UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

DECISION SUPPORT SYSTEM FOR FORM VERIFICATION OF MANUFACTURED PARTS

A DISSERTATION

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

degree of

Doctor of Philosophy

By

JUAN ANTONIO AGUIRRE CRUZ

Norman, Oklahoma
2007

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

DECISION SUPPORT SYSTEM FOR FORM VERIFICATION OF MANUFACTURED PARTS

A DISSERTATION APPROVED FOR THE
SCHOOL OF INDUSTRIAL ENGINEERING

BY

Dr. Shivakumar Raman, Chair

Dr. F. Hank Grant

Dr. Randa L. Shehab

Dr. Zahed Siddique

Dr. Theodore B. Trafalis

ACKNOWLEDGEMENTS

I want to thank God and Our Lady of Guadalupe for letting me enjoy each day that I have lived, each person that I have met, for the good times that had helped me be happier, and the bad times that had helped me be stronger. Special thanks to my parents and my sister that have been there for me cheering and supporting me through all my life. You are all my best friends and my role models. To the youngest member of our family, we love you and we remember you always.

My most sincere appreciation to my advisor and good friend Dr. Shivakumar Raman that gave me the opportunity to grow as an engineer and as a person. His support and guidance were an invaluable part for the development of the topics and projects that we worked on during the time I spent at the University of Oklahoma. In addition, I want to thank the faculty of the Industrial Engineering department, especially, the members of my Ph.D. committee, Dr. Randa Shehab, Dr. Theodore Trafalis, Dr. Hank Grant, and Dr. Zahed Siddique for their support and the time they gave me during this research. Also to Cheryl Carney, Jean Shingledecker, and Amy Piper for their assistance and friendship.

Last but not least, to all my friends, for their caring and constant support.

Partial financial support for this project was provided by the National Science Foundation (NSF), through grant DMI-0217563. It must however be noted that the opinions expressed in this dissertation are those of the author and not necessarily those of the NSF.

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

ABSTRACT

The form verification of manufactured parts is a process composed of a set of operations that are expensive and yet add no value to the product. Yet, the resources used to inspect the parts add a small but significant amount of noise that can affect the outcome of the process. For this reason, this research provides guidelines to effectively perform the inspection process by suggesting new mathematical models and approaches that can be used for the creation of a decision support system that can assist in the verification of the accuracy of machined parts.

The main goal of this research is to develop procedures that are simple to implement but at the same time are robust enough to provide reliable information that help the metrologist to make accurate decisions about the inspected parts. Form features such as spheres, cylinders, cones, frustums, and torus forms are commonly used to design complex parts. However, the procedures to verify most of these form features have not been developed yet by the national standards. Therefore, this research proposes new mathematical models that combine the concepts of analytic geometry and optimization to provide optimal solutions.

This research proposes two approaches to improve the robustness of the mathematical models from the noise induced by the inspection process. The Dynamic Angle Approach (DAA) and the Free Form Orientation approach (FFO) presented here focus on finding the parameters of the axes and origin of the form that counteract the inaccuracies of the inspection equipment.

The proposed approaches and mathematical models were verified using measurements from features that were perfectly aligned with the coordinate system of the inspection equipment and from features that were intentionally misaligned. The results showed that the models were accurate and robust enough to estimate the parameters and zone of error of the form features and they performed better than existing models.

In summary, this research suggests formalized methods for feature extraction, sampling, path planning, and form fitting, although the last mentioned received the most attention. It is believed that this comprehensive, integrated analysis will lead to the development of a decision support system.

CHAPTER 1

INTRODUCTION

1 Summary

The inspection process is a non-value added operation that is necessary to determine the quality and good reliability of the parts that are being produced. The form verification of machined parts is a type of manufacturing tolerance that focuses on evaluating if the surface of the part follows a desired individual geometry. The evaluation of this tolerance requires a set of operations for comparing the surface against an ideal form in order to determine the form deviations of the part. However, most of the operations used to collect the data from the surface are susceptible to random and systematic errors (Barry, 1978). For this reason, the present study proposes a mathematical methodology for each of the operations of the inspection process to determine an accurate procedure that can be used to create a decision support system for the form verification of manufactured parts.

1.1 Form Tolerance Verification Process

The accuracy of the form of manufactured parts is affected by the parameters of the manufacturing and inspection processes, such as the machining equipment, cutting tools, machining parameters, inspection equipment, the number of sampling points inspected, the location of the sampling points, the mathematical model, and the fitting algorithm. Although, the inspection process is different than the machining process, the latter has a direct impact on the surface of the part, because the conditions used to machine a workpiece leave a unique fingerprint over the surface that affects the tolerances (Hocken et al., 1993). Thus, the inspection process must be adaptable according to the manufacturing process that creates the part.

The inspection equipment is one of the most important factors in the inspection process since it is the interface that collects the measurements from the part, and its accuracy and precision are reflected on each inspected point. An example of the inspection equipment is the coordinate measuring machine (CMM) that inspects the part by collecting points around the surface in a discrete manner. Although the CMMs are highly used in academics and in industry to verify parts, they are affected by random and systematic errors that cause them to wrongly estimate the tolerances (Barry, 1978; Hocken et al. 1993). The factor sample size also affects the outcome of the process because it is used to determine the amount of sampling points that should be inspected from the part. If a small sample size is used often, lack of information leads to ineffective representation of the geometry of the part. On the other hand, if a higher sample size is chosen then more information is available but the inspection time and cost tends to increase (Hocken et al., 1993). The location of these sampling points plays a significant role in the inspection process because smaller samples sizes that are properly distributed over the surface of the part can be more efficient than larger datasets improperly distributed. For this reason, the inspection process should require a level of planning based on the geometry and the dimensions of the part that can be used to extract as much information with the optimal use of resources.

The mathematical model is used to determine the deviations of the part by comparing the inspected points against an ideal form. The accuracy and robustness of the model determines the amount of error contained in the part. Thus, a good model is one that produces the smallest error (precision) and the one that estimates the parameters of the inspected part closer to the results obtained from the size tolerance (accuracy). The estimation of these parameters is performed by a fitting algorithm that uses optimization methodology to find the magnitude of the parameters that minimizes the overall error of the part. Therefore, if the mathematical model accurately represents the geometry of the part then any fitting algorithm can provide precise and accurate results. However, if the model is not good enough then powerful fitting algorithms should be used to obtain a good solution that is accurate and precise. Therefore, the mathematical models and fitting algorithms are very important because a poor selection of these can cause an over- or under-estimation of the zone of error, which can be the difference between rejecting or accepting a part (Orady et al., 2000). For this reason, it is important to study each one of these factors in order to identify the correct factor combination that accurately determines the quality of the inspected part.

1.2 Problem Definition

In industry, there are several computer programs that automate and provide decision aid for the inspection process, such as Pro/E^{TM} and PC-DMIS^{TM}. Pro/E^{TM} is a computer aided design, manufacturing, and inspection (CAD/CAM/CAI) program that allows the user to create a solid model of the part, generate the NC sequences, and create the inspection path planning, while PC-DMIS^{TM} that stands for personal computer-dimensional measuring interface standard and is a metrology software that is used to evaluate manufacturing tolerances and the quality of the manufactured parts. Although these computer programs are very sophisticated at optimizing the inspection process, the algorithms and mathematical models that they use are proprietary, thus the parameters used to calculate the tolerances are unknown. On the other hand, in metrology literature, studies have focused on automated inspection systems, such as the computer aided inspection (CAI), which has been defined as the element that closes the loop between the design (CAD), the process planning (CAPP), and the manufacturing (CAM) systems (Cowling et al., 1989; Anjanappa et al., 1996; Jiang et al., 2002). The focus of these papers was to improve and automate the inspection process in general, but not on studying and improving the operations that compose the form tolerance verification process.

The form tolerance verification of spheres, cylinders, cones, frustums, and torus shapes requires a set of operations to collect data from the part and analyze them, to determine the accuracy of the forms. The resources used to collect the data often induce errors that have a direct impact on the output of the inspection process. These forms have not been studied at any length in the literature. Further, existing mathematical models to inspect these shapes are not robust enough to counteract the errors induced by the inspection equipment such as the improper alignment of the coordinate system of the part with the coordinate system of the inspection equipment. This problem leads to an over- or under-estimation of the width of the zone of error and an inappropriate estimation of the parameters of the inspected shape.

Most studies based on determining the efficiency of the mathematical models and fitting algorithms only analyze the width of the zone of error. The only information obtained from the width of the zone of error is the variability of the deviations within the tolerance zone. Thus, these studies focus only on the precision of the mathematical models and fitting algorithm but not on their accuracy.

1.3 Objective

This thesis aims to study the modules that compose the inspection process and to develop new mathematical formulations that can counteract the errors induced while inspecting the part. The modules include the developments of:

1. A feature recognition program of spheres, cylinders, cones, frustums, and torus forms from the CAD model.
2. A path plan that reduces the time of inspection and inaccuracies of the inspection equipment by collecting the data points using a vector that is normal to the surface of the part.
3. New mathematical models for spheres, cylinders, cones, frustums, and torus forms that are robust and accurate to estimate the parameters of the parts.
4. Two approaches that counteract the axis misalignments and origin offsets caused by setting up the part on the inspection equipment.
5. The development of residuals for the analysis of the data to provide information about the accuracy and precision of the existing and proposed mathematical models.

Therefore, the improvements and developments of each module can be automated for the creation of a decision support system.

1.4 Contributions

The principal contribution of this thesis is the development of new mathematical models and approaches for inspection of complex forms such as spheres, cylinders, cones, and torus forms. This is evidenced by the lack of standards in the inspection of these forms. A second important contribution is the development of methodology to compensate for errors caused by axis misalignments and/to Limacon approximation. The Limacon approximation has been used by Samuel et al. (2003) for circles; Kanada (1995) and Samuel et al. (2003) for spheres; Shunmugam (1987), Prakasvudhisarn (2002), and Prakasvudhisarn et al. (2004) for cones; and Aguirre Cruz (2003), Aguirre Cruz et al. (2005) for torus forms. These models are accurate as long as the origin of the CMM is properly transformed to the position of the geometric origin of the part. If both origins are not in the same position, then the performance of the models will decrease. According to Chetwynd (1979) the resolution of the approximation depends of the relation between the distance from the origin of the inspection equipment to the geometrical origin of the part and the ideal radius of the circular shape. This means that if the distance between the origins is smaller then the performance of the approximation will increase. This thesis proposes an approach called the dynamic angle approach (DAA) that focuses on improving the performance of this approximation by mathematically reducing the distance between origins regardless of the starting location of the two origins. Furthermore, the present research proposes a new approach called the free form orientation approach (FFO) that uses the principles of the dynamic angle approach and the flatness model to counteract the effect of the errors induced by the CMM.

CHAPTER 2

LITERATURE REVIEW

2 Summary

The literature review of this research is divided in seven sections, beginning with Decision Support System (DSS) that focuses on the studies that provided details about the history and guidelines that measure the efficiency of a good decision support system. At the same time, this section focuses on the studies that incorporated the decision support system principles on the tolerance verification process, such as the Computer Aided Inspection (CAI). The second section, Geometric Modeling (GM), studies the different techniques used to represent solids and surfaces on the 3-Dimensional environment, focusing on the Boundary Representation, also known as B-rep, which is one of the techniques most used in the literature. The third section is Feature Recognition (FR), which looks at the algorithms and procedures used to extract features of CAD models. The fourth section deals with the Coordinate Measuring Machine (CMM), which is the most common equipment used for inspecting tolerances of manufactured parts in industry. Form tolerance inspection is the fifth section focusing on a survey of previous studies on spherical, conical, and torus form feature inspection. The last two sections present Sampling Strategies and Fitting algorithms. The former focuses on the different sampling sequences that capture information from manufactured parts; while the latter aims to explain the mathematical algorithms used in previous studies for the calculation of the form tolerance.

2.1 Decision Support Systems

A decision support system (DSS) is an interactive computational tool used to assist users in the decision making process. The DSS manages and analyzes data using logical and mathematical algorithms; in addition, the system is able to present reliable information through a Graphic User Interface (GUI) (Sprague, 1980; Alter, 1977).

The DDS is the result of the development of two systems, Electronic Data Processing “EDP” and Management Information System “MIS”. The scope of the EDP was mainly to handle the performance measures at the bottom part of the organization, as well as storage and data processing, (Sprague, 1980). As time passed by, the needs of managing the information in a broad way became an issue for the EDP. For that reason a new platform with a superior scope was developed, called Management Information System. The goal of MIS was to regulate the flow of information from the operational levels to the managers, as well as to assist middle managers to present to top executives reliable and specific information about the behavior of the company. Although MIS introduced great advances in knowledge and technology, it was not powerful enough to assist top managers in the decision making process. This was the reason that triggered the development of a new system called Decision Support Systems, built using the EDP and MIS principles. The new aspect of these systems was the introduction of an interactive interface and advanced mathematical models capable of analyzing and displaying information to support the decision making process. The interaction capability upgraded this system into a more flexible, adaptable and multi-scenario platform, allowing the user to understand the problem and to determine the performance of different “what if” scenarios (Sprague, 1980; Fang, 2003a).

The study of Decision Support Systems was too broad in the sense that it involved both managerial subjects and advanced engineering topics. Alter (1978, 2002) studied these systems in a deep manner, focusing more on the managerial aspects and the man-machine interface; from the development of elements and characteristics to the implementation and improvements of the system. He found that an efficient DSS has to "Give Answers" and "Ask for Answers." "Give Answers" means that the system should be able to answer specific questions from the user in an accurate and precise manner; and "Ask for Answers" means that the system has to have enough resources such as menus, toolbars, and textboxes, to ask the user for specific data to generate the customized reports that the user requires.

Alter (1977) mentioned that DSS should be a better responsive system rather than an interactive system; where responsiveness is defined as a combination of power, to answer the most important questions; accessibility, to provide answers in the time given; and flexibility, to adapt to changing situations. Therefore, the best systems should be capable enough to understand, analyze and present data in the most robust way to facilitate the decision making process.

A decision system as any other system requires some kind of input or data that in fact is the output of another system; these data are evaluated and transformed into a desired output or useful information. Specifically, the data are samples collected from a particular system that have no meaning and cannot be used to make inferences about that system until some process of pattern recognition and analysis is applied to it. Once the data are processed, it becomes valuable information that shows patterns, relations and interaction between variables. However not all the data are good enough to be transformed into valuable information. The quality of the data is crucial because it will affect the output of the system.

The DSS have several stages that help to process data and generate information, such as classifying the data, applying mathematical algorithms, and presenting information to the user. In literature, many authors have studied each of these; some focused on studying better ways to handle the data using advanced databases; others studied applying accurate algorithms to look for patterns and extract as much information as possible. Further studies were developed on better man-machine interfaces, as well as on decision support systems as a managerial tool.

The design and application of DSS depends on the industries in which it is applied, and the dimension of the problem to be studied. The dimension of the system depends on the amount of variables and modules that the system has; if the number of variables increases then more powerful database algorithms are needed to process the data and to manage the information provided to the user. For example, Bohanec, et al. (2004) studied hierarchical multi-attribute decision models, which is a technique used to classify a large amount of data based on the value of the input variables. A good example of this technique is the process that a university follows to accept or reject applications of prospective students. The information of these applications is introduced into the computer and the information contained in it is evaluated and classified according to the standards of that school. However, the user or person in charge of the system can interact with the program and decide whether or not that person is accepted. This technique reduces the processing time for a large amount of data by classifying it according to a hierarchy (Meystel, 2003).

Using a different technique, Ferguson, et al. (1969) developed a decision system to schedule jobs in a production line, where the goal of the system was to generate an optimal and feasible schedule for all jobs, based on the simulated problems of a manufacturing environment and user interactions. However, this type of system can be very complex since the production environment has many variables that affect the production process. The production planner should be very organized and pay attention to all changes in the environment to manage well the system and to keep it updated and accurate. Pillai (1990) create a similar system for an automated wafer process, where the main decision system was created from small systems that manage different departments of the factory, such as cost, material handling system, yield, process, equipment utilization, among others. These modules allowed the systems to decompose and classified the data according to a hierarchy, evaluating the risk of each proposed decision, in order to provide the user with the decision that benefits all parts of the company.

In a similar way, as the principles of the decision support systems are applied to the production process, these can also be applied to develop systems capable of solving the common but complex problems in the metrology area, such as the number of sample points and the sampling strategy needed to capture the deviations from a machined part.

2.1.1 Computer Aided Inspection (CAI)

The inspection process is composed of a set of repetitive activities that determine the quality of machined parts. As mentioned before, the inspection process does not add any value to the product, since it is only for verifying that the machined parts are within the specified tolerances. For this reason, it is important to reduce the inspection time. One way to solve this problem is through the use of automated operations that will aid the user to speed up the process and accurately determine the integrity of the part. Computer aided inspection (CAI) is a buzzword that describes the inspection process as a system that uses the computer to automate operations. Many authors have defined computer aided inspection (CAI) as the bridge that closes the loop between computer aided design (CAD), computer aided process planning (CAPP), and computer aided manufacturing (CAM) (Cowling et al., 1989; Anjanappa et al., 1996; and Jiang et al., 2002). This means that the gap between the conceptual world (design) and the real world (manufacturing) is filled by the inspection process because it measures the machined part and compares its measurements against those dimensions required by the conceptual world. The differences between these two worlds are analyzed by an expert system that sends the feedback to the manufacturing processes in order to make the preventive or reactive changes, such as correcting the parameters of the manufacturing process, checking if the tool is still in good condition, checking the dimensions of the workpiece, and verifying the alignment of the workpiece, among others. The main advantage of the automated inspection is the time and cost that will be saved because all feedback analysis will be done by powerful mathematical algorithms that will detect the problems of the manufacturing processes (Anjanappa et al., 1996).

In the literature, there is an extensive collection of articles that focus on the computer aided inspection (CAI) systems Examples of these researches include, ElMaraghy et al. (1987) who developed an expert system for modeling and inspection planning using artificial intelligence techniques; Cowling et al. (1989) proposed an interface between the CAD system and the coordinate measuring machine, in which the measured points were used to create a “reformed model” based on Bezier curves that was compared against the CAD model to verify the quality of the part; Anjanappa et al. (1996) developed a data analyzer to evaluate the results obtained from the inspection process and to recognize patterns in the residuals in order to determine the cause of the defective part; Moroni et al. (1998) created an automated inspection system for non-rotational parts. The system focused on reading the CAD file, extracting the form features, creating the sampling plan, and generating the necessary NC code to inspect the part; and Jiang et al. (2002) proposed a statistical tool to determine the number of measuring points needed to determine the quality of the part, which was based on the features contained on the CAD model. Professional software such as ProE^{TM} and PCDMIS^{TM} focus on providing tools to automate the inspection process, however, as mentioned before the algorithms that they used are proprietary. Thus it is not possible to access the procedures that they use to execute the subroutines.

ElMaraghy et al. (1987) created an expert system for inspecting rotational and prismatic parts by using artificial intelligence techniques. The system consisted of modeling, feature recognition, and inspection planning blocks. The modeling block was used to analyze the features contained in the CAD file in order to establish relationships between the features contained in the part, and between manufacturing processes, geometric tolerances, and the features. The features were extracted from the CAD file using feature recognition algorithms. Then the features were clustered according to the tolerances that were required for inspection of the features. The author mentioned that the coordinate measuring machine software did not have subroutines to calculate some of the 3-Dimensional tolerances. Therefore he used a combination of 2-Dimensional tolerances to inspect the surfaces. In addition, the author used the same analysis for other surfaces since it was found that the 2-D analysis was faster and more efficient than the 3-D analysis. The inspection of the features was defined by the inspection planning block based on a set of principles: first, the features were divided into CMM features which can be measured using the CMM while the non CMM feature have to be measured using a different technique. Second, the inspected part was positioned on the measuring table of the CMM where the accessibility of each feature was verified. Third, the features that needed datum verification were identified. Fourth and fifth, selecting the proper probe for each feature, and clustering all features that were measured with the same probe so that all are measured together. The process was repeated until all features were measured. Then the part is reoriented on the measuring table and the remaining features are measured until all are inspected. The set of principles was coded in a logic programming language called Prolog, which stored the rules and artificial intelligence algorithm. The author mentioned that the efficiency of the system depended in great part on the selection of features that served as datums for others, because these determine the plan of action and the inspection order.

Cowling et al. (1989) proposed an interface between the CAD system and the coordinate measuring machine, in which the measured points were used to create a "reformed model" based on Bezier curves that was compared against the CAD model to verify the quality of the part. The author defined a model for each of the computer aided systems, "original CAD model" for CAD, "physical model" for CAM, and "reformed model" for CAI. The last model was created using the data points collected with the CMM, which were interconnected using Bezier rational square segments and B-spline curves. At the same time, the Bezier and B-spline curves were translated and rotated in order to correct some of the errors induced by the measuring table and the inspection instrument. The edges of the “reformed model” were reshaped according to the “original CAD model,” thus, the sharp or curve edges of the part were inspected thoroughly in order to help the fitting algorithms to properly fit the data points into the shape of the CAD model. The interface was verified by simulating the measurements of the CMM and fitting them into the CAD model to obtain the “reformed model.” Results from this test showed that the differences between the CAD and the reformed model represented the manufacturing errors, which were identified in terms of geometric features. It is important to notice that the differences between the CAD and the reformed model are equal to the manufacturing errors plus the errors due to the inspection process. The author did not recognize the errors due to inspection which can make the interface not very reliable since it could provide false alarms on the machining process.

Anjanappa et al. (1996) developed a data analyzer to evaluate the results obtained from the inspection process and to recognize patterns in the residuals in order to determine the cause for the defective part. The author proposed a computer aided inspection data analyzer (CAIDA) that focused on analyzing the residuals from parts that were nonconforming, in order to determine the cause of the errors and to elaborate the corrective actions. The system was created to recognize problems of prismatic parts such as slots, pockets, holes, and boundaries. The prismatic parts were created using a milling process which was analyzed in order to create the knowledge base that will recognize the cause of the errors. The author classified the milling process errors under four types: cutting tool errors (size, runout, wear, and deflection), machine errors (positioning, and thermal), fixture errors (part/fixture location, and chip control), and miscellaneous errors (stock size, workpiece deflection, and chatter). The errors were analyzed separately and combined, since errors can happen one at a time or with more than one present on a machined part. The data analyzer was tested by designing prismatic parts using the software Cadkey. The modeled parts were exported to a flexible manufacturing cell using the IGES exchange file. The features of the parts were extracted using an intelligent feature extractor that generated the process plan including the NC codes and the machining parameters. A CNC milling center was used to machine the two parts using aluminum alloy 6061 T6. The two parts were machined with known errors in order to verify whether the data analyzer was able to capture those inaccuracies. The two parts were inspected using a touch probe mounted on a coordinate measuring machine. Results from this experiment showed that the data analyzer was able to recognize the tool size error, tool runout, fixturing error, and boundary error. However the analyzer was not able to recognize tool wear since it was found that the magnitude of these errors was too small compared to the resolution of the CMM.

Moroni et al. (1998) created an automated inspection system for non-rotational parts. The system focused on reading the CAD file, extracting the form features, creating the sampling plan, and generating the necessary NC code to inspect the part. The proposed system was written in C++ and used the ACIS kernel to represent the solid parts by its boundary. Firstly, the author defined all non-rotational form features in terms of the boundary representation entities and other features. The system was able to recognize the following features: through hole, blind hole, hole, face list, through pocket, blind pocket, pocket, through slot, blind slot, slot, through step, blind step, step, round boss, non round boss, surface, parent form feature, and form feature. The system interface allowed the user to choose either the entities or the features that had to be inspected. The system checked for the accessibility of the features and determined the geometrical and topological entities that created the feature. The interface was not completely automated since the user had control over the selection of the features. However, the author found that the user interaction was a plus in the system, since it allowed selecting of the features that the user needed instead of spending time on verifying all features contained in the part.

2.2 Geometric Modeling

The Computer Aided Design and Computer Aided Manufacturing (CAD/CAM) systems are important elements in the development and growth of the technological advances in many industries. The integration of the computer into the design and manufacturing areas has allowed engineers to design and create products in a virtual environment, in which products can be tested under different virtual scenarios which will guarantee that the product will meet customer requirements and that it will be functional. In addition, the virtual designs can be exported as a set of instructions that are understandable by the machining processes, which perform the material removal operations under a set of optimized parameters established in the design phase. Therefore, once that the part leaves the production area, it is assumed that it has all quality characteristics and that the part will meet customer specifications. Unfortunately, there are several causes of variability that disrupt the performance of the ideal world. For this reason, there are a significant number of studies performed in the design and manufacturing areas that range from providing better design tools, to the development of decision support systems to accurately inspect the form error of parts, which is the goal of this research.

The geometric modeling is considered an important part of the CAD/CAM systems, since it is a way to represent geometric objects in a virtual environment, where physical properties and surface characteristics are embedded to the designed objects with the purpose of determining the best way to design and manufacture the object (Zeid, 1991). Geometric modeling can be considered an art since there are multiple ways in which one part can be designed. Thus, geometric modeling programs can represent an object in three different types, such as wireframes, surfaces, and solids. In the present study, the form tolerance evaluation is performed by extracting the form features of a modeled part, with the purpose of comparing the measured object against the CAD model in order to determine if the part is out of tolerance. Out of three types of geometric modeling, the solid modeling will be a great approach to represent the parts, since it stores more information from the designed object than the wireframe and surface models, which will simplify the feature recognition process, as is mentioned in the next lines. Although, solid modeling is the focus of this study, the wireframes and surface models are briefly discussed.

2.2.1 Wireframe modeling

The wireframe models are created by emulating the shape of the object using a virtual frame of wires, which are bent according to the edges and curvature of the desired design (Zeid, 1991; Coulibaly, 1998). The wireframe models uses points, lines, arcs, circles, conics, and curves, to reconstruct the shape of the object. The advantages of this type of models are that they are simple and do not require much time and computer memory to run. However, the models are not very friendly to the human eye since they become difficult to interpret as the complexity of the design increases. The models can be constructed with a set of analytic entities such as points, lines, arcs, circles, fillets, chamfers, and conics. If the design is more complex it can be built based on synthetic entities, such as cubic spline, B-spline, beta-spline, nu-spline, and Bezier curves. Figure 1 shows an example of a wireframe model.

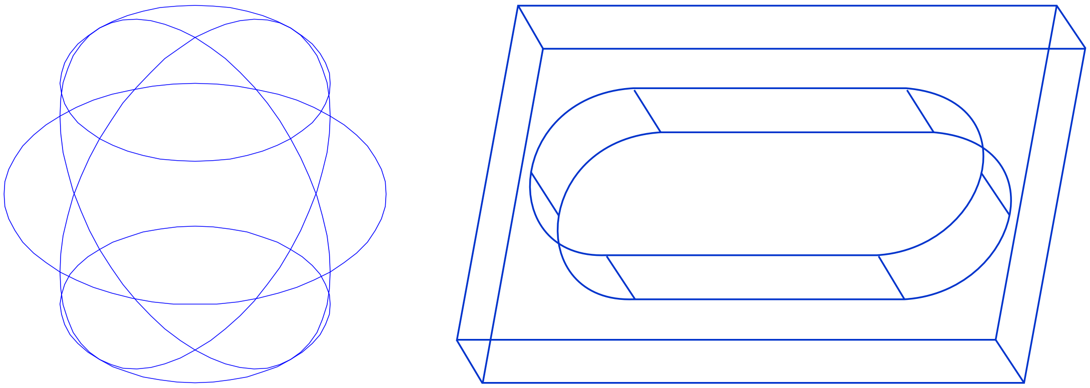


Figure 1. Parts created using wireframe modeling, a sphere on the left and a block with a slot on the right.

2.2.2 Surface modeling

The surface models are more complete than the wireframe models, since they include more information about the connections of objects. The surface models represent the designed part using a set of artificial lines called *mesh*, which break the design into a network of interconnected patches (Zeid, 1991). The advantages of the surface models are that the models can be used for volume and mass property calculation, finite element modeling, NC path generation, cross sectioning, and interference detections; also the appearance of the surface models is more realistic than the wireframe models. A disadvantage of these models is that the mesh will consume a significant amount of computer memory, since more information is needed to represent the object and this type of modeling does not have material properties such as density (Coulibaly et al., 1998). The surface models are constructed by analytical entities, such as plane surface, ruled surface, surface of revolution, and tabulated cylinder, and synthetic entities, such as bicubic Hermit spline surface, B-spline surface, rectangular and triangular Bezier patches, rectangular and triangular Coons patches, and Gordon surface. An example of a surface model is seen in Figure 2.

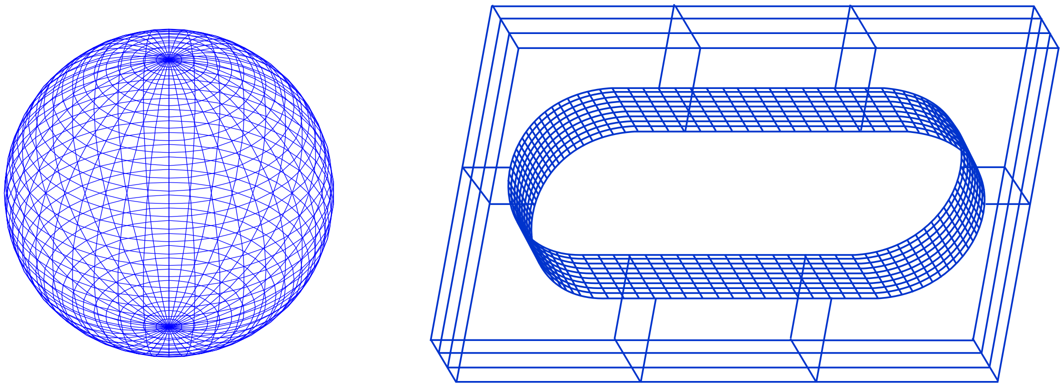


Figure 2. Parts created using surface modeling, a sphere on the left, and a block with a slot on the left.

2.2.3 Solid modeling

The solid modeling, also called volumetric modeling, is a modeling tool that represents designs in an unambiguous way, since it organizes and records the location of every entity in space, whether it is located inside, outside, or on the object because of spatial addressability. The wireframes and surface models are less powerful than the solid models, since they do not have the spatial addressability property and they only store the dimensions of the entities also called geometrical information. On the contrary, solid models have many advantages, such as the spatial addressability property, the models contain all physical properties for volume, weight, and inertia calculations, NURBS (Non-uniform rational B-spline) technology can be implemented in the model, and they store the dimension of the entities called geometrical information, as well as the relation between the entities also called topological information, (Zeid, 1991; Coulibaly et al., 1998; Ismail et al., 2005). The difference between geometry and topology is presented in Figures 3 and 4, which show an example between two parts that have the same geometry but a different topology, and parts with the same topology but a different geometry, respectively.

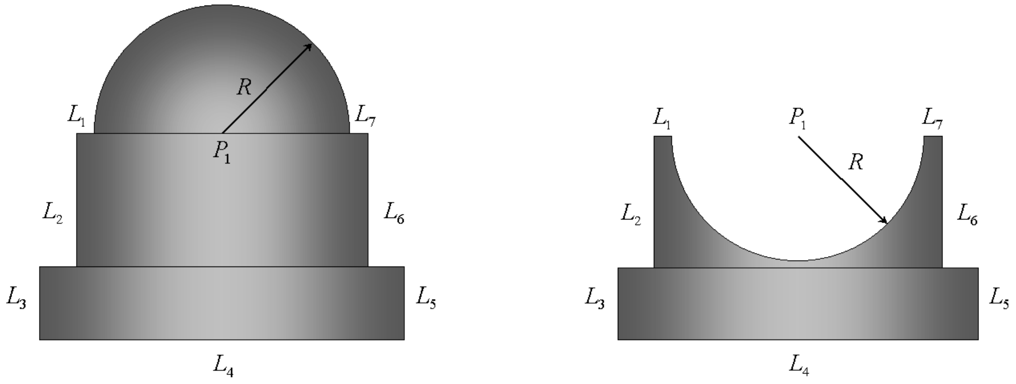


Figure 3. A part with the same geometry but a different topology. (Adapter from Zeid, 1991)

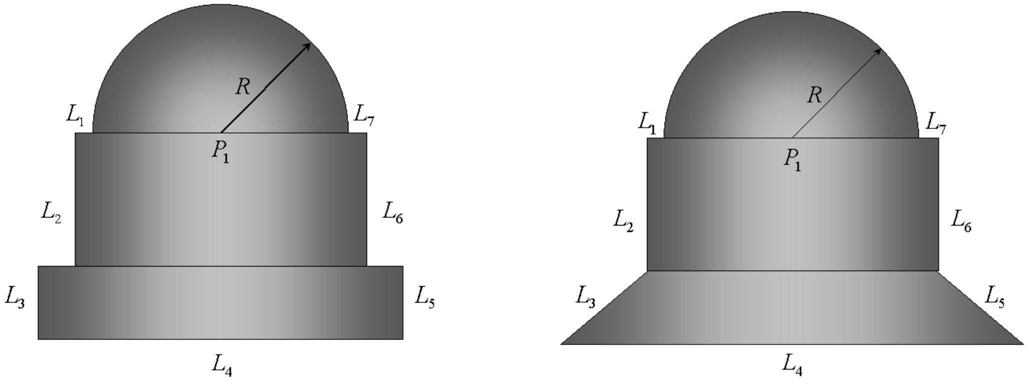


Figure 4. A part with the same topology but a different geometry. (Adapter from Zeid, 1991)

The solid models, due to their properties and characteristics, are used for mass property calculations, finite element modeling, computer aided process planning, and machine vision. The solid models are constructed based on solid entities called primitives, such as block, cylinder, cone, sphere, wedge, and torus; where the first four primitives are the most used since almost all of the mechanical designs can be represented using these shapes (Zeid, 1991). These primitives are mathematically combined to represent an object using the set theory by means of the Boolean operations, such as union, intersection, and difference, as shown in Figure 5.

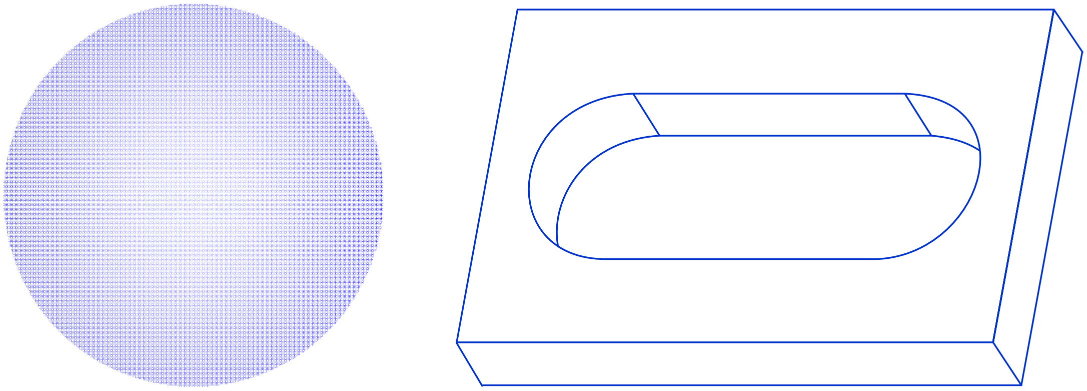


Figure 5. Parts created using solid modeling, a sphere on the right, and a block with a slot on the left.

The designs obtained from solid modeling are not unique since one design can be created using different procedures, which is reasonable since the Boolean operations possess the commutative property. Hence, there are nine procedures used to create solid models, starting with half spaces, boundary representation (B-rep), constructive solid geometry (CSG), sweeping, analytic solid modeling, cell decomposition, spatial enumeration, octree, encoding, and primitive instancing. The boundary representation is one of the most used procedures to generate solid models, which is one of the reasons that this research will use the boundary representation procedures to extract form features from CAD files.

2.2.3.1 Boundary Representation

The boundary representation also known in the literature as “B-rep,” is a way to define solids by their boundary based on topological entities such as faces, edges, and vertices (Ismail et al., 2005). The entities are bonded together which allow them to generate unique surfaces, where the inside or outside of the surface is recognized according to the position and orientation of its normal vector (Zeid, 1991). The main advantage of the boundary representation (B-rep) modeling is the fact that the solids are stored using geometrical and topological information, which allows the model to be tested as a physical object. The geometrical information of a solid model is obtained by applying Euclidian calculations in order to obtain surfaces, curves, and point coordinates, while the topological information is provided by the Euler law, defined as

$$F - E + V = 2 \tag{2.1}$$

where, F stands for faces, E stand for edges, and V stands for vertices. Equation (2.1) is the simplest representation of Euler’s law to describe a polyhedral; however, in the literature there are different variations of this law to represent complex solids (Zeid, 1991). The relation between topology and geometry is based on its entities, and it can be said that the topological entities define the geometrical entities, due to the fact that the faces provide information about the surface equation, the edges define the curve equation, and the vertices determine the location of the point coordinates.

Although, there are many ways to represent a solid, the boundary representation (B-rep) models are one of the most used in the literature and in the industrial world. According to Case (1993) extracting features from constructive solid geometry (CSG) models are not as straightforward as those created with boundary representation (B-rep) modeling due to the fact that the constructive solid geometry (CSG) models are not unique; this means that a part can be created in many different ways by using a finite combination of primitives, such as point, lines, circle, and spheres, among others. For this reason, the present research will use the boundary representation (B-rep) models as a part of the decision support system in order to obtain information about the part that has to be inspected. In addition, it can be seen in Section 2.4, called Feature Recognition, that the boundary representation models have been used to extract features from solid models and some of these studies have used the format of solid models that is proposed in the present research.

The solid modeling programs such as Pro-Engineering and Solidworks^{TM} to store solids as boundary representation (B-rep) models by using the ACIS file type. ACIS is a three dimensional modeler kernel that was created in the 1980's as an open source kernel which allowed it to be embedded in most of the CAD/CAM systems; however, ACIS was bought later on by Spatial Corporation a branch of Dassault Systemes (Corney et al., 2001; Spatial Corp., 2006). Neither the Spatial Corp. website nor the two bibliographical references found about ACIS, provided the meaning or origin of the name of the modeler ACIS, however, there are several websites online such as Wikipedia that mention that ACIS is an acronym of the name of the three people that developed the modeler "Alan, Charles, and Ian's System."

The CAD/CAM systems use the ACIS file as well as many other files to exchange information between different areas, such as design, manufacturing, and inspection, among others. Some files have different advantages, however, the purpose of these files is the same, which is to translate and exchange information between systems. A brief explanation about the multiple data exchange format is given in the following section.

2.3 Data Exchange Formats

The data exchange formats have the purpose of exchanging information between different areas of computer aided design (CAD) and computer aided manufacturing (CAM); as well as to translate geometric models from one representation to another, either CSG to B-rep, or B-rep to CSG, among others (Kroszynski et al., 1989; Zeid, 1991).

The exchange of information between CAD/CAM areas has to be performed in such a way that no information is lost between interfaces, and for this reason, the standards defined four types of data that need to be exchanged between systems, such as shape, non-shape, design, and manufacturing data. The shape data focus on the geometrical, topological, part, and form features; the non-shape data refers to the graphics data, as well as database information; the design data is the information used to perform mass property calculation, and finite element analysis; and the manufacturing data refers to the path planning, tolerances, process planning, and tool design (Zeid, 1991).

The ideal data exchange involves the four data types mentioned previously, however at the beginning of the development of the CAD/CAM areas there were few files capable of exchanging only geometrical information, such as Initial Graphics Exchange Specification (IGES) in the United States, ISO 10303 also known as Standard for the Exchange of Product model data (STEP), Standard for the Exchange of CAD data (SET) in France, and Organization of the Automobile Industry – Surface Translation Format (VDA-FS) in Germany (Kroszynski et al., 1989; Zeid, 1991; NIST, 2006).

According to Kroszynski et al. (1989) the national standards IGES, SET and VDA-FS have several limitations, which was one of the main reasons why the STEP file was chosen as an international data exchange format that has been developed by several countries such as the United States, France, Great Britain, Germany, Japan, the Netherlands, and Switzerland.

In the literature, authors had used different data exchange formats to extract features from design models, as well as from manufacturing parts, such as Li (1989) who used the IGES file to extract information of non rotational features from rotational parts; Prabhakar et al. (1992) used Romulus which is used as a CSG to B-rep converter, to extract features from boundary representation (B-rep) models using neural-networks techniques; Suliman et al. (2001) used the DFX file which is used in AutoCAD, to develop a recognition program to extract features from turned (rotational) parts using engineering drawings; and Case et al. (2000) proposed a system based on the design by features principle using the ACIS solid modeler; Ong et al. (2003) extracted STEP files from a design by features system using a developed feature recognizer; Ranjan et al. (2005) used DXF and IGES file formats to develop a recognition system that extracted machining features from a CAD model. Other data exchange formats include Gerber files, Parasolid, STL, among others.

2.4 Feature Recognition

Pattern recognition is a branch of machine learning that focuses on classifying input data by comparing it against predetermined rules and patterns stored in a knowledge database. In the literature, the pattern recognition field has been studied extensively, because it is a powerful tool that can be used in many applications, such as speech recognition, meteorology, medical, military, design and manufacturing, among others. The recognition process greatly depends on the part of the application in which the procedure is used. For example, the meteorology field focuses on extracting patterns to help predict storms, tornados, or hurricanes; the military field applies these techniques to decode radio transmissions in order to gather intelligence to capture the enemy (Alpaydin, 2004). In a similar way, the computer aided design (CAD) and the computer aided manufacturing (CAM) use file exchange programs that allow them to trade information, thus, once the part has been designed, the CAM software uses features recognition algorithms to extract all machinable features from the wireframe, surface or solid model file (Owusu-Ofori, 1990). Although, the exchange process between CAD and CAM looks like an attractive way to automate the production of machined parts, it has had serious problems in the later years, because of the existence of multiple CAD programs that define features in different ways than the CAM programs, causing trouble to the latter to identify the information from the CAD file (Li et al., 1989; Prabhakar et al., 1992; Case et al., 1993; Jha, 1995; Erard et al., 1996; Coulibaly et al., 1998; You et al., 1998). On the other hand, Maendl's (2003) study showed the CAM system successfully extracted all features when using data exchange files such as ACIS (*.sat), Parasolids (*.x_t), and Solidworks^{TM} (*.sldprt).

In the literature, the communication problem between CAD and CAM is being solved in two ways. The first one focuses on redefining the way in which the design is performed with the purpose of constraining the number of features that the designer can use to represent a part, also called design by features (Case, 1993), and the second is to standardize the data exchange files so that all CAD/CAM systems speak the same language and become able to recognize features from these files. The approach of designing by features allows the designer to create a part based on a finite list of predefined features, which simplify all processes after the design phase; however many complex shapes cannot be modeled since the list of features is limited. Therefore, this process is not yet robust enough to completely eradicate the feature recognition problems. On the other hand, the creation of a common language for the CAD/CAM systems has caused the authors to study the definition of the word "feature," which has had many different meanings according to the application in which it is used (Coulibaly, 1998). For example, Owusu-Ofori (1990) defined form features as the combination of solids created using a single machine cycle; Prabhakar et al. (1992) described feature as a function of topological variables and geometrical variables; and Jha (1995) classified the term feature as form features, precision features, and material features. Form features were defined as the shape of the part and its dimensions; the precision features were those attributes that determine the accuracy of the part such as tolerances and surface finish; and the material features that were defined as the material properties, types, surface treatments, etc. As can be seen, there are many definitions in the literature about the word "feature," which provide a difficult task for the feature recognition process, since they have had to adjust according to the types of features and the application in which they are used.

The accuracy of the feature recognition process plays an important role in the CAD/CAM systems, since it is the way in which both systems exchange information. The feature recognition process can be performed in many different ways, depending on the application in which are used. Prabhakar et al. (1992) classified the feature recognition process into human-assisted feature recognition and automatic feature recognition. The human-assisted feature recognition is when the user needs to help the system recognize the features by providing information about the designed part. The automatic recognition does not need any user intervention, and it is further classified into machining-region algorithms that focus on detecting those areas of the designed part that need to be machined. The ruled-based algorithms recognize features based on a series of rules that describe each feature of the part, while the graph-based algorithms use a dependency graph that shows the decomposition of the boundary representation model (B-rep) into faces, edges, and vertices that later on are decomposed into low level entities. Finally, the application-based algorithms are customized according to the application in which they are used. In a different study, Case (1993) identified two main approaches: the syntactic pattern recognition and the rule-based template approach to perform the feature recognition process. In the syntactic pattern recognition approach the features in the solid model are represented as a set of words that are compared against a set of defined rules that determine the type of feature and the class to which that feature belongs. The rule-based template approach is similar to the ruled-based algorithm defined by Prabahakar et al. (1992), where features are recognized by a set of predefined rules. Furthermore, Jha

(1995) divided the feature recognition systems into six groups called graph based, constructive solid geometry (CSG), volume decomposition, state transition and automata, syntactic pattern recognition, and expert system logic.

The decision support system developed in this research will focus on extracting the form features from the CAD file using the ACIS data exchange file, and it will recognize features and their dimensions defined in the word-base file as a way to perform the feature recognition process. A survey of studies that had been performed in the feature recognition field applied to the manufacturing area is presented in the next lines, with the purpose of showing the different ways in which authors had extracted features from drawings or CAD files. Li et al. (1989) created a feature recognition system to extract features from entities stored on an IGES file, which contained information from rotational parts that had non rotational features; Owusu-Ofori et al. (1990) developed a system to extract information from scanned engineering drawings of rotational parts. Prabhakar et al. (1992) extracted features from solid models using a five-layer perceptron algorithm. Jha (1995) developed a recognition system where he transformed the constructive solid geometry (CSG) representation into a destructive solid geometry (DSG) to determine the primitives contained in the designed model. Coulibaly et al. (1998) reviewed different feature recognition techniques used on CAD systems. Suliman et al. (2001) recognized entities of turning parts by comparing the information of 2-D drawings against a knowledge data base containing predefined entities. Ismail et al. (2005) proposed a system to recognize cylindrical and conical features based on a new technique called edge boundary classification (EBC).

Li et al. (1989) proposed a semi-automatic feature recognition system to extract non-turning features from rotational parts by using the 2-Dimensional engineering drawings exported in IGES file format. The system was created based on Li's (1988) study used a feature extraction system for rotational features, thus Li et al. (1989) extended the previous study, but with the purpose of including both turning and non-turning features. The author defined a rotational part as a symmetrical part that contains turning features and non-turning features. According to Li et al. (1989), the non-turning features are created by other non-turning features, which are classified as primary and secondary surfaces; where the primary surface is regarded as the tool entrance. Thus, he found that all non-turning features have one or more than one geometrical tolerances and surface roughness information.

The advantage of studying a part with only turning features was that a single view from CAD drawings was enough to extract all features from the part. However, this was not the case for non-turning features, since it was found that the system requires more pf the drawing's views to extract both types of features; thus the top view, front view, right hand side view, and the cross-section view were used in this research. The drawings were exported from a CAD modeling software using the IGES format, which was regarded as the input of the recognition system. The entities from the CAD model were classified as symmetric parts, non-symmetric parts, and text, which were categorized by using a part-definition data structure (PDDS) developed in Li's (1988) study, but modified to accommodate the non-turning features. The PDDS organized the extracted data into eight sections, such as general data, turning feature, turning surface, part geometry, non-turning feature, non-turning face, coordinate tolerance, and geometric tolerance. The system recognized three types of non-turning features, such as flats, holes (vertical, horizontal, and oblique), and slots (T-slot, square-slot, Y-slot, and dovetail-slot) that were extracted from the CAD drawings by analyzing their location based on the amount of vertical, horizontal, and oblique lines. However, the author found that there was no topological information that linked the lines of one view to the lines of a different view; therefore, the author analyzed each individual view and ordered the features according to the views and to their position within the view. The system performance was evaluated using a program developed in AutoCAD, where an IGES file that contained non-turning features was introduced to the system. It was found that the system successfully recognized flat features, holes, and slot features. The tolerances and the information from the surface roughness were also recognized, however, this information was gathered from the information provided by the user and not from the IGES files, which revealed the lack of robustness of the data exchange files.

Owusu-Ofori et al. (1990) developed a feature recognition algorithm to extract manufacturing features of turned (rotational) parts from scans of engineering drawings. The recognition model was divided into three levels: the low level, which had the purpose of recognizing geometric elements, such as lines and arcs; the medium level, which identified the shapes; and the high level that recognized the features based on the information gathered by the two preceding levels. In addition to these levels of recognition the authors added a knowledge database that contained information about dimensional, bounding, and manufacturing related attributes, which helped the system to increase the recognition accuracy by adding redundancy. The lines and arcs captured by the low level were supplied to the shape recognition or medium level that used a set of rules and the knowledge database to extract the shapes of the symmetric part. This level was regarded as a critical part of the system since any shape that was improperly recognized affected directly the feature recognition level. The system was able to detect twenty-four form features from turned (rotational) parts that were classified as grooves, transitions, recess-features, semi-open features, and others (open-diameter, chamfer, taper, and extrusion features). The feature classification was based on the parameters that affect the creation of turned (rotational) parts. The authors found that the form features of turned (rotational) parts were affected by five factors:

- Form of cutting edge tool.
- Tool motion (axial, radial, and combined).
- Tool motion constraints.
- Number of passes needed to machine the feature.
- Machining technology constraints.

The algorithm behind the shape and feature recognition was based on the set theory, where lines and arcs were combined to recreate the form features. The system was capable of recognizing single and multi-feature turned (rotational) parts following the mentioned logic, where the features were detected in order starting with the one located on the left side of the part.

Prabhakar et al. (1992) proposed an automatic recognition system using neural-network techniques to extract features from boundary representation (B-rep) models. The feature recognition system was modeled using a multilayer perceptron network, which used the threshold, transfer function, and weights to detect features from the boundary representation (B-rep) model. The solid model was exported from the ROMULUS solid modeler using a format file called PROLOG. The topological and geometrical information obtained from the file format was used as the input of the model; however, this information was preprocessed and transformed into a numerical matrix called the adjacency matrix, since it was simpler for the neural network algorithm to perform numerical operations instead of logical operations. The adjacency matrix was defined as a 2D array of integer vectors, where each of these vectors was composed of eight integers that held information of the feature entities (faces, edges, etc.) and the relationship between them. The rows of the adjacency matrix were composed of several vectors that described the characteristics and properties of a specific face. The rows were provided to the network one-by-one and were decomposed into vectors that at the same time were decomposed into integers. Thus, the network defined as a five-layer perceptron processed the integer values to generate the code that defined the feature hidden in each row. A feature library was created and trained with a specific network of features to be recognized, which were analyzed using the range nonlinearity developed by Prabhakar et al. (1992) since the standard nonlinearities were continuous functions that were not suitable for this application. The algorithm was tested using three complex parts that contained a through-hole, through-slots, and intersecting slots. Results from these tests showed that the system is able to identify features for which it has been trained, thus the accuracy of the system depended on the training of the network. Also the system showed difficulties to accurately identify all features that were intersecting due to the lack of information between the feature faces.

Jha (1995) developed a feature recognition system using as an input a constructive solid geometry (CSG) model and extracting the features using destructive solid geometry

(DSG). The author proposed that a good feature recognition system should be able to display the recognized features on the interface and to allow the user to provide input in order to make the system more flexible. Also the system should create one database that can recognize CAD features, and that can cope with the deficiencies of the solid modelers' databases, since these cannot store information about surface finish, tolerances, and material properties, among others. The proposed system received as input the constructive solid geometry (CSG) model that was created in I-DEAS and exported using a Universal File Format called UFF. The system used the Boolean operations performed in the constructive solid geometry (CSG) model to work backwards in order to extract the primitives. The extraction process was divided into four steps that were executed using eight software modules. The first step was to simplify the Boolean operations by expanding the terms and removing the parenthesis, which was performed by the first and second modules. The second step was to define the minimum envelope that enclosed each feature, which was performed using third, fourth and fifth modules. The third step was performed by the sixth module, which defined the different primitives used to remove the excess of material. Finally, the fourth step, performed in the seventh and eight modules, transformed the union primitives by the inverse and exported the output as an I-DEAS file. Although the feature recognition process was explained in the article, there was no information regarding any experimental procedure. The author suggested applying the system on parts created in a turning process, as well as on parts that are intersecting since they are common on constructive solid geometry (CSG) models.

Coulibaly et al. (1998) reviewed the feature recognition process for conventional and feature-based CAD systems. For conventional CAD systems, the authors found that the parts were defined as one entity instead of using features to define the part, such as the constructive solid geometry (CSG), which has the disadvantage that the models are not unique since they can be created using Boolean operations in different order, and the boundary representation (B-rep) approach that recognized features using topological information of the faces to determine the relationship between all shapes. For feature-based CAD systems, the system has complete information of every feature contained in the model simplifying the feature recognition process. The authors found two recognition methods for these types of systems, the recognition using the entity/relationship model in which the part was broke up into all features in order to determine the relationship between them, and the recognition using graph-grammar approach where the system goal was to extract words from either SET or STEP files in order to get the feature's characteristics and its relations. The graph grammar approach had a knowledge base with a list of features and their corresponding topological graphs, this was used to compare the words obtained from the STEP file and to determine the amount and type of features contained in the model. This type of recognition process is similar to the process proposed in the present research, which will be explained in the following chapters.

Suliman et al. (2001) developed a system to automatically recognize features of turned (rotational) parts using engineering drawings. The system was created using three main modules: structured modeling, feature recognition, and feature sequencing for machining. The structured modeling module aim was to extract all entities from 2D AutoCAD drawings that were exported using the DFX file. The drawings of the turned (rotational) parts were extracted as orthographic projections where the hidden features were described with dotted lines and the edge of the part with continuous lines. A program written in C language was used to extract information such as entity type, style, starting point, end point, and center point of an arc from the entities. The program had three stages that were used to recreate features based on the position of the lines in the drawing. The drawing of the part was symmetrical with respect to the center axis, therefore, on the first stage, all lines below the middle line were removed. In the second stage, some of the vertical continuous lines were removed according to the intersection between the continuous and dotted lines. The empty spaces left after removing the lines represented the beginning or ending of the internal features. Finally, in the third stage, the entities were arranged in a list where the external entities were ordered from right to left, while the internal entities were arranged first from left to right, or from right to left, according to the inaccessibility issues. The feature extraction module used the information from the previous module, where the recognition was performed automatically by comparing the AutoCAD information against a knowledge database that contained eight turning features, such as cylinders, end face, shoulder face, taper, chamfer, fillet, corner, and undercut turn. Each of these entities was represented by a major entity that held entity properties, related to style, type, convexity, peripherality, *x* and *y* coordinates' changes, threshold relativity, and tool accessibility. The major entity information was carried on to the final module, where the extracted features were sequenced for machining using a program written in C^{++}. The program used three algorithms to determine the rough and final turning operations as well as the tool motions that were sequenced according to the location of the internal and external features.

Ismail et al. (2005) proposed a feature recognition system developed by using a new technique called edge boundary classification (EBC), which was based on the spatial addressability property of the boundary representation (B-rep) models. The system extracted only cylindrical and truncated cone features, since the author found that they were the most used features in mechanical parts. The cylinders were defined in three ways: the single circular edge loop (SEL), in the case where the face of the cylinder was formed by the intersection between a cylinder and a normal plane. The multiple curve edge loop (MEL), in the case that the face of the cylinder was formed by the intersection of the cylinder and more than one tilted planes, or as semi-cylindrical faces, such as the case of a slot, where half of the cylindrical faces are located at the extremes of the slot. Hence, the system was able to recognize cylindrical features such as, through holes, blind holes, boss and internal undercuts. For conical shapes, the system only recognized the features of a truncated cone, such as chamfers and tapers, which were the ones that had two edge loops, where one of the loops was smaller than the other one. The system used the edge boundary classification (EBC) technique to define patterns of test points (t_{p1}, t_{p2}, and t_{pm}) for feature classification. The test points were created by membership classification that was defined as the process where points and lines are classified according to the location of the test points in, on, or off the solid. The test points were created by positioning two points on the edge of the solid face, and then calculating the middle point between them. Thus, the location of these three points with respect to the object determined the type of feature contained in the part. For example, a through hole was defined by the three points being outside the solid, while a boss was defined by three combinations. The first one was defined by t_{p1}, t_{p2} on the solid and , t_{pm} in the solid, the second one by , t_{p1} on the solid, and t_{p2} , t_{pm} in the solid, and the third one by t_{p1} , t_{pm} being in the solid, and t_{p2} being on the solid. The feature recognition algorithm was evaluated using a theoretical part in AutoCAD, where the proposed system was able to recognize all cylindrical features as well as other features that were not recognized in previous studies, such as a chamfer and a blind pocket. The reason that others did not recognize these features was that the later did not consider conical features in their knowledge base. Therefore, the above research agrees with the research proposed in the present study, where the use of boundary representation models for feature extraction is a good approach due to all properties and characteristics that are stored in these models.

2.5 Inspection Technologies

The application of the computers in the design and manufacturing areas have helped the development of products with tight tolerances and complex forms, which represent a big challenge to the inspection area, because the inspection equipment has to be designed with higher inspection capabilities in order to cope with the quality standards. There are many types of inspection instruments that range from simple gauges and calipers, to more sophisticated equipment such as profilometers, coordinate measuring machines (CMM's), and laser equipment. Each one of these instruments has the purpose of measuring different types of tolerances. Although these instruments are developed with the purpose of providing accurate measurements, they are susceptible to different types of environmental conditions that affect the performance and accuracy of their readings. Barry (1978) identified two types of errors that apply to any inspection instrument and occur while taking measurements: the first type called accidental or random errors, and the second type called systematic or constant errors. The accidental errors were defined as the random noise that affects the measurements, which can occur at any given time, and has no correlation between the magnitude and the sign of the previous or subsequent error. The systematic errors were defined as those errors that always induce noise into the readings with the same magnitude. These errors were subdivided into three categories called natural errors, instrumental errors, and personal errors. The natural errors were described as those inaccuracies caused by natural phenomena, such as thermal expansion, humidity, and atmospheric pressure. The instrumental errors were described as those errors caused by the instrument itself, which were related to the poor quality of the materials used to create the equipment, or related to the squareness of the axes, among other machine problems. The last type of systematic errors were the personal errors that were attributed to the person that is performing the measurements; some examples of this type of errors were identified as poor vision, slow reaction time to activate or halt the count of a stopwatch, among others.

In addition to the above errors, there are many other errors that affect the measurements and the analysis of these measurements. Hocken et al. (1993) identified three main factors that induce noise into the inspection process. The first is related to the translation of the information provided by the designers to the inspection process, which is similar to the CAD/CAM problem that was stated above in the feature recognition section. The next is the effect caused by the inspection equipment and sampling strategy into the measurement results. The noise introduced by the mathematical algorithms used to process and analyze the results is the third factor of noise induction. For this reason, researchers had focused in creating reliable inspection equipment able to measure and analyze data using powerful mathematical tools that counteract and recognize the effects of these errors.

In coordinate metrology, the inspection equipment is used to determine the tolerances of manufactured parts by collecting a predetermined amount of point coordinates. The collected sample is processed using a fitting algorithm that will determine the parameters and the form error of the part. The verification process is not very complex, but it takes a significant amount of time since it is necessary to have a controlled environment to reduce the systematic errors that could affect the results of the process. For this reason, several inspection technologies have been developed, which collect point coordinates under different circumstances in a better and faster rate.

Curless (1997) proposed a way to classify the inspection equipment based on the technology used to acquire information from the part, shown in Figure 6. The devices were classified as contact and non-contact technologies, where the contact technologies relied on probes and styluses to collect information from the part in a discrete way, while the non-contact technologies used lasers, electromagnetic waves, microwaves, and light to acquire the information. Contact technologies were found to be more common since they are cheaper and can achieve good levels of accuracy that makes them appropriate for many manufacturing shops; however, one problem that was encountered was that the data collection required a significant amount of time, since the equipment has to travel around the object in a collision free path. On the other hand, non-contact technologies were more versatile since they can acquire millions of data points in just a few seconds, but the cost of the equipment is much greater compared to those prices for the equipment that uses contact technologies.

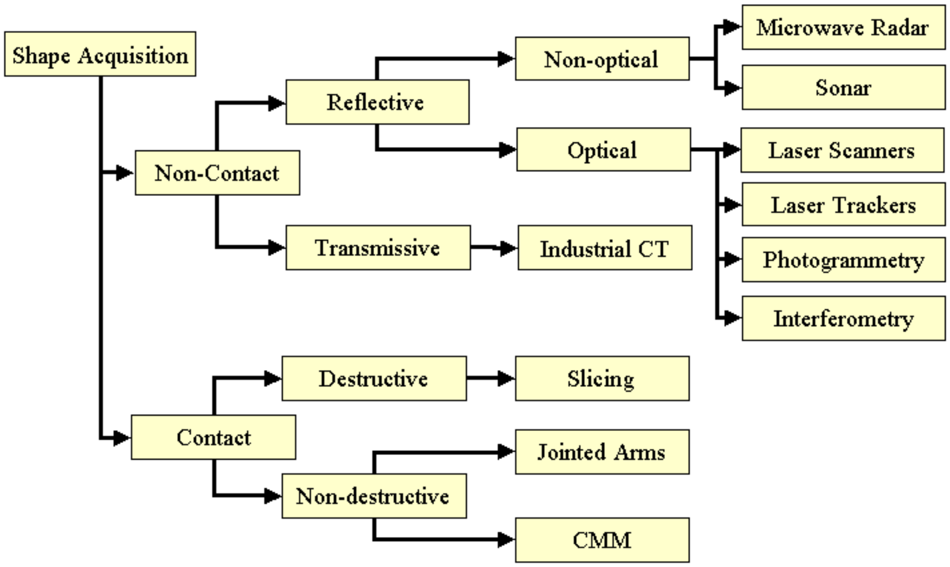


Figure 6. Taxonomy of the shape acquisition. (Adapted from Curless, 1997)

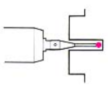
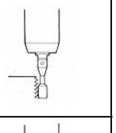
2.5.1 Coordinate Measuring Machine (CMM)

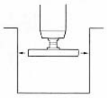
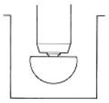
The coordinate measuring machines (CMM) are used to verify the dimensions and tolerances of manufactured parts in industries such as aerospace, automotive, electronics, and plastics (Dotson et al., 2003). One of the main reasons that the coordinate measuring machine is highly utilized is the fact that it is economically affordable and provides many other benefits than the traditional micrometers and gauges. However, there are other technologies such as lasers and interferometers that are more accurate, but the acquisition cost is normally high. For this reason, the CMM technology was chosen in this research as the main way to collect information from the machined parts. In addition, any new improvement and contribution found in this research will have an important impact in the research and industry world.

Groover (2001) described the CMM as an inspection equipment that is composed of four basic elements:

1. The *touch-trigger probe* that records point coordinates from the inspected part by measuring the displacements of a stylus. The probe is composed of a set of rollers and springs that are electronically connected to the machine controller, and are activated by the motion of the stylus (Anthony, 1986). The stylus is held by the probe head and physically touches the inspected part for collecting the measurements. Thus, the rollers and springs send an electronic signal to the machine controller that will decode and transform it into point coordinates each time that the stylus touches the inspected part, seen in Figure 13. The stylus can have many shapes, such as balls, cylinders, end pointers, and discs, and it can be found in different materials, such as ruby, steel, silicon nitride, zirconia, alumina, and tungsten carbide, which can vary according to the type of measurements and the environment in which the measurements are performed, as seen in Table 1; (Renishaw plc., 2005).

Table 1. Stylus tip types and properties. (Adapted from Renishaw plc., 2005)

Stylus Tip	**Deviations from spherical, cylindrical, or pointer forms**	**Coefficient of expansion at 25°C**	**Stylus Picture**
Ruby balls	5.12 µinch (0.13 µm)	4.5x10^{-6}/°C	
Silicon nitrate balls	5.12 µinch (0.13 µm)	3.2 x10^{-6}/°C	
Zirconia balls	5.12 µinch (0.13 µm)	10.5 x10^{-6}/°C	
Ruby ball ended cylinder	0.0157 µinch (4 µm)	4.5x10^{-6}/°C	
Silver Steel simple cylinder	0.0157 µinch (4 µm)	16 x10^{-6}/°C	
Tungsten carbide ended cylinder	+1575 µinch (+40 µm)	5 x10^{-6}/°C	
Silver Steel simple pointer	Cone angle 30°	16 x10^{-6}/°C	
Tungsten carbide radius pointer	Cone angle 30°	5 x10^{-6}/°C	

Silver Steel discs	0.39 µinch (1 µm)	16 x10^{-6}/°C	
Alumina hollow balls	0.39 µinch (1 µm)	8.1 x10^{-6}/°C	

2. A *set of servomotors* work together to activate the bearings that position the three perpendicular axes of the machine (Dotson et al., 2003). Smith (2002) identified five types of bearings that are commonly used in inspection equipment, such as dry bearings, ball bearings, hydrodynamic bearings, air-based hydrostatic bearings, and oil-based hydrostatic bearings. Out of the five types of bearings, the air bearings are the most widely used bearing in coordinate measuring machines. The author found that this type of bearing can provide high levels of accuracy and generates the least amount of friction compared to the other types of bearings. However, it was found that the humidity and the uncleanness of the compressed air could decrease the accuracies or even damage the equipment.

3. A *computer system* controls the machine during the numerical control and manual operations.

4. A *software tool* is used to operate the inspection machine and to analyze the collected data. In general, the software contains a set of subroutines that allow the user to verify and analyze the most common tolerances found in machined parts, with the purpose of reducing the time of inspection (Dotson et al., 2003). The last two components defined by Groover (2001) are dependent on one another, since one will define the requirements of the other. Computational systems are easier to upgrade than the software tools, thus in most of the cases the software tool will define the amount of computer resources. Some of the most common software

tools used in CMM equipment are PCDMIS^{TM}, Quindos^{TM}, Metrolog XG^{TM}, Metrosoft CM^{TM}, among others.

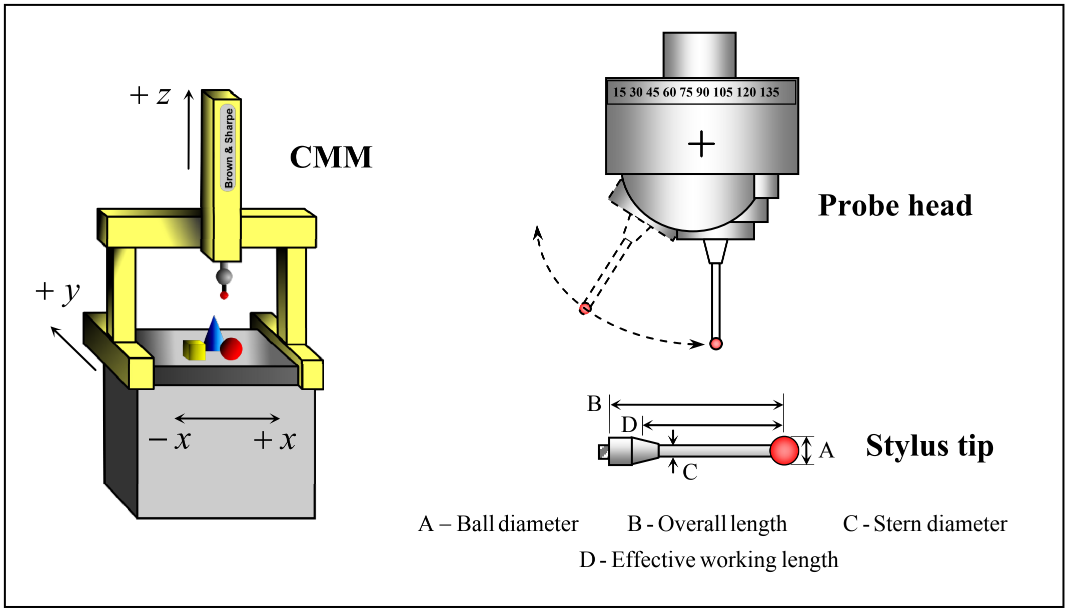


Figure 7. Coordinate Measuring Machine (CMM), probe head, and stylus tip.

The coordinate measuring machines (CMM) are operated in four different modes: manual, manual computer assisted, motorized computer assisted, and direct computer controlled (Dotson et al., 2003). The manual mode is the simplest of all the modes where the user measures data points by manually moving the probe to any desired location. The manual computer assisted mode is very similar to the manual mode, and the only difference is that in this mode the user is aided by a computer interface that provides information about the location of the probe and analysis of the data points. The motorized computer assisted mode allows the user to position the CMM's probe in all three directions by using a joystick. The direct computer controlled mode is the most accurate since it allows the creation of a program that manipulates the movements of the CMM by setting the positioning and measuring speeds, the approaching vector, and the point coordinates of the measuring points.

Independently of the operation modes, the CMM can be found in the different configurations that vary according to the work envelope and the way in which the three axes are arranged. The five configurations are column, bridge, cantilever, horizontal arm, and gantry types (Dotson et al., 2003). The column type has a similar configuration as a computer numerical control milling machine, where the *x* and *y* axes are located on the measuring table, and the *z* axis lowers and raises the probe arm. The bridge type is one of the most commonly used types of CMM and is the same type of CMM that is being used in the present research. The bridge carries all three axes, moves back and forth in the *y* axis, left to right in the *x* axis, and it lowers and raises the probe in the *z* axis. The cantilever type is similar to the column type, where the *x* axis is located in the measuring table, the *y* axis moves the cantilever arm, and the *z* axis is located in the cantilever arm that carries the probe. The horizontal arm type is similar to a robotic arm and is located on one side of the measuring table, where the *x* axis moves the arm from left to right, the *y* axis extends and retracts the arm, and the *z* axis lowers and raises the arm. Finally, the gantry type is similar to the bridge type, however, in this type of CMM the bridge is supported by two arcs that define the work envelope.

The concepts of accuracy and precision play an important role in the inspection process, since any reliable inspection requires an inspection instrument, in this case the coordinate measuring machine (CMM), to provide the same results within readings and with the smallest variability between them (Montogomery, 2001). However, studies performed by Hocken et al. (1993) showed that the mentioned CMM components are susceptible to external factors such as parametric, probe, thermal, and general inspection errors that induce noise into the readings. The parametric errors were defined as those errors that are induced by the CMM components each time the probe needs to be physically relocated. Although the errors seem to be unavoidable, the author found that these errors behave as systematic errors, thus they can be reduced by the use of an accurate sampling strategy. The probe errors are those errors that are induced each time the stylus tip is in contact with the inspected part. The errors are due in great part to the approach vector, approaching speed, contact force, and stylus configuration. However, suppliers of coordinate measuring machines, DEA (1995a, 1995b), and stylus tip, Renishaw plc. (2005) suggested procedures and guidelines to reduce the effect of these types of errors. DEA (1995a, 1995b) recommended that the stylus tip must be qualified and compensated before any measurement is taken. The tip qualification consists of measuring five points, which are used to calculate the geometrical center of the ruby ball, and to estimate the tip diameter from a calibration sphere. The tip compensation process requires the measurement of five points from the calibration sphere, which helps the CMM recognize the approaching vector of measurements and compensate for the probe radius. In addition, it was recommended that all measurements must be taken using the normal vector of the surface in order to increase the accuracy of the measurements (DEA, 1995a). On the other hand, Renishaw plc. (2005) suggested that the stylus must be as short as possible because the length of the stylus will affect the accuracy of the measurement. Moreover, it was recommended that the ball diameter must be as big as possible in order to reduce the possibility that the stern touches the measuring object before the stylus ball does.

Kanada (1997) studied the number of roundness measurements needed to estimate the sphericity error. He used a stylus tip to record the roundness measurements where he found that the accuracy of the measurements increased when performing the compensation procedure. In addition, studies done by Hocken et al. (1993) on the effect of probe approach rate, stylus length, and stylus orientation in geometric tolerances, showed that the probe errors are sometimes the magnitude of the desired tolerances. For this reason, he found that it is extremely important to perform the tip compensation and probe qualification before taking any measurement.

The thermal errors are very critical for the inspection process since studies done by Hocken et al. (1993) reported that these types of errors can have the same magnitude as the errors induced by the imperfections of the machine components, such as errors due to axes squareness, or straightness errors, among others. Moreover, it was found that these errors are not normally distributed, making it a difficult task to find a sampling strategy that can counteract their effect. The sources of these errors are room temperature, lights, number of people working close to the inspection machine, among others. Finally, the last type of errors is the inspection errors that are related to all the set up, inspection and data analysis procedures that induce noise to the measurements. Some examples of this type of errors include the improper alignment between the origin of the CMM and the geometrical origin of the part, accuracy of the fitting algorithms, stylus properties, uncontaminated part surface and measuring table, among others.

2.6 Geometrical Tolerances

The tolerances are used to describe the geometrical imperfections between the machined part and the ideal geometry developed during the design phase (CAD) (Dotson et al. 2003). Thus, the correct inspection of these tolerances will determine the accuracy of the part. The five types of geometrical tolerances are form, profile, orientation, location, and runout (Kalpakjian, 2001). The tolerance of form determines the deviations of the inspected part regarding how straight, flat, round, and cylindrical the part is. The profile tolerances specify the level of error of the profile of a line or surface. The form and profile tolerances determine the error of the machined parts based only on the specific curve of the surface being measured. On the other hand, the calculation of the tolerances of orientation, location, and runout, require the measured data to be compared against theoretical references called datums, in order to determine the level of error of the part, Henzold (1995). The orientation tolerances verify the deviation between the datum and a measured plane in order to determine if both are perpendicular, parallel, or tilted at a specific angle. The location and runout tolerances verify the differences between the datum and the measured features, in order to determine if both are at the same position and concentric or if they have a circular and total runout, respectively; as seen in Table 2 Geometrical Tolerances.

Table 2. Geometrical Tolerances. (Adapted from Kalpakjian, 2001)

Tolerance Type	Tolerance	Symbol
Form	Straightness	⏤
	Flatness	⏥
	Roundness	○
	Cylindricity	⌭
Profile	Profile of a line	⌒
	Profile of a surface	⌓
Orientation	Perpendicularity	⊥

	Angularity	∠
	Parallelism	//
Location	Position	⌖
	Concentricity	◎
Runout	Circular runout	↗
	Total runout	⌰

The national and international standards ANSI and ISO recognize the mentioned tolerances. However, the standards do not specify the procedures needed to evaluate the tolerances, which can lead to interpretation problems. Consequently, several researchers have focused on determining the algorithms that best determine the tolerances of machined parts, especially the form tolerances. Forms such as spherical, conical, conical frustum, torus, and prismical have not been defined by the standards yet. Therefore, it is important to focus on these complex shapes in order to find the best inspection parameters that yield the true form or error. Among these parameters are the sample size (geometry based, adaptive size), sampling strategy (Hammersley, Halton-Zaremba, random, aligned systematic, uniform, adaptive sampling), and fitting algorithm (linear or nonlinear), which are explained in the subsequent sections.

2.6.1 Circular and Spherical Form

The study of form verification of circular and spherical shapes is a problem that was created due to the needs of producing quality parts in the manufacturing environment. However, some of the principles that govern the algorithms that nowadays are used to verify circular and spherical shapes were studied during the 1800's and in the beginning of the 1900's. Clear good examples are the papers from Sylvester (1857) who identified the importance of finding the smallest circular shape to enclose a finite number of points, and Blumenthal (1941), who described a way to find a minimum circumscribed spherical surface that can bound a subset of n-dimensional Euclidian space, in which he referred to many other articles that were developed during the early 1800's. Unfortunately, the references from Blumenthal's (1941) paper were in German and French, making them difficult to read and almost impossible to obtain, due to the times when these papers were published.

One hundreds years after Sylvester's (1857) publication, Rademacher et al. (1957) developed an interesting algorithm to enclose *n*-points inside the smallest circumscribed circle. The procedure to compute the circle radius required the calculation of the distances between all points. Then after all distances were calculated, the "span" of the set of points was determined by choosing two points (P_1, P_2) that have the maximum distance between them. The maximum distance (d) was used to create two circles: one with radius d and origin in P_1 and another that had a radius d and origin in P_2. The intersection between these two circles created an area similar to the shape of a football where the rest of the points were positioned. The football shaped area had two points S_1 and S_2 that were the points where the two circles intersected. The distance between the two points represented the smallest diameter of a circle needed to enclose the *n*-points, shown in Figure 8.

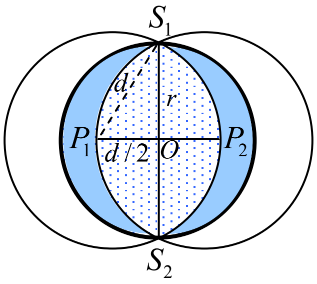


Figure 8. *n*-points enclosed by smallest circle (Adapted from Rademacher et al. 1957)

Thus, the radius of the enclosing circle was obtained by applying the Pythagorean Theorem to the triangle created by P_1, $S_1$and O,

$$r = \sqrt{d^2 - \left(\frac{d}{2}\right)^2} = \frac{d}{2}\sqrt{3} \approx 0.866\ d \tag{2.2}$$

Despite the results obtained, Rademacher (1957) proved under different scenarios that there was a smaller circle that can enclose the given set of points, which was found by applying H.W.E. Jung's theorem that stated that "*every set of diameters* d *in a Euclidian space* $\mathrm{E_n}$ *is contained in a ball in* $\mathrm{E_n}$ *of radius* $r = d\sqrt{(n/2(n+1))}$," therefore, the smallest circle that can enclose *n* given points was defined by the radius of $r = \frac{d}{3}\sqrt{3} \approx 0.577\ d$ (SpringerLink, 2006).

Similar to Sylvester (1857) and Rademacher et al. (1957), Elzinga et al. (1972) revisited the problem to find the smallest spherical shape that enclosed a finite number of points, where he formulated the problem as a quadratic programming dual problem and solved using the Simplex method. Although none of these studies were applied in the metrology area, it is important to mention that they provided an insight of the importance that the circular and spherical shapes have in the mathematical world, since they are very useful shapes that can be used to solve many different types of problems, ranging from the pure mathematical problems to the more applied problems such as the form verification of circular and spherical shapes.

The spherical surfaces are one of the most studied 3-Dimensional shapes in the literature due to their vast number of applications in the metrology area and to their simple mathematical models that define them. Several authors have published multiple articles on form verification of spheres, such as Kanada (1995, 1997) and Samuel et al. (2000, 2003). They had revisited the topic providing improvements to the existing models as well as exploring new fitting algorithms for the calculation of the parameters of the sphere. These articles and many others are explained in the following paragraphs and are arranged in a chronological order, starting first with those used to verify the form of circular shapes and then reviewing those related to form verification of spherical shapes.

2.6.1.1 Roundness (Circularity)

The term roundness or circularity defines the degree in which measured parts deviate from the true form of a circle. Chetwynd (1979) studied the properties and applications of the Limacon approximation for roundness measurements. The Limacon is as a mathematical shape used to linearly approximate the parameters of the circle with respect to the origin. The procedure also called "radius suppression," is applied to counteract the origin offsets between the coordinate system of the measuring instrument and the geometrical coordinate system of the round part that is being inspected.

Figure 9 provides a graphic explanation of the logic behind the Limacon approximation where R is the radial distance measured from the surface of the round part to the geometrical origin of the part O_c; and ρ is the radial distance measured from the surface of the round part to the origin of the measuring instrument O_s. The geometrical origin of the part O_c is located in polar coordinates at a distance e at an angle of ϕ degrees, or in Cartesian coordinate at a distance of a in the x axis and b in the y axis, or in polar coordinates.

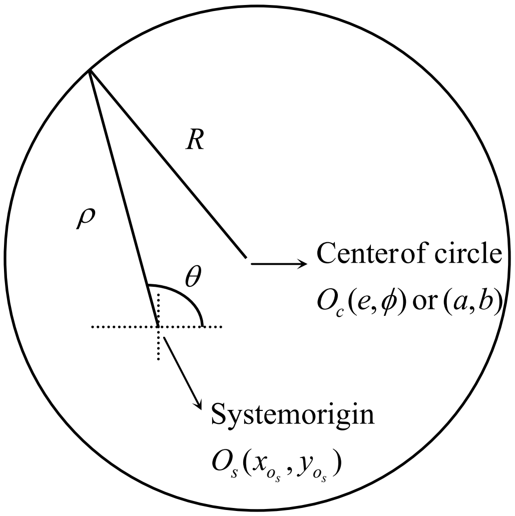


Figure 9. The Limacon approximation (Adapted from Chetwynd, 1979)

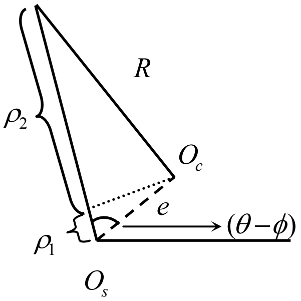


Figure 10. Limacon approximation explanation

Mathematically, the main idea of the Limacon approximation is to linearly calculate the value of ρ in terms of the radii R and the origin offsets (x_0, y_0). By doing this the measurements of a round part can be broken-down into these three factors in order to find real deviations of a circular part. Therefore, from the Figure 10 and Equation (2.3), the value of ρ can be calculated by adding the values of ρ_1 and ρ_2, where ρ_1 is obtained by using Equation (2.4) and ρ_2 by using the Pythagorean theorem, as described in Equation (2.5).

$$\rho = \rho_1 + \rho_2 \tag{2.3}$$

$$\rho_1 = e\cos(\theta - \phi) \tag{2.4}$$

$$\rho_2 = \left[R^2 - e^2 \sin^2(\theta - \phi)\right]^{1/2} \tag{2.5}$$

Equation (2.5) can be reduced to only R since the value of e is considered very small compared to the value of R, thus making the right side of the subtraction almost zero. In a similar way, Equation (2.4) is extended by substituting the value of cos(θ - ϕ) for the trigonometric identity of the sum of two angles cos(ϕ)cos(θ)+sin(ϕ)sin(θ). Therefore, the value of ρ is obtained by adding R to ρ_1, as seen in Equation (2.6).

$$\rho \cong e\cos(\phi)\cos(\theta) + e\sin(\phi)\sin(\theta) + R \tag{2.6}$$

The value of a and b that represent the location of the geometrical origin of the part can be defined as a = e cos (ϕ) and b = e sin (ϕ). Thus, if these trigonometric relations are substituted in Equation (2.7) then the value of ρ is given by

$$\rho \cong a\cos(\theta) + b\sin(\theta) + R \tag{2.7}$$

Finally, instead if a and b are called x_0 and y_0, then the equation describing the value of ρ can be used to verify mathematically that the form of a round manufactured part was created according to specifications. Therefore, the major contribution of Chetwynd (1979) is that round parts can be linearly verified since the origin offsets are no longer affecting the calculations of the parameters of the circle.

