Automatic Feature Recognition and Tool path Generation Integrated with Process Planning

by

Vinodhkumar Somavar Muniappan

A thesis

presented to the University of Waterloo

in fulfillment of the

thesis requirement for the degree of

Master of Applied Science

in

Mechanical Engineering

Waterloo, Ontario, Canada, 2012

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

The simulation and implementation of Automatic recognition of features from Boundary representation solid models and tool path generation for precision machining of features with free form surfaces is presented in this thesis. A new approach for extracting machining features from a CAD model is developed for a wide range of application domains. Feature-based representation is a technology for integrating geometric modeling and engineering analysis for the life cycle. The concept of feature incorporates the association of a specific engineering meaning to a part of the model. The overall goal of feature-based representations is to convert low level geometrical information into high level description in terms of form, functional, manufacturing or assembly features.

Using the boundary representation technique, the information required for manufacturing process can be directly extracted from the CAD model. It also consists of a parameterization strategy to extract user-defined parameters from the recognized features. The extracted parameters from the individual features are used to generate the tool path for machining operations regardless of the intersection of one or more features. The tool path generation is carried out in two phases such as roughing and finishing. Various types of tool paths such as one-way, zig-zag, contour parallel are generated according to the type of the feature for the roughing operation. The algorithm automatically plans the sequence of machining operation with respect to the feature location, and also selects the type of tool and tool path to be used according to the feature.

The finishing operation uses the tool path generation strategy in the same manner as used in roughing operation. The algorithm is implemented using the Solid works API library and verified with CNC milling simulator. The results of the work proved the efficiency of this approach and it demonstrate the applicability.

Acknowledgements

I am truly and deeply thankful to my Supervisor, Dr. Sanjeev Bedi for the amount of knowledge that he shared with me, along with the countless amounts of time that he spent for me. The enthusiasm and effort that he had put into this topic and to make me better off will always be remembered. I heartily thank him, for offering me this wonderful opportunity. Without his guidance and persistent help, this thesis would not have been possible. For all his help throughout the years, I am indebted to him.

I would like to expresses my gratitude to Dr. Gregory Glinka and Dr. Behrad Khamesee for being my thesis readers. Special thanks to Rajnish bassi and Kandarp patel who helped me in making shape to my ideas and discussing the latest advancements in the field.

I would like to expresses my gratitude for the enormous support that my family has given me throughout this entire process. I would not have made it this far, without the vast amount of love, support, and courage that they have given me. I especially want to thank my parents and brother who are the backbone to my success thus far. Their love shall be forever cherished.

To my parents and my brother…

Contents

List of Figures

Chapter 1

Introduction

Over the last few decades, three-dimensional (3D) geometric modeling has been widely used in various engineering fields, like Computer Aided Design and Computer Aided Manufacturing (CAD/CAM), rapid prototyping, virtual reality, etc. The requirement of geometric data processing varies with the fields of application. For example in rapid prototyping a triangulated model suffices whereas in engineering objects such as aerospace components, automobile components, injection moulds, turbine blades and dies, complex surfaces are required. The modelling system must be able to meet the needs of the various applications.

These geometric applications are aids to manufacture a part. The process of manufacturing these parts is an expensive and time consuming task. The global competition between the manufacturing industries has increased the demands to reduce production times and increase quality of the product. The attempt to automate processing of geometric data and reduce the manual effort for handling complex and huge geometries has been persistently pursued over the last two decades. This pursuit has resulted in drastic improvement in CAD systems that allow the designers to easily build complex geometries into the models.

The part design is pursued with CAD software and it is typically represented as a solid model by using either Boundary Representation or Constructive Solid Geometry or a hybrid scheme. The CAD software is a suite of modules designed to assist the designer in creation, visualization, analysis and other activities. The CAD software contains many modules such as a geometric module, an analyses module, a communication module, a collaborative module and an application module. The geometric module allows the user to perform several operations such as model construction, editing and manipulation of geometry, scaling, rotation, translation, drafting and documentation. The analyses module allows the user to analyze their design according to their application field such as electrical, architectural and mechanical. Figure 1.1 shows an example of a model created with a CAD package.

Figure 1.1 CAD model of a sample bracket

The communication module is used to interact between CAD/CAM systems, other computer systems and manufacturing facilities, translating databases between CAD/CAM systems using Initial Graphics Exchange Specification file (IGES) and Standard for the Exchange of Product model data (STEP). These neutral formats (IGES or STEP) pave a way for geometrical data exchange among different software packages for solid modelling. Most CAD/CAM systems have implemented Initial Graphics Exchange Specification file which provides the entities of points, lines, arcs, curves, curved surfaces and solid primitives to precisely represent CAD models. As each CAD system has its own internal method of representing geometry, both mathematically and structurally, there exists some loss of information while translating from one CAD data format to another. The concurrent work on the same part, assembly or drawing file can be done in real time over the web using a collaborative module. The application module allows a user to write custom modules. The Application Programming Interface (API) module is an important tool used by many manufacturing engineers as it allows customizing CAD system for certain design and manufacturing tasks which also enables extension of application functionalities. Almost all the CAD systems contain an Application Programming Interface module. This module allows the user to develop algorithms to automate the design process or to add additional functionalities to the CAD system.

Solid models are created in a CAD package using a graphical user interface (GUI) with the help of mathematical and parametric tools. The solid modeling based CAD package plays an important role during the design and manufacturing phases of a product life cycle. Through CAD modeling, a virtual representation of the final product that needs to be manufactured can be created. Initially, CAD was used as a drafting aid for making engineering drawings. In time it evolved into wireframe modeling and subsequently in to hidden line removed models.

The real development in the computer aided design field started with a search for methods to represent solids and the evolution of solid modeling. Solid modelling is a preferred method of representing mechanical parts because it is soundly based on mathematical theory and can be used to classify points, lines and spaces relative to the part. CAD modeling is also used to get solutions to geometric information such as volumes, surface area and sectional properties of the designed part. Figure 1.1 shows a CAD model of a sample bracket. CAD encompasses the entire array of computer tools that are used as design aids. The ability to write geometric algorithms is unique to solid modelling: previous geometric modelling methods such as triangulated surfaces, wire frame, hidden line removed models did not possess this capability. This ability is the key to the CAD systems today. The solid model serves as the geometry and topology database and other modules such as GUI, analyses, visualization, etc. use the database with interfacing geometric algorithms.

Feature Based Design

Solid modelling provides simple entities to create a part. If these primitive entities are used, then creation of a part can take a long time and make solid modelling unattractive to use. Hereto, the ability to write geometric algorithms using API is handy as it allows the development of modules that create geometries with a certain topology, using an algorithm and parameters supplied by the user. Such geometric modules that create new geometries with fixed topology are called **Geometric or Form Feature**. The concept of feature based design is the most fundamental aspect in creating a solid model. A part model starts with a base feature and then additional features are added one at a time until the accurate representation of a part's geometry is attained. A feature is a basic block that is used to build a design by merging with other features. The use of features to build solid models is such a common routine that simple features like blocks and cylinders are considered as standard parts of all solid modellers and are used as building blocks for complex parts. The designer can create a geometric feature in two ways, one is to sketch a section of the shape to be added and then extrude it, revolve or sweep it in order to create a shape which is termed as sketched geometric features. The second way uses pick and place geometric features. In this the designer simply chooses a geometric feature from the library and places it on the part at a specified location such as placing a hole on the model. This process of working with geometric feature based solid modelling method is like sculpting parts from solid material. Using pre-defined geometric features to design a solid model would substantially reduce the number of input commands required, which also makes redesigning an easy task.

Geometric features relate geometry and topology of a part. This relationship is also the key to many manufacturing operations. For example, tool paths for Computer Numerical Controlled (CNC) machining depend greatly on geometry and topology. These features that use the geometry and topology to assist with manufacturing operations are called manufacturing features. The idea of geometric features can be extended and the geometric algorithm can be used to store analyses and manufacturing information about the part. A system based on such geometric algorithms is called a **Feature Based Design (FBD) system**. When two types of information, namely, geometric and machining are integrated into one feature, some difficulties arise when features interact with one another.

It is easy to imagine a rectangular cavity feature being merged with a cylindrical Boss as shown in Figure 1.2 and to imagine the resulting geometries but the impact on CNC tool path is unclear. The cylindrical Boss that lies inside the rectangular cavity remains as an obstacle for tool movement. The issue with FBD is that a geometric algorithm must consider all possible cases of interaction between features. Such systems have been successful in companies with a strong engineering support coupled with a strong computer science group.

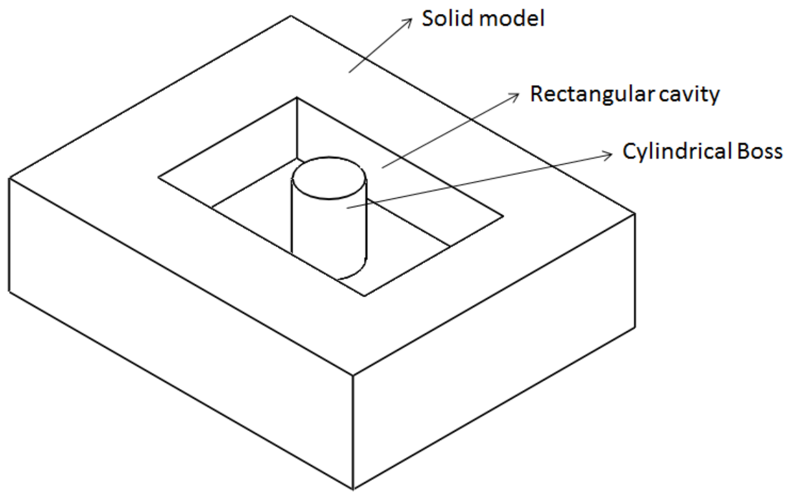


Figure 1.2 Cylindrical Boss inside rectangular cavity

Clearly FBD is the forte of large companies and is infrequently used by small companies. In small companies the geometry of the part is created independent of the machining data. The current commercially available CAD/CAM system requires some important human inputs such as side step, feed rates, tool sizes before tool paths can be created. Furthermore, the cost involved in the implementation of CAD and CAM is very high. It would be a great attraction for small to medium size companies if the machining information could be extracted automatically. One method explored in literature is called **Feature Based Machining**. The geometric features may differ from the manufacturing features in which case feature recognition becomes necessary. The development of these extraction algorithms involves recognition of all individual features of a solid model and its parameters and requires experienced engineers and programmers. Though the current available feature recognition module in the CAD system can recognize some basic geometric features from the solid model, it’s not designed in a way to identify complex features or interferences between multiple features, which is the primary and most challenging task in planning gouge-free tool paths. The feature based machining modules currently cited in literature [40][41] use tool offset methods to determine the tool path. In this method the tool is offset to the underlying geometry along a pre-specified tool path foot print. In this step of offsetting, the tool may gouge another feature or geometry. For example, assume a cylindrical Boss inside a slot feature, the tool may gouge the Boss feature as shown in figure 1.3. Gouging must be avoided as it destroys the part and is expensive. Although simple gouge avoidance strategies have been tested in literature [40][41][43] they are not robust and reliable. Consequently, feature based machining has not gained popularity in usage.

