #6[PLANE]));
------------------------------------------------------------------------------------------------------------------------------
#19=T_DIMENSION(#3, (nFI, 2, RFS), PD(#2, RFS));
```

```
#20=T_PARALLELISM(#2,  (FI, 1, RFS), PD(#3, RFS));
#21=T_DIMENSION(#3, (nFI, 1, RFS), PD(#0, RFS));
#22=T_SIZE(#2, (nFI, 1, RFS));
#23=DOF(#15, (SIZE_DOF, SHAPE_DOF));
#24=T_POSITION(#2, (FI, 0.5, RFS), PD(#3, RFS), SD(#4, RFS), TD(#5, RFS));
#25=T_SIZE(#3, (nFI, 1, RFS));
#26=DOF(#14, (SIZE_DOF, SHAPE_DOF));
#27=T_POSITION(#3, (FI, 1, RFS), PD(#4, RFS), SD(#5, RFS), TD(#6, RFS));
------------------------------------------------------------------------------------------------------------------------------------------
#28=STEP(step_1, #2, #3);
#29=STEP(step_2, #4, #5, #6);
#30=STEP(step_3, #7);
```

Storing which faces was machined in which step

This PCTF represents a virtual part with all the final and intermediate faces of the example part under four machining operations, as shown in Figure 32. This PCTF also include the geometrical errors that need to be accumulated along with the sequence of the datum transformation. This PCTF can be the input to the Tolerance analysis testbed to do tolerance analysis like we do for any other part.

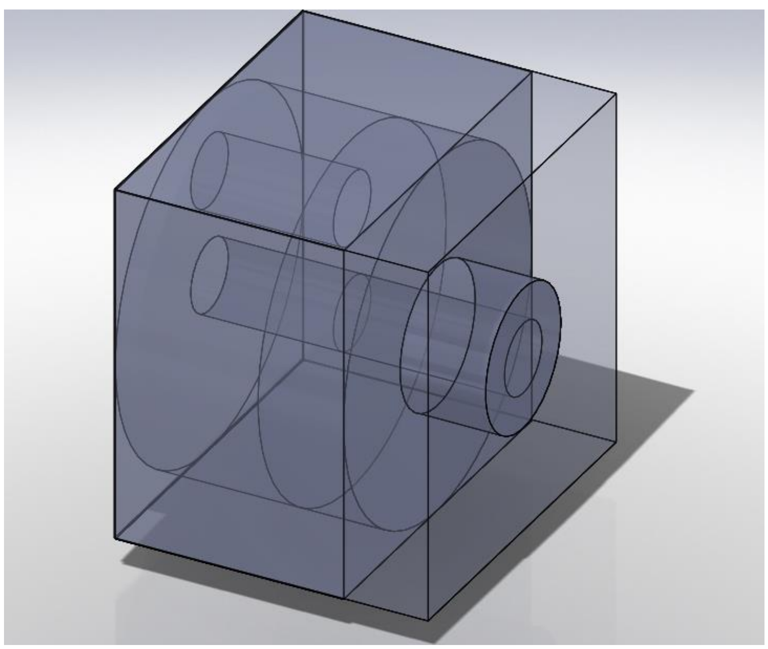


Figure 32: *Virtual part*

CHAPTER 7: CONCLUSION, LIMITATIONS/ASSUMPTIONS & FUTURE WORK

In current industry practice, there seems to be little time to objectively determine the goodness of a plan. In automated CAPP systems, minimization of production time appears to be the only measure of goodness used, and even that is applied to a few alternatives that have been generated in an ad-hoc manner. Another difficulty in evaluating process plans is that decisions are made at many different levels disconnected from each other, even though those decisions affect other aspects of the plan. The selection of stock, operations, setups, fixtures, machines, tools, machining sequence, cutting pattern, tool paths, etc. is done at different levels of planning, even though each decision constrains others. In this research a new approach to audit machining process plan by explicating the machining errors was suggested. This method is based on extracting the machining errors and representing them in the standard representation available for Geometrical and Dimensional tolerances. The System architecture, the data structure to store the Data, and the algorithms to create the m-maps, that are comparable with design tolerance maps, is presented in detail.