The advantages of incorporating the Limacon approximation to a 2-Dimensional or 3-Dimensional form evaluation model are that the model will compensate the origin offset between the measuring instrument and the measuring part, therefore, the model will provide information of the origin offset in x and y components and the true radial measurements of the inspected part. Furthermore, the Limacon approximation can be applied in a real manufacturing environment especially in the inspection process because it will allow the operator to decrease the time that he/she spends aligning the part to the instrument's coordinate system, thus reducing the cycletime of the process. In addition, the Limacon provides good results when the approximation is solved by means of the least squares method, due to its linear properties (Chetwynd, 1979).

2.6.1.2 Sphericity

The sphere is a 3-Dimensional surface in Euclidian space $\Re^3$ that is defined as a set of points that are equidistant from a center point called the origin (Weisstein, 1999). Figure 11 illustrates the two parameters that describe the form of the sphere, which are the origin of the sphere O (x_0, y_0, z_0), and the radius (R) which is the distance from the surface origin to a point P (x, y, z) contained on the spherical surface. Usually, the location of the origin is at O (0, 0, 0), however this location can change according to the application in which the spherical surface is being used. Equation (2.8) shows the relation between the radius of the sphere and the Cartesian coordinates (x, y, z).

The spherical forms can be found as closed surface such as Figure 1, or as partial surface such as in the case of the hemisphere. These surfaces are very useful shapes since they can be easily adapted to solve difficult geometrical problems, such as the calculation of the form deviation of cones, frustums, and torus forms, as shown in the subsequent chapters.

$$R^2 = \sqrt{x^2 + y^2 + z^2} \tag{2.8}$$

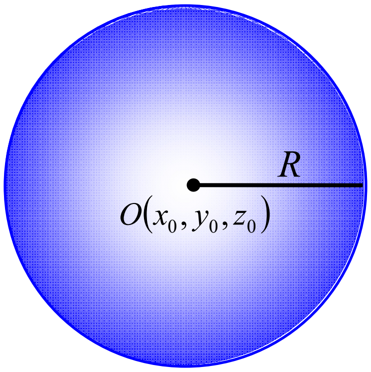


Figure 11. Parameters of the Sphere

The form evaluation of spherical shapes, also called sphericity, is defined as how much the form of an object is different from an ideal sphere. The sphericity error was determined based on sets of 2-Dimensional round planes. (Kanada, 1995, 1997; Huong, 1999; and Samuel et al., 2001), instead of measuring the sphere as a 3-Dimensional object; all of this due to the lack of procedures from ANSI and ISO to evaluate the tolerances of complex forms. For this reason, the following lines present a collection of previous studies that attempted to establish a 3-Dimensional model to evaluate the form tolerances of spherical forms.

The mathematical models that describe the spherical shape are essentially two, which are also the most frequently used in the literature. The first model is a linear model that uses the Limacon approximation as a reference to find the origin offsets (x_0, y_0, z_0)

and radii of the sphere (*R*). This research refers to the model as a linear spherical model; as seen in Equation (2.9),

$$d_i = r_i - \left[R + x_0 \cos\theta \sin\alpha + y_0 \sin\theta \sin\alpha + z_0 \cos\alpha\right] \tag{2.9}$$

The second model is a nonlinear model, which applies the Euclidian distance principle to find the parameters of the sphere, and is referred to through out this research as a nonlinear spherical model, shown in Equation (2.10).

$$d_i = R - \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2} \tag{2.10}$$

Although, one model is linear and the other is nonlinear, both aim to find the radii of the sphere and the origin offsets. Each of these models has advantages and disadvantages, which are explained in detail in the following chapters. Meanwhile, the next paragraphs show a summary of the spherical papers available in the literature, in which most of them apply the mentioned linear and nonlinear models. Thus for simplicity, the models will be referred to by name in the text and not in the equation form.

The sphericity evaluation is a 3-Dimensional measurement that requires accurate and precise instruments to collect data from the surface part, such as the Coordinate Measuring Machines (CMM). However, the CMM's are measuring instruments that were not developed until the late 1990's. Prior to this time, the sphericity evaluation was performed by using a combination of 2-Dimensional procedures such as sets of roundness or circular measurements where each of these measurements were located at 90 degrees between each of them, thus calculating the sphericity value by using the minimum circumscribed circle method (Kanada, 1995). The main reason that the spheres were evaluated as a combination of roundness measurements was that ANSI and ISO standards showed that the circle and the sphere share similar geometrical properties. However, authors such as Kanada (1995, 1997), Huong (1999), and Samuel et al. (2001) stated that the 2-Dimensional measurement of a sphere was not accurate since the spherical form is by nature a 3-Dimensional shape, and repetitive 90 degrees apart roundness measurements did not provide a good representation of the sphere error. Kanada (1997) used a statistical method to verify the form of spherical parts where his main purposes were to determine the efficiency of the measuring method proposed by the ISO and ANSI standards and the minimum number of roundness measurements needed to obtain a good representation of the sphericity error. The method used to accomplish his goals was to measure four spheres selected out of a set of nineteen spheres, with radii ranging from $^{11}/_{16}$ to 2 inches. The samples were measured using a rotational measuring table for roundness measurements, in which each sphere was evaluated using sixty round profiles arranged in groups of three. The reason why only three profiles were in a group was that only three 90 degree profiles could be measured from a spherical surface at a time. Thus, after the first group was measured, the sphere was rotated to a different position and three different round profiles were measured. The procedure was repeated until all sixty profiles were measured. The roundness measurements were calculated by means of the minimum circumscribed circle, in which confidence intervals were calculated for the roundness measurements and for the standard deviation of the measurements. Although the results were satisfactory, the author stated the difficulty of using 2-Dimensional instruments to evaluate 3-Dimensional shapes, since it was very difficult to keep each round profile 90 degrees apart from each other. Moreover, Kanada (1997) studied the minimum number of roundness measurements needed to obtain a good estimate of the sphericity error. The procedure used was to measure four spheres using different values of (*n*) number of round profiles per group, *n* going from 3, 6, 9,…, up to 57 profiles. Since 60 is the total number of profiles, then the maximum number of profile combinations for each value of *n* was limited to 20. The results were graphed using a scatter plot in which the *y*-axis corresponded to the sphericity values while the *x*-axis was the number of round profiles. Results showed that for small diameter balls the minimum number of round profiles needed are 9, while for big diameter ball the minimum number of recommended profiles are 18 or more.

In a similar study, Kanada (1995) focused on the evaluation of spherical forms by means of the iterative least squares and minimum zone models. Actual measurements were not collected; however, the author simulated the measuring data by using Laplace spherical function. The simulator helped the author to generate random noise used to simulate surface errors and to generate the angles that describe the position of the data point in spherical coordinates. The author simulated 7,192 data points that were distributed into five sets, where the simulator provided the sphere radii and standard deviation of the data. The data was evaluated first using the iterative least squares method in which a linear spherical model was used to calculate the parameters of the sphere. The data was also evaluated in a nonlinear spherical model using the minimum zone approach by applying the downhill simplex method. The simplex method uses a triangular shape in which each of the vertices represents different values of the objective function. The size of this vertex either increases or decreases according to the improvements given by following the vertices that provide the minimum value of the objective function. The method was finalized when the convergence criterion is fulfilled; in this case, that criterion was given by the magnitude of the standard deviation of the measurements. For the five sets of data the minimum zone approach outperformed the iterative least squares method, providing smaller zones of error and taking less computing time. In a similar way as Kanada's (1997) paper, the author analyzed the data in order to determine if all data were necessary for sphericity calculation. After performing least squares analysis, the author found that the data can be classified into three zones, such as "hills," "dales," and "medium high data." The hills represent the data that is above the least squares line, while the dales are the data below the line. The medium high data is the data that is in the middle, which does not affect the calculation of the sphericity value, since the sphericity is calculated based either on the subtraction of the minimum from the maximum residual or by minimizing the maximum deviation. Thus, the author proposed a criterion to determine the amount of data required to reduce the computational time. In addition, Kanada (1995) performed a comparison between analyzing the sphere as a set of roundness measurements and analyzing it as a 3-Dimensional surface, where he found no significant differences between the roundness measurement of the equatorial plane and the sphericity value. However, he found that the roundness measurements of longitudinal lines are one third of the sphericity value. Results from this research contradict the guidelines provided by the standards since the ANSI and ISO standards accept the evaluation of sphericity as a combination of roundness measurements. Therefore, the sphere should be inspected as a 3-Dimensional surface in order to avoid any underestimation of the surface deviations.

Huang (1999), Fan et al. (1999), and Samuel et al. (2001) studied the sphericity as a 3-Dimensional problem using different approaches than the ones already revised in the literature. Huang (1999) applied the 3D Voronoi approach to find the exact origin of the sphere that minimizes radial distances. Fan (1999) simulated a mechanical system using the minimum potential energy theory that emulates the analysis of the sphere deviations, while Samuel et al. (2001) applied a similar procedure like 3D Voronoi approach in which he used 3D convex hulls to inscribe and circumscribe the sphere deviations for the calculation of the minimum zone of the sphere.

The 3D Voronoi diagrams applied by Huang (1999) are defined as a plane that split the Euclidian distance of two measured points $A(x_a, y_a)$ and $B(x_b, y_b)$ into half spaces, in which all points on the plane are equidistant to the two measured points. One side of the half spaces that does not contain A is defined as the farthest region $F(A)$ while the space containing A is defined as the nearest region $N(A)$, thus the farthest region of A is the nearest region of the point B labeled $N(B)$, and the nearest region of A is the farthest region of point B labeled $F(B)$. The approach can be extended to multiple points since it can be used to generate n number of half spaces according to the number of paired points available, where the farthest and nearest regions are created by the intersections of the half spaces. The relation between the sphericity problem and the 3D Voronoi diagrams is the existence of a "*max region*" that contains two control points that define the minimum zone of spheres, since these points determine the radial distance between the inscribed and circumscribed spheres. Huang (1999) developed an automated procedure to calculate the minimum zone of spheres.

The steps followed during this procedure were to identify five points among the set of data points for the X-vertex calculation. Then, the inscribed and circumscribed spheres were calculated with the purpose of determining if all points on the set were enclosed by the two spheres. If all points were enclosed then the procedures ends, however, if some points were outside the spheres, then some of five points selected for the X-vertex calculation were replaced by these points outside the spheres, and then the procedure continues until all points are enclosed between the spheres. The procedure was verified under two similar conditions. First, a set of five points was used to determine the X-vertex that generated the minimum radial separation. The five points' coordinates are (4.8, 6.4, 6.0), (0, 0, -10), (0, 9, 0), (0, -9, 0), and (0, -9, 0). Among all possible vertices generated from the set of five points, the vertex (0, 0, 0) provided the minimum radial separation equal to 1. In addition, it was found that points 1, 2, and 3 described the position of the circumscribed sphere with radii of ten, while points 4 and 5 determined the position of the inscribed sphere with radii of nine. The solution of this procedure was used to determine the efficiency of the algorithm under a different condition. The five points were hidden in a set of ninety-five points, making the total sample points equal to one hundred. The new places of these points were 51, 62, 70,79, and 95. The one hundred points were evaluated in set of five points, and it was found that the algorithm needed five updates to locate the five known points. The results were identical to the ones obtained from the first experiment, proving the efficiency and accuracy of the procedure. In addition, the author created several scenarios in which he varied the number of evaluated points ranging from 10 to 100, with the purpose of determining the total number of updates needed to calculate the sphericity, as well as the total number of points that were on the final set. Results for 10 points showed that the average number of updates needed were 2.89 updates, while the average number of points on the final set was 7.81 points. Results for 100 points showed that the average number of updates needed was 6.09 updates, while the average number of points on the final set was 14.12 points.

In a different study, Fan et al. (1999) proposed an analogy to study the minimum zone of spheres, in which he applied the minimum potential energy theory to a mechanical system that behaves similarly to the sphericity problem. The mechanical system was defined by a set of two spherical surfaces that represent the circumscribed sphere with radii equal to $R_o = 1$, and the inscribed sphere with radii equal to $R_i=R_o/\alpha$ ($\alpha > 1$), as well as by a set of mechanical supports that were located in the same position as the measuring points; it was assumed that these supports can only withstand compression force. The system was controlled by a spring that stores elastic energy while contracting. The adjustment on the spring caused modifications to the dynamic of the whole system, since it was assumed that there are only five points that are in contact with the spherical surfaces. Hence, the contractions of the spring change the contact points between the inscribed and circumscribed spheres, until the elastic energy of the system was reduced to the minimum. The five supports in contact with the spherical surface behaved in different ways, which is the reason why the author defined four modes to describe the system. The first mode was called M4-1 in which four points are in contact with the circumscribed sphere while one point is in contact with the inscribed sphere. The second and third modes were called M3-2 in which there are three supports in contact with the circumscribed sphere and two with the inscribed sphere, and M2-3 in which two supports are in contact with the circumscribed sphere while two are in contact with the inscribed sphere. The last mode was called M1-4 where one support is in contact with the outer sphere while four supports are in contact with the inner sphere. The principles of virtual work and the equilibrium equations were applied to each of the described modes, where it was determining that if the supports were in compression then the reaction should be positive, but if the reaction was negative then the energy spent on the system could be reduced even more, giving chance for improvements. In a similar way as Huang's (1999) algorithm, Fan's procedure was to update the active support each time the reaction was negative; the procedure was repeated until all supports were evaluated and none of them yielded a negative reaction. The algorithm was verified using a set of fifty data points that were generated randomly. The points were evaluated first using the least squares approach in which the solution was used as initial conditions for the proposed algorithm. Results obtained from this procedure showed that the minimum potential energy analogy took 1.3 seconds using six iterations to calculate the zone of error of the sphere given by 7.66 μm.

Samuel et al. (2001) proposed a model to evaluate spherical forms similar to Huang's (1999) 3D Voronoi diagrams and to Fan's et al. (1999) analogy, where he employed the use of convex hulls for the calculation of the parameters of the sphere as well as to calculate the minimum zone of spheres. Three techniques were applied for the evaluation of spheres: the minimum circumscribed sphere, maximum inscribed sphere, and minimum zone spheres. The logic behind the construction of the 3D convex hulls lies on a new technique developed by the Samuel et al. (2003) called equidistance (ED) plane and line. The ED plane perpendicularly slices a line connecting two measured points P_1

and P_2 from a spherical surface, in which all points contained on the ED plane are equidistant from both mentioned points. Similar to the Voronoi diagrams, the procedure was adapted to n number of points. For three points, the intersection between all ED planes, created a center point among all points that was used to generate a perpendicular line also called as ED line, in which all points contained on this line are equidistance to the three mentioned points. Thus, any point over the ED line was used to determine the origin of the circumscribed or inscribed sphere.

The procedure used to calculate the circumscribed sphere was based on the calculation of the 3D convex outer hull, which used the logic behind the ED plane and the ED line. For a set of points, the procedure looked for identifying the nearest and farthest centers that passed through at least four vertices. The data were evaluated as triangular planes of three points that were updated based on the location of the centers, where the final error was obtained by selecting a center that provided the smallest radius that surrounded all measured points. The inscribed sphere or 3D convex inner hull was calculated based on the center found for the circumscribed sphere, and on the same procedure used to calculate the outer convex hull. In addition the data were transformed in order to simplify the calculations of the procedure. The last approach was the minimum zone spheres, defined as the zone between two concentric spheres that is the smallest radial distance that enclosed all measured points. For this procedure, the initial conditions were found by applying the information obtained from the outer and inner convex hulls' approaches. Five data points were required to construct two concentric spheres, thus the procedure passed several times through the data until it identified the five data points that provided the minimum radial separation between spheres.

An experiment was performed to verify the accuracy of the procedures where two sets of data points were used. The first points were a set of 26 data points found in the literature, while the second set was simulated based on those values obtained from the literature. Results from this experiment showed that the three proposed methods performed better than the least squares technique. The minimum zone spheres performed better than the results found in the literature, since they yielded a zone of error of 2.82842μm while those in the literature were in the order of 2.88μm. In addition, the author found that the proposed methods were not affected by the convergence value specified in each method, but by the data points that controlled the zone of error.

Samuel et al. (2003) performed a different study in which they analyzed the form errors of circular and spherical shapes by using coordinate measuring machine data. The data were analyzed using the linear spherical and nonlinear spherical models shown in Equations (2.9) and (2.10). For the nonlinear spherical and circular models the data were applied directly from the CMM output, while for the linear spherical and circular models the data were transformed into polar coordinates, which employed the limacon and limacoid surfaces as assessment features. The nonlinear circularity evaluation was determined first by applying an equation similar to the nonlinear spherical model from Equation (2.10), but reduced in dimensionality since the circular shapes only require the *x* and *y* coordinates instead of the *x*, *y*, and *z* coordinates required by the spherical shapes, as seen in Equation (2.11).

$$e_i = R - \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} \tag{2.11}$$

The nonlinear model used a circle as an assessment feature with an origin located at $O(x_0, y_0)$ and radii equal to R. The minimum zone of the circle or circularity was found by calculating the difference between the maximum deviation and the minimum deviation. The linear circularity evaluation was calculated in a different way than the nonlinear procedure, since it requires the data to be transformed into polar coordinates to apply the limacon shape as an assessment feature. The procedure used to transform was the follow. First, the nonlinear model from Equation (2.11) was used to calculate the origin location x_0 and y_0, and the radii R; where, instead of estimating the parameter using the whole set of data, the parameters were calculated based on three data points. then, the estimated parameters were used to calculate the residuals for all data points. These residuals were used to transform the radial distances r_i that were calculated by adding to each of the residuals, the negative value of the minimum residual plus the value of a variable d_0 that was chosen arbitrarily. The angle θ was calculated by the tangent inverse of $(y_i - y_0)/(x_i - x_0)$ where the x_0 and y_0 were obtained by means of the mentioned three data points. The linear model for circular shapes is similar to the sphere model shown in Equation (2.9) but reduced in dimensionality as mentioned before. The parameters of the nonlinear spherical model were calculated using Equation (2.10) and were assessed based on the spherical feature, while the parameters of the linear spherical models were obtained by transforming the data in a similar way as it was done for the circular shapes.

An experiment was performed to verify the accuracy of the two models where the data were obtained from Fan et al.'s (1999) study. The models were analyzed by means of the simplex search techniques. Results from this study showed that the output generated from the four models were very close to those results obtained in the literature. For circularity, Fan et al.'s (1999) found that the form error of the data was 29.2860µm, while Samuels et al.'s (2003) transformed model showed a form error of 29.2816µm and the untransformed model showed a form error of 29.2809µm. For sphericity, Fan et al.'s (1999) form error was 7.660µm, while Samuels et al.'s (2003) transformed model showed a form error of 7.66023µm and the untransformed model showed a form error of 7.66011µm. Regarding the zone of error the four models were comparable to the results and approach used in Fan et al.'s (1999) study. However, the transformed model does not represent the real parameters. Hence, this model will mislead the metrologist when trying to make inference regarding the state of the manufacturing and the inspection processes.

Balakrishna (2005) and Balakrishna and Raman (2007) applied the support vector regression (SVR) technique to evaluate the form tolerance of spherical shapes. The main goal of this study was to provide a new approach to overcome the deficiencies of the least square method. Since it was found that the least square method requires the data to follow the normal distribution and it is sensitive to outliers, it overestimates the zone of error. On the other hand, the support vector regression only requires the data to be independent and identically distributed. The author applied the ν-SVR technique that minimized the zone of error based on a parameter ν. This parameter was used to determine the fraction of errors that can lie outside the error tube. The sphere is by nature a nonlinear shape, which is why the author employed the radial basis kernel that allowed her to solve the sphericity problem in a higher dimension. The performance of the support vector regression approach was evaluated by comparing it to the linear least squares, the nonlinear least squares, the linear optimization approach, and the nonlinear optimization approach. The data were generated using the random sampling strategy with sample sizes of 20, 64, 100, 200, and 500 data points that were collected using a coordinate measuring machine (CMM). The collected (x, y, z) data points were transformed into spherical coordinates (r, θ, β) so that they can be used in the proposed approach. The ν-SVR approach was solved using the SPIDER© software that was implemented in Matlab®. Results from this study showed that the ν-SVR zone of error was smaller than the zones of error from the other methods, however, no statistical analysis was performed neither on the fitting algorithm nor on the sample size. In addition, no information was provided regarding the estimated parameters of the sphere, thus, the conclusions of these papers were based exclusively on the minimum zone of error.

Most of the existent research articles about the sphere tend to provide information about the zone of error of the inspected sphere, instead of providing a complete analysis based on the fitted parameters and the zone of error. The problem in focusing only on the zone of error makes the inspected sphere look as it was manufactured according to specifications, however, if the parameters of the sphere are taken into account together with the zone of error, then the analysis of the inspected sphere will be accurate since it can have a very low zone of error with an estimated radius that is higher or lower than the desired values, or it can have a high zone of error with estimated parameters being at the correct size of the radius. The most common mathematical models used by the mentioned authors are explained and analyzed later on in this research with the purpose of developing improvements that can describe the spherical forms in an accurate way.

2.6.2 Conical Form

The conical form can be found in the literature as a complete conical form or as a frustum form. The cone is described as a pyramid with a circular cross section (Weisstein, 1999). The cone has three parameters that describe it: the height of the cone (*h*), the radius of the base of the cone (*r*), and the slope (*s*) which is given by the relation of the first two parameters (γ), Equation (2.12). In the literature, most of the studies locate the position of the origin of the cone *O* (x_0, y_0, z_0) at the bottom of the surface; however, this research will consider the origin of the cone at the vertex, Figure 12.

$$\gamma = \tan^{-1}\left(\frac{r}{h}\right) \tag{2.12}$$

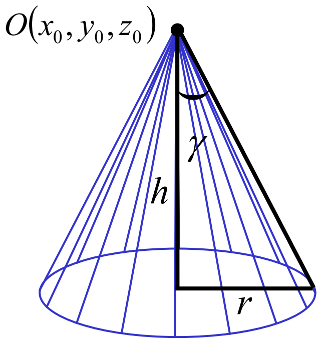


Figure 12. Parameters of the Cone.

On the other hand, the frustum is a conical surface that has the top part removed. Unlike the cone, the frustum has four parameters that describe its shape: the radius of the base (*r*), the radius of the top cross section (*r*"), the height of the cone (*h*'), and the slope (*s*) which is defined by the relation between the height of the cone and the both radii of the cone.

$$\gamma = \tan^{-1}\left(\frac{r - r''}{h'}\right) \tag{2.13}$$

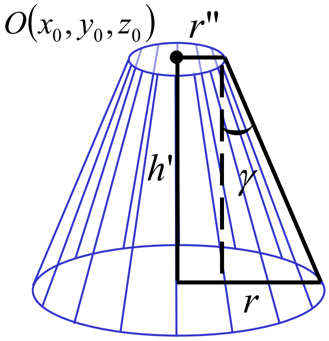


Figure 13. Parameters of the Frustum.

The form verification of conical forms has not been studied extensively in the literature. Studies from Tsukada et al. (1988), Chung (1997), Chatterjee et al. (1998), Chen et al. (2000), Prakasvudhisarn et al. (2004), Ahn et al. (2002), and Ahn (2004) focused on studying the form verification of these shapes by estimating the paratemers of the cone; the first, the second, and fifth studies are the only ones that report values of the zone of error.

Tsukada et al. (1988) proposed a model to calculate the zone of error of conical taper forms using the simplex method and the least squares method. The study employed fifty seven cones with diameters ranging from 6 to 60 mm, rates of taper from 1/50 to 1/3, and cone heights from 20 to 100 mm. The conical tapered parts were evaluated using a measuring machine designed to inspect cylindrical parts, where the radial deviations, defined as the distance from the surface of the cone to the base line of the measuring machine, were recorded. The conical form was inspected by collecting data points from the surface where five hundred positions were recorded around the cone surface by revolving the rotational table of the measuring equipment, and several positions were recorded along the axial direction ranging from 1.03 to 6 mm. Due to the imperfections of the rotational table, the author had to mathematicaly counteract the effects of the origin offsets and axis misalignments in order to provide valid values of the zone of error. For this reason, the definition of direction cosines was used to cancel out the origin offsets of the *x* and *y* components and the misalignments between the axis of the cone and the axis of the measuring machine. However, it was assumed that there is no offset on the *z* component, which means that the *z* component of the origin of the machine was placed on the same position as the *z* component of the geometrical origin of the part, causing invalid results as it was stated at the beginning of this section. The calculation of the zone of error was performed by enclosing the cone deviations using inscribed and circumscribed cones. The parameters of the cone were estimated by applying the least squares method and the simplex method, where for the later method, five variables were optimized using a simplex with six vertices. The convergence criterion for the simplex method was set to be the standard deviation of the objective function at each vertex that was compared against a constant set at 0.01μm. Results obtained from both fitting methods showed that the least squares zone was greater than the one obtained from the simplex method. The differences were caused due to the tendency of the least squares method to overestimate the zone of error.

Chatterjee et al. (1998) performed a study to develop mathematical models to verify the form of error of conical forms, in which the Chebychev approximation method was used to solve the proposed models. The conical form of error was defined for several scenarios in which the best fitted cone was estimated based on specific known data. The first scenario was used to prove that a cone can be constructed based on the the three given points and the vertex; as shown in Equation (2.14).

$$k = \frac{\left(X_i - X_0\right) \cdot P}{\left|X_i - X_0\right|\left|P\right|} \qquad \text{Equation (2.14)}$$

where k represents the positive and negavite half cones, **P** is a vector that represents the direction of the cone axis, X_i represents the data points (i = 1, 2, 3), and X_0 represents the vertex of the cone. The formulation generated eight combinations due to the fact that k can take positive or negative values for each of the point coordinates; it was found that four cones can be generated for the given data points that share the same vertex. The second scenario involved the use of the Chebychev approximation to generate two cones that enclosed the form deviations of the inspected cone given a vertex and four data points. The Chebychev cone was obtained first by applying the least squares method to a set of data points, where four points out of the data points set were selected. Two points out the four points were selected among the points that were above the middle line, while the other two were chosen from below the middle line. Then, the four points were processed with the Chebychev algorithm, by using as an initial solution the axis direction provided by the least squares method. The solution from the Chebychev was either a non-optimal or optimal solution. For the non-optimal solution, four given points were not able to create a fitted cone that enclosed all deviations, thus the four points were replaced and fitted again. For the optimal solution, it meaned that the created fitted cone was able to enclose all deviations given by the set of points. Thus, it was possible to create a fitted cone with four given points and a vertex based on the iterative process of the Chebychev approximation method. The third scenario focused on creating a fitted cone based on the axis of the cone and three given points. One of the advantages of this scenario was that the dimensionality of the problem was reduced since the only two unknowns were the position of the vertex of the cone and the magnitude of the half angle. The fitted cones enclosing the deviations were created in a similar way as the cones created in the first scenario, however, one of the main differences was the X_0 in Equation (2.14) that was subtituted by $(t \pm l)P$, which represents the vertex of the fitted cones. The modified equation was manipulated after sustituting the values of the three given coordinates, where it was found that only four fitted cones can be generated. The four cones were generated by rotating the points along the given axis. In this way each point had a unique location based on the direction of rotation and the position on the 3-Dimensional space, which was used to define the fitted cones. The last scenario was to contruct the fitted cones based only on a given set of points. The simplex search algorithm was used to determine the location of the vertex of the cone, where the Chebychev width was calculated at different positions in order to determine the best location of the vertex. The last scenario which is preferred, to evaluate the form of error of conical forms was verified by inspecting from a tappered conical form. The object was measured at different locations using sixty data points that were evaluated using the least squares and the Chebychev approximation. Results from this experiment showed that the Chebychev method provided a slightly smaller zone of error (0.0032 inch) compared to the least squares method (0.0035 inch). The axis and vertex location were very similar for both methods. However, the angle of the cone was identical for both fitting methods.

Chen et al. (2000) developed a study to estimate the parameters of conical tapered forms. The paper proposed a new mathematical model to calculate the minimum zone of error by applying the definition of the minimum circumscribed and maximum inscribed cones, which are the ones that have the minimum volume and maximum volume, respectively. The cone was mathematicaly defined by the axis of symmetry which had four parameters *a*, *b*, *l*, and *m*, where *a* and *b* were the eccentricity factors that compensate the origin offsets with respect to the *x* and *y* components, respectively. The parameters *l* and *m* defined the axis misalignments between the *x* and *z* component and between the *y* and *z* component, respectively. The four parameters were used to determine the minimum circumscribed and maximum inscribed cones. The two cones were calculated by estimating the half angle of the cone, which at the same time was estimated based on the vertex of either minimum or maximum cone, the inspected points, and the parameters *l* and *m* that are counteracting the effects of the axis misalignments. The half angles of both cones were used to determine the distance between the *i*th point and the axis line of the cone. The five points that had the maximum angles were chosen to estimate the parameters of the axis misalignments and the vertex of the minimum circumscbribed cone, and the five points that had the lower angles were used to estimate the parameters of the axis misalignments and the vertex of the maxumim inscribed cones. The output of the mathematical model showed estimated parameters for both maximum and minimum cone. For the minimum circumscribed cone, the estimated parameters were two points one on the upper surface (9.5570, 5.6574, 22.6476) and one in the lower surface (-18.4880, -0.5083, 30.8011); the radius of the upper base was 13.5001, while the radius of the main base was 18.4999, a half angle of 30°59', and a cone height of 8.3214. For the maximum inscribed cone, the parameters were two points, one on the upper surface (12.9543, 3.1074, 22.3751) and one in the lower surface (-18.4880, -0.5083, 30.8011); the radius of the upper base was 13.4100, while the radius of the main base was

18.4001, a half angle of 31°0'2", and a cone height of 8.3213. Although the authors are calculating the angles in an “interactive” way, they did not provide any information regarding the way the residuals are calculated and the the size of the zone of error. In addition, one risk that can be encountered while estimating both inscribed and circumscribed cones is that the model can provide different half angular values for both cones. Hence, the zone of error will be created by two cones that are not strictly parallel to each other, which will not follow the definition of the minimum zone provided by ANSI and ISO standards that state that the zone enclosing the part deviations should take on the shape of the inspected object.

Prakasvudhisarn (2002) and Prakasvudhisarn et al. (2004) studied the form evaluation of cones and conical frustum forms by evaluating the effects that several independent variables have on the zone of error of conical forms. The five independent variables used in this research were the sampling strategy, fitting algorithm, conical surface, conical specimen, and sample size. The sampling strategies defined as the procedure used to collect information from the surface of the part, such as Hammersley, Halton-Zaremba, and aligned systematic strategies, were adapted to fit conical and frustum forms since no previous study had defined them. In addition, the study provided two ways to inspect the form of conical features, using linear and nonlinear formulations, as can be seen in equations (2.15) and (2.16), respectively.

$$d_i = r_i - \left[R_0 + \left(\frac{R_1 - R_0}{h} \right) z_i + (mz_i + x_0)\cos(\theta_i) + (nz_i + y_0)\sin(\theta_i) \right] \tag{2.15}$$

$$d_i^{'} = \frac{\sqrt{(m^2 + n^2 + 1)\left[(x_i - x_0 - mz_i)^2 + (y_i - y_0 - mz_i)^2\right] - \left[n(x_i - x_0) - m(y_i - y_0)\right]^2}}{\sqrt{m^2 + n^2 + 1}} - \tag{2.16}$$

$$R_0 - \left(\frac{R_1 - R_0}{h}\right)\left(\frac{z_i}{\cos(\alpha)} + D\cos(\theta)\right)$$

where:

r_i is given by the Euclidian distance between the x_i and y_i of the ith point.

R_0 is the radii of the base of the cone.

R_1 is the radii of the upper cut of the frustum.

h is the height of the cone.

z_i is the z component of the ith point.

m is the first component of the axis of assessment

x_0 is the component that counteracts the origin offsets.

θ_i is the angle created by the x_i and y_i component of the ith point.

n is the second component of the axis of assessment.

D is the orthogonal distance from the measured point to the axis of assessment.

The conical forms were evaluted using an experiment in which the mentioned factors were analyzed. Eight conical forms were used, where four specimens were complete conical forms while the remaining four were conical frustums. The eight forms were held on the measuring table using conical areas, which were quadrangular blocks of 13.74 inch^2 and 22.71 inch^2. The form of these specimens was inspected using a Coordiante Measuring Machine, which automatically recorded the point coordinates from the surface of these parts. The number of points collected from each specimen were of 8, 16, 64, and 256 data points. The factor combinations were evaluated using the least squares and the minimum zone methods. A program written in Visual Basic for Applications in Microsoft® Excel was used to perform both of the least squares analyses, while LINGO® 6 was used to perform the minimum zone algorithms. Results from this experiment showed that higher sample sizes yield higher zones of error since more information is collected from the part surface. It was found that the nonlinear optimization method provided the smallest zone of error; however, no analysis was performed to determine which fitting method the parameters of the cone were closer to the ideal values. Thus, it is erroneous to conclude that a model performed better than others if there is no analysis on the parameters of the cone. Although both formulations provided a good estimate of the parameters and zone of error of the cone, none of them counteract the origin offsets with respect to the z component. Furthermore, the inspection of the cone is performed using radial distances that are calculated from the *x* and *y* coordinate components, instead of using information from all three components.

The studies by Ahn et al. (2002) and Ahn (2004) focused on the form inspection of 2-Dimensional and 3-Dimensional shapes analyzed by means of the least square orthogonal distance fitting. It is important to mention that both studies came out from the doctoral dissertation of the first author, where the first study was published in the year 2002 as a journal paper, and the second study was published in the year 2004 as a book. The objective of both studies was based on the computational efficiency of the developed models, reason why the later study was published as a “Lecture Notes in Computer Science.” Although, that the author focused on developing fitting models in the metrology area, the study did not focus on providing information regarding neither the sampling strategy nor the procedures used to collect the data. For this reason, the only inferences that can be made from these studies are regarding the fitting models, which will be explained in detail in the fitting algorithms section.

Ahn (2004) defined the cone as a function of the vertex angle created by the axis of the cone and the peripheral face of the cone; the radius of the base, and the vector composed by the Cartesian components, Equation (2.17). The conical function was evaluated under the orthogonal distance fitting, where the origin position and the rotational angles were defined as constraints based on the mass center of the measuring data points, Equation (2.18).

$$f(\psi, r, \mathbf{x}) \overset{\Delta}{=} x^2 + y^2 - \left(z \tan(\psi / 2)\right)^2 = 0 \tag{2.17}$$

$$f_{\mathrm{c}}(\mathbf{a}_p, \mathbf{a}_r) \overset{\Delta}{=} \left(\mathbf{X}_o - \overline{\mathbf{X}}\right)^{\mathrm{T}} \mathbf{r}_z(\omega, \varphi) = 0 \tag{2.18}$$

The author validated the proposed model by using ten point coordinates that were obtained from the German Federal Authority PTB. The point coordinates were used first to acquire the initial conditions for the fitting algorithm. The author proposed that the initial conditions were calculated based on geometric models that were simpler than the studied geometric shape. In this case, the initial conditions of the origin position and the radius magnitude were calculated by fitting measured data to the circularity and the cylindricity models, and the initial value of the vertex angle was set to zero, since none of these models calculated the vertex angle. The results reported in this study were based only on the parameters of the cone, therefore, the magnitude of the zone of error was not provided. The following table shows the results obtained from the circular, cylindrical, and conical models, where it can be seen that the initial conditions provided by the simpler models were very close to the final solution. However, the author mentioned that the orthogonal distance fitting (ODF) took a significant amount of time to converge due to the initial conditions that were provided, since it was found that the conditions provided by the cylinder fitting represented a local minima.

Table 3. Comparison of circle, cylinder, and cone parameters using ODF model.

Parameter	ψ**[rad]**	r**[mm]**	X_o**[mm]**	Y_o**[mm]**	Z_o**[mm]**	ω**[rad]**	φ**[rad]**
Circle	-	283.04	694.53	-889.73	-498.10	2.053	0.547
Cylinder	-	379.09	561.53	-702.15	-398.22	1.609	-0.157
Cone	1.4262	276.44	706.72	-890.52	-499.10	2.055	0.587

2.6.3 Torus Form

Hilbert et al. (1952) and Schwartz (1967) described the torus as a surface of revolution, created when a circle is revolved around the *z*-axis. It has a doughnut shape with the origin located at the center of the complete surface. However, Bourke (2007) mentioned that the torus can be formed using a finite number of spheres arranged around a circle, where the radius of the spheres represents the radius of the torus tube.

The torus equation has been defined for several coordinate systems, but it has two main parameters that describe it. These parameters are the internal radius of the tube (*a*), and the radius (*c*) that goes from the origin of the torus to the middle of the tube, as can be seen in Figure 14.

$$a^2 = \left(c - \sqrt{x^2 + y^2}\right)^2 + z^2 \tag{2.19}$$

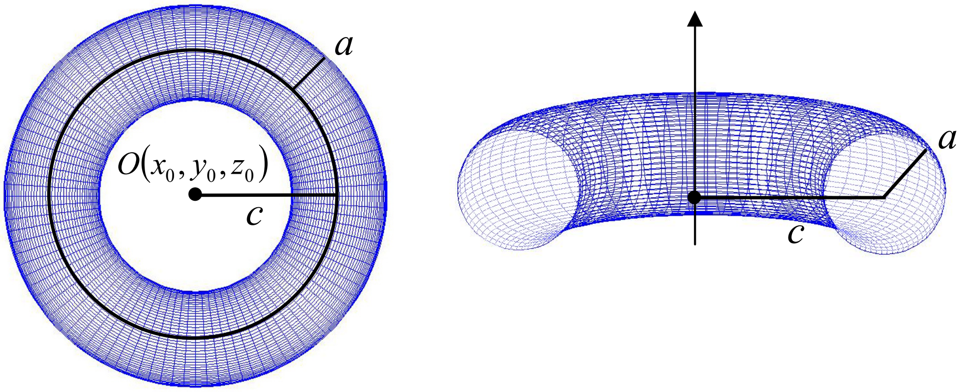
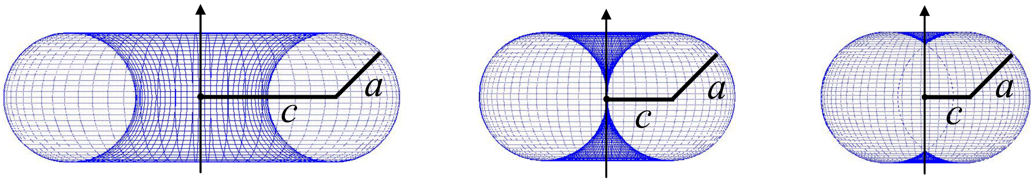


Figure 14. Parameters of the torus

According to Weisstein (1999), the torus can be named in different ways that depend on the values of the mentioned parameters; for example, if the torus has a radius *c* greater than radius *a,* it is called a Ring Torus. If the value of radius *c* is equal to radius *a*, then it is called a Horn Torus, and if the radius *c* is less than the radius *a*, it will be called Spindle Torus, Figure 15.



Ring Torus ($c > a$)

Horn Torus ($c = a$)

Spindle Torus ($c < a$)

Figure 15. Cross section of Torus forms

The study of form verification of torus forms is not as extensive as other geometrical shapes; this can be caused by either the complexity of shape or due to the lack of ISO and ANSI procedures to inspect these shapes. In the literature, there are several papers related to the application of torus shapes in other fields that are different from coordinate metrology, however, there are four papers by Ahn et al. (2002), Aguirre Cruz (2003), Ahn (2004), and Aguirre Cruz and Raman (2005) that proposed new methods to verify the form of torus forms. The second paper by Aguirre Cruz (2003) is the Master's thesis of the author of the present research, which was published later on. Therefore, the second and last, and the first and third studies are summarized together in the next paragraphs. The research by Ahn (2004), and Aguirre Cruz and Raman (2005) were focused on the study of form verification of torus shapes; however, each study applied different mathematical models to describe the geometry of the torus form as well as different procedures to analyze the data.

The studies by Ahn (2004) focused on the form inspection of 2-Dimensional and 3-Dimensional shapes, such as the case of the conical model explained in the previous section. Similar to the conical model, the author described the torus form as a quartic surface defined by two radii, r_2 that represents the radius of the torus, and r_1 that represents the radius of the torus tube, as seen in the following function defined in Cartesian coordinates.

$$\begin{aligned} f(r_1, r_2, \mathbf{x}) \overset{\Delta}{=} x^4 + y^4 + z^4 + 2\left(x^2y^2 + y^2z^2 + z^2x^2\right) - 2\left(r_2^2 + r_1^2\right)\left(x^2 + y^2\right) \\ + 2\left(r_2^2 - r_1^2\right)z^2 + \left(r_2^2 - r_1^2\right)^2 = 0 \end{aligned} \tag{2.20}$$

The initial conditions of the torus form were calculated using only the circular shape, which was applied to the orthogonal distance fitting (ODF) algorithm to calculate the parameters of the torus. It can be seen in the following table that the parameters obtained from the circle are not as close as the ones found for the conical form, due to the geometrical differences between the circular shape and the torus form.

Table 4. Comparison of circle and torus parameters using ODF model.

Parameter	r_1**[rad]**	r_2**[mm]**	X_o**[mm]**	Y_o**[mm]**	Z_o**[mm]**	ω**[rad]**	φ**[rad]**
Circle	-	9.059	0.383	1.527	4.716	0.493	0.358
Torus	2.510	7.512	1.316	1.955	3.232	0.527	0.272

The study performed by Aguirre Cruz and Raman (2005) focus mainly on the development of mathematical models to verify the form errors of torus shapes. The research was the first one in the literature to develop fitting algorithms as well as to adapt the existent sampling strategies into a torus form. Several geometric models were analyzed for torus form representation, however, only two "Complete Torus" approximation, and "Transformed Torus" approximation were found feasible for this study, where both approximation methods inspect the torus form with respect to the central axis *z*. The Complete Torus approximation, defined in polar coordinates, was intended to analyze the torus form by defining horizontal radial distances in terms of the parameters of the torus, *c* and *a*. The horizontal radial distance were measured using a virtual *xy* plane located at the same height of the measured point, thus the radial distances is given by using the Euclidian distance principle between the measured point $P(x_i, y_i, z_i)$ and the point where the virtual *xy* plane and the central axis *z* intersect $P(0, 0, z_i)$. The Transformed Torus approximation is analyzed by transforming the ring torus into a horn torus. The advantage of this transformation is to simplify the mathematical analysis. Thus, this approximation was intended to analyze the torus form using irregular-vertical cross sections that are perpendicular to the *x* axis. Therefore, the two approximation formulas were used to measure the minimum zone of error, which was calculated by comparing the measurements against the formula obtained from the approximation models, including the origin offsets equation obtained from the Limacon approximation as well as the misalignments equation that corrects the errors due to the axis of assessment.

An experiment was developed to verify the accuracy of the two approximation methods, where 10 torus specimens, 5 positive torus forms and 5 negative torus forms were used. Although, 10 specimens were used only 8 parts were used for the experiments in order to fulfill the randomization requirements. Three sample sizes were selected in the range from 40, 80 and 120 data points, where three sets of 40, 80 and 120 data points were generated using Align Systematic, Hammersley and Random sampling strategies. The 72 factor combinations were randomized in order to determine the inspection sequence. The data points were collected using a Coordinate Measuring Machine that measured the parts in an automatic mode. The measured points were evaluated using the linear least squares and the minimum zone method, where the torisity values or zone of error of torus parts was calculated. The 72 values of torisity were analyzed using the ANOVA procedure were it was found that the torisity value of the positive torus forms were smaller than the values of the negative torus forms, however, it was found that one negative part was not machined properly, thus causing the ANOVA procedure to detect the differences. The fitting algorithm was also a significant factor because the minimum zone method outperformed the linear least squares. In addition, the Complete Torus approximation performed much better than the Transformed Torus approximation, due to the robustness of the first approximation compared to the irregular-vertical cross sections of the Transformed Torus approximation. The sample strategy was used as a blocking factor since the main purpose of the study was to determine the accuracy of the approximations. Results from this study showed that the best factor combination for measuring torus forms was to apply a sample size of 80 data points, analyzing the data using the Complete Torus approximation and the minimum zone algorithm.

2.7 Sampling Strategies and Sample Size

In metrology, the inspection process plays a fundamental role in determining whether or not machined parts were created according to specifications. The process focuses on collecting data points from the part and fitting them using mathematical models that will determine the tolerances of the part. The data collection is indeed an expensive process because of the number of machined parts that have to be inspected from a lot and the number of data points that have to be collected in each part. Although, there is inspection equipment that can collect data points from a part in a matter of seconds, they are costly, which is why the coordinate measuring machine (CMM) is more popular in industry. Still a disadvantage of the CMM is that it samples the part in a discrete way, which increases the inspection time and cost.

The inspection process is an operation that does not add value to the product, since it is focusing on verifying if the machining processes were machining parts within the quality limits. The inspection process is susceptible to errors due to the fact that operators and machines are controlling the process and can induce errors. For this reason, Dodge et al. (1945) identified two risks, the producer's risk and the consumer's risk that are encountered as a result of the inspection process. The producer's risk is the probability of rejecting a part when the part is non-defective also known as type I error (α), while the consumer's risk is the probability of accepting a part when the part is defective also known as type II error (β), (Montgomery, 2001). Moreover, Dodge et al.

(1945) stated that the mentioned risks were not enough to determine an efficient sampling plan, since they tend to view the lots of parts as independent samples and not as dependent series. Due to this, he included the concepts of quality control to the inspection process, which allowed him to base his analysis not only on the actual conditions of the process but on the existing conditions too, with the purpose of minimizing the collection of data. The statements from Dodge et al. (1945) are very important and are going to be used later on to establish the sampling strategy and the sample size needed to determine the quality of a machined part.

The importance of evaluating the geometrical tolerances of machined parts is due to the fact that each manufacturing process leaves a unique fingerprint over the surface of the part that is created by a combination of amplitudes and frequencies. The high frequencies are referred to as the surface roughness, medium frequencies as the waviness and low frequencies as the form error (Hocken et al., 1993). Therefore, if a 100% sampling is performed then the inspection process will yield the true form error of the part. However, the main goal of an efficient inspection process is to verify the tolerances of machined parts by minimizing the cost and the time of inspection. For this reason, authors developed sampling strategies that were able to capture the form errors of machined parts. Some of the most known strategies are Hammersley, Halton-Zaremba, random, aligned systematic, and uniform, among others. The majority of the mentioned strategies are based on a fixed or semi-random point distribution that sometimes caused them to perform a poorly, since at times they are not able to sample the regions that govern the width of the zone of error (Edgeworth et al., 1999). Although there are no perfect sampling strategies, there is a possibility to study the relationship between the type of process, the machining parameters and the form of the part, which will provide information about the fingerprint that the specific combination of factors produced on the part surface. Therefore, if this relationship is known, then it will be possible to develop a new generation of sampling strategies that focus its point distribution only on those areas that define the error of the machined part. These sampling strategies are commonly known as adaptive sampling, since they adapt the search pattern according to the quality of the machined part (Edgeworth et al., 1999). The drawback of these strategies is the high cost involved in their development, because of the need of iterative algorithms used to determine the sample size, the location of the data points, and the patterns between the mentioned factors. In addition to the mentioned strategies, there are other sampling strategies that have not been studied in depth and appear to be more efficient than some of the mentioned ones. Such is the case with the honeycomb lattice distribution proposed by Narayanan Namboothiri et al. (1995) whom found that the honeycomb lattice requires less inspection time than the aligned systematic, and the spiral strategy proposed by Collins, Berrocal, Aguirre Cruz, and Raman (2006, 2007) whom created a strategy that focuses its point distribution on those zones of the workpiece that contain the maximum deviations.

In the literature, the accuracy of the sampling strategies is determined by evaluating the discrepancy of the sequence of data points. Studies performed by Roth (1954) on data sequences produced a lower bound discrepancy in terms of the sample size and the dimension of the inspected feature, which was used by Woo et al. (1993) to test the accuracy of the Hammersley and uniform sampling strategies, and by Collins, Berrocal, Aguirre Cruz, and Raman (2006, 2007) to verify the accuracy of the spiral sampling strategy. Roth (1954) created the lower bound discrepancy by mathematically integrating the residuals (errors) of the sequence with respect to size of the rectangular inspection zones. The zones had a maximum area of 1 unit^2, and the shapes were squares or hypercubes, according to the dimension of the problem. The square zones were used to calculate the residuals (errors) by gradually incrementing the size of the squares and comparing them against the number of data points located inside them. The residuals (errors) were small when the data points contained inside the squares or hypercubes were proportional to their size, and vice versa. Woo et al. (1993) used the mentioned methodology to verify the accuracy of the Hammersley and uniform strategies, since he found that the discrepancy of the Hammersley sequence had a quadratic reduction compared to the lower bound discrepancy proposed by Roth (1954), which means that the Hammersley sequence needs only 10 data points compared to the 100 points needed by other sequences.

The two sampling sequences were compared by measuring data points from two machined surfaces that were simulated using the Wiener model and the Gaussian model. The differences between these two modeling techniques is that the first produced smoother surfaces similar to those surfaces obtained from a single point rotary cutting; while the latter generates rougher surfaces similar to those obtained from an electro-discharge machining or sand blasting processes. The Hammersley sequence had sample sizes of 12, 32, 64, and 128 data points, while the uniform sequence had sample sizes of 100, 625, 2500, and 15625 points arranged in grids of 10x10, 25x25, 50x50, and 125x125, respectively. The reason why the sample sizes between distributions are different is because the authors tried to emulate the efficiency of the Hammersley sequence. The sequences were compared based on the root mean square error (RMS), which showed indeed that the Hammersley distribution is more efficient than the uniform distribution, since it was found that the former needs a quadractic reduction in the number of points to produce the same results than the later.

In a different study, Woo et al. (1995) followed a similar comparison procedure to verify the accuracy of the Hammersley, Halton-Zaremba, and uniform sampling sequences. The data points generated from the Halton-Zaremba strategy were found to be very similar to those obtained from the Hammersley distribution. However, it was found that the former had two main differences. Firstly, it was defined only in two dimensions, and secondly, the number of data points generated must be a power of 2. For this reason, the author compared the accuracy of the three strategies by using the lower bound discrepancy obtained by Roth (1954). The author found that the Hammersley as well as the Halton-Zaremba sequences needed fewer points (quadratic reduction) to achieve the same level of accuracy obtained using the uniform distribution. In order to probe these results, the author evaluated the three strategies by measuring the roughness error and the flatness error from simulated parts. The surfaces of the machined parts were simulated using the Wiener process, the average Wiener process, and isotropic process, where each surface was defined as a grid of 300 x 300 points. The surfaces were used to evaluate the roughness and the flatness errors. The roughness error was evaluated in terms of the root mean square error (RMS) that was obtained by applying the point distribution of each sampling strategy using sample sizes of 16, 64, and 256 data points. The results obtained from the RMS analysis showed that the performance of Hammersley and Halton-Zaremba strategies produced very similar results, while the uniform distribution produced much higher RMS values. For the evaluation of flatness error, the least squares and minimum zone algorithms were compared using the same sample sizes and same number of replications as the ones used before. The results obtained from the RMS values showed that the three strategies performed similarly, except for RMS values obtained from the surfaces generated using the isotropic process, which showed a higher value of RMS values for all strategies. In general, the author found no significant differences between the Hammersley and Halton-Zaremba strategies in 2 dimensions. In addition, the author found that both strategies are preferred when compared to the uniform distribution, but their accuracy in practice was not as efficient as the quadratic reduction found in the theory.

Narayanan Namboothiri et al. (1995) proposed a collision free path that was used to find a distribution sequence that provided the smallest inspection time and highest level of precision. The authors first compared the efficiency of the systematic sampling versus the stratified random sampling by using the variance of the samples. The data points were generated for the systematic sampling by selecting a random data point from different batches of *k* units each, where the magnitude of the first data point determined the distribution of the sequence. On the other hand, the data points from the stratified random sampling were generated by selecting a random point from each of the zones in which the sampling area was divided. The variance of each strategy was defined in terms of the sample size for the systematic sequence, while the variance for the stratified random sequence was defined in terms of the sample size and in the size of the inspection zones. The comparison of variances showed that the effectiveness of the systematic sampling depended in great part on the sample size used, while the stratified random sampling was affected only when the size of the inspection zone was greater than the unit. Based on these findings the author developed a collision free path for square lattice honeycomb sampling sequences. The lattices were evaluated based on the density of the measurement points and on the connective constant (μ) that defines the number of available steps that can be performed without colliding with the workpiece. Results showed that the honeycomb lattice had a smaller connective constant than the square lattice, however the honeycomb lattice got a better density of the measurement points. In addition, it was found that the honeycomb lattice required less measurement time since it required 23% less measurements than the square lattice to measure the same area. Thus, the reason why the author proposed the honeycomb lattice as a much better sequence was due to the fact that the sequence was similar to the unaligned systematic sampling. Yet, the author recommended the use of the stratified sampling sequence since it was found to be more precise than the systematic sequence.

Lee et al. (1997) proposed feature-based methodology to apply the Hammersley, uniform, and random sampling strategies to 3-Dimensional geometric features. The author modified the procedure used to generate the Hammersley points in order to minimize the systematic error captured due to the nature of the sampling sequence. The sequences were adapted to different shapes such as circles, cones, conical frustums, and hemispheres. For each shape, the author defined two cases to distribute the data points. The first case consisted of generating the sample points based on the central point of the curves or surfaces of revolution; this case was created for those surfaces that have a good inner surface but bad edges. The second case focused on generating data points based on the edge point; the purpose of this case was to evaluate those surfaces that have uniform surface finish. The three sampling strategies and the two cases were evaluated using an experimental procedure that consisted of verifying the surface roughness of a multi-feature part. The part was virtually created and it consisted of three surfaces of revolution, a hemisphere, a cylinder, and a conical frustum. For this reason, the part was assumed to be machined using a turning process, which helped the author to reduce the dimensionality of the problem by analyzing the features using polar coordinates instead of the Cartesian coordinates. The surfaces of the three features were simulated using the Wiener process and a modified isotropic process, similar to the simulations developed by Woo et al. (1993, 1995). The sampling strategies used to verify the roughness of these surfaces were the stratified Hammersley with randomization of the first data point, the random, and the uniform distribution. The root mean square error (RMS) was used to determine the accuracy of each factor combination, where each combination of factors was replicated 100 times. The results from this experiment showed that the random sampling was not affected by the periodic variation, as the uniform sampling was. It was found that the Hammersley distribution decreased the probability of capturing systematic errors by randomizing the first data point. In general, the Hammersley distribution performed better than the uniform distribution. On the other hand, the random distribution had the same levels of accuracy as the uniform distribution. Therefore, the author recommended the stratified Hammersley distribution to evaluate the tolerances of multi-feature parts.

Narayanan Namboothiri et al. (1999) proposed a new methodology to determine an optimal sample size that can predict the form error of manufactured parts. The methodology consisted of modeling the errors of the measuring points as if they were sampled from a standard normal distribution, which allowed the author to define the errors in terms of the sample size n and two constants a_n and b_n. The three variables were used to define a mathematical model to determine the probability (P_f) that a given error is less than the predicted value. The model was verified by simulating straightness errors, using profiles that contained systematic noise, random noise, and the combination of two noises. The advantage of simulating the profiles instead of physically measuring profiles on parts was the straightforward way to determine the efficiency of the proposed model since all parameters of the profiles are known. The profiles were virtually inspected using the random sampling distribution with sample sizes ranging from 3 to 40 data points. For each sample predicted form error and the probability were estimated, while the actual form error was calculated using the least squares method. Results from this experiment showed that the mathematical model is more sensitive to the predicted from error rather than the sample size. At the same time it was found that the factor sample size had less effect on the predicted form error as the sample size increases. For all factor combinations, the predicted error was always greater than the form error obtained from the least squares method. The probability (P_f) was found to have an increasing trend for smaller sample sizes and a steadier trend for higher sample sizes. The reason that this effect was found was due to the fact that the probability of capturing a maximum error was smaller at a smaller sample size, and it started to increase as the sample size increased. For this reason, the author recommended to use this methodology in order to reduce the time of inspection, since higher values of probability will determine the chances of capturing the maximum deviation. The proposed approach seems to be very effective, however, it will be difficult to apply it in industry or in the research area since most of the time the total error of the machined parts is unknown.

Edgeworth et al. (1999) developed an iterative and adaptive sampling method to inspect the form error of machined parts. The method consisted of comparing the position and surface normal vector of the measured data points against the nominal values in order to determine the deviation of the surface of the machined part. The adaptive sampling method was executed online, meaning that the method was evaluating the accuracy of the data points at the same time that the machine was performing the measurements. An experiment was performed to verify the accuracy of the proposed method. The machined part consisted of a circular contour of 300 mm, which was inspected using 360 measurements. The data points were collected using a probing head of a coordinate measuring machine. The data was fitted to the nominal geometry after each measurement was taken. The position and surface normal vector was calculated based on the results from the fitting algorithm that were used to create an interpolation curve to determine the next point of inspection. The consequent inspection points were determined by three rules: first, the extreme points of the interpolation curve should be sampled if the curve exceeds bounding box offset; second, if the distance between sample points is greater than the maximum allowed then the next sample points should be in the middle of the two points; and third, if the distance between sample points is smaller than the minimum allowed then stop sampling. The method was more accurate than the uniform distribution since it needs fewer measurements to achieve the same level of accuracy. However, it was found that the method has drawbacks because it is very sensitive to the starting point of the sequence. In addition, it is important to notice that the algorithm provided by the author does not generate a collision free path plan, rather it only generates the coordinates of the sampling points. Thus, the functionality of this technique depends in great part on the capability of the coordinate measuring machine used.

2.8 Fitting Algorithms

The fitting algorithms play an important role in the tolerance verification of manufacturing parts, since they have the task of estimating the geometrical parameters and the zone of error of the inspected parts. The parameters are obtained by comparing the measurements against a nominal form (Choi et al., 1999), where the difference of these two represents the zone of error, which is composed of errors from the machining process and inspection process. Thus, the accuracy and robustness of these algorithms will determine whether a part is acceptable or not.

In the literature, there are two common fitting algorithms that have been used extensively, the least squares method and the minimum zone method. The least squares method is the most used method in the metrology area, due to its simplicity and to the efficient computational time required to analyze the data. This method is used to estimate the regression coefficients by minimizing the sum of squares of error in a multiple linear regression method (Montgomery, 2001; Choi et al., 1999). However, results provided by the least squares method are not very precise since it is insensitive to outliers, which means that it tends to overestimate the zone of error (Orady et al., 2000). For this reason, the ANSI and ISO standards have adopted the minimum zone as a method to evaluate the geometric tolerances of features. The minimum zone method looks for fitting a set of measurements into a predetermined tolerance zone, thus if all points are enclosed by this zone then the part is regarded as a good part. The minimum zone is also known as a minimax fit and has the objective of minimizing the maximum deviation, which provides a tighter fit than the least squares method. For this reason, the least squares method will generate type I errors (rejects good parts) since it focuses on finding a centerline that minimizes the objective function, while the minimum zone will generate type II errors (accepts bad parts) since it focuses on minimizing the maximum error between points and a fitted line (Choi et al., 1999; Huang, 1999).

Prakasvudhisarn (2002) and Kethara Pasupathy et al. (2003) provided two different classifications for the tolerance zone methods. Prakasvudhisarn (2002) classified the minimum zone methods into two categories, computational geometry approaches and numerical approaches. The computational geometry approaches are computationally efficient and are used to estimate the error of basic geometric features; some examples of this approach are convex hull, eigen-polygon, and Voronoi diagram. On the other hand, the numerical approaches use linear and nonlinear optimization methods to evaluate geometric forms. A disadvantage of these methods is that they require a significant amount of computational time to analyze large data sets. Some examples of this approach are the genetic algorithms, and neural networks. Kethara Pasupathy et al. (2003) proposed a classification of these methods based on the mathematical approaches used: offsets method (Minkowski operation, and conditional tolerance zone from half-spaces), parametric space method, algebraic methods, homogenous transformation methods, user-defined offset zone by parametric curves, and statistical analyses such as the process capability index. Although the two classifications showed the methodologies used in the literature to calculate the tolerance zone of manufactured parts, the one proposed by

Prakasvudhisarn (2002) contains some of the most widely used algorithms in the literature, as is explained below.

Dowling et al. (1995) compared the orthogonal least squares against the minimum zone approach to estimate straightness and flatness errors. Huang (1999) used the 3D Voronoi diagram to calculate the zone of error of spherical surfaces. Prakasvudhisarn (2002) applied the linear least squares, nonlinear least squares, linear minimum zone, and nonlinear minimum zone to study the geometric tolerances of cones and conical frustums. Aguirre Cruz (2003) compared the performance of the linear least squares against the linear minimum zone on the tolerance verification of torus forms. Malyscheff et al. (2002) defined the use of support vector machines (SVMs) for representing the minimum zone problem, while Prakasvudhisarn et al. (2003), and Balakrishna (2005) used the support vector regression (SVR) to analyze the form tolerance of flat and spherical surfaces, respectively. Support vector regression (SVR) is a tool that was developed as a part of statistical learning theory, which has not been extensively applied in the metrology area to verify form tolerances, as mentioned before. One big advantage of the SVM algorithms is that the data does not have to follow the normality assumption, which makes these algorithms very robust (Prakasvudhisarn et al., 2003). For this reason, the proposed support system will compare the performance of the support vector regression against the minimum zone method. The former will be applied for the first time to analyze the tolerance zone of conical and torus forms.

The 3D Voronoi diagrams applied by Huang (1999) are defined as a plane that split the Euclidian distance of two measured points $A(x_a, y_a)$ and $B(x_b, y_b)$ into two equal spaces, in which all points on the plane are equidistant to the two measured points. One side of the two spaces that does not contain *A* is defined as the farthest region *F*(*A*), while the space containing *A* is defined as the nearest region *N*(*A*), thus the farthest region of *A* is the nearest region of the point *B* labeled *N*(*B*), and the nearest region of *A* is the farthest region of point *B* labeled *F*(*B*). The approach can be extended to multiple points since it can be used to generate *n* number of half spaces according to the number of paired points available, where the farthest and nearest regions are created by the intersections of the half spaces.

The relation between the sphericity problem and the 3D Voronoi diagrams is the existence of a "*max region*" that contains two control points that define the minimum zone of spheres, since these points determine the radial distance between the inscribed and circumscribed spheres. The region contains two types of vertices that are defined as Y-vertex and X-vertex, which are the extreme points of this region. The Y-vertex created before the existence of the "*max region*" is defined by the four either nearest of farthest regions. The X-vertex was defined by five measuring points, which describe the intersection between a Voronoi diagram and an edge, where two measured points defined the diagram and three points defined the edge. Huang (1999) defined two lemmas with the purpose of demonstrating that the exact minimum zone value was obtained by using the Voronoi diagrams and the information obtained from the "*max region.*" The first lemma provided evidence that the X-vertex contained within the "*max region*" is the only one that can minimize the radial separation center for the inscribed and circumscribed spheres. The author proved that the Y-vertex cannot provide the minimum radial center since it was found that the Y-vertex on the nearest region can provide values of the objective function that are less or equal in magnitude to those generated by the Y-vertex in the “*max region.*” Thus, the X-vertex was the only vertex in the region that provided the smallest radial separation center. The second lemma focused on verifying that the sphericity value for a given set of points is always smaller than the sphericity value calculated by including new point to the set. The reason that the zone of error of the sphere cannot change is due to the two control points inside the “*max region*,” which determine the zone of error. If two points from a known set provide the smallest maximum separation between the inscribed and circumscribed spheres, then any new point cannot decrease the size of the zone of error. However, any of the control points can be replaced, if and only if the new added points cause an increase in value to the zone of error of the sphere.

Based on these statements, Huang (1999) developed an automated procedure to calculate the minimum zone of spheres. The steps followed during this procedure were to identify five points among the set of data points for the X-vertex calculation. Then, the inscribed and circumscribed spheres were calculated with the purpose of determining if all points on the set were enclosed by the two spheres. If all points were enclosed then the procedure ends, however, if some points were outside the spheres, then some of five points selected for the X-vertex calculation were replaced by these points outside the spheres, and then the procedure continues until all points are enclosed between the spheres. The procedure was verified under two similar conditions. First, a set of five points was used to determine the X-vertex that generated the minimum radial separation. The five points; coordinates are (4.8, 6.4, 6.0), (0, 0, -10), (0, 9, 0), (0, -9, 0), and (0, -9, 0). Among all possible vertices generated from the set of five points, the vertex (0, 0, 0) provided the minimum radial separation equal to 1. In addition, it was found that points 1, 2, and 3 described the position of the circumscribed sphere with radii of ten, while points 4 and 5 determined the position of the inscribed sphere with radii of nine.

The solution of this procedure was used to determine the efficiency of the algorithm under a different condition. The five points were hidden into a set of ninety-five points, making the total sample of points equal to one hundred. The data points are not shown in this chapter, but they will be used to verify the new sphere model proposed in this research. The five points used in the previous experiment were randomly hidden within the data. The new places of these points were 51, 62, 70, 79, and 95. The one hundred points were evaluated in set of five points, and it was found that the algorithm needed five updates to locate the five known points. The results were identical to the ones obtained from the first experiment, proving the efficiency and accuracy of the procedure. In addition, the author created several scenarios in which he varied the number of evaluated points ranging from 10 to 100, with the purpose of determining the total number of updates needed to calculate the sphericity, as well as to calculate the total number of points that were on the final set. Results for 10 points showed that the average number of updates needed were 2.89 updates, while the average number of points on the final set was 7.81 points. Results for 100 points showed that the average number of updates needed were 6.09 updates, while the average number of points on the final set was 14.12 points. Although, the 3D Voronoi approach looks appealing, it leaves room for questioning its performance, since the data used were simulated and not measured on a coordinate measuring machine. It will be interesting to evaluate its performance using real data, where the maximum deviations are unknown. In addition, the approach will be useful under the assumption that an efficient sampling strategy was used to collect the data Moreover if this approach is used online then the time needed to find the control points will increase into regions not accepted by industrial standards.

In summary, it can be said that the minimum zone method outperformed the least squares method, due to the reasons explained above. However, in a few cases the least squares method performs better than the minimum zone method, which is due to the fact that the minimum zone method does not provide the global optima as a solution, instead the algorithm gets trapped in a local minima solution.

CHAPTER 3

DECISION SUPPORT SYSTEM TO EVALUATE FORM TOLERANCES OF SURFACES OF REVOLUTION

3 Summary

The proposed decision support system is composed of five major phases. Some have been already studied in the literature and are improved here, while others are newly developed, taking into account the weaknesses of existing procedures. The system is created with the purpose of offering a tool to verify the form tolerances of rotational parts. The system begins by extracting information from a CAD file provided by the user. The file contains the geometric model of the part based on the boundary representation, which includes dimensions, form characteristics, position, and orientation of the part. Then, the obtained information is used to determine the location and amount of data points needed to determine the form tolerances of the inspected part. Once the location and amount of points are obtained, then the sampling points are generated according to the requirements of the software of the CMM. The data are loaded in the CMM computer and the points collected using the numerical control mode of the CMM. The measured point coordinates are sorted and compared against an ideal form feature to determine the form tolerance of the desired feature in the inspected part. Two different analyses are performed on the measuring data; one that compares the measured data points against and ideal form; and the other that extracts the parameters of the feature from the measuring data. The residuals, the parameters, and the magnitude of the zone of error of both analyses are compared in order to determine if the part was created according to specifications. The pattern of the residual is stored in the computer, taking into account the manufacturing process, machine, material, and form feature used to create the part, in order to create a knowledge-base that can be used in the future to generate adaptive sampling strategies for reducing inspection time and increasing sampling efficiency. The information is presented to the user in a suitable manner to facilitate decision making.

The five phases of the proposed decision support system are:

Phase I – Feature Recognition

Phase II – Path Planning

Phase III – Data Collection

Phase IV – Form Verification

Phase V – Information Analysis

The first phase Feature Recognition is explained in Chapter 4, while Path Planning and Data Collection are explained in Chapter 5. These three phases are standard procedures used in the CAD/CAM areas to machine and inspect manufactured parts. Although these procedures have been studied in the past, two programs are developed for cleaning and for extracting the form features from the CAD file. However, the contributions that can be claimed from the first three phases are minimal, since these procedures have been well-researched in the past. The remaining two phases explained in chapters 6 and 7 represent the main contributions that this research adds to the science and engineering content in metrology. Although, the phases are explained in the following chapters, a brief summary of each phase is presented below.

3.1 Phase I – Feature Recognition

In this phase, the user provides a CAD file that contains the design of the part that is inspected. The systems supports CAD files generated in some of the most common modeling software, such as AutoCAD^{TM}, ProE^{TM}, and Solidworks^{TM}. The algorithm behind this phase has the purpose of extracting all the information about the part from the CAD file. The information stored in the CAD files can vary and depend in great part on the type of file used. There are several exchange data files that provide different types of information, as mentioned in Chapter 2. In this research, the ACIS file was chosen as the exchange data file, for the reason that it stores the part information using boundary representation modeling and word-based syntaxes. The code of the ACIS file is cleaned and rearranged in the form of a matrix, which simplifies the feature recognition process Thus, the data matrix is used to recognize the types of features, the dimensions, the location, and orientation of the part. Finally, the algorithm saves the information in a series of matrices that are used in the next phases to verify the form tolerances of all features. Figure 16 shows a flowchart of the first phase of the support system, explained below.

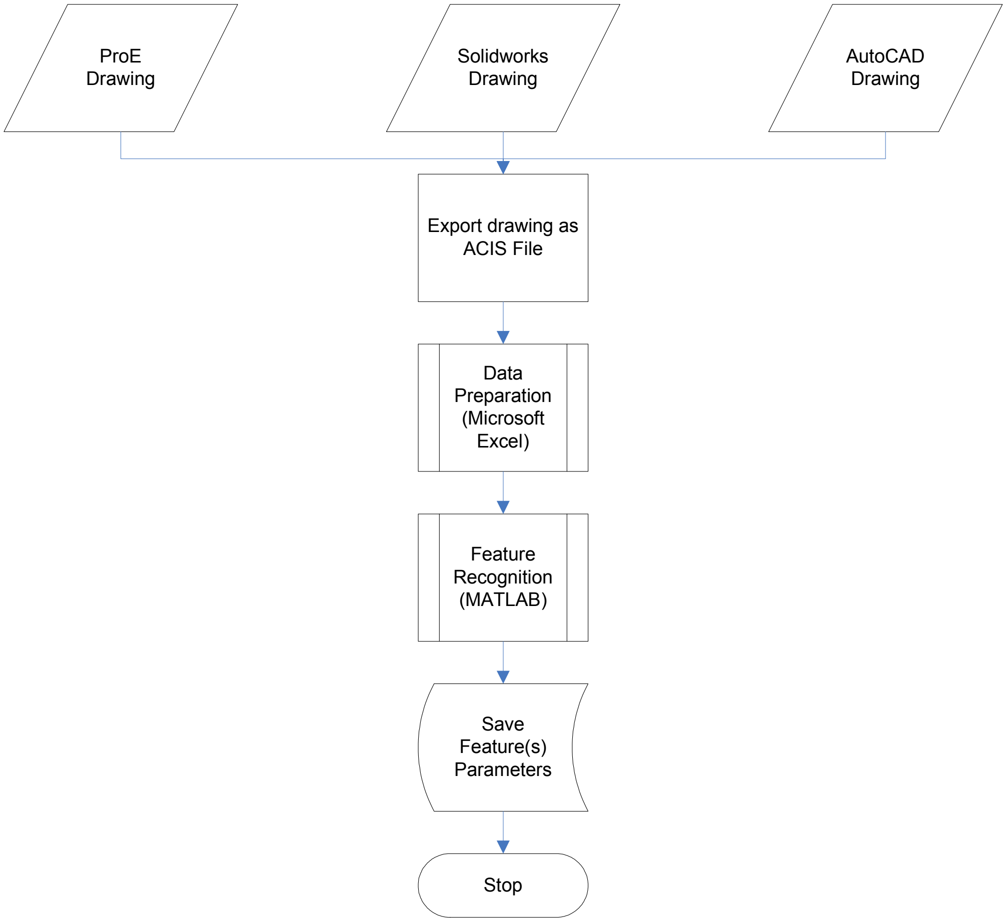

ProE Drawing
Solidworks Drawing
AutoCAD Drawing
Export drawing as ACIS File
Data Preparation (Microsoft Excel)
Feature Recognition (MATLAB)
Save Feature(s) Parameters
Stop

Figure 16. Flow diagram of the feature recognition block.

3.1.1 ACIS File

The decision support system begins by reading the ACIS file (*.sat) obtained by exporting the solid model from either AutoCAD^{TM}, Pro-Engineering^{TM}, or Solidworks^{TM} programs. The proposed system accepts an ACIS file exported as text using a version no older than ACIS V12.0. However, if a previous version of this file is provided, then the system will not be able to determine the form features, dimension, location and orientation of the part. An example of a cone, frustum, cylinder, sphere, and torus model exported as an ACIS file can be seen in the Appendix A.

3.1.2 Data Preparation

The data preparation block has the goal of transforming the ACIS file into a matrix composed of codes that represent the form features and its properties. The structure of the ACIS file can vary according to the number of features and to the orientation of the part in the virtual world of the modeling program. In addition, the file contains many standard lines that are generated when exporting the solid model as a *.sat file; however some of those lines do not provide any information about the features, and they could increase the processing time of the feature recognition program if they are left in the data matrix, which is why they are deleted. The remaining lines are efficiently rearranged so that the feature recognition program can read the information and extract all details from each line in a faster rate, as is shown in Figure 17.

Finally, after transforming all the lines, the program adds an extra column at the right side that contains all the key information that later on is used to speed up the feature recognition process. The key information is composed of fourteen cells that contain the information from all geometrical/topological element counters, such as number of lines, faces, loops, planes, spheres, cones, torus, coedges, edges, vertexes, straight curves, ellipse curves, points, and shells. Then, the program saves the data matrix as a text file (*.txt) called ACIS in a location specified by the user.

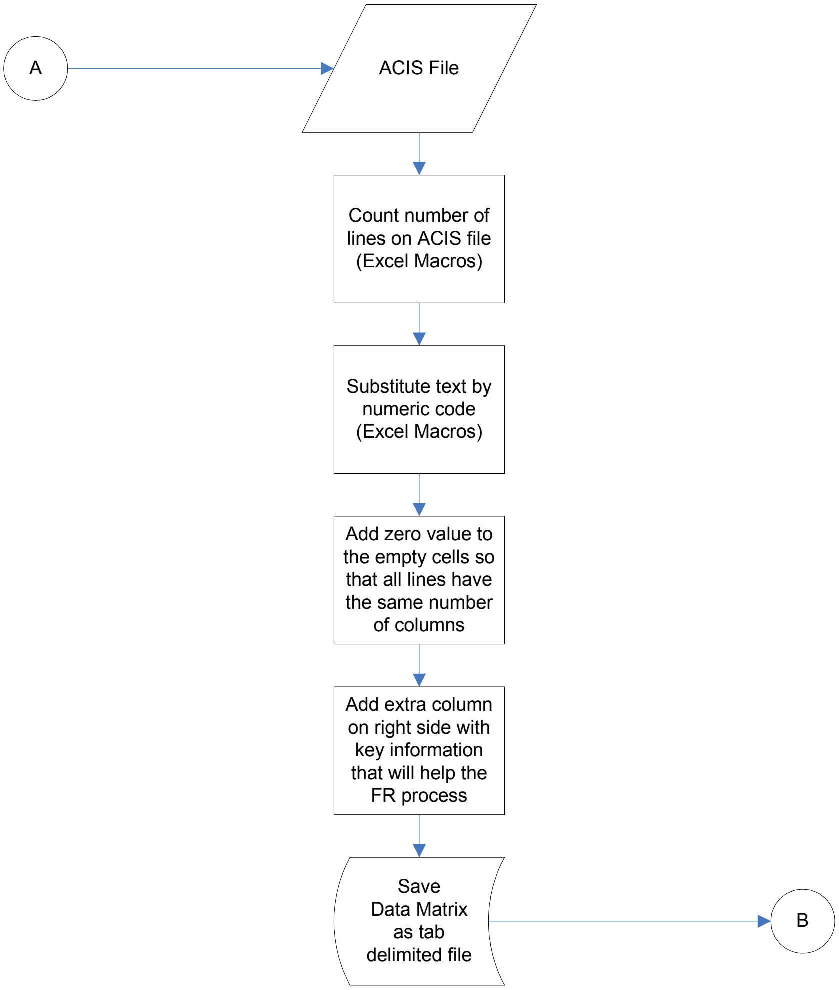

A
ACIS File
Count number of lines on ACIS file (Excel Macros)
Substitute text by numeric code (Excel Macros)
Add zero value to the empty cells so that all lines have the same number of columns
Add extra column on right side with key information that will help the FR process
Save Data Matrix as tab delimited file
B

Figure 17. Flowchart of the data preparation block

3.1.3 Feature Recognition

The feature recognition block receives as input the data matrix developed in the data preparation block. The feature recognition algorithm extracts the feature(s) from the file by reading the last column of the data matrix, which contains the key information about the total number of lines and the number of topological and geometrical entities. Based on this information, the program recognizes whether the file contains a single feature or a multi-feature part. For single feature parts, the program reads the file and extracts the information in a matrix form for each of the topological and geometrical entities. Then the program determines the position and orientation of the feature. In the case that the user needs to recognize a single feature but the data matrix contains more than one feature, then the program focuses its resources to determine the position and orientation of that specific feature within the multi-feature object. For multi-feature objects, the program sorts the vertexes and point coordinates from high to low, and it relates the coordinate points to the type of surface that they belong to. Finally, a matrix containing a summary of all the recognized features including their dimension, position, and orientation, as well as the data matrix and the matrices of the topological and geometrical entities is stored in memory. The matrices are used later on either to generate the sampling points for form tolerance verification, or to extract the features from the measured point coordinates. Figure 18. shows the flow diagram of the feature recognition block.

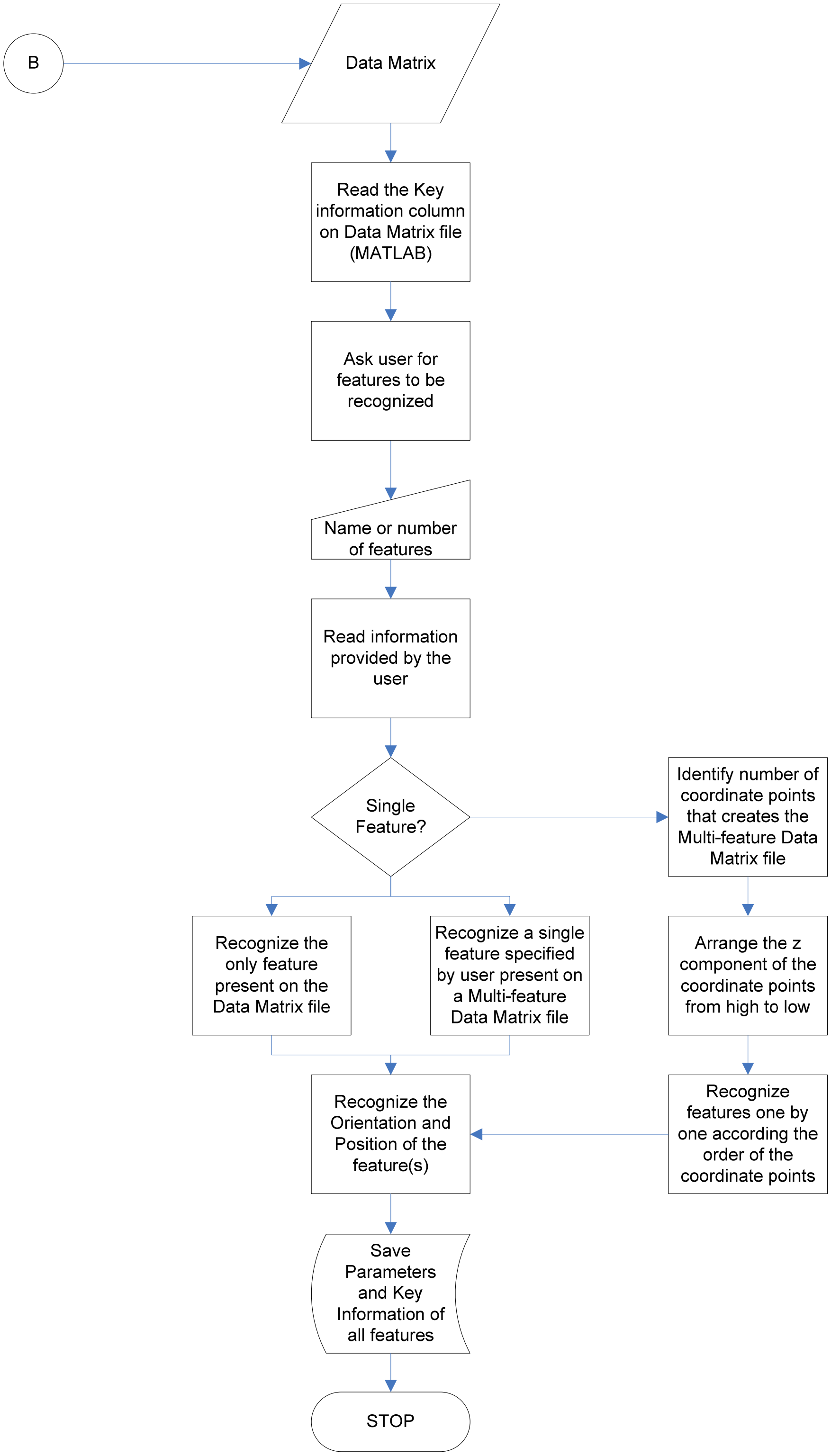

B
Data Matrix
Read the Key information column on Data Matrix file (MATLAB)
Ask user for features to be recognized
Name or number of features
Read information provided by the user
Single Feature?
Identify number of coordinate points that creates the Multi-feature Data Matrix file
Recognize the only feature present on the Data Matrix file
Recognize a single feature specified by user present on a Multi-feature Data Matrix file
Arrange the z component of the coordinate points from high to low
Recognize the Orientation and Position of the feature(s)
Recognize features one by one according the order of the coordinate points
Save Parameters and Key Information of all features
STOP

Figure 18. Flowchart of the feature recognition process.