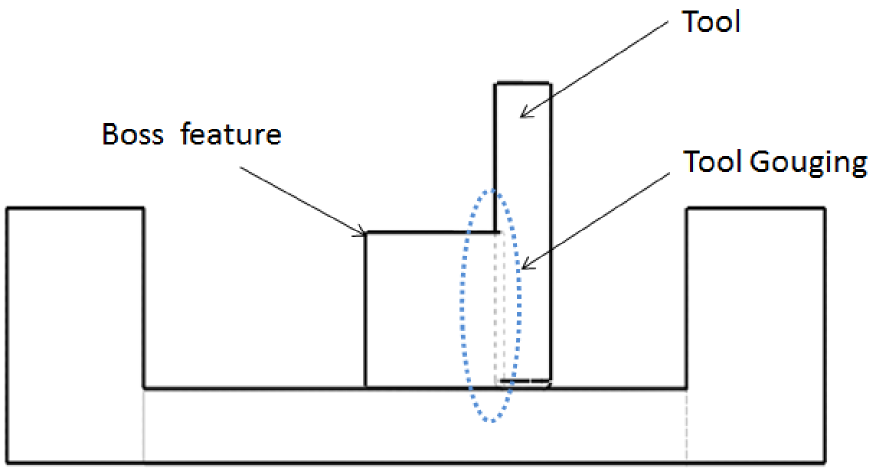


Figure 1.3 Tool gouging with a Boss feature

The tool path is the trajectory followed by the cutter center to remove the material on the surface. The tool path generation is the collection of a sequence of points from the design surface, which the tool follows to remove material on the surface. The tool path generation can be classified as either cutter contact based method or cutter location based method depending on the surface type. The cutter contact based method generates tool paths by dividing the part surface into a sequence of cutter contact points which is then converted to cutter location points. This method is carried out based on three methods: parametric, drive surface and guide plane methods. The cutter location method generates the tool path by creating a cutter location surface on the design surface. Based on coordinates extracted from the surface, tool path planning can be either Cartesian based or parametric based. The Cartesian based tool path is planned on the x-y plane of Cartesian coordinates by composing the intersecting curves between the surfaces and the planes. In parametric based tool path, the surface data can be directly used in tool path generation.

There is a need to develop a system that will automatically generate tool paths from the part design. Part design can be in various formats; however a solid model of the part is a universally accepted standard. Thus a part defined as a Boundary Representation solid model will be used in this work. The B-Rep model will be processed to identify geometric entities (features) that have topological similarity. From the feature the topology is known and thus an appropriate tool path will be planned.

A method capable of recognizing machining features from the CAD model and automatically planning machining sequence for a 3-axis CNC machine in the form of a tool path is proposed in this work. This algorithm has been developed from a generative point of view, meaning that it generates the tool path for each identified feature. The algorithm also creates a process plan which is used to determine the sequence of machining.

The initial step towards automatic feature recognition from a solid model is the extraction of geometric and topological information and representing it in a structured manner. The geometric information is defined in terms of low level entities such as lines, planes, circles. Topological information defines the relationship between the geometric entities. By identifying the type of entity and computing the relation between a set of entities, a feature type can be defined. It is also important to define the parameters of the feature which are necessary for generating tool paths. All operations performed by a feature recognition module become the primary input for a tool path planning module to generate cutter contact path.

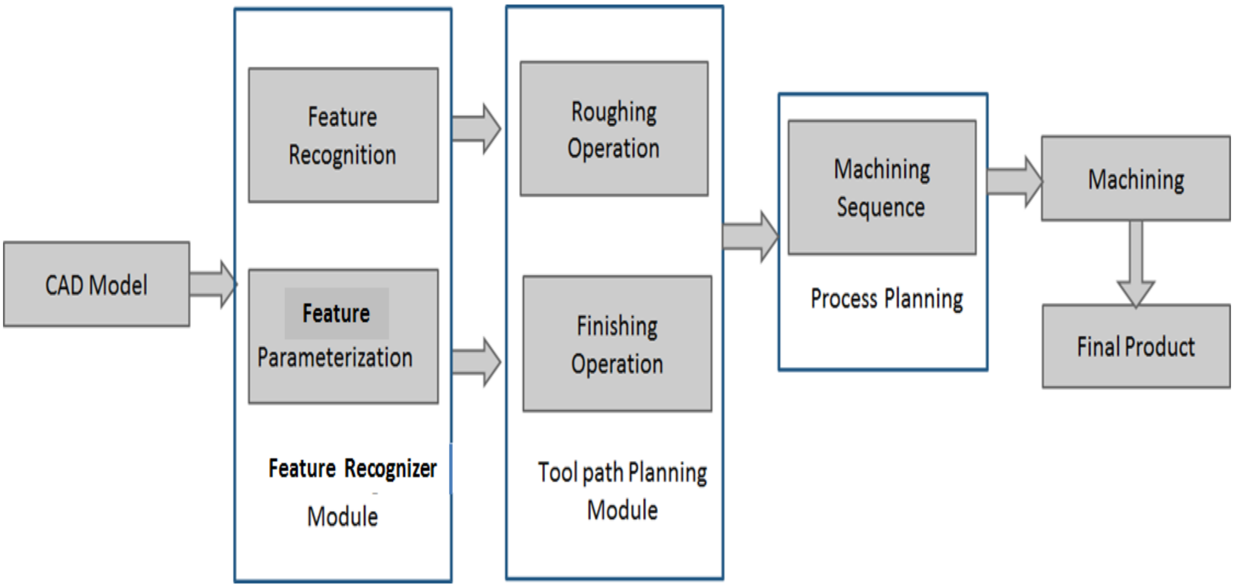


Figure 1.4 Schematic representation of proposed strategy

The tool path module generates two different tool paths for roughing and finishing operations. Though the cutter contact path and cutter contact points are generated by the tool path planning module, it’s important to compose a gouge-free tool position. Apart from planning gouge-free tool positions, the interconnected Process Planning module also illustrates the machining sequence of geometric features according to its type and location. The parameters of geometric features such as its size, location, depth, etc. are given by the feature recognition module. Three essential elements are required to machine the effective volume of the machining feature, namely, the tool approach direction, the cutting tool position face and the cutting depth. The process planning module extracts this information from the feature recognition module to decide the exact tool position, the tool travelling direction and the depth of cut.

The current CAD/CAM systems require a user decision to do the process and hence it cannot provide an automated and intelligent decision to optimize a 3-axis machining strategy. Considering this problem, it is desirable to have a system that can automatically extract features and generate tool paths which can make intelligent decisions without any user input. Such system will be proposed in the current work.

1.1 Research Objectives

The goals of the current research is to combine the concepts of automatic feature recognition and feature based machining with process planning, which can generate an optimal tool path planning system for the 3-axis CNC machining. The cost of CAD, CAM and CNC machining can be reduced by coupling these three concepts in an automated system. A combined system can solve various manufacturing issues by offering functionality of design and manufacturing in a single package.

Similar systems have already been created; such is the case of *WatCAD/CAM* at University of Waterloo: a web-based CAD and CAM system which allows users to easily design and manufacture table legs. The software is developed by linking a solid modelling package with a custom CAM package and is used with a CNC milling lathe designed specifically to carve wooden legs. Another example of this type of system is *WatSign* [40] consisting of a web-based solid modeller paired with CAM software which allows users to design plaques online and download tool paths required to machine plaque in 3-axis CNC milling machine. However, these systems are not based on features of the solid model instead the solid model is represented by a triangulated mesh to generate a tool path. Though the triangular meshes are commonly used to represent sculptured surfaces, a few drawbacks make it difficult to generate tool paths. Topological information is not provided for the triangular facets and an accurate representation requires a large number of triangular facets which increases processing time. In feature based machining, the topological information can be extracted easily and also the processing time is shorter than the time required to process triangulated surfaces.

In particular, the objectives of this current work are:

1. To automatically recognize all the features of the solid model including freeform surfaces such as Bezier, B-spline surfaces and to extracts its parameters.
2. To design an algorithm capable of generating the tool path according to the identified feature and also to be capable of automatically selecting the tool path type to be generated with respect to the feature.
3. To conduct machining tests either with a machine or a simulator and to compare with other machining strategies to demonstrate the efficiency of the proposed 3-axis machining methodology.

1.2 Thesis Layout

In Chapter 1, a general introduction is given to highlight the need for automatic feature recognition and tool path planning, its background and also to outline the objective of the thesis. Chapter 2 describes the important concepts related to computer aided design and 3-axis machining. The conditions required to generate the machining configuration are also explained. This chapter also presents a survey of the literature related to feature recognition and tool path planning. The proposed methodology for feature recognition is described in detail in Chapter 3. The procedures required for extracting the features, as well as, its parameters and the results obtained are explained. In Chapter 4 detailed descriptions of the proposed methodology for tool path generation based on the identified features and methods for planning the sequence of machining is given. This chapter also illustrates the implementation method and the results obtained from the proposed work. The results demonstrate the efficiency and reliability of the proposed work. Chapter 5 discusses the final conclusion and considerations about the further developments to this work.

Chapter 2

Theoretical background

Solid modeling is a mathematical modelling technique for algorithmically building a complete physical representation of solid objects. Solid modeling in computers is done using various techniques such as Constructive Solid Geometry (CSG) or Body Representation (B-Rep). A solid modeler based purely on CSG modeling has an easy to use interface but it is difficult to classify and to visualize the models created in CSG whereas a B-Rep model is hard to construct but lends itself to efficient classification and viewing. Most of current solid modelers are hybrids which use CSG as a user interaction tool along with other tools and maintain the internal representation in terms of B-Rep model.

2.1 Boundary Representation (B-Rep)

Boundary representation is one of the solid modelling techniques in which the geometry of a part is described in terms of its bounding surfaces. This geometric information associated with four topological entities such as face, loop, edge and vertex forms the basic constituents of Boundary-representation models. B-Rep is a geometric implementation of the mathematical theory of 2D manifolds. The solid model of a part is composed of bounding faces. A face is a mathematically defined surface with a defined boundary. A face may contain several bounding loops. Each face has one outer loop, but may have many internal loops. Internal loops lie within an outer loop. Loops inside an inner loop do not belong to the given face and therefore describes a new face. Each loop is made of edges. Each edge is a curve with a defined start and end vertex. Figure 2.1 illustrates the basic constituents of B-Rep models.