7.1 Conclusion

To verify a manufacturing process it is necessary to model the machining errors and analyze the correspondence of produced parts with the functional tolerances. In order to tackle this problem, we propose to use a semantic model for manufacturing errors which is consistent with our metric Bi-level model for GD&T, tolerance maps (T-maps). Since this new model maps the manufacturing errors into an n-dimensional Euclidian point space, we have named it ‘m-map’. In this research, we have outlined a framework for extracting the manufacturing errors and their corresponding datums. We have also designed a data structure (PCTF), that includes all the GD&T errors along with the geometry of the part under machining processes which is a general and machine readable data structure which can be used for any system and any tolerance analysis method.

This data can be used to automate conformance checking. Based on the PCTF we can find the m-maps that need to be created. Tolerances with no datum and errors that do not go through datum transformation have no m-map. Thus we search in PCTF for each dimensional or geometrical error on each feature (or face) that goes through datum transformation. When all the m-maps are found and accumulated m-maps are created, in the last phase, we can compare the m-maps with the T-maps. Basically if the m-map fit in the T-map we can say that in each direction, corresponding to one type of tolerance-error, the process plan can meet the design specification; if not, we may need to generate alternative plans to curb machine errors exceeding the allowed tolerances in design.

In current industry practice, there seems to be little time to objectively determine the goodness of a process plan. In automated CAPP systems, minimization of production time appears to be the only measure of goodness used, and even that is applied to a few alternatives that have been generated in an ad-hoc manner. Another difficulty in evaluating process plans is that decisions are made at many different levels disconnected from each other, even though those decisions affect other aspects of the plan.

7.2 Limitations and Future work

In this research the method proposed is a general and expandable approach for doing conformance analysis of process plans. We have talked about all types of geometrical and

Dimensional Tolerance errors to be extracted and a data structure that can convey all the information. Also the m-map model is a comprehensive model like t-maps, which has already been developed for most of tolerance types. As we have discussed, this work is divided into three steps. In the first step, we discussed how to extract errors from the process plan and developed library for fixturing methods and tables showing the errors correspondent to each machining process (machines, tools, etc used). These libraries and tables are now limited and do not cover all types of fixturing and machining process. These can be further studied and expanded to cover more cases. Also, with spit of the fact that we have shown how this step can be automated, it is considered to be interactive process. Thus a user has to extract the errors based on these libraries and heuristics, and assign them on the CAD model in test-bed. Therefore in the next level, this step can be automated too.

The second step is already and automated process, which takes in the PCTF and based on them, finds and creates the necessary individual m-maps. But the current state of the third steps is now a proof of concept which is limited to planar and cylindrical faces, since the mathematic model for tolerance transfer of only cylindrical and planar datum transformation are developed. Also Minkowski sum mathematical models for summation of m-maps are limited to planar faces for now. As a result, to have an automated system to work with all different types of features, other transformations and summation methods are needed.

REFERENCES

1. American National Standard ASME Y14.5 (2009). Dimensioning and Tolerancing, The American Society of Mechanical Engineers, New York

2. ISO 1101: Geometric Product Specifications (GPS) - Geometrical tolerancing – Tolerances of form, orientation, location and run-out, Int Stds Org.

3. Shah J, Ameta G, Shen Z, Davidson J, Navigating the Tolerance Analysis Maze, CAD & Applications Journal, V4 N5, pp 705-719, 2007.

4. Shen, Z., Ameta, G., Shah, J. J. and Davidson, J. K., 2005, "A Comparative Study of Tolerance Analysis Methods", ASME Transactions, J. of Computing & Information Science in Eng, V5(3), September, 2005.

5. T.j. Drozda, C. Wick, "Tool and manufacturing engineers handbook", Volume 1, Chapter 2, 1982.

6. Rashpal S. Ahluwalia and Andres V. Karolin, "CATC — a computer aided tolerance control system", Journal of Manufacturing Systems, 3(2), pp. 153–160, 1986.

7. J. K. Li, C. Zhang and Y. Z. Zhang, "Operational dimensions and tolerances calculation in CAPP systems for precision manufacturing", Annals CIRP, 38(1), pp. 403–406, 1989.

8. K. Whybrew, G. A. Britton, D. F. Robinson and Y. Sermsutianuwat, "A graph-theoretic approach to tolerance charting", International Journal of Advanced Manufacturing Technology, 5, pp. 175–183, 1990.

9. Zhengshu Shen, "Development of a framework for a set of computer-aided tools for tolerance analysis", PhD Thesis, 2005.