3.2 Phase II – Path Planning

The path-planning phase has the purpose of determining the way in which the part is inspected. The input for this phase are the three matrices generated in the feature recognition phase. The first, which is the most important, holds the summary of all the extracted features and their properties. The second is the matrices of the topological and geometrical entities obtained from phase I. And third is the data matrix that contains a summary information from the part. The process includes establishing the sample size as well as the location of the sampling points. Several studies have been done in the literature to determine, which sampling strategy yields the best information from the surface of the part. However, it is well known that the manufacturing process contains an infinite number of variables, in which the same feature forms can be produced in many different ways. Therefore, it will be erroneous to inspect a part based only on the form of the object. For this reason, it is important to understand the way in which the parts were created including the manufacturing parameters. By doing this, the user can related the defects of the part with the common problems caused by that particular manufacturing process. For example, if the part was created using the turning operation, then more focus is given to the highest part of the object as well as to the sides in order to detect the accuracy of the axis of rotation.

The support system follows the Hammersley sampling strategy as a baseline to determine the distribution of the points across the part surface. Hammersley was chosen because it is one the most used strategies in the literature, and has a good point distribution. The system also generates data points based on two different sampling strategies: random and aligned systematic.

3.3 Phase III – Data Collection

The data collection phase employs the CMM to measure the point coordinates created in the path-planning phase. The points are measured automatically using the numeric control function of the CMM. The data points are collected in a random manner or by following a strategy set by the user, which could be either based on descending or ascending height.

In addition, the support system provides a set of rules that the user needs to follow in order to set the part on the measuring table, as well as the necessary guidelines to translate the location of the origin of the CMM to the location of the geometrical origin of the part.

3.4 Phase IV – Form Verification

The form verification phase has the purpose of analyzing the data points obtained from the data collection process. The data points are analyzed using new developed mathematical models that define the shape of the inspected form, such as spheres, cylinders, cones, frustums, and torus. The measurements are analyzed according to the number of inspected features. For single features, the data points are analyzed in a conventional way, where the data points are plugged into the form verification models and the parameters of the feature are calculated. On the other hand, for parts that have multiple features, the points are separated based on the inspected features. Then, each of the features are analyzed individually in order to obtain all the features parameters and see whether there is a correlation on the form error between features. At the same time, the parts with multiple features are analyzed as one entity; this means that all features are regarded as a cylindrical form, assuming that the part was created using the turning process. This step allows to find out whether the axis of rotation had any influence on the form of error, and to determine the amount of error that each feature contributes to the total zone of error. Figure 19 shows the flowchart for individual and multiple feature form verification.

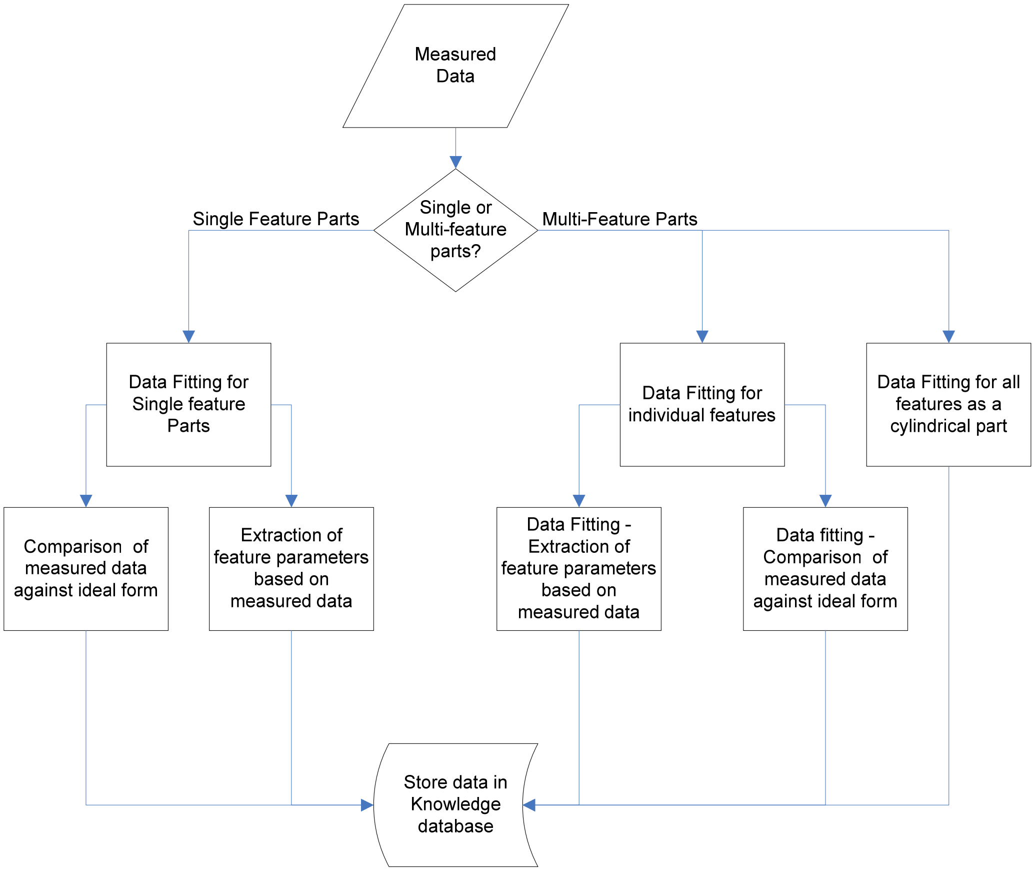

Measured Data
Single or Multi-feature parts?
Single Feature Parts
Multi-Feature Parts
Data Fitting for Single feature Parts
Data Fitting for individual features
Data Fitting for all features as a cylindrical part
Comparison of measured data against ideal form
Extraction of feature parameters based on measured data
Data Fitting - Extraction of feature parameters based on measured data
Data fitting - Comparison of measured data against ideal form
Store data in Knowledge database

Figure 19. Flowchart of individual and multiple feature form verification.

3.5 Phase V – Information Analysis

Finally, the information analysis phase presents a set of graphs and charts that help the user to identify the reasons why the part was either under specifications or out of tolerance. The goal of the support system is to help the user identify the regions of each form that contribute to the zone of error, as well as to identify if the problem was due to the setup of the inspection process or due to the tools or materials of the manufacturing process. By doing this, the system will aid the user to recognize patterns and trends that occurs when using different sets of inspection and machining parameters.

The following Chapters are organized according to the order of the phases of the decision support system, starting with Chapter 4 explains the feature recognition process. Chapter 5 focuses on the sampling strategies and path planning. Chapter 6 presents the approaches and mathematical models developed in this research. Chapter 7 is related to the experimental methodology needed to verify the performance of the proposed mathematical models. Chapter 8 shows the results of the experiments and Chapter 9 gives complete details about the contributions, conclusions, recommendations, and future work.

CHAPTER 4

FEATURE RECOGNITION

4 Summary

The feature recognition algorithms focus on extracting information from a set of data, based on patterns and rules stored on knowledge based systems. Computational advances and higher quality standards revolutionized the design and manufacturing areas, in which the products were conceptualized and materialized under processes that were controlled by computers. This caused the development of new systems capable of ensuring the quality of the parts, such as the computer aided design (CAD) and computer aided manufacturing (CAM) systems. The exchange of information between the two systems was based on feature recognition algorithms due to the fact that the features used in the design (lines, arcs, etc.) were different from the features used in the manufacturing area (chamfers, holes, etc.). Thus, the feature recognition algorithms provided the link needed to simplify the task. Furthermore, these algorithms have also been implemented in the inspection area to verify the form tolerances of manufactured parts, such as is the case of the present research.

4.1 Feature Recognition

The CAD/CAM process involves a set of steps needed for creating a manufactured part: the first step is the conceptualization of the idea, that is, an area where the idea becomes a design by virtually modeling it using a computer program. Then, the design becomes a reality when the virtual model is electronically transferred to the processing unit of the computer numerical control machine that will translate the model into a set of instructions needed to physically manufacture the part. Finally, the virtual model is also imported into the processing unit of the inspection machine that recognizes the feature(s) and its dimensions in order to create a path plan that is used to inspect the manufacturing tolerances of the machined part (Zeid, 1991). Thus, the communication and transfer of information between the design, manufacturing, and inspection systems is very important because any misinterpretation of the information will cause the part to be out of tolerances. For this reason the feature recognition algorithms have been incorporated into the design and manufacturing areas with the purpose of avoiding misinterpretations of information and connecting the design and manufacture modules (Owusu Ofori et al., 1990).

The CAD programs such as AutoCAD^{TM}, ProE^{TM}, and Solidworks^{TM}, are solid modeling software that use different types of data exchange files to transfer the information to the CAM modules. Among the exchange files there are three that were found in the literature to be the common data exchange files: IGES, STEP, and ACIS. These files use a solid model representation to convert the designed part into categorized geometrical and topological entities that can be easily transferred and recognized by other machines (Coulibaly et al., 1998). Although these files provide enough information about the features contained in the design, Case et al. (1993) found that more information is required in order to make the model suitable for the whole CAD/CAM process, such as tolerances, surface finish, materials, and tool access directions. Additionally, Maendl (2003) found the cycle time of the product is reduced if the file has the right information.

Therefore, it is important to choose the right exchange file that has the right information needed for that specific application.

The goal of the present research is to develop a feature recognition system that can extract the features, dimensions, and orientation from data exchange files in order to verify the manufacturing tolerances of machined parts. Among all the data exchange files surveyed, the ACIS file was chosen for this research, due its word-base syntaxes and capability to define the parts in terms of their boundaries. In addition, the feature recognition system developed for this research requires information about dimensions and orientation of the part, which is similar to the information provided by the CAD programs, since these use geometry as their official language (Jha, 1995). Hence, there is no need to translate the ACIS file; instead the feature recognition program needs to recognize the relationship between geometrical and topological entities to determine the type of features that are contained in the virtual design.

4.2 ACIS

ACIS is a geometric modeler that uses the boundary representation format to model the shapes of parts. The reconstruction of these parts is created by using a set of curves and surfaces that are interconnected to each other, in which the inside and outside of the curves and surfaces are distinguished by the modeler (Corney et al., 2001). The ACIS file can be obtained from a solid modeler by exporting the solid part as text file (*.sat) or as binary file (*.sab). The former being the one used in the feature recognition program of the present research.

The ACIS code uses geometrical and topological entities to process at a fast rate the mathematics involved in the representation of the solids. The geometric entities are used to define the graphic representation of the part (Coulibaly et al., 1998), while the topological entities define the way in which the geometrical entities are interconnected (Corney et al., 2001). The solid modeling program uses three geometrical entities, points, curves, and surfaces, and three topological entities such as faces, edges, and vertices. However, the ACIS modeler uses more topological entities, coedges, loops, shells, lumps, and bodies, in order to increase the processing speed of the information (Corney et al., 2001).

A solid part is composed of one or many form features, which vary according to the complexity of the part. Based on those features, the ACIS modeler arranges the entities into categories according to their hierarchy, with the highest topological entity called body. The body represents the complete solid part, which can be either a single feature or multi-feature part. At the same time, the body is composed by a set of lumps that are broken down into sets of shells. The lumps and shells are also topological entities that are contained in solid parts, which vary in number according to the amount of features in the part. The shells are defined by a set of topological entities called the faces. The faces are constructed by surfaces and by loops. The surfaces are geometrical entities that define the type of the feature contained in the part, such as cone, plane, spline, sphere, and torus, while the loops are a topological entity that represent the boundary of the faces and are composed by a chain of edges and coedges. The difference between the edges and the coedges is that the edges are entities that are shared by multiple faces, while the coedges are unique for each face (Corney et al., 2001). In addition, the edges are defined by a geometrical entity called curves that can be of two types, either an ellipse or a straight curve. Finally, each edge/curve is bounded at the end points by the topological entity vertex and by the geometrical entity point.

4.2.1 ACIS Example

The following lines show an example of a conical form that was created using Solidworks^{TM}. The example explains the way in which the solid modeler ACIS stores the solid part in a *.sat file and the way in which the data is interpreted from the file. The cone presented in this example is shown in Figure 20 and has a height (h) of 60, a radius of the base (r) of 30, an angle (γ) equal to 26.57 degrees, a vertex located at the origin (0, 0, 0), and an axis of revolution in the same orientation as the z-axis.

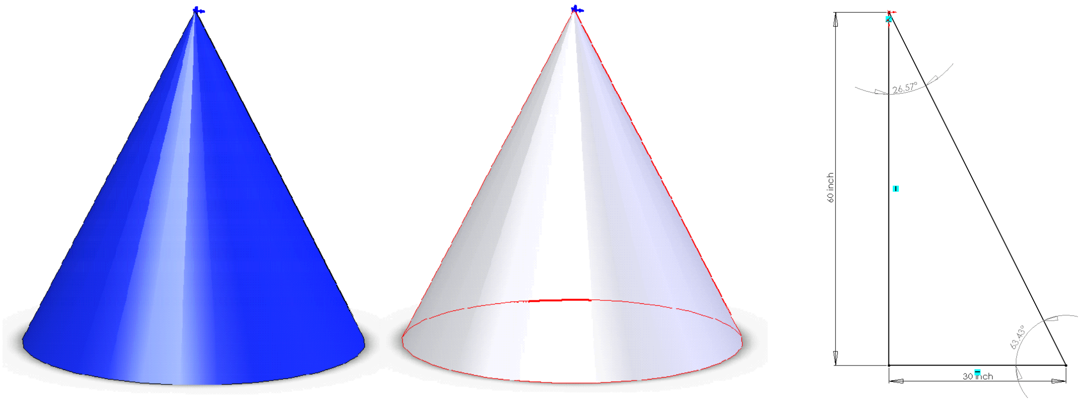


Figure 20. Conical form created in Solidworks^{TM}.

The code shown in the following lines was partially cleaned. This means that the decimals were reduced to 5 digits because the complete file contains a precision of at least 16 decimals. However, the complete code used in this example is found at the end of this dissertation in Appendix A. In order to simplify the explanation, the following example uses color codes to represent the topological and geometrical entities. It begins by explaining the general lines of the code and then goes from the lower level entities to the higher level entities.

The first three lines of the code contain general information about the file, such as the program used to generate the code, the type of file, either *.sat or *.sab, the version of the ACIS modeler, and the date in which the file was created, among others, while the last line of the code contains a tag named "End-of-ACIS-data."

```
1400 0 1 2
37 SolidWorks(2005143)-Sat-Convertor-2.0 12 ACIS 14.0 NT 24 Wed Feb 07 19:25:07 2007
1 9.9999999999999995e-007 1e-010
⋮
End-of-ACIS-data
```

The middle part of the code contains the representation of the part in terms of the geometrical and topological entities. The amount of information of each entity depends on the version of the ACIS modeler. The final part has the lower level entities, such as points, vertices, curves, and edges, which are arranged in no particular order. Most of the modelers enumerate each of the lines using a dash followed by the number, but there are some versions of ACIS that do not provide this information. Each line is terminated by the symbol (#), and the entities are called by other entities using the symbol ($) before enumerating the desired entity.

The lines 1 and 3 shown below, are common to all ACIS files. The only difference is that these lines represent the attributes of the solid model. The attributes shown in line 1 have an effect on the entity in line 0, which corresponds to the body that is the highest topological entity. Another attribute is the color, which is seen in the last three numbers in line 3. The numbers represent the RGB color system used by AutoCAD^{TM}, ProE^{TM}, and Solidworks^{TM} to render the faces of the solid. In this example, the three colors are Red (0.09803 or 25), Green (0.18431 or 47), and Blue (0.91764 or 234), in scales from 0 to 1 and from 0 to 255, respectively.

```
-0 body $1 -1 -1 $-1 $2 $-1 $-1 F #
-1 name_attrib-gen-attrib $-1 -1 $-1 $-1 $0 keep keep_kept ignore copy @8 Cone #
-3 rgb_color-st-attrib $-1 -1 $-1 $-1 $2 0.09803 0.18431 0.91764 #
```

The points and vertices are the lower level entities, and normally they are found on the last lines of the ACIS file. However, the code builder arranges the middle body of the code in no particular order. Table 5 shows the points and vertices needed to construct the conical solid model. The three lines on the left represent the number of points, where the last three columns of each line describe the location of each point. For example, point 32 is located on the coordinate (0, 0, 0). On the other hand, the vertices are located on three lines on the right, which are topological entities that are related to each one of the points. The last two columns of each vertex line are pointing to an edge, which is a higher order topological entity, and to a point, which defines the vertices, respectively.

Table 5. Three points and three vertices used to create a conical solid model.

Points (Geometrical Entities)	Vertices (Topological Entities)
-32 point $-1 -1 -1 $-1 0 0 0 #	-25 vertex $-1 -1 -1 $-1 $22 $32 #
-33 point $-1 -1 -1 $-1 –30 0 -60 #	-26 vertex $-1 -1 -1 $-1 $28 $33 #
-35 point $-1 -1 -1 $-1 30 0 -60 #	-29 vertex $-1 -1 -1 $-1 $28 $35 #

From Table 6, it can be seen that the conical form is composed by four edges. Each of these lines has four pointers: the first two refer to the vertices that create that edge; the third pointer refers to the coedge associated with that edge; and the fourth pointer refers to the data in Table 7 that defines the type of curve used to construct the edge, either straight or ellipse. For example, the edge described in line 19 is constructed by vertices 25 and 26, has a coedge number 18, and the pointer 27 defines the edge as a straight curve. Figure 21 shows the point, vertex, edge, and curve entities that create the cone, where the geometrical and topological entities are outside and inside the parentheses, respectively.

Table 6. Four edges of a conical form.

Edges (Topological Entities)
-19 **edge** $-1 -1 –1 $-1 $**25** 0 $**26** 0.067082 $**18** $**27** forward @7 unknown F #
-22 **edge** $-1 -1 –1 $-1 $**25** 0 $**29** 0.067082 $**16** $**30** forward @7 unknown F #
-24 **edge** $-1 -1 –1 $-1 $**29** 0 $**26** 3.14159 $**17** $**31** forward @7 unknown F #
-28 **edge** $-1 -1 –1 $-1 $**26** -3.14159 $**29** 0 $**21** $**34** forward @7 unknown F #

Table 7. Four curves that define the type of edge.

Curves (Geometrical Entities)
-27 **straight-curve** $-1 -1 -1 $-1 0 0 0 -447.21359 0 -894.42719 I I #
-30 **straight-curve** $-1 -1 -1 $-1 0 0 0 447.21359 0 -894.42719 I I #
-31 **ellipse-curve** $-1 -1 -1 $-1 0 0 -60 0 0 1 30 0 0 1 I I #
-34 **ellipse-curve** $-1 -1 -1 $-1 0 0 -60 0 0 1 30 0 0 1 I I #

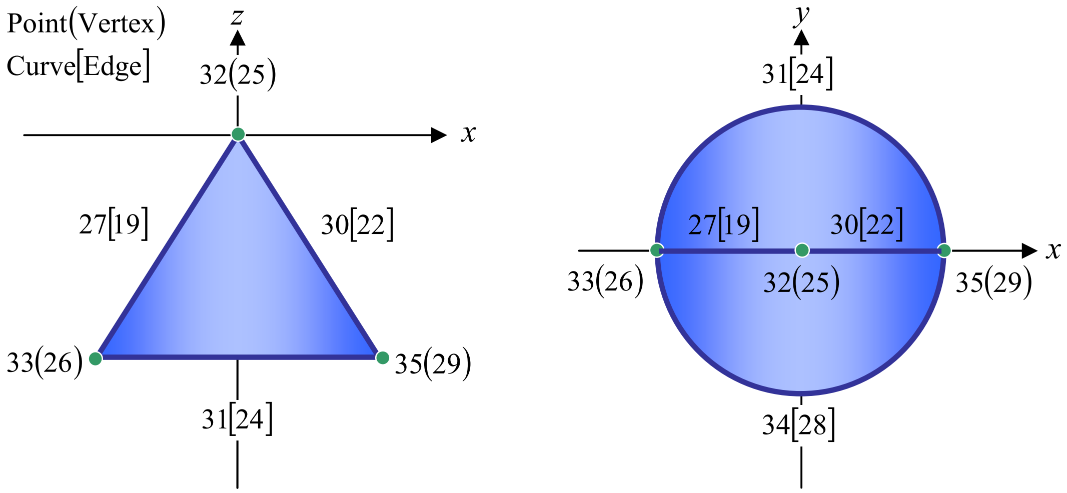


Figure 21. Position of points, vertices, edges, curves of a conical form.

The coedges are topological entities that are associated with each edge and are unique to each face. Table 8 shows the eight coedges that create the conical form. Each coedge line is defined by five pointers: the first three are referring to the next, the previous, and the partner coedges, the fourth pointer refers to the edge where the coedge is located, and the fifth pointer refers to the loop created by that coedge. For example, the coedge in line 12 has as next coedge the number 16, as previous coedge the number 17, and as partner coedge the number 18. This coedge is in the same position as edge 19, and it is part of loop number 7.

Coedge 12 can be seen in the form of a white arrow at the top view of Figure 22. The direction of the arrow determines the next coedge, that is, number 16, and the previous coedge, that is, number 17, which together create loop number 7. Coedge 12 is located in the same position as edge 19, and it is in front of coedge 18, which makes them partner coedges. This means that coedge 12 and coedge 18 share the same edge but do not belong to the same face.

Table 8. Coedges of the conical form.

Coedges (Topological Entities)
-12 **coedge** $-1 –1 -1 $-1 $**16** $**17** $**18** $**19** reversed $**7** $-1 #
-15 **coedge** $-1 –1 -1 $-1 $**18** $**21** $**16** $**22** reversed $**10** $-1 #
-16 **coedge** $-1 –1 -1 $-1 $**17** $**12** $**15** $**22** forward $**7** $-1 #
-17 **coedge** $-1 –1 -1 $-1 $**12** $**16** $**23** $**24** forward $**7** $-1 #
-18 **coedge** $-1 –1 -1 $-1 $**21** $**15** $**12** $**19** forward $**10** $-1 #
-20 **coedge** $-1 –1 -1 $-1 $**23** $**23** $**21** $**28** reversed $**13** $-1 #
-21 **coedge** $-1 –1 -1 $-1 $**15** $**18** $**20** $**28** forward $**10** $-1 #
-23 **coedge** $-1 –1 -1 $-1 $**20** $**20** $**17** $**24** reversed $**13** $-1 #

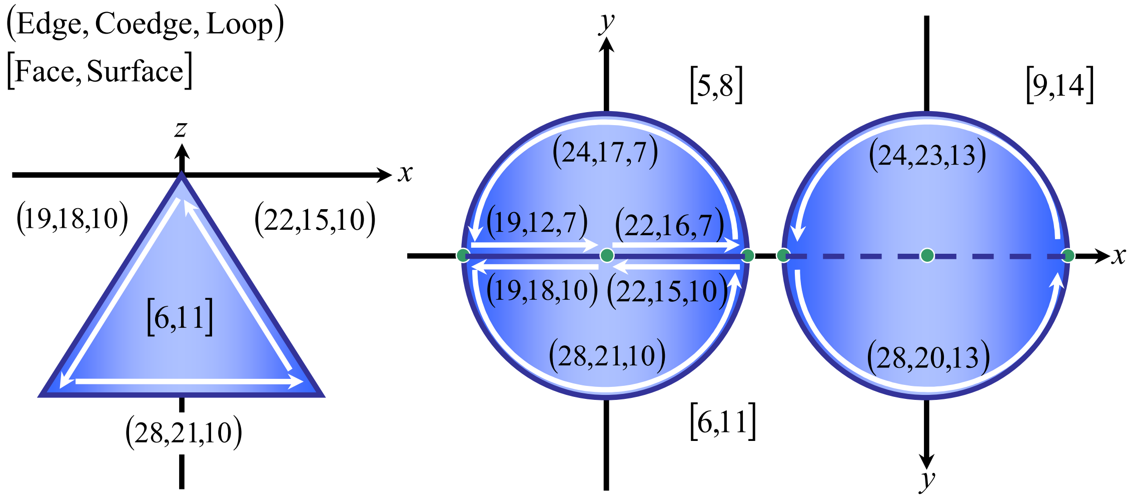


Figure 22. Front, top, and bottom views of the entities of the conical form.

The loops are created by a set of coedges, which create a face. The conical form has three loops and thus three faces and three surfaces, seen in Tables 9 to 11 respectively. The loops have two pointers, one pointing to the first coedge on the loop and the other refers to the face created by that loop. From Figure 22, it can be seen that the cone has three faces: two are half conical faces, while the third face represents the base of the cone. The faces have four pointers: the first pointer refers to the next face on the shell, the second pointer refers to a loop, the third is pointing to the shell, and the fourth determines the surface that creates that face.

Table 9. Loops created by the conical form.

Loops (Topological Entities)
--7 **loop** $-1 -1 –1 $-1 $-1 $**12** $**5** F unknown #
-10 **loop** $-1 -1 -1 $-1 $-1 $**15** $**6** F unknown #
-13 **loop** $-1 -1 -1 $-1 $-1 $**20** $**9** F unknown #

Table 10. Three faces of the conical form.

Faces (Topological Entities)
-5 **face** $-1 -1 -1 $-1 $**6** $**7** $**4** $-1 $**8** forward singleFT -2.23606 -2.10190 0 3.14159 #
-6 **face** $-1 -1 -1 $-1 $**9** $**10** $**4** $-1 $**11** forward singleFT -2.23606 -2.10190 -3.14159 0#
-9 **face** $-1 -1 -1 $-1 $-1 $**13** $**4** $-1 $**14** forward singleFF #

Table 11. Three surfaces of the conical form.

Surfaces (Geometrical Entities)
-8 **cone-surface** $-1 -1 -1 $-1 0 0 -1000 0 0 -1 -500 0 0 1 I I 0.44721 0.89442 500 forward I I I I #
-11 **cone-surface** $-1 -1 -1 $-1 0 0 -1000 0 0 -1 -500 0 0 1 I I 0.44721 0.89442 500 forward I I I I #
-14 **plane-surface** $-1 -1 -1 $-1 0 0 -60 0 0 -1 -1000 0 0 forward_v I I I I #

Finally, all the mentioned entities are tied to the high level entities which are normally located in lines 0, 2, and 4. The first numbered line 0 represents the higher-level topological entity called body, which as mentioned before is composed by a set of lumps. It can be seen in this line that the body is pointing to line number 2, which is the only lump needed to represent the conical form. At the same time, the lump is pointing to line number 4, which is the shell that is pointing to line number 5, which is a face.

```
-0 body $1 -1 -1 $-1 $2 $-1 $-1 F #
-2 lump $3 -1 -1 $-1 $-1 $4 $0 F #
-4 shell $-1 -1 -1 $-1 $-1 $-1 $5 $-1 $2 F #
```

In summary, it can be said that the ACIS code needs three points, three vertices, four edges, four curves, eight coedges, three loops, three faces, three surfaces, one shell, one lump, and one body to create a conical form and to determine the position, orientation, and dimensions of the solid. This information, as well as the entities of other features, is used to train the feature recognition program which is explained in the following sections. For this reason, Table 12 shows the geometrical and topological entities needed to create the form features used in this research.

Table 12. Geometrical and topological entities of complex forms features.

		Form Feature			
		Cylinder	Cone	Sphere*	Torus
Geometrical Entities	Points	4	3	6	4
	Curves	6	4	6	8
	Surfaces	4	3	2	4
Topological Entities	Vertex	4	3	6	4
	Edges	6	4	8	8
	Coedge	12	8	16	16
	Loop	4	3	4	4
	Faces	4	3	4	4
	Shell	1	1	1	1
Total Number of Entities		45	32	53	53

*The orientation of the sphere in the CAD program can increase or decrease the number of entities.

4.3 Feature Recognition

A decision support system to verify form tolerances should be an automated tool that gets input, in this case the CAD model from the user, and uses it to extract valuable information that allows the system to recognize the type of feature, the dimensions of the feature, and the orientation of the feature. For this reason, the present study proposes the use of a feature recognition system that is built based on the ACIS format shown above. The system asks the user to provide the ACIS file of a solid model. Then the system cleans the ACIS code by using a program developed using Microsoft® Excel macros, which is used to remove all unnecessary characters and information. The program generates a data matrix that becomes the input of the feature recognition program developed in Matlab® that uses the entities' information shown in Table 12 to determine the type of feature and its dimensions. Then the program generates the data points needed to inspect the feature based on the strategy selected by the user. The following lines show an example of the steps needed to recognize the conical form shown above in the ACIS example.

First, the ACIS file of the conical form is imported as a text file into a Microsoft Excel® worksheet, where the data on the file is separated into columns by spaces. The cleaning program that is stored in the form of macros is executed, and it goes line by line removing unnecessary information and leaving only the entities, dimensions, and the pointers that relate the entities to one another. The program removes the names of all the entities and replaces them by using the seven-bit Binary Counter code (BC-code), seen in Table 13. The macro reads the name of the entity and substitutes its name with the corresponding five bit binary code. The last two remaining bits correspond to the counter that tells the program the number of times that the entity is used in the code.

Table 13. Binary code used to create data matrix

Entities	Binary code	Counter	BC-Code
Face	10001	00	1000100
Loop	10010	00	1001000
Plane surface	10011	00	1001100
Sphere surface	10100	00	1010000
Cone surface	10101	00	1010100
Torus surface	10110	00	1011000
Coedge	10111	00	1011100
Edge	11000	00	1100000
Vertex	11001	00	1100100
Straight curve	11010	00	1101000
Ellipse curve	11011	00	1101100
Point	11100	00	1110000
Shell	11101	00	1110100

An example of the cleaning process for each type of entity is shown in the next lines and in Table 14. The unnecessary information in each of the lines of the ACIS code is shown in white font color and black background, while the wanted information is shown in black font color and white background. The required information for each entity is as follows:

- Face: the type of entity, and the pointers of face, loop, shell, and surface.
- Loop: the first coedge of the loop and the number of the face created by the loop.
- Plane surface: the point that determines the location of the plane and the normal vector of the plane.
- Cone surface: the relationship between the vertical and horizontal components that create the slope of the cone.
- Coedge: the next, previous, and partner coedges, the edge, and the loop.
- Edge: the vertices that define the edge, the coedge, and the type of curve.
- Vertex: the edge and the point.
- Straight curve: type of curve (0).
- Ellipse curve: type of curve (1).
- Point: the location of the point.
- Shell: shell counter.

Table 14. Cleaning process of the geometrical and topological entities.

Face	
ACIS	-5 face $-1 -1 -1 $-1 $6 $7 $4 $-1 $8 forward single F T -2.23 -2.10 0 3.14 #
BC code	5 1000101 6 7 4 8
Loop	
ACIS	-7 $-1 -1 -1 $-1 $-1 $12 $5 F unknown #
BC code	7 1001001 12 5
Plane Surface	
ACIS	-14 plane-surface $-1 -1 -1 $-1 0 0 -60 0 0 -1 -1000 0 0 forward_v I I I #
BC code	14 1001101 0 0 -60 0 0 -1
Cone Surface	
ACIS	-8 cone-surface $-1 -1 -1 $-1 0 0 -1000 0 0 -1 -500 0 0 1 I 0.447 0.894 500#
BC code	8 1010101 0.447 0.894
Coedge	

ACIS	-12 coedge $-1 -1 -1 $-1 $16 $17 $18 $19 reversed $7 $-1 #
BC code	12 1011101 16 17 18 19 7
Edge	
ACIS	-19 edge $-1 -1 -1 $-1 $25 0 $26 0.0670 $18 $27 forward @7 unknown F#
BC code	19 1100001 25 0 26 0 18 27
Vertex	
ACIS	-25 vertex $-1 -1 -1 $-1 $22 $32 #
BC code	25 1100101 22 32
Straight Curve	
ACIS	-27 straight-curve $-1 -1 -1 $-1 0 0 0 -447.21 0 -894.42 I I #
BC code	27 1101001 0
Ellipse Curve	
ACIS	-31 ellipse-curve $-1 -1 -1 $-1 0 0 -60 0 0 1 30 0 0 1 I I#
BC code	31 1101101 1
Point	
ACIS	-32 point $-1 -1 -1 $-1 0 0 0 #
BC code	32 1110001 0 0 0
Shell	
ACIS	-4 shell $-1 -1 -1 $-1 $-1 $-1 $5 $-1 $2 F #
BC code	4 1110101 1

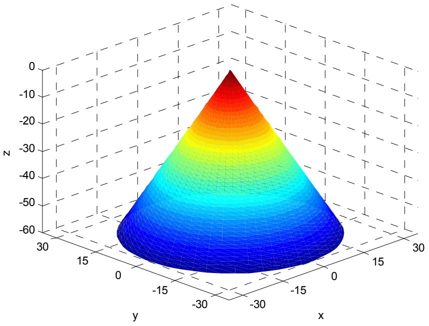
The second step is the creation of the data matrix. After cleaning each line, the macro creates a data matrix of *n*x13, *n* being the number of entities on the ACIS file, seen in Table 15. The last column of the matrix represents the counter of all entities, which are used by the feature recognition program to extract the information. For the cone feature, the last column reads: 32 entities, 3 faces, 3 loops, 1 plane surface, 0 sphere surface, 2 cone surface, 0 torus surface, 8 coedges, 4 edges, 3 vertex, 2 straight curve, 2 ellipse curve, 3 points, and 1 shell.

Table 15. Data matrix that represents the entities of a conical feature.

4	1110101	1	0	0	0	0	0	0	0	0	0	32
5	1000101	6	7	4	8	0	0	0	0	0	0	3
6	1000102	9	10	4	11	0	0	0	0	0	0	3
7	1001001	12	5	0	0	0	0	0	0	0	0	1
8	1010101	0.447	0.894	0	0	0	0	0	0	0	0	0
9	1000103	-1	13	4	14	0	0	0	0	0	0	2
10	1001002	15	6	0	0	0	0	0	0	0	0	0
11	1010102	0.447	0.894	0	0	0	0	0	0	0	0	8

12	1011101	16	17	18	19	7	0	0	0	0	0	4
13	1001003	20	9	0	0	0	0	0	0	0	0	3
14	1001101	0	0	-60	0	0	-1	0	0	0	0	2
15	1011102	18	21	16	22	10	0	0	0	0	0	2
16	1011103	17	12	15	22	7	0	0	0	0	0	3
17	1011104	12	16	23	24	7	0	0	0	0	0	1
18	1011105	21	15	12	19	10	0	0	0	0	0	0
19	1100001	25	0	26	0	18	27	0	0	0	0	0
20	1011106	23	23	21	28	13	0	0	0	0	0	0
21	1011107	15	18	20	28	10	0	0	0	0	0	0
22	1100002	25	0	29	0	16	30	0	0	0	0	0
23	1011108	20	20	17	24	13	0	0	0	0	0	0
24	1100003	29	0	26	3	17	31	0	0	0	0	0
25	1100101	22	32	0	0	0	0	0	0	0	0	0
26	1100102	28	33	0	0	0	0	0	0	0	0	0
27	1101001	0	0	0	0	0	0	0	0	0	0	0
28	1100004	26	-3	29	0	21	34	0	0	0	0	0
29	1100103	28	35	0	0	0	0	0	0	0	0	0
30	1101002	0	0	0	0	0	0	0	0	0	0	0
31	1101101	1	0	0	0	0	0	0	0	0	0	0
32	1110001	0	0	0	0	0	0	0	0	0	0	0
33	1110002	-30	0	-60	0	0	0	0	0	0	0	0
34	1101102	1	0	0	0	0	0	0	0	0	0	0
35	1110003	30	0	-60	0	0	0	0	0	0	0	0

The third step is to execute the feature recognition program created in Matlab®. The first line of the program is to load the data matrix into memory, and then the program reads the last column of the matrix and defines the type of feature that is contained in the matrix. The program compares the number of entities in the matrix against those values obtained in Table 12, which allow it to determine whether the matrix contains a single- or multi-feature part. If the program matches the total number of entities of the data matrix with one of the values of Table 12 then the program proceeds to determine the dimensions and orientation of the feature by reading the information of the geometrical entities such as the normal vectors of the surfaces and the location of the points. For example, the data matrix shown above has 32 entities in total that fit the description of a conical form shown in Table 12. Then, the program reads the normal vectors, the inclination and the position of the surfaces, which help it determine that the cone has a vertex located at (0, 0, 0) and the axis of revolution in the same position as the *z*-axis. Once the program has finished recognizing the object, then it displays a summary of the objects including a set of matrices of the geometrical and topological entities, as shown below.



$$\text{Coedge} = \begin{bmatrix} 12 & 16 & 17 & 18 & 19 & 7 \\ 15 & 18 & 21 & 16 & 22 & 10 \\ 16 & 17 & 12 & 15 & 22 & 7 \\ 17 & 12 & 16 & 23 & 24 & 7 \\ 18 & 21 & 15 & 12 & 19 & 10 \\ 20 & 23 & 23 & 21 & 28 & 10 \\ 21 & 15 & 18 & 20 & 28 & 10 \\ 23 & 20 & 20 & 17 & 24 & 13 \end{bmatrix}$$

$$\text{Feature} = \text{Cone} \qquad \text{Shell} = [1]$$

$$\text{Face} = \begin{bmatrix} 5 & 7 & 8 \\ 6 & 10 & 11 \\ 9 & 13 & 14 \end{bmatrix} \qquad \text{Point} = \begin{bmatrix} 32 & 25 & 0 & 0 & 0 \\ 33 & 26 & -30 & 0 & -60 \\ 35 & 29 & 30 & 0 & -60 \end{bmatrix}$$

$$\text{Vertex} = \begin{bmatrix} 25 & 32 \\ 26 & 33 \\ 29 & 35 \end{bmatrix} \qquad \text{Edge} = \begin{bmatrix} 19 & 25 & 26 & 51 & 27 \\ 22 & 25 & 29 & 54 & 30 \\ 24 & 29 & 26 & 55 & 31 \\ 28 & 26 & 29 & 55 & 34 \end{bmatrix}$$

$$\text{Curve} = \begin{bmatrix} 27 & 0 \\ 30 & 0 \\ 31 & 1 \\ 34 & 1 \end{bmatrix} \qquad \text{Surface} = \begin{bmatrix} 8 & 0.447 & 0.894 & 0 & 0 & 0 & 0 \\ 11 & 0.447 & 0.894 & 0 & 0 & 0 & 0 \\ 14 & 0 & 0 & -60 & 0 & 0 & -1 \end{bmatrix}$$

Figure 23. Feature and entities matrices extracted using feature recognition program.

On the other hand, if the program was not able to match the total number of entities with the totals of Table 12, then the program identify the type of surfaces and the total number of points in the file and sorts the points from highest to lowest. Then the features are recognized based on the location and number of points that belong to the feature. Finally, the form feature and the entities matrices are used to determine the way in which the feature is inspected. This involves the number of data points and the strategy used to collect them, as shown in the next chapter.

CHAPTER 5

SAMPLE POINT GENERATION, MACHINE SETUP, AND DATA COLLECTION

5 Summary

The feature recognition process explained in the previous chapter generated matrices that contained information regarding the dimensions and orientation of the form feature(s) that needed to be inspected. These matrices were used to generate the sampling points, to set up the machined part on the inspection equipment, and to collect the data using a collision free path. Each of these procedures is vulnerable to many different sources of error that can affect the readings as well as the analysis of the form tolerances (Hocken et al., 1993). For this reason, the following sections provide recommendations and insights that can be used to avoid these errors.

5.1 Sample Point Generation

One of the biggest problems in the inspection process of manufactured parts is the calculation of the number of sample points needed to obtain a good estimate of the zone of error. Larger data sets require longer inspection times than the small data sets, but smaller data sets do not provide enough information from the surface of the part. In the literature, several authors have focused on solving this problem by studying the mathematical sequences that yield a small amount of error, such as the Hammersley sampling strategy (Woo et al., 1993), and on those sequences that are able to capture the fingerprint on the surface of the part left by the manufacturing process (Hocken et al., 1993), such as the adaptive sampling strategies (Edgeworth et al., 1999; Kim et al., 1999; Badar et al., 2003) and spiral sampling strategy (Collins, Fay, Aguirre Cruz, and Raman, 2006, 2007). In addition to these strategies, others have been studied in the literature but are not as accurate as the Hammersley distribution, such as the Halton-Zaremba that can be considered as a special case of the Hammersley distribution since the main difference between both strategies is that the Halton-Zaremba strategy generates its data points in batches that are powers of two. The random strategy has the disadvantage that the generated sample points are not evenly distributed. Rather they are clustered in different zones over the surface of the part. The uniform and aligned systematic strategies generate the datasets based on location of the first data points, causing the sequences to ignore several regions of the surface of the part. Most of the papers agree that the Hammersley strategy has a better point distribution than the mentioned sampling strategies. For example, Woo et al. (1993) and Lee et al. (1997) found that the Hammersley distribution needs fewer points to produce great levels of accuracy compared to the uniform distribution. In a different study, Woo et al. (1995) studied the surface roughness of manufacturing parts by comparing the efficiency of the Hammersley, Halton-Zaremba and the uniform sampling strategies. The authors found that there were not significant differences between the Hammersley and the Halton-Zaremba sampling strategies, while the uniform strategy generated the highest root mean square error. In contrast, Kim et al. (1999) found that the Halton-Zaremba strategy has a smaller discrepancy than the Hammersley, random, and uniform distributions when applied to the evaluation of flatness error.

In addition to these strategies, there are sequences that focus on adapting the distribution of the data points according to the resources used to machine a part. This type is called adaptive sampling and was studied by Edgeworth et al. (1999), who used an iterative method of inspection and evaluation to determine the surface error, and by Collins, Fay, Aguirre Cruz, and Raman (2006, 2007), who used a spiral sampling method with the purpose of capturing the pattern left by the face milling and turning processes. The application of these strategies requires the development of an algorithm that evaluates the performance of the points as they are being collected. However, an algorithm cannot be created if the mathematical models contained in it are not very accurate. For this reason, the present research focuses on developing accurate and robust mathematical models that can be used to evaluate the data points. By doing this, the residuals can be analyzed in order to determine the points that create the boundaries of the zone of error of the inspected part. The developed mathematical models must be evaluated using an accurate sampling strategy, and since the Hammersley distribution has been proved to be a good sequence, the present study uses the Hammersley distribution to analyze the form verification models that are developed in Chapter 6. It is important to mention that the decision support system should be able to generate any type of sampling strategy in order to create a system that can be adaptable to the requirements of the user.

5.2 Hammersley Sampling Strategy

The Hammersley sampling strategy was chosen in the present research as the strategy to collect the data points due to the fact that it has been proved in the literature to provide a good sample point distribution. The Hammersley distribution has been applied to circles, spheres, cones, (Lee et al. 1997, Prakasvudhisarn 2002) and torus forms (Aguirre Cruz 2003). This section provides the mathematical formulation used to apply the Hammersley sampling strategy to the inspection path planning for the form features studied in this research.

The foundation of the Hammersley distribution came from a low error mathematical sequence developed by Van der Corput in 1935 that later on was improved by Roth (1954), Hammersley (1960) and Lee et al. (1997) into a *d* dimensional randomized sampling sequence. Thus, the main parameters of the Hammersley sampling strategy were defined by

$$\begin{aligned} x_{Hi} &= \frac{i}{N} + x_{rand} \qquad ; \text{if} \left(\frac{i}{N} + x_{rand} \right) < 1 \\ &= \left(\frac{i}{N} + x_{rand} \right) - 1 \ ; \text{otherwise} \end{aligned} \tag{5.1}$$

$$\begin{aligned} y_{Hi} &= \left(\sum_{j=0}^{k-1} b_{ij} 2^{-j-1} \right) + y_{rand} \qquad ; \text{if} \left(\left(\sum_{j=0}^{k-1} b_{ij} 2^{-j-1} \right) + y_{rand} \right) < 1 \\ &= \left[\left(\sum_{j=0}^{k-1} b_{ij} 2^{-j-1} \right) + y_{rand} \right] - 1 \quad ; \text{otherwise} \end{aligned} \tag{5.2}$$

where (*N*) is the total number of data points, (*i*) represents the *i*th data point in the range of [0,…, *N*-1], (*k*) represents the total number of bits needed to define the desired sample size and is obtained by calculating the logarithm with base two of the total number of data points (*N*), (*j*) represents the position of the bits defined in the range of [0,…, *k*], (b_{ij}) represents the *j*th bit of the *i*th binary number (b_i), and (x_{rand}, y_{rand}) is a point coordinate drawn at random that affects the location of all data points.

Features such as spheres, cones, cylinders, frustums, and torus forms have in common that they all can be viewed from a plane perpendicular to the axis of revolution as a partial circle in the case of the torus form or as a full circle in the case of the other form features. Therefore, Lee et al. (1997) proposed a transformation of the Hammersley parameters from Cartesian to polar coordinates, which allow the definition of the parameters of the features using a central point. The mathematical expression of this transformation was represented by

$$r_{Hi} = r\sqrt{y_{Hi}} \tag{5.3}$$

$$\theta_{Hi} = 360° x_{Hi} \tag{5.4}$$

where (r) is the radius of the form feature being inspected, and 360 represents the location of the radial distance on the circular shape. Finally, the polar coordinates defined above were adapted to three dimensional form features by substituting the calculated radius and angle into the equation that represents the height of the form feature being inspected. The following lines show an example of eight Hammersley coordinates that were adapted to spherical, cylindrical, conical, frustum, and torus form features. In the final section of this chapter (Data Collection), the Hammersley coordinates are regarded as measuring points since they represent those points that are inspected by the coordinate measuring machine. In addition, the polar coordinate obtained from this procedure must be transformed into Cartesian coordinates due to the fact that the coordinate measuring machine requires the latter format as an input of the numerical control (NC) mode.

5.2.1 Hammersley Sequence Applied to Surfaces of Revolution

For simplicity, the following example assumes that all form features have the same radius (*r*). This means that the radius of the sphere, the radius of the cylinder, the radius of the base of the cone, and the distance from the origin of the torus to its boundary are all equal to 1.25 inches. The random point, given by x_{rand} = 0.39 and y_{rand} = 0.7, is added to the each one of the Hammersley coordinates, as shown in Table 16.

Table 16. Example of eight data points generated using the Hammersley strategy.

i	b_i	$b_{i2}2^{-2-1}$	$b_{i1}2^{-1-1}$	$b_{i0}2^{-0-1}$	$\sum b_{ij}2^{-j-1}$	i/N	x_{Hi}	y_{Hi}	r_{Hi}	θ_{Hi}
0	(0,0,0)	0	0	0	0	0	0.390	0.700	1.046	140.4
1	(0,0,1)	0	0	0.500	0.500	0.1	0.490	0.200	0.559	176.4
2	(0,1,0)	0	0.250	0	0.250	0.2	0.590	0.950	1.218	212.4
3	(0,1,1)	0	0.250	0.500	0.750	0.3	0.690	0.450	0.839	248.4
4	(1,0,0)	0.125	0	0	0.125	0.4	0.790	0.825	1.135	284.4
5	(1,0,1)	0.125	0	0.500	0.625	0.5	0.890	0.325	0.713	320.4
6	(1,1,0)	0.125	0.250	0	0.375	0.6	0.990	0.075	0.342	356.4
7	(1,1,1)	0.125	0.250	0.500	0.875	0.7	0.090	0.575	0.948	32.4

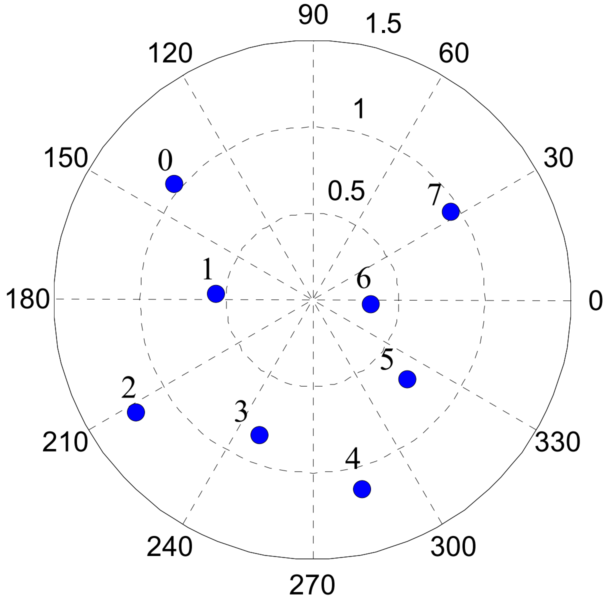


Figure 24. Polar plot of eight Hammersley coordinates.

The polar coordinates shown in Figure 24 are transformed into Cartesian coordinates for each feature so that they can be used as measuring points for the collision free path program of the coordinate measuring machine. Firstly, the polar coordinates are adapted to inspect spherical form features, where the *x* and *y* components are calculated by multiplying the Hammersley radii (r_{Hi}) by the cosine and sine functions of the angle (θ_{Hi}), respectively. The *z* component is found by substituting the x_{Mi} and y_{Mi} components into the equation of the sphere and solving for z_{Mi}, as seen in Equations (5.1) to (5.4). The measuring points of the spherical form are shown in Table 17, where the geometrical origin of the sphere is located at the center of the shape.

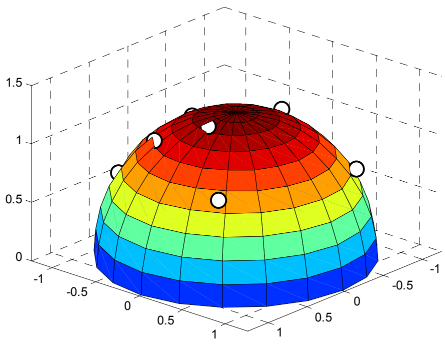
$$x_{Mi} = r_{Hi} \cos\theta_{Hi} \tag{5.1}$$

$$y_{Mi} = r_{Hi} \sin\theta_{Hi} \tag{5.2}$$

$$r = \sqrt{x_{Mi}^2 + y_{Mi}^2 + z_{Mi}^2} \tag{5.3}$$

$$z_{Mi} = \sqrt{r^2 - \left(x_{Mi}^2 + y_{Mi}^2\right)} \tag{5.4}$$

Table 17. Measuring points of the spherical form.

x_{Mi}	y_{Mi}	z_{Mi}	
-0.806	0.667	0.685	
-0.558	0.035	1.118	
-1.029	-0.653	0.279	
-0.309	-0.780	0.927	
0.282	-1.100	0.523	
0.549	-0.454	1.027	
0.342	-0.021	1.202	
0.800	0.508	0.815	

The measuring points of the cylindrical form are obtained in a different way than the ones obtained for the spherical form, due to the fact that the cylinder has a constant radius that makes necessary that the Euclidian distance between the *x* and *y* components to yield the radius of the cylinder. For this reason, the *x* and *y* components of the cylindrical form are calculated by multiplying the radius of the cylinder (*r*) times the cosine and sine functions of the angle (θ_{Hi}), respectively. The *z* component is obtained by using a virtual conical form with the same radius of the base (*r*) and height (*h*=1) as the cylindrical form. By doing this, the *z* is calculated by the negative product between the Euclidian distance (r_{Mi}) of Equations (5.1) and (5.2) times the relation between the radius and the height of the conical form, as seen in Equations (5.5) to (5.8) and Table 18. It is important to mention that the geometrical origin of the cylinder is set to be on the center of the top plane.

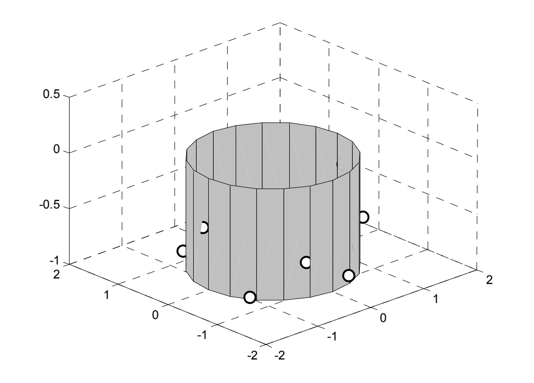
$$x_{RMi} = r\cos\theta_{Hi} \tag{5.5}$$

$$y_{RMi} = r\sin\theta_{Hi} \tag{5.6}$$

$$r_{Mi} = \sqrt{x_{Mi}^2 + y_{Mi}^2} \tag{5.7}$$

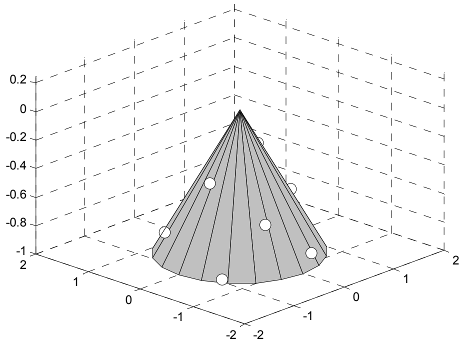
$$z_{RMi} = -r_{Mi}\left(\frac{h}{r}\right) \tag{5.8}$$

Table 18. Measuring points of the cylindrical form.

x_{Mi}	y_{Mi}	z_{Mi}	
-0.963	0.797	-0.837	
-1.248	0.078	-0.447	
-1.055	-0.670	-0.975	
-0.460	-1.162	-0.671	
0.311	-1.211	-0.908	
0.963	-0.797	-0.570	
1.248	-0.078	-0.274	
1.055	0.670	-0.758	

The measuring points of conical forms are obtained by using the Equations (5.1) and (5.2) for the calculations of the *x* and *y* components, respectively, while the *z* component is obtained by following the Equation (5.8) developed for the cylindrical form. Table 19 shows the eight measuring points of the conical form, where the geometrical origin of the cone is located at the vertex and the height of the cone (*h*) is equal to 1.

Table 19. Measuring points of the conical form.

x_{Mi}	y_{Mi}	z_{Mi}	
-0.806	0.667	-0.837	
-0.558	0.035	-0.447	
-1.029	-0.653	-0.975	
-0.309	-0.780	-0.671	
0.282	-1.100	-0.908	
0.549	-0.454	-0.570	
0.342	-0.021	-0.274	
0.800	0.508	-0.758	

The measuring points of the frustum form requires the Hammersley radial distance (r_{Hi}) to be calculated in a different way from the one shown in Table 16. The new radial distance (r_{FHi}) is calculated by multiplying the difference between the radius of the frustum base (*r*) and the radius of the top circular plane (*r*') by the square root of the vertical Hammersley component (y_{Hi}). Then, the *x* and *y* components of the measuring points are calculated by adding the new radial distance (r_{FHi}) and the radius of the top circular plane (*r*') and multiplying them by the cosine and sine functions of the angle (θ_{Hi}), respectively. Finally, the *z* component is obtained by multiplying the negative Euclidian distance between the *x* and *y* components of the measuring points times the relation between the height of the conical form (*h*) and the radius of the frustum base (*r*). It is important to notice that the height of the conical form is not the height of the frustum form but the height of a complete cone. In addition, the geometrical origin of the frustum is located at the vertex of the complete conical form. Table 20 shows the eight coordinates of the frustum form.

$$r_{FHi} = (r - r')\sqrt{y_{Hi}} \tag{5.9}$$

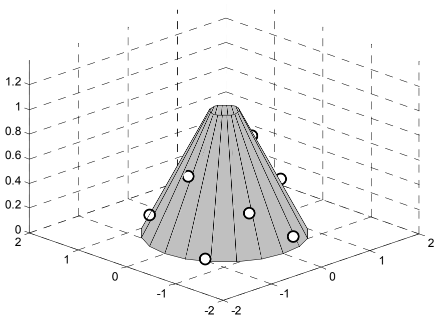
$$x_{FMi} = (r_{FHi} + r')\cos\theta_{Hi} \tag{5.10}$$

$$y_{FMi} = (r_{FHi} + r')\sin\theta_{Hi} \tag{5.11}$$

$$r_{FMi} = \sqrt{x_{FMi}^2 + y_{FMi}^2} \tag{5.12}$$

$$z_{Mi} = -r_{FMi}\left(\frac{h}{r}\right) \tag{5.13}$$

Table 20. Measuring points of the frustum form.

x_{Mi}	y_{Mi}	z_{Mi}	
-0.837	0.693	-0.837	
-0.696	0.044	-0.447	
-1.034	-0.656	-0.975	
-0.339	-0.856	-0.671	
0.288	-1.122	-0.908	
0.632	-0.523	-0.570	
0.523	-0.033	-0.274	
0.851	0.540	-0.758	

Similar to the frustum form, the torus forms requires the Hammersley radial distance (r_{Hi}) to be calculated in a different way from the one shown in Table 16. For this reason, the new radial distance (r_{THi}) is calculated by adding two terms: the first term is obtained by multiplying two times the radius of the torus tube (a=0.25) by the square root of the vertical Hammersley component (y_{Hi}), while the second term is the radius of the torus hole, given by the difference between the radius of the torus (c=1) and the radius of the torus tube (a). Then, the x and y components of the measuring points are calculated by multiplying the new radial distance (r_{THi}) by the cosine and sine functions of the angle (θ_{Hi}), respectively. Finally, the *z* component is calculated using the Pythagorean Theorem between the radius of the torus tube (*a*) and the horizontal radial distance, where the horizontal radial distance is obtained by the difference between the radius of the torus (*c*) and the Euclidian distance between the *x* and *y* components of the measuring points. Table 21 shows the eight coordinates of the torus form.

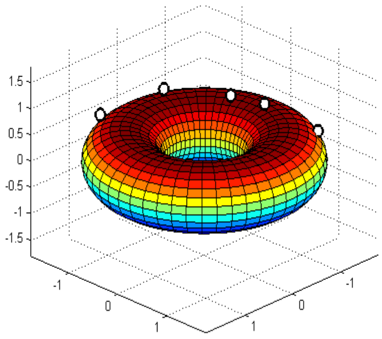
$$r_{THi} = 2a\sqrt{y_{Hi}} + (c - a) \tag{5.14}$$

$$x_{TMi} = r_{THi} \cos \theta_{Hi} \tag{5.15}$$

$$y_{TMi} = r_{THi} \sin \theta_{Hi} \tag{5.16}$$

$$z_{TMi} = \sqrt{a^2 - \left(c - \sqrt{x_{TMi}^2 + y_{TMi}^2}\right)^2} \tag{5.17}$$

Table 21. Measuring points of the torus form.

x_{Mi}	y_{Mi}	z_{Mi}	
-1.287	1.065	0.462	
-1.182	0.074	0.622	
-1.556	-0.988	0.196	
-0.539	-1.361	0.587	
0.438	-1.705	0.361	
1.031	-0.853	0.619	
0.965	-0.061	0.557	
1.328	0.843	0.535	

5.3 Coordinate Measuring Machine Setup

The form tolerance verification of manufactured parts is a procedure for collecting data points from the surface of the part and analyzing them to verify the geometry of the form feature. The evaluation of this tolerance requires a set of tools that ensures the accurate estimation of the parameters of the features, such as: the sampling strategy that determines the location and amount of sampling points, as explained above; a collision free path plan that tells the inspection equipment the movements required to inspect the sampling points, explained in the next section; and the inspection equipment that collects and analyzes the data points, explained in this section.

The inspection equipment such as the CMM allows the metrologist to inspect the tolerances of parts by following predetermined procedures set by the manufacturer of the equipment. Each type of CMM is unique according to the instructions that need to be followed to operate the machine. Most of these procedures are general information that the metrologist needs to follow so that software can interpret and analyze the data. DEA (1995a, b), the manufacturers of the software, TUTOR for Windows^{TM}, that is used in this research, suggested procedures for qualifying and calibrating the stylus tip, translating the origin of the CMM, and inspecting manufacturing tolerances, among others. One of the most important operations that needs to be done before inspecting the tolerances of a part is to determine the location of the origin of the machine with respect to the geometrical origin of the part. Most mathematical models available in the literature require the origin of the machine to be located in the same or very close to the position to the geometrical origin of the part. If this operation is not followed, then the mathematical models will not be able to determine the parameters of the part that is being inspected or they will need powerful algorithms to extract those parameters from the measured data. For this reason, it is very important to collect the data according to the way in which it will be analyzed. It is important to mention that the software of the CMM as well as many other computer programs (Polyworks$^{®}$ and Geomagic$^{®}$) contain powerful algorithms that can fit primitive or free forms on the measured data regardless of the location of the origin of the machine. However, the algorithms that they use are proprietary. Thus, the mathematics that they apply to analyze the data are unknown.

The software used in this research, TUTOR for Windows^{TM}, provides four options to translate the origin of the machine. The first is “preset on circle,” which allows the user to collect four or more data points on a circular shape with the purpose of estimating the center of the circle, thus positioning the origin of the two involved Cartesian components in that location. For example, if a circular shape is located on the *xy* plane, then the machine will set the origin of the *x* and *y* components to zero at the same position as the origin of the circular shape. The last three options are “x preset,” “y preset,” and “z preset,” which allow the user to collect one point from any surface and the machine to locate the origin of the component over the virtual plane in which the point was collected. For example, if the user needs to locate the origin of a square block on the southwest corner, then the user needs to select the “x preset” and collect one point from the south face of the square block. Next the user needs to select the “y preset” and collect one point from the west face of the square block. Finally, the user needs to select the “z preset” and collect a point from the top face of the block. Therefore, the intersection between the three planes yields a point located at the southwest corner of the block.

In a similar way, the present research uses these four options to translate the origin of the machine into the same position as the geometrical origin of spheres, cylinders, cones, frustums, and torus forms. The translation is not straightforward since for some shapes such as the torus that has its geometrical origin in the middle of the doughnut hole. For this reason, the following lines show the way in which the origin of the machine can be translated in order to fulfill the requirements of the mathematical models. In addition, the experiments in Chapter 7 show examples of setting the origin of the machine in an inaccurate position, while Chapter 8 shows the consequences of this mistake.

The following procedures take into account that the CMM used in the present research can only measure the top part of the feature forms. This means that the CMM can only inspect the top hemisphere of a complete sphere surface, all the cone surface, all the cylinder surface, all the frustum surface, and the top half of the torus surface. In addition, it is assumed that the origin of the machine is located at O_s (x_s, y_s, z_s).

The procedure to translate the origin of the machine into the geometrical origin of the sphere (located at the center of the shape) requires three steps that use the option "preset on circle":

1. The first step is to set the x and y components of the origin along the axis of revolution of the sphere. This is done by inspecting four points on the north, east, south, and west positions. The points must be contained on the same plane, where the location of the plane z=a should be as close as possible to the equator of the sphere, as seen on the left side of Figure 25.
2. The second step is to set the z component of the origin at the center of the sphere. In order to do this, the user needs to choose between the x=0 or y=0 planes, which are used to inspect four points on the arc that belong either to the xz or yz planes, respectively. By doing this, the two components of the origin (xz or yz) are set to zero at the center of this

arc, as seen on the right side of Figure 25. Since it is very difficult to find the *x*=0 or *y*=0 plane, then a third step is needed.

3. The third step is basically to repeat the first step in order to ensure that the *x* and *y* components of the origin were not affected by the second step.

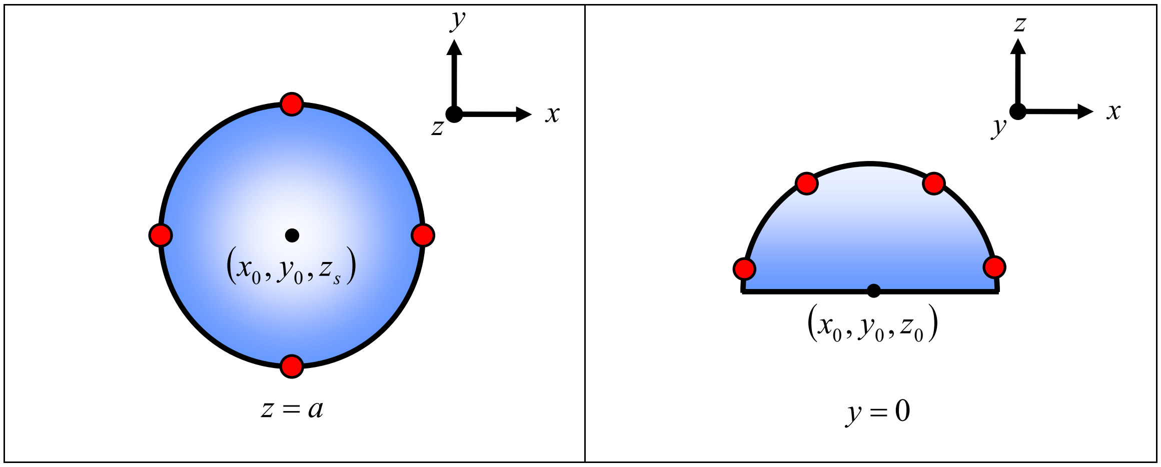


Figure 25. Top (left) and front (right) view of the procedure used to translate the origin of the machine into the location of the geometrical origin of the sphere.

The procedure to translate the origin of the machine into the geometrical origin of the cylinder (located at the top-center of the shape) requires two steps that use the option "preset on circle," and "z preset," respectively:

1. The first step is to set the *x* and *y* components of the origin along the axis of revolution of the cylinder. This is done by inspecting four points on the north, east, south, and west positions. The points must be contained on the same plane, where the plane z=*a* can be located at any height, as seen on the left side of Figure 26.

2. The second step is to set the *z* component of the origin at top-center of the cylinder. In order to do this, the option "z preset" is selected, which

allows the user to inspect one point at the desired height of the *z*-axis. Thus, a point is inspected at the top of the cylinder, and since the *x* and *y* components are already set at the center, then the origin's new position is located at the top-center of the cylinder, as seen on the right side of Figure 26.

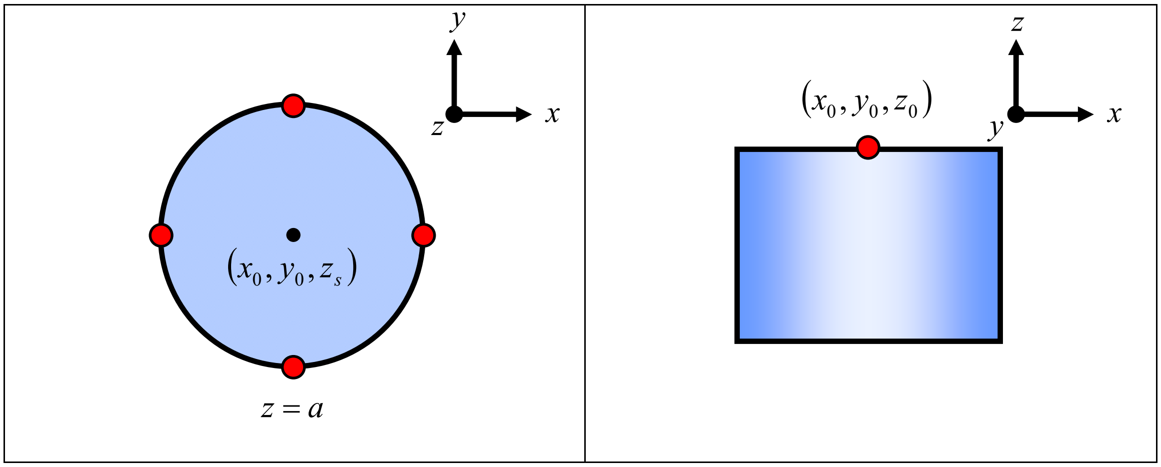


Figure 26. Top (left) and front (right) view of the procedure used to translate the origin of the machine into the location of the geometrical origin of the cylinder.

The procedure to translate the origin of the machine into the geometrical origin of the cone (located at the vertex of the shape) requires two steps that use the option "preset on circle," and "z preset," respectively:

1. The first step is to set the *x* and *y* components of the origin along the axis of revolution of the cone. This is done by inspecting four points on the north, east, south, and west positions. The points must be contained on the same plane, where the plane z=*a* can be located at any height, as seen on the left side of Figure 27.

2. The second step is to set the *z* component of the origin at the vertex of the cone. In a similar way as the cylinder, the option "z preset" is

selected and a point is inspected at the vertex of the cone, and since the *x* and *y* components are already set at the center, then the origin's new position is located at the vertex of the cone, as seen on the right side of Figure 27.

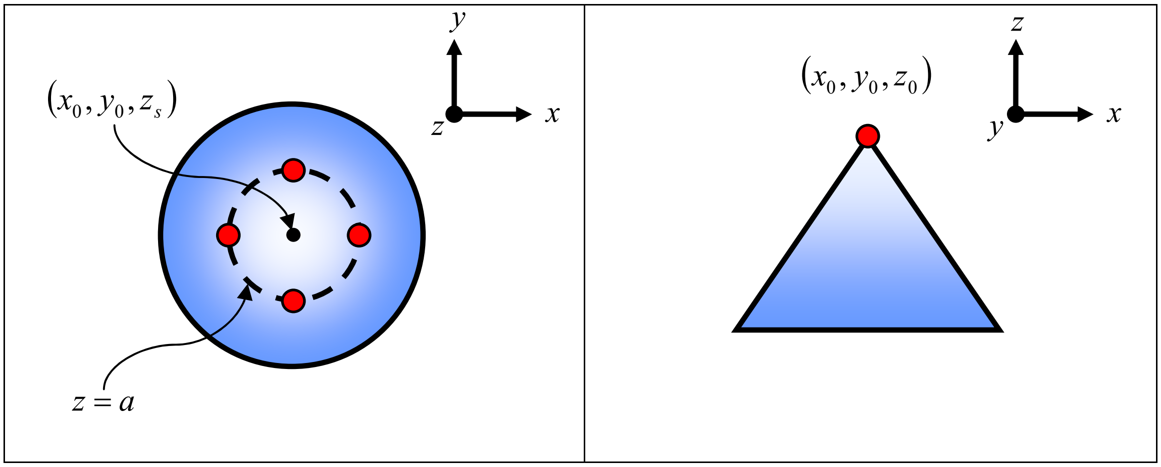


Figure 27. Top (left) and front (right) view of the procedure used to translate the origin of the machine into the location of the geometrical origin of the cone.

The procedure to translate the origin of the machine into the geometrical origin of the frustum (located at the top-center plane of the shape) requires two steps that use the option "preset on circle," and "z-axis," respectively:

1. The first step is to set the *x* and *y* components of the origin along the axis of revolution of the frustum. This is done by inspecting four points on the north, east, south, and west positions. The points must be contained on the same plane, where the plane z=*a* can be located at any height, as seen on the left side of Figure 28.

2. The second step is to set the *z* component of the origin at the top-center plane of the frustum. Thus, the option "z preset" is selected and a point is inspected at any location on the top plane of the frustum, and since

the *x* and *y* components are already set at the center, then the origin's new position is located at the top-center plane of the frustum, as seen on the right side of Figure 28.

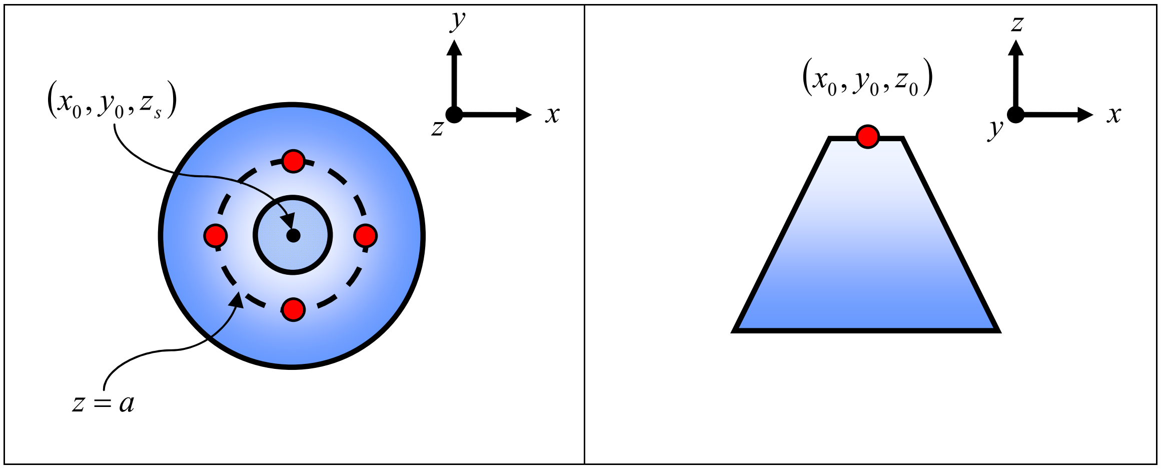


Figure 28. Top (left) and front (right) view of the procedure used to translate the origin of the machine into the location of the geometrical origin of the frustum.

The procedure to translate the origin of the machine into the geometrical origin of the torus (located at the center of the hole of the torus) requires three steps that use the option "preset on circle":

1. The first step is to set the *x* and *y* components of the origin along the axis of revolution of the torus. This is done by inspecting four points on the north, east, south, and west positions. The points must be contained on the same plane, where the location of the plane *z*=*a* should be as close as possible to the equator of the torus, as seen on the left side of Figure 29.
2. The second step is to set the *z* component of the origin. In a similar way as the spherical form, the user needs to choose between the *x*=0 or *y*=0 planes, which are used to inspect four points on the arc of the torus tube

that belong either to the *xz* or *yz* planes, respectively. By doing this, the two components of the origin (*xz* or *yz*) are set to zero at the center of this arc, as seen on the right side of Figure 29. However, since the origin of the torus is not located at the center of the torus tube, then a third step is needed.

3. The third step is basically to repeat the first step in order to set the *x* and *y* components of the origin.

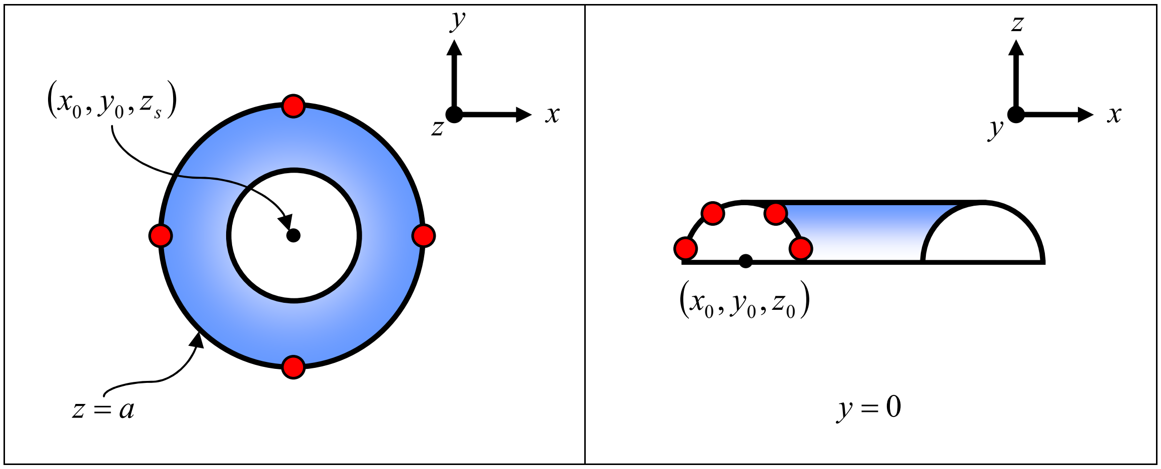


Figure 29. Top (left) and front (right) view of the procedure used to translate the origin of the machine into the location of the geometrical origin of the torus.

5.4 Data Collection

The path planning process is defined as the distance traveled by the inspection equipment to collect all data points from the surface of the part. The purpose of the path planning is to create a pathway that collects all information using the minimum amount of time, which also implies that the inspection device has to travel the shortest path without colliding with the part.

The present study investigates two types of path planning approaches: the raindrop approach and the hexagonal mesh approach. The raindrop approach is a heuristic and simple way to generate a path plan. This approach uses three types of points: the measuring, the positioning, and z-plane points. Basically, the approach moves the stylus tip horizontally using the z-plane points that are located above the part. Each one of the z-plane points is located above the positioning points, thus, the stylus tip is lowered as a raindrop from the z-plane point to the positioning point. Then, the stylus moves to the measuring point using a vector that is normal to the surface. Once that the point is inspected, the stylus moves back to the z-plane in order to locate the next point. Although the raindrop approach is simple, it consumes a lot of time since it is not an optimized sequence. An example of the input files and CMM program used to generate this approach can be seen in Appendix B.

For this reason, the present research proposes an optimized approach based on a hexagonal mesh, which can be easily adapted to any surface. In addition, the approach will collect the data points using the normal vector of the surface, which will increase the reliability and accuracy of the readings. The approach calculates the shortest distance between one measuring point and the next and uses the vertices and links of the hexagonal mesh to find the shortest traveling path. An example of the file and CMM program used to generate this approach can be seen in Appendix B.

5.4.1 Raindrop Path Planning

The raindrop approach is a collision free path plan that uses three different types of points to inspect a part. The points are defined as: measuring points, which are the actual points that belong to the surface of the part that is being inspected. The positioning points are offset in the outward direction at a specific distance that is normal to the surface of the part. The location of these points determines the approaching vector that the inspection equipment uses to collect the measuring points. Finally, the z-plane points are located on a plane that is normal to the axis of revolution of the part that is being inspected. The plane is located above the part, which is used by the inspection equipment to position the stylus tip above the positioning points.

The inspection equipment collects the data points by first positioning the stylus tip above the first positioning point. Then, the machine lowers the tip until it reaches the positioning point. Next, the stylus moves from the positioning point to the measuring point and collects the information from the surface. Subsequently, the machine automatically retracts to the positioning point and then retracts the stylus tip in a vertical direction to the location of z-plane point. Once in the z-plane the machine moves to a different z-plane point and continues the process explained above until all measuring points are collected. In order to simplify the understanding of this process, the raindrop approach is explained by using an example that calculates the positioning and z-plane points of the surfaces of revolution, with the measuring points obtained in the Table 16 as a reference.

5.4.1.1 Raindrop Approach Applied to Surfaces of Revolution

The example shows the parameters and calculations needed to obtain the positioning and z-plane points. In addition, the example assumes that the distance between the measuring points and the positioning points is 0.20 inches; this distance is regarded as the positioning offset (o_P) in this example. The location of the z-plane varies according to the form because the origin of the part is located at different positions. In addition, the example uses three symbols to differentiate the three types of points: a circle for the measuring points, a square for the positioning points, and a star as z-plane points.

The positioning and z-plane points for the spherical form are found by calculating the angle (α_{Si}) that determines the relation between the *xy* horizontal component and the *z* component. The main reason for the calculation of this angle is that it can be used to break up the positioning offset (o_P) into *x*, *y*, and *z* components. Thus, the (x_{SPi}) and (y_{SPi}) components of the positioning points are calculated by adding the ideal radius of the sphere (r) and the positioning offset (o_P) and multiplying them by the trigonometric functions related to each of the components, while the (z_{SPi}) component is obtained by adding the ideal radius of the sphere (r) and the positioning offset (o_P) and multiplying them by the cosine of the angle (α_{Si}), as seen in Equations (5.19) to (5.21).

Finally, the components of the z-plane points are obtained as follows: the (x_{SZi}) and (y_{SZi}) components are the same as the positioning points, while the (z_{SZi}) component is obtained by adding the ideal radius of the sphere plus the positioning offset, as seen in Equations (5.22) to (5.24). Table 22 and Figure 30 show the values and location of the positioning and z-plane points.

$$\alpha_{Si} = \tan^{-1}\left(\frac{\sqrt{x_{Mi}^2 + y_{Mi}^2}}{z_{Mi}}\right) \tag{5.18}$$

$$x_{SPi} = (r + o_P)\cos\theta_{Hi}\sin\alpha_{Si} \tag{5.19}$$

$$y_{SPi} = (r + o_P)\sin\theta_{Hi}\sin\alpha_{Si} \tag{5.20}$$

$$z_{SPi} = (r + o_P)\cos\alpha_{Si} \tag{5.21}$$

$$x_{SZi} = x_{SPi} \tag{5.22}$$

$$y_{SZi} = y_{SPi} \tag{5.23}$$

$$z_{SZi} = \left(r + o_P\right) \tag{5.24}$$

Table 22. Positioning and z-plane point coordinates of a spherical form.

Positioning Points			**Z-Plane Points**		
x_{SPi}	y_{SPi}	z_{SPi}	x_{SZi}	y_{SZi}	z_{SZi}
-0.935	0.773	0.794	-0.935	0.773	1.450
-0.647	0.041	1.297	-0.647	0.041	1.450
-1.193	-0.757	0.324	-1.193	-0.757	1.450
-0.358	-0.904	1.075	-0.358	-0.904	1.450
0.328	-1.276	0.607	0.328	-1.276	1.450
0.637	-0.527	1.191	0.637	-0.527	1.450
0.396	-0.025	1.395	0.396	-0.025	1.450
0.928	0.589	0.945	0.928	0.589	1.450

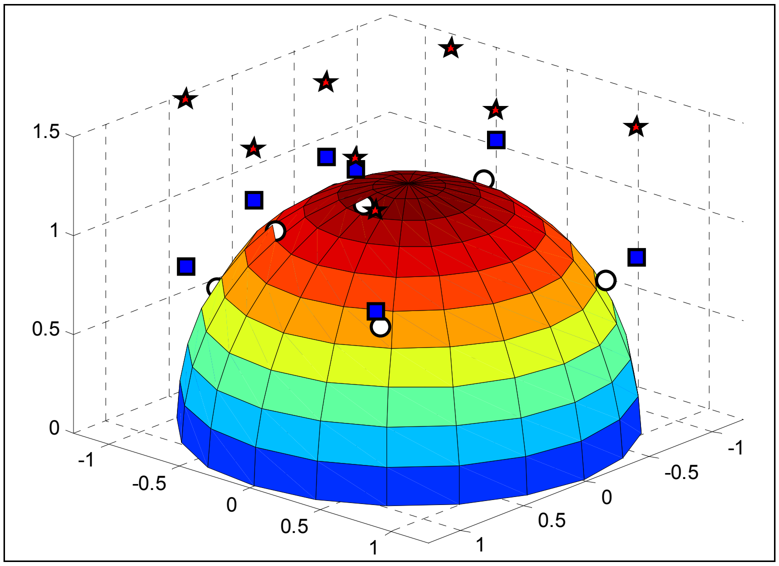


Figure 30. Measuring (circle), positioning (square), and z-plane (star) points of a spherical form.