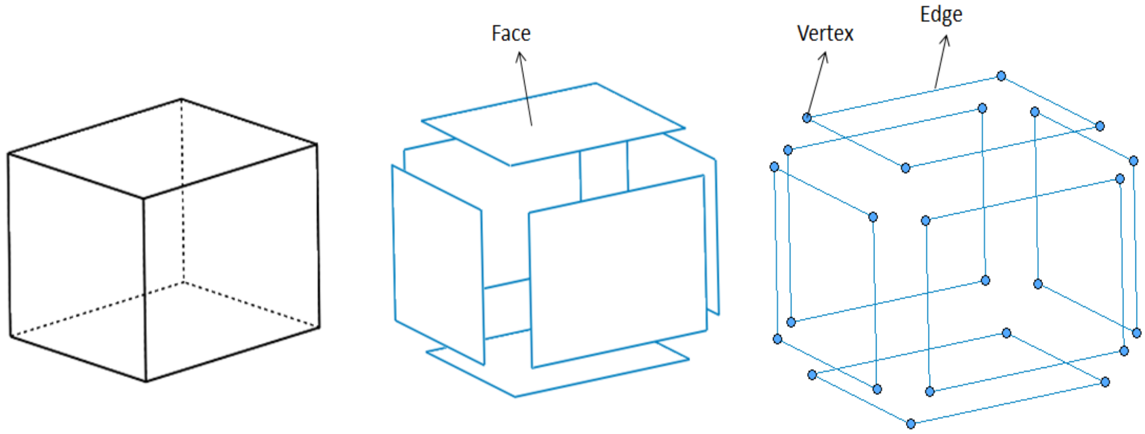


Figure 2.1 Constituents of B-Rep models

The geometric information is contained in the face and edge equations and vertex coordinate. The relation between faces, edges and vertices describes the topological information of the part. The topological information in B-Rep is typically implemented via a winged data structure. The data structure is built using doubly linked lists and can be traversed forward and backward to obtain topology information. The additional geometrical data which is stored in a boundary representation technique includes transformation, rotation, angles, distances, area, etc. However, to extract the features from the solid model the topology relation must be stated between each face in order to identify the shape and volume of the model. For example, the angle between two adjacent faces can be determined easily using the geometric data of B-rep models. Much of the research in the past few decades used boundary representation to extract machining features. The property of B-Rep models to describe the topological relation is the key to developing custom algorithms for feature recognition.

2.2 Constructive solid geometry (CSG)

In the constructive solid geometry modelling technique, the solid model of the part design consists of a variety of different solid primitives combined together using regularized Boolean operations to create a solid model. Cuboids, cylinders, pyramids, spheres, cones and prisms are some of the predefined primitives available in CAD packages today. The regularized Boolean operation such as union, intersection and difference are the key to B-Rep based solid modellers; however, they are computationally expensive and thus have given way to other easier methods of building a solid such as sketching, extrusion etc. For example, Figure 2.2 illustrates two rectangular blocks which are combined together by rotating one block to a vertical position and by combining them using a union operation. The final model is obtained by subtracting a cylinder from the previous model.

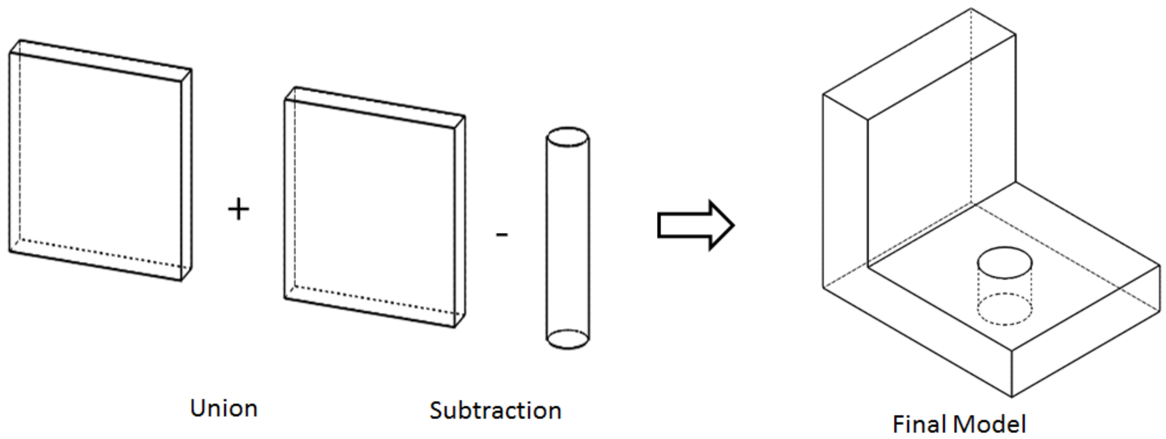


Figure 2.2 Constructive Solid Geometry

Constructive solid geometry allows the user to describe a solid model easily when compared to boundary representation. The CSG based modelling is a desirable method for user interfaces, but extracting geometric and topological information from CSG model is not an easy task.

As discussed in the previous chapter, in order to automate the manufacturing process, the CAD and CAM must be integrated. Feature recognition can play a significant role to facilitate this integration between CAD/CAM. Feature Based techniques are the key for efficient feature recognition and for developing machining features. Before machining features can be developed an understanding of 3-axis CNC machine is essential. Given below are some common terms and definitions used in 3-axis CNC machining.

2.3 3-Axis CNC Machining

In order to utilize the identified machining feature to manufacture a part, a 3-Axis CNC machine is used. A CNC machine holds the part stock in its workspace by fixing it to the machine table and moves it relative to a rotating tool. The relative motion removes material from the stock taking it closer to the desired shape. The series of movements of the part relative to the tool is called the tool path or tool trajectory. In a 3-Axis CNC machine there are three linear axes. The axes are used to move the tool relative to the part. Each axis is aligned along one of three principle coordinate axis and the part or tool is moved along it. The tool motion is described in the work piece coordinate system using a tool path. The Cartesian coordinate system is a simple way of defining the location of the tool with respect to the part. Figure 2.3 shows the general setup of the 3-axis machine. Each CNC machine has two coordinate systems, the machine coordinate system and the work piece coordinate system. The machine coordinate system of the 3-axis CNC machine is defined by the manufacture and cannot be changed. The homing or zeroing of each axis is a process for calibrating the machine coordinate system and it is done every time after a restart.

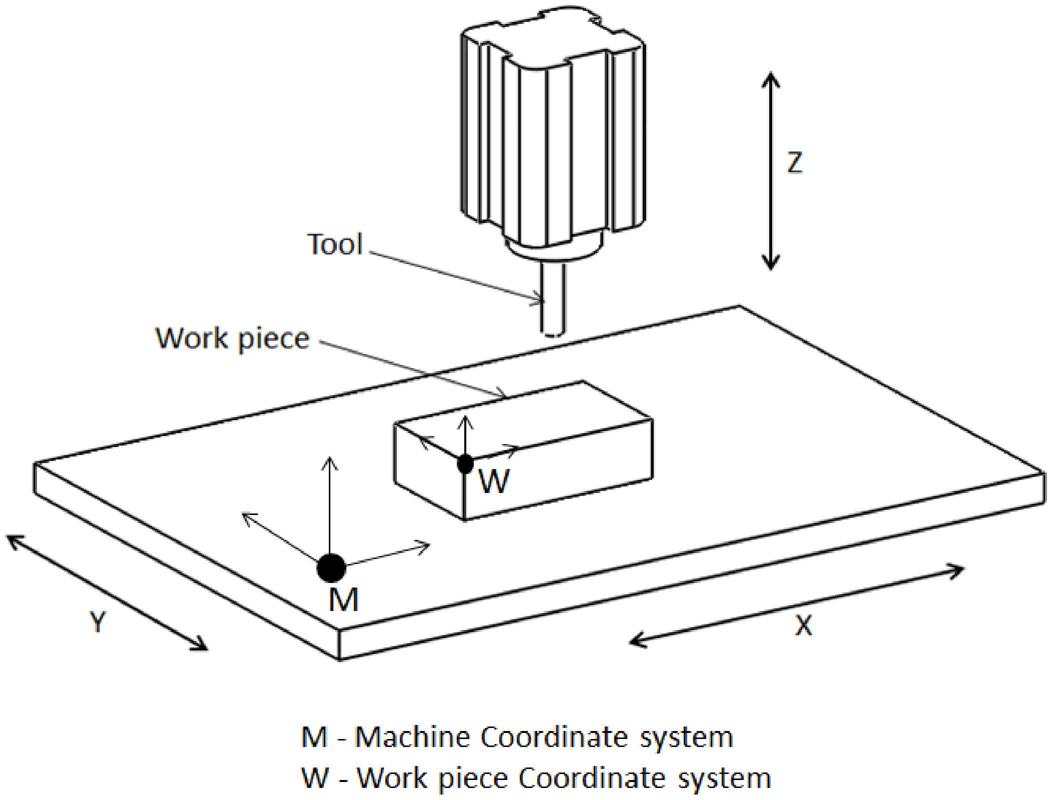


Figure 2.3 3-axis machining setup

The work piece coordinate system is the coordinates of the work piece defined by the programmer in respect to the machine coordinate system. This work piece coordinate system can be changed at any time by the user. A tool path relative to these coordinates systems is composed of several tool positions and the tool moves from one position to the next. The motion between points is along a linear vector joining the start and end points or along a circular arc of a user defined radius. Curved surfaces are machined with a zig-zag tool path with each leg or path being composed of a number of tool positions separated by a feed forward distance. The spacing between the legs of the zig-zag path is called a side step or step over distance. Generally, the tool path is described in terms of the center point of the tool. Based on the tool geometry, the tools are classified into several types as discussed below.

2.3.1 Types of tools

Ball nosed, flat end mill and toroidal end mills are the three most commonly used end milling cutters. The cutting operation occurs by rotating the tool about its axis and translating it through the stock along a defined trajectory. The rotational cutting speed of end milling cutters is much higher than their translational feed rate. This allows the ball nosed cutter to be modeled as a sphere, the flat-end milling cutter to be modeled as a cylinder, and the toroidal end mill to be modeled as a torus.