10. Y. Zhang, W. Hu, Y. Rong, David W. Yen, "Graph-based set-up planning and tolerance decomposition for computer-aided fixture design", International Journal of Production Research, Vol. 39, Iss. 14, 2001.

11. Shah, Y. Yan, "Representation and mapping of geometric dimensions from design to manufacturing", Proceeding of ASME DETC/CIE conferences, August 1996.

12. G.Thimm, J.Lin, "Redimensioning parts for manufacturability: a design rewriting system", The International Journal of Advanced Manufacturing Technology, August 2005, Volume 26, Issue 4, pp 399-404G.

13. Wilma Polini, Giovanni Moroni. Manufacturing Signature for Tolerance Analysis, J. Comput. Inf. Sci. Eng. 15, 021001 (2015) (5 pages); Paper No: JCISE-14-1300.

14. Krulikowski, A., 2002, Tolerance Stacks—A Self-Study Course, Effective Training, Westland, MI.

15. Shen, Z., Ameta, G., Shah, J. J., and Davidson, J. K., 2005, "A Comparative Study of Tolerance" pp. 247–256.

16. Joe Davidson, Jami J. Shah, T-Maps: A mathematical model for representing GDT Tolerance classes, Technical Report, Arizona state University, 2010.

17. Vemulapalli P, Mohan P, Zhang Z., Savaliya S., Shah, J. J., Machining Plan Variability Analysis for ifab**,** DARPA-ifab Project, Final Technical Report, ASU/DAL/ifab/2012-02-R1, May, 2012.

18. Rao S, Haghighi P, Shah J, Davidson, "Library of T-Maps for Dimensional and Geometric Tolerances", ASME Design Tech Conference, DETC/CIE, 2014, Paper 35273, Aug, 2014, Buffalo, New York.

19. Mohan P, Haghighi P, Shah J, Davidson J, "Automatic Detection Of Directions of Dimensional Control In Mechanical Parts", Proceedings of ASME 2014 International Manufacturing

Science and Engineering Conference, MSEC2014, June 9-13, 2014, Detroit, Michigan, Paper 4143.

20. Wu Y., Shah J., Davidson J., "Computer modeling of geometric variations in mechanical parts and assemblies", *ASME Transactions*, J. of Computing & Information Science, V3(1), March 2003.

21. Shen Z, Shah J, Davidson J, "Analysis neutral data structure for GD&T", *J. Intelligent Manufacturing*, V19 (4), pp 455-472, 2008.

22. Jian, K.., Davidson, J. K.., Jianhua, L. and Shah, J. J., 2005, "Using tolerance maps to validate machining tolerances for transfer of cylindrical datum in manufacturing process", Int J Adv Manuf Tech, 73: 465-478.

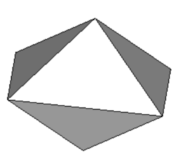
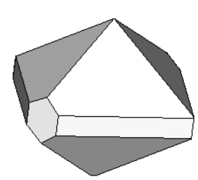
APPENDIX A

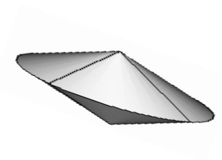
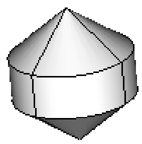
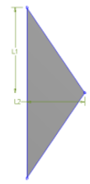
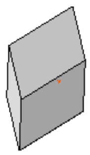
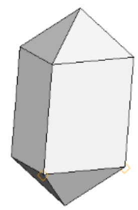
T-MAPS

T-map is a mathematical model for representing 3-D geometric variations in a hypothetical point space. Every Tolerance-Map is a convex set. The shape, size and internal subsets of T-Map represent the possible variations in size, form, orientation and position of the feature [18]. A general procedure for construction of T-Maps is given in reference [20]. T-map can handle tolerance interactions (MMC, LMC), form tolerances, floating zones and is consistent with Rule#1 in Y14.5 [1]. Also, since all the axes in T-Maps have same units, it makes feasible to compare different specifications on a feature. Also, to find the accumulated Tolerances zone of various tolerances, we can do Minkovski Sum of their T-maps. The resultant would be the worst case boundary for the toleranced feature. T-map makes stack-up relations apparent in an assembly, and these can be used to allocate size and orientational tolerances; the same relations also can be used to identify sensitivities for these tolerances. Stack-up relations are developed for parts where the centers of faces are offset laterally. All stack-up relations can be met for 100% interchangeability or for a specified probability.