The positioning and z-plane points for the cylindrical form are found in a simpler way than the spherical form, because the Euclidian distance between the *x* and *y* components yields the radius of the cylinder. The (x_{RPi}) and (y_{RPi}) components of the positioning points are calculated by adding the ideal radius of the cylinder (r) and the positioning offset (o_P) and multiplying them by the cosine and sine functions of the angle (θ_{Hi}), respectively; while the (z_{RPi}) component is identical in magnitude to the (z_{Mi}) component of the measuring points, as seen in Equations (5.25) to (5.27).

Finally, the components of the z-plane points are obtained as follows: the (x_{RZi}) and (y_{RZi}) components are the same as the positioning points, while the (z_{RZi}) component is given by the positioning offset (o_P), as seen in Equations (5.28) to (5.30). Table 23 and Figure 31 show the values and location of the positioning and z-plane points of the cylindrical form.

$$x_{RPi} = (r + o_P)\cos\theta_{Hi} \tag{5.25}$$

$$y_{RPi} = (r + o_P)\sin\theta_{Hi} \tag{5.26}$$

$$z_{RPi} = z_{Mi} \tag{5.27}$$

$$x_{RZi} = x_{RPi} \tag{5.28}$$

$$y_{RZi} = y_{RPi} \tag{5.29}$$

$$z_{RZi} = o_P \tag{5.30}$$

Table 23. Positioning and z-plane point coordinates of a cylindrical form.

Positioning Points			**Z-Plane Points**		
x_{RPi}	y_{RPi}	z_{RPi}	x_{RPi}	y_{RPi}	z_{RPi}
-1.117	0.924	-0.837	-1.117	0.924	0.200
-1.447	0.091	-0.447	-1.447	0.091	0.200
-1.224	-0.777	-0.975	-1.224	-0.777	0.200
-0.534	-1.348	-0.671	-0.534	-1.348	0.200
0.361	-1.404	-0.908	0.361	-1.404	0.200
1.117	-0.924	-0.570	1.117	-0.924	0.200
1.447	-0.091	-0.274	1.447	-0.091	0.200
1.224	0.777	-0.758	1.224	0.777	0.200

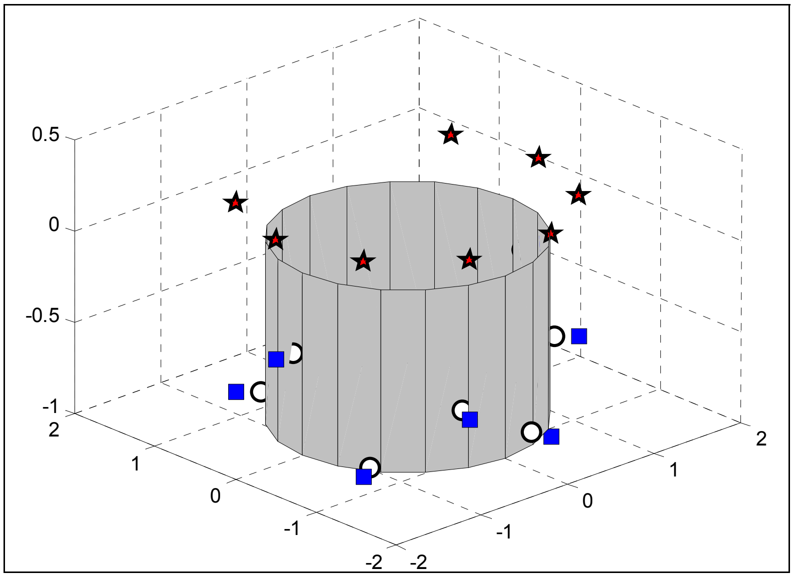


Figure 31. Measuring (circle), positioning (square), and z-plane (star) points of a cylindrical form.

The positioning and z-plane points for the conical form are found by calculating the angle (α_{Ci}) that determines the relation between the radius of the base of the cone (r) and the height of the cone (h), as shown in Equation (5.31). In a similar way to the spherical form, this angle is calculated primarily because it can be used to break up the positioning offset (o_P) into x, y, and z components. In addition, the (xy_i) horizontal component is calculated in order to simplify the calculation of the individual x and y components. Thus, the (xy_i) component is calculated by adding the Euclidian distance between the (x_{Mi}) and (y_{Mi}) components of the measuring points plus the product between the positioning offset (o_P) times the sine of the angle (α_{Ci}), seen in Equation (5.32).

Consequently, the (x_{CPi}) and (y_{CPi}) components of the positioning points are calculated by multiplying the (xy_i) component by the cosine and sine functions of the angle (θ_{Hi}), respectively. The (z_{CPi}) component is obtained by adding the (z_{RMi}) component of the measuring points plus the product between the positioning offset (o_P)

and the cosine of the angle (α_{Ci}), as seen in Equations (5.33) to (5.35). Finally, the components of the z-plane points are obtained as follows: the (x_{CZi}) and (y_{CZi}) components are the same as the positioning points, while the (z_{CZi}) component is given by the positioning offset (o_P), as seen in Equations (5.36) to (5.38). Table 24 and Figure 32 show the values and location of the positioning and z-plane points.

$$\alpha_{Ci} = \tan^{-1}\left(\frac{r}{h}\right) \tag{5.31}$$

$$xy_i = \sqrt{x_{Mi}^2 + y_{Mi}^2} + o_P \sin\alpha_{Ci} \tag{5.32}$$

$$x_{CPi} = xy_i \cos\theta_{Hi} \tag{5.33}$$

$$y_{CPi} = xy_i \sin\theta_{Hi} \tag{5.34}$$

$$z_{CPi} = z_{RMi} + o_P \cos\alpha_{Ci} \tag{5.35}$$

$$x_{CZi} = x_{CPi} \tag{5.36}$$

$$y_{CZi} = y_{CPi} \tag{5.37}$$

$$z_{CZi} = o_P \tag{5.38}$$

Table 24. Positioning and z-plane point coordinates of a conical form.

Positioning Points			**Z-Plane Points**		
x_{CPi}	y_{CPi}	z_{CPi}	x_{CZi}	y_{CZi}	z_{CZi}
-0.926	0.766	-0.7117	-0.926	0.766	0.200
-0.714	0.045	-0.3223	-0.714	0.045	0.200
-1.161	-0.737	-0.8497	-1.161	-0.737	0.200
-0.366	-0.925	-0.5459	-0.366	-0.925	0.200
0.321	-1.251	-0.7834	0.321	-1.251	0.200
0.669	-0.554	-0.4451	0.669	-0.554	0.200
0.498	-0.031	-0.1489	0.498	-0.031	0.200
0.932	0.592	-0.6333	0.932	0.592	0.200

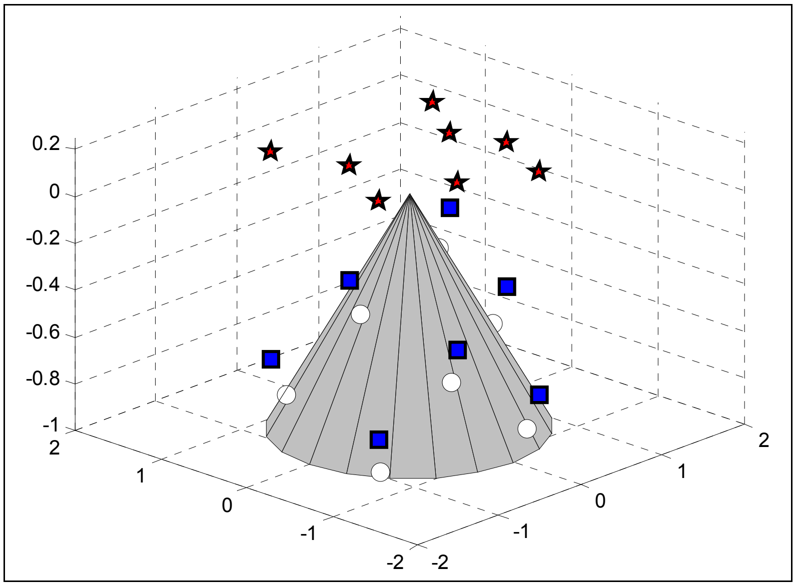


Figure 32. Measuring (circle), positioning (square), and z-plane (star) points of a conical form.

The positioning and z-plane points of the frustum form are obtained by using the equations obtained from the conical form, where Equations (5.33) to (5.35) describe the positioning points, while Equation (5.36) to (5.38) describe the z-plane points. Although, the formulas are the same, the position of the point coordinates differs due to the different form of both surfaces. Thus, Table 25 and Figure 33 show the values and location of the positioning and z-plane points.

Table 25. Positioning and z-plane point coordinates of a frustum form.

Positioning Points			**Z-Plane Points**		
x_{Pi}	y_{Pi}	z_{Pi}	x_{Zi}	y_{Zi}	z_{Zi}
-0.958	0.792	-0.712	-0.958	0.792	0.200
-0.852	0.054	-0.322	-0.852	0.054	0.200
-1.166	-0.740	-0.850	-1.166	-0.740	0.200
-0.396	-1.001	-0.546	-0.396	-1.001	0.200
0.327	-1.273	-0.783	0.327	-1.273	0.200
0.752	-0.622	-0.445	0.752	-0.622	0.200
0.679	-0.043	-0.149	0.679	-0.043	0.200
0.983	0.624	-0.633	0.983	0.624	0.200

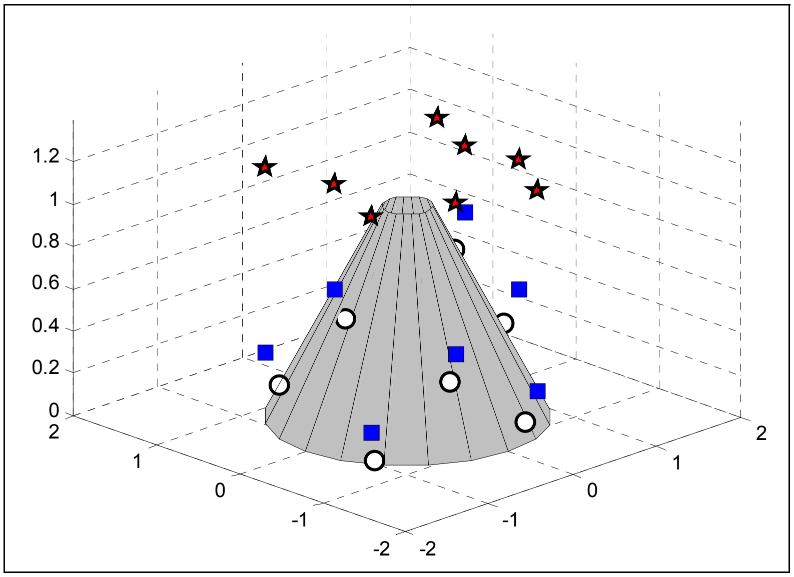


Figure 33. Measuring (circle), positioning (square), and z-plane (star) points of a frustum form.

The positioning and z-plane points for the torus form are found by calculating the angle (α_{Ti}) that determines the relation between the *xy* horizontal component and the *z* component inside the torus tube, as seen in Equation (5.39). This angle is calculated because it can be used to break up the positioning offset (o_P) into *x*, *y*, and *z* components. Thus, the (x_{TPi}) and (y_{TPi}) components of the positioning points are calculated by evaluating the position of the point with respect to the ideal radius of the torus (c). If the point has an angle (α_{Ti}) greater than 90 degrees, then the *xy* horizontal component of the point is smaller than the ideal radius of the torus (c), and the (x_{TPi}) and (y_{TPi}) components must be calculated by subtracting the horizontal component of the torus tube from the ideal radius of the torus (c), and multiplying the result by cosine and sine of the angle (θ_{Hi}), respectively. On the other hand, if the point has an angle (α_{Ti}) smaller than 90 degrees, then the *xy* horizontal component of the point is greater than the ideal radius of the torus (c) and the (x_{TPi}) and (y_{TPi}) components must be calculated by adding the horizontal component of the torus tube plus the ideal radius of the torus (c), and multiplying the result by cosine and sine of the angle (θ_{Hi}), respectively. It is important to mention that the horizontal component of the torus tube is obtained by adding the ideal radius of the torus tube (a) plus the positioning offset (o_P), and multiplying the result by the cosine of the angle (α_{Ti}).

The (z_{TPi}) component is obtained by adding the ideal radius of the torus tube (a) plus the positioning offset (o_P) and multiplying the result by the sine of the absolute value of the angle (α_{Ti}). The three components of the positioning points are seen in Equations (5.40) to (5.42). Finally, the components of the z-plane points are obtained as follows: the (x_{TZi}) and (y_{TZi}) components are the same as the positioning points, while the (z_{TZi}) component is obtained by adding the ideal radius of the torus tube (a) plus the positioning offset (o_P), as seen in Equations (5.43) to (5.45). Table 26 and Figure 34 show the values and location of the positioning and z-plane points.

$$\alpha_{Ti} = \tan^{-1}\left(\frac{z_{TMi}}{r_{Hi} - c}\right) \tag{5.39}$$

$$\begin{array}{ll} \text{if } \alpha_{Ti} > 0 & x_{TPi} = \left[c + \left(a + o_P\right)\cos\alpha_{Ti}\right]\cos\theta_{Hi} \\ \text{otherwise} & x_{TPi} = \left[c - \left(a + o_P\right)\cos\alpha_{Ti}\right]\cos\theta_{Hi} \end{array} \tag{5.40}$$

$$\begin{array}{ll} \text{if } \alpha_{Ti} > 0 & y_{TPi} = \left[c + \left(a + o_P\right)\cos\alpha_{Ti}\right]\sin\theta_{Hi} \\ \text{otherwise} & y_{TPi} = \left[c - \left(a + o_P\right)\cos\alpha_{Ti}\right]\sin\theta_{Hi} \end{array} \tag{5.41}$$

$$z_{TPi} = \left(a + o_P\right)\sin\left|\alpha_{Ti}\right| \tag{5.42}$$

$$x_{TZi} = x_{TPi} \tag{5.43}$$

$$y_{TZi} = y_{TPi} \tag{5.44}$$

$$z_{TZi} = a + o_P \tag{5.45}$$

Table 26. Positioning and z-plane point coordinates of a torus form.

x_{TPi}	y_{TPi}	z_{TPi}	x_{TPi}	y_{TPi}	z_{TPi}
-1.391	1.151	0.610	-1.391	1.151	0.925
-1.161	0.073	0.820	-1.161	0.073	0.925
-1.717	-1.089	0.259	-1.717	-1.089	0.925
-0.564	-1.424	0.775	-0.564	-1.424	0.925
0.478	-1.863	0.476	0.478	-1.863	0.925
1.052	-0.870	0.817	1.052	-0.870	0.925
0.875	-0.055	0.736	0.875	-0.055	0.925
1.415	0.898	0.706	1.415	0.898	0.925

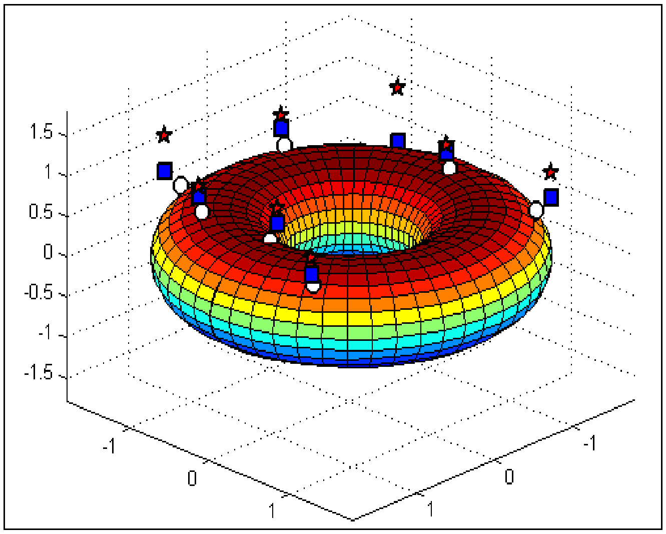


Figure 34. Measuring (circle), positioning (square), and z-plane (star) points of a torus form.

5.4.2 Hexagonal Mesh Path Planning

Without a doubt, the hexagonal mesh approach is better than the raindrop approach due to the mathematical principles behind it, as shown below. However, some problems were encountered while retrieving the measured coordinates from the output file provided by the CMM, because the software of the CMM records the measured coordinates without subtracting or compensating from the radius of the stylus tip. This problem generates point coordinates from an inspected part that have bigger dimensions than the real dimensions of the part. For this reason, the hexagonal mesh was not used to collect the data of the present research; instead the raindrop approach was applied. It is important to mention that the program and files created for the hexagonal mesh approach are functional (seen in Appendix B), but the CMM software was the main cause of this setback.

Furthermore, the CMM software provides a summary of the dimensions and parameters of the inspected part, which are accurate according to the dimensions of the part. This led us to conclude that the CMM software has some kind of proprietary protection that does not allow the user to obtain the real point coordinates, while using the type of program seen in Appendix B.

The hexagonal mesh approach is applied only to spherical forms, due to the reason mentioned above. However, this approach can be applied to any type of form feature. The hexagonal mesh approach is an optimized path plan that uses three hexagonal shapes that are rotated 60 degrees from each other. This arrangement creates a hexagonal frustum that encloses the part, where the vertices and links that creates the frustum form are used to guide the movements of the CMM.

Similar to the raindrop approach, the hexagonal mesh uses three types of points: the measurement, the positioning, and the mesh points. The measuring points lie on the surface of the inspected part. The positioning points are located on a positioning sphere that is bigger in size than the inspected sphere, while the mesh points, which are similar as the z-plane points explained in the raindrop approach, are a fixed net of points that are located on the positioning sphere that has a radius (r_H), which is given by:

$$r_H = r + o_P \tag{5.46}$$

where (r) is the radius of the spherical shape being inspected and (o_P) is the positioning offset explained in the raindrop approach. Thus, instead of making the CMM go up and down as in the raindrop approach, the mesh helps the CMM to position itself in a faster rate to the location of the next measuring point. The main components of the hexagonal approach are seen in Figure 35.

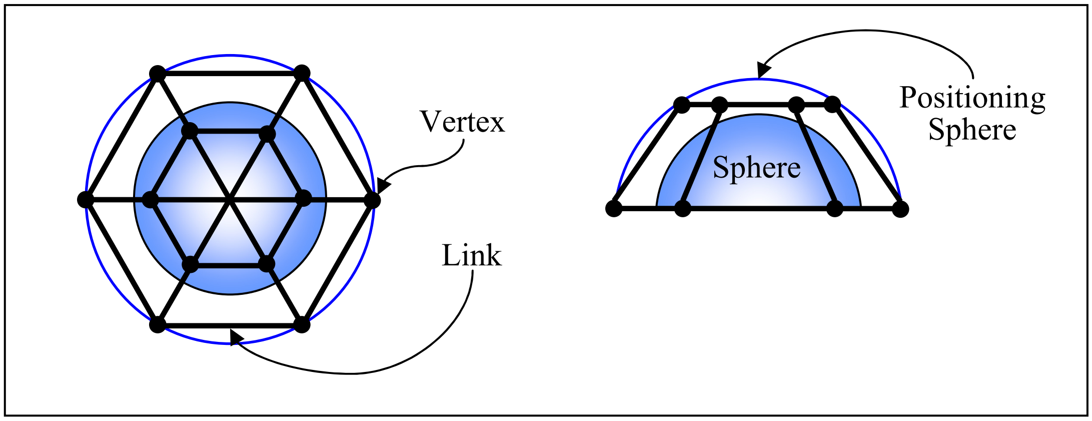


Figure 35. Top (left) and front (right) view of hexagonal mesh components.

The hexagonal mesh has twelve vertices; six from the top-small hexagon (A, B, C, D, E, F) and six from the bottom-big hexagon (1, 2, 3, 4, 5, 6), where the point coordinates of these vertices are calculated based on the radius of the positioning sphere (r_H), as seen on the left side of Figure 36 and in Tables 27 and 28. Moreover, the vertices from the top-small hexagon are defined by a circle that is half of the size ($0.5r_H$) of the circular reference that defines the position of the vertices of the bottom-big hexagon (r_H). Thus, the two circular references and the hexagonal mesh split the inspected part into twelve zones, as seen on right side of the Figure 36. The zones determine the path that the CMM needs to follow to collect all measuring data points. For example, if the previous and next measuring points are located in the same zone, then there is no need to use any mesh points, since the CMM can travel between the previous and next positioning points without colliding with the part. On the other hand, if the previous measuring point is located in zone *a* and the next measuring point is located in zone *d*, then the several mesh points are used to guide the movements of the CMM around the part, as explained in the following example.

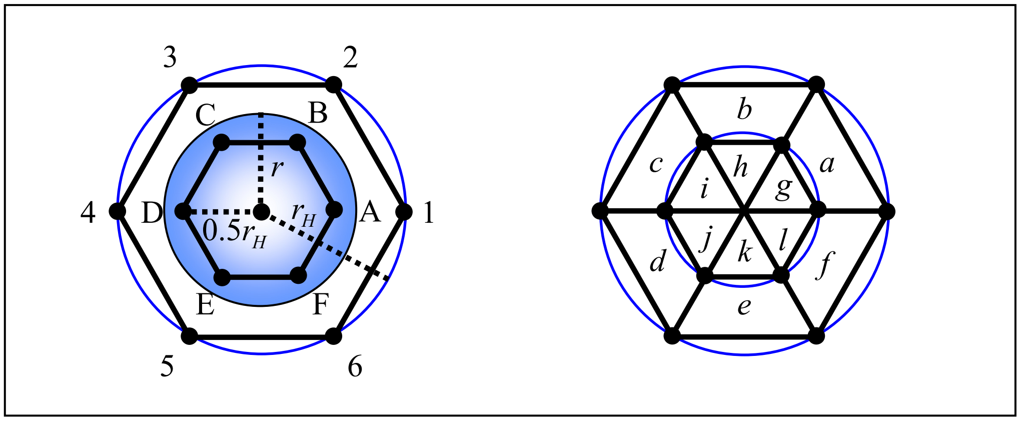


Figure 36. Top view of the vertices of the two hexagonal shapes (left) and top view of the twelve zones used to partition the inspected part (right).

Table 27. Point coordinates of vertices of bottom-big hexagon.

Vertex	Point Coordinates		
	x	***y***	***z***
1	r_H	0	0
2	$\frac{r_H}{2}$	$\sqrt{r_H^2-\left(\frac{r_H}{2}\right)^2}$	0
3	$-\frac{r_H}{2}$	$\sqrt{r_H^2-\left(\frac{r_H}{2}\right)^2}$	0
4	$- r_H$	0	0
5	$-\frac{r_H}{2}$	$-\sqrt{r_H^2-\left(\frac{r_H}{2}\right)^2}$	0
6	$\frac{r_H}{2}$	$-\sqrt{r_H^2-\left(\frac{r_H}{2}\right)^2}$	0

Table 28. Point coordinates of vertices of top-small hexagon.

Vertex	Point Coordinates		
	x	y	z
A	$\frac{r_H}{2}$	0	$\sqrt{r_H^2-\left(\frac{r_H}{2}\right)^2}$
B	$\frac{r_H}{4}$	$\sqrt{\left(\frac{r_H}{2}\right)^2-\left(\frac{r_H}{4}\right)^2}$	$\sqrt{r_H^2-\left(\frac{r_H}{2}\right)^2}$
C	$-\frac{r_H}{4}$	$\sqrt{\left(\frac{r_H}{2}\right)^2-\left(\frac{r_H}{4}\right)^2}$	$\sqrt{r_H^2-\left(\frac{r_H}{2}\right)^2}$
D	$-\frac{r_H}{2}$	0	$\sqrt{r_H^2-\left(\frac{r_H}{2}\right)^2}$
E	$-\frac{r_H}{4}$	$-\sqrt{\left(\frac{r_H}{2}\right)^2-\left(\frac{r_H}{4}\right)^2}$	$\sqrt{r_H^2-\left(\frac{r_H}{2}\right)^2}$
F	$\frac{r_H}{4}$	$-\sqrt{\left(\frac{r_H}{2}\right)^2-\left(\frac{r_H}{4}\right)^2}$	$\sqrt{r_H^2-\left(\frac{r_H}{2}\right)^2}$

5.4.2.1 Hexagonal Mesh Approach Applied to Spherical Form

The following example was created in Microsoft® Excel, in which all possible paths were calculated in advance in order to simplify the process. However, a program can be created to avoid the calculation of the paths. The examples uses the measuring and positioning data points generated for the spherical form of radius 1.25 inches, shown above in Tables 16 and 17. Thus, it uses the same positioning offset (o_P) of 0.20 inches, which creates a positioning sphere with a radius equal to 1.45 inches. Also, it is assumed that the initial position of the CMM is at the point (0, 0, 8).

The following steps explain the creation of the path plan needed to collect the eight Hammersley data points:

1. Firstly, the program should determine the zone containing each measuring point and the closest vertex to each point. Table 29 shows

the eight measuring and positioning points including the zone and closest vertex for each data point. A graphic representation of this step is seen in Figure 37.

Table 29. Data needed to generate the hexagonal mesh path planning.

Point	x_{Mi}	y_{Mi}	z_{Mi}	x_{SPi}	y_{SPi}	z_{SPi}	**Zone**	**Closest Vertex**
0	-0.806	0.667	0.685	-0.935	0.773	0.794	*c*	C
1	-0.558	0.035	1.118	-0.647	0.041	1.297	*i*	D
2	-1.029	-0.653	0.279	-1.193	-0.757	0.324	*d*	5
3	-0.309	-0.780	0.927	-0.358	-0.904	1.075	*e*	E
4	0.282	-1.100	0.523	0.328	-1.276	0.607	*e*	6
5	0.549	-0.454	1.027	0.637	-0.527	1.191	*f*	F
6	0.342	-0.021	1.202	0.396	-0.025	1.395	*l*	A
7	0.800	0.508	0.815	0.928	0.589	0.945	*a*	B

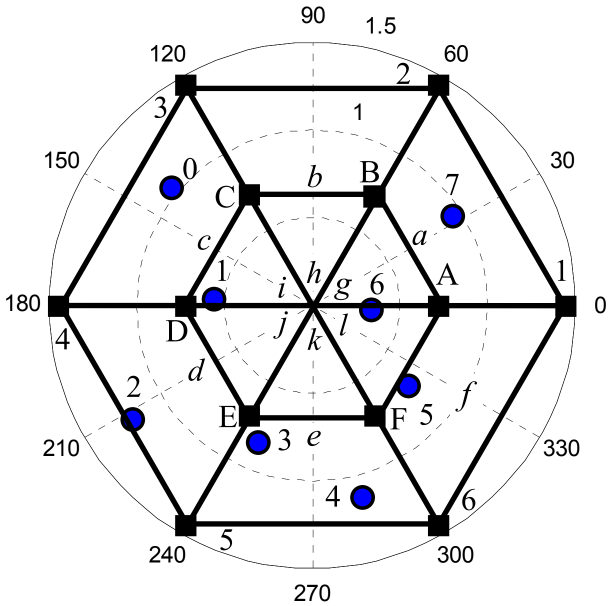


Figure 37. Location of the eight Hammersley points in the hexagonal mesh.

2. Then, it is necessary to move the stylus of the machine to the vertex that is closer to the first measuring point “0”. It is important to mention

that the first vertex must be located on the top-small hexagon in order to avoid any collision. In this example, the first measuring point "0" has as its closest vertex the point "C," located on the top-small hexagon. Thus, only one positioning movement is required to get to the closest vertex, given by the point coordinate of the vertex "C". In addition, since the first movement of the CMM is a positioning point, the first line of the code should start with the letter "P", as seen below.

```
1        P        -0.363        0.628        1.256
```

3. The second line of the code is also a positioning movement, since it is necessary to move the stylus probe from vertex "C" to the positioning point of the first measuring point (x_{SP0}, y_{SP0}).

```
2        P        -0.935        0.773        0.794
```

4. The third line is given by the point coordinate of the first measuring point (x_{SM0}, y_{SM0}), which makes the CMM measure the data point and automatically retract to the location of the positioning point. The line begins with the letter "M" since this coordinate belongs to a measuring point.

```
3        M        -0.806        0.667        0.685
```

5. The next measuring point "1" has as its closest vertex the point "D," and is located in zone "*i*," a different zone than from the previous measuring point. However, the order in which the Hammersley sequence generates the data points in a counterclockwise direction, which simplifies the positioning movements between points, since only one positioning movement is required from one point to another.

Therefore, the fourth line of the code is a positioning movement from the previous positioning point (x_{SP0}, y_{SP0}) to the closest vertex of point "1," given by the point coordinate of the vertex "D."

```
4    P    -0.725    0.000     1.256
```

6. In the next movement, the CMM moves the stylus to the positioning point of point "1" (x_{SP1}, y_{SP1}), which creates the fifth line of the code given by the point coordinate of this positioning point.

```
5    P    -0.647    0.041     1.297
```

7. The sixth line is given by the point coordinate of measuring point "1" (x_{SM1}, y_{SM1}), which makes the CMM measure the data point and automatically retract to the location of the positioning point.

```
6    M    -0.558    0.035     1.118
```

8. The next three lines of the code are given by: the movement of the CMM from the previous positioning point (x_{SP1}, y_{SP1}) to the closest vertex "5" of the next measuring point "2," the movement from the vertex "5" to the positioning point of point "2" (x_{SP2}, y_{SP2}), and the movement from the positioning point (x_{SP2}, y_{SP2}) to the measuring point (x_{SM2}, y_{SM2}), respectively.

```
7    P    -0.725    -1.256    0.000
8    P    -1.193    -0.757    0.324
9    M    -1.029    -0.653    0.279
```

9. Then, the next three lines of the code are given by: the movement of the CMM from the previous positioning point (x_{SP2}, y_{SP2}) to the closest vertex "E" of the next measuring point "3," the movement from the vertex "E" to the positioning point of point "3" (x_{SP3}, y_{SP3}), and the

movement from the positioning point (x_{SP3}, y_{SP3}) to the measuring point (x_{SM3}, y_{SM3}), respectively.

10	P	-0.363	-0.628	1.256
11	P	-0.358	-0.904	1.075
12	M	-0.309	-0.780	0.927

10. The next measuring point “4” is located in the same zone as point “3.” Thus, there is no need to move the stylus to the closest vertex, since it is safe for the machine to move within a zone. Therefore, only two movements are required: the movement from the previous positioning point (x_{SP3}, y_{SP3}) to the next positioning point (x_{SP4}, y_{SP4}), and the movement from the positioning point (x_{SP4}, y_{SP4}) to the measuring point (x_{SM4}, y_{SM4}), respectively.

13	P	0.328	-1.276	0.607
14	M	0.282	-1.100	0.523

11. The next measuring point “5” is located in a different zone than the previous one, thus, three lines of the code are needed to inspect the point. The first line is the movement of the CMM from the previous positioning point (x_{SP4}, y_{SP4}) to the closest vertex “F” of the next measuring point “5.” The second line is the movement from the vertex “F” to the positioning point of point “5” (x_{SP5}, y_{SP5}), and the last line is given by movement from the positioning point (x_{SP5}, y_{SP5}) to the measuring point (x_{SM5}, y_{SM5}).

15	P	0.363	-0.628	1.256
16	P	0.637	-0.527	1.191
17	M	0.549	-0.454	1.027

12. Then, the next three lines of the code are given by: the movement of the CMM from the previous positioning point (x_{SP5}, y_{SP5}) to the closest

vertex “A” of the next measuring point “6,” the movement from the vertex “A” to the positioning point of point “6” (x_{SP6}, y_{SP6}), and the movement from the positioning point (x_{SP6}, y_{SP6}) to the measuring point (x_{SM6}, y_{SM6}), respectively.

18	P	0.725	0.000	1.256
19	P	0.396	-0.025	1.395
20	M	0.342	-0.021	1.202

13. The last three lines of the code are given by: the movement of the CMM from the previous positioning point (x_{SP6}, y_{SP6}) to the closest vertex “B” of the next measuring point “7,” the movement from the vertex “B” to the positioning point of point “7” (x_{SP7}, y_{SP7}),and the movement from the positioning point (x_{SP7}, y_{SP7}) to the measuring point (x_{SM7}, y_{SM7}), respectively.

21	P	0.363	0.628	1.256
22	P	0.928	0.589	0.945
23	M	0.800	0.508	0.815

14. Finally, all lines of the code are arranged in order and are modified according to the requirements of the CMM. The first and last line are code lines that tells the CMM the beginning and end of the point coordinates, while two columns of 1’s are added as a part of the syntaxes required by the CMM software, as seen in Appendix B.

CHAPTER 6

FORM TOLERANCE VERIFICATION OF COMPLEX FORMS

6 Summary

The purpose of this chapter is to explain the analysis, improvements, and implementation of existing and newly developed form verification models. In the literature, mathematical models to verify spheres, cylinders, cones, frustums, and torus forms have been reported. The majority of these papers have focused on spheres and cylinders, a few focused on cones and frustums, and a smaller number have focused on torus forms. Most of the mathematical models presented in those researches performed reasonably; however, this research proposes a number of improvements that can be done to the existing models in order to increase their accuracy. At the same time, this research proposes several new unique models that help to estimate the form error of manufactured parts. This is done with the purpose of providing better tools to the metrologists so that they can make accurate decisions regarding the status of the machining and inspection process.

A secondary goal of this chapter is to create robust geometric models that are not dependant on powerful fitting algorithms to extract the parameters of the part. This means that, if the geometric model is capable of representing the form deviations of the part that is being inspected, then the zone of error will be accurately calculated regardless of the fitting algorithm used. Furthermore, the geometric models must also be as simple as possible in order to make them easy to apply and thus computationally efficient.

The achievement of these goals requires the use of two developed approaches: called the Dynamic Angle Approach (DAA) and the Free Form Orientation (FFO). The former focuses on the dynamic calculation of the magnitude of the angles and the latter on the correction of the axis misalignments. In addition, this chapter also provides information about the implementation of each of the mathematical models, such as constraints that must be used in order to maintain the performance of the models.

6.1 Form Tolerance Verification

Form tolerance verification is a procedure used to determine the quality of a manufactured part. The procedure involves the use of three elements: the measurements; a mathematical model that represents the geometry of the part, the origin offsets and the misalignments due to the axis of assessment; and, last but not least, a fitting algorithm that is used to compare the measurements against the mathematical model in order to extract the dimensions and form of error from the measurements. Therefore, if a mathematical model is not robust enough to represent the form of error of the part, then a powerful fitting algorithm must be used in order to get the accurate parameters and form deviations. On the other hand, if a robust mathematical model is created, then any good fitting algorithm is able to extract the information from the measurements.

The present study proposes newly developed mathematical models and two approaches that improve the performance of the existing models. Hence, the models are analyzed using a simple but strong fitting algorithm, such as the minimum zone method in the form of a nonlinear programming problem. In the literature, there are several fitting algorithms that have been applied to study the form tolerances of manufactured parts, such as the least squares method, minimum zone, support vector regression, Voronoi diagrams, convex hulls, and simplex method, among others. Among all of these algorithms, the least squares method is one of the most used due to its simplicity and the fact that is computationally efficient. However, it has been proved that the algorithm tends to overestimate the zone of error, which causes the rejection of good parts (Anthony 1986). For this reason the ANSI and ISO standards have adopted the use of the minimum zone method that focuses on creating an envelope form that encloses the deviations (d_i). The shape of the envelope is identical to the form being inspected, such as the zone of error of the torus form shown in Figure 38. The envelope is created by minimizing the maximum normal deviation. Hence, the magnitude of the zone of error (H) of any form is based on residuals that are normal to the form being inspected, and it is calculated by subtracting the smallest negative deviation from the largest positive deviation, as shown in Equation (6.1).

$$H = \max(d_i) - \min(d_i) \tag{6.1}$$

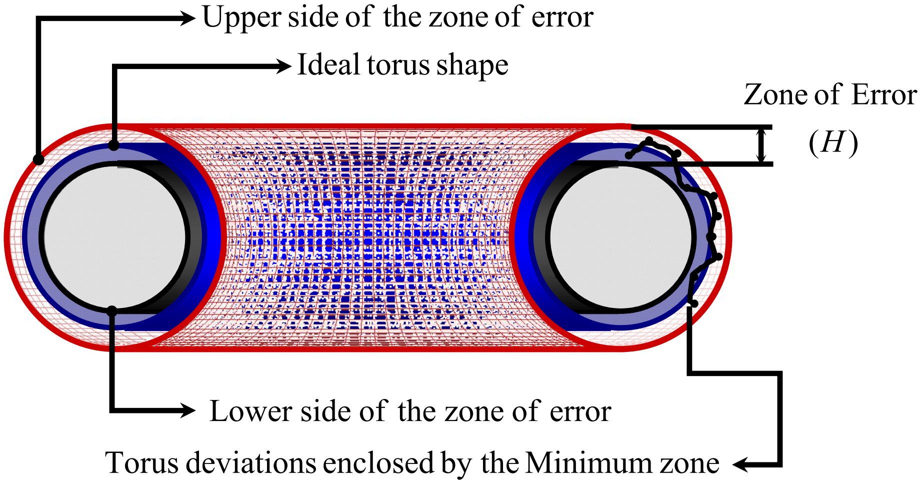

Upper side of the zone of error
Ideal torus shape
Zone of Error
(H)
Lower side of the zone of error
Torus deviations enclosed by the Minimum zone

Figure 38. Graphical representation of a minimum zone of error of a torus form.

The minimum zone algorithm is solved using the minimax approach which models each deviation function (d_i) as an inequality constraint. According to Winston (1995) each variable in the function that is unrestricted in sign should be modeled as the difference between two positive variables, while the variables that are always positive must be left without modification. Hence, the constraints needed to evaluate each mathematical model or deviation function are presented at the end of the section of each form feature.

6.2 Circular Form

The circle or curve segment has two parameters that describe its form: the radius (R) and the location of the origin (x_0, y_0). Shunmugam (1987) and Samuel et al. (2003) studied two different approaches in which the circular form can be evaluated; one is known as the linear circular model and the other as the nonlinear circular model. The linear model was developed based on the principles of the Limacon approximation developed by Chetwynd (1979), in which it is assumed that the axis of the assessed circle is oriented perfectly with the z-axis and that there exist origin offsets between the origin of the machine (O_S) and the geometrical origin of the part (O_C). Equation (6.2) shows the Limacon approximation where the value of (ρ) is the radial distance from the origin of the machine to the location of the ith point, and it is calculated by adding the radius of the circle plus or minus the terms that compensate the horizontal and vertical differences between the machine and the part origins.

$$\rho \cong R + x_0 \cos(\theta_i) + y_0 \sin(\theta_i) \tag{6.2}$$

where (R) represents the radius of the circle, (x_0) is the variable used to represent the origin offsets in the x direction, (y_0) is the variable used to represent the origin offsets in the y direction, and (θ_i) represents the location of the ith point on the xy plane and it is obtained using Equation (6.3).

$$\theta_i = \tan^{-1}\left(\frac{y_i}{x_i}\right) \tag{6.3}$$

In general, the form deviations (d_i) are obtained by subtracting the ideal form from the measured data points. For circular shapes, the ideal form is obtained by using the Limacon approximation shown in Equation (6.2), while the assessed data is obtained from the radial distances (r_i) shown in Equation (6.4) that are found by using the Euclidian distance between the ith point and the origin of the machine $O_S(0,0)$. Hence, the model to verify circular forms is found by substituting Equations (6.3) and (6.4) in Equation (6.5).

$$r_i = \sqrt{x_i^{\,2} + y_i^{\,2}} \tag{6.4}$$

$$d_i = r_i - \left[R + x_0\cos(\theta_i) + y_0\sin(\theta_i)\right] \tag{6.5}$$

The model to verify circular forms shown in Equation (6.4) has been proven to be effective in the literature. However, several researchers have applied the Limacon approximation principle to evaluate form tolerances of circles (Shunmugam, 1987, Samuel et al., 2003), spheres (Shunmugam, 1987, Balakrishna et al., 2005), cylinders (Shunmugam, 1987), cones (Prakasvudhisarn et al., 2004), and torus (Aguirre Cruz et al., 2005) and have regarded the angle (θ_i) as a constant term, obtained by using Equation (6.3). It can be seen from Figure 39, that the angle (θ_i) depends on the location of the origin of the inspection equipment. Thus, if the origin changes, then the angle (θ_i) should be updated as well, resulting in a more accurate approach to find the parameters of the feature than the approach used when the angle (θ_i) is regarded as a constant term. For this reason, the present research proposes an approach called the "*Dynamic Angle Approach (DAA)*," which is an improvement to the Limacon approximation that is used to avoid this problem.

Figure 39 shows two perfect circles that are hypothetically evaluated using the constant angle approach (left circle) and the proposed dynamic angle approach (right circle). It is assumed that the origin of the machine is in a different position than the geometrical origin of the part. Therefore, for this example the geometrical origin (O_C) of both circles is located a few inches to the southwest side of the machine origin (O_S).

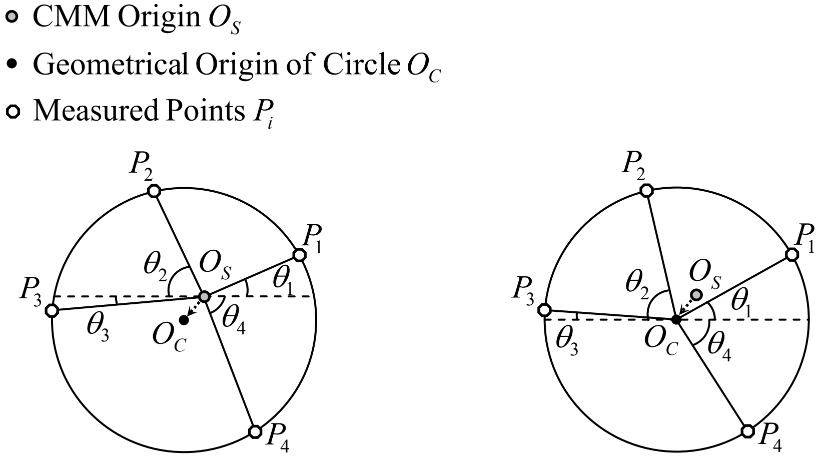


Figure 39. Circular form evaluated using constant angle approach (left), and dynamic angle approach (right).

Then a fitting algorithm is used to extract an optimal solution for the parameters (R), (x_0), and (y_0) that minimizes the form of error. Thus, if the algorithm is applied to the circle in the left, then the solution of this problem will provide a slightly different radius (R) due to the fact that the angles are not updated to a closer location where the optimal solution lies. The solution then is based on inaccurate radial distances and angular magnitudes. On the other hand, when the optimization tool is applied to the circle in the right, then the mathematical model updates the origin offsets as well as the radial distances, which provides a better estimate of the parameters of the circle. Hence, this problem can be solved either by redesigning the existing models in order to make them insensitive to origin offset or avoiding the use of angles in the mathematical models. Both proposed solutions are explained in the following sections and are applied to the mathematical models of sphere, cylinder, cone, frustum, and torus forms.

In contrast, the nonlinear model developed by Shunmugam (1987) is claimed to be more accurate than the linear model since it obtains the circle deviations by subtracting an ideal radial distance (R) from the measured radial distance. The latter distance is referred in this research as (r_{is}) and is obtained in a different way than the radial distance (r_i) because this uses the location of the origin of the circle $O_c(x_0, y_0)$ as a part of the Euclidian distance that represents the measurements. Equation (6.7) shows the nonlinear model that is obtained by subtracting the ideal radial distance (R) from the measured radial distance (r_{is}) shown in Equation (6.6).

$$r_{is} = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} \tag{6.6}$$

$$d_i = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} - R \tag{6.7}$$

6.2.1 Dynamic Angle Approach Applied to Circular Forms

The mathematical principle behind the dynamic angle approach is to define the measured radial distances (r_{is}) in terms of the origin offsets and angles of the Limacon approximation in terms of the *x*, *y*, and *z* components of the measured points and the origin offsets. The adjustment will force the fitting algorithms to find an optimal value for the parameters of the geometrical form that minimizes the zone of error based on the magnitude of the origin offsets, the radius of the circle, and the angular location of the data points. In the literature, authors have developed mathematical models to verify the form tolerance of different forms. Out of these papers, the studies performed by Chen et al. (2000) and Samuel et al. (2003) came closest to updating the magnitude of the angles. Chen et al. (2000) developed a model to inspect conical parts based on the maximum inscribed cone and minimum circumscribed cone. The authors defined the half angle of the cone in terms of the axis of assessment and the radial distances, which helped them to identify the data points that gave the maximum and minimum angles in order to determine the parameters of the tapered cone. Samuel et al. (2003) studied the form tolerances of round parts by using the linear and nonlinear mathematical models to describe these shapes. The point coordinates used in both approaches was identical; the only difference was that the data used for the linear approach were transformed into radial distances and angular values. The transformation involved the use of a reference circle with radius (r_R) and origin (x_R, y_R) that was generated by selecting three points from the dataset. The origin of the reference circle was used to calculate the *i*th angle of each data point, as seen below.

$$\theta_i = \tan^{-1}\left(\frac{y_i - y_R}{x_i - x_R}\right) \tag{6.8}$$

The numerical calculation of the angles was performed before applying any fitting algorithm. Although the angle was updated based on the circle origin, the fitting algorithm did find the parameters of the circle based only on the radius and origin offsets, missing the fact that the magnitude of the angles are also affected by the calculation of the origin offsets. Therefore, the next lines show the changes needed to redesign the linear model shown in Equation (6.5).

First, the model is adapted to the principles of the dynamic angle approach. To do this, the original calculation of the angle (θ_i) given by Equation (6.3) as well as the sine and cosine identities must be replaced by an expression defined in terms of the origin offsets (x_0, y_0), which will aid the model to update the angles each time the fitting algorithm provides a new origin position. Thus, the angle (θ_i) and the trigonometric identities are symbolically calculated in terms of the *i*th point coordinate and the origin offsets, as seen in Equations (6.9) to (6.11).

$$\theta_i = \tan^{-1}\left(\frac{y_i - y_0}{x_i - x_0}\right) \tag{6.9}$$

$$\cos\theta_i = \cos\left[\tan^{-1}\left(\frac{y_i - y_0}{x_i - x_0}\right)\right] \tag{6.10}$$

$$\sin\theta_i = \sin\left[\tan^{-1}\left(\frac{y_i - y_0}{x_i - x_0}\right)\right] \tag{6.11}$$

It is important to mention that the trigonometric functions are blind regarding the position of the (x_i) and (y_i) components. This means that sometimes the sign obtained by these identities does not represent the real position of the point in space. For example: if a point is located at 225 degrees, then the sign of the trigonometric function of x=-3 and y=-3 should be negative, but if this coordinate is used, then the value obtained will be as if the point were located at 45 degrees on the first quadrant instead of the third quadrant, seen in Equation (6.12). Therefore, Equations (6.10) and (6.11) should be modified as shown in Equations (6.13) and (6.14) to avoid this problem.

$$\theta_i = \tan^{-1}\left(\frac{-3-0}{-3-0}\right) = \tan^{-1}(1) = 0.7853\,\text{radians} \cong 45^\circ \tag{6.12}$$

$$\cos\theta_i = \operatorname{sgn}(x_i)\left|\cos\left[\tan^{-1}\left(\frac{y_i - y_0}{x_i - x_0}\right)\right]\right| \tag{6.13}$$

$$\sin\theta_i = \operatorname{sgn}(y_i)\left|\sin\left[\tan^{-1}\left(\frac{y_i - y_0}{x_i - x_0}\right)\right]\right| \tag{6.14}$$

where the trigonometric identities are enclosed into an absolute value and the sine and cosine identities are multiplied by the sign of the *x* and *y* components, respectively. This provides the accurate location of the point. Finally, the linear model in Equation (6.5) is modified by replacing the constant angle identities by the dynamic angle identities, seen in Equation (6.15).

$$d_i = \sqrt{x_i^{\,2} + y_i^{\,2}} - \left[R + x_0\operatorname{sgn}(x_i)\left|\cos\left[\tan^{-1}\left(\frac{y_i - y_0}{x_i - x_0}\right)\right]\right| + y_0\operatorname{sgn}(y_i)\left|\sin\left[\tan^{-1}\left(\frac{y_i - y_0}{x_i - x_0}\right)\right]\right|\right] \tag{6.15}$$

In theory, the previous model should be more efficient than the existing model, since it takes into consideration the updated angles. However, both models can be theoretically improved if the measured radial distance is calculated using Equation (6.6) instead of Equation (6.4). Thus, the model to inspect circular parts is obtained by replacing the radial distance (r_i) by (r_{is}), as shown in Equation (6.16).

$$d_i = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} - \left[R + x_0\operatorname{sgn}(x_i)\left|\cos\left[\tan^{-1}\left(\frac{y_i - y_0}{x_i - x_0}\right)\right]\right|\right.$$

$$+\,y_0\,\operatorname{sgn}(y_i)\left|\sin\left[\tan^{-1}\left(\frac{y_i-y_0}{x_i-x_0}\right)\right]\right|\Bigg] \qquad (6.16)$$

6.2.2 Form Verification of Circular Forms (Circularity/Roundness)

The form tolerance to verify circular forms is known as circularity or roundness, and it is defined by ANSI Y14.5M-1994 (ASME 1995) as the condition in which all assessed points are equidistant from an axis and belong to a plane that is perpendicular to that axis. The circular form is evaluated using four mathematical models that follow the definition of the standards, the two models proposed by Shunmugam (1987) and the two developed models that contain the principles of the dynamic angle approach, shown in Equations (6.5) to (6.7) and Equations (6.15) and (6.16), respectively.

The magnitude of the zone of error of circular forms is obtained by subtracting the smallest negative normal deviation from maximum positive normal deviation. In this case, the residuals provided by the four mathematical models are already normalized and so there is no need to transform the data. In addition, all variables that are unrestricted in sign must be represented as a difference between two positive variables, while the positive ones should remain the same. Hence the additional constraints for the four models are:

$$\begin{aligned} &x_0 = x_0^{'} - x_o^{''} \\ &y_0 = y_0^{'} - y_o^{''} \\ &R, x_o^{'}, x_o^{''}, y_0^{'}, y_o^{''} \geq 0 \end{aligned} \qquad (6.17)$$

6.3 Spherical Form

The sphere represents challenges to the manufacturing environment due to its complex form. Several factors can cause the form to be out of the spherical shape, such as machine vibrations, tool wear, depth of cut, work misalignments, and properties of the material, among others. For this reason, it is important to study the mathematical models that describe these shapes, in order to capture as many form errors as possible.

In Chapter 2, it was mentioned that the spherical forms are evaluated by means of two commonly used mathematical models, the linear spherical model and the nonlinear spherical model. Although the sphere is a nonlinear shape by nature and the spherical nonlinear model should perform better than the linear model, Samuel et al. (2003) found slightly different results. He compared the accuracy of the nonlinear and linear spherical models against the procedure used by Fan et al. (1999). Samuel et al. (2003) applied 50 Cartesian coordinates obtained from Fan et al.'s (1999) research to the nonlinear spherical model. In addition, the 50 point coordinates were transformed into polar coordinates and applied to the linear spherical model. The zones of error obtained for the nonlinear and linear models were 7.66023 micrometers and 7.66011 micrometers, respectively; while the results obtained from Fan's research were 7.66 micrometers. Therefore, the latter shows that both models can perform in a similar way, regardless of the properties of the models. For this reason, this section focuses on transforming the linear model into a nonlinear model by applying the principles of the dynamic angle approach. Equations (6.18) and (6.19) were introduced in Chapter 2; however they are rewritten in the next lines in order to ease the reading of this paper.

$$d_i = r_i - \left[R + x_0\cos(\theta_i)\cos(\alpha_i) + y_0\sin(\theta_i)\cos(\alpha_i) + z_0\sin(\alpha_i)\right] \tag{6.18}$$

$$d_i = R - \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2} \tag{6.19}$$

The linear spherical model compares the measurements (r_i) against the ideal form composed by the parameters (R), (x_0), (y_0), and (z_0), where the last three parameters are multiplied by the calculated vertical and horizontal components of the angles (θ_i), and (α_i), where the angle (θ_i) represents the relation between the x and y components and the angle (α_i) represents the relation between the z and xy components. The measurements are calculated by applying the Euclidian distance between the ith point and the origin of the system O_s (0, 0, 0), as seen in Equation (6.20).

$$r_i = \sqrt{(x_i - 0)^2 + (y_i - 0)^2 + (z_i - 0)^2} \tag{6.20}$$

The linear spherical model has been studied by Kanada (1995), Shunmugam (1987), Samuel et al. (2003), and Balakrishna et al. (2005), who regarded the magnitude of the angles as a constant value. Kanada (1995) used a Laplace spherical function to simulate the spherical errors where the angles (θ_i), and (α_i) were generated using the variables n and m obtained from the simulation algorithm. The angle (θ_i) described the relation between the z and the xy components, while the angle (ϕ_i) defined the relation between the x and y components, shown in Equations (6.21) and (6.22).

$$\theta_{ij} = (i-1)\frac{\pi}{N} \quad (i = 1,2,\ldots,N+1) \tag{6.21}$$

$$\varphi_{ij} = (j-1)\frac{2\pi}{M} \quad (j = 1,2,\ldots,M) \tag{6.22}$$

Samuel et al. (2003) obtained the two mentioned angles by applying the concepts of trigonometry, as seen in Equations (6.23) and (6.24); where (θ_i) represented the position of the point on the *x* and *y* components, and (β_i) described the position of the point between the *z* and *xy* components. The angles were calculated by subtracting from each measured point a reference coordinate, which is obtained by selecting three points on a reference sphere and calculating the location of the origin. This location provided the origin offsets that were subtracted from the measured points, similar to the procedure used in the circular form.

$$\theta_i = \tan^{-1}\left(\frac{y_i - y_R}{x_i - x_R}\right) \tag{6.23}$$

$$\beta_i = \tan^{-1}\left(\frac{z_i - z_R}{\sqrt{(x_i - x_R)^2 + (y_i - y_R)^2}}\right) \tag{6.24}$$

Finally, Balakrishna et al. (2005) applied similar formulas to the model used by Samuel et al. (2003). The only difference is that the first authors did not subtract the reference point; they only calculated the angles based on the measured data. From Equations (6.25) and (6.26) it can be seen that (θ_i) shows the position of the point coordinate between the *x* and *y* components, while the angle (α_i) shows the position between the *z* and the *xy* components.

$$\theta_i = \tan^{-1}\left(\frac{y_i}{x_i}\right) \tag{6.25}$$

$$\alpha_i = \tan^{-1}\left(\frac{z_i}{\sqrt{x_i^2 + y_i^2}}\right) \tag{6.26}$$

Although the calculation of these angles varies according to the three mentioned authors, all of them estimated and used the angular values as a constant. This means that once the value of the angle is calculated, it will not change regardless of the value of the estimated parameters of the sphere. This procedure might induce some errors because the purpose of the sphericity algorithm is to find the radii and origin offsets that minimize the zone of error. Thus, if the location of the origin changes, then the estimated angles should change in the same proportion. This means that the sphericity algorithm should estimate the angles in a dynamic way instead of fixing them. Samuel et al. (2003) used the correct formulas since they adjusted the angle calculation by using a reference origin. However, they applied a fixed angle value on the fitting algorithm instead of using a dynamic one.

This research proposes three mathematical models to study the verification of spherical forms by applying the dynamic angle approach. The first two models are similar to the two models developed for the circular forms, while the third model is different from the other two since it does not contain trigonometric functions, as is explained below.

6.3.1 Dynamic Angle Approach Applied to Spherical Forms

The first two models are obtained by modifying the constant angles of Equation (6.18) into dynamic angles. The angles (θ_i) and (α_i), which represent the position of the data point between the x and y components and the position of the point between the z and xy components, respectively, are calculated symbolically by adding the components of the origin of the system (O_s), as shown in Equations (6.27) and (6.28) and Figure 40.

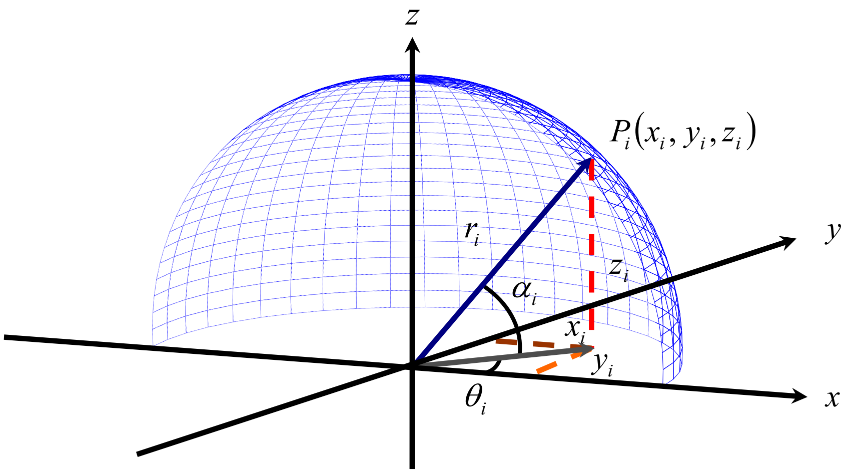


Figure 40. Parameters used in the calculation of the spherical form error.

$$\theta_i = \tan^{-1}\left(\frac{y_i - y_0}{x_i - x_0}\right) \tag{6.27}$$

$$\alpha_i = \tan^{-1}\left(\frac{z_i - z_0}{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}}\right) \tag{6.28}$$

Hence, Equations (6.29) and (6.30) show the first two models to inspect the spherical form that are found by substituting the dynamic angles and the proper sign of each trigonometric function into the linear spherical model. The difference between the two models is basically the way in which the measured radial distance is modeled, where the first model uses the (r_i) distance from Equation (6.18) and the second model uses the (r_{is}) radial distance.

$$d_i = \sqrt{x_i^2 + y_i^2 + z_i^2} - \Big[R$$

$$+ x_0 \operatorname{sgn}(x_i)\left|\cos\left[\tan^{-1}\left(\frac{y_i - y_0}{x_i - x_0}\right)\right]\cos\left[\tan^{-1}\left(\frac{z_i - z_0}{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}}\right)\right]\right|$$

$$+ y_0 \operatorname{sgn}(y_i) \left| \sin\left[\tan^{-1}\left(\frac{y_i - y_0}{x_i - x_0} \right) \right] \cos\left[\tan^{-1}\left(\frac{z_i - z_0}{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}} \right) \right] \right|$$

$$+ z_0 \operatorname{sgn}(z_i) \left| \sin\left[\tan^{-1}\left(\frac{z_i - z_0}{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}} \right) \right] \right| \Bigg] \tag{6.29}$$

$$d_i = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2} - \Bigg[R$$

$$+ x_0 \operatorname{sgn}(x_i) \left| \cos\left[\tan^{-1}\left(\frac{y_i - y_0}{x_i - x_0} \right) \right] \cos\left[\tan^{-1}\left(\frac{z_i - z_0}{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}} \right) \right] \right|$$

$$+ y_0 \operatorname{sgn}(y_i) \left| \sin\left[\tan^{-1}\left(\frac{y_i - y_0}{x_i - x_0} \right) \right] \cos\left[\tan^{-1}\left(\frac{z_i - z_0}{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}} \right) \right] \right|$$

$$+ z_0 \operatorname{sgn}(z_i) \left| \sin\left[\tan^{-1}\left(\frac{z_i - z_0}{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}} \right) \right] \right| \Bigg] \tag{6.30}$$

6.3.2 Model of Spherical Form without Trigonometric Identities

Following the second solution proposed in Section 6.2 to improve the form verification models, the third model is constructed by removing all trigonometric functions in order to determine if these are causing inaccuracies in the model. Firstly, according to Weisstein (1999) the definition of the cosine of an angle is given by the relation between the adjacent side of the triangle and the hypotenuse, while the sine is given by the relation between the opposite side of the triangle and the hypotenuse. The cosine and sine functions of the *i*th point are defined as shown in Figure 41 and Equations (6.31) to (6.34).

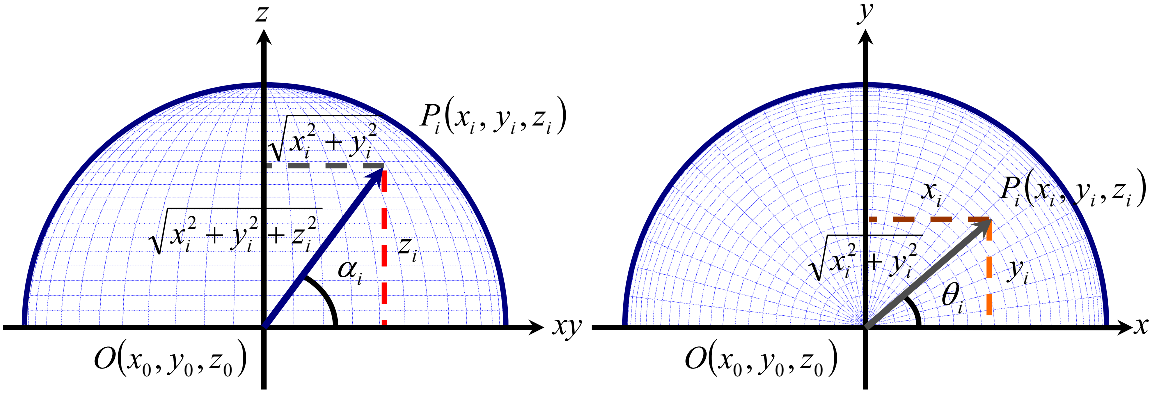


Figure 41. Cartesian and trigonometric components of the *i*th point.

$$\cos(\theta_i) = \frac{(x_i - x_0)}{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}} \tag{6.31}$$

$$\sin(\theta_i) = \frac{(y_i - y_0)}{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}} \tag{6.32}$$

$$\cos(\alpha_i) = \frac{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}}{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2}} \tag{6.33}$$

$$\sin(\alpha_i) = \frac{(z_i - z_0)}{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2}} \tag{6.34}$$

Then, each of these Equations is substituted in the linear model shown in Equation (6.18), which provides a model without trigonometric functions, as seen in Equation (6.35).

$$d_i = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2}$$

$$-\left[R + x_0 \frac{(x_i - x_0)}{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}} \frac{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}}{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2}}\right.$$

$$+ y_0 \frac{(y_i - y_0)}{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}} \frac{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}}{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2}}$$

$$\left. + z_0 \frac{(z_i - z_0)}{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2}}\right] \tag{6.35}$$

From the previous Equation, the middle terms of (x_0) and (y_0) can be simplified by canceling out the Euclidian distances between *x* and *y*. Thus, the simplified model is shown in Equation (6.36).

$$d_i = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2}$$

$$-\left[R + x_0 \frac{(x_i - x_0)}{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2}}\right.$$

$$+ y_0 \frac{(y_i - y_0)}{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2}}$$

$$\left. + z_0 \frac{(z_i - z_0)}{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2}}\right] \tag{6.36}$$

Finally, it can be seen that the terms that model the origin offsets contain the Euclidian distance between *x*, *y* and *z*. Hence, if the right side is representing the ideal form then all of these distances can be substituted by the ideal radius of the sphere (*R*), which simplifies the model even more, as seen in Equation (6.37).

$$d_i = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2}$$

$$-\left[R+\frac{1}{R}\left[x_0(x_i-x_0)+y_0(y_i-y_0)+z_0(z_i-z_0)\right]\right] \tag{6.37}$$

6.3.3 Form Verification of Spherical Form (Sphericity)

The form verification of spherical forms is also known as sphericity. In the present research, three sphericity models have been developed that analyze the zone of error of spherical forms in different ways. However, they all provide residuals that are normal to the surface. Thus, the zone of error is obtained in a straightforward way. The additional constraints needed for the above models are:

$$\begin{aligned}
&x_0 = x_0^{'} - x_0^{''}\\
&y_0 = y_0^{'} - y_0^{''}\\
&z_0 = z_0^{'} - z_0^{''}\\
&R, x_0^{'}, x_0^{''}, y_0^{'}, y_0^{''}, z_0^{'}, z_0^{''} \geq 0
\end{aligned} \tag{6.38}$$

6.4 Cylindrical Form

The cylinder is created by joining together both ends of a 3-Dimensional plane where every point on the surface is equidistant to a central axis, and its upper and lower ends are bounded by two parallel planes. The parameters of the cylinder include the radius (*r*), which is the perpendicular distance from the central axis to the surface of the cylinder; and the height of the cylinder (*h*); Figure 42.

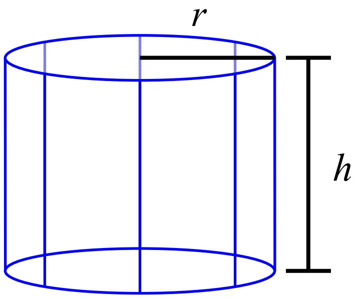


Figure 42. Parameters used in the analysis of the cylindrical form.

The height is not as important as the radius because the cylindrical form does not depend on the height; it only depends on the magnitude of the radius. The cylinder was studied by Shunmugam (1987), Lai et al. (1996), Carr et al. (1995), Narayannan Namboothiri (1998), and Prakasvudhisarn (2002). Shunmugam (1987), Narayannan Namboothiri (1998) and Prakasvudhisarn (2002) studied the cylindrical form using two models: they all used the same linear model that is improved in the present research, and they used two different nonlinear models. Lai et a. (1996) studied the cylindrical form by transforming it into a planar surface, while Carr et al. (1995) verified the cylinder deviations using two approaches: one by extracting the parameters by fitting the measured data into a reference cylinder and the other by fitting the data into a common cylindrical axis. The results obtained from Carr's (1995) and Prakasvudhisarn's (2002) nonlinear models are compared against the nonlinear model developed in the present research.

The present research proposes two models to inspect the cylindrical form. The first model adapts the dynamic angle approach to a model studied by Shunmugam (1987), Narayannan Namboothiri (1998), and Prakasvudhisarn (2002), and the second model is a nonlinear approach that uses information from a 3 dimensional point coordinate to inspect the cylinder with spherical radial distances.

6.4.1 Model of the Cylindrical Form using the Spherical and the Dynamic Angle Approaches

The linear model shown in Equation (6.39) was used by Shunmugam (1987), Narayannan Namboothiri (1998), and Prakasvudhisarn (2002). The model uses the (x_i) and (y_i) components of the *i*th point coordinate and transforms the data into radial distances that depend on the axis of revolution of the cylinder. Although the manufactured parts are created using computer controlled machines, they are not error free. Thus, the model has to compensate for any axis misalignments due to the machining process or induced by the inspection process. The axis misalignment is represented by the vector of the cylinder axis [*m*, *n*, 1], which is the normal vector of the cylindrical feature. Then the model analyzes the measured data in order to find the inclination of the cylindrical form that minimizes the form of error.

$$d_i = r_i - \left[r + mz_i\cos(\theta_i) + x_0\cos(\theta_i) + nz_i\sin(\theta_i) + y_0\sin(\theta_i)\right] \tag{6.39}$$

where the angle (θ_i) defines the location of the point on the *xy* plane. Therefore, if this angle is defined as a dynamic angle (Equation 6.40) then the modified cylindrical model with the dynamic angle approach principle will look like Equations (6.41) and (6.42).

$$\theta_i = \tan^{-1}\left(\frac{y_i - y_0}{x_i - x_0}\right) \tag{6.40}$$

$$d_i = \sqrt{x_i^2 + y_i^2} - \left[r + \left(mz_i + x_0\right)\mathrm{sgn}\left(x_i\right)\left|\cos\left(\tan^{-1}\left(\frac{\left(y_i - y_0\right)}{\left(x_i - x_0\right)}\right)\right)\right| \right.$$

$$\left. + \left(nz_i + y_0\right)\mathrm{sgn}\left(y_i\right)\left|\sin\left(\tan^{-1}\left(\frac{\left(y_i - y_0\right)}{\left(x_i - x_0\right)}\right)\right)\right| \right] \qquad (6.41)$$

$$d_i = \sqrt{\left(x_i - x_0\right)^2 + \left(y_i - y_0\right)^2} - \left[r \right.$$

$$+ \left(mz_i + x_0\right)\mathrm{sgn}\left(x_i\right)\left|\cos\left(\tan^{-1}\left(\frac{\left(y_i - y_0\right)}{\left(x_i - x_0\right)}\right)\right)\right|$$

$$\left. + \left(nz_i + y_0\right)\mathrm{sgn}\left(y_i\right)\left|\sin\left(\tan^{-1}\left(\frac{\left(y_i - y_0\right)}{\left(x_i - x_0\right)}\right)\right)\right| \right] \qquad (6.42)$$

6.4.2 Model of the Cylindrical Form with Free Form Orientation

The nonlinear model presented in this section studies the form deviations of the cylindrical form by using information from the three components of the *i*th point. This means that the cylinder is studied using a finite number of spheres or a finite number of radial distances that are transformed into normal deviations.

From Figure 43, it can be seen that the radius of the cylinder of the *i*th point (r_{hi}) is given by multiplying the radial distance (r_{is}) times the cosine of the angle (α_i). However, the angle (α_i) is not a constant term for all radial distances (r_i). Thus the angle (α_i) is obtained with the inverse tangent of the relation between the vertical component (z_i) and the ideal radius of the cylinder (r), as seen in Equations (6.43) and (6.44).

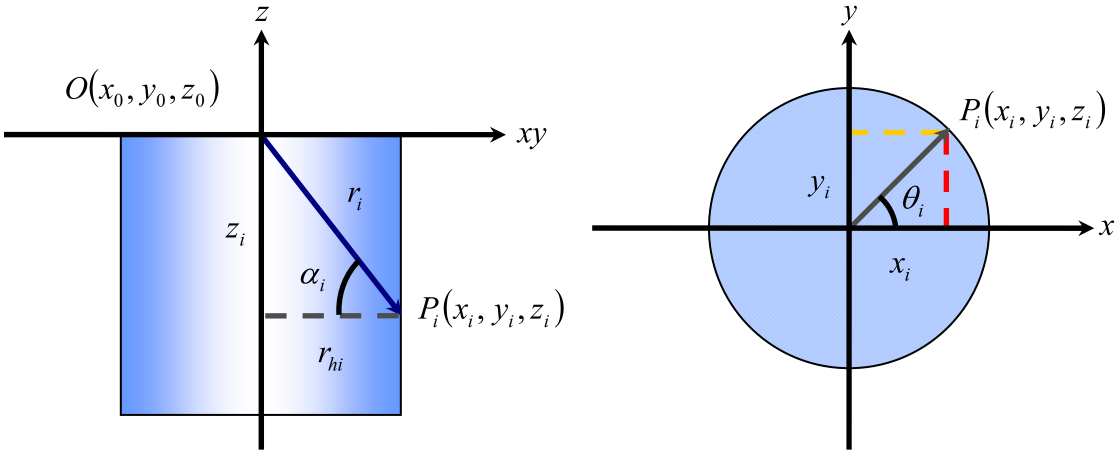


Figure 43. Front and top views of the free form orientation model for cylindrical forms.

$$\alpha_i = \tan^{-1}\left(\frac{z_i}{r}\right) \tag{6.43}$$

$$r_{hi} = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2}\cos\left(\tan^{-1}\left(\frac{z_i}{r}\right)\right) \tag{6.44}$$

The relation shown in Equation (6.44) assumes that the axis of the cylinder is perfectly aligned with the *z*-axis of the inspection equipment. Thus, it is necessary to compensate for any axis misalignment. The (z_i) used to calculate the angle (α_i) must be corrected in order to apply the free form orientation approach to the model. A simple way to do this is to apply the flatness model developed by Shunmugam (1987) to each of the (z_i)'s used in the calculation of the angle, as shown in Equations (6.45) and (6.46), respectively. By doing this, the model rectifies the value of (z_i) according to the inclination of the axis of the cylinder and calculates the normal vector of each *i*th plane that minimizes the zone of error. Therefore, the model can find the measured points in any location of 3-dimensional space.

$$z_{fi} = \frac{z_i - (mx_i + ny_i + z_0)}{\sqrt{m^2 + n^2 + 1}} \tag{6.45}$$

$$r_{hi} = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2} \cos\left(\tan^{-1}\left(\frac{z_i - (mx_i + ny_i + z_0)}{r\sqrt{m^2 + n^2 + 1}} \right)\right) \tag{6.46}$$

In addition, it can be seen from Equation (6.44) that the (z_i) contained in the Euclidian distance is not affected by the flatness model, due to the fact that the distance obtained from that calculation belongs to the cylinder surface regardless of the inclination of the cylindrical form. Hence, the only (z_i) affected is the one used to calculate the angle (α_i). Finally, the nonlinear model to verify the form deviations of the cylindrical form is obtained by subtracting the radius of the cylinder (r) from the radius of the cylinder of the *i*th point (r_{hi}), as seen in Equation (6.47) and in Figure 44.

$$d_i = \left[\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2} \cos\left(\tan^{-1}\left(\frac{z_i - (mx_i + ny_i + z_0)}{r\sqrt{m^2 + n^2 + 1}} \right)\right) \right] - r \tag{6.47}$$

The advantages of using this model compared to the one above are that:

- The model can find the cylindrical feature anywhere in the 3-dimensional space.
- The cylinder axis does not have to align with the z-axis of the inspection equipment, and
- Complete information from the *i*th point coordinate is being used to extract the parameters from the data.

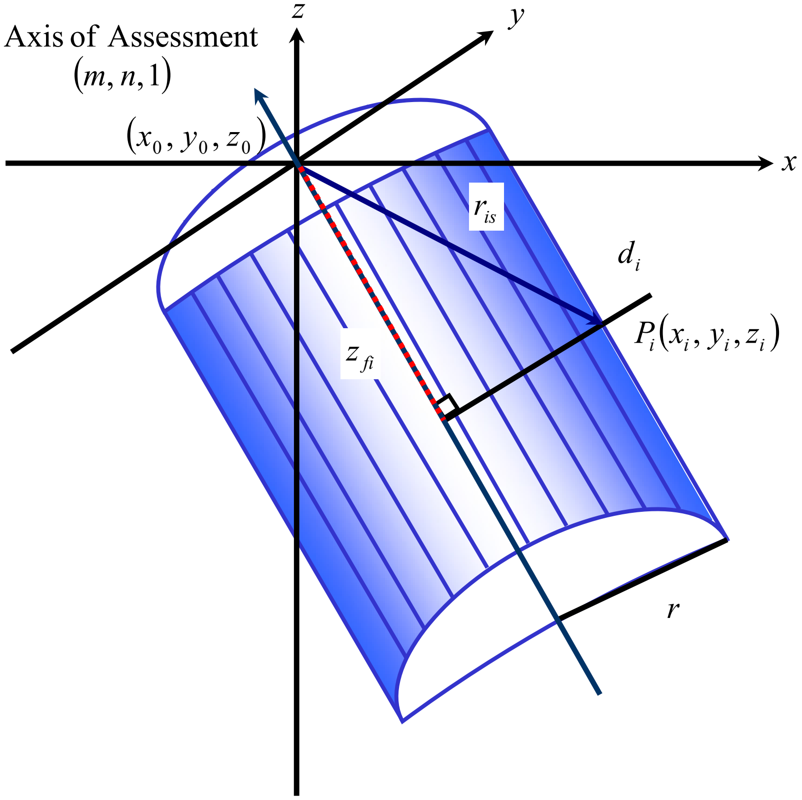


Figure 44. Free Form Orientation approach applied to the study of cylindricity.

6.4.3 Form Verification of Cylindrical Forms (Cylindricity)

The form tolerance zone of cylindrical forms is also known as cylindricity, which is studied in the present research using three mathematical models shown above. The residuals provided by the three models are normal to the surface, so there is no need to modify them. The additional constraints needed to solve the cylindricity models are defined as follows:

$$\begin{aligned}
&x_0 = x_0^{'} - x_0^{''} \\
&y_0 = y_0^{'} - y_0^{''} \\
&z_0 = z_0^{'} - z_0^{''} \\
&m = m^{'} - m^{''} \\
&n = n^{'} - n^{''} \\
&r, x_0^{'}, x_0^{''}, y_0^{'}, y_0^{''}, z_0^{'}, z_0^{''}, m^{'}, m^{''}, n^{'}, n^{''} \geq 0
\end{aligned} \qquad (6.48)$$

6.5 Complete Conical Form

The conical form can be found in several mechanical assemblies in two common configurations: the complete conical form and the conical frustum form. The latter feature is the most used in industry and it is also known as tapered cone, which was studied by Chen et al. (2000), Liu et al. (2004), and Prakasvudhisarn et al. (2002), while Charterjee et al. (1998) and Prakasvudhisarn et al. (2002) studied the complete conical form. The present research analyzes both conical forms in a separate way, where two different mathematical models are used to describe the form of these shapes.

The complete conical form is studied using two brand new models. The first developed model uses the spherical form and the dynamic angle approach to inspect the form of the cone. This modification makes the model less sensitive to origin misalignment, but the only disadvantage is that the axis of the conical form needs to be aligned as close as possible with the *z*-axis of the inspection equipment. For this reason, a new nonlinear model is developed that applies the free form orientation principles that were adapted to the cylindrical form.

6.5.1 Model of the Complete Conical Form using the Spherical and the Dynamic Angle Approaches

The mathematical analysis of the complete conical form requires the use of the three Cartesian components of the measured data, which are used to extract the parameters of the cone. The main idea behind the model is to measure the cone as a finite number of virtual spheres. The origin of the cone and the origin of these spheres are located at the vertex of the measured cone. Thus, each measurement (P_i) provides a radial distance from the vertex of the cone (origin of the sphere) to the location of the point (P_i) (which lies on the surface of the cone and on the surface of the virtual sphere). Since the cone is measured (n) number of times, then each distance can be represented as a radial distance. For this reason, the analysis of the cone is performed as if it were composed of a set of (n) virtual spheres of different radii, as seen in Figure 45. The radii of the virtual spheres can be transformed into the cone deviations that are normal to the surface by applying the principles of analytic geometry, trigonometry, and nonlinear programming algorithms.

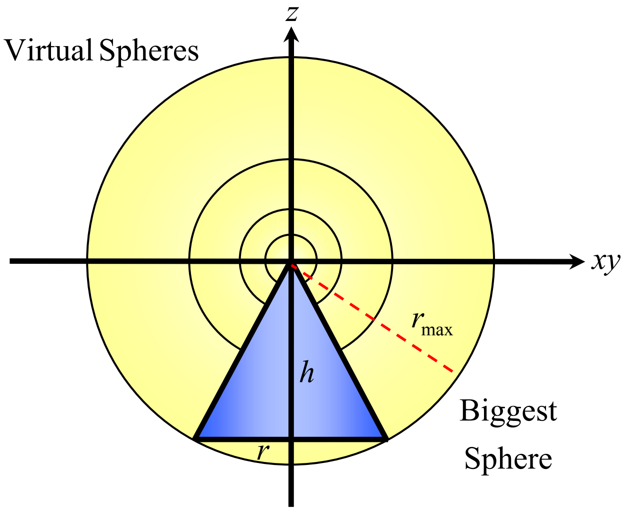


Figure 45. Virtual spheres superimposed on the conical form.

The conical form has three main parameters: the height of the cone (*h*), the radius of the base of the cone (*r*), and the angle (γ_i) that determines the relation between the height of the cone and the length of its base. In addition, there is one more parameter that can be obtained based on the magnitude of the parameters mentioned above. The angle (α_i) together with the 90-degree angle and the angle (γ_i) creates the internal angles of one of the triangles that composed the cross section of the cone. Figure 46 portrays the cone in the form of a kite. The shape of the cone is the top part described as a shaded cross section, while the bottom of the kite is used to calcualte the normal deviations of the cone.

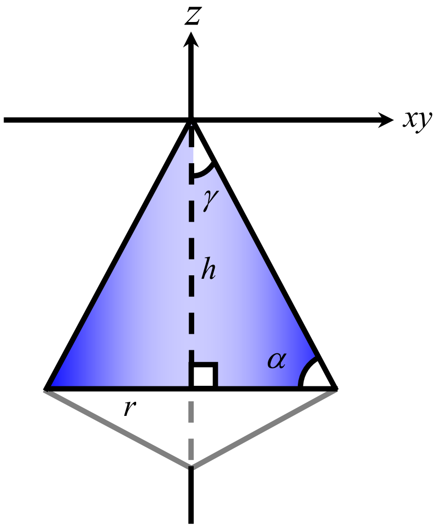


Figure 46. Parameters used in the analysis of the complete conical form.

The geometrical deviations of the cone (d_i) are found by comparing the surface measurements of the manufactured cone against an ideal cone surface. In this model, the surface measurements are converted into radial distances (r_i) that are compared against calculated ideal radial distances (r_{Si}), as seen below.

$$d_i = r_i - r_{Si} \tag{6.49}$$

The proposed model is explained by using an example based on a single inspection point, which is divided into two parts. The first part is concerning the definition of the ideal identities needed to perform the analysis, while the second part concerns the application of those identities to develop the mathematical model that estimates the cone deviations.