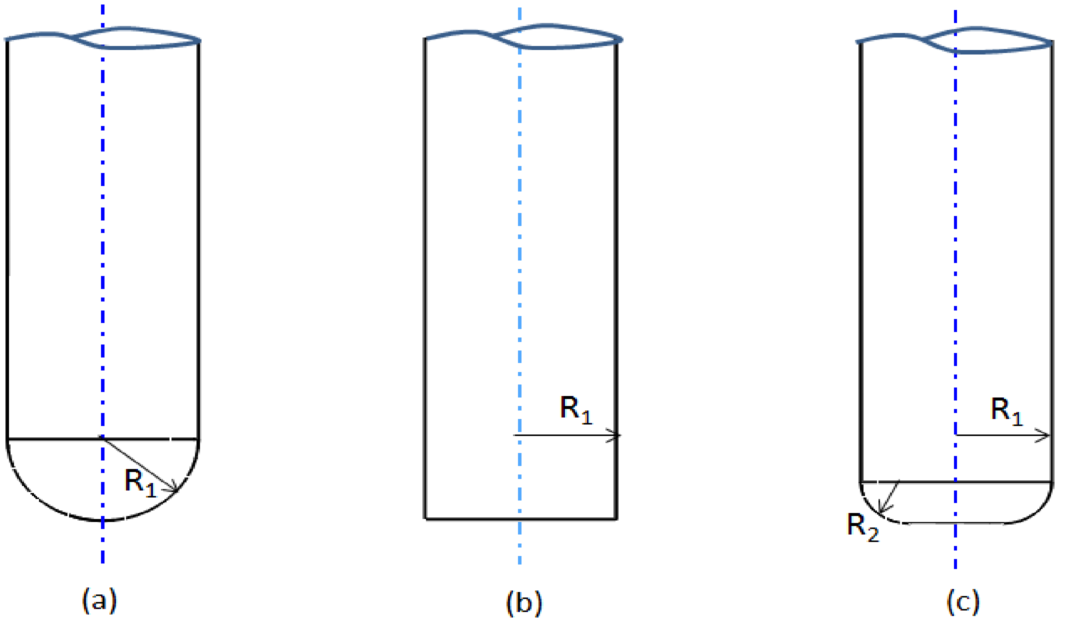


Figure 2.4 (a) ball nose (b) flat end (c) toroidal end mill

Figure 2.4 shows the geometry of the three end mill cutters. The radius of the bottom edge of flat-end mill is filleted with a desired radius, in order to produce a toroidal end mill. A toroidal end mill which has a circular insert of a minor radius (R1) and a major radius (R2) can be used to represent the shape of both ball nose and flat-end mills. Thus, if R1 = 0 then the geometry of the toroidal cutter will be the same as an end milling cutter and, if R2 = 0 then it would resemble a ball nosed cutter.

Generally, ball nose end mill cutters are the commonly used cutters for machining complex surfaces of the part. But when using a ball nose cutter to machine curved or flat surfaces, a portion of material is left uncut between the tool passes, since the tool geometry does not exactly match the surface geometry. This uncut area is called a scallop. The ball nose end mill has a fixed spherical cutting surface, which eventually leads to larger scallop heights and requires more passes to machine a part.

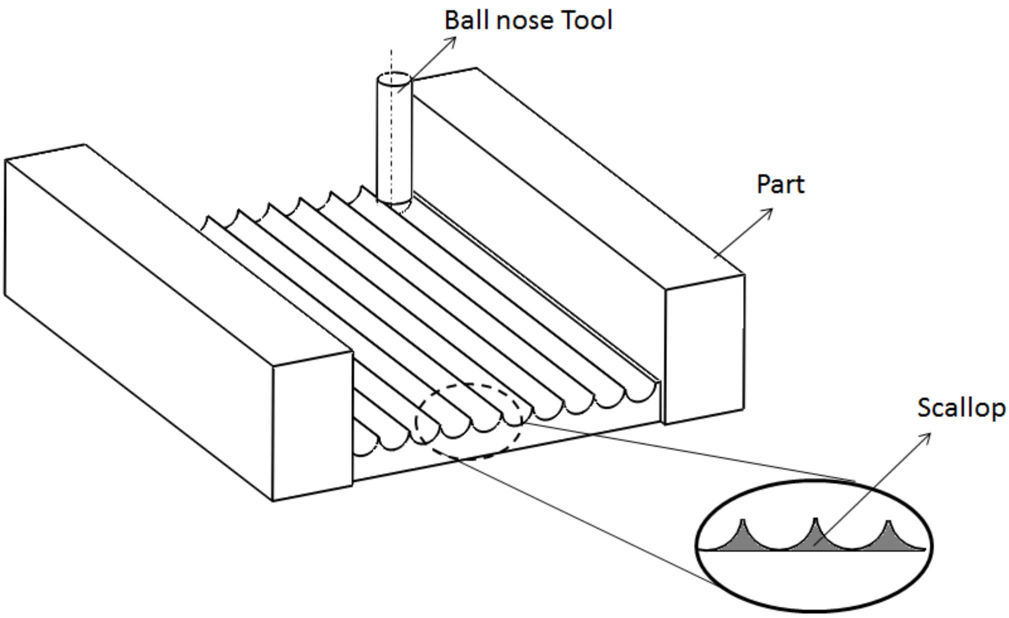


Figure 2.5 Scallop heights in machining a slot using ball nose cutter

Figure 2.5 illustrates the scallops formed during machining a slot using ball nose end mill. The low scallop height in 3-axis machining may require long machining times. Furthermore, the surface finish of the work piece is poor with a ball nose end mill, since the tip of the ball has zero radius cutting speed. The flat-end mills have more surface contact with the work piece and result in smaller scallop heights. But the zero corner radius flat-end mills may produce marks along the feed direction, thereby increasing surface roughness. Since the toroidal cutters do not have zero corner radiuses, it has the merits of both ball nose and end milling without their deficiencies [7].

2.4 Feature Based Design

A feature is the set of information which relates the geometry and the topology obtained from the design to the tool path parameters for manufacturing. The vital role of feature based machining is the integration of design and manufacturing. The various types of manufacturing features are discussed below.

2.4.1 Manufacturing features

Li [8] classified the manufacturing features into four categories: transition feature, machining feature, replicate feature and region feature. By keeping this classification as central, the machining features are further classified into compound and tool path features based on its accessibility for tool path planning. Figure 2.6 illustrates the classification of manufacturing features.

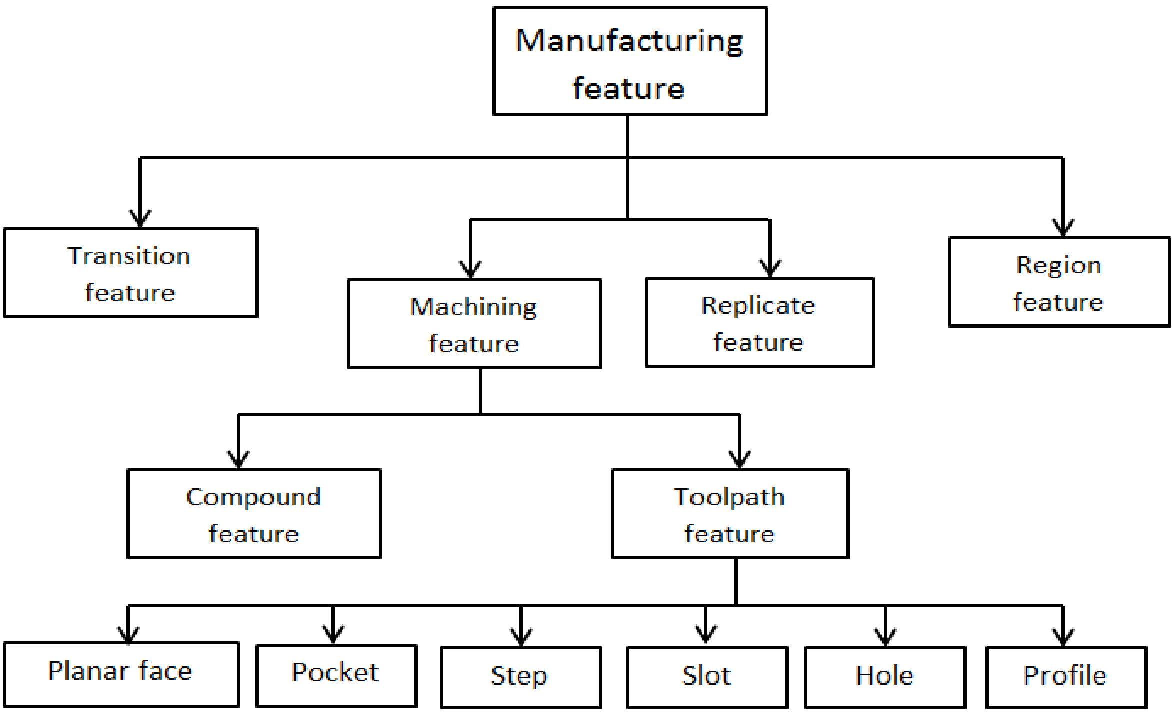
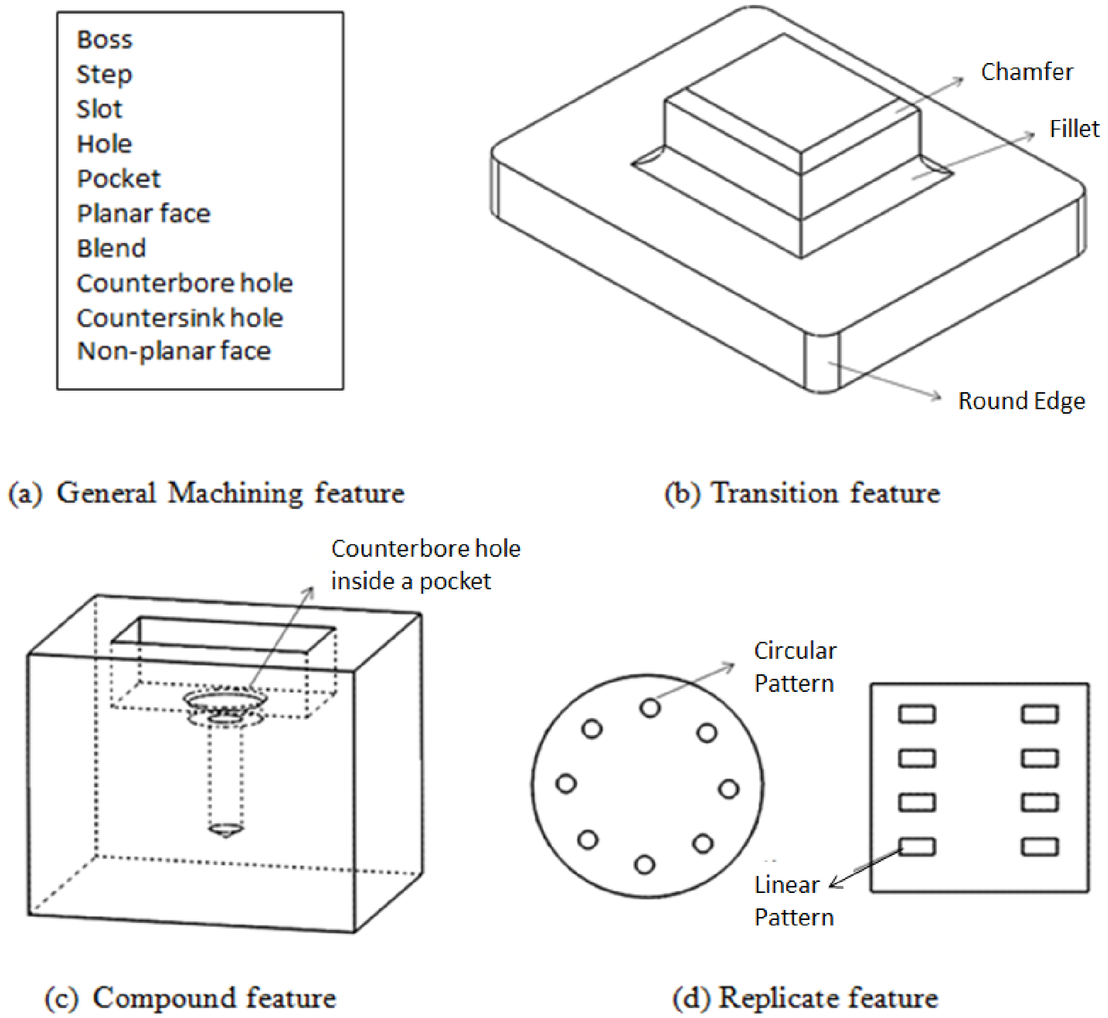


Figure 2.6 Classification of manufacturing features

The most essential manufacturing features are said to be machining features such as hole, pocket, slot and step, etc., and a group of these machining features which are arranged in three patterns such as circular, rectangular and general patterns are said to be replicate features.