A an example, out of the fifty T-map models that have been developed so far based on combinations of target feature, tolerance type and datum type, a few are shown in Table A1. More examples and description of T-maps are in the references. [20-22].

Table A1: *Library of a few T-maps*

T-map	Geometry, tolerance, datum	T-map	Geometry, tolerance, datum
	Geom: Rect bar; plane Tol class: size Datum: none		Geom: Rect bar; plane Tol class: size + orient Datum: planar face

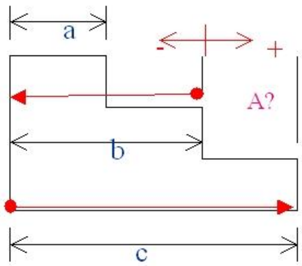
	Geom: Round bar; plane Tol class: size Datum: none		Geom: Round bar; plane Tol class: size + orient Datum: offset axis
	Geom: Round bar; plane Tol class: size + orient Datum: planar face		Geom: Planar circular face Tol class: circular runout Datum: axis
	Geom: traing bar; plane Tol class: size Datum: none		Geom: Rect bar; plane Tol class: size + orient Datum: two datums

APPENDIX B

TOLERANCE CHARTS

Traditionally, process planners verify manufacturing tolerances by conducting one dimensional (1D) analysis [1]. The 1D approach does not account for DRFs, datum precedence, zone tolerances and neglects contributors from other directions. That means they consider each 1D stack to be uncoupled from others. Designers also employ a variety of approaches, in some cases supported by computer software. This includes the 1D manual charting method, often taught in ASME professional development classes. However, unlike the process planner's 1D analysis method, such 1D charts consider all tolerance types as per the Y14.5 standard.

However, it is a manual bookkeeping procedure for 1-D stack calculation used typically with engineering drawings. A 1D coordinate system is set up with the origin at the left side of the unknown dimension as shown in Figure B1 (or lower end for radial stack), with positive direction to the right and negative to the left. The rationale for this convention is that if A comes out to be positive it is a clearance, and negative means interference (for assemblies); for part level analysis a negative value means that the feature disappears. A stack is a path from the origin to the other side of the analyzed dimension obtained by traversing a series of known dimensions. All tolerances encountered in traversing the stack are accounted for by rules that are specific to each class. The chart contains two main columns in which a value and a sign are entered for each tolerance contained in the stack, based on these rules. For size tolerances, the rule is that if the travel is in the positive direction, the maximum limit of the size is entered in the first column with a positive sign and its minimum in column 2 also with a positive sign. If travel is in the negative direction, the minimum value is put in column 1 and max in column 2, both with negative sign. The arithmetic sum of column 1 gives the max value of A and column 2 gives the min value of A. In the example shown, the stack is –b+c, so column 1 sum will be (-bmin+cmax) and column 2 will be (-bmax+cmin).



		COLUMN 1		COLUMN 2		
Part #	Stack Contributor	sign	**Maximum** max/min	sign	**Minimum** min/max	**Δ**
	TOTALS					

Figure B1: *Format of a traditional tolerance chart*

Space does not permit us to discuss rules for all geometric tolerance classes. Suffice to say that this method does take into account all Y14.5M rules, such as material conditions, Rule#1, bonus and shift. However, only worst case analysis can be done; no statistical analysis can be done since no algebraic expression for the analyzed dimension in terms of the contributors is generated by this method. Also, since some contributors are not aligned with the direction of analysis, they are ignored, which may yield incorrect results. Charts can also be constructed for worst case analysis of clearances in assemblies. The analyst mentally “positions” parts in a way that gives the worst cases (min. or max. value of analyzed dimension). Separate charts have to be done for each worst case.

Although tolerance charting, both design and process planning versions, are practiced primarily as manual procedures, there have been a few attempts at automating them. Notable research has been done by Ahluwalia and Karolin [6], Li and Zhang [7], Whybrew et al. [8], and Shen [9]. Besides, researchers have been trying to automate process planning and making interactive computer tools for it. Y. Zhang et al [10] proposed a computerized graph based setup/fixture planning using GD&T. Shah et al [11] have developed a dimensional model that facilitates the conversion of dimensions and tolerances from design models to machining features extracted automatically by a feature recognition system. For geometric tolerances, Thimm [12] explores a term rewriting system to derive alternative geometric and size design specifications with the aim of improving the manufacturability of a design.