The calculation of the ideal distances requires the analysis of five perfect triangles that are obtained from the kite shape explained above. From Figure 47, it can be seen that the first three triangles share the same proportions of its sides, this is the reason why they all have the same magnitude of their angles. The first triangle can be described as one half of the cross section of the conical form and has a height (v_1) equal to (h), and a radius of the base of the cone given by (r). The hypotenuse of this triangle is the maximum ideal radial distance ($r_{i\max}$), which is the radius of the biggest sphere that can be generated and can be obtained symbolically by applying the Pythagorean Theorem. In a similar way, the relations of the angle (γ) are obtained by using the trigonometric identities, as seen in Equations (6.50) to (6.54).

$$v_1 = h \tag{6.50}$$

$$r_{i\max} = \sqrt{h^2 + r^2} \tag{6.51}$$

$$\sin\gamma = \frac{r}{\sqrt{h^2 + r^2}} \tag{6.52}$$

$$\cos\gamma = \frac{h}{\sqrt{h^2 + r^2}} \tag{6.53}$$

$$\tan\gamma = \frac{r}{h} \tag{6.54}$$

The second triangle is located under the first triangle and is created by using the radius of the base of the cone and the ideal maximum normal distance of the cone ($a_{i\max}$). Equation (6.55) shows the calculation of the hypotenuse ($a_{i\max}$) of this triangle using a different relation than the one develop in Equation (6.53).

$$\cos\gamma = \frac{r}{a_{i\max}}$$

$$a_{i\max} = \frac{r}{\cos\gamma} \tag{6.55}$$

Accordingly, the vertical side (v_2) can be obtained symbolically in terms of the tangent of the angle (γ), as seen in Equation (6.56).

$$\tan\gamma = \frac{v_2}{r}$$

$$v_2 = r\tan\gamma \tag{6.56}$$

The third triangle is located at the right of the first triangle and is created by using the ideal maximum radius of the virtual sphere ($r_{i\max}$) and the maximum normal distance of the cone ($a_{i\max}$). The height of this triangle is obtained by adding together the height of the first and second triangles, seen in Equation (6.57).

$$v_3 = v_1 + v_2$$

$$v_3 = h + r\tan\gamma \tag{6.57}$$

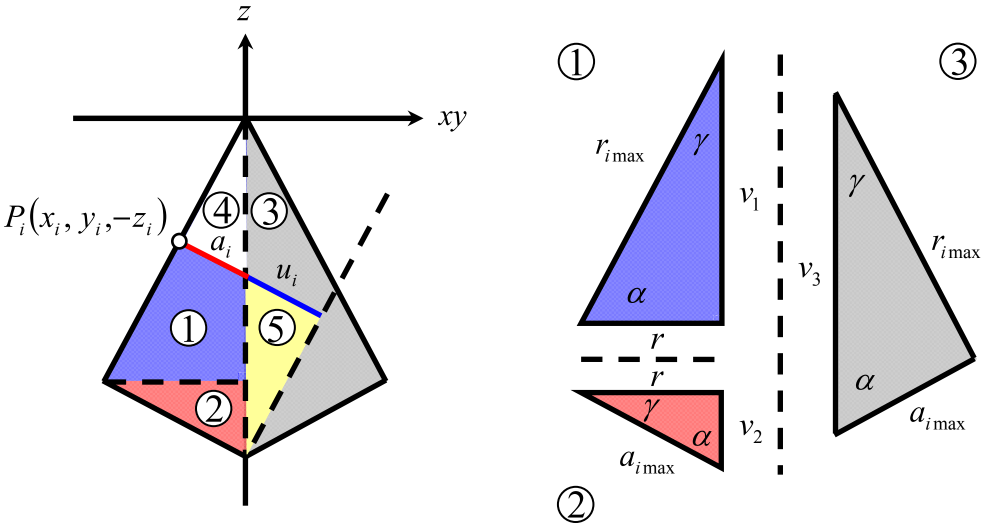


Figure 47. Right triangles used to analyze the complete conical form.

The fourth and fifth triangles are generated according to the location of the *i*th measuring point. Thus, the explanation of these triangles is included in the example shown in the next lines. The measuring point P_i (x_i, y_i, -z_i) used in this example has a negative (z_i) due to the fact that the point is located under the origin located at the vertex. It can be seen from Figure 47 that the location of the point (P_i) generates the fourth and fifth triangles, which are similar in proportions as the other three triangles.

The measurement of the *i*th point is converted into a radial distance (r_{is}) or radius of the *i*th virtual sphere by applying the Euclidian distance principle, where the variables that control the origin offsets are introduced to the formula in order to compensate for those misalignments, similarly as it was done in the previous geometric shapes shown above.

$$r_i = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2} \tag{6.58}$$

Equation (6.58) is compared against the ideal radial distance (r_{Si}), which is obtained by following the next steps. Firstly, the mathematical formula that describes the ideal radial distance must be defined using the parameters of complete conical form in order to use them to extract the parameters of the assessment cone. Therefore, the ideal radius of the *i*th sphere (r_{Si}) is calculated by transforming the ideal normal distances (a_i) of the fourth triangle into radial distances, which can be achieved by multiplying each (a_i) by the inverse of Equation (6.54), shown in Equation (6.59).

$$r_{Si} = \left(\frac{1}{\tan \gamma} \right) a_i$$

$$r_{Si} = \left(\frac{h}{r} \right) a_i \tag{6.59}$$

The second step is to obtain the value of the ideal normal distance (a_i) by subtracting the complementary ideal normal distance (u_i) of the fifth triangle from the ideal maximum normal distance ($a_{i\max}$) obtained in Equation (6.55), seen in Equation (6.60).

$$r_{Si} = \left(\frac{h}{r} \right) \left[a_{i\max} - u_i \right]$$

$$r_{Si} = \left(\frac{h}{r} \right) \left[\frac{r}{\cos \gamma} - u_i \right] \tag{6.60}$$

The third step is to calculate the vertical side of the fifth triangle, which is going to be used to obtain the value of (u_i). From Figure 48, it can be seen that the height of the fifth triangle is obtained by the height of the first (v_1), second (v_2), third (v_3), and fourth (v_4) triangles. The height of the first, second, and third triangles was already obtained in

Equations (6.50), (6.56), and (6.57). The only variable remaining is the height of the fourth triangle that is obtained by transforming the third Cartesian component (z_i) into a height that represents the total height of the kite or height of the third triangle. The main reason that this transformation needs to be done is because the ideal normal distances (a_i) are created using the height of the third triangle instead of using only the original height of cone (v_1). Therefore, the height of the fourth triangle is calculated by using the similarities between the first and third triangles, where the (z_0) component is added to the (z_i) value in order to counteract the effect of the origin offsets, as seen in Equation (6.61).

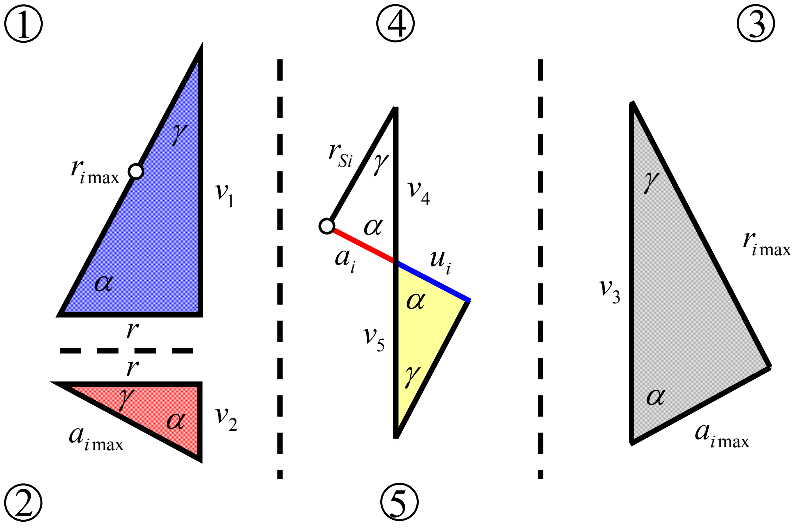


Figure 48. Ideal triangles used to calculate the cone deviations.

$$v_4 = \left(\left|z_i - z_0\right|\right)\frac{v_3}{v_1}$$

$$v_4 = \left(\left|z_i - z_0\right|\right)\frac{\left(r\tan\gamma + h\right)}{h} \tag{6.61}$$

Thus, the height of the fifth triangle is determined by two terms, which is the height of the second triangle (Equation 6.56) and the difference between the height of the first triangle (v_1) minus the height of the fourth triangle (v_4), as it is described in Equation (6.62).

$$v_5 = r \tan\gamma + (v_1 - v_4)$$

$$v_5 = r \tan\gamma + \left[h - \left[(|z_i - z_0|) \frac{(r \tan\gamma + h)}{h} \right] \right] \tag{6.62}$$

From the first triangle, it is known that the tangent of the angle (γ) is given by the ratio between the height of the cone (h) and the radius of the base of the cone (r). Thus, Equation (6.62) can be simplified by implementing Equation (6.54)

$$v_5 = r\left(\frac{r}{h}\right) + \left[h - \left[(|z_i - z_0|) \frac{\left(r\left(\frac{r}{h}\right) + h \right)}{h} \right] \right]$$

$$v_5 = \frac{r^2}{h} + \left[h - \left[(|z_i - z_0|) \frac{\left(\frac{r^2}{h} + h \right)}{h} \right] \right] \tag{6.63}$$

The fourth step is to calculate the value (u_i), which is given by applying the sine identity to the fifth triangle and solving for (u_i), as described in Equations (6.64) and (6.65).

$$\sin\gamma = \frac{u_i}{v_5} \tag{6.64}$$

$$u_i = \left[\frac{r^2}{h} + \left[h - \left[\left(\left| z_i - z_0 \right| \right) \frac{\left(\frac{r^2}{h} + h \right)}{h} \right] \right] \right] \sin\gamma \tag{6.65}$$

Equation (6.65) can be simplified by substituting Equation (6.52) that was obtained from the first triangle, as seen below.

$$u_i = \left[\frac{r^2}{h} + \left[h - \left[\left(\left| z_i - z_0 \right| \right) \frac{\left(\frac{r^2}{h} + h \right)}{h} \right] \right] \right] \frac{r}{\sqrt{h^2 + r^2}} \tag{6.66}$$

Finally, the fifth step is to substitute Equations (6.53) and (6.66) in Equation (6.60) in order to obtain the ideal value of the radial distances in terms of the parameters of the cone. Equation (6.67) shows the final equation, which was algebraically simplified in order to reduce the computing time.

$$r_{Si} = \left(\frac{h}{r} \right) \left[\frac{r\sqrt{h^2 + r^2}}{h} - \left[\left[\frac{r^2}{h} + \left[h - \left[\left(\left| z_i - z_0 \right| \right) \frac{\left(\frac{r^2}{h} + h \right)}{h} \right] \right] \right] \frac{r}{\sqrt{h^2 + r^2}} \right] \right]$$

$$r_{Si} = \left[\sqrt{h^2 + r^2} - \left[\frac{\left[r^2 + h^2 - \left[\left(\left| z_i - z_0 \right| \right) \left(\frac{r^2}{h} + h \right) \right] \right]}{\sqrt{h^2 + r^2}} \right] \right]$$

$$r_{Si} = \frac{h^2 + r^2 - \left[r^2 + h^2 - \left[\left(\left| z_i - z_0 \right| \right) \left(\frac{r^2 + h^2}{h} \right) \right] \right]}{\sqrt{h^2 + r^2}}$$

$$r_{Si} = \left[\frac{h^2 + r^2}{h\sqrt{h^2 + r^2}} \right] \left(\left| z_i - z_0 \right| \right) \tag{6.67}$$

Equation (6.68) shows the calculation of the residuals of the cone that are obtained by substituting Equations (6.58 and 6.67) in Equation (6.49).

$$d_i = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2} - \left[\left[\frac{h^2 + r^2}{h\sqrt{h^2 + r^2}} \right] \left(\left| z_i - z_0 \right| \right) \right] \tag{6.68}$$

The residuals obtained by Equation (6.68) are not the true normal deviations of the conical surface, since they come from the radial distances. Hence, in order to generate the normal deviations, Equation (6.68) should be multiplied by Equation (6.54), which is a relation that comes from the third triangle, as seen in Equation (6.69).

$$d_i = \frac{r}{h} \left[\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2} - \left[\left[\frac{h^2 + r^2}{h\sqrt{h^2 + r^2}} \right] \left(\left| z_i - z_0 \right| \right) \right] \right] \tag{6.69}$$

The mathematical relation shown in Equation (6.69) to determine the normal deviations of the cone can be used to verify the conical form under the assumption that the manufacturing process used to machine the part is the only source of error. This means that in an ideal scenario with no misalignments from the inspection instrument, this relation will provide the width of the zone of error. However, this is not possible since it was mentioned in Section 2.5 of Chapter 2 that the inspection instruments induce different types of errors that can affect the output of the inspection process. Moreover, at the beginning of this section, it was explained that the complete conical form is analyzed using a finite number of spheres and for this reason, it is necessary to add the last terms employed in the sphericity model that counteract the effects of origin offsets of the spheres. However, there are some differences between the last terms of the spherical model and the conical model because the latter has the angle (*γ*) that is constant for all data points, therefore the last term of each origin offset must be changed from cosine to sine and from sine to cosine, as see in Equation (6.70) and (6.71).

$$\begin{aligned} d_i = \frac{r}{h}\Bigg[& \sqrt{x_i^{\,2}+y_i^{\,2}+z_i^{\,2}} - \Bigg[\Bigg[\Bigg[\frac{h^2+r^2}{h\sqrt{h^2+r^2}}\Bigg]\big(|z_i|-z_0\big)\Bigg] \\ & + x_0\,\mathrm{sgn}(x_i)\Bigg|\cos\Bigg[\tan^{-1}\bigg(\frac{y_i-y_0}{x_i-x_0}\bigg)\Bigg]\sin\Bigg[\pi-\tan^{-1}\bigg(\frac{r}{h}\bigg)\Bigg]\Bigg| \\ & + y_0\,\mathrm{sgn}(y_i)\Bigg|\cos\Bigg[\tan^{-1}\bigg(\frac{y_i-y_0}{x_i-x_0}\bigg)\Bigg]\sin\Bigg[\pi-\tan^{-1}\bigg(\frac{r}{h}\bigg)\Bigg]\Bigg| \\ & + z_0\,\mathrm{sgn}(z_i)\Bigg|\cos\Bigg[\pi-\tan^{-1}\bigg(\frac{r}{h}\bigg)\Bigg]\Bigg|\Bigg]\Bigg] \end{aligned} \tag{6.70}$$

$$\begin{aligned} d_i = \frac{r}{h}\Bigg[& \sqrt{(x_i-x_0)^2+(y_i-y_0)^2+(z_i-z_0)^2} - \Bigg[\Bigg[\Bigg[\frac{h^2+r^2}{h\sqrt{h^2+r^2}}\Bigg]\big(|z_i|-z_0\big)\Bigg] \\ & + x_0\,\mathrm{sgn}(x_i)\Bigg|\cos\Bigg[\tan^{-1}\bigg(\frac{y_i-y_0}{x_i-x_0}\bigg)\Bigg]\sin\Bigg[\pi-\tan^{-1}\bigg(\frac{r}{h}\bigg)\Bigg]\Bigg| \\ & + y_0\,\mathrm{sgn}(y_i)\Bigg|\cos\Bigg[\tan^{-1}\bigg(\frac{y_i-y_0}{x_i-x_0}\bigg)\Bigg]\sin\Bigg[\pi-\tan^{-1}\bigg(\frac{r}{h}\bigg)\Bigg]\Bigg| \\ & + z_0\,\mathrm{sgn}(z_i)\Bigg|\cos\Bigg[\pi-\tan^{-1}\bigg(\frac{r}{h}\bigg)\Bigg]\Bigg|\Bigg]\Bigg] \end{aligned} \tag{6.71}$$

6.5.2 Model of the Complete Conical Model with Free Form Orientation

The model presented in Equation (6.72) behaves similar as the cylindrical model shown before, in which the (z_i) component of Equation (6.69) is substituted by the flatness model shown in Equation (6.45) in order to compensate for any axis misalignments and origin offsets. Figure 49 presents a graphic representation of Equation (6.72).

$$d_i = \frac{r}{h}\left[\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2} - \left[\left(\frac{h^2 + r^2}{h\sqrt{h^2 + r^2}}\right)\left(\left|\frac{z_i - (mx_i + ny_i + z_0)}{\sqrt{m^2 + n^2 + 1}}\right|\right)\right]\right] \tag{6.72}$$

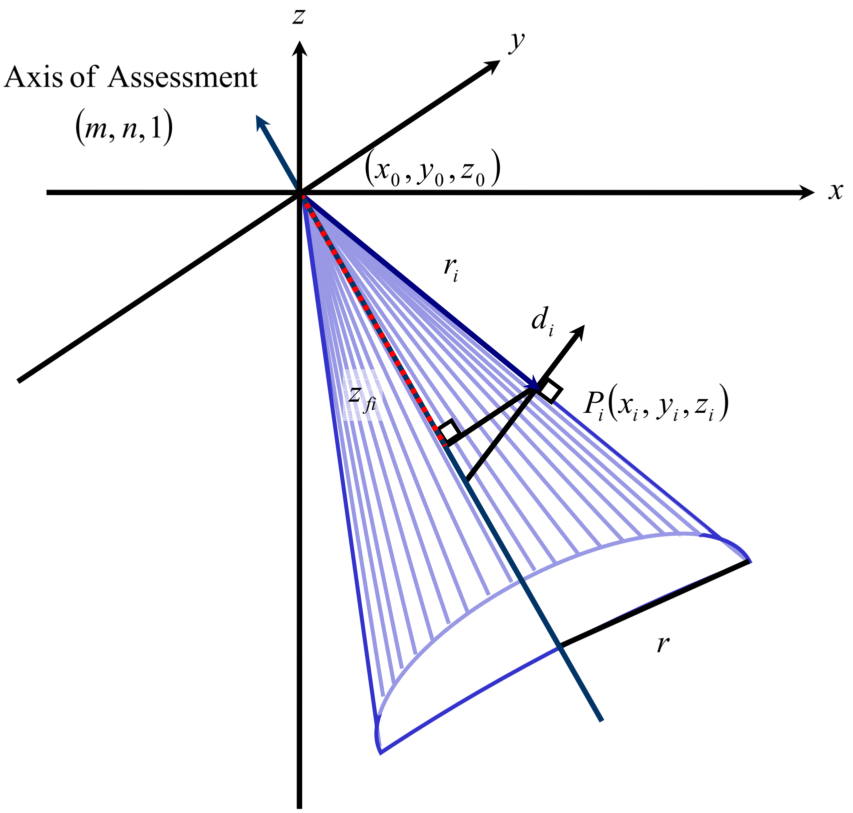


Figure 49. Free Form Orientation approach applied to the study of conical forms.

6.5.3 Form Verification of Conical Forms (Conicity)

The form verification of conical forms is also known as conicity, which is study in the present research using three mathematical models shown above. The residuals provided by the three models are normal to the surface, this is the reason why there is no need to modify them. The additional constraints needed to solve the conicity models are defined as follow:

$$\begin{aligned} x_0 &= x_0^{'} - x_0^{''} \\ y_0 &= y_0^{'} - y_0^{''} \\ z_0 &= z_0^{'} - z_0^{''} \\ m &= m^{'} - m^{''} \\ n &= n^{'} - n^{''} \\ R, h, r, x_0^{'}, x_0^{''}, y_0^{'}, y_0^{''}&, z_0^{'}, z_0^{''}, m^{'}, m^{''}, n^{'}, n^{''} \geq 0 \end{aligned} \tag{6.73}$$

6.6 Conical Frustum Form

The conical frustum is defined as a cone that has the top part sliced, or in other words is a conical section bounded by two parallel planes (Weisstein 1999). Although, the complete cone is more recognizable than the frustum form, the frustum is more utilized in industrial equipment in the form of conical bearings, conical fixtures, and in some other parts in the form of chamfers. The frustum form can be evaluated in a similar way as the verification of complete conical form. However, it is necessary to take into consideration that the top part does not exits, and thus making impossible for the inspection equipment such as the coordinate measuring machine (CMM) to set its origin in the same position as the vertex of the cone. For this reason, the formulas developed for complete conical forms should be modified so that can be suitable to verify conical frustums. The frustum form is analyzed using two models. The first model uses the principles of the dynamic angle approach and represents the frustum deviation using a finite number of spheres. The second model uses the free form orientation approach that allows the model to extract the parameters of the frustum by searching the feature anywhere in the 3-dimensional space.

6.6.1 Model of the Conical Frustum Form using the Spherical and the Dynamic Angle Approaches

The parameters of the conical frustum form are slightly different than the ones used to describe the complete conical form. The former has six parameters where three of them are the same parameters as the complete conical form, such as the radius of the base of the frustum given by (r), and the angles (α_i, γ_i) used to describe the proportions of the sides of the frustum. The remaining three parameters are: the radius of the top plane given by (r'), the height of the frustum given by (h''), and the height of the peak (h') given by the distance between the top plane and the imaginary vertex of the frustum, as seen in Figure 50.

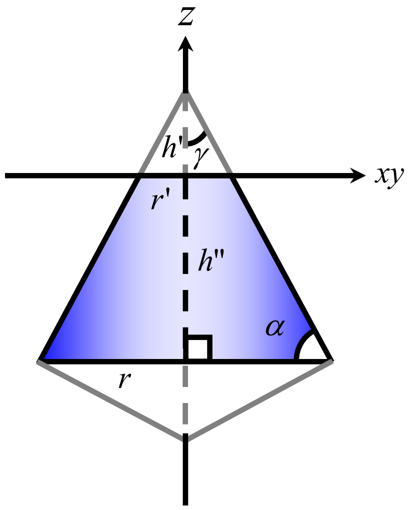


Figure 50. Parameters used in the analysis of the frustum form.

The deviations of the conical frustum (d_i) are calculated in a similar way as the deviations of the complete conical form, where virtual spheres are used to calculate the measured radial distance (r_i) and the ideal radial distance (r_{Si}). However, the frustum form does not posses a physical vertex, which makes it impossible to locate the origin of the measuring instrument in that location. Hence, the origin is positioned at the center of the top plane, which causes some minor changes to the analysis.

The mathematical analysis of the frustum form requires the calculation of eight ideal triangles and one ideal rectangle that are going to be used to generate a symbolic equation in terms of the parameters of the frustum in order to determine the ideal radial distances (r_{Si}). The first four triangles and the rectangular shapes are explained below, while the remaining triangles are explained later on due to the fact that these triangles are created according to the position of the *i*th measuring point. From Figure 51, it can be seen that the rectangle and first triangle are the two shapes that create the frustum form. However, the proportion of the sides of the second triangle is the one that define the width of the rectangle and radius of the first triangle.

The rectangular shape, which creates a cylinder in 3-dimensions, has a height (h") created by the height of the frustum, and a width (r') that is created by the radius of the top circular plane. The first triangle is located at the left side of the rectangle and has parameters given by a height (v_1), which is the same as the height of the rectangle and the radius of the base (r") that is given by the difference between the base of the frustum (r) and the width of the rectangle (r'). In addition, the relation between the height and the radius of the base of this triangle is controlled by the angle (γ), which is obtained by applying the parameters and the trigonometric identity shown in Equations (6.74) to (6.76).

$$v_1 = h'' \tag{6.74}$$

$$r'' = r - r' \tag{6.75}$$

$$\tan\gamma = \frac{r''}{h''} \tag{6.76}$$

The second triangle is created by the conical cap that is removed from the cone in order to create the frustum form. The parameters of this triangle are the height (v_2), which is given by the distance from the center of the top circular plane to the imaginary vertex of the frustum, and the radius of the base (r'), which is given by the dimensions of the top plane. Similarly to the relation between the parameters of the first triangle, the angle (γ) controls the relation of the sides of the second triangle; as shown by the tangent and sine of the angle (γ) in Equation (6.77) to (6.79).

$$v_2 = h' \tag{6.77}$$

$$\tan\gamma = \frac{r'}{h'} \tag{6.78}$$

$$\sin\gamma = \frac{r'}{\sqrt{r'^2 + h'^2}} \tag{6.79}$$

The third and fourth triangles are identical in form and magnitude to the second and third triangles used in the analysis of the complete conical form. For the third triangle the corresponding Equations are (6.55) and (6.56), while for the fourth triangle the corresponding Equation is (6.57). The role that these two triangles play in this analysis is not as strong as the role that they played in the complete conical form. The three remaining cones are explained as a part of the example since those triangles are created according to the location of the *i*th measuring point. The point used in the following example is located in the same position as the location of the point used above.

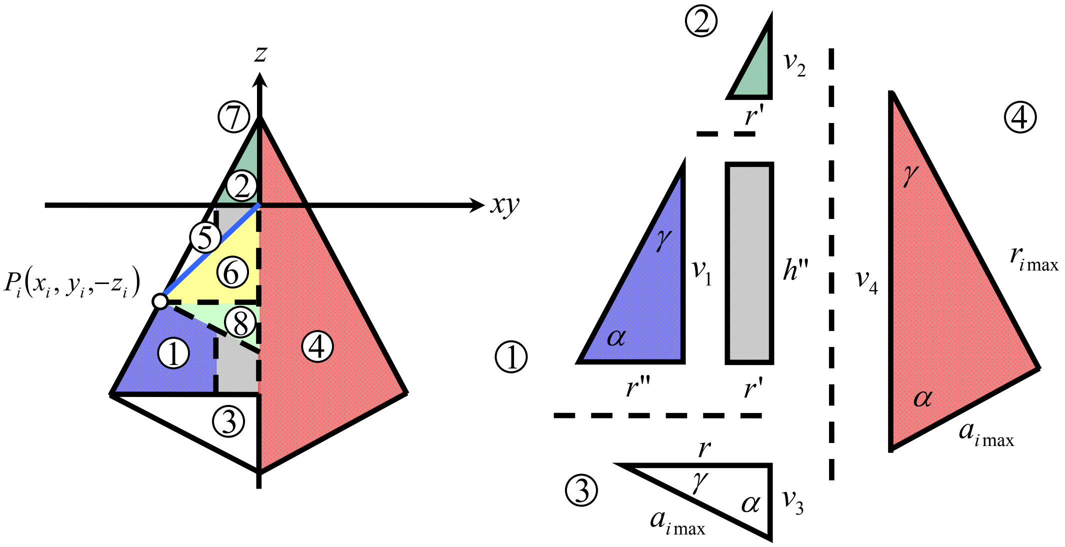


Figure 51. Right triangles used to analyze the conical frustum form.

The point (P_i) is used to calculate the measured radial distance, which is calculated by using the Euclidian distance from the origin of the frustum to the location of the point. However, it can be seen from Figure 51 that the radial distance is not placed over the surface of the frustum form, due to the fact that the origin is not located on the vertex. For this reason, the formulas developed for the complete conical form need to be modified. Moreover, the formulas do not contain all parameters of the frustum form, making the previous methodology not very attractive for the frustum form. Hence, the measured radial distance (r_i) is calculated by modifying the *z* component. This modification involves the addition of an extra term that is given by Equation (6.77), which is the height of the second triangle (v_2). The numerical value of the height of the second triangle is substituted in Equation (6.80) with the purpose of forcing the fitting algorithm to extract the height between the origin from the center of the radial plate to the imaginary vertex of the frustum. The reason that the height (*h*') is negative is because the offset will move the origin up making all z_i's be further down than where they were before the offset.

$$r_i = \sqrt{{x_i}^2 + {y_i}^2 + (z_i - h')^2} \tag{6.80}$$

The ideal radial distance (r_{Si}) is obtained by transforming the Euclidian distance that goes from the actual origin of the frustum to the location of the point, into a Euclidian distance that goes from the new origin located at the imaginary vertex to the location of the point. The transformation is explained using the following steps. The first step is to calculate the base of the fifth triangle, shown in Figure 52. The triangle is similar in form to the first triangle, where the proportion of the sides is given by the location of the *i*th point. The triangle has a height (v_5) given by the absolute value of the third component of the Cartesian coordinate (z_i), where the (z_0) component is added to the (z_i) value in order to counteract the effect of the origin offsets, as seen in Equation (6.81). Furthermore, since the first and fifth triangles are similar in proportions, the base of the triangle (r_{Pi}) is given by multiplying the tangent identity obtained for the first triangle in Equation (6.76) by the height of the triangle (v_5), as seen in Equation (6.82).

$$v_5 = |z_i - z_0| \tag{6.81}$$

$$r_{Pi} = v_5 \tan\gamma$$

$$r_{Pi} = (|z_i - z_0|)\left(\frac{r''}{h''}\right) \tag{6.82}$$

The second step is to calculate the base of the sixth triangle. The sixth triangle is used to describe the location of the *i*th point with respect to the actual origin of the frustum. This triangle has a height (v_6) that is the same as the height of the fifth triangle (v_5) and the base of this triangle (r_{Ti}) that is given by the addition of the width of the rectangle (r') plus the base of the fifth triangle (r_{Pi}) obtained in Equation (6.82), as seen in Equations (6.83) and (6.84).

$$v_6 = v_5 = \left| z_i - z_0 \right| \tag{6.83}$$

$$r_{Ti} = r_{Pi} + r'$$

$$r_{Ti} = \left(\left| z_i - z_0 \right| \right) \left(\frac{r''}{h''} \right) + r' \tag{6.84}$$

The third step is to convert the horizontal distance (r_{Ti}) obtained in Equation (6.84) into an ideal radial distance (r_{Si}) by applying the relations of the seventh triangle. The seventh triangle is similar in proportions to the first, second, and fifth triangles and shares the magnitude (r_{Ti}) with the sixth triangle. The triangle has a height (v_7) given by adding the height of the second triangle (v_2) and the height of the sixth triangle (v_6). It can be seen from Figure 52 that the hypotenuse of this triangle is given by the ideal radial distance (r_{Si}). Thus, the ideal radial distance (r_{Si}) is given by applying the sine of the angle (γ) to this triangle, and substituting Equation (6.84) in Equation (6.85), as seen in Equation (6.86).

$$r_{Si} = \frac{r_{Ti}}{\sin \gamma} \tag{6.85}$$

$$r_{Si} = \left[\left(\left|z_i - z_0\right|\right)\left(\frac{r''}{h''}\right) + r' \right]\left(\frac{1}{\sin\gamma}\right) \tag{6.86}$$

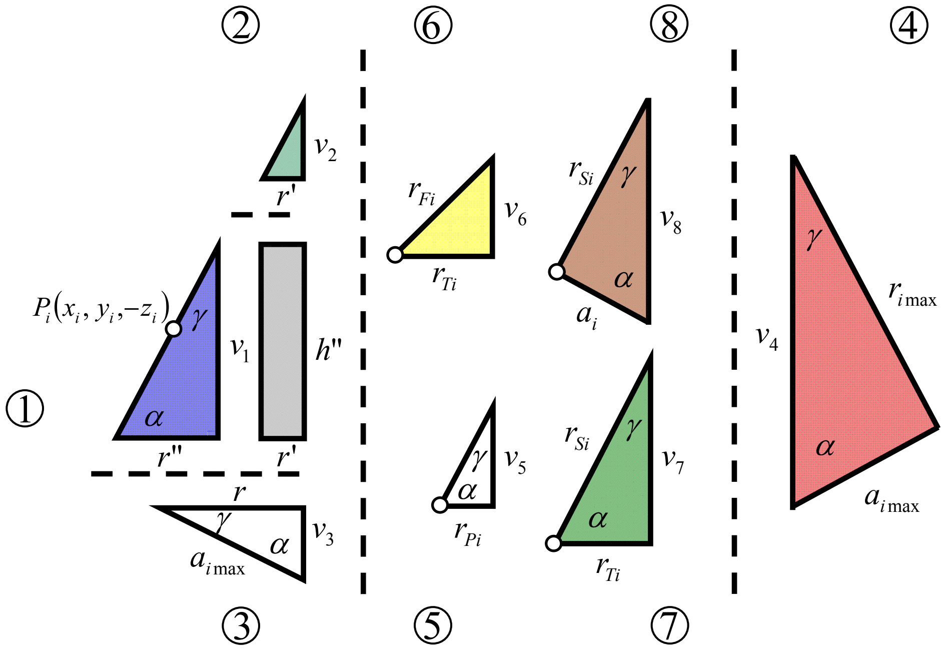


Figure 52. Ideal triangle used to calculate the frustum form deviations.

Finally, the symbolic equation to calculate the ideal radial distances (r_{Si}) that involve all parameters of the frustum is found by substituting Equation (6.79) in Equation (6.86).

$$r_{Si} = \left[\left(\left|z_i - z_0\right|\right)\left(\frac{r''}{h''}\right) + r' \right]\left(\frac{\sqrt{r'^2 + h'^2}}{r'}\right) \tag{6.87}$$

Equation (6.88) shows the calculation of the residuals of the frustum that are obtained by substituting Equations (6.80) and (6.87) in Equation (6.49).

$$d_i = \sqrt{x_i^{\,2} + y_i^{\,2} + (z_i - h')^2} - \left[\left[\left(\left|z_i - z_0\right|\right)\left(\frac{r''}{h''}\right) + r' \right]\left(\frac{\sqrt{r'^2 + h'^2}}{r'}\right) \right] \tag{6.88}$$

The residuals obtained by Equation (6.88) are not the true normal deviations of the frustum surface, since they come from the radial distances of the sphere. In order to generate the normal deviations, the Equation (6.88) is multiplied by the sine relation obtained from the eighth triangle. However, this relation is equivalent to the relation shown in Equation (6.76), which comes from the first triangle and is based on the ideal parameters of the frustum. Therefore, Equation (6.88) is multiplied by the ratio between the parameters of the first triangle, as seen in Equation (6.89).

$$d_i = \frac{r''}{h''}\left[\sqrt{x_i^{\,2} + y_i^{\,2} + (z_i - h')^2} - \left[\left[\left(|z_i - z_0|\right)\left(\frac{r''}{h''}\right) + r'\right]\left(\frac{\sqrt{r'^2 + h'^2}}{r'}\right)\right]\right] \tag{6.89}$$

The Equation (6.89) can be used to inspect frustum forms that are aligned as close as possible with the *z*-axis of the inspection instrument. Furthermore, the frustum is analyzed using a finite number of spheres that follows the same geometry as the cone. For this reason, it is necessary to add the last terms employed in the conicity model to counteract the origin offset, as seen in Equation (6.71).

$$\begin{aligned} d_i = \frac{r''}{h''}\Bigg[& \sqrt{x_i^{\,2} + y_i^{\,2} + (z_i - h')^2} - \Bigg[\left[\left[\left(|z_i - z_0|\right)\left(\frac{r''}{h''}\right) + r'\right]\left(\frac{\sqrt{r'^2 + h'^2}}{r'}\right)\right] \\ & + x_0 \operatorname{sgn}(x_i)\left|\cos\left[\tan^{-1}\left(\frac{y_i - y_0}{x_i - x_0}\right)\right]\sin\left[\pi - \tan^{-1}\left(\frac{r}{h}\right)\right]\right| \\ & + y_0 \operatorname{sgn}(y_i)\left|\cos\left[\tan^{-1}\left(\frac{y_i - y_0}{x_i - x_0}\right)\right]\sin\left[\pi - \tan^{-1}\left(\frac{r}{h}\right)\right]\right| \\ & + z_0 \operatorname{sgn}(z_i)\left|\cos\left[\pi - \tan^{-1}\left(\frac{r}{h}\right)\right]\right|\Bigg]\Bigg] \end{aligned} \tag{6.90}$$

6.6.2 Model of the Conical Frustum Form with Free Form Orientation

The model presented in Equation (6.89) assumes that the frustum form is perfectly oriented with the *z*-axis of the inspection instrument. However, this is not always the case because sometimes the axis of both the manufactured part and the inspection instrument are not always aligned. Hence, the (z_i) component of Equation (6.89) is substituted by the flatness model shown in Equation (6.45). Therefore, the final model to inspect frustum forms is shown in Equation (6.91) and in Figure 53.

$$d_i = \frac{r''}{h''}\left[\sqrt{x_i^2+y_i^2+(z_i-h')^2} - \left[\left[\left(\left|\frac{z_i-(mx_i+ny_i+z_0)}{\sqrt{m^2+n^2+1}}\right|\right)\left(\frac{r''}{h''}\right)+r'\right]\left(\frac{\sqrt{r'^2+h'^2}}{r'}\right)\right]\right] \tag{6.91}$$

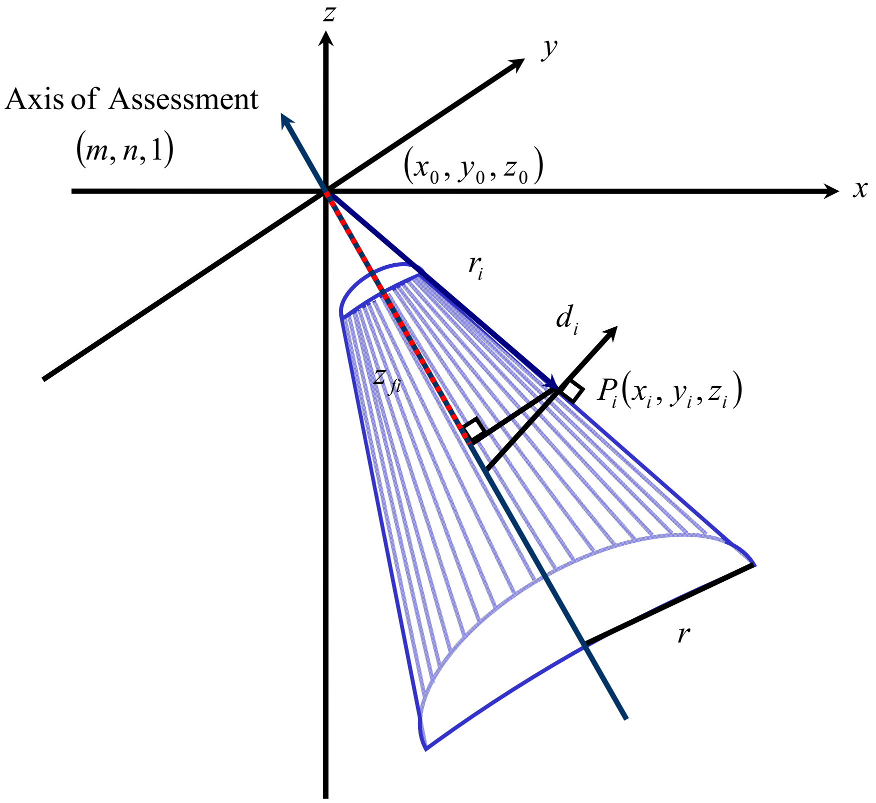


Figure 53. Free Form Orientation approach applied to the study of conical frustum.

6.6.3 Form Verification of Conical Frustum Form (Conicity)

The form verification of conical forms is also known as conicity, which is studied in the present research using two mathematical models shown above. The residuals provided by the three models are normal to the surface, reason why there is no need to modify them. The additional constraints needed to solve the conicity models are defined as follows:

$$
\begin{aligned}
&x_0 = x_0' - x_0'' \\
&y_0 = y_0' - y_0'' \\
&z_0 = z_0' - z_0'' \\
&m = m' - m'' \\
&n = n' - n'' \\
&R, h'', r'', h', r', x_0', x_0'', y_0', y_0'', z_0', z_0'', m', m'', n', n'' \geq 0
\end{aligned}
\tag{6.92}
$$

6.7 Torus Form

The torus form has received very little attention in the literature. Aguirre Cruz (2003), Aguirre Cruz and Raman (2005), and Ahn et al. (2004) were the only three researches that have studied the tolerance verification of torus form. The form error of the torus is analyzed using three different models. The first model is based on the approach developed by Aguirre Cruz (2003) and Aguirre Cruz and Raman (2005). The second model is the model proposed in this research, which focuses on measuring the torus form using an origin as a reference instead of using the axis of assessment, similar to the approach used for the complete cone and frustum forms. Finally, the third model inspects the torus deviations using the free form orientation approach.

6.7.1 Linear Model of Torus Form

Aguirre Cruz (2003) and Aguirre Cruz and Raman (2005) developed a linear model to inspect torus forms, which was based on radial distances that were defined from the axis of assessment to the location of the *i*th point, as shown in Equation (6.93). The experimental procedure developed in these researches confirms the accuracy of the model to represent the form deviations. However, the model applies the constant angle approach. Hence, the present research compares the performance of this model against those new models that have the dynamic angle approach.

$$d_i = \sqrt{x_i^2 + y_i^2} - \left[c \pm a\cos(\varphi_i) + (mz_i + x_0)\cos(\theta_i) + (nz_i + y_0)\sin(\theta_i)\right] \qquad (6.93)$$

6.7.2 Model of the Torus Form using the Spherical and the Dynamic Angle Approaches

Although the torus shape is completely different from conical forms, the form verification model is similar, in which the torus is evaluated using radial distances that provide the normal deviations of the inspected shape. The main idea is to assess the torus form as a finite combination of virtual spheres. A similar idea is found in Aguirre Cruz (2003), in which he proposed to transform the torus form into a sphere form. He found that the torus equation is a particular case of the sphere because if the radius *c* of the torus equation is set to zero then the remaining equation is identical to the equation of the sphere. Therefore, the present study adapts that idea, and instead of transforming the torus shape into a sphere, it uses several spheres to inspect the torus form, as is explained in the following lines.

The first step is to calculate the measured radial distance (r_i) and the symbolic radial distance (r_{is}) that goes from the origin of the inspection system (O_s) to the location of the *i*th point $P_i(x_i, y_i, z_i)$, as seen in Equations (6.94) and (6.95).

$$r_i = \sqrt{x_i^{\,2} + y_i^{\,2} + z_i^{\,2}} \tag{6.94}$$

$$r_{is} = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2} \tag{6.95}$$

The next step is to calculate all the three angles created by the *i*th radial distance: the angle (θ_i) that describes the location of the *i*th point on the *x* and *y* plane; the angle (φ_i) that provides information about the location of the *i*th point with respect to horizontal distance and the elevation of the point inside the torus tube; and last but not least, the angle (α_i) that describes the location of the *i*th point by using the relation between height of the point and the horizontal distance that goes from the origin of the torus to the location of the point.

$$\theta_i = \tan^{-1}\left(\frac{y_i - y_0}{x_i - x_0}\right) \tag{6.96}$$

$$\varphi_i = \tan^{-1}\left(\frac{z_i - z_0}{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} - c}\right) \tag{6.97}$$

$$\alpha_i = \tan^{-1}\left(\frac{z_i - z_0}{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}}\right) \tag{6.98}$$

Once all angles are calculated, it is necessary to find a mathematical relationship to represent the radial distances (r_i) and (r_{is}) in terms of the parameters of the torus. The purpose of this step is to create a relationship that can be used to extract the parameters of the torus from the measured radial distances. As mentioned above, Aguirre Cruz (2003) and Aguirre Cruz and Raman (2005) developed a model to assess torus form using horizontal distances that changed in magnitude according to the location of the point on the torus surface. The horizontal distances were calculated by adding together the radius of the torus (c) plus/minus the product between the torus tube radius (a) and the cosine of the angle (φ_i), shown in Equation (6.99). The addition or subtraction of these two values depends on the location of the measured point, for example if the horizontal location of the point is greater than the value of the torus radius (c) then it is necessary to add both values, but if the horizontal location is less than the value of the torus radius then it is necessary to subtract them. Figure 54 provides a description of the mentioned procedure.

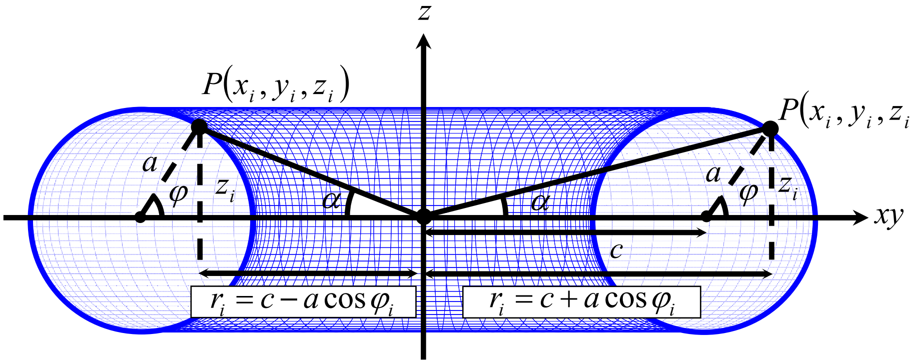


Figure 54. Parameters used in the analysis of torus forms.

$$\left(c \pm a\cos\left(\left|\tan^{-1}\left(\frac{z_i - z_0}{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} - c}\right)\right|\right)\right) \tag{6.99}$$

The previous equation provides the value of the horizontal distance; however it is necessary to convert this horizontal distance into a radial distance. Therefore, if Equation (6.99) is divided by the cosine of the angle (α_i), it will provide an ideal radial distance based on the one that was calculated using the measured data, as can be seen in Equation (6.100)

$$\frac{\left(c \pm a\cos\left(\left|\tan^{-1}\left(\frac{z_i - z_0}{\sqrt{\left(x_i - x_0\right)^2 + \left(y_i - y_0\right)^2} - c}\right)\right|\right)\right)}{\cos\left(\left|\tan^{-1}\left(\frac{z_i - z_0}{\sqrt{\left(x_i - x_0\right)^2 + \left(y_i - y_0\right)^2}}\right)\right|\right)} \qquad (6.100)$$

The torus zone of error can be found by substituting Equation (6.100) in the Equations (6.29) and (6.30) on the same place as the radius of the sphere. Thus, the model to verify torus forms is given by Equations (6.101 and 6.102)

$$\begin{aligned} d_i = \sqrt{x_i^2 + y_i^2 + z_i^2} - \Bigg[& \frac{\left(c \pm a\cos\left(\left|\tan^{-1}\left(\frac{z_i - z_0}{\sqrt{\left(x_i - x_0\right)^2 + \left(y_i - y_0\right)^2} - c}\right)\right|\right)\right)}{\cos\left(\left|\tan^{-1}\left(\frac{z_i - z_0}{\sqrt{\left(x_i - x_0\right)^2 + \left(y_i - y_0\right)^2}}\right)\right|\right)} \\ & + x_0 \operatorname{sgn}\left(x_i\right)\left|\cos\left[\tan^{-1}\left(\frac{y_i - y_0}{x_i - x_0}\right)\right]\cos\left[\tan^{-1}\left(\frac{z_i - z_0}{\sqrt{\left(x_i - x_0\right)^2 + \left(y_i - y_0\right)^2}}\right)\right]\right| \\ & + y_0 \operatorname{sgn}\left(y_i\right)\left|\sin\left[\tan^{-1}\left(\frac{y_i - y_0}{x_i - x_0}\right)\right]\cos\left[\tan^{-1}\left(\frac{z_i - z_0}{\sqrt{\left(x_i - x_0\right)^2 + \left(y_i - y_0\right)^2}}\right)\right]\right| \\ & + z_0 \operatorname{sgn}\left(z_i\right)\left|\sin\left[\tan^{-1}\left(\frac{z_i - z_0}{\sqrt{\left(x_i - x_0\right)^2 + \left(y_i - y_0\right)^2}}\right)\right]\right|\Bigg] \qquad (6.101) \end{aligned}$$

$$
\begin{aligned}
d_i = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2} \\
- \left[\frac{\left(c \pm a \cos\left(\left| \tan^{-1}\left(\dfrac{z_i - z_0}{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 - c}} \right) \right| \right) \right)}{\cos\left(\left| \tan^{-1}\left(\dfrac{z_i - z_0}{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}} \right) \right| \right)} \right. \\
+ x_0 \operatorname{sgn}(x_i) \left| \cos\left[\tan^{-1}\left(\frac{y_i - y_0}{x_i - x_0} \right) \right] \cos\left[\tan^{-1}\left(\frac{z_i - z_0}{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}} \right) \right] \right| \\
+ y_0 \operatorname{sgn}(y_i) \left| \sin\left[\tan^{-1}\left(\frac{y_i - y_0}{x_i - x_0} \right) \right] \cos\left[\tan^{-1}\left(\frac{z_i - z_0}{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}} \right) \right] \right| \\
\left. + z_0 \operatorname{sgn}(z_i) \left| \sin\left[\tan^{-1}\left(\frac{z_i - z_0}{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}} \right) \right] \right| \right] \qquad (6.102)
\end{aligned}
$$

Finally, the normalized residuals (d_{ni}) are found by multiplying Equation (6.101) and (6.102) by the cosine of the difference between the angle (φ_i) and the angle (α_i).

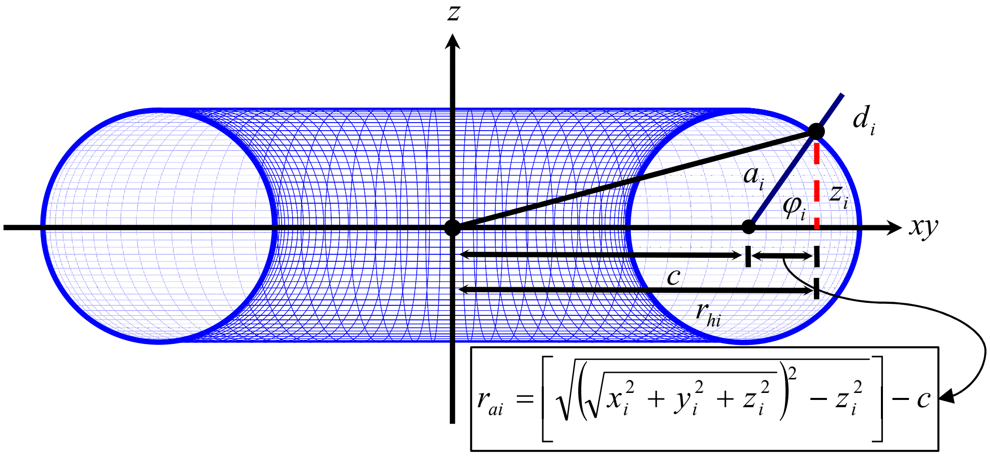
$$
d_{ni} = d_i \cos(\varphi_i - \alpha_i) \qquad (6.103)
$$

6.7.3 Model of the Torus Form with Free Form Orientation

The free form orientation model for torus forms is developed in a different way than the models for cone, cylinder, and frustum forms. The objective of the model is to generate the normalized residuals by comparing the measured value of the radius of the torus tube (a_i) against the ideal value of the radius of the torus tube (a). Thus, the measured radial distance (r_i) must be converted into the measured value of the radius of the torus tube (a_i). The transformation is done by following the next steps: First, the Pythagorean Theorem is used to convert the measured radial distance (r_i) into a horizontal distance (r_{hi}), as seen below.

$$r_{hi} = \sqrt{\left(\sqrt{x_i^2 + y_i^2 + z_i^2}\right)^2 - z_i^2} \tag{6.104}$$

The (z_i) component of the *i*th point which describes the height of the triangle seen in Figure 55 needs to be modified according to the principles of the free form orientation. This means that the (z_i) needs to be adapted for any axis misalignments. Thus, the flatness model from Equation (6.45) is applied to this value. On the other hand, the (z_i) value inside the square root of the measured radial distance (r_i) is not affected by this procedure because it is used as a coordinate of a 3-dimensional space and not on its own.



z
d_i
a_i
φ_i
z_i
xy
c
r_{hi}
$r_{ai} = \left[\sqrt{\left(\sqrt{x_i^2 + y_i^2 + z_i^2}\right)^2 - z_i^2}\,\right] - c$

Figure 55. Nonlinear free form orientation of torus form.

$$r_{hi} = \sqrt{\left(\sqrt{x_i^2 + y_i^2 + z_i^2}\right)^2 - \left[\frac{z_i - \left(mx_i + ny_i + z_0\right)}{\sqrt{m^2 + n^2 + 1}}\right]^2} \tag{6.105}$$

Second, the radius of the torus (c) is then subtracted from the horizontal radial distance (r_{hi}), which generates the horizontal distance (r_{ai}) inside the torus tube.

$$r_{ai} = \sqrt{\left(\sqrt{x_i^2 + y_i^2 + z_i^2}\right)^2 - \left[\frac{z_i - \left(mx_i + ny_i + z_0\right)}{\sqrt{m^2 + n^2 + 1}}\right]^2} - c \tag{6.106}$$

Third, the Pythagorean Theorem is applied between Equation (6.106) and the (z_i) component in order to obtain the measured value of the radius of the torus tube (a_i). Finally, the ideal radius of the torus tube (a) is subtracted from the measured value (a_i) (Equation 6.106) providing the nonlinear torus model with free form orientation capabilities, as seen in Equation (6.107) and in Figure 56.

$$d_i = \left[\left[\sqrt{\left(\sqrt{\left(x_i - x_0\right)^2 + \left(y_i - y_0\right)^2 + \left(z_i - z_0\right)^2}\right)^2 - \left[\frac{z_i - \left(mx_i + ny_i + z_0\right)}{\sqrt{m^2 + n^2 + 1}}\right]^2} - c\right]^2 + \left[\frac{z_i - \left(mx_i + ny_i + z_0\right)}{\sqrt{m^2 + n^2 + 1}}\right]^2\right]^{1/2} - a \tag{6.107}$$

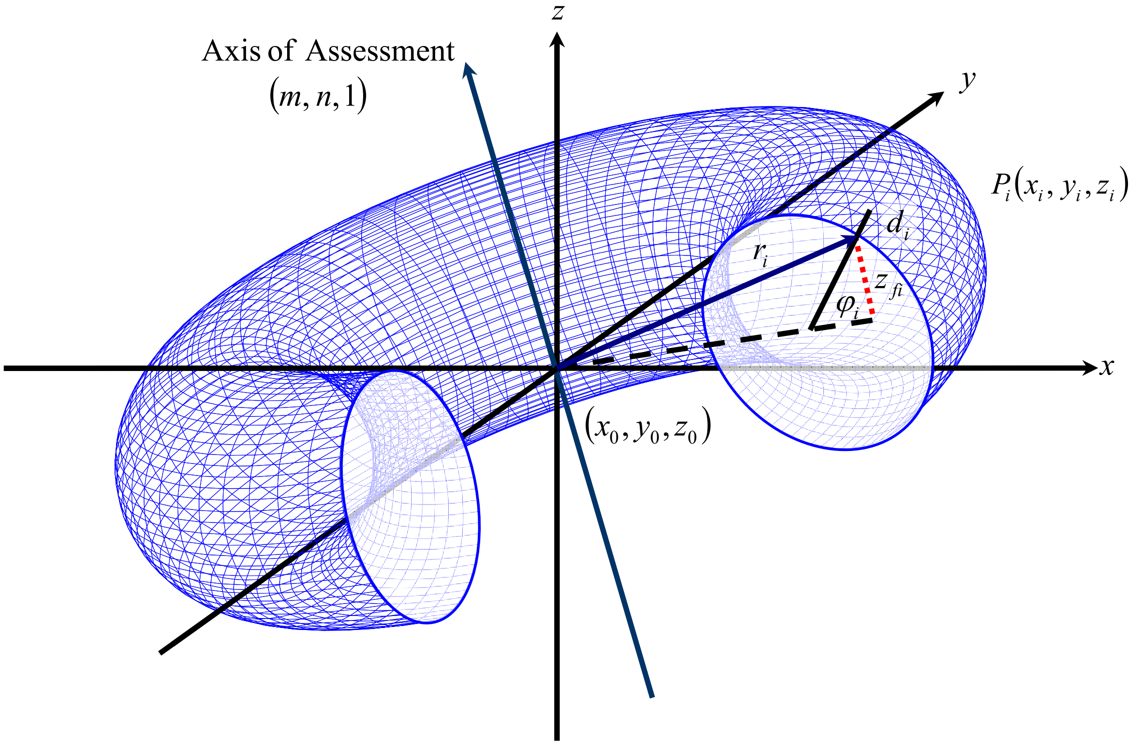


Figure 56. Free Form Orientation applied to the study of torisity.

6.7.4 Form Verification of Torus Forms (Torisity)

The form verification of torus forms is also known as torisity, which is studied in the present research using three mathematical models shown above. The residuals provided by the first two models (Equations 6.101 and 6.102) are not normal. For this reason, the Equation (6.103) must be added to each model in the form of a dynamic angle approach in order to solve the minimum zone method using the normal residuals. On the other hand, the third model provides residuals that are normal to the surface, which is why there is no need to modify them. The additional constraints needed to solve the cylindricity models are defined as follows:

$$\begin{aligned}
&x_0 = x_0^{'} - x_0^{''} \\
&y_0 = y_0^{'} - y_0^{''} \\
&z_0 = z_0^{'} - z_0^{''} \\
&m = m^{'} - m^{''} \\
&n = n^{'} - n^{''} \\
&c, a, x_0^{'}, x_0^{''}, y_0^{'}, y_0^{''}, z_0^{'}, z_0^{''}, m^{'}, m^{''}, n^{'}, n^{''} \geq 0
\end{aligned} \tag{6.108}$$

6.8 Multiple Feature Form Evaluation

The evaluation of multiple form features is performed by solving simultaneously the mathematical models that are contained in the part. Hence, no special mathematical models are developed for the multiple feature part. The multiple feature analysis presented in the chapter 8 should be viewed as a guideline for developing future research in the field and is by no mean intended to serve as a comprehensive analysis of feature interactions, orientation tolerance verification, and /or part-based tolerance analysis. It is interesting, however to note that adjacent feature can be analyzed jointly to identify axis errors, and fixturing defects.

CHAPTER 7

EXPERIMENTAL METHODOLOGY

7 Summary

The proposed form verification models were evaluated by using a series of experiments in order to determine their performance and accuracy. The experiments were divided according to the form that was inspected, due to the fact that the presented research studied many types of form features. Each of the form features studied here are independent in itself and hence the type of feature form was not considered as an independent variable in the experiment.

7.1 Experimental Tools and Apparatus

The inspection equipment used to collect the data from the surface of the form features was a Brown & Sharpe MicroVal PFx^{TM} 454 coordinate measuring machine (CMM) with a touch trigger probe head. The movements of the CMM were controlled using the numerical control (NC) mode that was programmed using the CMM software. The machine had a measurement repeatability of 0.00015 inches and a linear displacement accuracy of 0.0002 inches for each of the axes. The measuring table of the CMM had threaded holes that allowed the use of clamps to hold the workpieces in place and to avoid any movements of the parts during the inspection. The elements used to hold the workpiece were composed of four metric studs, four flange nuts, four screws, and four serrated end clamps.

The machine recorded the data points from the part surface using a Renishaw (PH9/PH10) manual touch trigger probe head that was fixed at the bottom part of the *z*-axis and had the capability of holding M2 and M3 styluses. The two styluses used in the following experiments were: a Renishaw M2 ruby tipped stylus (A-5003-0577) with an overall length of 0.787 inches and a ball radius of 0.0276 inches, and a Renishaw M3 ruby tipped stylus (A-5003-0052) with an overall length of 1.2 inches and a ball radius of 0.0787 inches.

A personal computer with Windows^{TM} 98 as an operating system was used as the link between the experimenter and the coordinate measuring machine. The computer had an interface software called TUTOR for Windows^{TM} that helped the experimenter to: qualify and calibrate each of the styluses' tips, to translate the origin of the machine into the same position as the geometrical origin of the part, to run the subroutines to inspect the form features, and to execute the NC programs needed to collect the point coordinates.

7.2 Experimental Design

The response variable for these experiments was set as the width of the zone of error produced by the mathematical models. The width of the zone of error was affected by two independent variables: sample size and the type of mathematical model. The main reason that these two variables were chosen was because one of the goals of this study was to determine the accuracy and precision of the developed models in the presence of *n* number of data points.

The fitting algorithm used to analyze the data was not set as an independent variable in these experiments since only one algorithm was used, mentioned below as the minimax algorithm. Furthermore, the specimens used in these experiments were considered the blocking factor since they were not identically machined.

The levels of the two independent variables were set as:

- Sample size: 8, 32, 128, and 512 data points.
- Mathematical model: varies according to the feature being inspected. Refer to Tables 30 to 35.

A balanced factorial design model with fixed factors was used to analyze the variance of the width of the zone of error. The statistical model was defined by:

$$y_{ijk} = \mu + A_i + B_j + \beta_k + (AB)_{ij} + \varepsilon_{ijk} \tag{8.1}$$

where:

A_i is the effect of the ith sample size,

$i = 1, \ldots , 4$

B_j is the effect of the jth mathematical model,

$j\ 1, \ldots , b$

$(AB)_{ij}$ is the interaction between the sample size and the mathematical model,

β_k is the blocked effect of the kth specimen,

$k = 1, \ldots, 5$

ε_{ijk} is the random error

7.3 Experimental Procedure

The experiments were performed in the precision laboratory located in Carson Engineering Center room CEC33 where the temperature of the room was set at 65 degrees Fahrenheit. The aluminum parts and the measuring table of the CMM were cleaned using alcohol and a soft cloth in order to avoid any scratches to the surface of the parts. The type of features and the factor combinations of each feature were inspected in a random order, which was: the cylinder, sphere, cone, torus, multiple feature part, and the frustum.

Once the data points were collected, the data points were stored for the analysis in a Dell® Inspiron 8100 laptop computer with an Intel Pentium III mobile CPU. The text files with the data points were cleaned using a program written in Microsoft® Excel 2000 macros. The data points were applied to each of the mathematical models of the inspected feature using the worksheets of Microsoft® Excel 2000. Then, the parameters of the features were extracted from the data points using the function *fminimax* from Matlab® 6.5 where the only information provided to the function solver was the starting point. It is important to mention that no upper and lower boundaries were provided to the function in order to verify if the models were accurate enough to yield the correct parameters of the form feature. Finally, the parameters and residuals were exported and analyzed using the Microsoft® Excel worksheets.

7.3.1 Experimental Procedure for Spherical Feature

The sphericity error was evaluated by inspecting five hemispheres that were used in Balakrishna’s (2005) research. The spherical forms were machined out of an aluminum workpiece and were created at the top of two cylindrical features using a radius of 1.25 inches. The manufacturing process and the tolerances used to create the five parts were unknown. However, it was known that the five parts were not identically machined since some of them had an incomplete hemisphere form. This means that the top part of the hemisphere feature was accurate in form but the bottom part did not meet the *xy* plane at a radial value of 1.25 inches causing a bump in the interface between the hemisphere and the cylindrical part. Figure 57 shows a drawing of one of the specimens used to evaluate the spherical form.

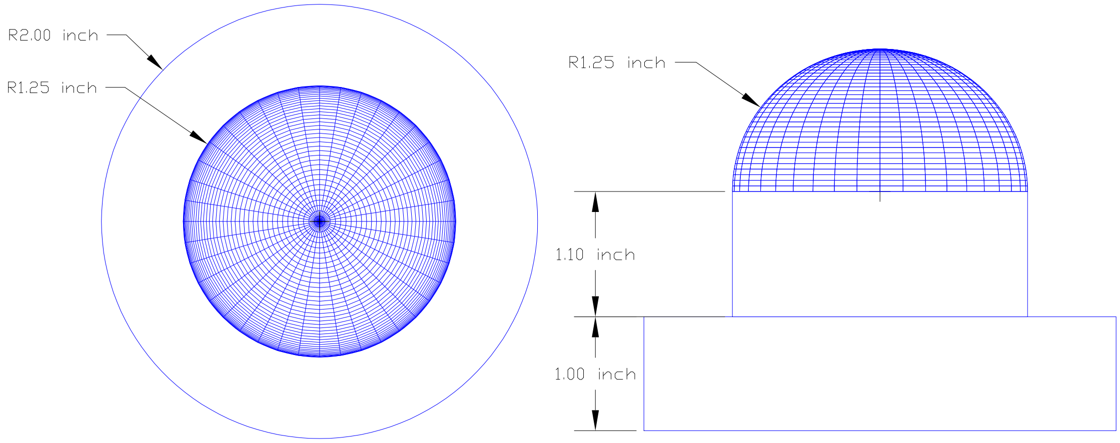


Figure 57. Top and front view of a hemisphere part used to verify the spherical form.

The spherical form was verified using two experimental procedures: the first experiment consisted of evaluating the performance of the two dynamic angle approach models (SQRT and NSQRT), the nonlinear model with no angles (NOANG), and Shunmugam's linear (SL) and nonlinear (SNL) spherical models, shown in Table 30. The models were evaluated by collecting information about the surface from the five hemispheres using four sample sizes of 8, 32, 128, and 512 data points generated using the Hammersley sampling strategy. The stylus tip used in this experiment was the M2 stylus which was used to translate the origin of the CMM into the geometrical origin of the sphere. The procedure used for the translation of the origin was the same procedure explained in Chapter 5, and it was used in order to make sure that both origins were located very close to each other. Finally, the data points were collected using the NC mode of the CMM that was programmed using the software of the CMM. This approach was used in order to collect the data points by applying an approaching vector that was normal to the surface of the hemisphere.

Table 30. Mathematical models used in the evaluation of spherical form.

Acronym	**Type of Model**	**Equation**
SL	Shunmugam's linear model	(6.18)
SNL	Shunmugam's nonlinear model	(6.19)
NSQRT	DAA with No square root	(6.29)
SQRT	DAA with Square root	(6.30)
NOANG	No angles	(6.37)

The second experiment consisted on evaluating the performance of the models mentioned above by setting the origin of the CMM in a different position than the geometrical origin of the sphere. The former was located approximately 0.25 inches above the geometrical origin of the part. The procedure used to locate the origin at that position was the same as the one used above. The only difference was that the *z* offset was changed by positioning the stylus 0.25 inches above the location of the origin and recording a point at that height using the "z-preset" function. The five workpieces were inspected using the M3 stylus tip and a sample size of 32 data points that followed the Hammersley distribution.

Finally, the twenty factor combinations and the five factor combinations of the first and second experiments respectively were randomly evaluated in order to avoid any systematic errors, where the randomization function from Microsoft® Excel 2000 was used to determine the random order of the experiment.

7.3.2 Experimental Procedure for Cylindrical Feature

The cylindricity error was evaluated by inspecting the cylindrical features that were contained in the parts used in Balakrishna's (2005) research. Although, the hemispheres were the principal features of these parts, the cylindrical features were machined accurately using a radius of 1.25 inches.

The cylindrical features were evaluated using the dynamic angle approach model (SQRT), Shunmugam's linear model (SL), and the nonlinear model with free form orientation capabilities (NL), as shown in Table 31. The cylindrical features were inspected using four sample sizes of 8, 32, 128, and 512 data points that followed the Hammersley distribution. The parts were not aligned perfectly with the coordinate system of the CMM. The misalignment was intentionally induced in order to determine if the models were accurate enough to capture the misalignments and to see if they could estimate the parameters and the zone of error in a precise way. The inclination of the part was induced by positioning the part above the first step of the serrated end clamp located at the right side of the part, which approximately rotated the axis of the cylinder to a new position [0.033, 0, 1], as seen in Figure 58. It is important to mention that not all the cylindrical feature was inspected due to the limitations of the CMM and in order to avoid any collision with the part. This was due to the fact that the parts were inspected by programming the NC mode of the CMM using the software. For this reason, a small region of the top part of the cylinder was inspected which had a height of 0.35 inches below the origin of the hemisphere. This means that the origin of the CMM was translated into the geometrical origin of the hemisphere where the 0.35 inches below the hemisphere form were inspected.

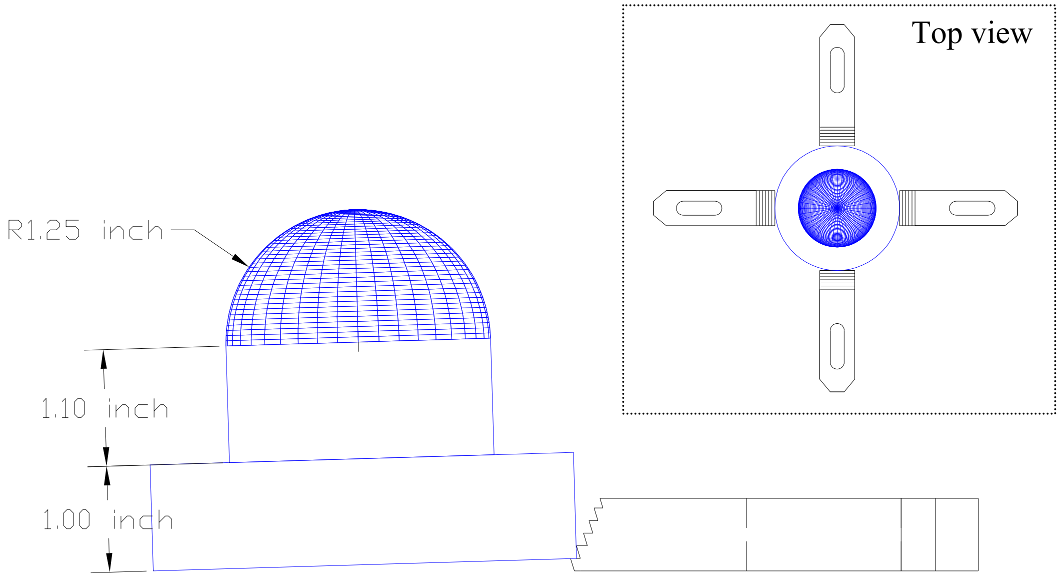


Figure 58. Inspection of cylindrical feature using an axis of revolution of [0.033, 0, 1]

Table 31. Mathematical models used in the evaluation of cylindrical form.

Acronym	**Type of Model**	**Equation**
SL	Shunmugam's linear model	(6.39)
SQRT	DAA with Square root	(6.42)
NL	FFO Nonlinear model	(6.47)

7.3.3 Experimental Procedure for Conical Feature

The conicity error was evaluated by inspecting five conical features that were used in Prakasvudhisarn's (2002) research. The conical forms were machined out of an aluminum workpiece using a radius of the base of the cone of 1.5 inches and a height of the cone of 2.5 inches. The conical parts had a small pin at the base of the cone that was used to fix the part over a small square base in order to avoid movements during the inspection process. Figure 59 shows the conical form and the square base used in this experiment.

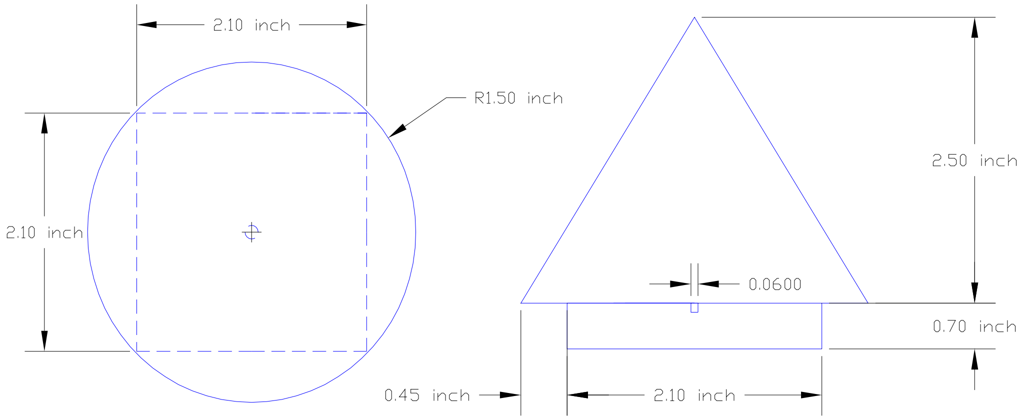


Figure 59. Workpiece of a conical form used to verify the conical form.

The conical parts were inspected using one experiment that consisted on evaluating the performance of the two dynamic angle approach models (SQRT and NSQRT) and the nonlinear model with free form orientation capabilities (NL), shown in Table 32. The parts were inspected using four sample sizes of 8, 32, 128 and 512 data points that followed the Hammersley distribution. The translation of the origin between the CMM and the conical part was performed using the procedures explained in Chapter 5. All the data points were recorded using the M2 stylus tip by programming the NC mode of the CMM using the software. In addition, the twenty factor combinations were randomly evaluated in order to avoid any systematic errors.

Table 32. Mathematical models used in the evaluation of conical form.

Acronym	**Type of Model**	**Equation**
NSQRT	DAA with No square root	(6.70)
SQRT	DAA with Square root	(6.71)
NL	FFO Nonlinear model	(6.72)

7.3.4 Experimental Procedure for Frustum Feature

The conical frustum error was evaluated by inspecting five frustum features that were used in Prakasvudhisarn's (2002) research. The frustum forms were machined out of an aluminum workpiece using a radius of the base of the cone of 2 inches and a frustum height of 3 inches. In a similar way as the conical parts, the frustums had a small pin at the base that was used to fix the part over a small square base in order to avoid movements during the inspection process. Figure 60 shows the frustum form and the square base used in this experiment.

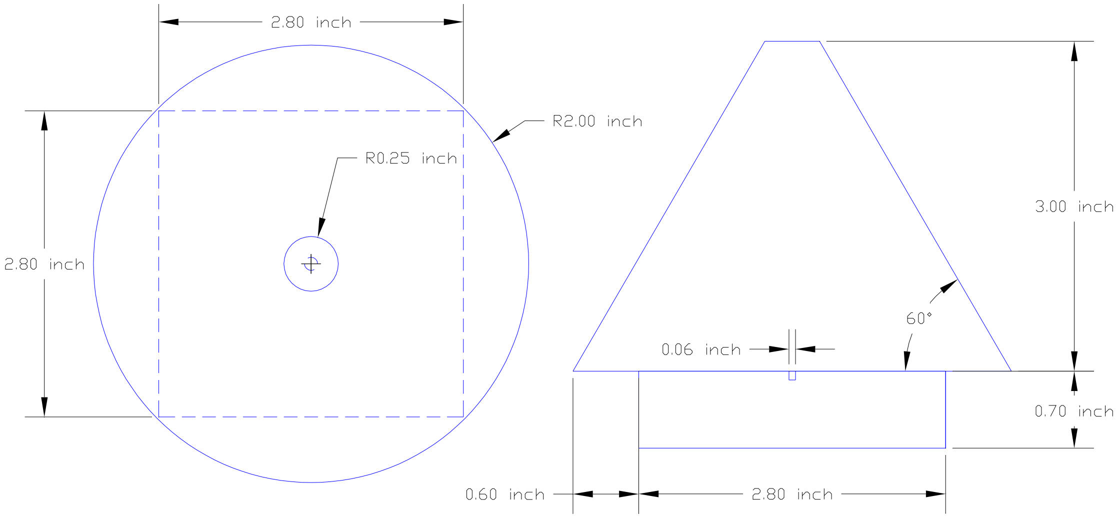


Figure 60. Top and front view of a frustum form used to verify the zone of error.

The frustum features were evaluated by comparing the performance of two mathematical models: one which used the dynamic angle approach (NSQRT) and the nonlinear model that uses the free form orientation capabilities (NL), shown in Table 33. The forms were inspected using four sample sizes of 8, 32, 128, and 512 data points that followed the Hammersley distribution. The data points were collected with a M2 stylus by programming the NC mode of the CMM using the software. In addition, all factor combinations were randomly evaluated in order to diminish the probability of capturing systematic errors.

Table 33. Mathematical models used in the evaluation of frustum form.

Acronym	Type of Model	Equation
NSQRT	DAA with No square root	(6.90)
NL	FFO Nonlinear model	(6.91)

7.3.5 Experimental Procedure for Torus Feature

The torisity error was evaluated by inspecting five torus features that were used in Aguirre Cruz's (2003) research. The torus forms were machined out of an aluminum workpiece with a radius of the torus of 1.25 inches, and a radius of the torus tube of 0.75 inches. The parts were not all machined in the same way, since they presented similar machining inaccuracies as the ones explained in the spherical parts. Figure 61 shows a drawing of the torus forms used in these experiments.

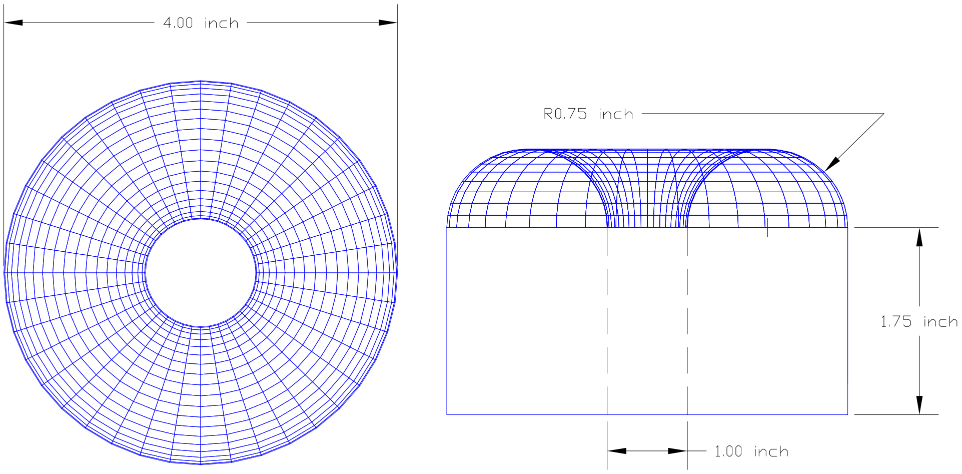


Figure 61. Workpiece of a torus form used in the verification of form error.

The torus forms were evaluated using two experiments: the first experiment consisted on inspecting the five torus parts using the M2 stylus and the NC mode of the CMM programmed using the software, where four sample sizes were used of 8, 32, 128, and 512 data points that followed the Hammersley distribution. These factors were used to compare the performance of the two dynamic angle approach models (SQRT and NSQRT), Aguirre Cruz's Thesis model (Thesis), and the nonlinear model with free form orientation capabilities (NL), as shown in Table 34.

The second experiment consisted on inspecting the five torus parts using the M3 stylus. The parts were not aligned properly with the coordinate system of the CMM, in order to determine whether the models could capture the inclination of the torus form. The inclination of the parts was done by setting the part above the first step of the serrated end clamp which provided an axis of assessment of [0.033, 0, 1]. The parts were inspected using one sample size of 32 data points that followed the Hammersley distribution, which were used to verify the performance of the four models shown in Table 34.

Table 34. Mathematical models used in the evaluation of torus form.

Acronym	Type of Model	Equation
Thesis	Aguirre Cruz's thesis model	(6.93)
NSQRT	DAA with No square root	(6.101)
SQRT	DAA with Square root	(6.102)
NL	FFO Nonlinear model	(6.107)

7.3.6 Experimental Procedure for Multiple Feature Parts

The multiple feature part was verified using the workpieces used in Balakrishna's (2005) research. The reason that this parts were chosen was because they had two of the most used features in the literature: a spherical feature and a cylindrical feature. The features were verified by solving the models of both features at the same time, where the mathematical model chosen for the spherical form was the dynamic angle approach model with square root (SQRT), while the mathematical model chosen for cylindrical form was the nonlinear model with free form orientation capabilities (NL), shown in Table 35. The purpose of this experiment was to determine if both features were created using the same manufacturing process. If both features were created using one machining sequence, then both features must share the same origin and the width of the zone of error of the multiple feature analysis should be equal to the maximum width of the zone of error of the individual feature analysis. However, if both features were machined using different machining sequences then the minimax algorithm should show a higher zone of error for the analysis of the multiple feature part than ones found for the individual feature analysis.

Table 35. Mathematical models used in the evaluation of multiple feature part.

Acronym	**Type of Model**	**Equation**
SQRT	DAA with Square root (Sphere)	6.30
NL	FFO Nonlinear model (Cylinder)	6.47

The five parts were inspected using a sample size of 64 data points that followed the Hammersley distribution. The sample size was divided in two parts, where the first 32 points were used to inspect the spherical feature, while the remaining points were used to inspect the cylindrical feature. The points were collected using the NC mode of the CMM that was programmed using the software. The M3 stylus tip was used to collect the data points since it was the longest tip available that was able to reach most of the cylindrical feature. Finally, the parts were randomly evaluated in order to avoid any systematic errors.

CHAPTER 8

RESULTS AND ANALYSES

8 Summary

The results presented in this chapter were arranged by features in the order of spherical, cylindrical, conical, frustum, torus, and multiple feature forms. The parameters and zone for error of each of these features were analyzed using the statistical analysis of variance to determine the precision of the mathematical models.

8.1 Results of Spherical Form

The spherical form was evaluated using two experiments. The first experiment focused on evaluating the performance of the mathematical models by inspecting five spherical parts that were perfectly aligned with the coordinate system and origin of the coordinate measuring machine. Each one of the five parts was inspected using four sample sizes of 8, 32, 128, and 512 data points for a factor combination of twenty. The results obtained from this experiment showed no difference between the performances of the models. The estimated parameters and zones of error were identical for all models. Figure 62 shows the plot of the zones of error of the spherical parts at all sample sizes. It can be seen that the width zone of error increases as the sample size increases proving that more information from the surface is being captured as the sample size increases. In addition, the plot contains the value called “Worst,” which is given by comparing the measured data points (x_i, y_i, z_i) against an ideal radius (r), assuming no origin offsets. Hence, the value “Worst” is obtained as follows:

$$\text{Worst} = \sqrt{x_i^2 + y_i^2 + z_i^2} - r \tag{8.1}$$

The main purpose of the comparison between the performance of the models and the value “Worst” was that if the width of the zone of error of the models had been found identical to the value “Worst” then it would imply that the origin of the CMM was theoretically positioned almost in the same spot as the geometrical origin of the part. If the models performed even worse than the value “Worst,” then it would imply that the models were not robust enough to be attracted to the optimal solution. Finally, if the width of the zone of error of the models was located well below the value “Worst,” then this would suggest that the origin of the CMM and the geometrical origin of the part were not very well aligned. Furthermore, it can be seen from Figure 63 that the estimated radii values of all the models were closer to the ideal value of the radius of the measured spheres, which was specified at 1.25 inches.

The analysis of variance (ANOVA) was not conducted in this experiment, due to the fact that all models generated the same width of the zone of error and the same radii values. Therefore, it was assumed that all models have the same performance.

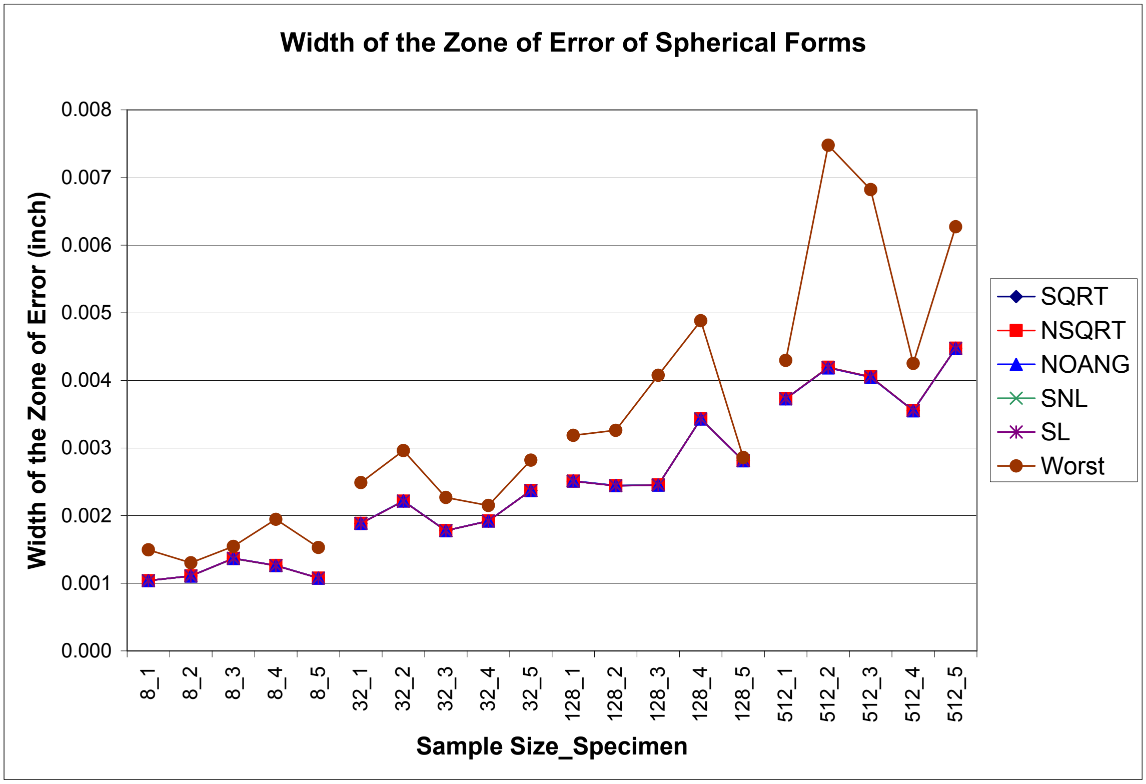


Figure 62. Zones of error of spherical form at different sample sizes.