(a) General Machining feature (b) Transition feature

(c) Compound feature (d) Replicate feature

Figure 2.7 Manufacturing features

The auxiliary parts of the machining features which act as connecting parts between them are said to be transition features such as chamfer and fillet. The compound feature is a combination of several machining features together into a single complex feature, and the features which determine the surfaces for freeform milling are region features. The region feature can be described by two methods as follows,

1) Explicit geometry

   It is the feature that has been determined by using a surface for freeform milling. It is most commonly used to define the region features.

2) Implicit geometry

   It is the feature that has been designed using profile and path parameters. This is the method used to describe most of the machining features. For example, a circular profile and a linear path define a hole.

A seamless interface can be established between product design and manufacturing applications by the feature recognition algorithm, which identifies the manufacturing features by using a geometric model or design by feature model. Although Li[8] proposed this classification he has not implemented and tested it for verifying its adequacy.

2.5 Feature Recognition

A CAD model has its own geometric and topology data that is stored in entities, such as vertex, edge, face, etc. Similarly, a CAM machining model has its own data, such as Tools Accessing Direction (TAD), tool path, and the cutting depth, etc., which is used to generate a machining plan to form a desired CAD model. In order to combine the operation of CAD/CAM, it is necessary to extract the features from the CAD model.

Based on the geometric modelling representation, feature recognition techniques can be generally classified as boundary based schemes and volumetric feature recognition schemes and the feature recognition approach which is integrated with the design by feature approach is the hybrid scheme. The boundary based scheme is based on boundary representation method which uses the basic geometric configuration of CAD model such as faces, loops, edges and vertices to identify and extract the manufacturing features of the part. This boundary based scheme can be further classified into rule based, graph based, hint based and artificial neural networks based approaches.

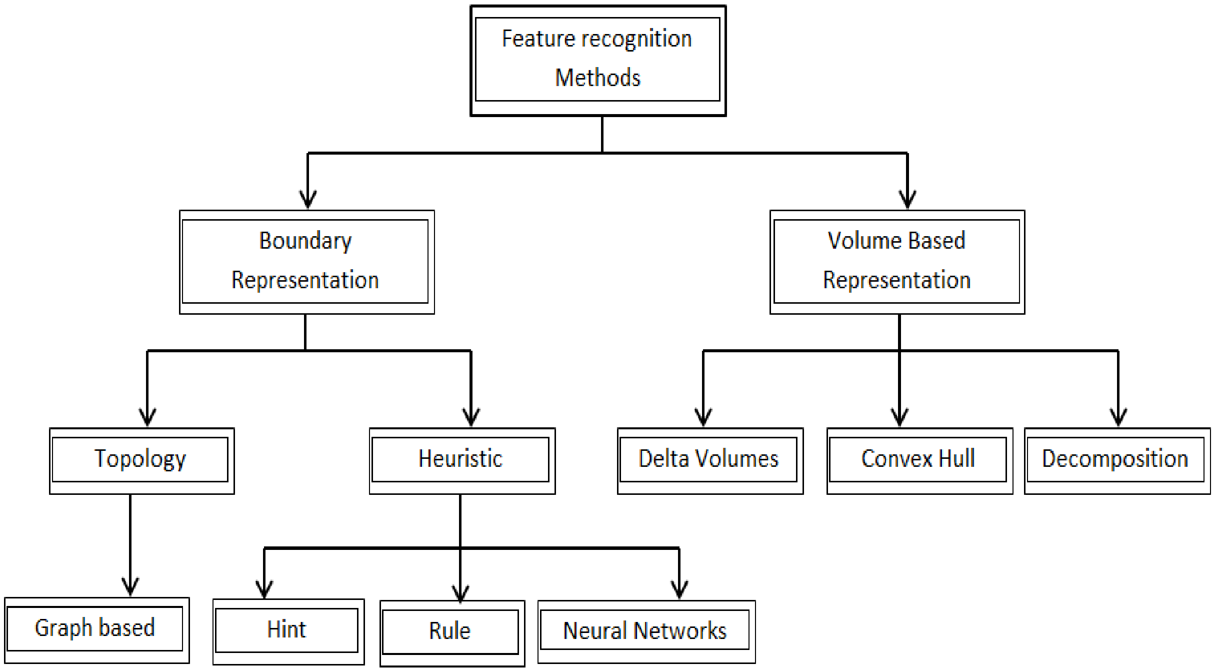


Figure 2.8 Taxonomy of Feature Recognition methods [5]

2.5.1 Rule based approach

Rule based approach determines some typical template patterns of features, which expresses the characteristic relationship between the entities such as faces, loops, edges and vertices. The characteristic relationships may be parallel, perpendicular, equality, adjacency, convexity and concavity which must be determined to indicate some particular patterns and constraints of features. Graph based approach is a graphical representation of the organized boundary entities and the relevant information of the CAD model with attributes. The features with identical topologies and different geometry can be easily recognized using this approach and it also describes the simplified boundary representation of the design model. This approach basically extracts features by applying graph manipulation and matching algorithms which consists of two types such as face-edge graph and edge-vertex graph. Hint based approach extracts the manufacturing features from the geometric patterns left in the nominal geometry using some hints of features. The incomplete patterns in a CAD model associated with features are said to be the feature hints. Artificial neural networks based approach initially describes the part model as a graph. Then secondly face adjacency matrix which acts as the input for the neural network is formed by encoding the above graph for pattern recognition. The most promising approach to recognize the various types of features by solving the interacting features is neural networks based approach.

2.5.2 Volumetric based approach

The volumetric based scheme represents a set of features based on volumetric operation by decomposing the input CAD model. Either volumetric representation or boundary representation models can be used as a input model in this scheme. The volumetric based scheme can be further classified as convex hull approach and decomposition approach. The convex hull approach is the method of decomposing the solid CAD model into several volumetric features by applying regularized Boolean operations and convex hull operation. The decomposition or volume growing approach adds the volumetric features by finding the hints of the feature from the CAD model such as group of faces from a model or loops from the part, and again converts these volumetric features in to part from the hints through volume enclosure operations. The process continues and it stops if no hints and features are identified from the part.

2.5.3 Hybrid scheme

The hybrid scheme integrates the design by features model and feature recognition system to extract the machining features. This process takes place in two steps, firstly the geometric model is developed by using the predefined features available in the database through interactive graphics system. Secondly the feature recognition system extracts the features by comparing the possible matches in the database with the feature used in the model.

2.6 Literature on Feature Recognition

Nasr [6] in 2006 developed an intelligent Feature Recognition Methodology for 3D prismatic parts created using CSG techniques. CSG models are built by combining primitives. If CSG is based on primitives that can be machined, then these primitives can be used for tool path creation. However, the currently popular solid modelling systems do not use CSG, Nasr resorts to algorithmically determining these CSG primitives. The input to the feature recognition system is a neutral file in IGES format which allows the algorithm to communicate with various CAD/CAM systems. Once the data has been imported, the information from the file is converted to manufacturing information. The features are recognized from the geometrical information based on a geometric reasoning algorithm. A feature can be declared based on its geometric relation with adjacent faces.

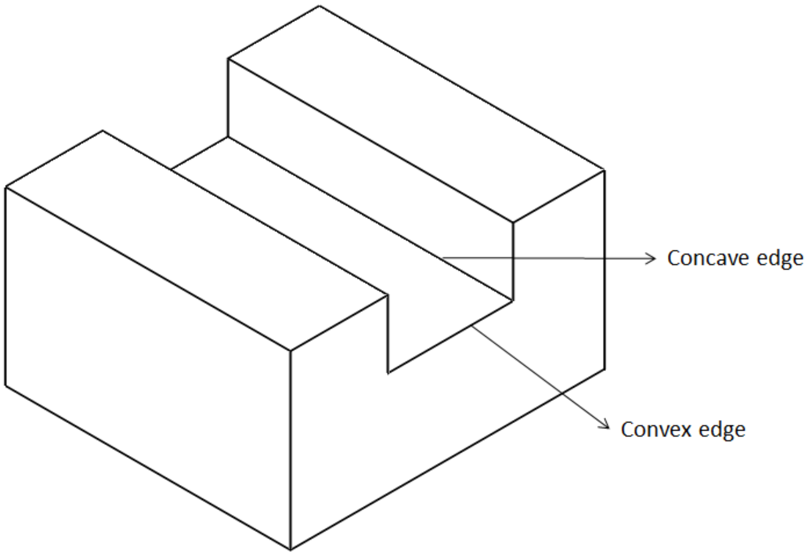


Figure 2.9 Recognition of slot

In this approach, the concavity of all edges in a face is related with adjacent faces in order to identify the type of the feature. For example, a slot feature can be identified if a face has two concave and two convex edges as shown in figure 2.9. The algorithm is mainly used to recognize simple features such as step, simple holes etc.

In order to integrate the design and manufacturing, the entity level description of the solid model must be converted into information for machining operation. Madurai [2] developed a rule-based system for automatic extraction and recognition of features for rotational part features. In that work, the input file used is in the format of IGES, from which the geometric and topological data are read by the feature extraction data compactor. Geometry to feature translator captures the manufacturing features in its decision logic which is expressed as rules. These rules are predefined in the algorithm and define the type of feature. For example, a hole is identified based on the predefined rule such as a convex edge having adjacent cylindrical face. Although their work demonstrates that geometric reasoning can be used to identify topologically similar shapes, it was limited to round parts and stopped at recognizing features.

A concept of recognition of machining features from 3D boundary representation model by building attributed adjacency graph was developed by Joshi [11]. This work was carried out by building a graph which shows the relation between the adjacent faces and according to the relation, a feature type is determined. The nodes in the graph represent faces, arcs represent edges. The features are identified by comparing the relation between these nodes and arcs. Their work was limited to polyhedral features such as pockets, slots, steps, blind slots and polyhedral holes. Kulkarni [24] has also proposed a graph-based approach to recognize features. An attributed face adjacency graph which consists of geometric and topological attributes is used to represent feature. Their work involves finding similar sub graphs from the part graph and evaluating those sub graphs to declare the type of the feature. The feature recognizer developed by them was able to address interactions between the features. And also it consists of parameterization module to extract user defined parameters from the recognized features. Their work has also described some important feature interactions such as multiple base and virtual interactions.

Fu [3] proposed an approach for recognizing design and machining features from a data exchanged part model. Their work facilitates feature identification and extraction based on relationship between part's geometry and topology entities. Their approach was able to identify intersecting features or compound features. Though their algorithm identifies various type of features such as step, slot, pocket, hole, etc., they are confined to features with plane base surface. Extraction of machining features with non-planar base surface is not discussed in this paper.

2.7 Tool path planning methods

A part is created by removing material from a large block with a rotating tool. The trajectory followed by the tool to remove material is called **tool path**. The tool path for a particular feature can be determined based on its geometry and topology coupled with some additional tool parameters. For example, consider the pocket feature shown in figure 2.10. This pocket can be machined in one or two passes using a flat end mill tool. The number of passes depends on the tool radius. However, larger tools exert higher forces and require larger machines. The larger tools also leave lot of material in the corners. In order to avoid this, the tool paths are normally planned with tools having smaller radius. The radius is determined by acceptable radius of the corners.

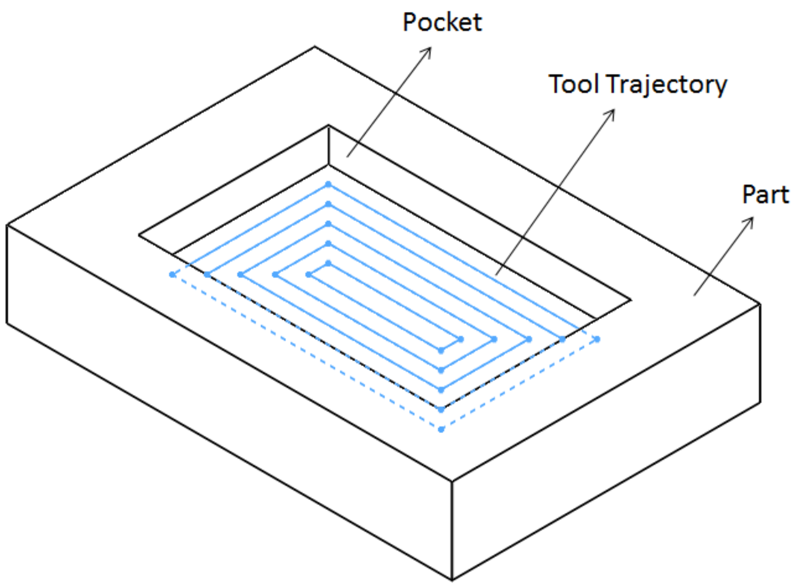


Figure 2.10 Tool path for machining pocket feature

In order to plan the tool path, the part is studied to determine number of orientation and its accessibility relative to tool axis. First, it should be analyzed whether all surfaces of the solid model can be machined by a tool rotating about a selected axis. If all features of the part cannot be accessed by a tool rotating about fixed axes, then the part may have to be re-oriented and rejigged to machine all the features. The alternate is to machine it on a 5-axis machine. Two main approaches are commonly used in planning a tool path once the part has been oriented relative to the tool axis and jigged on to the machine table. The tool path can be constructed using either a Cartesian or a parametric method.

2.7.1 Cartesian Method

The Cartesian co-ordinate system determines the machine tool movements in the orthogonal coordinate system used to define the part. The Cartesian co-ordinate system is specified relative to one point called the origin. Any other point is specified in terms of its distance from the origin along three orthogonal axis. The tool path is specified as trajectory of a point on the tool typically the center of key geometry on the cutter. The key geometry for a ball nosed cutter is the center of a sphere. The trajectory of this point is specified in terms of the destination point and interpolation scheme. The interpolation scheme is of three types: linear, circular and un-coordinated. Linear interpolation moves the key point from its current location to the destination along a straight line; the circular interpolation takes the key point along a circular path to the destination point; and in un-coordinated interpolation, the tool moves from the current point to the destination at full specified speed and with all axis moving independently. The tool path comprises of two distinct types of tool path segments which are discussed below.

Point to Point movements: The point to point path movement takes the tool from one activity at one point to the next point. The tool moves in the air as it moves from one point to next. Once it reaches the destination an operation is performed with cutting tool. This path generation system is used to locate accurately on the part specific features for operations such as drilling, reaming, boring and punching, etc.

Continuous path movements: This type of movement is used to machine surface features such as pockets, edges and faces etc.

2.7.2 Parametric method

The basic idea of the parametric-plane-based tool path generation is for the tool paths to be parallel straight lines on the parametric plane. The parametric plane-based method is the popular method for sculptured surface machining because the parametric surface data can be directly utilized in the tool path generation.

2.8 Tool path generation

Tool path generation begins with selection of geometric entities to be machined. The entities may define a slot, a boss, a patch, a hole etc. This task is typically done manually. Once the entities have been defined, a tool path strategy is selected. The tool path strategy describes the shape of the tool path. Some popular tool path strategies are zig-zag, to and fro, radial offset etc. Once the tool path strategy is defined, the tool is selected along with tool parameters such as emersion, RPM etc. Next the tool approach, tool entry and tool exit strategies are defined. Once these data have been choosen, the tool path to machine the specific entities can be produced. Similar tool paths are created for the grouping of entities until the whole part has been machined. In tool path generation, the tool is moved along an offset of the geometric entities. For some simple entities it is possible to ensure that the tool does not machines any entity other than the generating entities. Such unplanned machining of entities is called gouging. Gouge avoidance or detection is left up to the machinist in most cases.

To determine an efficient tool path, a small tool interval must be calculated which should be used as constant offset among each path. This makes easy to define constant isoparametric offset, thereby satisfying the surface accuracy. The tool path generation methods are classified into Cutter Contact-based method or CL-based method.

2.8.1 Cutter Contact-based method

The cutter contact based method generates tool path by dividing the part surface into a sequence of cutter contact points which is then converted in to the conventional tool path generation which is based on cutter location points. The cutter contact based method can be further classified as guide plane method, parametric method and drive surface method.

The *Guide plane method* initially generates tool path on a 2D plane in the form of either line or contours and then projects it on the design surface. The guide plane in 3-axis machining is the plane perpendicular to the tool axis. The main advantage of this method is that the feature shape or the part surface to be machined can be taken into account during tool path planning on the guide plane. The *Parametric method* generates the tool path by extracting the surface parameters at finite intervals and by sequencing the corresponding coordinates on the surface. Due to the non-uniform transformation between the Parametric and Euclidean spaces, the uniform parametric lines in the parametric domain results in non-uniformly spaced points on the surface which in turns results in non-uniform surface finish with varying scallop heights.

The *Drive surface method* generates tool paths by creating a series of planes along the surface, and by identifying the intersection between the planes and surface. The intersection of surface and planes result in intersecting curve, which are used to generate tool paths. This method which creates parallel intersecting planes for machining is known as iso-planar machining. This method can handle complex surfaces easily and is robust and reliable.

2.8.2 Cutter Location method

The cutter location methods generate tool paths, by creating a cutter location surface relative to the design surface. The cutter location surface is used to generate the tool path segments which are connected to build the tool path. The cutter location surface is created by offsetting the surface with an interval distance typically equal to the tool radius.

2.8.3 Offset Surface method

The offset surface method generates tool path by adopting offset on design surface and also to avoid the accuracy of surface finish. It is divided into two steps as follows,

a. Discretization of the surface

This method will help to take certain resolution on design surface by assigning z-coordinate values along with x, y co-ordinates. Under varying surface, the intersecting plane is used to get initial curves on surface and these intersecting planes must be vertical and parallel to each other at constant arc length.

b. Inverse tool offset

The inverse tool offset is used to generate the offset surface and it helps in inverse of tool in Z-direction, with the center of inverted tool on design surface. This tool will provide center location on every discretized point so the tool is tangent to the surface.

2.9 Literature on Tool path Planning

Hwang [34], presented a method for generating interference free tool paths from parametric compound surfaces. This method was able to obtain tool paths from a surface model in a short time. In this method tool paths are generated in two steps, first points are obtained from a compound surface by converting it into a triangular polyhedron from which tool paths are generated. In that algorithm, an efficient method was used in the calculation of cutter location data and planar tool paths to make it suitable for metal cutting. Hatna [33] proposed that the tool paths can be also generated by offsetting the 3D constant Z-height contours on parametric surfaces. This method allows the iterative calculation of interference free 3D offset contours and it is independent of reference frame used to define the surface. The main elements of that method are the iterative offset of loops and parametric segments, with the corresponding trimming and connection process. As discussed earlier, in 3-axis sculptured surface machining, the most commonly used cutter is ball end mill cutter, whereas the flat and filleted end mill is less frequently used. A method to generate effective tool path for all these cutter types is presented by Hwang [30].

Patel [43] developed an automatic web-based tool path planning for machining sculptured surfaces as are found in wooden plaques using 3-Axis CNC machines. This work has integrated a web-based CAD system with web-based tool path planning system to automatically generate tool paths using optimal cutter with desired tolerances. In his method, the tool path planning was divided into cutter contact path and tool positioning. The tool path foot print is the path described in a plane perpendicular to tool axis along which the projection of the tool moves as shown in figure 2.11. It is developed by taking a projection of the CAD model on the foot print surface. After generating foot print, it is discretized into several equally spaced points and cutter position at each point is found. The maximum depth at which the cutter can penetrate without gouging the surface is calculated. The tool path is constructed by moving the tool linearly between two consecutive points on the tool path foot print. His work mainly focused on gouge-free cutter positions for all points along the foot print.