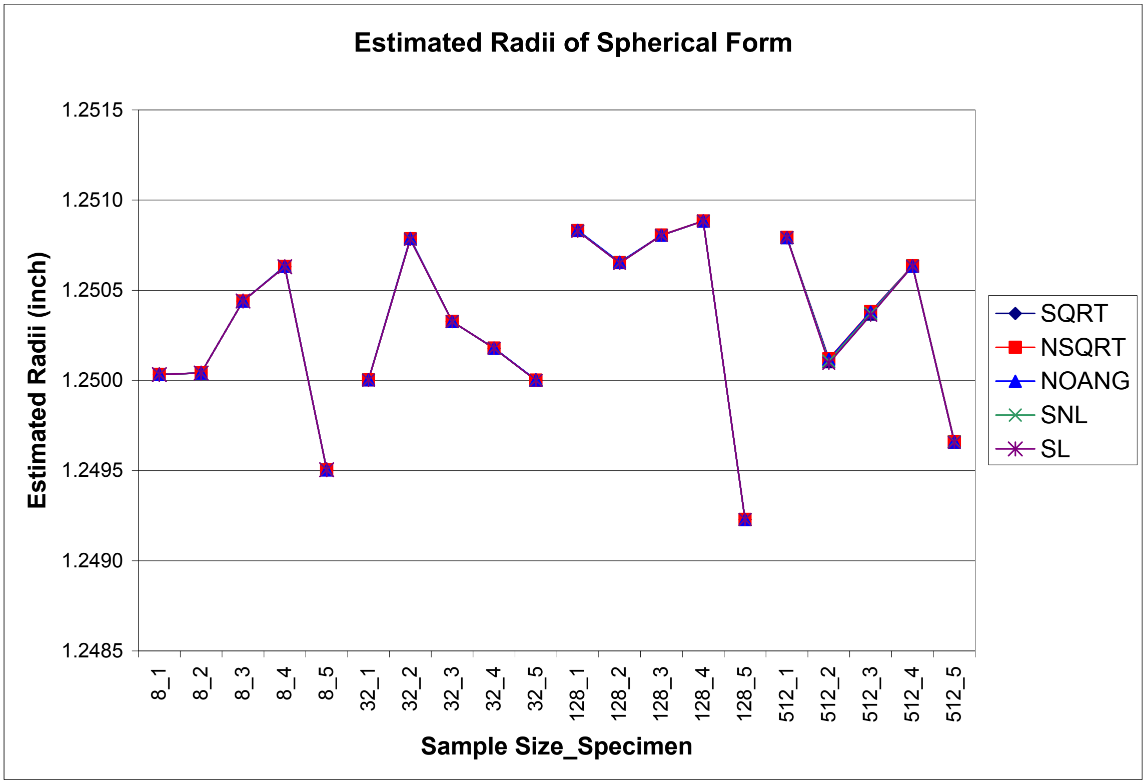


Figure 63. Estimated radii values of the five spherical models.

The second experiment consisted of placing the origin of the CMM at 0.25 inches above the geometrical origin of the part. The five parts used in this experiment were identical to the parts used above. The only difference was that the parts were inspected using only a sample size of 32 data points. Results from this experiment showed different performance between the models. For this reason the width of the zone of error of the five models was analyzed using the analysis of variance (ANOVA), where the width of the zone of error was calculated by subtracting the minimum deviation (d_i) from the maximum deviation (d_i). For this experiment, the type of mathematical model was the only independent variable, due to the fact that only one sample size was applied. Figure 64 shows the width of the zone of error generated by each mathematical model and by the coordinate measuring machine (CMM).

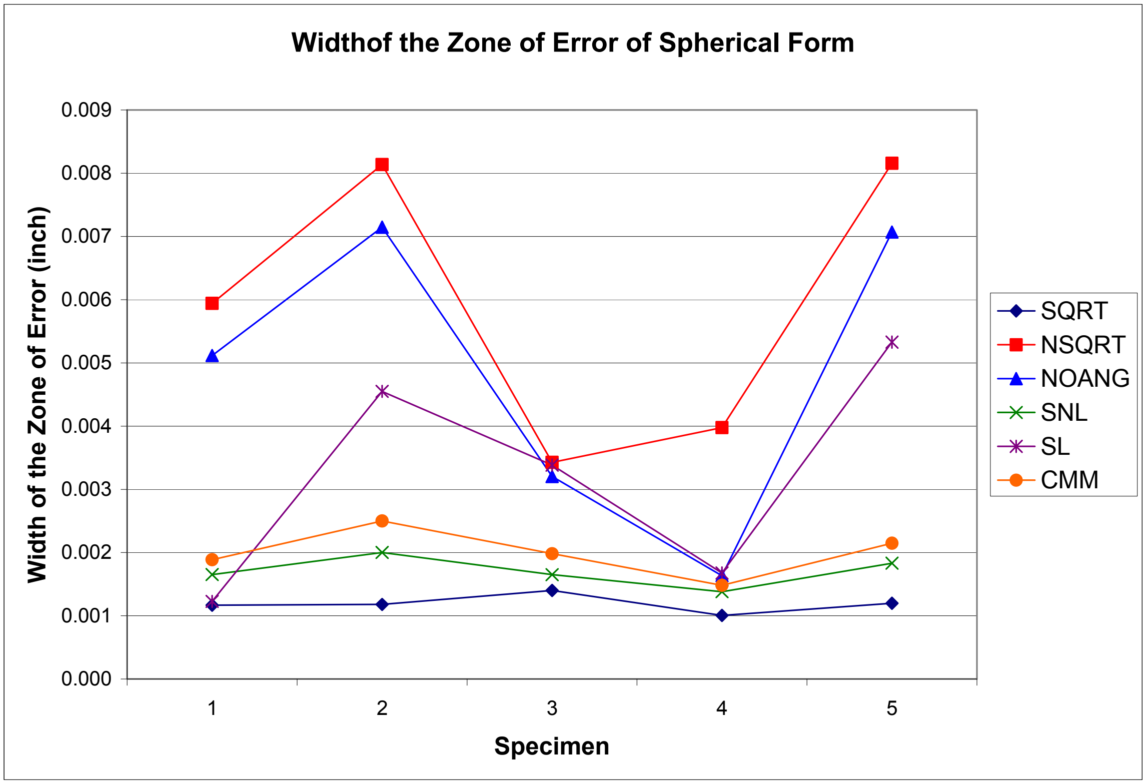


Figure 64. Width of the zone of error of spherical models.

It is important to mention that the magnitude of the zone of error of the CMM was included in the plots but was not included in the analysis of variance, due to the fact that the CMM is not a level of the factor "mathematical model" and that the type of model that the CMM used to calculate the parameters is unknown.

In order to detect any violations of the ANOVA assumptions, the data were analyzed for normality, equal variances, and independence, which can be seen in Appendix D. The analysis showed that the normality assumption was validated by the four tests and the normal probability plot generated using the *proc univariate* function. The plots for equal variances showed a pattern close to a funnel which implied that the models have different variances. However, ANOVA is robust to unequal variances when using a balance fixed effects model (Montgomery 2001). The plot of residuals versus sequence was not generated due to the fact that the same data for each factor combination was applied to each of the mathematical models. Thus no specific order was used to apply the data to a particular model.

Results from the analysis of variance showed significant differences between the performances of the five mathematical models. According to the results showed by the Tukey test using an alpha value of 0.01, the means of all models were found to be divided in three groups, shown in Figure 65 below. The first group was composed by the dynamic angle approach model with no square root (NSQRT) with a mean of 0.005929, the model with no angles (NOANG) with a mean of 0.004834, and Shunmugam's linear model (SL) with a mean of 0.003233. The second group was composed of the model with no angles (NOANG) with a mean of 0.004834, Shunmugam's linear model (SL) with a mean of 0.003233, and Shunmugam's nonlinear model (SNL) with a mean of 0.001704, while the third group was composed of Shunmugam’s linear model (SL) with a mean of 0.003233, Shunmugam’s nonlinear model (SNL) with a mean of 0.001704, and the dynamic angle approach model with square root (SQRT) with a mean of 0.001191.

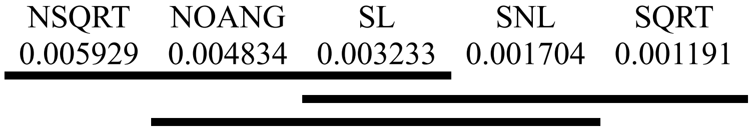


Figure 65. Means comparison of the factor Mathematical Model of the spherical form using the Tukey test.

The results obtained from the Tukey test implied that the mean value of the width of the zone of error of the models was statistically significant. The first group had the highest magnitude of the means, while the third group had the smallest magnitude of the means. In theory, the smaller the width of the zone of error the better, since it means that the mathematical models were able to locate an optimal solution that represents the form deviations. However, this does not mean that the models were accurate regarding the estimated radius of the sphere because they could have generated a small zone of error but at a radius and origin offsets that do not represent the dimensions of the real part. For this reason, further analysis was done in order to determine the accuracy of the mathematical models.

The accuracy of the models was evaluated by constructing a plot that combined the width of the zone of error and the estimated radii of the mathematical models. The plot was constructed by calculating a new type of residual called the residuals for the accuracy comparison (r_{ac}) that was calculated by adding the value of the residuals (d_i) to the correspondent estimated radius of the mathematical model.

$$r_{ac} = \hat{r} + d_i \tag{8.2}$$

The advantage of using this type of residual was that it allowed the experimenter to understand the position and variability of each mathematical model with respect to the ideal value of the radius of the inspected sphere. Figure 66 confirms the findings obtained by the Tukey test since it shows that the dynamic angle approach model without square root (NSQRT), the model with no angles (NOANG), and Shunmugam's linear model (SL) were the ones that had the highest zones of error.

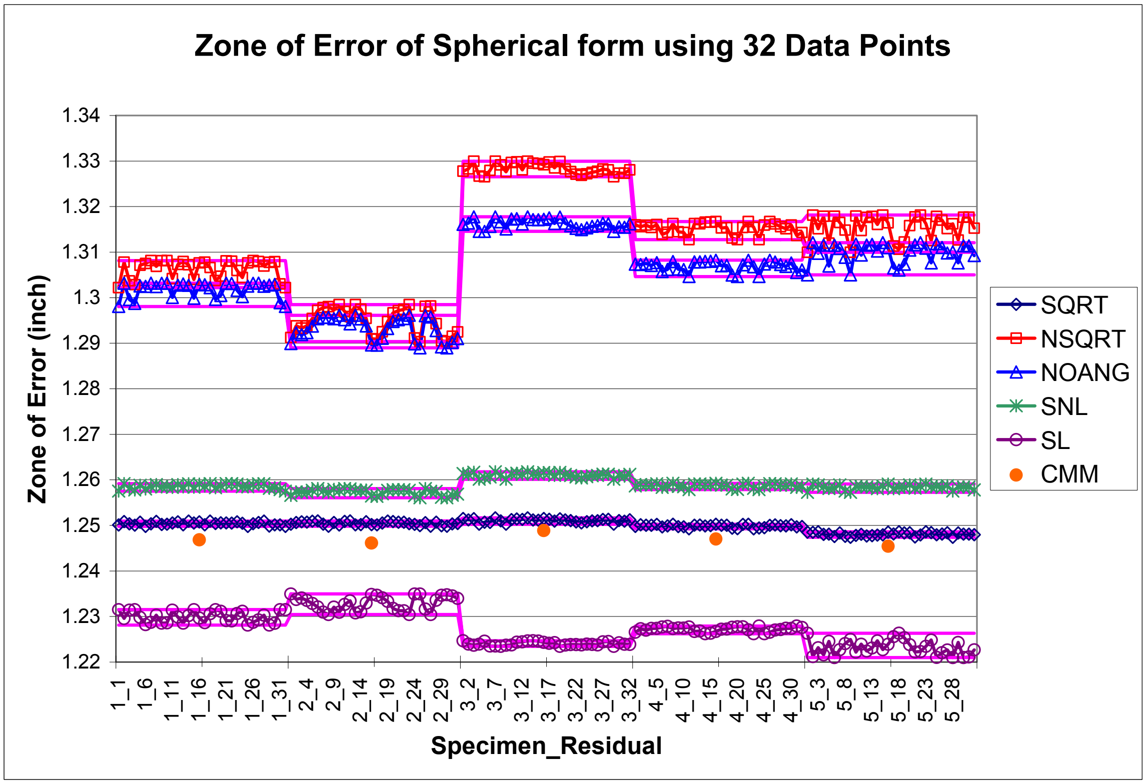


Figure 66. Real zone of error of spherical models.

In addition, it can be seen that the dynamic angle approach model without square root (NSQRT), the model with no angles (NOANG), and Shunmugam's nonlinear model (SNL) overestimated the radius of the sphere; the first two models performed the worst, while the latter overestimated the radius of the sphere by approximately ten microinches.

On the other hand, Shunmugam's linear model (SL) was the only model that underestimated the radius of the sphere. Out of all five models the dynamic angle approach model with square root (SQRT) was the only model that performed similarly to the results obtained from the CMM and the only one that included the value of the ideal radius of the sphere inside the zone of error.

Finally, from Figure 67 it can be seen that the dynamic approach angle model with square root (SQRT) showed the shortest distance between the point (0,0,0) and the geometrical origin of the part. The main reason that this effect happed was because both the measured data and the ideal form are in motion while the fitting algorithm finds the optimal solution. In other models the measurements are used as reference while the ideal form is in motion.

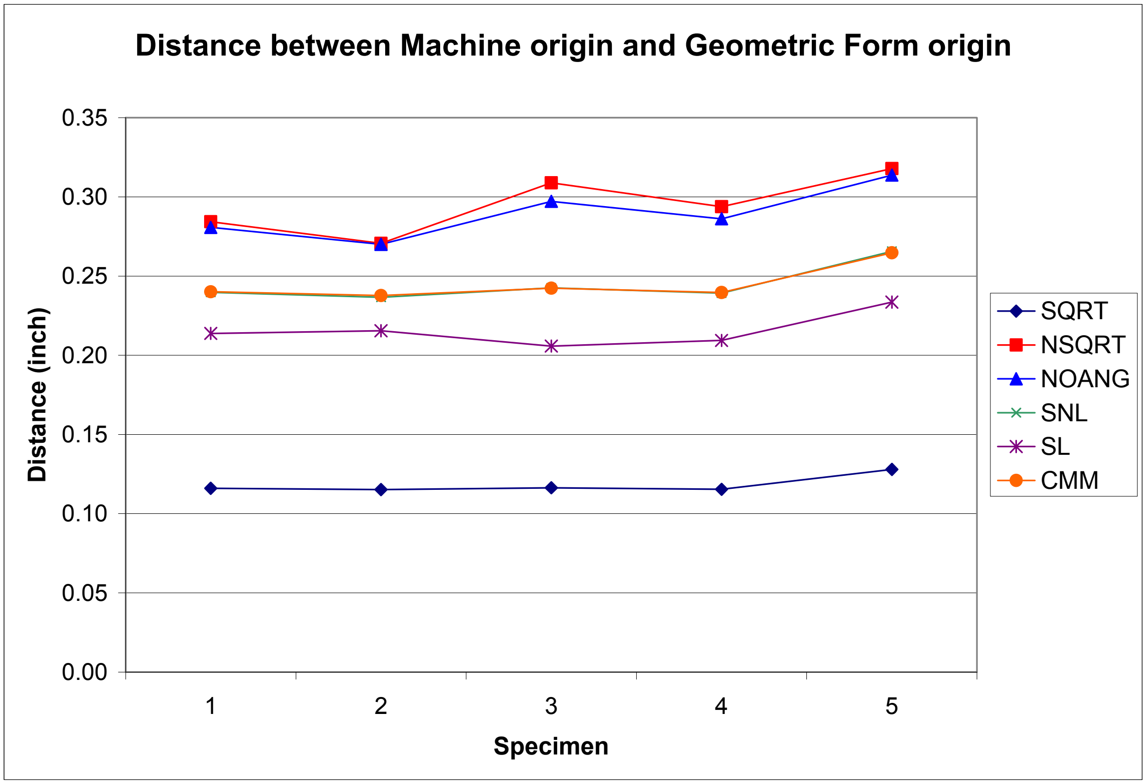


Figure 67. Euclidian distance between the origins of the machine and of the part.

8.2 Results of Cylindrical Form

The performance of the models developed for the cylindrical form was evaluated using two procedures: the first was an experiment similar to those shown above and the second procedure was to apply the data available in the literature to the models and compare the generated parameters against those that were obtained by others.

The first experiment evaluated the cylindrical parts using an induced error, in which the parts were not oriented in the same direction as the coordinate system of the CMM. The axis of the cylinder was oriented approximately [0.033, 0, 1], which was intentionally done to determine the accuracy of the models. The experiment compared the performance of the following models: dynamic angle approach model with square root (SQRT), Shunmugam's linear model (SL), and the nonlinear model with free form orientation capabilities (NL), using four sample sizes of 8, 32, 128, and 512 data points.

The width of the zone of error obtained from the three models was analyzed using the analysis of variance, where the ANOVA assumptions were verified using the tests and plots shown in Appendix D. Results from this analysis showed that the ANOVA model was significant, being that the main factors were significant, while the interaction was not significant. The magnitude of the mean of the main factors was compared using the Tukey test with an alpha value of 0.01, shown in Figures 68 and 69. Results from the main factor sample size showed the means arranged in three groups. The first group was composed of sample size 512. The second group was composed of sample sizes of 32 and 128, while the third group was composed of a sample size of 8 data points. Results for the main factor mathematical models showed the means arranged in two groups: the first group was composed by the dynamic angle approach model with square root (SQRT) and Shunmugam’s linear model (SL), while the second group was composed by the nonlinear model with free form orientation capabilities (NL).

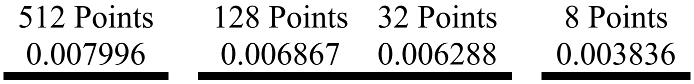


Figure 68. Means comparison of the factor Sample Size of the cylindrical form using the Tukey test.

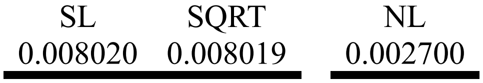


Figure 69. Means comparison of the factor Mathematical Model of the cylindrical form using the Tukey test.

Figure 70 shows the graphical explanation of the width of the zone of error for the three mathematical models and the four sample sizes. The performance of the first two models: the dynamic angle approach with square root (SQRT) and Shunmugam’s linear model (SL) were identical, while the last model the nonlinear model with free form orientation capabilities (NL), was different from them. Although the latter generated the smallest zone of error, all models showed an incremental increase as in the width of the zone of error as the sample size increases. Furthermore, the value “Worst” for cylindrical form was calculated using Equation (8.3), but was not shown in the plots because the values were in the order of tenths of microns.

$$\text{Worst} = \sqrt{x_i^2 + y_i^2} - r \qquad (8.3)$$

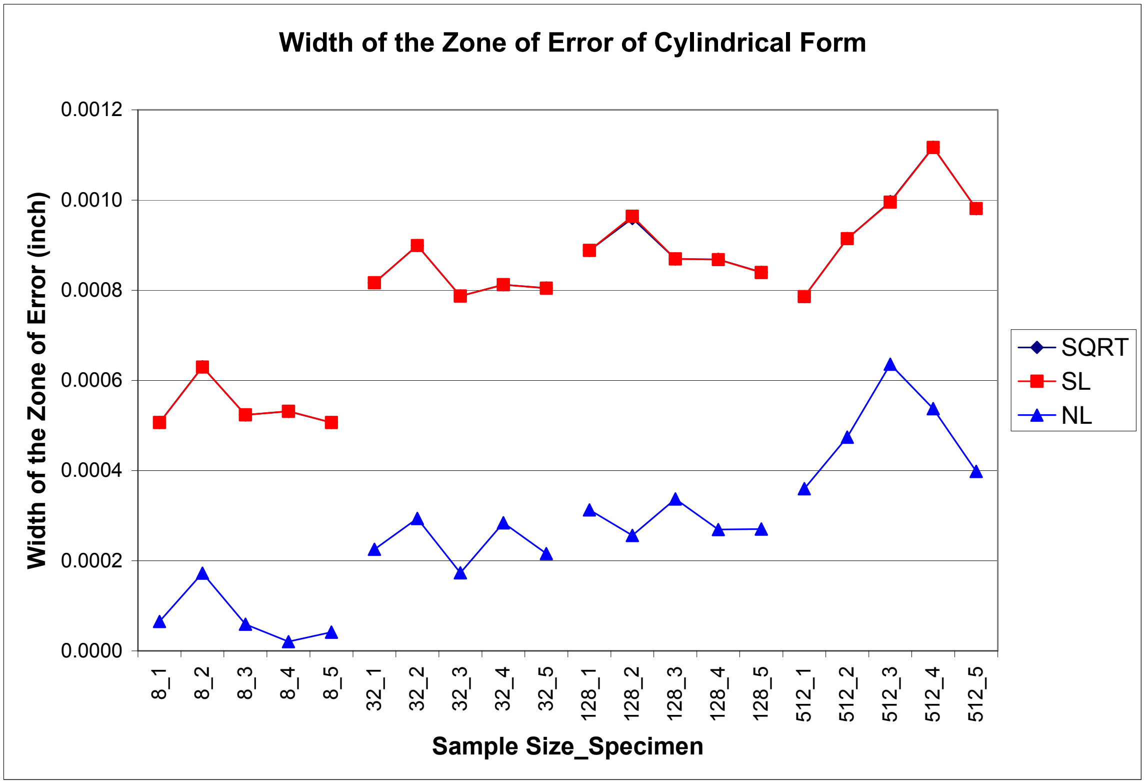


Figure 70. Zones of error of cylindrical form at different sample sizes.

The accuracy of the performance of the models was analyzed using the plot of the real zone of error. The residuals for the accuracy comparison (r_{aci}) were calculated following Equation (8.4), which adds to each residual (d_i) the estimated radius value of the mathematical model.

$$r_{aci} = \hat{r} + d_i \tag{8.4}$$

Figure 71 shows the real zone of error of the cylindrical models. It can be seen that the three models found a solution closer to the ideal radius of the cylinder. The difference between the ideal radius of the cylinder ranges between a few nanoinches for the nonlinear model (NL) and a few microinches for the first two models (SQRT and SL). In addition, it can be seen that the nonlinear model (NL) performed very close to the parameters obtained from the CMM, which means that the model was capable of accurately extracting the inclination and parameters of the cylindrical form.

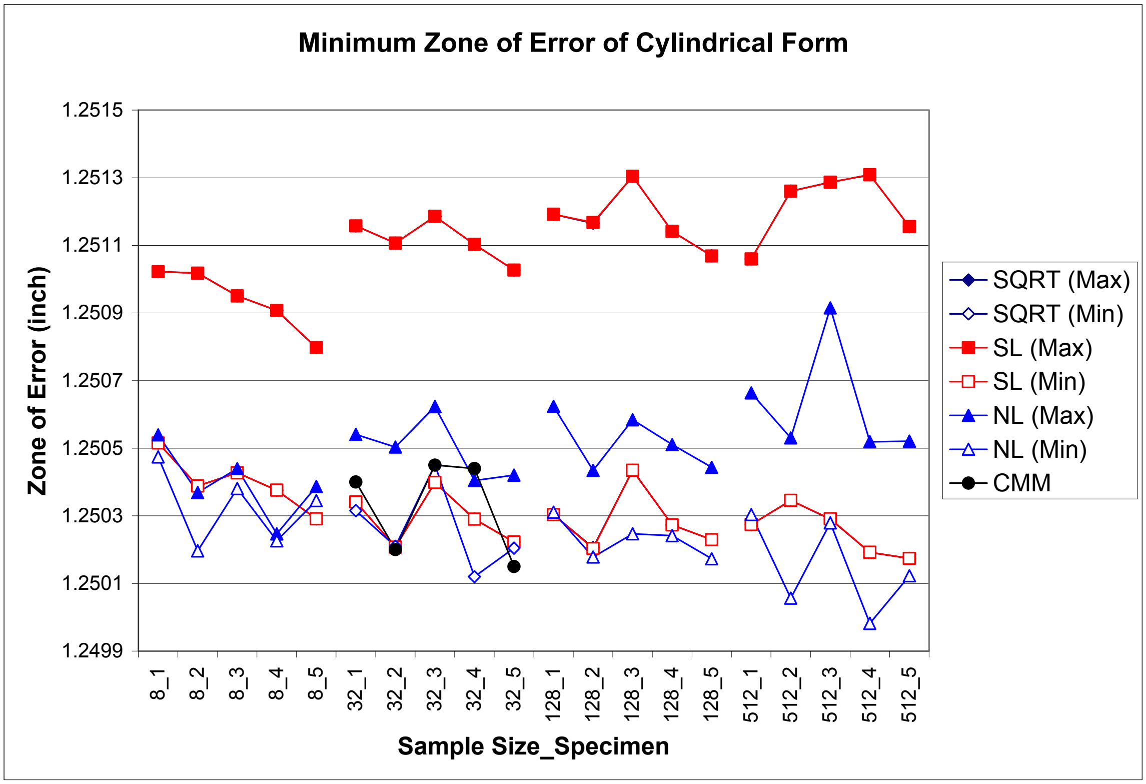


Figure 71. Real zone of error of cylindrical models.

In addition to the experiment mentioned above, the cylindrical models were verified using datasets available in the literature, specifically from Carr et al.'s (1995) and Roy et al.'s (1995) studies. Prakasvudhisarn (2002) presented a comparison of several studies that had analyzed datasets presented in Carr et al.'s (1995) and Roy et al.'s (1995) studies. In this comparison, Prakasvudhisarn (2002) presented the results obtained from his nonlinear model solved using the generalized reduce gradient method, against the results obtained from Carr et al.'s (1995) algorithm, and from Wang's (1992) sequential quadratic programming model. This comparison is used in the present research to determine the performance and accuracy of the dynamic angle approach model with square root (SQRT) and the nonlinear model with free form orientation capabilities (NL).

Carr et al. (1995) studied the cylindrical form by developing a linear programming algorithm to determine the maximum inscribed cylinder, the minimum circumscribed cylinder, and the straightness of the cylinder axis. The authors verified their algorithms by using three datasets of 40, 20, and 40 data points that can be seen in Appendix C. The first dataset was generated by NIST Algorithm Testing System (ATS), where 40 data points were generated randomly from two cylinders that had their axes of revolution perfectly aligned with the z-axis. The cylinders had a radius of 30.000 and 30.001 respectively. Thus, the data had a known cylindricity solution of 0.001. Table 36 shows a comparison of the models and algorithms used by Carr et al. (1995), Wang (1992), Prakasvudhisarn (2002), and the two proposed cylindricity models. The results obtained from this comparison showed that both proposed models outperformed Wang's (1992) and Prakasvudhisarn's (2002) studies. The performance of the nonlinear model with free form orientation capabilities (NL) generated a zone of error very close to the value obtained by Carr et al. (1995).

Table 36. Results obtained from the analysis of Carr et al.'s (1995) dataset 1.

Model	**Carr et al. (1995)**	**Wang (1992)**	**P[1] (2002)**	**DAA (SQRT)**	**FFO (NL)**
Minimum Zone	0.001	0.004809865	0.004809871	0.00380700	0.001425359
r	-	30.00206564	30.00206564	30.000882	30.000366
m	-	8.71539E-06	8.71526E-06	0.000043	5.017E-05
$\boldsymbol{x_0}$	-	0	0	-0.000234	0.000599
n	-	0	0	-0.000057	-9.649E-06
$\mathbf{y_0}$	-	0	0	0.000579	0.0002312
$\boldsymbol{z_0}$	-	-	-	-	-1.599E-05

[1] Prakasvudhisarn (2002)

The last two datasets from Carr et al.'s (1995) study were generated using the NIST Algorithm Testing System (ATS) and had random errors and unknown cylindricity values. Table 37 shows the results obtained from the analysis of the second dataset. Results showed that both proposed models outperformed Carr et al.'s (1995) algorithms, Wang's (1992) model, and Prakasvudhisarn's (2002) model, since the nonlinear model with free form orientation (NL) generated the smallest zone of error.

Table 37. Results obtained from the analysis of Carr et al.'s (1995) dataset 2.

Model	**Carr et al. (1995)**	**Wang (1992)**	**P[1] (2002)**	**DAA (SQRT)**	**FFO (NL)**
Minimum Zone	0.18396	0.194828343	0.194828184	0.036460163	0.031963025
r	-	60.00478139	60.00478136	60.000544	59.996995
m	-0.00062	0	0	0.001915	0.0018534
$\boldsymbol{x_0}$	-	0.005930704	0.00593079	-0.009502	0.0183155
n	-0.00292	0	0	0.001025	0.0011124
$\boldsymbol{y_0}$	-	0	0	-0.011912	0.0246282
$\boldsymbol{z_0}$	-	-	-	-	0.0037978

[1] Prakasvudhisarn (2002)

The third dataset was slightly different than the first two datasets, since it had a cylinder that was not perfectly aligned with the *z*-axis. Therefore the dynamic angle approach with square root was not used during this comparison. Table 38 compares the performance of the four models and shows that the nonlinear model with free form orientation capabilities (NL) outperformed the parameters from the previous studies, since it generated the smallest zone of error and produced a good estimate of the radius of the cylinder.

Table 38. Results obtained from the analysis of Carr et al.'s (1995) dataset 3.

Model	**Carr et al. (1995)**	**Wang (1992)**	**P[1](2002)**	**DAA (SQRT)**	**FFO (NL)**
Minimum Zone	0.00941	0.009410136	0.009410193	-	0.003367416
r	-	49.99953289	49.99953288	-	50.00032
m	1.000052	1.000058463	1.000058461	-	1.000249
x_0	-	3.998826419	3.998826401	-	1.748899
n	1.000052	1.000046417	1.000046416	-	1.000049
y_0	-	4.998937542	4.998637497	-	2.748737
z_0	-	-	-	-	-2.24918

[1] Prakasvudhisarn (2002)

The last dataset was obtained from Roy et al.'s (1995) study which was applied to the proposed models in order to compare their performances against studies from Roy et al. (1995), Wang (1992), and Prakasvudhisarn (2002). Table 39 provides a summary of this comparison, which showed that the dynamic angle approach model with square root (SQRT) and the nonlinear model with free form orientation (NL) outperformed the results provided by other studies. In addition, it can be seen that the latter showed the smallest zone of error.

Table 39. Results obtained from the analysis of Roy and Zu's (1995) dataset.

Model	**Roy et al. (1995)**	**Wang (1992)**	**P[1] (2002)**	**DAA (SQRT)**	**FFO (NL)**
Minimum Zone	0.02881336	0.029548125	0.029548126	0.026281926	0.011538552
r	-	5.000751946	5.000751947	5.0000452	4.998422
m	0.001567376	0.001017511	0.001017512	0.0011712	0.002661
x_0	-0.00911603	0	0	-0.0011626	-0.006810
n	0.00054574	0.000104874	0.000104875	-8.925E-05	-0.001700
y_0	-0.00182085	0	0	-0.0020074	0.006266
z_0	-	-	-	-	0.001682

[1] Prakasvudhisarn (2002)

8.3 Results of Conical Form

The conical form was evaluated using only one experiment in which the cone was oriented perfectly with the coordinate system of the machine. However, a second experiment with the conical form tilted to the left side was planned but not performed due to the fact that the slope of the cone was too steep and could have caused the probe head of the CMM to collide with the cone surface.

The experiment included as independent variables the sample size with four levels:8, 32, 128, and 512, and the mathematical models with three levels: dynamic angle approach with square root (SQRT) and without square root (NSQRT), and the nonlinear model with free form orientation capabilities (NL). The width of the zone of error of the models was chosen as the dependent variable.

The width of the zone of error was analyzed using the analysis of variance in order to determine if the mathematical models were statistically different. The ANOVA assumptions were validated by analyzing the plots of normality and equal variances, as shown in Appendix D. The results obtained from the analysis of variance showed that the main factors as well as the interaction between them were statistically significant. The plots of the main factors are found in Appendix D, while the interaction plot can be seen in Figure 72.

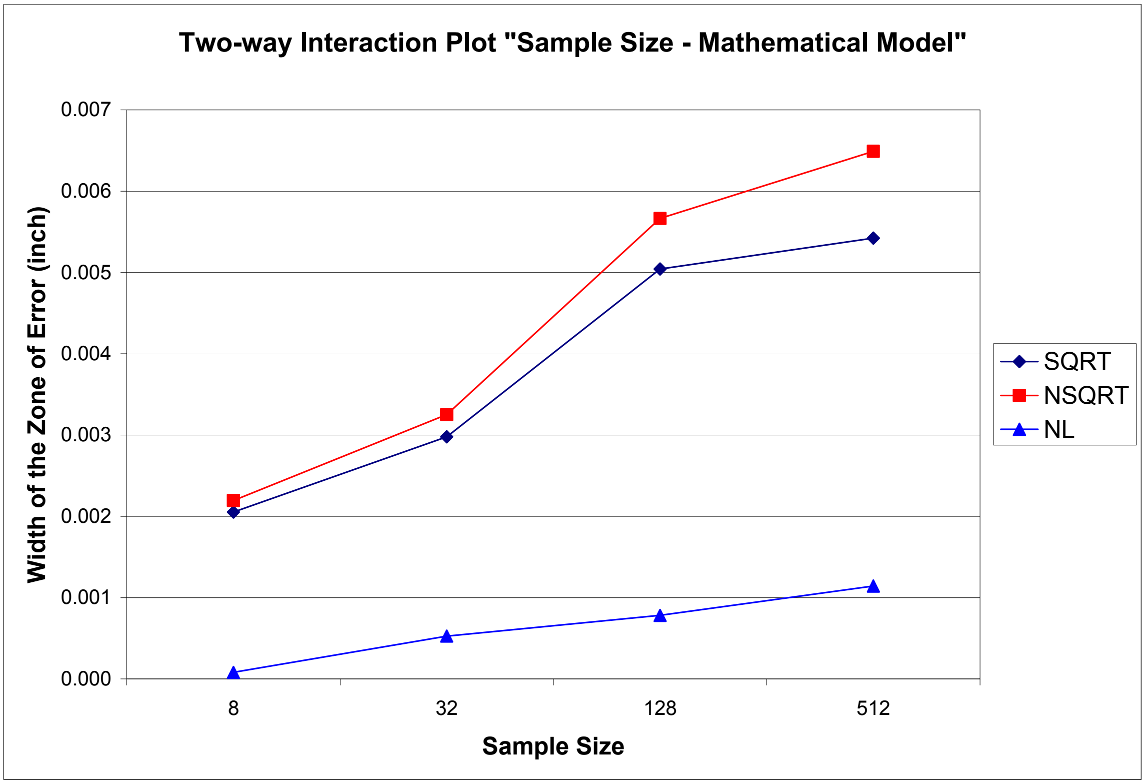


Figure 72. Two way interaction plot between sample size and mathematical model.

The means of the two main factors were analyzed using the Tukey test with an alpha value of 0.01, as shown in Figures 73 and 74. Results for sample size showed that the means were arranged in two groups; the first group was the sample sizes of 8 and 32, and the second group was 128 and 512. The first group generated the smallest width of the zone of error. Results for the mathematical model showed that the two dynamic angle approach models (SQRT and NSQRT) were not significantly different. The nonlinear model with free form orientation capabilities (NL) was the only significant model, since it produced a smaller zone of error than the other two models. The effect of sample size on the former was smaller than in the other two models, making the former less sensitive to increments in sample size, as seen in the interaction plot.

512 Points	128 Points	32 Points	8 Points
0.004352	0.003829	0.002252	0.001443

Figure 73. Means comparison of the factor Sample Size of conical form using the Tukey test.

NSQRT	SQRT	NL
0.004400	0.003874	0.001443

Figure 74. Means comparison of the factor Mathematical Model of the conical form using the Tukey test.

The accuracy of the performance of the models was also analyzed, since it is important to verify that the models produced not only a small zone of error but also accurate parameters that describe the shape of the cone that is being inspected. The analysis was done by plotting the width of the zone of error of each factor combination, as well as the real zone of error of the cone. Figure 75 shows a plot that describes the performance of the models at different sample sizes, which included the value "Worst". This value was determined by using Equation (8.5), which assumes that the measurements were perfectly aligned with the *z*-axis and without any origin offsets. The purpose of this value was to determine the accuracy of the measurements, similar to the analysis done for the spherical forms.

$$\text{Worst} = \left(\frac{r}{h}\right)\left(\sqrt{x_i^2 + y_i^2 + z_i^2} - |z_i|\left(\frac{r^2 + h^2}{h\sqrt{r^2 + h^2}}\right)\right) \qquad (8.5)$$

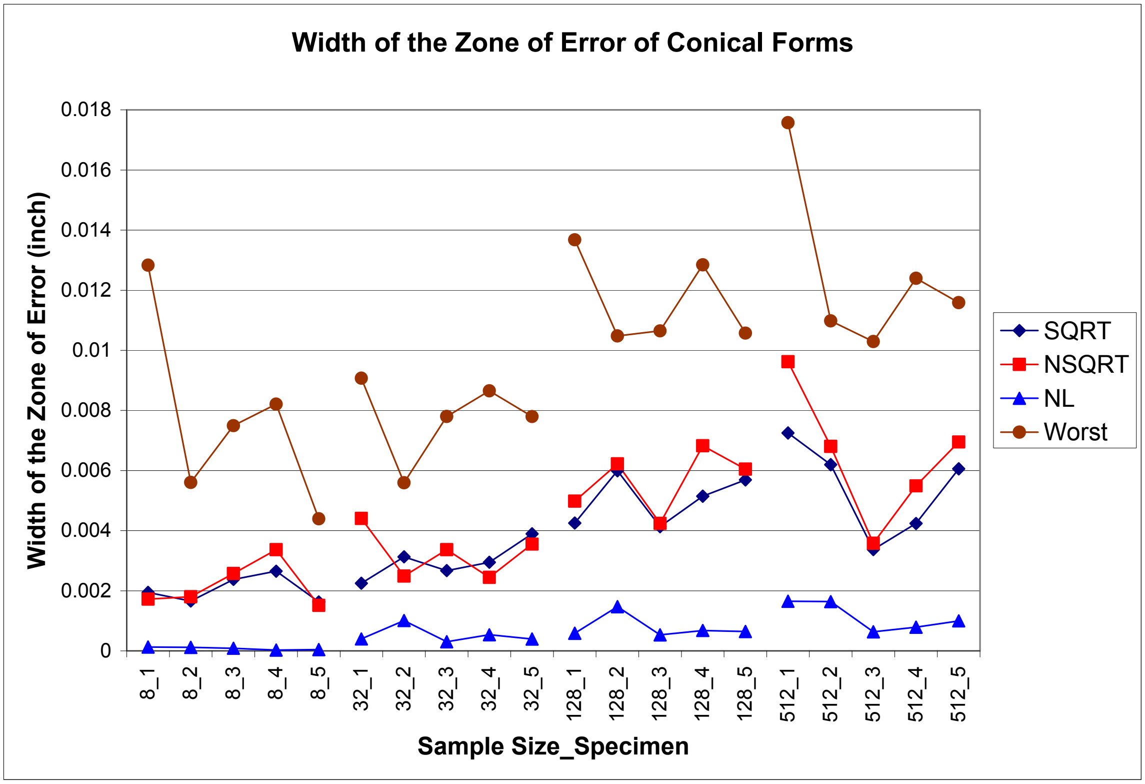


Figure 75. Zones of error of conical form at different sample sizes.

Figure 75 shows the same effect found on the interaction plot, where the zone of error of the models increases as the sample size increases. All models performed better than the value "Worst" showing that the models were able to find the axis misalignments and origin offsets from the data. Out of the three models, the nonlinear model (NL) had the lowest width of the zone of error compared to the two dynamic angle approach models (SQRT and NSQRT). It can be seen that the performance of the last two models was more defined at larger sample sizes than at low sample sizes. At large sample sizes (128 and 512) the (SQRT) model performed slightly better than the (NSQRT), while at low sample sizes (8 and 32) both produced mixed results.

The second analysis was done by plotting the real zone of error that studied the accuracy and precision of the performance of the mathematical models. The plot was created by using a formula that defined the zone of error of the cone. Therefore, the residuals of the conical form were calculated using the formula developed in Chapter 6 for the ideal maximum normal distance ($a_{i\max}$). This parameter was chosen because it combines the three main parameters of the conical form: the radius of the base (r), the height of the cone (h), and the angle (α) that describe the relation between the radius and the height of the cone. Thus, the residuals for the accuracy comparison (r_{aci}) were calculated by adding to each estimated normal distance ($a_{i\max}$) the value of each residual (d_i), which can be visualized as a Christmas tree, as seen in Figure 76.

$$r_{aci} = \frac{\hat{r}}{\cos\left(\tan^{-1}\left(\frac{\hat{r}}{\hat{h}}\right)\right)} + d_i \tag{8.6}$$

where the estimated radius and height are the values obtained in each of the mathematical models. For example, all residuals obtained from the factor combination, sample size 8 and model (SQRT), were transformed by adding to them the estimated normal distance that was obtained based on the values of the parameters from the model (SQRT). Thus, each factor combination should have different set of parameters.

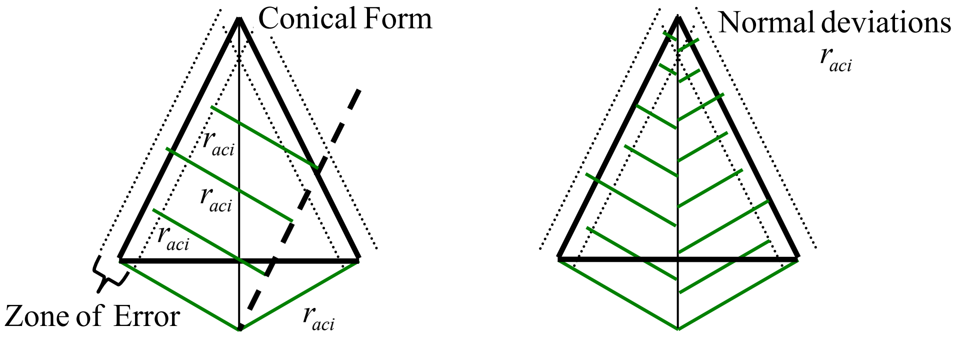


Figure 76. Visualization of the zone of error of conical forms.

Figure 77 shows the real zone of error of the conical form. Although the models were different in performance regarding the width of the zone of error, they generated comparable results regarding the estimated parameters of the cone. All models produced maximum normal distances lower than the ideal value. The ideal value was calculated by substituting the ideal parameters of the cone on Equation (8.6). However, it is unknown if the conical parts were machined according to specifications. Therefore, it is not possible to determine whether the models underestimated the parameters of the cone. Finally, it can be seen that for all sample sizes the three models generated comparable results; this means that all models recognized specimens one and four as the worst machined and specimen two and five as the best machined.

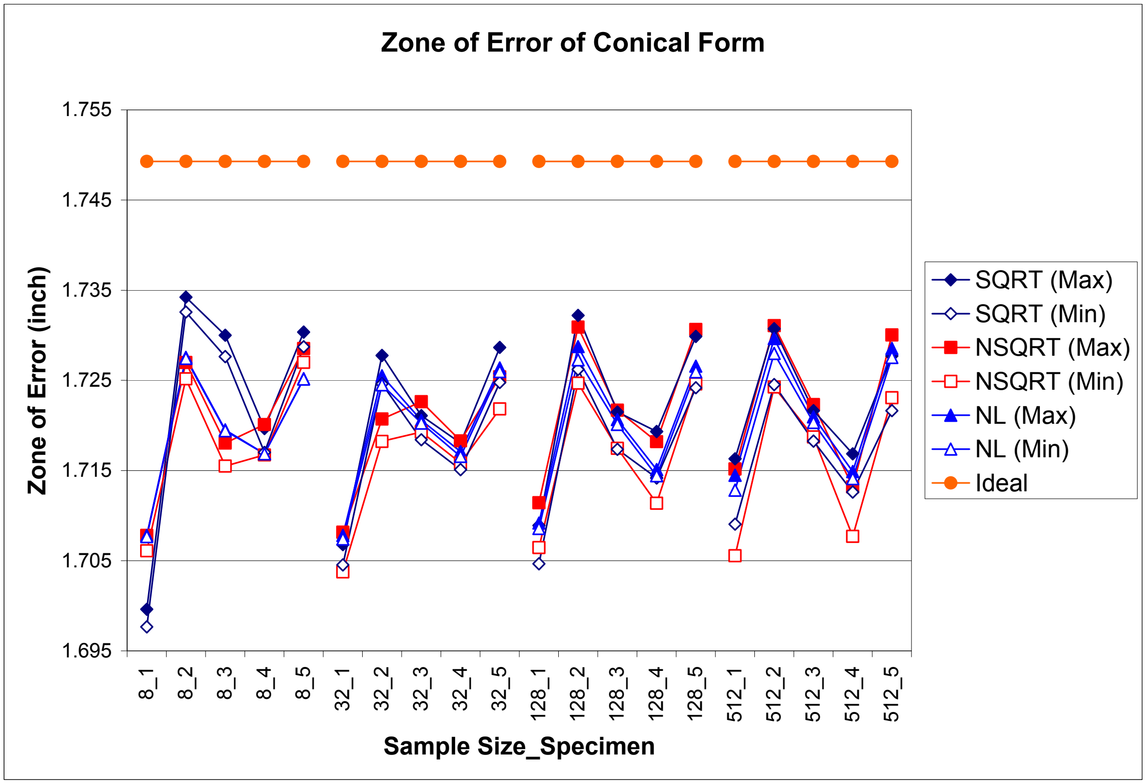


Figure 77. Real zone of error of conical models.

8.4 Results of Frustum Form

The performance of the frustum models was evaluated using one experiment, which used five frustum parts that were perfectly aligned with the coordinate system and origin of the coordinate measuring machine and were inspected using four sample sizes of 8, 32, 128 and 512. The mathematical models that were evaluated were the dynamic angle approach without square root (NSQRT) and the nonlinear model with free form orientation capabilities (NL).

The width of the zone of error obtained from the two models was analyzed using the statistical analysis of variance, where the ANOVA assumptions were verified using the tests and plots shown in Appendix D. Results showed that the ANOVA model was significant, and the main factor sample size was the only significant term of the model. The means of the sample size were further analyzed using the Tukey test with an alpha value of 0.01, shown in the Figure 78. The test showed that the means of the sample size were arranged in three groups. The first group was composed of the sample sizes of 512 and 128 data points. The second group was composed of the sample sizes of 128 and 32 data points, while the third group was composed of the sample sizes of 32 and 8 data points.

512 Points	128 Points	32 Points	8 Points
0.009367	0.007420	0.004896	0.002564

Figure 78. Means comparison of the factor Sample Size of the frustum form using the Tukey test.

Figure 79 shows the effect that the mathematical model and the sample size has over the width of the zone of error. It can be seen that the width of the zone of error increases as the sample size increases, agreeing with the findings of the Tukey test.

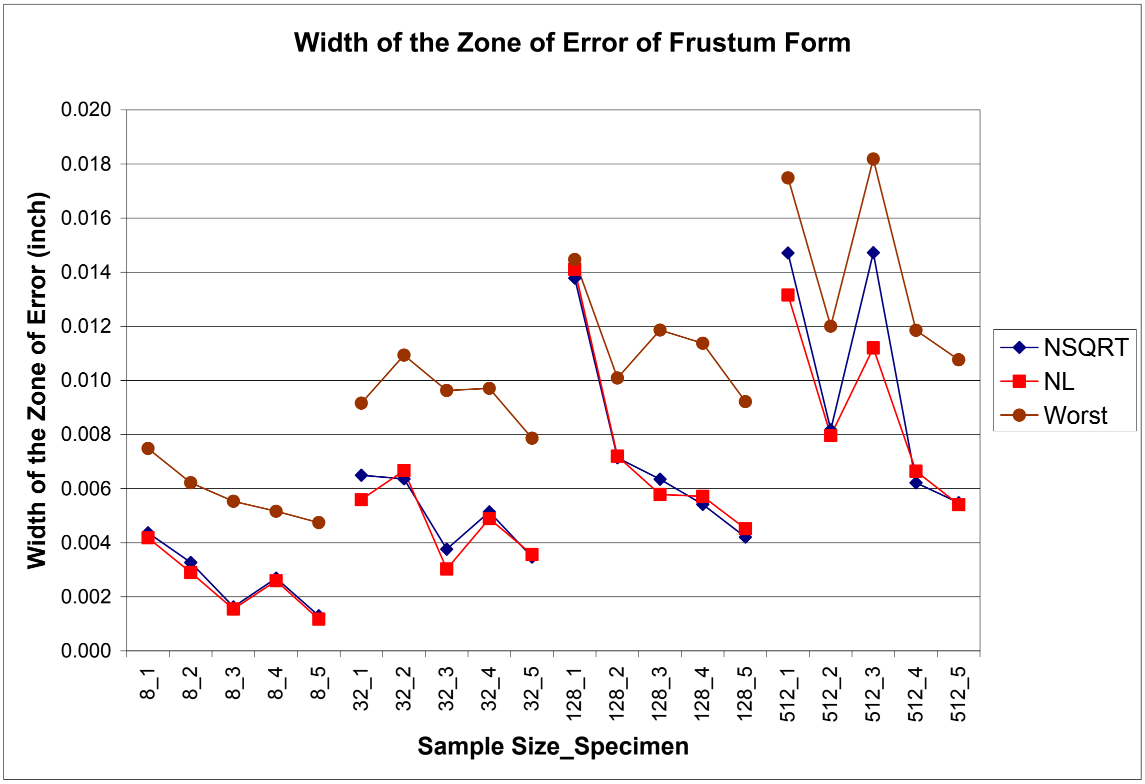


Figure 79. Zones of error of frustum form at different sample sizes.

Furthermore, the statistical analysis showed that there were no differences in the performance of the two mathematical models, and it can be seen that both models accurately identified the origin offsets and axis misalignments since both performed better than the value "Worst." The latter was calculated using Equation (8.7), which assumed that the frustum was perfectly aligned with the coordinate system of the machine.

$$\text{Worst} = \frac{r''}{h''}\left[\sqrt{x_i^{\,2} + y_i^{\,2} + \left(z_i - h'\right)^2} - \left[\left[\left|z_i\right|\left(\frac{r''}{h''}\right) + r'\right]\left(\frac{\sqrt{r'^2 + h'^2}}{r'}\right)\right]\right] \qquad (8.7)$$

The data from this experiment was also analyzed using a plot of the real zone of error that described the accuracy and precision of the performance of the mathematical models. In a similar way as the conical feature, the plot of the frustum form was created by using a formula that defined the zone of error of the cone. Therefore, the residuals of the frustum form were calculated using the formula developed in Chapter 6 for the ideal maximum normal distance ($a_{i\max}$). This formula was chosen because it combines all the parameters used to generate a frustum form. Thus, the residuals for the accuracy comparison (r_{aci}) were calculated by adding to the estimated normal distance ($a_{i\max}$) the value of each residual (d_i).

$$r_{aci} = \frac{\left(\hat{r}' + \hat{r}''\right)}{\cos\left(\tan^{-1}\left(\frac{\hat{r}' + \hat{r}''}{\hat{h}' + \hat{h}''}\right)\right)} + d_i \qquad (8.8)$$

Figure 80 shows the plot of the real zone of error of the frustum forms. It can be seen that the accuracy of the two developed models is very high since both models were able to obtain results that were very close to the ideal maximum normal distance of the frustum form. The nonlinear model with free form orientation capabilities (NL) performed the best because it showed less variability and it was less affected by the location of the data points. The dynamic angle approach without square root (NSQRT) was more affected by the location of the data points because it produced different widths of the zone of error for each of the specimens at different sample sizes, while the nonlinear model was more consistent in its results.

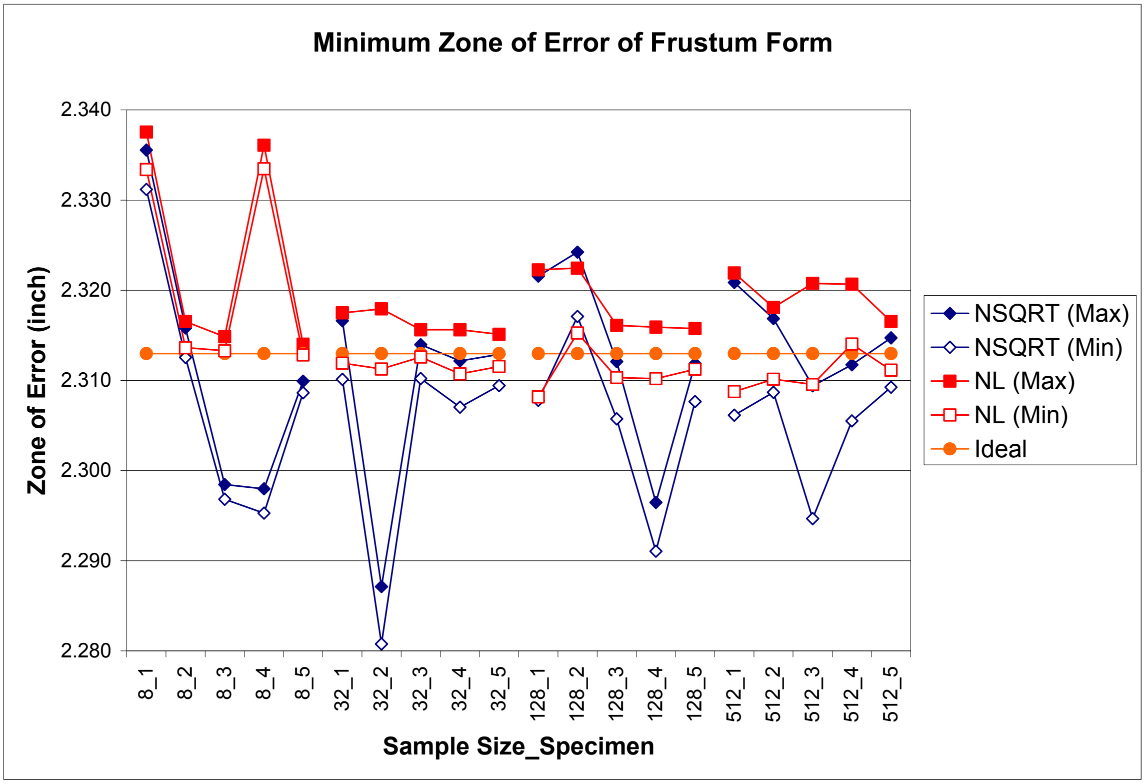


Figure 80. Real zone of error of frustum models.

8.5 Results of Torus Form

The mathematical models developed for the torus form were evaluated using two experiments. The first experiment consisted of inspecting five torus forms using four sample sizes of 8, 32, 128, and 512 data points, where the parts were aligned perfectly with the coordinate system and origin of the coordinate measuring machine. The mathematical models used were the dynamic angle approach model with square root (SQRT) and without square root (NSQRT), Aguirre Cruz‘s thesis model (Thesis), and the nonlinear model with free form orientation capabilities (NL). The calculated width of the zone of error of each model was analyzed using the method of the analysis of variance. The ANOVA assumptions were analyzed using the tests and plots shown in Appendix D.

The normality assumption was violated on this experiment; however ANOVA is very robust to any deviation from normality (Montgomery, 2001). The ANOVA model was found to be significant, thus the means of the main factors were further analyzed using the Tukey test with an alpha value of 0.01, shown in Figures 81 and 82. It is important to mention that the interaction between the two main factors was not significant. Results for the main factor sample size showed the means arranged in three groups. The first group was composed of the sample sizes of 512 and 128 data points. The second group was composed of the sample size of 128 and 32 data points, while the third group was composed of the sample size of 8 data points. The last group generated the smallest width of the zone of error. Results obtained from the main factor mathematical model showed the mean arranged in two groups. The first group was composed of the nonlinear model with free form orientation capabilities (NL) and Aguirre Cruz's thesis model (Thesis), while the second group was composed of Aguirre Cruz's thesis model (Thesis), the dynamic angle approach with square root (SQRT) and the model without square root (NSQRT).

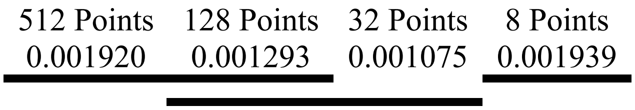


Figure 81. Means comparison of the factor Sample Size of the torus form using the Tukey test. (Experiment I)

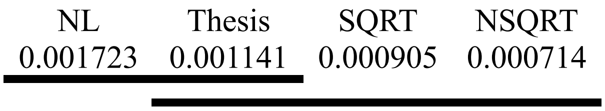


Figure 82. Means comparison of factor Mathematical Model of the torus form using the Tukey test. (Experiment I)

Figure 83 shows the width of the zone of error for all factor combinations as well as for the value "Worst." The latter was calculated using Equation (8.9), which assumes that the data comes from a perfectly aligned torus part.

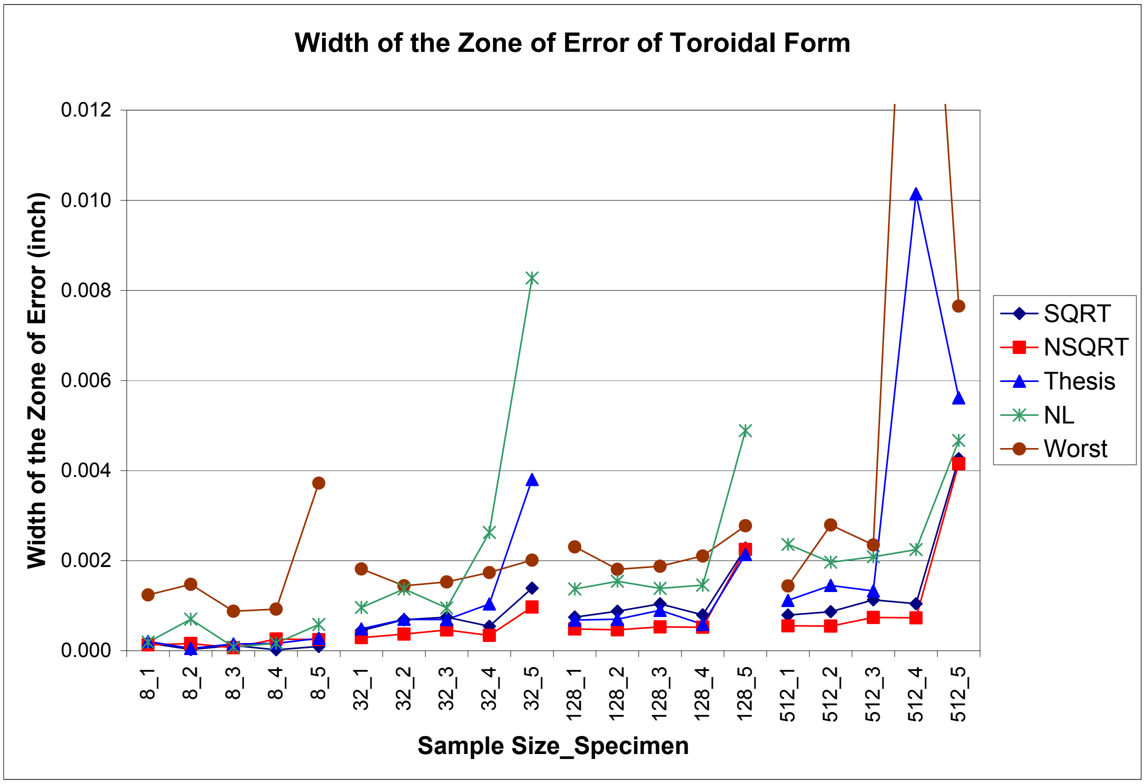


Figure 83. Width of the zone of error of torus forms. (Experiment I)

In general, it can be seen that for all models the width increases as the sample size increases. The sample size of 8 data points was the only level in which all the models performed better than the value "Worst." For the remaining sample sizes, four out of the five models performed better than the value "Worst" with the nonlinear approach with free form orientation capabilities (NL) performing the worst. On the other hand, the dynamic angle approach models with square root (SQRT) and without square root (NSQRT) generated lower widths of the zone of error, with the latter the smallest. The Aguirre Cruz's thesis model (Thesis) was very sensitive to certain data points, since for some of the specimens it produced very large zones of error; however the generated widths were above the dynamic angle approach models but below the nonlinear model.

$$\text{Worst} = \cos(\varphi_i - \alpha_i)\left[\sqrt{x_i^2 + y_i^2 + z_i^2} - \left[\frac{(c \pm a\cos(\varphi_i))}{\cos(\alpha_i)}\right]\right] \tag{8.9}$$

The accuracy of the models was evaluated by verifying the real zone of error of the torus. The plot shown in Figure 84 shows the zone of error defined by the radius of the torus tube (*a*). It can be seen that the model that had the worst performance was the dynamic angle approach without square root (NSQRT). The model underestimated the radius of the torus tube for smaller sample sizes but started to improve its performance as the sample size increased. Aguirre Cruz's thesis model (Thesis) was consistent through all sample sizes, since it generated radii of the torus tube that were very close in magnitude to those of the ideal form (0.75 inch). However, it can be seen that the model was too sensitive to outliers. The dynamic angle approach with square root (SQRT) and the nonlinear model with free form orientation capabilities (NL) performed very badly for the sample size of 8 data points but started to improve their performance as the sample size increased. However, the latter had a higher width of the zone of error.

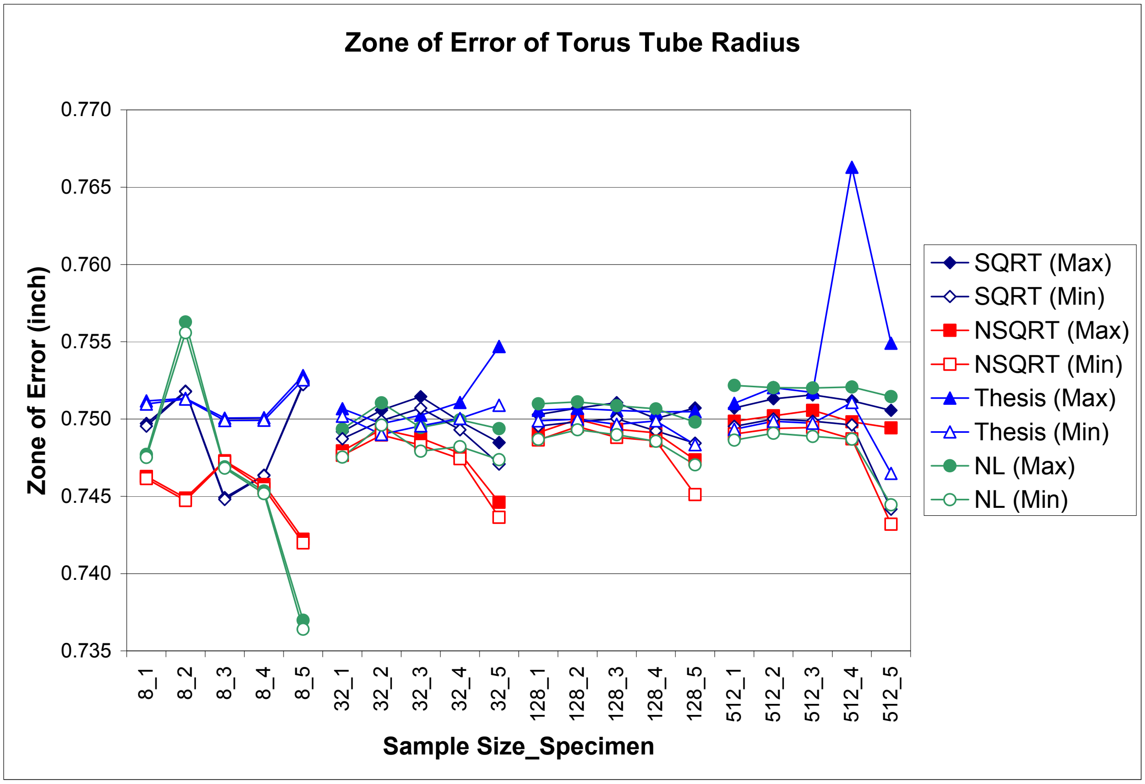


Figure 84. Real zone of error of the torus using the torus tube radius. (Experiment I)

The second experiment consisted of inspecting the same five torus parts using the sample size of 32 data points, where the parts were intentionally tilted in order to determine if the models were capable of recognizing the axis misalignment. The four mathematical models used to analyze the measured data points were the same models used in the experiment above.

The width of the zone of error generated by each of the mathematical models was evaluated using the analysis of variance. The ANOVA assumptions were analyzed using the tests and plots shown in Appendix D. The ANOVA model was found to be significant, thus the means of the main factors were further analyzed using the Tukey test with an alpha value of 0.01, shown in Figure 85. Results for the main factor mathematical model showed the means arranged in three groups: the first group was composed of the dynamic angle approach with square root (SQRT) and Aguirre Cruz’s thesis model (Thesis). The second group was composed by the dynamic angle approach model without square root (NSQRT), while the third group was composed of the nonlinear model with free form orientation capabilities (NL). The third model produced the smallest width of the zone of error.

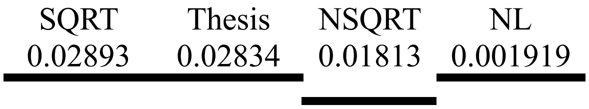


Figure 85. Means comparison of factor Mathematical Model for the torus form using the Tukey test. (Experiment II)

Figure 86 shows the behavior of the width of the zone of error for each of the mathematical models. It can be seen that all models performed better than the value “Worst,” and the nonlinear model with free form orientation capabilities (NL) produced the smallest width of the zone of error. The dynamic angle approach with square root (SQRT) and Aguirre Cruz’s thesis model (Thesis) produced similar widths of the zone of error, which were statistically significant as pointed out by the Tukey test. Finally, the dynamic angle approach without square root (NSQRT) produced widths of the zone of error that were slightly smaller than the two previously mentioned models but higher than the nonlinear model.

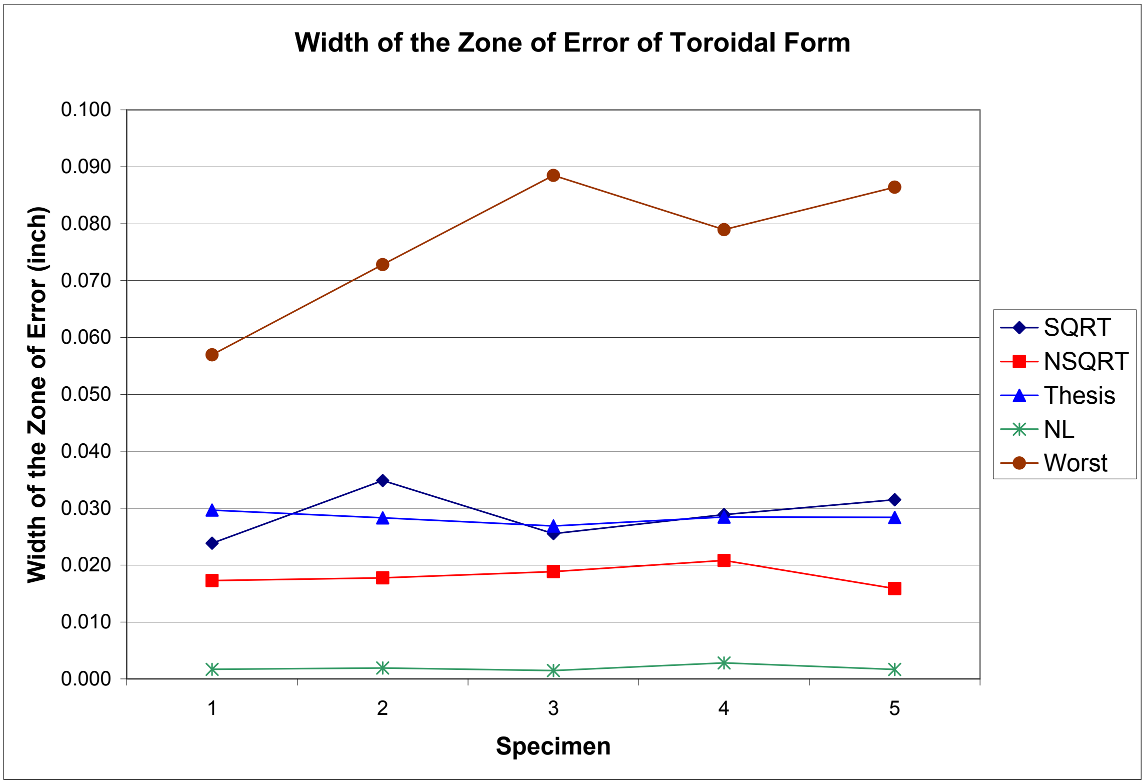


Figure 86. Width of the zone of error of torus forms. (Experiment II)

The accuracy of the models was evaluated by verifying the real zone of error of the torus. The plot shown in Figure 87 shows the zone of error defined by the radius of the torus tube (*a*). It can be seen that the two dynamic angle approach models produced the worst performance since they were very sensitive to the inclination of the torus parts. On the other hand, the nonlinear model with free form orientation (NL) performed the best, because the model was consistent for all specimens and it produced values of the radii of the torus tube that were very close to the ideal value. Finally, Aguirre Cruz's thesis model (Thesis) was also consistent for all specimens; however, this model overestimated the radius of the torus tube.

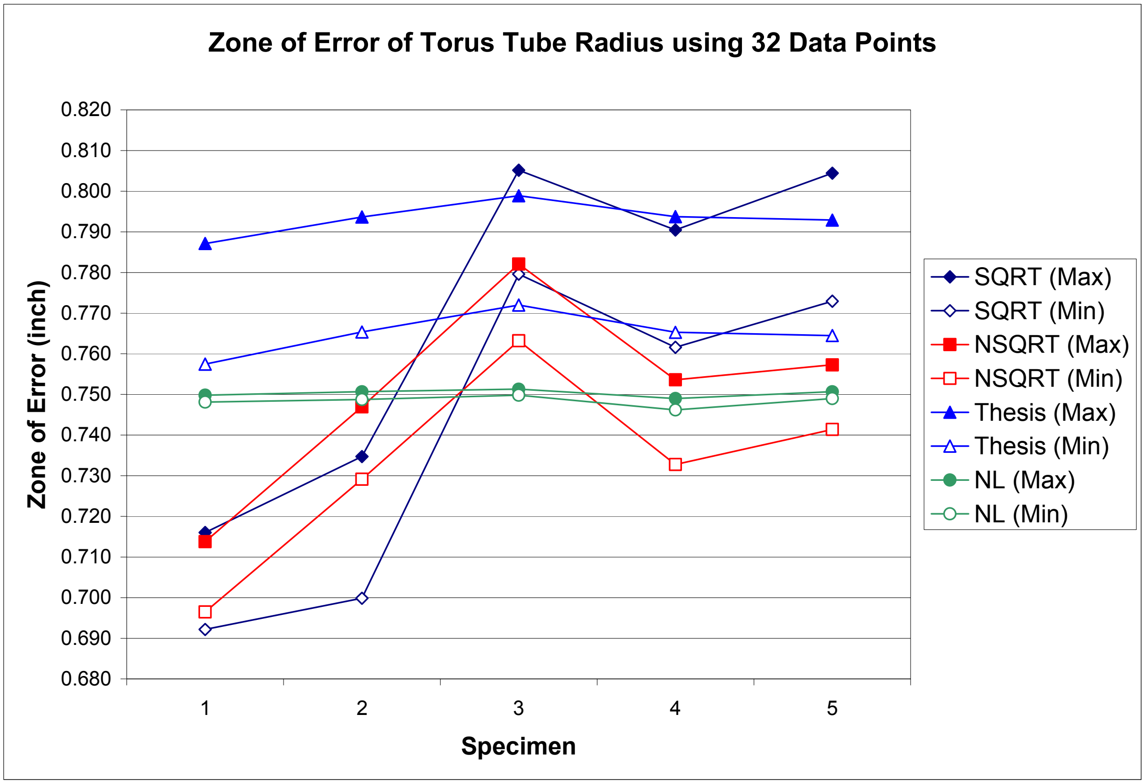


Figure 87. Real zone of error of torus forms. (Experiment II)

8.6 Results of Multiple Feature Part

As a sample, the spherical and cylindrical features contained in a single part were used to verify the multifeature part. The parts were inspected using a sample size of 32 data points, which were analyzed using the dynamic angle approach with square root (SQRT) for the spherical form, and the nonlinear model with free form orientation capabilities (NL) for the cylindrical form. The main purpose of the analysis of multiple feature parts was to identify if the width of the zone of error of individual features was the same as the width of the zone of error obtained when analyzing both features at the same time.

The analysis of the width of the zone of error of individual features was identical as the analyses shown above, while the analysis of multiple features consisted of solving both mathematical models simultaneously, which in this case are the models of the sphere and the cylinder. The two mathematical models had in common the variables of the origin offsets and its radii. Thus, the only difference between the two models was the variables of the axis of assessment from the cylindrical model. Results obtained from the individual and multiple feature analyses are shown in the Figures 88 and 89. The first figure shows the width of the zone of error for all the features as well as for the value "Worst." The latter was obtained by combining the two mathematical models of the two features, substituting the ideal radius into the model, and setting the origin offsets and axis misalignments to zero. It can be seen that the width of the zone of error of the individual analyses for both features was smaller than the width obtained from the multiple feature analysis. At the same time, both individual and multiple feature widths of the zone of error were smaller than the width provided by the value "Worst."

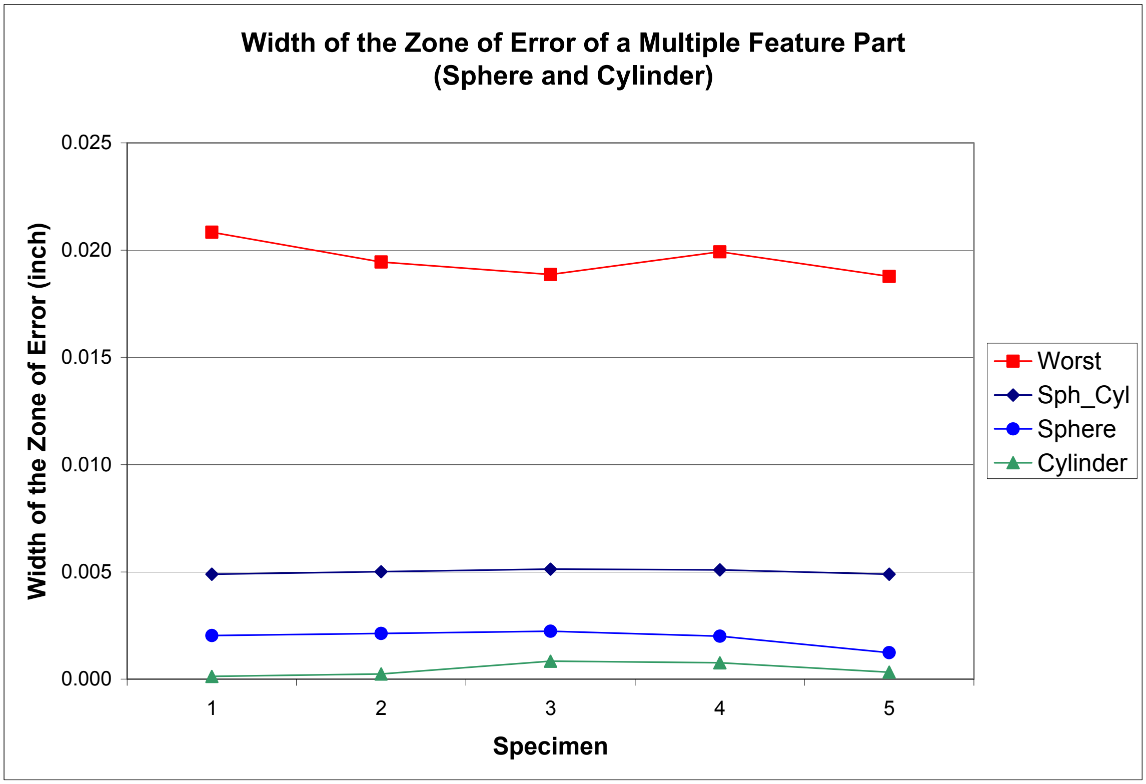


Figure 88. Width of the zone of error of multiple feature part.

The accuracy of both analyses was verified by plotting the real zone of error of individual and multiple feature parts. Figure 89 shows that both analyses included the ideal radii of the sphere and the cylinder inside the zone of error, however the difference between the performances of the two analyses was the dimension of the width of the zone of error, with the zone of error of multiple features wider than the individual zones.

In addition, results failed to show that the zone of error of multiple features is given by the maximum zone of error of the individual features. One of the reasons that this might have happened is due to the fact that the two features were not machined using a single machining sequence, thus causing them to have their geometrical origins in different locations.

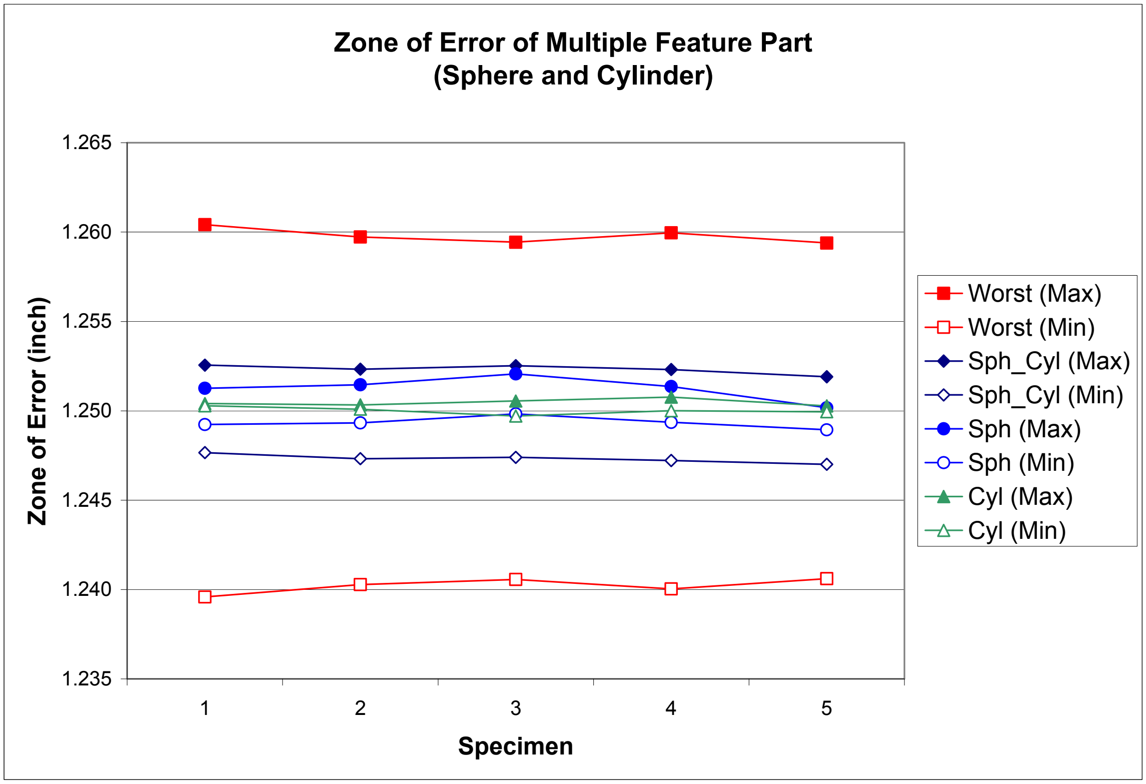


Figure 89. Real zone of error of multiple feature part.

CHAPTER 9

CONTRIBUTIONS, CONCLUSIONS, RECOMMENDATIONS, AND FUTURE WORK

9 Summary

The guidelines and mathematical models presented in this research contribute to a better understanding of the form verification of complex features on manufactured parts. Although much work needs to be done to completely determine the accuracy of machined parts, the conclusions from this research show that simple and robust mathematical models can be used to estimate inspection parameters and form tolerances that are close to optimal solutions.

9.1 Contributions

The main contributions of this research belong to the last two phases of the decision support system presented in this thesis. New mathematical models and approaches were developed to improve the estimation of the form tolerance of manufactured parts in these phases. These contributions are summarized below.

1. Two approaches were developed to counteract the errors induced by the inspection process: the dynamic angle approach (DAA) and the free form orientation (FFO) approach. The dynamic angle approach was developed with the purpose of allowing the fitting algorithm to dynamically calculate the magnitude of the angles that minimizes the zone of error, instead of using them as a constant value. This approach had

two alternatives: one that described the measurements as a Euclidian distance affected by the origin offsets (SQRT) and another that used the measurements as a constant term (NSQRT). In theory the first alternative should be more accurate since it ensures that the origin of the machine and the origin of the part will be closer. The second alternative focuses on identifying the distance between the origin of the machine and the origin of the part that minimizes the deviations. The free form orientation approach was developed to recognize the position and orientation of the part anywhere in the 3-dimensional space. The approach applied the concepts of the flatness model that allows it to counteract the effect of the axis misalignments, and the principles of the dynamic angle approach with square root (SQRT) alternative that allows it to calculate the angles in a dynamic way counteracting the origin offsets.

2. New mathematical models were developed to inspect cylindrical, conical, frustum and torus forms. The models were created by using a finite number of spheres to measure the parts. The advantage of using this approach is that the parameters of these forms are simpler to calculate since they are no longer constrained by the axis of revolution but only by a central point.

3. Two sets of analysis were performed to evaluate the performance of the mathematical models. The first analysis was done by using the width of the zone of error that determined the variability of the residuals. The second analysis was done by evaluating the estimated parameters of the features using the “residuals for the accuracy comparison” that determined the accuracy and precision of the models. In addition, the analyses were carried out by comparing the performance of the models against the value

"Worst" that allows the metrologist to analyze the impact that the axis misalignments and origin offsets have on the mathematical models.