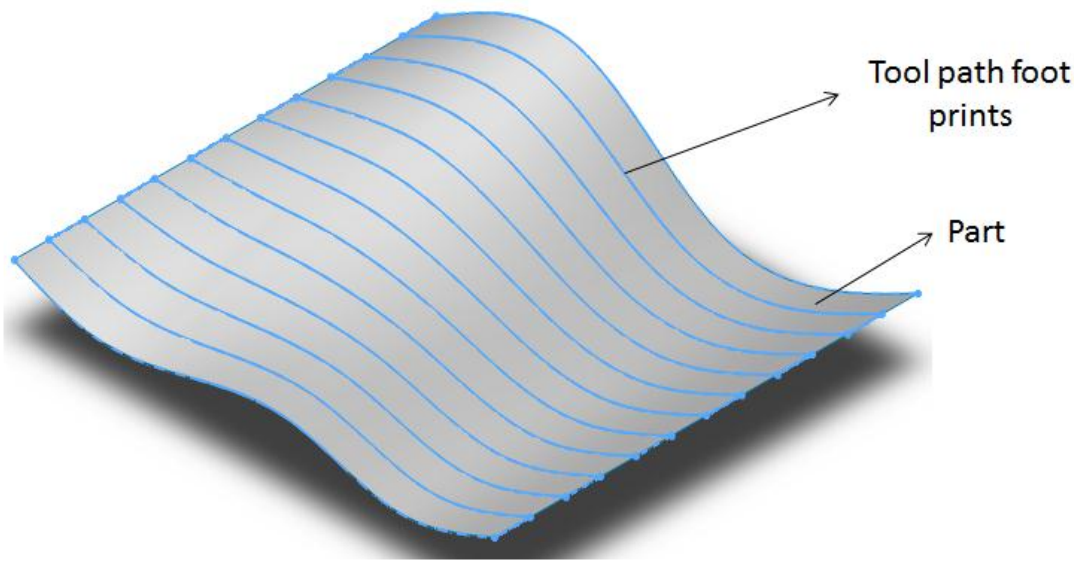


Figure 2.11 Cutter contact path

Though the tool paths for various features are generated, it’s necessary to plan the sequence of machining process to be followed by the tool. The study made by Kayacan [13] proposes an approach of process planning of prismatic parts. The feature recognition process is achieved with the B-rep modelling method in order to identify the vector direction and adjacent surface relationships using STEP interface. The database achieved from the feature recognition module is used to define the operation type and sequence of operation for prismatic parts. The work proposed by Han [17] also integrates the two activities such as feature recognition and process planning. That work presents the efforts done towards feature recognition for manufacturability and setup minimization, feature dependency construction and generation of an optimal feature based machining sequence.

Mawussi [42] proposed a general approach for generating machining process based on machining knowledge in which the CAD model of complex forging die is decomposed in geometrical features. In this work, the machining features are created by aggregating technological data and topological relations to a geometrical feature. After machining features have been identified, a machining process model is proposed to formalize the links between information imbedded in the machining features and the parameters of cutting tools and machining strategies. Finally all the machining sequences are grouped in order to generate complete die machining process.

Discussion

From the above brief review, it becomes apparent that most of the studies relied on theoretical examples of the proposed methodology. Though some works has been implemented in software, these studies are focused either on feature recognition from CAD models or on process planning or on tool path generation. No study integrates all these three process due to the following problems.

1. The first problem behind the difficulty in this integration is recognizing interacting or compound feature from the part design. As features interact their topology changes and makes it harder to recognize the resulting geometry. Fu [3] proposed an approach for recognizing compound features, but stopped short of integration with machining operations.
2. Secondly, planning tool path for compound features is a challenging task. Because, when two or more features interact each other, the chances of tool collision or gouging with the part are high. In order to avoid this tool gouge, planning a gouge-free tool path is necessary. As discussed earlier, Hatna [33] and few other researchers have proposed methods for planning gouge or interference free tool path using 3D-offset contours. Still, the integration of feature recognition or process planning with tool path planning has not been their focus.
3. Thirdly, though the gouge-free tool path for each feature is planned, it is difficult to produce an efficient machining operation without sequencing the activities into a process plan that transforms the raw stock into final product through various machining sequence. Though some researchers like Kayacan [13] and Han [17] proposed various process planning techniques for prismatic parts along with feature recognition, the recognition of compound features and tool path planning has not been addressed. Development of a tool path for compound features is still an open area for research.

Taking the above discussion into consideration and keeping the techniques developed by the above mentioned researchers in mind, a modified approach that combines both feature recognition and tool path generation with process planning for all types of features was developed and verified experimentally. The proposed work combines automatic feature recognition, tool path generation and associated planning of machining sequence. This combination successfully links engineering design and shop floor manufacturing. Machining tests conducted demonstrate the efficiency of the proposed feature based machining.

As discussed above, in this work the integration of automatic feature recognition, tool path generation and process planning is achieved by combining them together. Feature recognition module identifies all entities that form a specific topology and groups them together for tool path planning and process planning. All the features in the part design are recognized based on its geometry and topology relationships, and from these features parameters for machining are extracted by the feature recognition module. This information acts as an input for tool path planning module. The task of the tool path planning module is to generate tool paths for this group of identified entities in the feature. Typically tool path generation involves identifying the trajectory of the tool center required to maintain a cutter contact on the identified entities. This is achieved using offsets and surface normal. In this step of tool path planning the trajectory of the cutter contact for a particular feature is algorithmically embedded in the feature. This algorithm takes the cutter contact and the tool radius and offsets it by the tool radius in the appropriate direction to produce the trajectory of the cutter center. This trajectory only ensures that the tool contacts the feature surfaces at only one point. In typical parts, simple features combine together to form complex features. When such an interaction happens the tool path planning algorithm cannot ensure that there is only one contact point between tool and feature surfaces. In this case gouging can occur and remove wanted material from the part. Gouging must be avoided at any cost. Thus to ensure gouge free tool paths and add the ability to deal with complex features, it is proposed to break tool path planning into two phases, namely, Cutter Contact Path and Cutter Location Path. The Cutter Contact Path is determined first for all features. The Cutter Contact Paths are discretized into closely points and the tool is positioned individually at each point using a gouge-free tool positioning method. The tool path trajectory is then obtained by moving the tool linearly between neighboring points. Although the discretization can result in chamfering of some edges, this can be avoided algorithmically or minimized by reducing the spacing between points. It is the hypothesis of this work, thus breaking up tool path planning into tool path footprint and gouge free tool positioning and algorithmically integrating it with feature recognition and process planning can be used to automatically machine a part with simple and compound features.

To verify this hypothesis a rule based feature recognition system was developed and is discussed in chapter 3. This feature recognition system was then integrated with a tool path generation/planning and process planning modules. This tool path generator/planner and process planner is discussed in chapter 4.

Chapter 3

Feature Recognition system

Feature recognition links the important steps of process planning and computer aided manufacturing. Features describe the geometric and topological relationship between entities and can also be used to store semantics and manufacturing process details. Thus features can have multiple roles that depend on the application, for example, from the view of functionality and manufacturing. In this work, features refer to the manufacturing features, which can be determined as volume to be removed in order to obtain the final product as described in CAD model using a machining process. The relationship between part entities and features can be established in two ways. In the first method, as discussed in chapter 1, the part is built from features. In this case the relationship of entities and features is known and one can proceed directly to tool path generation and planning as discussed in chapter 4. This method is called Feature Based Design (FBD). In the second method the part is generated independent of feature definition. In this case the part entities are analyzed for geometric and topological relationships. When all geometric and topological relations match a definition, the entities are labeled as a feature. The process continues until all entities have been labeled. This step is called Feature Recognition. FBD works well when the part is built from well identifiable features. However, when features interact the machining method of one feature may result in the machining of another feature. This is not desirable. Feature recognition depends on the definition of features. A large library can result in identifying a large number of features, but increase complexity of feature definition. Most Feature Recognition systems, cited in literature are based on simple features. These suffer from the same problem as a FBD system. When features interact and the interaction can be detected, the machining method for one feature can gouge the other feature. This inability of current features is addressed in this work by dividing tool path planning into cutter contact path and cutter location path. Cutter contact paths are first developed based on features recognition and tool positioning with gouge detection and avoidance is done on the whole part, thereby avoiding gouging on portion of the part. A Feature Recognition system was developed to prove the hypothesis. The Feature Recognition system is based on the work of Joshi [11], but has been modified to separate Tool path planning into two parts, namely cutter contact path and cutter location path. Recognizing these manufacturing features is the initial task in the process of linking features, tool path generation and process planning. The following topics discuss the methods and steps involved in recognizing the manufacturing features in this work.

3.1 Feature extraction from CAD model

In this work, a rule based method for solids represented using Boundary Representation schemes is used to recognize features. In this rule based approach features are computed from geometric entities like lines, planes, circles, etc., and these entities are connected in a specific topology such as loops etc. As the geometry and topology of a part are central to features, the method of defining geometry and topology in a solid model is described next.

A solid is represented by its surfaces in the B-Rep format. Each solid is composed of a linked list of faces. Each face in turn is defined by the edges. Edges that define the outer boundary of the face make up a loop. This sequenced loop of outer boundary edges is called the outer loop. All faces have one outer boundary loop. Similarly a face can have inner edge loops as well. A loop of edge that defines cutouts in a face, when sequenced, forms the inner loops. A face can have many inner loops. Figure 3.1 shows a face that has one outer loop and three inner loops. Each edge in turn is made of two vertices, a start vertex and end vertex. The linked lists allow the traversal of the data structure. It is easy to write algorithms to find neighboring faces, edges and classification of points, lines and entities. This base algorithm makes B-Rep ideal for feature recognition. The geometrical information embedded in Boundary-representation of the solid model is analyzed by a Feature Recognition algorithm based on predefined rules. The proposed Feature Recognition Algorithm is able to recognize the following features: slots, pockets, holes, steps, etc.

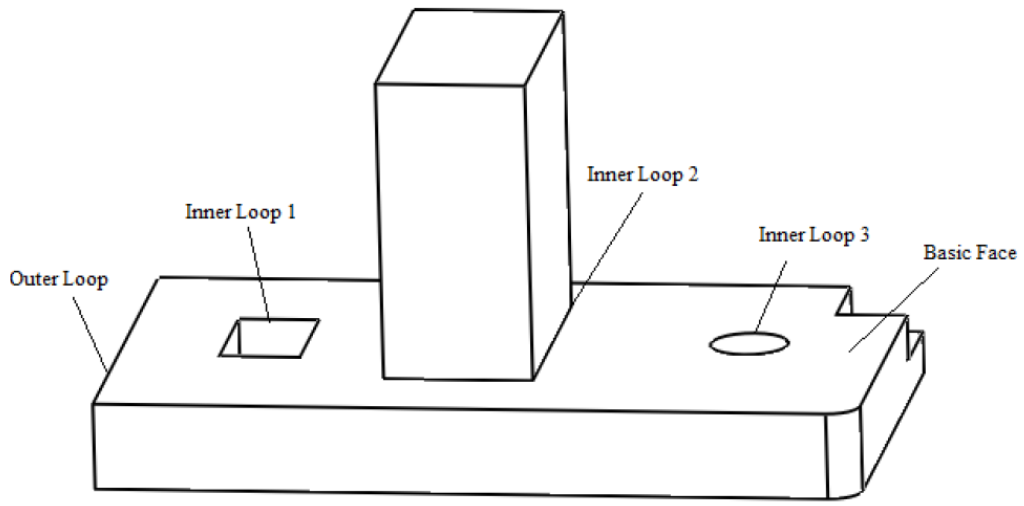


Figure 3.1 Loop classification

3.1.1 Concept behind Feature Recognition

A typical Feature Recognition system is based on geometry and topology of the part. In this work, the Feature Recognition system also considers the tool path planning during the feature recognition stage. The process of machining a part is accomplished by moving a tool along a predetermined path through a stock of raw material. The material removed by the moving tool is called machining volume. Most parts will require a variety of tools to machine the entire part. Each tool will describe its own machine volume. Further one tool may be used to remove material in one portion of the part and may also be used to remove material from another portion of the part. These removed portions describe two independent machining volumes. Process planning is a method of sequencing the machining volumes. The first machining volume is removed initially. It exposes other machining volumes which can be removed subsequently.