4. The minimum zone of error was evaluated using the "residual for the accuracy comparison" that was developed to describe the way in which the maximum and minimum deviations enclosed the zone of error. These residuals were developed for spherical, cylindrical, conical, frustum, and torus forms.

5. Finally, the analysis performed on multiple feature parts was developed to compare the width of the zone of error of individual features against the width of the zone of error of the combined features. This was done with the purpose of determining if the maximum width of the zone of error of individual features governs the width of the zone of error of the combined features. Thus, the analysis allows the metrologist to recognize the feature that has the worst form deviation and to determine if the two features were created within the same setup and process sequence.

Other contributions of this research described in the first three chapters of this dissertation. These contributions are based on the

1. A feature recognition program was developed to extract the form features from CAD files of popular commercial systems such as AutoCAD®, Solidworks®, and ProEngineering®. The program arranges in a set of matrices the topological and geometrical entities of the extracted form features. The matrices are used in the consequent phases of the decision support methodology for the creation of the sampling strategies, path planning, and form verification analysis. The feature recognition program is capable of recognizing spherical, cylindrical, conical, frustum, and torus features.

2. A hexagonal mesh path plan was developed that allows the CMM to collect measuring points using approaching vectors that are normal to the surface of the part. The path plan uses the vertices and links of the hexagonal mesh to guide the CMM through the collection of the points. The advantage of the path plan is that it requires less time to inspect a part due to the fact that it calculates the shortest distance between measurements.

3. Guidelines were provided to improve the quality of the measurements that show the user how to translate the origin of the coordinate measuring machine into the origin of the inspected part. These guidelines were developed for inspecting spherical, cylindrical, conical, frustum, and torus forms.

9.2 Conclusions and Recommendations

The procedures to verify the form tolerances of spherical, cylindrical, conical, cone frustum, and torus form features were studied. The research developed a set of guidelines to accurately inspect these forms, from extracting the CAD file of the form features of the part, to the analysis of the accuracy and precision of the mathematical models for each form feature. The conclusions are summarized according to the order in which guidelines were presented in this thesis.

The methodology for setting up the origin of the coordinate measuring machine in the same place as the geometrical origin of the part was studied for each form feature. The spherical form was evaluated by setting up the origin of the machine in a different position than the geometrical origin of the part. It was found that alignment of the axes created a big impact on the performance of the models. However, the performance of the models was significantly different when the origin of both the CMM and the part was offset. Hence, it is recommended to spend more time at the beginning of inspection in aligning the coordinate systems, because this will help to ensure that the mathematical models will have better estimates of the parameters and zone of error of the form features.

The present study proposed two approaches: the dynamic angle approach (DAA) and the free form orientation (FFO) approach. These were developed to improve the performance of the existing mathematical models as well as to increase the robustness of the proposed mathematical models. The performance of these approaches was different for each form feature, and the following paragraphs summarize the conclusion for the form features.

1. Five models were used to study the spherical forms: the dynamic angle approach models with square root (SQRT), without square root (NSQRT), the model with no angles (NOANG), and Shumugam's linear (SL) and nonlinear (SNL) models. The performance of the five mathematical models was evaluated using data that belonged to a perfectly aligned part. The results showed that the models had an identical performance since the parameters and the width of the zone of error were the same. This meant that the five mathematical models provided good results in the presence of data that belongs to a perfectly aligned part. However, the same conclusion could not be drawn when the data belonged to parts that were not aligned with the coordinate system of the machine, because the performance of the mathematical models depended in great part on the geometrical principles used to create them. Out of the five mathematical models the dynamic angle approach with square root (SQRT) was the only model that included the

ideal dimensions of the spherical form within the zone of error. Also this model provided the smallest width of the zone of error and was very close to the performance of the algorithms used by the CMM. Therefore, it is recommended to use the dynamic angle approach model with square root (SQRT) when analyzing spherical form. It is important to mention that the three proposed models to inspect spherical forms required the origin of the machine to be located inside the spherical form.

2. Three models were used to study the cylindrical forms: the dynamic angle approach models with square root (SQRT), the nonlinear model with free form orientation capabilities (NL), and Shunmugam's linear model. The nonlinear model with free form orientation capabilities (NL) generated the smallest widths of the zone of error and had a better performance in the analysis of the parameters of the cylinder. The main reason that this model outperformed the others was that the cylindrical forms were not aligned with the coordinate system of the machine. This made it very difficult for the other models to counteract the axis misalignments. In addition, the free form orientation model (NL) proved to be very robust and accurate since it outperformed the results obtained by Carr et al. (1995), Wang (1992), Prakasvudhisarn (2002), and Roy et al. (1995). Therefore, the use of the free form orientation model (NL) in combination with Hammersley sampling strategy and a sample size between 32 and 128 data points to inspect cylindrical parts is recommended, since the model was found to be robust enough to be attracted to an optimal solution.

3. Three models were used to study the conical forms: the dynamic angle approach with square root (SQRT) and without square root (NSQRT), and the nonlinear model with free form orientation capabilities (NL). The latter generated the smallest

width of the zone of error and was less affected by the increments of the sample size. However, the three models had a similar performance when analyzing the magnitude of the estimated parameters of the cone. For this reason, it is recommended to use the free form orientation model (NL) as a baseline to analyze the conical forms. This was followed by the dynamic angle approach with square root (SQRT) that determined slightly smaller widths of the zone of error at higher sample sizes than the model with no square root (NSQRT).

4. Two models were used to study the frustum forms: the dynamic angle approach without square root (NSQRT) and the nonlinear model with the free form orientation capabilities (NL). Both performed similarly with respect to the width of the zone of error. However, the analysis of the estimated parameters of the frustum showed that the free form orientation model was more accurate since it included the ideal dimension of the frustum parameters within the zone of error. This is a clear example that the width of the zone of error did not provide enough information to determine the accuracy of the models. For this reason, it is important to draw conclusions about a model based on both the estimated parameters and the width of the zone of error. In conclusion, it is recommended to use the nonlinear model with free form orientation capabilities (NL) in combination with a sample size between 32 and 128 data points to inspect the frustum forms.

5. Four models were used to study torus forms: the dynamic angle approach with square root (SQRT) and without square root (NSQRT), the nonlinear model with free form orientation capabilities (NL), and Aguirre Cruz thesis model (Thesis). In the presence of data from a perfectly aligned part, the two dynamic angle approaches (SQRT

and NSQRT) and Aguirre Cruz thesis model (Thesis) outperformed the free form orientation model while determining the width of the zone of error. The Aguirre Cruz thesis model (Thesis) was the only model that generated values closer to the ideal torus parameters for all sample sizes, while the other models produced mixed results at small sample sizes and were more accurate at higher sample sizes. One disadvantage of Aguirre Cruz thesis model (Thesis) is that it is too sensitive to data points that are located at the top of the torus due to the use of constant angles. For this reason, it is recommended to adapt the dynamic angle approach principles to this model to increase its accuracy. On the other hand, in the presence of data from a tilted torus part, the nonlinear model with free form orientation capabilities (NL) outperformed the other models since it was able to accurately recognize the inclination of the torus feature. Thus, it is recommended to apply the nonlinear model with free form orientation capabilities (NL) to torus forms, but caution must be taken since it is not very robust in the presence of data that belong to a perfectly aligned part.

In general for all form features, the fact that the mathematical models or the fitting algorithms generated a small width of the zone of error does not mean that the model performed the best. It just means that the model will be able to detect form deviations on parts that have tighter tolerances. For this reason, it is recommended to analyze the width of the zone of error as well as the estimated parameters of the features in order to determine the accuracy of the mathematical models.

The nonlinear model with free form orientation capabilities (NL) performed the best in all cases due the properties which allow it to counteract the effects of the axis misalignments and the effects of the origin offsets, excluding results obtained from the first experiment of the torus form. In addition, this model proved to be accurate and robust because it looks for the normalized vector of the form feature by reorienting only the *z* component (flatness model) instead of reorienting the *x* and *y* components.

Although the dynamic angle approach model with square root (SQRT) did not have the best performance, it proved to be a simple and accurate way to improve the performance of the mathematical models because it was constructed with the purpose of eliminating the noise induced by the origin offsets.

Finally, the factor sample size affected the zone of error of all form features, since it was found that the width of the zone of error increased as the sample size increased. It is recommended to use bigger sample sizes to observe when the width of the zone of error tails off in order to determine a specific sample size needed to obtain a good estimate of the width of the zone of error.

9.3 Future Work

1. The dynamic angle and the free form orientation approaches could be applied to the study of regular and irregular prismatic forms. Aguirre Cruz and Raman (2006) developed a study to verify the form tolerances of prismatic forms by using the Fourier series analysis. The regular and irregular prism forms were transformed into periodic waves with the purpose of generating one mathematical formula that describes the whole prism form; where the periodic waves were generated based on the normal axis of the part, similar to the study of cylindrical forms. For this reason, the present study recommends applying the principles of the nonlinear model for cylindrical forms (NL) to the study of prism forms in order to improve the accuracy of the model. Figures 90 and

91 show the transformation of quadrangular (regular) and rectangular (irregular) prism forms into periodic waves.

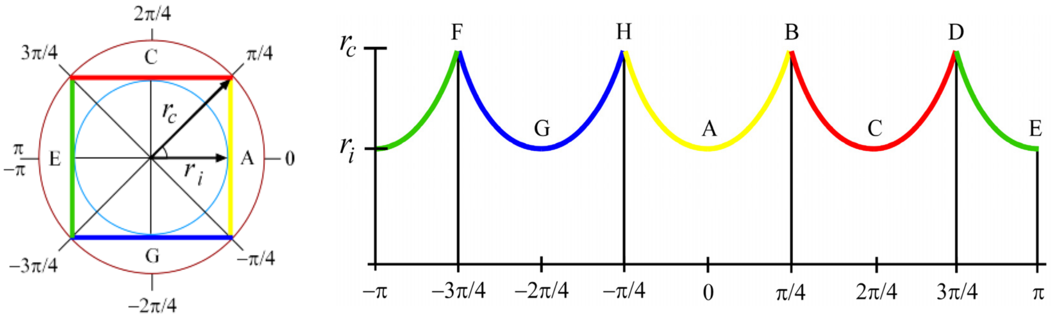


Figure 90. Quadrangular prism form verified using Fourier series analysis.

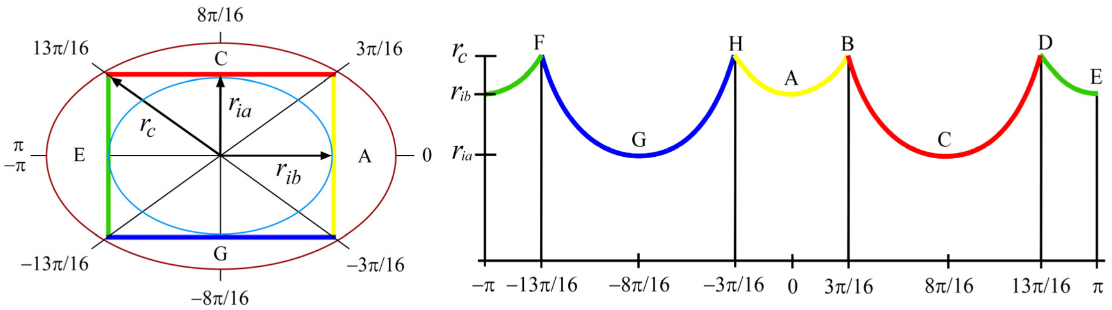


Figure 91. Rectangular prism form verified using Fourier series analysis.

2. The nonlinear model with free form orientation capabilities for cylindrical forms can be applied to the estimation of the machining parameters using coordinate metrology. This is similar to the study performed by Kilic, Aguirre Cruz, and Raman (2007) in which they developed a model to inspect cylindrical forms after turning. The model included a parameter that modeled the deflection of the part and the machining parameters, such as depth of cut and tool nose radius. In this way, the authors inspected the cylindrical form using point coordinates to estimate the parameters applied to machine the cylindrical form.

3. The performance of the mathematical models presented in this research should be studied using different fitting algorithms and different sampling strategies in order to determine if the accuracy of the models is consistent under different scenarios.

4. The errors induced by the inspection equipment should be studied and mathematically modeled in order to find the real form deviations of the inspected parts. More studies should be done to understand the capabilities of the inspection equipment. For example, the measuring table of the coordinate measuring machine cannot be perfectly aligned with the gantry that carries the stylus probe. For this reason it is important to determine the error induced by the table. To do so, optical interferometry can be used to obtain the inclination of the machine table [j, k, 1]. This information can be added to each variable that describes the normalized vector [m+i, k+n, 1] of the form features so that the fitting algorithm can obtain the real normalized vector of the part being inspected.

5. More research should be done in experimental design to develop a procedure to analyze an unbalanced design in which the levels of the factor have unequal variances. The advantage of studying this procedure is that all the residuals from the models could be analyzed to determine their variability within and between factors, instead of using the width of the zone of error that is constructed using only the maximum and the minimum deviations. The secondary advantage of analyzing all residuals is that the power of the experiment will increase and the normality assumption will be validated.

6. The guidelines and mathematical models proposed in this study could be used to develop an automated application for the analysis of the form tolerance of manufactured parts. It is foreseen that the experiments and mathematical analysis

presented here will become the basis for the development of a comprehensive decision support system for verification of geometrical tolerances.

REFERENCES

Aguirre Cruz, J.A. (2003). *Inspection of Torus Form Using Coordinate Metrology*, Published Masters thesis, University of Oklahoma, Norman, OK.

Aguirre Cruz, J.A., and Raman, S. (2005) Torus Form Inspection Using Coordinate Sampling, *ASME Journal of Manufacturing Science and Engineering*, *125*(1), 84 – 95.

Aguirre Cruz, J.A., and Raman, S. (2007). Mathematical Tolerance Verification of N-Sided Prisms Using Fourier Analysis. Online-First, To Appear in the *International Journal of Advanced Manufacturing Technology*.

Ahn, S.J., Rauh, W., Cho, H.S., and Warnecke, H.J. (2002). Orthogonal Distance Fitting of Implicit Curves and Surfaces. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, *4*(5), 620 – 638.

Ahn, S.J. (2004). *Least Square Orthogonal Distance Fitting of Curves and Surfaces in Space*. Berlin, Germany: Springer-Verlag, 17 – 77.

Alpaydin, E. (2004). *Introduction to Machine Learning*. Cambridge, Massachusetts: MIT Press.

Alter S. (1977). Why is Man-Computer Interaction Important for Decision Support Systems?, *Interfaces*, *7*(2), 109 – 115.

Alter S. (1978). Development Patterns for Decision Support Systems. *MIS Quarterly*, *2*(3), 33 – 42.

Alter S. (2002). *A Work System View of DSS in its Fourth Decade*. Eight Americas Conference on Information Systems.

Anjanappa, M., Dickstein, J.J., and Subbacharya, M. (1996). Computer-Aided Inspection Data Analyzer. *The International Journal of Advanced Manufacturing Technology*, *12*, 93 – 102.

Anthony, D.M. (1986). *Engineering Metrology*. Oxford, UK: Pergamon Press.

ASME Y14.5M-1994 (1995). *Dimensional and Tolerancing*. New York, NY: The American Society of Mechanical Engineers.

Badar, M.A., Raman, S., and Pulat, P.S. (2003). An Intelligent Search-Based Selection of Sample Point for Straightness and Flatness Estimation. *Journal of Manufacturing Science and Engineering*, *125*(2), 263 – 271.

Balakrishna P. (2005). *Coordinate Inspection of Spherical Forms Through Mathematical Optimization*. Published Masters Thesis, University of Oklahoma, Norman, OK.

Balakrishna P., and Raman, S. (2007). Support Vector Regression for Determining the Minimum Zone Sphericity. *International Journal of Advanced Manufacturing Technology*. Online-First.

Barry, A.B. (1978). *Errors in Practical Measurement in Science, Engineering, and Technology*. John Wiley & Sons: New York, 1 – 14.

Blumenthal, L.M., and Wahlin, G.E. (1941). On the Spherical Surface of Smallest Radius Enclosing a Bounded Subset of *n*-Dimensional Euclidian Space. *American Mathematical Society Bulletin*, *47*, 771 – 777.

Bohanec, M. and Zupan, B. (2004). A function-decomposition method for development of hierarchical multi-attribute decision models. *Decision Support Systems*, *36*, 215 – 253.

Bourke, P. (2007). *The torus and Supertoroid*. Published on the Internet: (http://astronomy.swin.edu.au/~pbourke/surfaces/torus/)

Carr, K., and Ferreira, P. (1995). Verification of Form Tolerances Part II: Cylindricity and Straightness of a Median Line. *Precision Engineering*, *17*, 144 – 156.

Case, K., and Gao, J. (1993). Feature technology: and overview. *International Journal of Computer Integrated Manufacturing*, 6(1-2), 2 – 12.

Case, K., and Wan Harun, W.A. (2000). Feature-Based Representation for Manufacturing Planning. *International Journal of Production Research*, *38*(17), 4285 – 4300.

Chatterjee, G., and Roth, B. (1998). Chebychev Approximation Methods for Evaluating Conicity. *Measurement*, *23*, 63 – 76.

Chen C.K., and Wu, S.C. (2000). A Mathematical Model for the Evaluation of Conically Taper Parts. *Journal of Materials Processing Technology*. *108*, 8 – 11.

Chetwynd, D.G., (1979). Roundness Measurement Using Limacons. *Precision Engineering*, *1*(3), 137-141.

Chetwynd, D. G. (1985). Applications of Linear Programming to Engineering Metrology. *Proceedings of the Institute of Mechanical Engineers*, *199*(B2), 93 – 100.

Choi, W., and Kurfess, T. R. (1999). Dimensional Measurement Data Analysis, Part 1: A zone fitting algorithm. *Journal of Manufacturing Science and Engineering*, *121*(2), 283-245.

Chung, S. H. (1997). *Methods for Determination of the Conicity Tolerance.* Published M.S. Thesis, University of Oklahoma, Norman, OK.

Ciarlini, P., Cox, M. G., Monaco, R. and Pavese, F. (1993). Advance Mathematical Tools in Metrology. *Proceedings of the International Workshop*. Singapur: World Scientific, 1-273.

Collins, C., Berrocal, E.F., Aguirre Cruz, J.A., and Raman, S. (2006). *Form Tolerance Inspection to detect Manufacturing Errors.* Published REU Program Project, University of Oklahoma, Summer 2006.

Collins, C., Berrocal, E.F., Aguirre Cruz, J.A., and Raman, S. (2007). Alternate Methods for Sampling in Coordinate Metrology. To Appear in the *Journal of Engineering Manufacture*.

Corney, J., and Lim, T. (2001). *3D Modeling with ACIS*. Stirling UK: Bell & Bain Ltd, Glasgow, 1 – 62 and 361 – 382.

Coulibaly, A., and Mutel, B. (1998). CAD/CAM Systems as the Basis of Integration Feature Based Technologies in Mechanical Engineering. *International Journal of Modelling and Simulation*, *18*(1), 26 – 35.

Cowling, G.J., and Mullineux, G. (1989). Toward a Intelligent CAD-CMM Interface. *Engineering with Computers*, *5*, 133-141.

Curless B.L. (1997). New Methods for Surface Reconstruction from Range Images. Published Ph.D. dissertation, Stanford University, Palo Alto, CA.

DEA spa. (1995a). *TUTOR for Windows: User's Guide*. Code: M1U038EB. Moncalieri, Italy: author.

DEA spa. (1995b). *TUTOR for Windows: Application Guide*. Code: M1U041EB. Moncalieri, Italy: author.

Dodge, H.F., and Romig, H.G. (1945). *Sampling Inspection Tables: Single and Double Sampling.* New York, NY: John Wiley & Sons, Inc., 1 -12.

Dotson, C., Harlow, R., and Thompson, R.L., (2003). Canada: Thompson Delmar Learning.

Dowling, M.M., Griffin, P.M., Tsui, K.L., and Zhou, C. (1995). A Comparision of the Orthogonal Least Squares and Minimum Enclosing Zone Methods for Form Error Estimation. *Manufacturing Review*, *8*(2), 120-138.

Edgeworth, R., Wilhelm. R.G. (1999). Adaptive Sampling for Coordinate Metrology. *Precision Engineering*, *23*, 144 – 154.

ElMaraghy, H.A., and Gu, P.H. (1987). Expert System for Inspection Planning. *Annals of CIRP*, *36*, 85 – 89.

ElMaraghy, H.A. (1998). *Geometric Design Tolerancing: Theories, Standards and Applications*. London, UK: Chapman & Hall.

Elzinga, D.J., and Hearn, D.W. (1972). The Minimum Covering Sphere Problem. *Management Science*, *19*(1), 96 – 104.

Erard, P.J., Belhi, A., Gardan, Y., Van der Donck, B. (1996). Feature in CIM: What are Problems?, *IEEE Proceedings, Conference on Emerging Technologies and Factory Automation, Nov. 1996. 1*, 372 – 377.

Fan, K.C., and Lee, J.C. (1999). Analysis of Minimum Zone Sphericity Error Using Minimum Potential Energy Theory. *Precision Engineering*, *23*, 65 – 72.

Fang, L., Hipel, K.W., Kilgour, M., and Peng, X. (2003a). A Decision Support System for Interactive Decision Making-Part I: Model Formulation. *IEEE Transactions on systems, man, and cybernetics-Part C: Applications and Reviews*, 33(1), 42 – 55.

Fang, L., Hipel, K.W., Kilgour, M., and Peng, X. (2003b). A Decision Support System for Interactive Decision Making-Part II: Analysis and Output Interpretation. *IEEE Transactions on systems, man, and cybernetics-Part C: Applications and Reviews*, *33*(1), 42-55.

Ferguson, R.L., Jones, C.H. (1969). A Computer Aided Decision System. *Management Science*, *15*(10), B550-B61.

Gou, J.B., Chu, Y.X. and Li, Z.X. (1999). A Geometry Theory Form, Profile, and Orientation Tolerances. *Precision Engineering*, *23*, 79-93.

Groover, M.P. (2001). *Automation, Production Systems, and Computer-Integrated Manufacturing*. Upper Saddle River, NJ: Prentice Hall.

Hammersley, J.M. (1960). Monte Carlo Methods for Solving Multivariate Problems. *Annals of the New York Academy of Science*, *86*, 844 – 874.

Henzold, G. (1995). *Handbook of geometrical tolerancing*. Chichester, UK: John Wiley & Sons.

Hilbert, D., and Cohn-Vossen, S. (1952). Geometry and the Imagination. New York, NY: Chelsea Publishing Company.

Hocken, R.J., Raja, J., Babu, U. (1993). Sampling Issues in Coordinate Metrology. *Manufacturing Review*, *6*(4), 282-304.

Huang, J. (1999). An Exact Minimum Zone Solution for Sphericity Evaluation. *Computer-Aided Design*. *31*, 845-853.

Ismail, N., Abu-Bakar, N., and Juri, A.H. (2005). Recognition of cylindrical and conical features using edge boundary classification. *International Journal of Machine Tools & Manufacture*, *45*, 649 – 655.

Jha, P. (1995). Integration of CAD to CAM Through Feature Recognition Based Solid Modeling. *Proceedings of the Computers in Engineering Conference and the Engineering Database Symposium, ASME 1995*, 489 – 505.

Jiang, B.C., and Chiu, S.D. (2002). Form Tolerance-Based Measuring Points Determination with CMM. *Journal of Intelligent Manufacturing*, *13*, 101 – 108.

Kalpakjian, S., and Schnid, S.R. (2003). *Manufacturing Processes for Engineering Materials*. Upper Saddle River, NJ: Prentice Hall.

Kanada, T., and Suzuki, S. (1993). Evaluation of minimum zone flatness by means of nonlinear optimization techniques and its verification. *Precision Engineering*, *15*(2), 93-99.

Kanada, T. (1995). Evaluation of Spherical Form Errors – Computation of Sphericity by Means of Minimum Zone Method and Some Examinations with Using Simulated Data. *Precision Engineering*, *17*(1), 281-289.

Kanada, T. (1997). Estimation of Sphericity by Means of Statistical Processing for Roundness of Spherical Parts. *Precision Engineering*, *20*(1), 117-122.

Kethara Pasupathy, T.M., Morse E.P., and Wilhelm R.G. (2003). A Survey of Mathematical Methods for the Construction of Geometric Tolerance Zones. *ASME Journal of Computing and Information Science and Engineering*, *3*(1), 64 – 75.

Kilic, B., Aguirre Cruz, J.A., and Raman, S. (2007). Inspection of the Cylindrical Surface Feature After Turning Using Coordinate Metrology. To Appear in the *International Journal of Machine Tools and Manufacture*.

Kim, W.S., and Raman, S. (2000). On the Selection of Flatness Measurement Points in Coordinate Measuring Machine Inspection. *International Journal of Machine Tools and Manufacture*, *40*, 427 – 443.

Kroszynski, U.I., Palstroem, B., Trostmann, E., and Schlechtendahl, E.G. (1989). Geometric Data Transfer Between CAD Systems: Solid Models. *IEEE Computer Graphics & Applications*, *9*(5), 57 – 71.

Lai, J.Y., and Chen, I.H. (1996). Minimum Zone Evaluation of Circles and Cylinders. *International Journal of Machine Tools and Manufacture*, *36*(4), 435 – 451.

Li, R.K, and Hwang, T.Z. (1988). A Part-Feature Recognition System for Rotational Parts. *International Journal of Production Research*, *26*, 1451 – 1475.

Li, R.K, and Hwang, T.Z. (1989). A Part-Feature Recognition System for Rotational Parts – Non-Turning Features. *International Journal of Computer Integrated Manufacturing*, *l*(5), 257 – 267.

Liu, C.H., Jywe, W.Y., and Chen, C.K. (2004). Quality Assessment on a Conical Taper Part Based on the Minimum zone Definition Using Genetic Algorithms. *International Journal of Machine Tools and Manufacture*, *44*, 183 – 190.

Lee, G., Mou, J., Shen, Y. (1997). Sampling Strategy Design for Dimensional Measurement of Geometric Features Using Coordinate Measuring Machine. *International Journal of Machine Tools Manufacture*, *37*(7), 917-934.

Maendl, M. (2003). Automatic Feature Recognition Slashes CAM work. *Machine Design*, *75*(18); 104 – 105.

Malyscheff, A.M., Trafalis, T.B., and Raman, S. (2002). From Support Vector Machine Learning to the Determination of the Minimum Enclosing Zone. *Computers and Industry Engineering*, *42*(1), 59–74.

Meystel, A.M. (2003). Multiresolutional Hierarchical Decision Support Systems. *IEEE Transactions on systems, man, and cybernetics-Part C: Applications and Reviews*, *33*(1), 86-101.

Miyata, S., Machida, H. (2001). Development of the Half Toroidal CVT POWERTOROS unit (3). *Motion & Control*, *11*, 11-18.

Montgomery, D.C. (2001). *Introduction to Statistical Quality Control*. New York, NY: John Wiley & Sons.

Montgomery, D.C., Runger, G.C. and Hubele, N.F. (2001). *Engineering Statistics.* New York, NY: John Wiley & Sons.

Moroni G, and Polini, W. (1998). Form Feature Selection in Computer-Aided Inspection Planning. *International Journal of Flexible Automation and Integrated Manufacturing*, *6*(1-2), 29 – 51.

Narayanan Namboothiri, V.N., and Shunmugam, M.S. (1995). On inspection planning for Co-ordinate Metrology. *Proceedings of SPIE. Modeling, Simulation, and Control Technologies for Manufacturing*. *2596*, 40-48.

Narayanan Namboothiri, V.N., and Shunmugam, M.S. (1998). Function-Oriented form Evaluation of Engineering Surfaces. *Precision Engineering*, *22*, 98 – 109.

Narayanan Namboothiri, V.N., and Shunmugam, M.S. (1999). On Determination of Sample Size in Form Error Evaluation Using Coordinate Metrology. *International Journal of Production Research*, *37*(4), 793-804.

NIST (2006). *IGES*. Published on the Internet. http://www.nist.gov/iges/

Ong, S.K., Li, W.D., and Nee, A.Y.C. (2003). STEP-Based Integration of Feature Recognition and Design-by-Feature for Manufacturing Applications in a Concurrent Engineering Environment. *International Journal of Computer Applications in Technology*, *18*(1-4), 78 – 92.

Orady, E., Li, S. and Chen, Y. (2000). Evaluation of Minimum Zone Straightness by a Nonlinear Optimization Method. *Journal of Manufacturing Science and Engineering*, *122*, 795-797.

Owusu-Ofori S., and Chen C.S. (1990). A knowledge-based approach to form-feature recognition from engineering drawings. *Proceedings of Manufacturing International '90, Volume I Intelligent Manufacturing Structure, Control, and Integration, presented at Atlanta, Georgia, 25–28 March*, 57–66.

Pillai, D.D. (1990). Developing a Decision Support System for Optimizing Automated Wafer Fabrication. *IEEE Transactions on components, hybrids, and manufacturing technology*, 13(1), 94-102.

Prakasvudhisarn, C. (2002). *Dimensional Measurement of Conical Features Using Coordinate Metrology*. Published Ph. D. Dissertation, University of Oklahoma, Norman, OK.

Prakasvudhisarn, C., and Raman, S. (2004). Framework for Cone Feature Measurement using Coordinate Measuring Machines. *ASME Journal of Manufacturing Science and Engineering*, *126*(1), 169-177.

Prakasvudhisarn, C., Trafalis, T. B., and Raman, S. (2003), Support Vector Regression for Determination of Minimum Zone. *ASME Journal of Manufacturing Science and Engineering*, *125*(4), 736–739.

Prabhakar, S., and Henderson, M.R. (1992). Automatic Form-Feature Recognition Using Neural-Network-Based Techniques on Boundary Representation of Solid Models. *Computer-Aided Design*, *24*(7), 381 – 393.

Rademacher, H., and Toeplitz, O. (1957). *The Enjoyment of Mathematics*. Princeton University Press, Princeton, N. J., 103 – 110.

Rajan, R., Kumar, N., Pandey, R.K., and Tiwari, M.K. (2005). Automatic Recognition of Machining Features from a Solid Model Using the 2D Feature Pattern. *International Journal of Advanced Manufacturing Technology*, *26*, 861 – 869.

Renishaw plc. (2007). *Explanation of Styli and Stylus Types*. Published on the Internet. http://www.renishaw.com/client/product/UKEnglish/PGP-147.shtml

Roth, K.F. (1954). On Irregularities of Distribution. *Mathematika*, *1*(2), 73 – 79.

Roy, U., and Xu, Y. (1995). Form and Orientation Tolerance Analysis for Cylindrical Surfaces in Computer-Aided Inspection. *Computers in Industry*, *26*(2), 127 – 134.

Russell, R.S., and Taylor III, B.W. (1998). *Operation Management: Focusing on Quality and Competitiveness*. Prentice Hall: Upper Saddle River, NJ, 712 – 714.

Samuel, G.L. and Shunmugam, M.S. (2000). Evaluation of Circularity from Coordinate and Form Data Using Computational Geometric Techniques. *Precision Engineering*, *24*, 251 – 263.

Samuel, G.L. and Shunmugam, M.S. (2001). Evaluation of Sphericity Error from Coordinate Measurement Data Using Computational Geometric Features. *Computer Methods in Applied Mechanics Engineering*, *190*, 6765 – 6781.

Samuel, G.L., and Shunmugam, M.S. (2003). Evaluation of Circularity and Sphericity from Coordinate Measuring Data. *Journal of Materials Processing Technology*, 139(1), 90 – 95.

Schwartz, A. (1967). *Calculus and Analytic Geometry*. New York, NY: Holt, Rinehart and Winston.

Shunmugam, M.S. (1987). Comparison of Linear and Normal Deviations of Forms of Engineering Surfaces. *Precision Engineering*, *9*(2), 96 – 102.

Smith, G.T. (2002). *Industrial Metrology Surfaces and Roundness*. London, UK: Springer-Verlag, 145 – 157.

Spatial Corp (2006). *ACIS*. Published on the Internet. http://www.spatial.com/

Sprague, R.H.Jr. (1980). A Framework for the Development of Decision Support Systems. *MIS Quarterly*.

SpringerLink (2006), *Encyclopaedia of Mathematics: "Circle"*. Published on the Internet. http://eom.springer.de/J/j054430.htm#j054430_00c2

STEP (2006). *STEP Tools*. Published on the Internet. http://www.steptools.com/library/standard/step_1.html

Suliman, S.M.A., and Awan, K.A. (2001). Automatic Recognition of Turning Features Using 2-D Drawing Files. *Japan Society of Mechanical Engineering International Journal, Series C*, 44(2), 527 – 533.

Sylvester, J.J. (1857). A Question in the Geometry of Situation. *Quarterly Journal of Pure and Applied Mathematics*, *1*, 79.

Tsukada T., Shukui, L., and Kanada, T. (1988). A Verification Technique of Minimum Zone Conical Taper Form. *Japan Society of Precision Engineering. 22*(3), 211 – 215.

Wang, Y. (1992). Minimum Zone Evaluation of Form Tolerances. *Manufacturing Review*, *5*(3), 213 – 220.

Weisstein, E. W. (1999). *CRC Concise Encyclopedia of Mathematics*. Boca Raton: Chapman & Hall.

Winston, W.L. (1995). *Introduction to Mathematical Programming*. Belmont, California: Dubury Press.

Woo, T.C., and Liang, R. (1993). Dimensional Measurement of Surfaces and Their Sampling. *Journal of Computer Aided Design. 25*(4), 233 – 239.

Woo, T.C., Liang, R., Hsieh, C.C., Lee, N.K. (1995). Efficient Sampling for Surface Measurements. *Journal of Manufacturing Systems. 14*(5), 345 – 354.

You, C.F., and Yang, S.S. (1998). Automatic Feature Recognition from Engineering Drawings. *The International Journal of Advanced Manufacturing Technology*, *14*, 495 – 507.

Zeid, I. (1991). *CAD/CAM Theory and Practice.* New York, NY: McGraw-Hill Inc., 153 – 177, 335 – 407, 442 – 447.

APPENDIX A

ACIS CODE OF SURFACES OF REVOLUTION

ACIS Code of a Spherical Form

```
1400 0 1 2
37 SolidWorks(2005143)-Sat-Convertor-2.0 12 ACIS 14.0 NT 24
1 9.9999999999999995e-007 1e-010
-0 body $1 -1 -1 $-1 $2 $-1 $-1 F #
-1 name_attrib-gen-attrib $-1 -1 $-1 $-1 $0 keep keep_kept ignore copy @10 Sphere #
-2 lump $3 -1 -1 $-1 $-1 $4 $0 F #
-3 rgb_color-st-attrib $-1 -1 $-1 $-1 $2 0.94117647058823528 0.67450980392156867 0.11764705882352941 #
-4 shell $-1 -1 -1 $-1 $-1 $-1 $5 $-1 $2 F #
-5 face $-1 -1 -1 $-1 $6 $7 $4 $-1 $8 forward single F F #
-6 face $-1 -1 -1 $-1 $9 $10 $4 $-1 $11 forward single F F #
-7 loop $-1 -1 -1 $-1 $-1 $12 $5 F unknown #
-8 sphere-surface $-1 -1 -1 $-1 0 0 0 50.000000000000142 1 0 0 0 0 1 forward_v I I I I #
-9 face $-1 -1 -1 $-1 $13 $14 $4 $-1 $15 reversed single F F #
-10 loop $-1 -1 -1 $-1 $-1 $16 $6 F unknown #
-11 sphere-surface $-1 -1 -1 $-1 0 0 0 50.000000000000142 1 0 0 0 0 1 forward_v I I I I #
-12 coedge $-1 -1 -1 $-1 $17 $18 $19 $20 reversed $7 $-1 #
-13 face $-1 -1 -1 $-1 $21 $22 $4 $-1 $11 forward single F F #
-14 loop $-1 -1 -1 $-1 $-1 $23 $9 F unknown #
-15 plane-surface $-1 -1 -1 $-1 0 0 0 1 0 0 0 0 -1000 forward_v I I I I #
-16 coedge $-1 -1 -1 $-1 $24 $25 $26 $27 reversed $10 $-1 #
-17 coedge $-1 -1 -1 $-1 $18 $12 $25 $28 forward $7 $-1 #
-18 coedge $-1 -1 -1 $-1 $12 $17 $29 $30 reversed $7 $-1 #
-19 coedge $-1 -1 -1 $-1 $31 $32 $12 $20 forward $33 $-1 #
-20 edge $-1 -1 -1 $-1 $34 -1.5707963267948966 $35 0 $19 $36 forward @7 unknown F #
-21 face $-1 -1 -1 $-1 $-1 $33 $4 $-1 $8 forward single F F #
-22 loop $-1 -1 -1 $-1 $-1 $26 $13 F unknown #
-23 coedge $-1 -1 -1 $-1 $37 $29 $24 $38 forward $14 $-1 #
-24 coedge $-1 -1 -1 $-1 $25 $16 $23 $38 reversed $10 $-1 #
-25 coedge $-1 -1 -1 $-1 $16 $24 $17 $28 reversed $10 $-1 #
-26 coedge $-1 -1 -1 $-1 $39 $40 $16 $27 forward $22 $-1 #
-27 edge $-1 -1 -1 $-1 $41 -3.1415926535897931 $34 -1.5707963267948966 $26 $42 forward @7 unknown F #
-28 edge $-1 -1 -1 $-1 $34 -2.7837141606445961e-015 $43 1.5707963267949001 $17 $44 forward @7 unknown T -
1.7457416805546961e-013 -1.3918570803223021e-013 0 50.000000000000142 50.000000000000142 0 #
-29 coedge $-1 -1 -1 $-1 $23 $45 $18 $30 forward $14 $-1 #
-30 edge $-1 -1 -1 $-1 $35 0 $43 1.5707963267948966 $29 $46 forward @7 unknown F #
-31 coedge $-1 -1 -1 $-1 $32 $19 $45 $47 reversed $33 $-1 #
-32 coedge $-1 -1 -1 $-1 $19 $31 $39 $48 reversed $33 $-1 #
-33 loop $-1 -1 -1 $-1 $-1 $19 $21 F unknown #
-34 vertex $-1 -1 -1 $-1 $28 $49 #
-35 vertex $-1 -1 -1 $-1 $30 $50 #
-36 ellipse-curve $-1 -1 -1 $-1 0 0 0 -1.2246063538223773e-016 -1 0 0 0 50.000000000000142 1 I I #
-37 coedge $-1 -1 -1 $-1 $45 $23 $40 $51 forward $14 $-1 #
-38 edge $-1 -1 -1 $-1 $43 1.5707963267948966 $41 3.1415926535897931 $23 $52 forward @7 unknown F #
-39 coedge $-1 -1 -1 $-1 $40 $26 $32 $48 forward $22 $-1 #
-40 coedge $-1 -1 -1 $-1 $26 $39 $37 $51 reversed $22 $-1 #
-41 vertex $-1 -1 -1 $-1 $51 $53 #
-42 ellipse-curve $-1 -1 -1 $-1 0 0 0 -1.2246063538223773e-016 -1 0 0 0 50.000000000000142 1 I I #
-43 vertex $-1 -1 -1 $-1 $30 $54 #
-44 ellipse-curve $-1 -1 -1 $-1 0 0 0 0 0 1 50.000000000000142 0 0 1 I I #
-45 coedge $-1 -1 -1 $-1 $29 $37 $31 $47 forward $14 $-1 #
-46 ellipse-curve $-1 -1 -1 $-1 0 0 0 -1 0 0 0 0 50 1 I I #
-47 edge $-1 -1 -1 $-1 $55 -1.5707963267948966 $35 0 $45 $46 forward @7 unknown F #
-48 edge $-1 -1 -1 $-1 $34 1.5707963267948992 $55 3.1415926535897967 $39 $56 forward @7 unknown T -1.7151265217091364e-
013 -50.000000000000142 -6.1232339957367834e-015 50.000000000000142 -1.3016524707046322e-013 -1.5940645318506716e-
029 #
-49 point $-1 -1 -1 $-1 50 -1.391857080322298e-013 0 #
-50 point $-1 -1 -1 $-1 0 0 50 #
-51 edge $-1 -1 -1 $-1 $41 3.1415926535897931 $55 4.7123889803846897 $37 $52 forward @7 unknown F #
-52 ellipse-curve $-1 -1 -1 $-1 0 0 0 -1 0 0 0 0 50 1 I I #
-53 point $-1 -1 -1 $-1 0 0 -50 #
-54 point $-1 -1 -1 $-1 -1.7861563276672136e-013 50 0 #
-55 vertex $-1 -1 -1 $-1 $47 $57 #
-56 ellipse-curve $-1 -1 -1 $-1 0 0 0 0 1.2246467991473532e-016 -1 0 50.000000000000142 6.1232339957367834e-015 1 I I #
-57 point $-1 -1 -1 $-1 -1.7861563276672136e-013 -50.000000000000142 -6.1232339957367661e-015 #
End-of-ACIS-data
```

ACIS Code of a Cylindrical Form

```
1400 0 1 2
37 SolidWorks(2005143)-Sat-Convertor-2.0 12 ACIS 14.0 NT 24
1 9.9999999999999995e-007 1e-010
-0 body $1 -1 -1 $-1 $2 $-1 $-1 F #
-1 name_attrib-gen-attrib $-1 -1 $-1 $-1 $0 keep keep_kept ignore copy @8 Cylinder #
-2 lump $3 -1 -1 $-1 $-1 $4 $0 F #
-3 rgb_color-st-attrib $-1 -1 $-1 $-1 $2 0.94117647058823528 0.67450980392156867 0.11764705882352941 #
-4 shell $-1 -1 -1 $-1 $-1 $-1 $5 $-1 $2 F #
-5 face $-1 -1 -1 $-1 $6 $7 $4 $-1 $8 forward single F F #
-6 face $-1 -1 -1 $-1 $9 $10 $4 $-1 $11 forward single F T -1.9999999999999964 3.4914833611093821e-015 -3.1415926535897931 0
#
-7 loop $-1 -1 -1 $-1 $-1 $12 $5 F unknown #
-8 plane-surface $-1 -1 -1 $-1 1.0474450083328164e-013 0 30.00000000000005 3.4914833611093821e-015 0 1 1000 0 -
3.4914833611093821e-012 forward_v I I I I #
-9 face $-1 -1 -1 $-1 $13 $14 $4 $-1 $15 forward single F T -1.9999999999999964 3.4914833611093821e-015 0
3.1415926535897931 #
-10 loop $-1 -1 -1 $-1 $-1 $16 $6 F unknown #
-11 cone-surface $-1 -1 -1 $-1 1.0474450083328146e-013 0 30 3.4914833611093821e-015 0 1 30 0 -1.0408340855860843e-013 1 I I
0 1 30 forward I I I I #
-12 coedge $-1 -1 -1 $-1 $17 $17 $18 $19 forward $7 $-1 #
-13 face $-1 -1 -1 $-1 $-1 $20 $4 $-1 $21 forward single F F #
-14 loop $-1 -1 -1 $-1 $-1 $22 $9 F unknown #
-15 cone-surface $-1 -1 -1 $-1 1.0474450083328146e-013 0 30 3.4914833611093821e-015 0 1 30 0 -1.0408340855860843e-013 1 I I
0 1 30 forward I I I I #
-16 coedge $-1 -1 -1 $-1 $23 $24 $25 $26 reversed $10 $-1 #
-17 coedge $-1 -1 -1 $-1 $12 $12 $24 $27 forward $7 $-1 #
-18 coedge $-1 -1 -1 $-1 $28 $25 $12 $19 reversed $14 $-1 #
-19 edge $-1 -1 -1 $-1 $29 0 $30 3.1415926535897931 $12 $31 forward @7 unknown F #
-20 loop $-1 -1 -1 $-1 $-1 $32 $13 F unknown #
-21 plane-surface $-1 -1 -1 $-1 -1.0474450083328127e-013 0 -29.999999999999947 -3.4914833611093821e-015 0 -1 -1000 0
3.4914833611093821e-012 forward_v I I I I #
-22 coedge $-1 -1 -1 $-1 $25 $28 $32 $33 forward $14 $-1 #
-23 coedge $-1 -1 -1 $-1 $34 $16 $35 $36 forward $10 $-1 #
-24 coedge $-1 -1 -1 $-1 $16 $34 $17 $27 reversed $10 $-1 #
-25 coedge $-1 -1 -1 $-1 $18 $22 $16 $26 forward $14 $-1 #
-26 edge $-1 -1 -1 $-1 $37 -0.059999999999999894 $30 1.0302869668521453e-016 $25 $38 forward @7 unknown F #
-27 edge $-1 -1 -1 $-1 $30 -3.1415926535897931 $29 0 $17 $39 forward @7 unknown F #
-28 coedge $-1 -1 -1 $-1 $22 $18 $34 $40 reversed $14 $-1 #
-29 vertex $-1 -1 -1 $-1 $40 $41 #
-30 vertex $-1 -1 -1 $-1 $26 $42 #
-31 ellipse-curve $-1 -1 -1 $-1 1.0474450083328183e-013 0 30.000000000000103 3.4914833611093821e-015 0 1 30 0 -
1.0408340855860843e-013 1 I I #
-32 coedge $-1 -1 -1 $-1 $35 $35 $22 $33 reversed $20 $-1 #
-33 edge $-1 -1 -1 $-1 $43 0 $37 3.1415926535897931 $22 $44 forward @7 unknown F #
-34 coedge $-1 -1 -1 $-1 $24 $23 $28 $40 forward $10 $-1 #
-35 coedge $-1 -1 -1 $-1 $32 $32 $23 $36 reversed $20 $-1 #
-36 edge $-1 -1 -1 $-1 $37 -3.1415926535897931 $43 0 $23 $45 forward @7 unknown F #
-37 vertex $-1 -1 -1 $-1 $26 $46 #
-38 straight-curve $-1 -1 -1 $-1 -29.999999999999893 3.6739403974420592e-015 30.000000000000103 3.4914833611093821e-012
0 1000 I I #
-39 ellipse-curve $-1 -1 -1 $-1 1.0474450083328183e-013 0 30.000000000000103 3.4914833611093821e-015 0 1 30 0 -
1.0408340855860843e-013 1 I I #
-40 edge $-1 -1 -1 $-1 $43 -0.059999999999999894 $29 1.0658141036401502e-016 $34 $47 forward @7 unknown F #
-41 point $-1 -1 -1 $-1 30.000000000000103 0 30 #
-42 point $-1 -1 -1 $-1 -29.999999999999893 3.6739403974420592e-015 30.000000000000206 #
-43 vertex $-1 -1 -1 $-1 $33 $48 #
-44 ellipse-curve $-1 -1 -1 $-1 -1.0474450083328108e-013 0 -29.999999999999893 3.4914833611093821e-015 0 1 30 0 -
1.0408340855860843e-013 1 I I #
-45 ellipse-curve $-1 -1 -1 $-1 -1.0474450083328108e-013 0 -29.999999999999893 3.4914833611093821e-015 0 1 30 0 -
1.0408340855860843e-013 1 I I #
-46 point $-1 -1 -1 $-1 -30.000000000000107 3.6739403974420726e-015 -29.99999999999979 #
-47 straight-curve $-1 -1 -1 $-1 30.000000000000103 0 29.999999999999893 3.4914833611093821e-012 0 1000 I I #
-48 point $-1 -1 -1 $-1 29.999999999999893 0 -30 #
End-of-ACIS-data
```

ACIS Code of a Conical Form

```
1400 0 1 2
37 SolidWorks(2005143)-Sat-Convertor-2.0 12 ACIS 14.0 NT 24
1 9.9999999999999995e-007 1e-010
-0 body $1 -1 -1 $-1 $2 $-1 $-1 F #
-1 name_attrib-gen-attrib $-1 -1 $-1 $-1 $0 keep keep_kept ignore copy @4 Cone #
-2 lump $3 -1 -1 $-1 $-1 $4 $0 F #
-3 rgb_color-st-attrib $-1 -1 $-1 $-1 $2 0.94117647058823528 0.67450980392156867 0.11764705882352941 #
-4 shell $-1 -1 -1 $-1 $-1 $-1 $5 $-1 $2 F #
-5 face $-1 -1 -1 $-1 $6 $7 $4 $-1 $8 forward single F T -1.6666666666666636 -1.566666666666664 0 3.1415926535897931 #
-6 face $-1 -1 -1 $-1 $9 $10 $4 $-1 $11 forward single F T -1.6666666666666636 -1.566666666666664 -3.1415926535897931 0 #
-7 loop $-1 -1 -1 $-1 $-1 $12 $5 F unknown #
-8 cone-surface $-1 -1 -1 $-1 -3.2819943594428191e-012 0 -940 -3.4914833611093821e-015 0 -1 -750.00000000000205 0
2.6645352591003757e-012 1 I I 0.60000000000000109 0.79999999999999916 750.00000000000205 forward I I I I #
-9 face $-1 -1 -1 $-1 $-1 $13 $4 $-1 $14 reversed single F F #
-10 loop $-1 -1 -1 $-1 $-1 $15 $6 F unknown #
-11 cone-surface $-1 -1 -1 $-1 -3.2819943594428191e-012 0 -940 -3.4914833611093821e-015 0 -1 -750.00000000000205 0
2.6645352591003757e-012 1 I I 0.60000000000000109 0.79999999999999916 750.00000000000205 forward I I I I #
-12 coedge $-1 -1 -1 $-1 $16 $17 $18 $19 reversed $7 $-1 #
-13 loop $-1 -1 -1 $-1 $-1 $20 $9 F unknown #
-14 plane-surface $-1 -1 -1 $-1 0 0 0 3.4914833611093821e-015 0 1 1000 0 -3.4914833611093821e-012 forward_v I I I I #
-15 coedge $-1 -1 -1 $-1 $18 $21 $16 $22 reversed $10 $-1 #
-16 coedge $-1 -1 -1 $-1 $17 $12 $15 $22 forward $7 $-1 #
-17 coedge $-1 -1 -1 $-1 $12 $16 $20 $23 reversed $7 $-1 #
-18 coedge $-1 -1 -1 $-1 $21 $15 $12 $19 forward $10 $-1 #
-19 edge $-1 -1 -1 $-1 $24 0 $25 0.074999999999999845 $18 $26 forward @7 unknown F #
-20 coedge $-1 -1 -1 $-1 $27 $27 $17 $23 forward $13 $-1 #
-21 coedge $-1 -1 -1 $-1 $15 $18 $27 $28 reversed $10 $-1 #
-22 edge $-1 -1 -1 $-1 $24 0 $29 0.074999999999999872 $16 $30 forward @7 unknown F #
-23 edge $-1 -1 -1 $-1 $25 0 $29 3.1415926535897931 $20 $31 forward @7 unknown F #
-24 vertex $-1 -1 -1 $-1 $19 $32 #
-25 vertex $-1 -1 -1 $-1 $28 $33 #
-26 straight-curve $-1 -1 -1 $-1 2.0948900166656292e-013 0 60 -600.00000000000409 0 -799.9999999999975 I I #
-27 coedge $-1 -1 -1 $-1 $20 $20 $21 $28 forward $13 $-1 #
-28 edge $-1 -1 -1 $-1 $29 -3.1415926535897931 $25 0 $27 $34 forward @7 unknown F #
-29 vertex $-1 -1 -1 $-1 $28 $35 #
-30 straight-curve $-1 -1 -1 $-1 2.0948900166656292e-013 0 60 599.99999999999829 7.3478807948841329e-014 -
800.00000000000125 I I #
-31 ellipse-curve $-1 -1 -1 $-1 0 0 0 -3.4914833611093821e-015 0 -1 -45 0 1.5711675124992219e-013 1 I I #
-32 point $-1 -1 -1 $-1 2.0948900166656292e-013 0 60 #
-33 point $-1 -1 -1 $-1 -45.000000000000007 0 3.1671131103978845e-013 #
-34 ellipse-curve $-1 -1 -1 $-1 0 0 0 -3.4914833611093821e-015 0 -1 -45 0 1.5711675124992219e-013 1 I I #
-35 point $-1 -1 -1 $-1 45.000000000000007 5.5109105961630903e-015 4.4610853639631695e-015 #
End-of-ACIS-data
```

ACIS Code of a Frustum Form

```
1400 0 1 2
37 SolidWorks(2005143)-Sat-Convertor-2.0 12 ACIS 14.0 NT 24
1 9.9999999999999995e-007 1e-010
-0 body $1 -1 -1 $-1 $2 $-1 $-1 F #
-1 name_attrib-gen-attrib $-1 -1 $-1 $-1 $0 keep keep_kept ignore copy @11 Frustum #
-2 lump $3 -1 -1 $-1 $-1 $4 $0 F #
-3 rgb_color-st-attrib $-1 -1 $-1 $-1 $2 0.94117647058823528 0.67450980392156867 0.11764705882352941 #
-4 shell $-1 -1 -1 $-1 $-1 $-1 $5 $-1 $2 F #
-5 face $-1 -1 -1 $-1 $6 $7 $4 $-1 $8 forward single F T -1.166190378969062 4.3292329318285554e-031 0 3.1415926535897931 #
-6 face $-1 -1 -1 $-1 $9 $10 $4 $-1 $11 forward single F T -1.166190378969062 4.3292329318285554e-031 -3.1415926535897931 0
#
-7 loop $-1 -1 -1 $-1 $-1 $12 $5 F unknown #
-8 cone-surface $-1 -1 -1 $-1 6.0952280304518356e-028 -1.7457416805546915e-013 0 3.4914833611093825e-015 -1 0 50
1.7457416805546915e-013 0 1 I I 0.51449575542752579 0.85749292571254465 50 forward I I I I #
-9 face $-1 -1 -1 $-1 $13 $14 $4 $-1 $15 forward single F F #
-10 loop $-1 -1 -1 $-1 $-1 $16 $6 F unknown #
-11 cone-surface $-1 -1 -1 $-1 6.0952280304518356e-028 -1.7457416805546915e-013 0 3.4914833611093825e-015 -1 0 50
1.7457416805546915e-013 0 1 I I 0.51449575542752579 0.85749292571254465 50 forward I I I I #
-12 coedge $-1 -1 -1 $-1 $17 $18 $19 $20 forward $7 $-1 #
-13 face $-1 -1 -1 $-1 $-1 $21 $4 $-1 $22 forward single F F #
-14 loop $-1 -1 -1 $-1 $-1 $23 $9 F unknown #
```

```
-15 plane-surface $-1 -1 -1 $-1 0 0 0 3.4914833611093825e-015 -1 0 1000 3.4914833611093825e-012 0 forward_v I I I I #
-16 coedge $-1 -1 -1 $-1 $24 $25 $17 $26 reversed $10 $-1 #
-17 coedge $-1 -1 -1 $-1 $27 $12 $16 $26 forward $7 $-1 #
-18 coedge $-1 -1 -1 $-1 $12 $27 $28 $29 reversed $7 $-1 #
-19 coedge $-1 -1 -1 $-1 $30 $30 $12 $20 reversed $21 $-1 #
-20 edge $-1 -1 -1 $-1 $31 0 $32 3.1415926535897931 $12 $33 forward @7 unknown F #
-21 loop $-1 -1 -1 $-1 $-1 $19 $13 F unknown #
-22 plane-surface $-1 -1 -1 $-1 -1.74574168055469e-013 49.999999999999972 0 -3.4914833611093825e-015 1 0 -1000 -
3.4914833611093825e-012 0 forward_v I I I I #
-23 coedge $-1 -1 -1 $-1 $34 $34 $27 $35 forward $14 $-1 #
-24 coedge $-1 -1 -1 $-1 $28 $16 $30 $36 forward $10 $-1 #
-25 coedge $-1 -1 -1 $-1 $16 $28 $34 $37 reversed $10 $-1 #
-26 edge $-1 -1 -1 $-1 $32 -0.058309518948453092 $38 0 $17 $39 forward @7 unknown F #
-27 coedge $-1 -1 -1 $-1 $18 $17 $23 $35 reversed $7 $-1 #
-28 coedge $-1 -1 -1 $-1 $25 $24 $18 $29 forward $10 $-1 #
-29 edge $-1 -1 -1 $-1 $31 -0.058309518948453098 $40 0 $28 $41 forward @7 unknown F #
-30 coedge $-1 -1 -1 $-1 $19 $19 $24 $36 reversed $21 $-1 #
-31 vertex $-1 -1 -1 $-1 $29 $42 #
-32 vertex $-1 -1 -1 $-1 $36 $43 #
-33 ellipse-curve $-1 -1 -1 $-1 -1.745741680554689e-013 49.999999999999936 0 3.4914833611093825e-015 -1 0 20
6.9388939039072284e-014 0 1 I I #
-34 coedge $-1 -1 -1 $-1 $23 $23 $25 $37 forward $14 $-1 #
-35 edge $-1 -1 -1 $-1 $40 0 $38 3.1415926535897931 $23 $44 forward @7 unknown F #
-36 edge $-1 -1 -1 $-1 $32 -3.1415926535897931 $31 0 $24 $45 forward @7 unknown F #
-37 edge $-1 -1 -1 $-1 $38 -3.1415926535897931 $40 0 $34 $46 forward @7 unknown F #
-38 vertex $-1 -1 -1 $-1 $35 $47 #
-39 straight-curve $-1 -1 -1 $-1 -50 -3.4914833611093831e-013 6.1232339957367661e-015 -514.49575542752279 -
857.49292571254648 6.3007558005921896e-014 I I #
-40 vertex $-1 -1 -1 $-1 $29 $48 #
-41 straight-curve $-1 -1 -1 $-1 50 0 0 514.49575542752882 -857.49292571254284 0 I I #
-42 point $-1 -1 -1 $-1 19.999999999999826 50 0 #
-43 point $-1 -1 -1 $-1 -20.000000000000174 49.999999999999865 4.2862637970157365e-015 #
-44 ellipse-curve $-1 -1 -1 $-1 0 0 0 3.4914833611093825e-015 -1 0 50 1.7457416805546915e-013 0 1 I I #
-45 ellipse-curve $-1 -1 -1 $-1 -1.745741680554689e-013 49.999999999999936 0 3.4914833611093825e-015 -1 0 20
6.9388939039072284e-014 0 1 I I #
-46 ellipse-curve $-1 -1 -1 $-1 0 0 0 3.4914833611093825e-015 -1 0 50 1.7457416805546915e-013 0 1 I I #
-47 point $-1 -1 -1 $-1 -50 -3.4914833611093831e-013 6.1232339957367661e-015 #
-48 point $-1 -1 -1 $-1 50 0 0 #
End-of-ACIS-data
```

ACIS Code of a Torus Form

```
1400 0 1 2
37 SolidWorks(2005143)-Sat-Convertor-2.0 12 ACIS 14.0 NT 24
1 9.9999999999999995e-007 1e-010
-0 body $1 -1 -1 $-1 $2 $-1 $-1 F #
-1 name_attrib-gen-attrib $-1 -1 $-1 $-1 $0 keep keep_kept ignore copy @5 Torus #
-2 lump $3 -1 -1 $-1 $-1 $4 $0 F #
-3 rgb_color-st-attrib $-1 -1 $-1 $-1 $2 0.94117647058823528 0.67450980392156867 0.11764705882352941 #
-4 shell $-1 -1 -1 $-1 $-1 $-1 $5 $-1 $2 F #
-5 face $-1 -1 -1 $-1 $6 $7 $4 $-1 $8 forward single F T -1.0474450083328143e-014 3.1415926535898002 0 3.1415926535897931 #
-6 face $-1 -1 -1 $-1 $9 $10 $4 $-1 $11 forward single F T -3.1415926535897931 5.0487097934144758e-030 0 3.1415926535897931
#
-7 loop $-1 -1 -1 $-1 $-1 $12 $5 F periphery $8 F #
-8 torus-surface $-1 -1 -1 $-1 0 0 0 -3.4914833611093825e-015 0 -1 60 20 -1 0 3.4914833611093825e-015 forward_v I I I I #
-9 face $-1 -1 -1 $-1 $13 $14 $4 $-1 $15 forward single F T -1.0658141036401503e-014 3.1415926535898002 -3.1415926535897931
0 #
-10 loop $-1 -1 -1 $-1 $-1 $16 $6 F periphery $11 F #
-11 torus-surface $-1 -1 -1 $-1 0 0 0 -3.4914833611093825e-015 0 -1 60 20 -1 0 3.4914833611093825e-015 forward_v I I I I #
-12 coedge $-1 -1 -1 $-1 $17 $18 $19 $20 reversed $7 $-1 #
-13 face $-1 -1 -1 $-1 $-1 $21 $4 $-1 $22 forward single F T -3.1415926535897931 5.0487097934144758e-030 -
3.1415926535897931 0 #
-14 loop $-1 -1 -1 $-1 $-1 $23 $9 F periphery $15 F #
-15 torus-surface $-1 -1 -1 $-1 0 0 0 -3.4914833611093825e-015 0 -1 60 20 -1 0 3.4914833611093825e-015 forward_v I I I I #
-16 coedge $-1 -1 -1 $-1 $24 $25 $17 $26 reversed $10 $-1 #
-17 coedge $-1 -1 -1 $-1 $27 $12 $16 $26 forward $7 $-1 #
-18 coedge $-1 -1 -1 $-1 $12 $27 $28 $29 reversed $7 $-1 #
-19 coedge $-1 -1 -1 $-1 $30 $31 $12 $20 forward $14 $-1 #
-20 edge $-1 -1 -1 $-1 $32 -1.5707963267949001 $33 1.5707963267948863 $19 $34 forward @7 unknown F #
-21 loop $-1 -1 -1 $-1 $-1 $35 $13 F periphery $22 F #
```

```
-22 torus-surface $-1 -1 -1 $-1 0 0 0 -3.4914833611093825e-015 0 -1 60 20 -1 0 3.4914833611093825e-015 forward_v I I I I #
-23 coedge $-1 -1 -1 $-1 $31 $30 $27 $36 reversed $14 $-1 #
-24 coedge $-1 -1 -1 $-1 $28 $16 $37 $38 reversed $10 $-1 #
-25 coedge $-1 -1 -1 $-1 $16 $28 $39 $40 forward $10 $-1 #
-26 edge $-1 -1 -1 $-1 $32 0 $41 3.1415926535897931 $17 $42 forward @7 unknown F #
-27 coedge $-1 -1 -1 $-1 $18 $17 $23 $36 forward $7 $-1 #
-28 coedge $-1 -1 -1 $-1 $25 $24 $18 $29 forward $10 $-1 #
-29 edge $-1 -1 -1 $-1 $33 0 $43 3.1415926535897931 $28 $44 forward @7 unknown F #
-30 coedge $-1 -1 -1 $-1 $23 $19 $45 $46 reversed $14 $-1 #
-31 coedge $-1 -1 -1 $-1 $19 $23 $35 $47 forward $14 $-1 #
-32 vertex $-1 -1 -1 $-1 $26 $48 #
-33 vertex $-1 -1 -1 $-1 $46 $49 #
-34 ellipse-curve $-1 -1 -1 $-1 -60 0 4.189780033331259e-013 0 -1 0 0 0 -20 1 I I #
-35 coedge $-1 -1 -1 $-1 $39 $37 $31 $47 reversed $21 $-1 #
-36 edge $-1 -1 -1 $-1 $41 -3.1415926535897896 $43 -1.2446030555722282e-062 $27 $50 forward @7 unknown F #
-37 coedge $-1 -1 -1 $-1 $35 $45 $24 $38 forward $21 $-1 #
-38 edge $-1 -1 -1 $-1 $33 1.5707963267948863 $32 4.7123889803846861 $37 $51 forward @7 unknown F #
-39 coedge $-1 -1 -1 $-1 $45 $35 $25 $40 reversed $21 $-1 #
-40 edge $-1 -1 -1 $-1 $43 -1.2446030555722282e-062 $41 3.1415926535897967 $25 $52 forward @7 unknown F #
-41 vertex $-1 -1 -1 $-1 $36 $53 #
-42 ellipse-curve $-1 -1 -1 $-1 0 0 0 -3.4914833611093825e-015 0 -1 -80 0 2.7931866888875051e-013 1 I I #
-43 vertex $-1 -1 -1 $-1 $36 $54 #
-44 ellipse-curve $-1 -1 -1 $-1 0 0 0 -3.4914833611093825e-015 0 -1 -39.999999999999993 0 1.3965933444437528e-013 1 I I #
-45 coedge $-1 -1 -1 $-1 $37 $39 $30 $46 forward $21 $-1 #
-46 edge $-1 -1 -1 $-1 $43 -3.1415926535897931 $33 0 $45 $55 forward @7 unknown F #
-47 edge $-1 -1 -1 $-1 $41 -3.1415926535897931 $32 0 $31 $56 forward @7 unknown F #
-48 point $-1 -1 -1 $-1 -80 0 4.8880767055531349e-013 #
-49 point $-1 -1 -1 $-1 -39.999999999999993 0 2.119382952648577e-013 #
-50 ellipse-curve $-1 -1 -1 $-1 60 7.3478807948841184e-015 0 -1.2246467991473532e-016 1 4.275833922458848e-031 -
20.000000000000004 -2.4492935982947065e-015 0 1 I I #
-51 ellipse-curve $-1 -1 -1 $-1 -60 0 4.189780033331259e-013 0 -1 0 0 0 -20 1 I I #
-52 ellipse-curve $-1 -1 -1 $-1 60 7.3478807948841184e-015 0 -1.2246467991473532e-016 1 4.275833922458848e-031 -
20.000000000000004 -2.4492935982947065e-015 0 1 I I #
-53 point $-1 -1 -1 $-1 80 9.7971743931788261e-015 -6.9829667222187641e-014 #
-54 point $-1 -1 -1 $-1 39.999999999999993 4.8985871965894122e-015 0 #
-55 ellipse-curve $-1 -1 -1 $-1 0 0 0 -3.4914833611093825e-015 0 -1 -39.999999999999993 0 1.3965933444437528e-013 1 I I #
-56 ellipse-curve $-1 -1 -1 $-1 0 0 0 -3.4914833611093825e-015 0 -1 -80 0 2.7931866888875051e-013 1 I I #
End-of-ACIS-data
```

APPENDIX B

INPUT AND PROGRAM FILES OF RAINDROP AND HEXAGONAL MESH PATH PLANS

Program 1 used to execute the Raindrop Approach

```
program DSS[WM1,WM2]
element_array MEMORY[512]
!The number 512 represents the maximum number of data points that will be allocated in memory
real XX,YY,ZZ,XP,YP,ZP,XM,YM,ZM,XPZ,YPZ,ZPZ
!The variables are declared for measuring, positioning, and z-plane points.
  pspeed 80
  mspeed 15
  acc 50
! Positioning speed, measuring speed, and acceleration parameters.
INCH_MODE
  XX=.0
  YY=.0
  ZZ=7
!
  ncmove
  move (X=XX,Y=YY,Z=ZZ)
! Enables NC mode and moves probe to position (0, 0, 7)

  open ("C:\Juan\T321.txt")
! Point coordinates are saved on a file named "T321.txt" that is located in the directory "Juan"
  format (pick,x=meas,y=meas,z=meas)
  file
! File 1 contains the positioning and measuring points. An example of file 1 can be seen below.
  openf (f1,"C:\Juan\PM_T321.TXT")
  reset (f1)
! File 2 contains the z-plane points. An example of the file 2 can be seen below.
  openf (f2,"C:\Juan\Z_T321.TXT")
  reset (f2)
!
! A cycle is created to obtain n number of data points
  for I=1 to 32 by 1
    readln (f2,XPZ,YPZ,ZPZ)
    dy (XPZ," ",YPZ," ",ZPZ," ")
    move (X=XPZ,Y=YPZ,Z=ZPZ)
!
    readln (f1,XP,YP,ZP,XM,YM,ZM)
    dy (XP," ",YP," ",ZP," ",XM," ",YM," ",ZM," ")
    move (X=XP,Y=YP,Z=ZP)
!
    msh (MEMORY[I],1)
    movetf (X=XM,Y=YM,Z=ZM)
!
    move (X=XPZ,Y=YPZ,Z=ZPZ)
  end_for
  move (X=0,Y=0,Z=8)
endstat
end_program
```

File 1 contains the positioning point coordinates and the measuring point coordinates of the raindrop approach for a spherical form. It is important to mention that the file cannot contain any headers.

-0.935	0.773	0.794	-0.806	0.667	0.685
-0.647	0.041	1.297	-0.558	0.035	1.118
-1.193	-0.757	0.324	-1.029	-0.653	0.280
-0.358	-0.904	1.075	-0.309	-0.780	0.927
0.328	-1.276	0.607	0.282	-1.100	0.523
0.637	-0.527	1.191	0.549	-0.454	1.027
0.396	-0.025	1.395	0.342	-0.021	1.202
0.928	0.589	0.945	0.800	0.508	0.815

File 2 contains the z-plane point coordinates of the raindrop approach for a spherical form. It is important to mention that the file cannot contain any headers.

-0.9348	0.7733	1.45
-0.6472	0.04072	1.45
-1.1933	-0.7573	1.45
-0.3581	-0.9044	1.45
0.32753	-1.2757	1.45
0.63693	-0.5269	1.45
0.39632	-0.0249	1.45
0.92835	0.58915	1.45

Program 2 used to execute the Hexagonal Mesh Approach

```
program HM[WM1,WM2]
 element_array MEMORY[512]
!The number 512 represents the maximum number of data points that will be allocated in memory
  pspeed 80
  mspeed 25
  acc 50
! Positioning speed, measuring speed, and acceleration parameters.

  inch_mode
  deg_angle
  set_dim (dm=whole, a=full, arel=acute, amode=a360)
  dy

! The NC mode of the CMM is enabled and is used to move the stylus to the location of the point
coordinates specified in the file "HM.pat"

 ncmove
 msh (MEMORY[3],25) path PT1
endstat
end_program

#include c:\wtutor\prog\HM.PAT
```

File HM.PAT contains measuring, positioning, and mesh points of the hexagonal mesh approach for a spherical form.

```
#path PT1 |
 P      1      1      -0.363       0.628       1.256
 P      1      1      -0.935       0.773       0.794
 M      1      1      -0.806       0.667       0.685
 P      1      1      -0.725       0.000       1.256
 P      1      1      -0.647       0.041       1.297
 M      1      1      -0.558       0.035       1.118
 P      1      1      -0.725      -1.256       0.000
 P      1      1      -1.193      -0.757       0.324
 M      1      1      -1.029      -0.653       0.279
 P      1      1      -0.363      -0.628       1.256
 P      1      1      -0.358      -0.904       1.075
 M      1      1      -0.309      -0.780       0.927
 P      1      1       0.328      -1.276       0.607
 M      1      1       0.282      -1.100       0.523
 P      1      1       0.363      -0.628       1.256
 P      1      1       0.637      -0.527       1.191
 M      1      1       0.549      -0.454       1.027
 P      1      1       0.725       0.000       1.256
 P      1      1       0.396      -0.025       1.395
 M      1      1       0.342      -0.021       1.202
 P      1      1       0.363       0.628       1.256
 P      1      1       0.928       0.589       0.945
 M      1      1       0.800       0.508       0.815
#endpath
```

APPENDIX C

CYLINDRICITY DATA

Table C.1. Dataset 1 of a cylindrical form obtained from Carr et al.'s (1995).

Point	*x*	*y*	*z*
1	29.1831	-52.425813	0.858363
2	-44.611479	40.126533	1.428767
3	60.004021	0.002953	4.659634
4	-38.031517	46.406943	4.962716
5	-59.414689	8.367685	6.744233
6	-39.72767	44.927386	8.152067
7	55.019556	-23.935132	11.525362
8	-34.039535	-49.395345	12.669581
9	-8.525012	59.390772	13.334342
10	34.130168	-49.375581	15.501363
11	-57.987024	15.413912	15.770917
12	53.851944	26.459793	19.191642
13	-23.228283	-55.317965	19.745199
14	-58.959058	-11.131969	19.839382
15	57.39863	17.478407	20.471642
16	57.378476	17.628739	21.998357
17	-10.250952	-59.117084	22.680775
18	-40.28493	-44.418701	22.937715
19	50.759016	-32.045645	22.751741
20	48.073499	-35.97318	27.101743

Table C.2. Dataset 2 of a cylindrical form obtained from Carr et al.'s (1995).

Point	*x*	*y*	*z*
1	-25.918505	15.109304	42.071436
2	-12.893256	-27.088078	56.081574
3	-12.893685	-27.088981	56.081574
4	-25.432673	-15.911603	55.82619
5	-25.433521	-15.912134	55.82619
6	25.363036	-16.024245	45.731882
7	25.36219	-16.03271	45.731882
8	-20.170149	-22.206503	45.384629
9	-20.171721	-22.207243	45.384629
10	-25.917641	15.108801	42.071436
11	-26.057635	14.866056	41.206363
12	-26.058504	14.866552	41.206363
13	28.699869	8.739422	40.731883
14	28.698913	8.739131	40.731883
15	25.130294	16.386834	39.235138
16	25.129457	16.386287	39.235138
17	28.324273	9.889165	31.782012
18	28.323328	9.888835	31.782012
19	17.044534	-24.688942	31.615727
20	17.043966	-24.688119	31.615727
21	-22.315616	20.050269	31.164982
22	-22.316359	20.050938	31.164982
23	1.056395	-29.982395	27.519008
24	1.05636	-29.981396	27.519008
25	29.953442	-1.688577	21.92032
26	29.952444	-1.68852	21.92032
27	-2.62016	-29.88536	19.694054
28	-2.620248	-29.886356	19.694054
29	-29.366428	-6.132937	13.137551
30	-29.367407	-6.133141	13.137551
31	30.001	0.001475	7.892267
32	30	0.001475	7.892267
33	-22.304724	20.062385	4.010534
34	-22.305468	20.063053	4.010534
35	-2.34494	29.908214	2.847871
36	-2.345018	29.909211	2.847871
37	14.612286	-26.201929	2.074327
38	14.611799	-26.201056	2.074327
39	-14.26238	-26.39288	0.461891
40	-14.262855	-26.393767	0.461891

Table C.3. Dataset 3 of a cylindrical form obtained from Carr et al.'s (1995).

Point	*x*	*y*	*z*
1	23.930322	23.878386	-41.82064
2	-11.820859	50.421254	-15.81738
3	18.527457	61.577469	-13.68042
4	48.919097	55.614254	-13.26661
5	33.207329	64.844079	-10.66548
6	49.576671	65.965076	-6.501629
7	68.59397	33.580936	-6.118165
8	65.713317	2.841028	3.498858
9	79.257377	49.418921	4.727043
10	46.632786	80.517454	4.866333
11	75.507564	20.208045	6.298139
12	-37.543275	31.718373	8.573268
13	82.820384	38.516367	9.148307
14	58.357492	87.161327	11.960644
15	96.781947	53.421231	22.908004
16	-4.153639	67.427229	23.451422
17	10.366902	80.249947	26.965969
18	84.570573	18.219363	28.224203
19	13.598993	83.519129	30.375
20	3.553158	75.111087	30.738097
21	89.036231	21.72231	35.086999
22	30.009532	-24.696147	35.870356
23	57.311572	-9.09605	38.123767
24	-26.871029	3.103967	39.48246
25	-3.608026	-24.493246	39.678687
26	81.087477	11.573666	46.319607
27	-18.366214	2.837799	46.415679
28	-3.793621	-14.263808	46.897322
29	-18.623157	23.988046	47.691608
30	58.416292	-4.557784	48.525368
31	66.363729	0.636729	49.246025
32	2.322453	-10.802862	51.268799
33	42.403448	-6.693162	56.567707
34	-5.898713	21.39033	60.097056
35	31.694971	-2.169579	63.538387
36	3.141412	52.730721	67.919265
37	67.398986	16.520701	79.062822
38	48.408528	15.833662	81.511728
39	22.371	47.845956	88.060867
40	34.46129	41.806234	94.623903

Table C.4. Dataset 1 of a cylindrical form obtained from Roy et al.'s (1995).

Point	*x*	*y*	*z*
1	-5.011	0.019	0
2	-4.701	1.715	0
3	-3.625	3.449	0
4	-2.376	4.401	0
5	0.035	4.999	0
6	2.018	4.572	0
7	3.901	3.115	0
8	4.718	1.626	0
9	4.992	0.059	0
10	4.726	-1.589	0
11	3.815	-3.22	0
12	2.356	-4.405	0
13	-0.792	-4.928	0
14	-2.014	-4.569	0
15	-3.912	-3.117	0
16	-4.802	-1.384	0
17	-5.001	0.019	5
18	-4.638	1.885	5
19	-3.984	3.019	5
20	-2.226	4.472	5
21	-0.007	4.988	5
22	2.219	4.472	5
23	3.802	3.238	5
24	4.745	1.566	5
25	4.97	0.52	5
26	4.64	-1.875	5
27	3.768	-3.288	5
28	2.321	-4.434	5
29	0.001	-5.015	5
30	-1.712	-4.705	5
31	-3.278	-3.788	5
32	-4.415	-2.365	5
33	-4.99	0	10
34	-4.644	1.823	10
35	-3.531	3.527	10
36	-1.704	4.703	10
37	0.387	4.99	10
38	1.496	4.78	10
39	3.782	3.29	10
40	4.778	1.502	10
41	4.999	0.085	10
42	4.803	-1.404	10
43	3.596	-3.484	10
44	1.913	-4.625	10
45	0.19	-4.989	10
46	-1.868	-4.641	10
47	-3.493	-3.58	10
48	-4.641	-1.834	10
49	-4.989	0.001	15
50	-4.636	1.852	15
51	-3.51	3.56	15
52	-1.688	4.706	15
53	0.413	5	15
54	1.473	4.799	15
55	3.798	3.291	15
56	4.796	1.487	15
57	5.022	0.066	15
58	4.816	-1.399	15
59	3.6	-3.472	15
60	1.923	-4.605	15
61	0.23	-4.987	15
62	-1.798	-4.643	15
63	-3.46	-3.588	15
64	-4.614	-1.878	15

APPENDIX D

MAIN EFFECT PLOTS AND SAS OUTPUT OF FORM FEATURES

ANOVA Results for Spherical Form

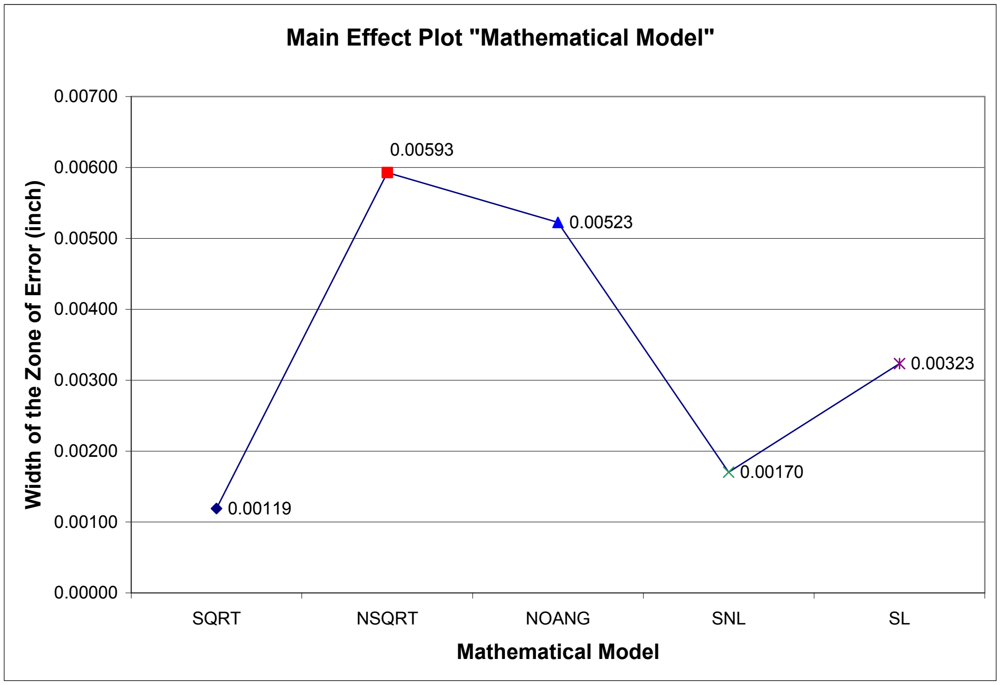


Figure D.1. Main Effect Plot of the Factor “Mathematical Model”

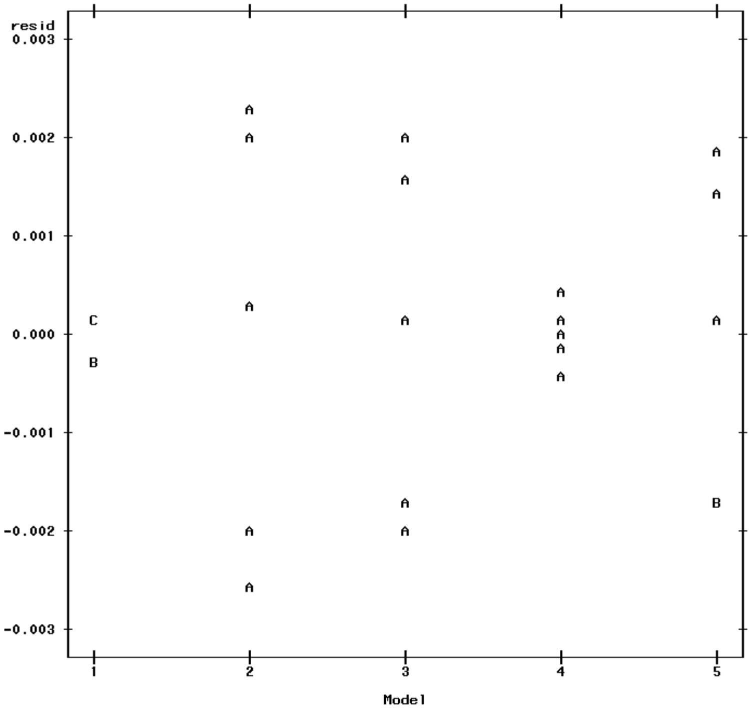
SAS Output Code for Spherical Form (Experiment 2)

Model 1 = SQRT, 2 = NSQRT, 3 = NOANG, 4 = SNL, and 5 = SL.

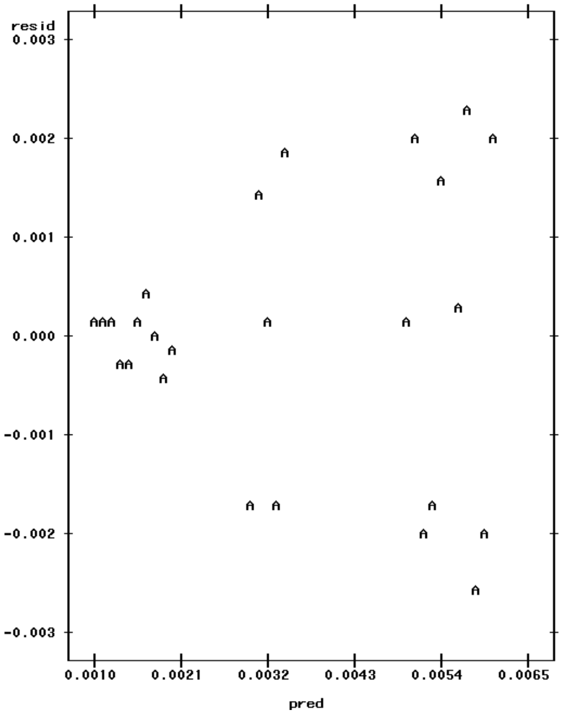
```
                                The GLM Procedure
                             Class Level Information
                         Class          Levels     Values
                         Model               5     1 2 3 4 5
                         Speci               5     1 2 3 4 5

                            Number of observations    25

                                The GLM Procedure
 Dependent Variable: Zone
                                             Sum of
      Source                      DF        Squares     Mean Square    F Value    Pr > F
      Model                        5     0.00008812      0.00001762       7.28    0.0006
      Error                       19     0.00004602      0.00000242
      Corrected Total             24     0.00013414

                      R-Square     Coeff Var      Root MSE     Zone Mean
                      0.656903      45.02470      0.001556      0.003457

      Source                      DF      Type I SS     Mean Square    F Value    Pr > F
      Model                        4     0.00008749      0.00002187       9.03    0.0003
      Speci                        1     0.00000062      0.00000062       0.26    0.6180

      Source                      DF    Type III SS     Mean Square    F Value    Pr > F
      Model                        4     0.00008749      0.00002187       9.03    0.0003
      Speci                        1     0.00000062      0.00000062       0.26    0.6180

                                The GLM Procedure
                     Tukey's Studentized Range (HSD) Test for Zone
    NOTE: This test controls the Type I experiment wise error rate, but it generally has a higher
                             Type II error rate than REGWQ.

                        Alpha                                    0.01
                        Error Degrees of Freedom                   19
                        Error Mean Square                     2.422E-6
                        Critical Value of Studentized Range   5.33342
                        Minimum Significant Difference         0.0037
                Means with the same letter are not significantly different.
```

```
Tukey Grouping          Mean       N    Model
       A          0.0059286       5    2
       A
   B   A          0.0052266       5    3
   B   A
   B   A    C     0.0032334       5    5
   B        C
   B        C     0.0017038       5    4
            C
            C     0.0011910       5    1
```

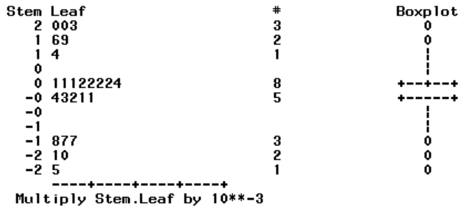
Plot of resid*Model. Legend: A = 1 obs, B = 2 obs, etc.



Plot of resid*pred. Legend: A = 1 obs, B = 2 obs, etc.

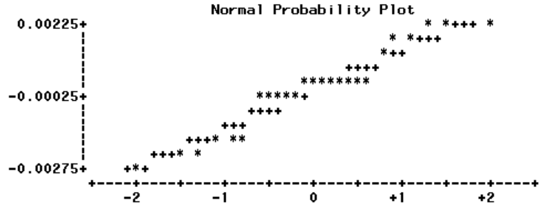


```
                      The UNIVARIATE Procedure
                          Variable:  resid
                              Moments
N                          25    Sum Weights                 25
Mean                        0    Sum Observations             0
Std Deviation      0.00138478    Variance            1.91762E-6
Skewness           -0.0958608    Kurtosis            -0.6978853
Uncorrected SS     0.00004602    Corrected SS        0.00004602
Coeff Variation             .    Std Error Mean      0.00027696

                     Basic Statistical Measures
         Location                    Variability
     Mean     0.000000     Std Deviation             0.00138
     Median   0.000114     Variance               1.91762E-6
     Mode      .           Range                     0.00482
                           Interquartile Range     0.0008442

                     Tests for Location: Mu0=0
        Test           -Statistic-    -----p Value------
        Student's t    t         0    Pr > |t|    1.0000
        Sign           M       1.5    Pr >= |M|   0.6900
        Signed Rank    S       6.5    Pr >= |S|   0.8653

                        Tests for Normality
   Test                  --Statistic---    -----p Value------
   Shapiro-Wilk          W     0.927327    Pr < W      0.0754
   Kolmogorov-Smirnov    D     0.151884    Pr > D      0.1401
   Cramer-von Mises      W-Sq  0.138295    Pr > W-Sq   0.0329
   Anderson-Darling      A-Sq  0.775191    Pr > A-Sq   0.0398

                      Quantiles (Definition 5)
                      Quantile        Estimate
                      100% Max       0.0023200
                      99%            0.0023200
                      95%            0.0020320
                      90%            0.0020062
                      75% Q3         0.0004098
                      50% Median     0.0001136
                      25% Q1        -0.0004344
                      10%           -0.0020236
                      The UNIVARIATE Procedure
                          Variable:  resid
                      Quantiles (Definition 5)
                      Quantile        Estimate
                      5%            -0.0020632
                      1%            -0.0025006
                      0% Min        -0.0025006

                        Extreme Observations
         -------Lowest------        ------Highest------
              Value      Obs             Value      Obs
         -0.0025006       12         0.0016202       23
         -0.0020632       17         0.0018724       25
         -0.0020236       13         0.0020062       22
         -0.0017822        5         0.0020320        8
         -0.0017422       18         0.0023200        7
```



```
Stem Leaf                     #             Boxplot
   2 003                      3                0
   1 69                       2                0
   1 4                        1                |
   0                                           |
   0 11122224                 8             +--+--+
  -0 43211                    5             +-----+
  -0                                           |
  -1                                           |
  -1 877                      3                0
  -2 10                       2                0
  -2 5                        1                0
     ----+----+----+----+
 Multiply Stem.Leaf by 10**-3
```

```
                      The UNIVARIATE Procedure
                          Variable:  resid
```



```
                       Normal Probability Plot
 0.00225+                                      * *+++ *
        |                                  * *+++
        |                                 *++
        |                             ++++
        |                        ********
-0.00025+                   *****+
        |                  ++++
        |               +++
        |           +++* **
        |       +++* *
-0.00275+    +*+
         +----+----+----+----+----+----+----+----+----+----+
             -2        -1         0        +1        +2
```

ANOVA Results for Cylindrical Form

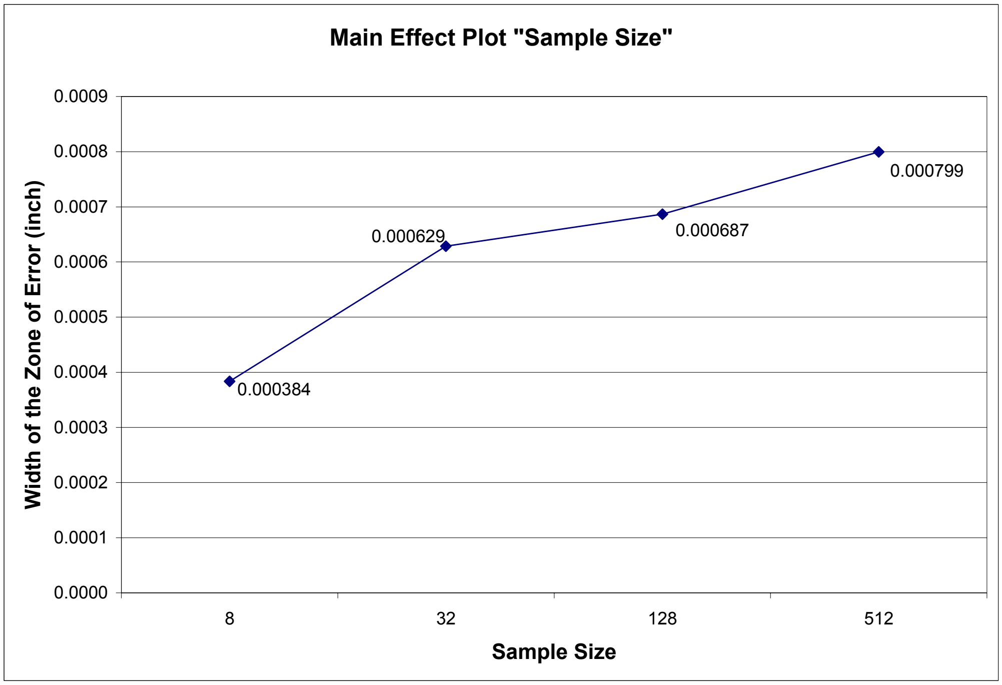


Figure D.2. Main Effect Plot of the Factor “Sample Size”

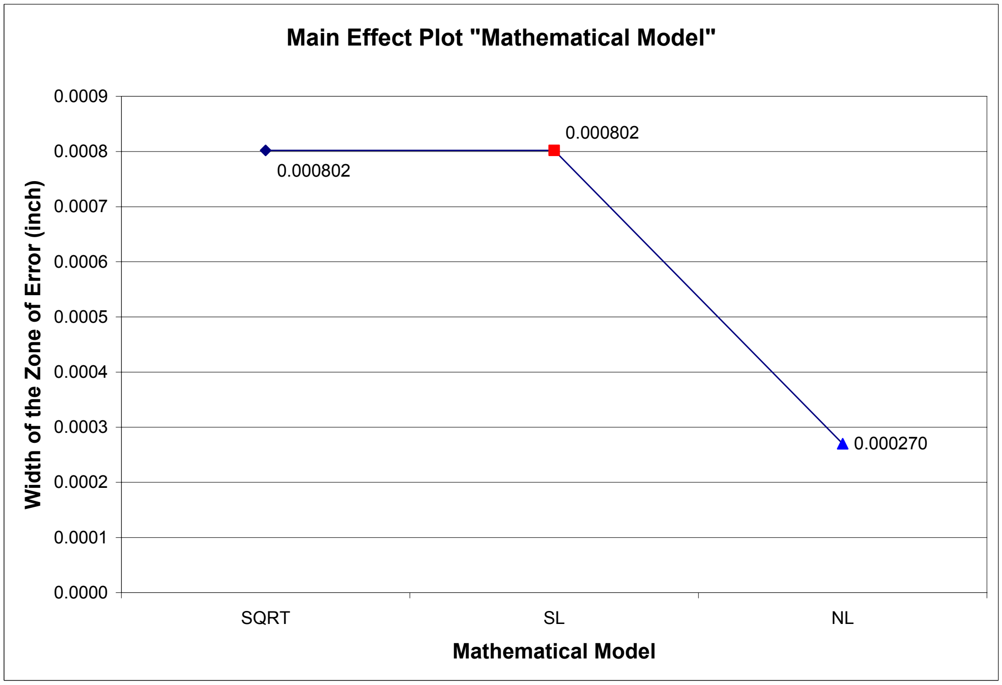
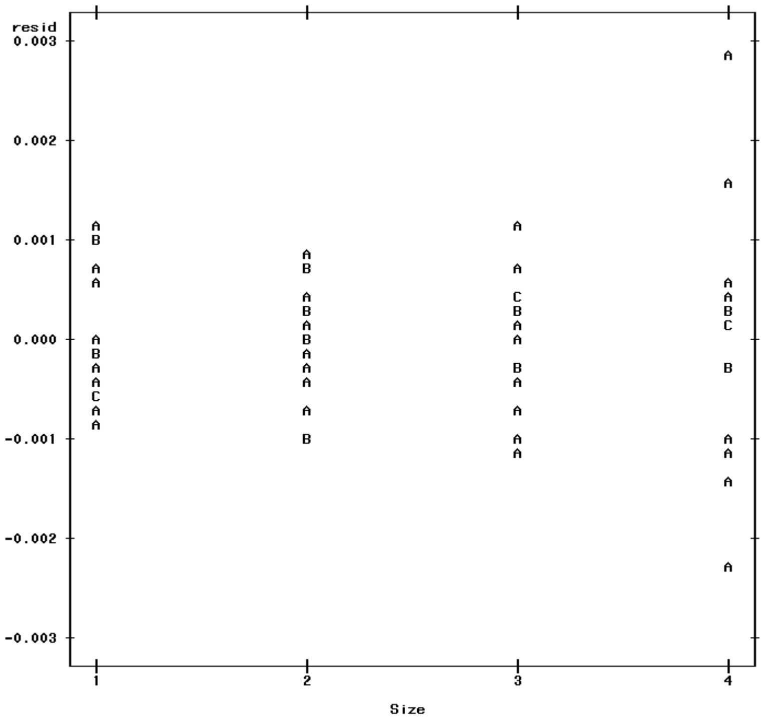


Figure D.3. Main Effect Plot of the Factor “Mathematical Model”

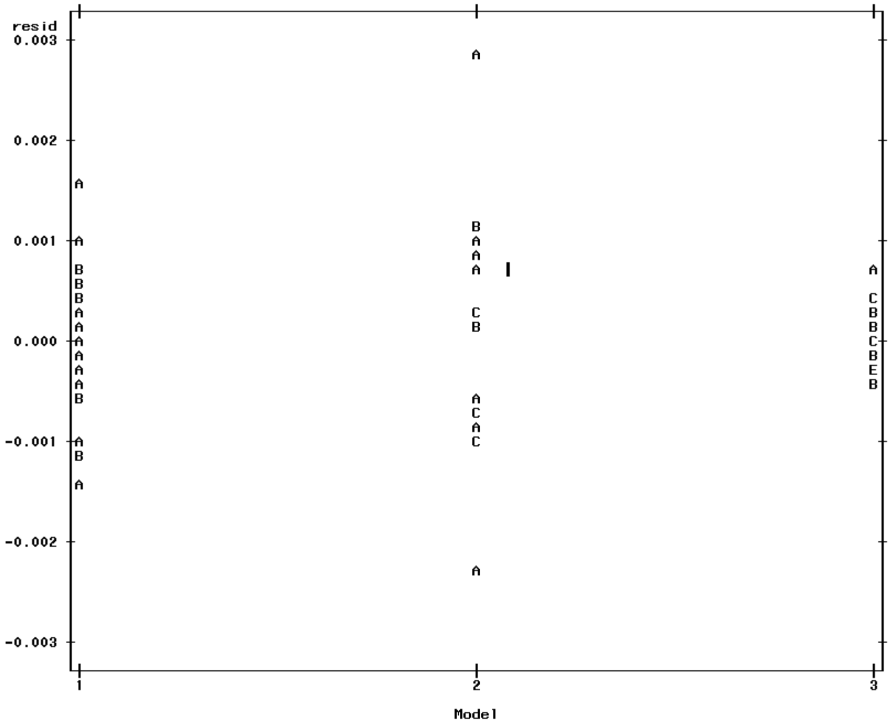
SAS Output Code for Cylindrical Form
Size 1 = 8, 2 = 32, 3 = 128, and 4 = 512.
Model 1 = SQRT, 2 = NSQRT, and 3 = NL.

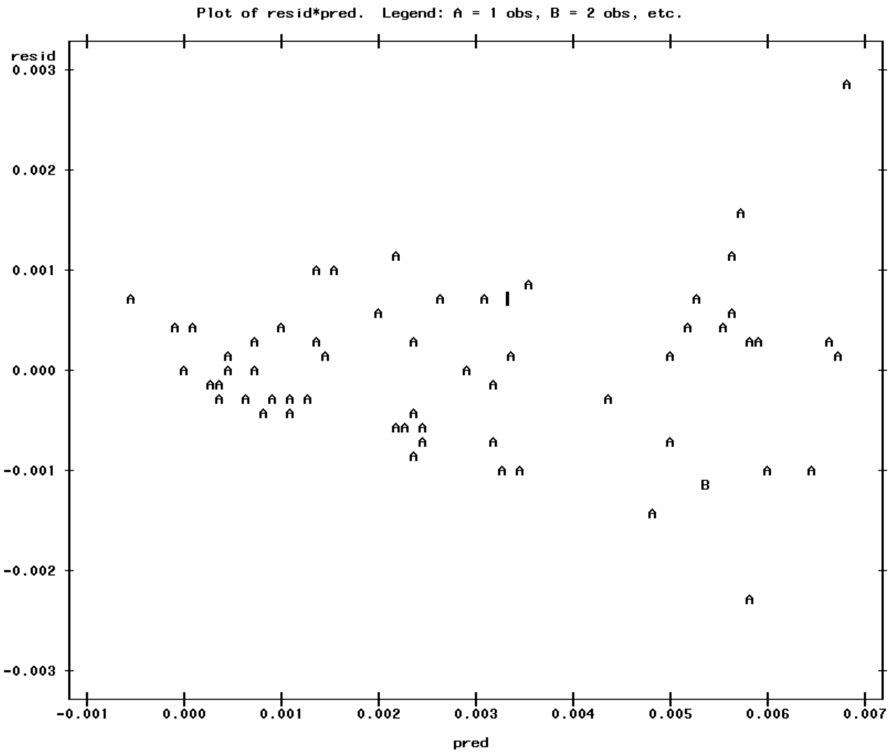
```
                                  The GLM Procedure
                               Class Level Information
                            Class         Levels    Values
                            Size               4    1 2 3 4
                            Model              3    1 2 3
                            Speci              5    1 2 3 4 5

                               Number of observations    60

                                  The GLM Procedure
 Dependent Variable: Zone
                                                Sum of
       Source                      DF          Squares     Mean Square    F Value    Pr > F
       Model                       15     5.2678626E-6    3.5119084E-7      82.09    <.0001
       Error                       44      1.882477E-7    4.2783568E-9
       Corrected Total             59     5.4561103E-6

                        R-Square     Coeff Var      Root MSE     Zone Mean
                        0.965498      10.47101      0.000065      0.000625

       Source                      DF        Type I SS     Mean Square    F Value    Pr > F
       Size                         3     1.3887034E-6    4.6290114E-7     108.20    <.0001
       Model                        2     3.7720472E-6    1.8860236E-6     440.83    <.0001
       Size*Model                   6     4.6895999E-8    7.8159999E-9       1.83    0.1158
       Speci                        4     6.0215928E-8    1.5053982E-8       3.52    0.0141

       Source                      DF      Type III SS     Mean Square    F Value    Pr > F
       Size                         3     1.3887034E-6    4.6290114E-7     108.20    <.0001
       Model                        2     3.7720472E-6    1.8860236E-6     440.83    <.0001
       Size*Model                   6     4.6895999E-8    7.8159999E-9       1.83    0.1158
       Speci                        4     6.0215928E-8    1.5053982E-8       3.52    0.0141

                                   The GLM Procedure
                      Tukey's Studentized Range (HSD) Test for Zone
 NOTE: This test controls the Type I experimentwise error rate, but it generally has a higher Type II
                                 error rate than REGWQ.

                          Alpha                                   0.01
                          Error Degrees of Freedom                  44
                          Error Mean Square                   4.278E-9
                          Critical Value of Studentized Range  4.66734
                          Minimum Significant Difference        0.0001

                   Means with the same letter are not significantly different.

                    Tukey Grouping          Mean      N    Size
                                 A    0.00079961     15    4
                                 B    0.00068665     15    3
                                 B
                                 B    0.00062882     15    2
                                 C    0.00038360     15    1

                                   The GLM Procedure
                      Tukey's Studentized Range (HSD) Test for Zone
 NOTE: This test controls the Type I experimentwise error rate, but it generally has a higher Type II
                                 error rate than REGWQ.

                          Alpha                                   0.01
                          Error Degrees of Freedom                  44
                          Error Mean Square                   4.278E-9
                          Critical Value of Studentized Range  4.34390
                          Minimum Significant Difference        0.0001

                  Means with the same letter are not significantly different.

                   Tukey Grouping          Mean      N    Model
                                A    0.00080201     20    2
                                A
                                A    0.00080192     20    1
                                B    0.00027008     20    3
```

Plot of resid*Size. Legend: A = 1 obs, B = 2 obs, etc.



Plot of resid*Model. Legend: A = 1 obs, B = 2 obs, etc.





```
                    The UNIVARIATE Procedure
                       Variable:  resid
                           Moments
N                          60    Sum Weights                 60
Mean                        0    Sum Observations             0
Std Deviation      0.00005649    Variance            3.19064E-9
Skewness             0.194629    Kurtosis            0.65943151
Uncorrected SS     1.88248E-7    Corrected SS        1.88248E-7
Coeff Variation             .    Std Error Mean      7.29228E-6

                  Basic Statistical Measures
         Location                    Variability
     Mean             0     Std Deviation           0.0000565
     Median   4.483E-6      Variance               3.19064E-9
     Mode             .     Range                   0.0002782
                            Interquartile Range     0.0000735

                  Tests for Location: Mu0=0
        Test           -Statistic-    -----p Value------
        Student's t    t          0    Pr > |t|    1.0000
        Sign           M          1    Pr >= |M|   0.8974
        Signed Rank    S          5    Pr >= |S|   0.9710

                     Tests for Normality
   Test                  --Statistic---    -----p Value------
   Shapiro-Wilk          W     0.965908    Pr < W      0.0918
   Kolmogorov-Smirnov    D     0.113304    Pr > D      0.0540
   Cramer-von Mises      W-Sq  0.083383    Pr > W-Sq   0.1907
   Anderson-Darling      A-Sq  0.642394    Pr > A-Sq   0.0918

                   Quantiles (Definition 5)
                  Quantile           Estimate
                  100% Max       1.49742E-04
                  99%            1.49742E-04
                  95%            1.01896E-04
                  90%            4.81852E-05
                  75% Q3         3.65968E-05
                  50% Median     4.48258E-06
                  25% Q1        -3.69316E-05
                  10%           -7.48697E-05
```

```
            The UNIVARIATE Procedure
                Variable:  resid
            Quantiles (Definition 5)
           Quantile           Estimate
           5%               -8.68600E-05
           1%               -1.28435E-04
           0% Min           -1.28435E-04

              Extreme Observations
--------Lowest-------          -------Highest-------
       Value      Obs                 Value      Obs
-1.28435E-04       46           5.82577E-05        6
-1.28061E-04       47           6.85587E-05       33
-8.68904E-05       49           1.35234E-04       56
-8.68296E-05       50           1.35449E-04       55
-7.64727E-05       48           1.49742E-04       54
```

```
          The UNIVARIATE Procedure
              Variable:  resid

Stem Leaf                   #            Boxplot
  28 3                      1               0
  26
  24
  22
  20
  18
  16
  14 3                      1               |
  12 01                     2               |
  10 2                      1               |
   8 67                     2               |
   6 45167                  5               |
   4 025903                 6            +-----+
   2 1346124                7            |     |
   0 157445                 6            *--+--*
  -0 9971                   4            |     |
  -2 1998750                7            |     |
  -4 7930                   4            +-----+
  -6 77743                  5               |
  -8 863                    3               |
 -10 4930                   4               |
 -12                                        |
 -14 0                      1               |
 -16
 -18
 -20
 -22 6                      1               0
     ----+----+----+----+
 Multiply Stem.Leaf by 10**-4
```

```
             The UNIVARIATE Procedure
                 Variable:  resid
                      Normal Probability Plot
 0.0029+                                                *
       |
       |
       |
       |                                                  +
       |                                                ++
       |                                             +++
       |                                           +*
       |                                        +**
       |                                      +*
       |                                   ++**
       |                                 +***
       |                              +***
 0.0003+                           ****
       |                         ***
       |                       **
       |                    ****
       |                  ***
       |               ***
       |             **+
       |        ** **
       |        +++
       |      *+
       |   +++
       | ++
       |+
-0.0023+  *
        +----+----+----+----+----+----+----+----+----+----+
            -2        -1         0        +1        +2
```

ANOVA Results for Conical Form

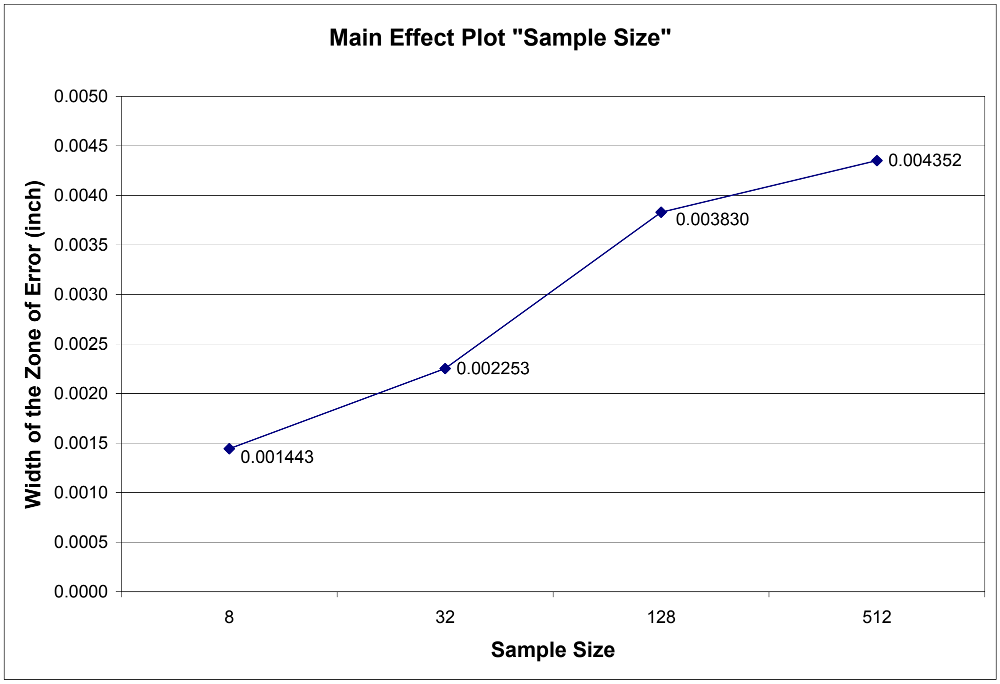


Figure D.4. Main Effect Plot of the Factor "Sample Size"

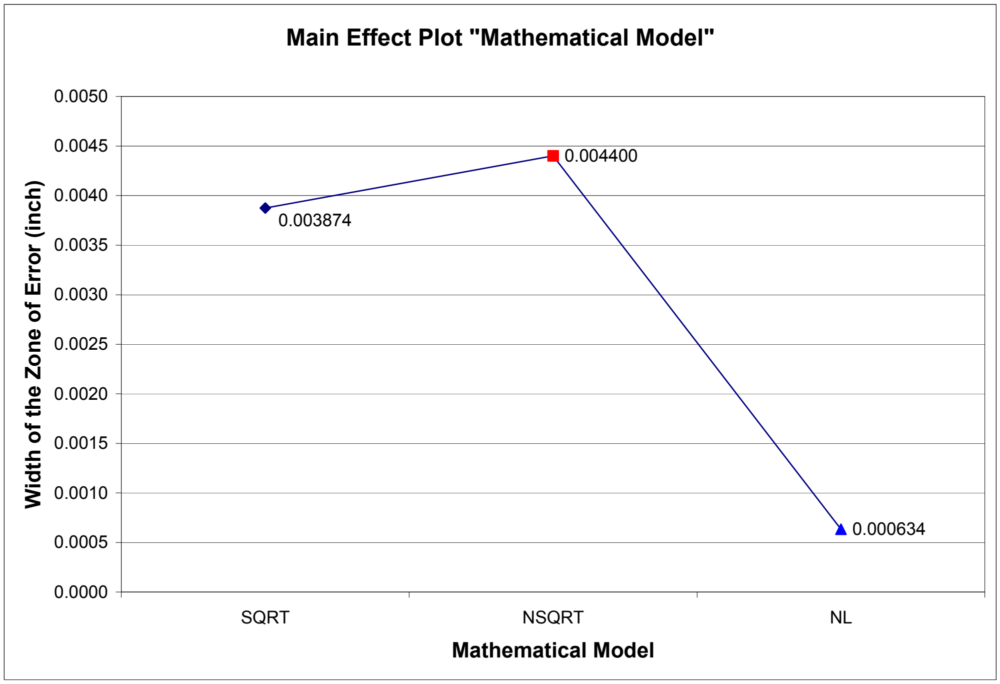


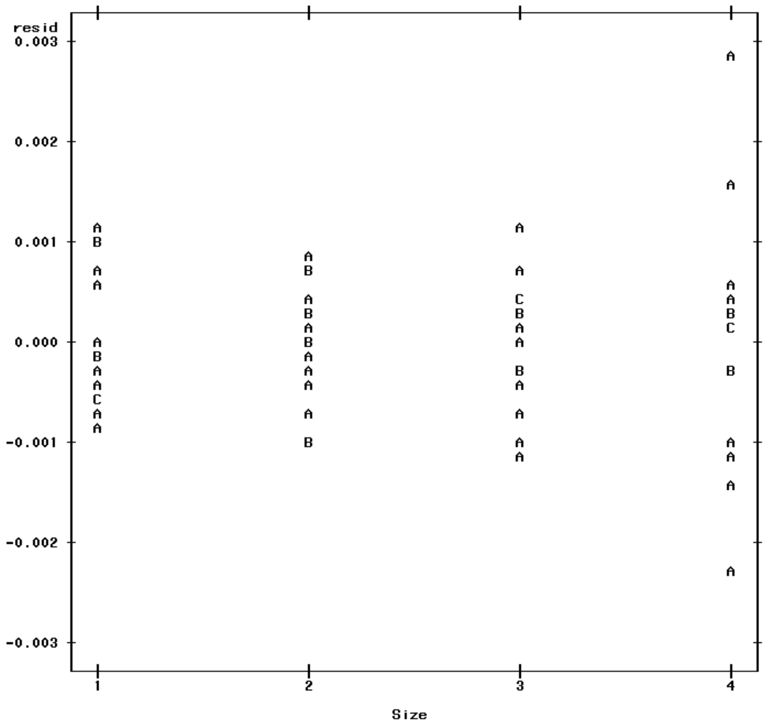
Figure D.5. Main Effect Plot of the Factor "Mathematical Model"

SAS Output Code for Conical Form

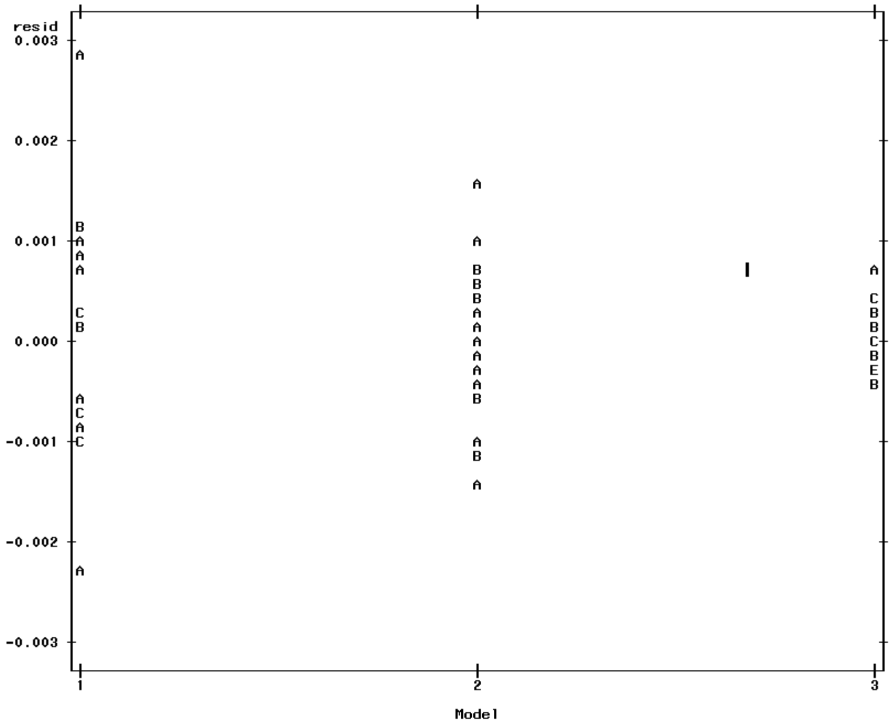
Size 1 = 8, 2 = 32, 3 = 128, and 4 = 512.
Model 1 = SQRT, 2 = NSQRT, and 3 = NL.

```
                                         The GLM Procedure
                                      Class Level Information
                                 Class         Levels    Values
                                 Size               4    1 2 3 4
                                 Model              3    1 2 3
                                 Speci              5    1 2 3 4 5

                                    Number of observations    60

                                         The GLM Procedure
Dependent Variable: Zone
                                                Sum of
      Source                      DF          Squares     Mean Square    F Value    Pr > F
      Model                       15       0.00027664      0.00001844      20.97    <.0001
      Error                       44       0.00003870      0.00000088
      Corrected Total             59       0.00031534

                         R-Square     Coeff Var      Root MSE     Zone Mean
                         0.877290      31.58039      0.000938      0.002970

      Source                      DF        Type I SS     Mean Square    F Value    Pr > F
      Size                         3       0.00008242      0.00002747      31.24    <.0001
      Model                        2       0.00016646      0.00008323      94.64    <.0001
      Size*Model                   6       0.00002070      0.00000345       3.92    0.0032
      Speci                        4       0.00000706      0.00000176       2.01    0.1102

      Source                      DF      Type III SS     Mean Square    F Value    Pr > F
      Size                         3       0.00008242      0.00002747      31.24    <.0001
      Model                        2       0.00016646      0.00008323      94.64    <.0001
      Size*Model                   6       0.00002070      0.00000345       3.92    0.0032
      Speci                        4       0.00000706      0.00000176       2.01    0.1102

                                          The GLM Procedure
                            Tukey's Studentized Range (HSD) Test for Zone
NOTE: This test controls the Type I experimentwise error rate, but it generally has a higher Type II
                                       error rate than REGWQ.

                            Alpha                                     0.01
                            Error Degrees of Freedom                    44
                            Error Mean Square                     8.794E-7
                            Critical Value of Studentized Range  4.66734
                            Minimum Significant Difference        0.0011

                     Means with the same letter are not significantly different.

                      Tukey Grouping          Mean      N    Size
                                   A     0.0043522     15    4
                                   A
                                   A     0.0038297     15    3
                                   B     0.0022529     15    2
                                   B
                                   B     0.0014432     15    1

                                          The GLM Procedure
                            Tukey's Studentized Range (HSD) Test for Zone
NOTE: This test controls the Type I experimentwise error rate, but it generally has a higher Type II
                                       error rate than REGWQ.

                            Alpha                                     0.01
                            Error Degrees of Freedom                    44
                            Error Mean Square                     8.794E-7
                            Critical Value of Studentized Range  4.34390
                            Minimum Significant Difference        0.0009

                    Means with the same letter are not significantly different.

                     Tukey Grouping          Mean      N    Model
                                  A     0.0044004     20    1
                                  A
                                  A     0.0038746     20    2
                                  B     0.0006336     20    3
```

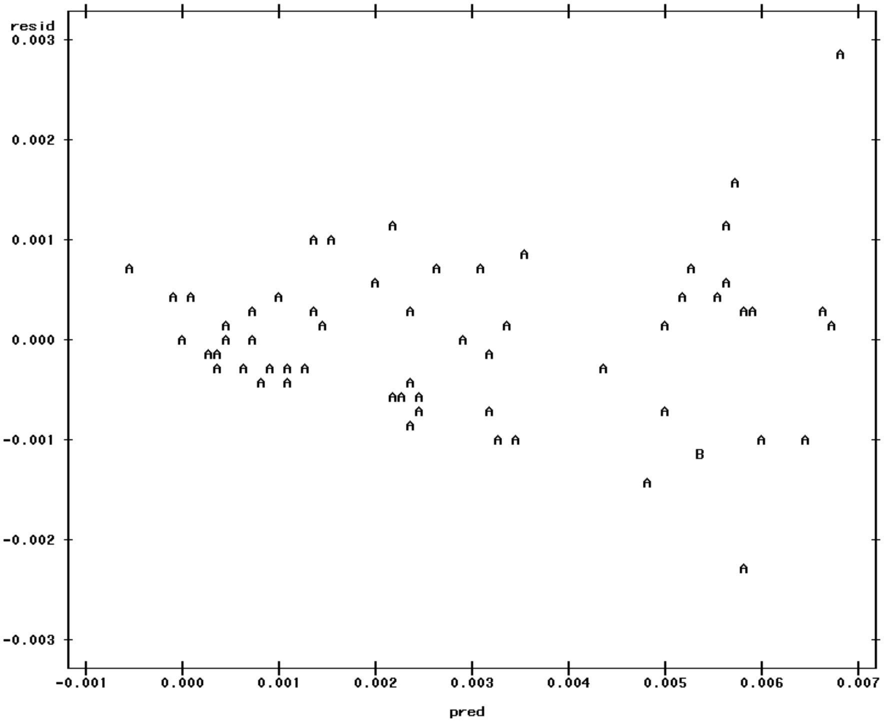
Plot of resid*Size. Legend: A = 1 obs, B = 2 obs, etc.



Plot of resid*Model. Legend: A = 1 obs, B = 2 obs, etc.



Plot of resid*pred. Legend: A = 1 obs, B = 2 obs, etc.

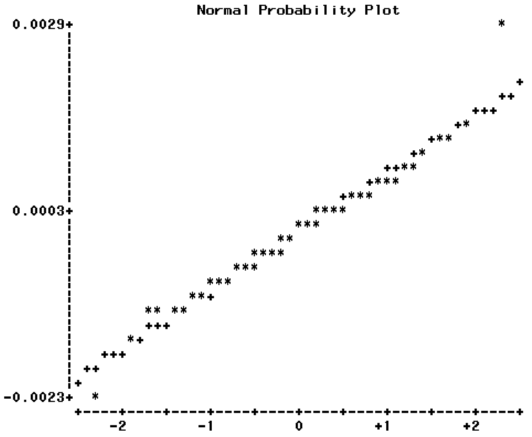


```
                         The UNIVARIATE Procedure
                            Variable:  resid
                                Moments
N                          60    Sum Weights                 60
Mean                        0    Sum Observations             0
Std Deviation      0.00080985    Variance            6.55854E-7
Skewness           0.36315884    Kurtosis            1.97047879
Uncorrected SS      0.0000387    Corrected SS         0.0000387
Coeff Variation             .    Std Error Mean      0.00010455

                        Basic Statistical Measures
         Location                    Variability
     Mean            0     Std Deviation           0.0008098
     Median   0.000029     Variance               6.55854E-7
     Mode            .     Range                     0.00510
                           Interquartile Range     0.0009998

                        Tests for Location: Mu0=0
            Test           -Statistic-    -----p Value------
            Student's t    t         0    Pr > |t|    1.0000
            Sign           M         1    Pr >= |M|   0.8974
            Signed Rank    S       -12    Pr >= |S|   0.9305

                           Tests for Normality
       Test                  --Statistic---    -----p Value------
       Shapiro-Wilk          W     0.973694    Pr < W      0.2204
       Kolmogorov-Smirnov    D     0.056367    Pr > D     >0.1500
       Cramer-von Mises      W-Sq  0.031779    Pr > W-Sq  >0.2500
       Anderson-Darling      A-Sq  0.292659    Pr > A-Sq  >0.2500

                         Quantiles (Definition 5)
                        Quantile          Estimate
                        100% Max       2.83447E-03
                        99%            2.83447E-03
                        95%            1.20827E-03
                        90%            9.13050E-04
                        75% Q3         4.66967E-04
                        50% Median     2.92667E-05
                        25% Q1        -5.32858E-04
                        10%           -9.91708E-04
```

```
          The UNIVARIATE Procedure
              Variable:  resid
          Quantiles (Definition 5)
         Quantile           Estimate
         5%             -1.11448E-03
         1%             -2.26137E-03
         0% Min         -2.26137E-03

            Extreme Observations
-------Lowest-------        -------Highest------
      Value      Obs            Value      Obs
-0.00226137       52       0.00102283        7
-0.00140297       53       0.00120337       40
-0.00114363       56       0.00121317       10
-0.00108533       32       0.00153087       47
-0.00102713       17       0.00283447       46

          The UNIVARIATE Procedure
              Variable:  resid
```

```
   Stem Leaf                     #           Boxplot
     28 3                        1              0
     26
     24
     22
     20
     18
     16
     14 3                        1              |
     12 01                       2              |
     10 2                        1              |
      8 67                       2              |
      6 45167                    5              |
      4 025903                   6           +-----+
      2 1346124                  7           |     |
      0 157445                   6           *--+--*
     -0 9971                     4           |     |
     -2 1998750                  7           |     |
     -4 7930                     4           +-----+
     -6 77743                    5              |
     -8 863                      3              |
    -10 4930                     4              |
    -12                                         |
    -14 0                        1              |
    -16
    -18
    -20
    -22 6                        1              0
        ----+----+----+----+
    Multiply Stem.Leaf by 10**-4
```

```
          The UNIVARIATE Procedure
              Variable:  resid
```



ANOVA Results for Frustum Form

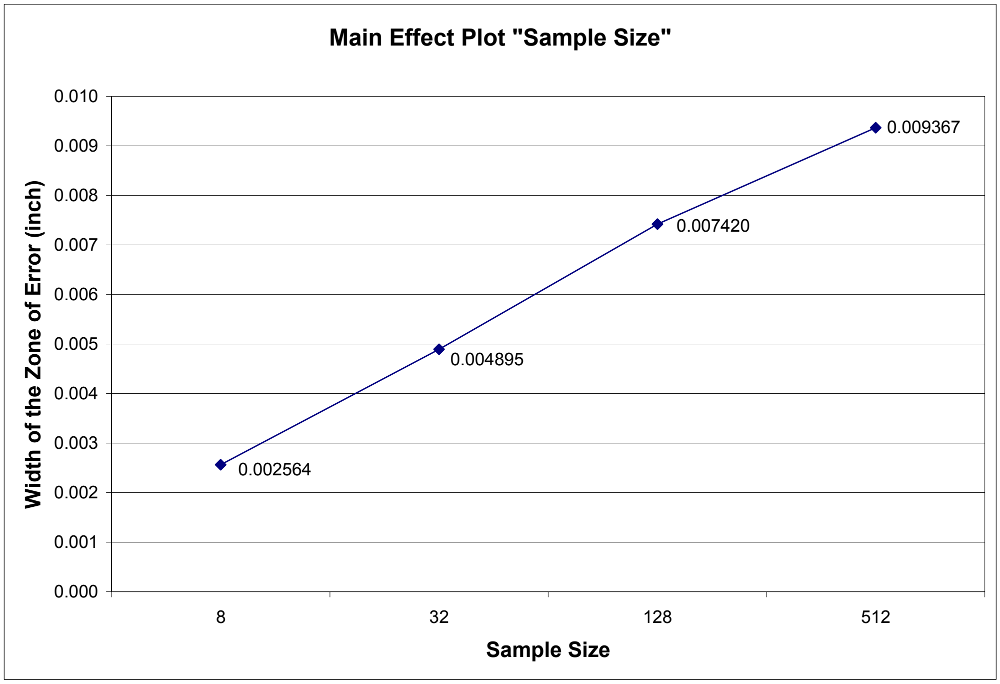


Figure D.6. Main Effect Plot of the Factor “Sample Size”

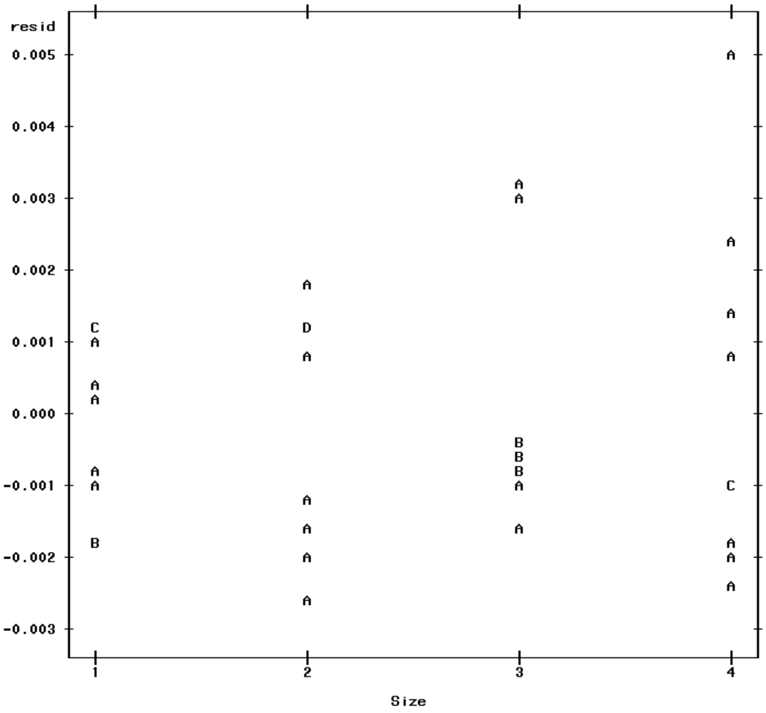
SAS Output Code for Frustum Form

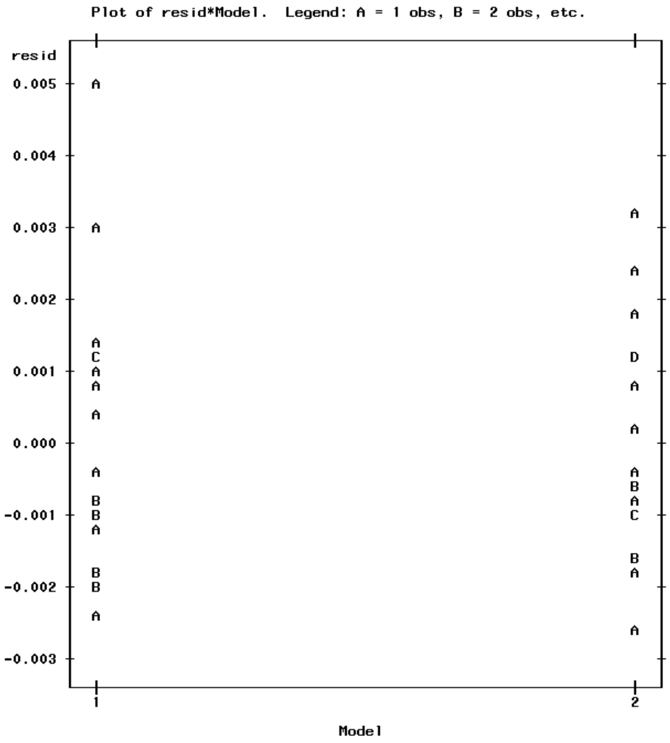
Size 1 = 8, 2 = 32, 3 = 128, and 4 = 512.
Model 1 = NSQRT, and 2 = NL.

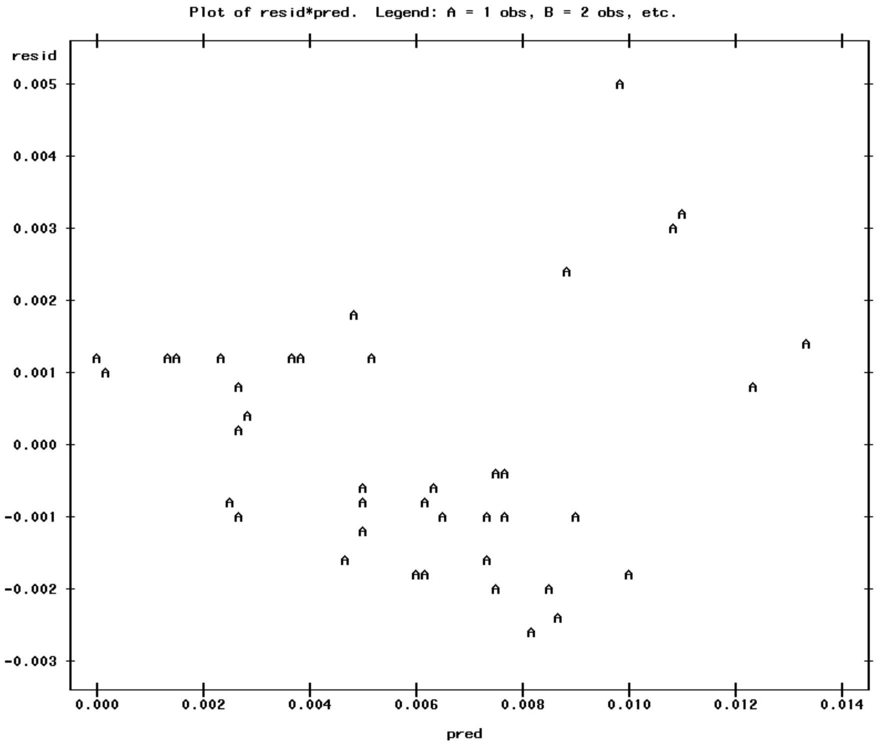
```
                                The GLM Procedure
                             Class Level Information
                         Class          Levels     Values
                         Size                4     1 2 3 4
                         Model               2     1 2
                         Speci               5     1 2 3 4 5

                            Number of observations    40

                                The GLM Procedure
 Dependent Variable: Zone
                                       Sum of
       Source                    DF         Squares    Mean Square    F Value    Pr > F
       Model                     11      0.00042145     0.00003831       9.53    <.0001
       Error                     28      0.00011260     0.00000402
       Corrected Total           39      0.00053405

                     R-Square     Coeff Var      Root MSE     Zone Mean
                     0.789155      33.08334      0.002005      0.006062

       Source                    DF       Type I SS    Mean Square    F Value    Pr > F
       Size                       3      0.00026365     0.00008788      21.85    <.0001
       Model                      1      0.00000116     0.00000116       0.29    0.5948
       Size*Model                 3      0.00000158     0.00000053       0.13    0.9409
       Speci                      4      0.00015506     0.00003876       9.64    <.0001

       Source                    DF     Type III SS    Mean Square    F Value    Pr > F
       Size                       3      0.00026365     0.00008788      21.85    <.0001
       Model                      1      0.00000116     0.00000116       0.29    0.5948
       Size*Model                 3      0.00000158     0.00000053       0.13    0.9409
       Speci                      4      0.00015506     0.00003876       9.64    <.0001
```

```
                              The GLM Procedure
                 Tukey's Studentized Range (HSD) Test for Zone
NOTE: This test controls the Type I experimentwise error rate, but it generally has a higher Type II
                            error rate than REGWQ.

                 Alpha                                    0.01
                 Error Degrees of Freedom                   28
                 Error Mean Square                    4.022E-6
                 Critical Value of Studentized Range   4.82959
                 Minimum Significant Difference         0.0031

          Means with the same letter are not significantly different.

              Tukey Grouping          Mean      N    Size
                        A      0.0093670     10    4
                        A
                   B    A      0.0074200     10    3
                   B
                   B    C      0.0048955     10    2
                        C
                        C      0.0025638     10    1

                              The GLM Procedure
                 Tukey's Studentized Range (HSD) Test for Zone
NOTE: This test controls the Type I experimentwise error rate, but it generally has a higher Type II
                            error rate than REGWQ.

                 Alpha                                    0.01
                 Error Degrees of Freedom                   28
                 Error Mean Square                    4.022E-6
                 Critical Value of Studentized Range   3.90784
                 Minimum Significant Difference         0.0018

          Means with the same letter are not significantly different.

           Tukey Grouping          Mean      N    Model
                     A      0.0062322     20    1
                     A
                     A      0.0058910     20    2
```

Plot of resid*Size. Legend: A = 1 obs, B = 2 obs, etc.

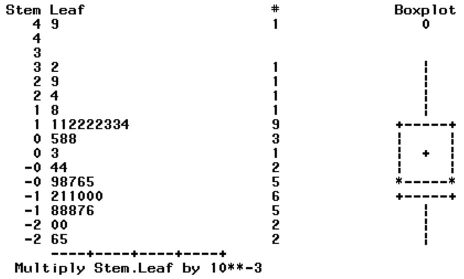






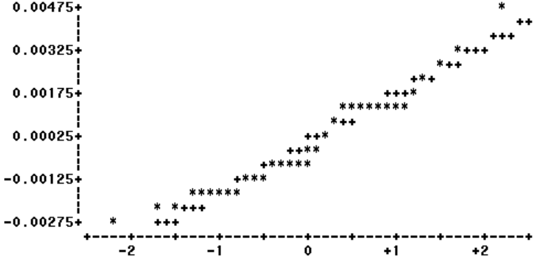
```
                     The UNIVARIATE Procedure
                        Variable:  resid
                            Moments
N                          40    Sum Weights                 40
Mean                        0    Sum Observations             0
Std Deviation      0.00169919    Variance            2.88725E-6
Skewness           0.69917819    Kurtosis             0.3363695
Uncorrected SS      0.0001126    Corrected SS         0.0001126
Coeff Variation             .    Std Error Mean      0.00026867
```

```
                 Basic Statistical Measures
    Location                    Variability
Mean      0.00000     Std Deviation            0.00170
Median   -0.00047     Variance              2.88725E-6
Mode       .          Range                    0.00757
                      Interquartile Range      0.00237

                 Tests for Location: Mu0=0
     Test           -Statistic-    -----p Value------
     Student's t    t         0    Pr > |t|    1.0000
     Sign           M        -2    Pr >= |M|   0.6358
     Signed Rank    S       -12    Pr >= |S|   0.8742

                    Tests for Normality
Test                  --Statistic---    -----p Value------
Shapiro-Wilk          W     0.944106    Pr < W      0.0476
Kolmogorov-Smirnov    D     0.139851    Pr > D      0.0471
Cramer-von Mises      W-Sq  0.133726    Pr > W-Sq   0.0393
Anderson-Darling      A-Sq  0.762561    Pr > A-Sq   0.0446

                  Quantiles (Definition 5)
                 Quantile          Estimate
                 100% Max       0.004924012
                 99%            0.004924012
                 95%            0.003038422
                 90%            0.002080749
                 75% Q3         0.001214835
                 50% Median    -0.000467704
                 25% Q1        -0.001153955
                 10%           -0.001896101

                  The UNIVARIATE Procedure
                      Variable:  resid
                  Quantiles (Definition 5)
                 Quantile          Estimate
                 5%            -0.002265400
                 1%            -0.002642185
                 0% Min        -0.002642185

                    Extreme Observations
     -------Lowest-------        -------Highest------
           Value      Obs              Value      Obs
     -0.00264219       12         0.00177321       14
     -0.00249574       37         0.00238829       36
     -0.00203506       11         0.00292035       21
     -0.00196031       39         0.00315649       22
     -0.00183190       33         0.00492401       35
```



```
Stem Leaf                     #             Boxplot
   4 9                        1                0
   4
   3
   3 2                        1                |
   2 9                        1                |
   2 4                        1                |
   1 8                        1                |
   1 112222334                9             +-----+
   0 588                      3             |     |
   0 3                        1             |  +  |
  -0 44                       2             |     |
  -0 98765                    5             *-----*
  -1 211000                   6             +-----+
  -1 88876                    5                |
  -2 00                       2                |
  -2 65                       2                |
     ----+----+----+----+
 Multiply Stem.Leaf by 10**-3
```

```
                  The UNIVARIATE Procedure
                      Variable:  resid
```



ANOVA Results for Torus Form (Experiment 1)

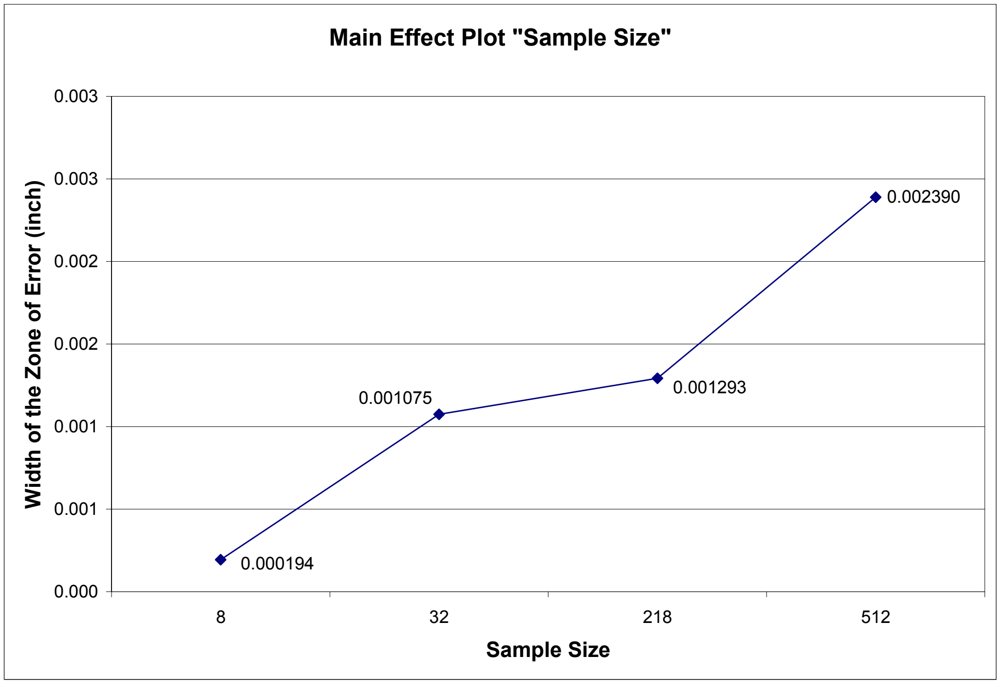


Figure D.7. Main Effect Plot of the Factor “Sample Size”

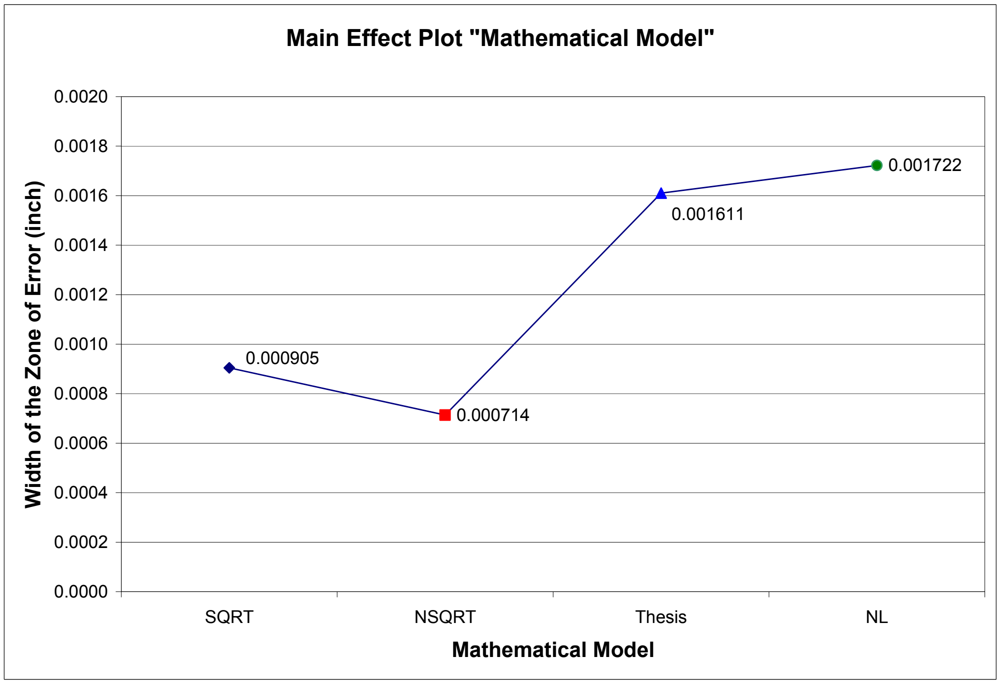


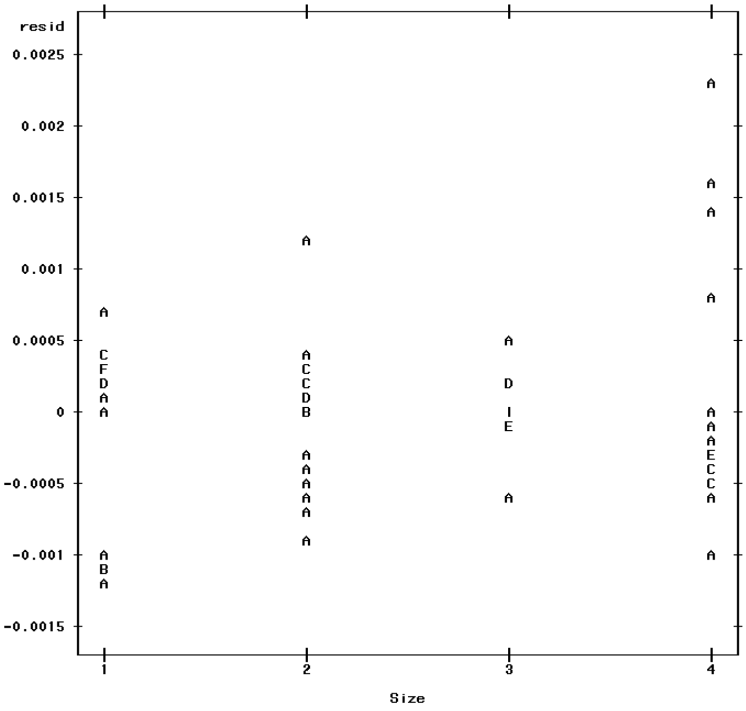
Figure D.8. Main Effect Plot of the Factor “Mathematical Model”

SAS Output Code for Torus Form (Experiment 1)

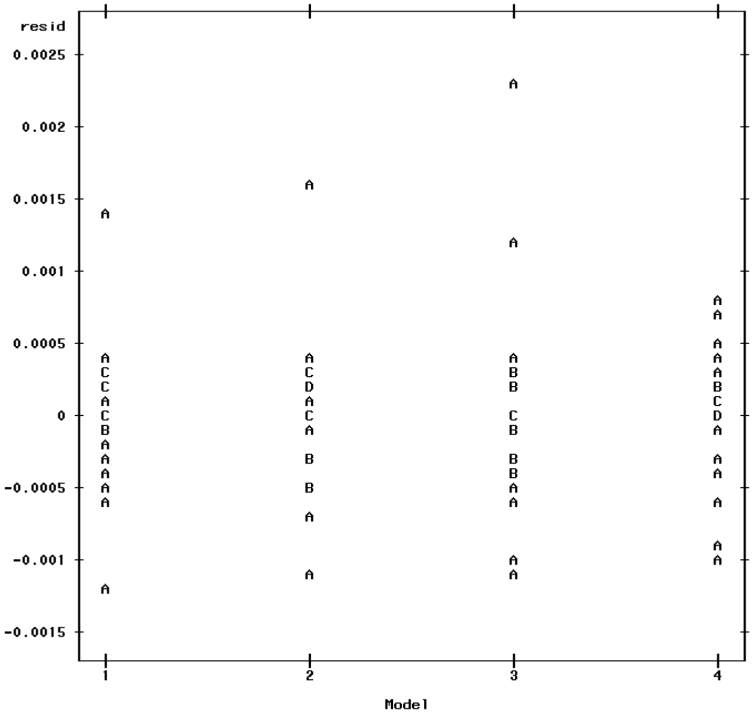
Size 1 = 8, 2 = 32, 3 = 128, 4 = 512.
Model 1 = SQRT, 2 = NSQRT, 3 = Thesis, and 4 = NL.

```
                                 The GLM Procedure
                              Class Level Information
                           Class         Levels    Values
                           Size               4    1 2 3 4
                           Model              4    1 2 3 4
                           Speci              5    1 2 3 4 5

                              Number of observations    80

                                 The GLM Procedure
 Dependent Variable: Zone
                                            Sum of
       Source                      DF         Squares     Mean Square    F Value    Pr > F
       Model                       19      0.00007571      0.00000398       9.39    <.0001
       Error                       60      0.00002547      0.00000042
       Corrected Total             79      0.00010118

                       R-Square     Coeff Var      Root MSE      Zone Mean
                       0.748248      58.14773      0.000652       0.001121

       Source                      DF       Type I SS     Mean Square    F Value    Pr > F
       Size                         3      0.00003061      0.00001020      24.03    <.0001
       Model                        3      0.00001150      0.00000383       9.03    <.0001
       Size*Model                   9      0.00000361      0.00000040       0.95    0.4938
       Speci                        4      0.00002999      0.00000750      17.66    <.0001

       Source                      DF     Type III SS     Mean Square    F Value    Pr > F
       Size                         3      0.00003061      0.00001020      24.03    <.0001
       Model                        3      0.00001150      0.00000383       9.03    <.0001
       Size*Model                   9      0.00000361      0.00000040       0.95    0.4938
       Speci                        4      0.00002999      0.00000750      17.66    <.0001

                                 The GLM Procedure
                    Tukey's Studentized Range (HSD) Test for Zone
 NOTE: This test controls the Type I experimentwise error rate, but it generally has a higher Type II
                                error rate than REGWQ.
                         Alpha                                0.01
                         Error Degrees of Freedom               60
                         Error Mean Square                4.245E-7
                         Critical Value of Studentized Range  4.59418
                         Minimum Significant Difference        0.0007

                  Means with the same letter are not significantly different.

                      Tukey Grouping          Mean       N    Size
                                   A     0.0019204      20    4
                                   A
                              B    A     0.0012930      20    3
                              B
                              B          0.0010748      20    2
                                   C     0.0001939      20    1

                                 The GLM Procedure
                    Tukey's Studentized Range (HSD) Test for Zone
 NOTE: This test controls the Type I experimentwise error rate, but it generally has a higher Type II
                                error rate than REGWQ.
                         Alpha                                0.01
                         Error Degrees of Freedom               60
                         Error Mean Square                4.245E-7
                         Critical Value of Studentized Range  4.59418
                         Minimum Significant Difference        0.0007

                   Means with the same letter are not significantly different.

                      Tukey Grouping          Mean       N    Model
                                   A     0.0017225      20    4
                                   A
                              B    A     0.0011413      20    3
                              B
                              B          0.0009046      20    1
                              B
                              B          0.0007137      20    2
```

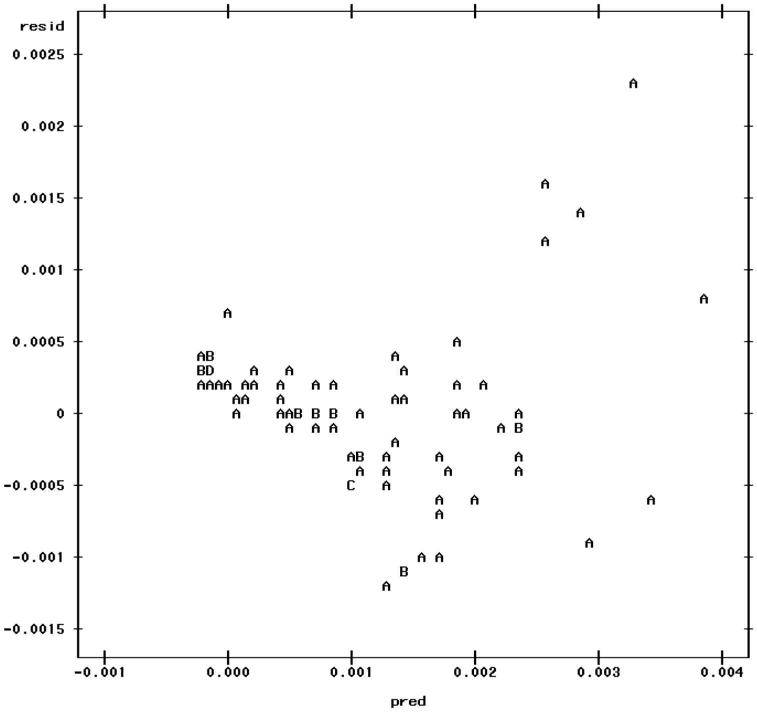
Plot of resid*Size. Legend: A = 1 obs, B = 2 obs, etc.



Plot of resid*Model. Legend: A = 1 obs, B = 2 obs, etc.



Plot of resid*pred. Legend: A = 1 obs, B = 2 obs, etc.



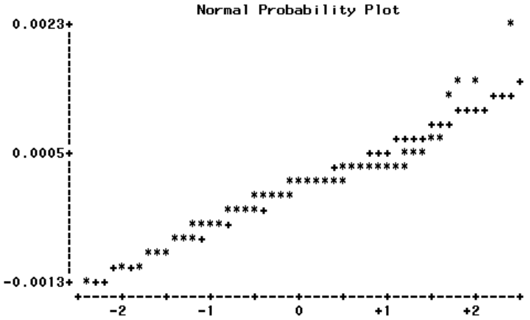
```
                         The UNIVARIATE Procedure
                            Variable:  resid
                                Moments
N                             80    Sum Weights                 80
Mean                           0    Sum Observations             0
Std Deviation         0.00056783    Variance           3.22427E-7
Skewness              0.95402386    Kurtosis           3.78794771
Uncorrected SS        0.00002547    Corrected SS       0.00002547
Coeff Variation                .    Std Error Mean     0.00006348

                      Basic Statistical Measures
         Location                    Variability
     Mean            0     Std Deviation            0.0005678
     Median   0.000011     Variance                3.22427E-7
     Mode            .     Range                      0.00355
                           Interquartile Range      0.0005662

                      Tests for Location: Mu0=0
        Test           -Statistic-    -----p Value------
        Student's t    t         0    Pr > |t|    1.0000
        Sign           M         3    Pr >= |M|   0.5764
        Signed Rank    S       -37    Pr >= |S|   0.8604

                         Tests for Normality
   Test                  --Statistic---    -----p Value------
   Shapiro-Wilk          W     0.908517    Pr < W     <0.0001
   Kolmogorov-Smirnov    D     0.143549    Pr > D     <0.0100
   Cramer-von Mises      W-Sq  0.331572    Pr > W-Sq  <0.0050
   Anderson-Darling      A-Sq  2.014533    Pr > A-Sq  <0.0050

                       Quantiles (Definition 5)
                      Quantile           Estimate
                      100% Max       2.33686E-03
                      99%            2.33686E-03
                      95%            1.00671E-03
                      90%            4.19396E-04
                      75% Q3         2.48705E-04
                      50% Median     1.07669E-05
                      25% Q1        -3.17463E-04
                      10%           -6.19248E-04
```

```
                 The UNIVARIATE Procedure
                    Variable:  resid
                 Quantiles (Definition 5)
                Quantile           Estimate
                5%             -9.89547E-04
                1%             -1.21439E-03
                0% Min         -1.21439E-03

                   Extreme Observations
--------Lowest-------         -------Highest-------
       Value      Obs                Value      Obs
-0.001214390       17          0.000777881       80
-0.001149400       18          0.001235545       39
-0.001116620       19          0.001420031       77
-0.000993539       75          0.001584106       78
-0.000985556       20          0.002336857       79
```

```
Stem Leaf                     #             Boxplot
  22 4                        1                *
  20
  18
  16
  14 28                       2                0
  12 4                        1                0
  10
   8
   6 88                       2                |
   4 135                      3                |
   2 0044456668001369        16             +-----+
   0 001122568033677778      18             *--+--*
  -0 110986555422            12             |     |
  -2 873310871                9             +-----+
  -4 418753                   6                |
  -6 4220                     4                |
  -8 992                      3                |
 -10 52                       2                |
 -12 1                        1                0
     ----+----+----+----+
 Multiply Stem.Leaf by 10**-4
```

```
                 The UNIVARIATE Procedure
                    Variable:  resid
```

Normal Probability Plot



ANOVA Results for Torus Form (Experiment 2)

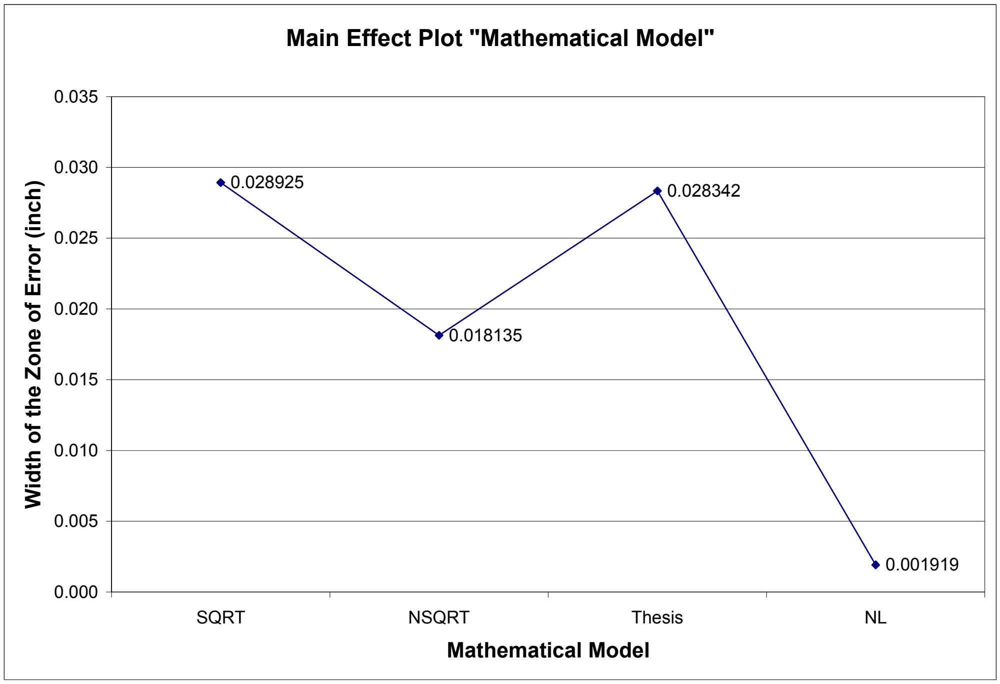


Figure D.9. Main Effect Plot of the Factor "Mathematical Model"

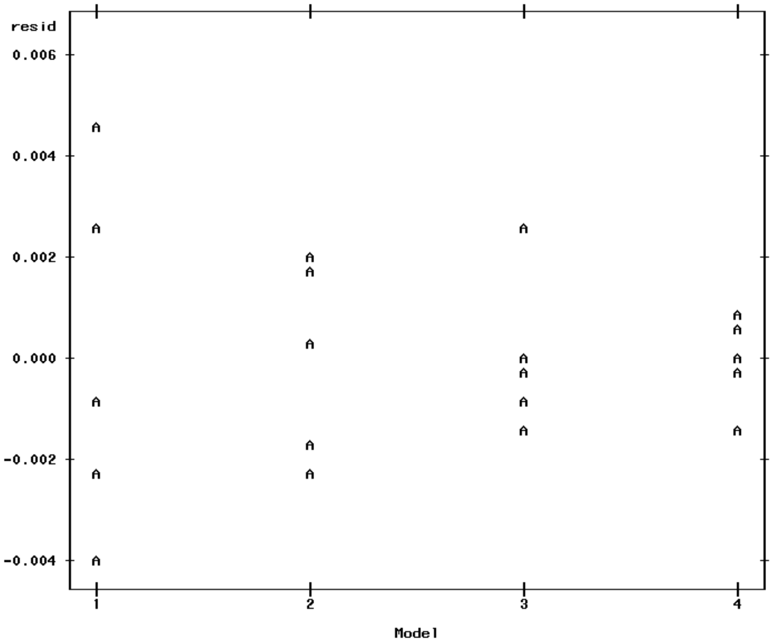
SAS Output Code for Torus Form (Experiment 2)

Model 1 = SQRT, 2 = NSQRT, 3 = Thesis, and 4 = NL.

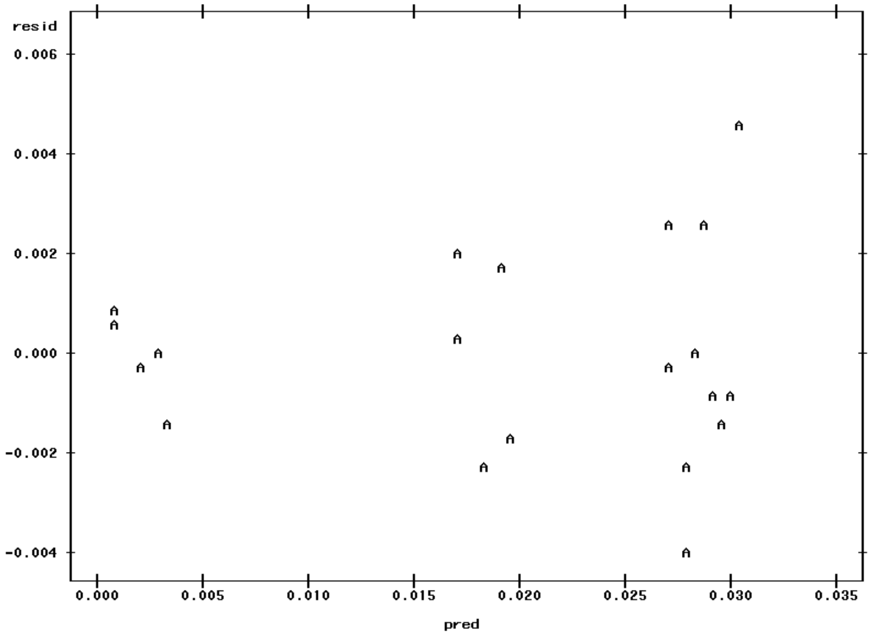
```
                                   The GLM Procedure
                                Class Level Information
                             Class          Levels    Values
                             Model               4    1 2 3 4
                             Speci               5    1 2 3 4 5

                                Number of observations    20

                                   The GLM Procedure
 Dependent Variable: Zone
                                              Sum of
       Source                      DF        Squares     Mean Square    F Value    Pr > F
       Model                        7     0.00241127      0.00034447      54.57    <.0001
       Error                       12     0.00007575      0.00000631
       Corrected Total             19     0.00248702

                        R-Square     Coeff Var      Root MSE     Zone Mean
                        0.969543      12.99751      0.002512      0.019330

       Source                      DF      Type I SS     Mean Square    F Value    Pr > F
       Model                        3     0.00238934      0.00079645     126.17    <.0001
       Speci                        4     0.00002193      0.00000548       0.87    0.5105

       Source                      DF    Type III SS     Mean Square    F Value    Pr > F
       Model                        3     0.00238934      0.00079645     126.17    <.0001
       Speci                        4     0.00002193      0.00000548       0.87    0.5105

                                   The GLM Procedure
                    Tukey's Studentized Range (HSD) Test for Zone
 NOTE: This test controls the Type I experimentwise error rate, but it generally has a higher Type II
                                 error rate than REGWQ.
                         Alpha                                   0.01
                         Error Degrees of Freedom                  12
                         Error Mean Square                   6.312E-6
                         Critical Value of Studentized Range  5.50160
                         Minimum Significant Difference        0.0062

                  Means with the same letter are not significantly different.
```

```
Tukey Grouping          Mean       N    Model
             A      0.028925       5    1
             A
             A      0.028342       5    3
             B      0.018135       5    2
             C      0.001919       5    4
```

Plot of resid*Model. Legend: A = 1 obs, B = 2 obs, etc.

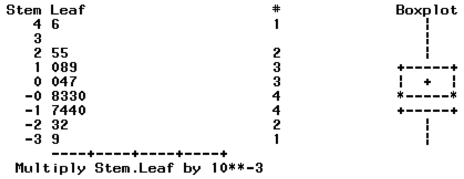


Plot of resid*pred. Legend: A = 1 obs, B = 2 obs, etc.

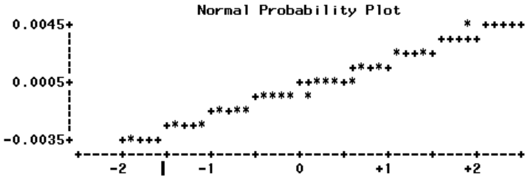


```
                       The UNIVARIATE Procedure
                          Variable:  resid
                              Moments
N                           20    Sum Weights                 20
Mean                         0    Sum Observations             0
Std Deviation       0.00199668    Variance            3.98675E-6
Skewness            0.34217002    Kurtosis            0.23910872
Uncorrected SS      0.00007575    Corrected SS        0.00007575
Coeff Variation              .    Std Error Mean      0.00044647
```

```
                 Basic Statistical Measures
    Location                    Variability
Mean      0.00000     Std Deviation            0.00200
Median   -0.00015     Variance              3.98675E-6
Mode       .          Range                    0.00843
                      Interquartile Range      0.00278

                 Tests for Location: Mu0=0
   Test           -Statistic-    -----p Value------
   Student's t    t         0    Pr > |t|    1.0000
   Sign           M        -1    Pr >= |M|   0.8238
   Signed Rank    S        -1    Pr >= |S|   0.9854

                    Tests for Normality
Test                  --Statistic---    -----p Value------
Shapiro-Wilk          W     0.984446    Pr < W      0.9780
Kolmogorov-Smirnov    D     0.096448    Pr > D     >0.1500
Cramer-von Mises      W-Sq  0.024815    Pr > W-Sq  >0.2500
Anderson-Darling      A-Sq  0.173169    Pr > A-Sq  >0.2500

                  Quantiles (Definition 5)
                 Quantile          Estimate
                 100% Max       0.004550377
                 99%            0.004550377
                 95%            0.003542628
                 90%            0.002534137
                 75% Q3         0.001378199
                 50% Median    -0.000147198
                 25% Q1        -0.001403924
                 10%           -0.002256198
                  The UNIVARIATE Procedure
                      Variable:  resid
                  Quantiles (Definition 5)
                 Quantile          Estimate
                 5%            -0.003075391
                 1%            -0.003879819
                 0% Min        -0.003879819

                    Extreme Observations
   -------Lowest-------         -------Highest------
         Value      Obs              Value      Obs
   -0.00387982        1         0.00177772       14
   -0.00227096       18         0.00186951       10
   -0.00224143        9         0.00253340       17
   -0.00174253        6         0.00253488        3
   -0.00142709        7         0.00455038        5
```



```
 Stem Leaf                     #             Boxplot
    4 6                        1                |
    3                                           |
    2 55                       2                |
    1 089                      3             +------+
    0 047                      3             |  +   |
   -0 8330                     4             *------*
   -1 7440                     4             +------+
   -2 32                       2                |
   -3 9                        1                |
      ----+----+----+----+
 Multiply Stem.Leaf by 10**-3
```



```
                       Normal Probability Plot
 0.0045+                                            * +++++
       |                                         +++++
       |                                    *++*+
       |                               +*+*+
 0.0005+                         ++****+*
       |                    +**** *
       |               +*+**
       |          +*++*
-0.0035+     +*+++
        +----+----+----+----+----+----+----+----+----+----+
            -2    |   -1          0        +1        +2
```