Features in this work merge Boundary representation with process planning and tool path generation. In other words, features can be said to merge machining volumes with Boundary representation. The faces of B-Rep model may be of any type of surfaces like ruled surface, plane, spline surface, etc. Again, the edge can be of any type like line, circular arc, conic arc, etc. A loop formed by the edges is the key factor to determine the existence of machining volume in a B-Rep model. The main machining volume on a face was represented by its outer loop and the location of protrusion or depression volumes is represented by its inner loop. Thus the number of features or machining volumes which lies on a common face can be determined from the number of inner loops of the face. A face can be a part of two machining volumes but an edge is part of only one machining volume. This can be visualized in the figure 3.2. Figure 3.2 show two features a circular boss and a rectangular cavity that intersect each other. The three machining volume are identified sequentially as shown in figure 3.2 (b),

(c) and (d). In the figure 3.2(c) the circular face is part of two machining volumes. All edges belong to distinct machine volume.

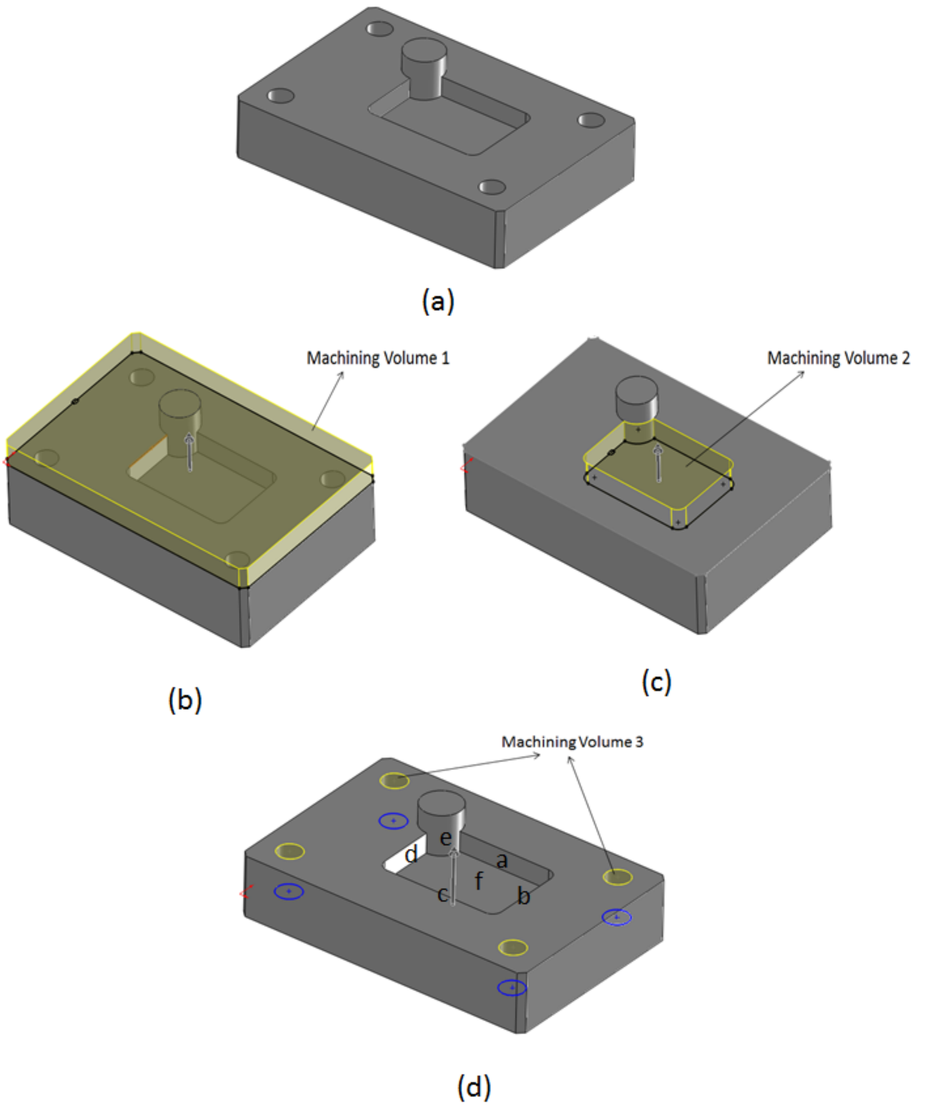


Figure 3.2 (a) Part model (b) First machining volume (c) Second machining volume (d) Third machining volume

The relationship between the B-Rep model and the machining volumes is the key to Feature Recognition. Consider the part shown in figure 3.2(a). This part is machined using the machining volumes which are sequenced in order to make a process plan as shown in figure 3.2(b), (c) and (d). The machining volumes share face and edge with the B-Rep model. For example, in figure 3.2(d) faces labeled as a, b, c, d, e and f is common to the part and the machining volume of that base face. These shared faces and edges represent machining feature or form feature. Features are incomplete solids. This boundary i.e. collection of edge with only one neighbor face, form a loop that belongs to the base face. The concept of base face has been introduced to integrate Feature Recognition with part machining and is discussed later. The base face forms the bottom of the machining volume. When the entire base faces of all features in a part is protruded in the direction of tool axis, the volumes formed by each feature do not intersect each other. This strategy is the key for planning gouge-free tool path for un-occluded base faces and is discussed in next chapter.

3.1.2 Classification for Feature Recognition

Form features can be categorized into **interior form feature** and **exterior form feature** based on the attributes of its geometric entities and topological relationship. The parent face which contains the bounding loop of the features in it is defined as the base face. The feature which lies inside the base face are said to be Interior form features. The base surface combined with its adjacent faces forms a feature which is said to be exterior form features. A loop is used as a basic indication to extract the interior and exterior form features. The interior form features, can be further divided into two categories, convex interior form feature and concave interior form feature. The convex block in the center of the part shown in figure 3.3 is a convex interior form feature for the base face 1. If the top most faces is considered, the two faces which lies in the same plane is the base face 2 and the slot form an exterior form feature for this base face 2. The concave geometric sections in face 1 are interior concave form feature for base face 1 and are shown in Figure 3.3. A base face containing a blind cylindrical hole or a pocket is also an interior concave form feature.

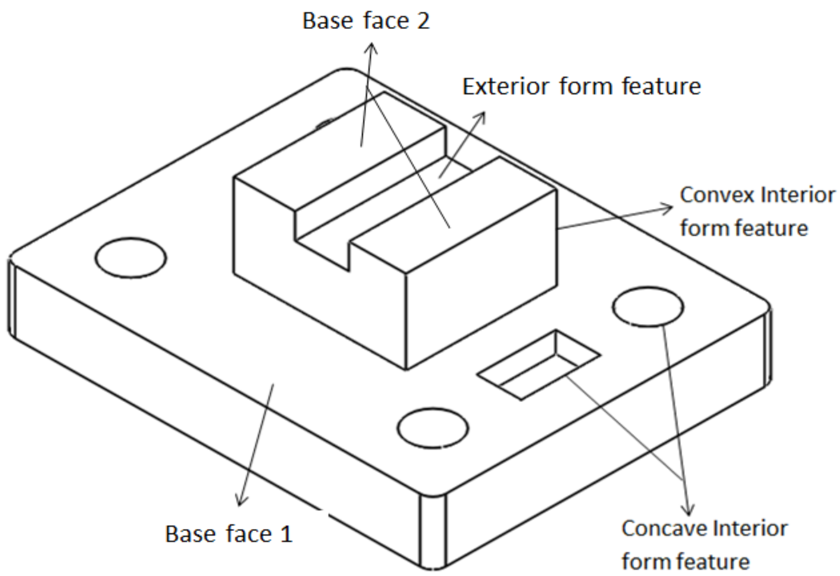


Figure 3.3 Classification of form features

In a solid model, start vertex and end vertex determine an edge and these are expressed in terms of three dimensional coordinates. By using several parameters related to the edge and by performing concavity test on the normal of faces separated by the edge, an edge can be declared as concave or convex edge. A concave feature is determined by identifying a loop in which all edges separate concave faces.

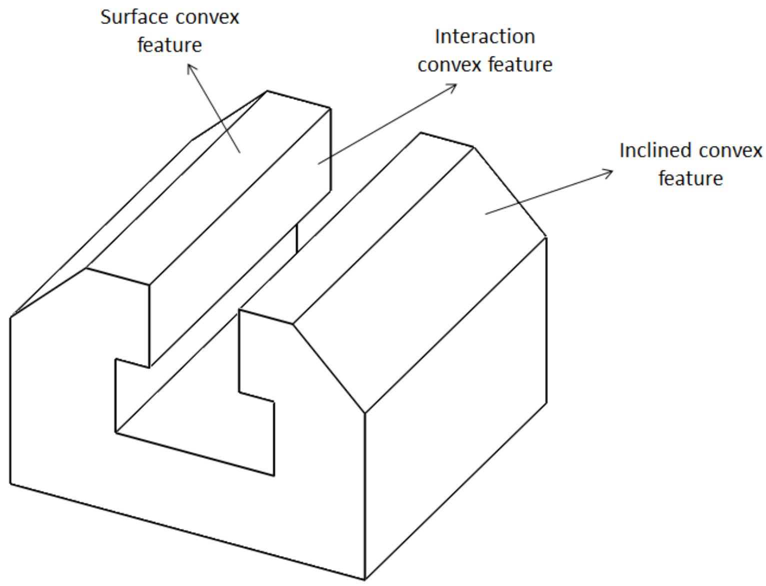


Figure 3.4 Types of convex features

Similarly, a convex feature is determined by a loop in which all edges separate convex adjacent faces. There are several types of convex features such as inclined, interaction or surface as shown in figure 3.4. The group of convex faces which are not parallel or perpendicular to the smallest surrounded envelope of the designed object are inclined convex features. When two or more features interacts each other, it is said to be interaction convex features. The features that are located on the exterior enclosing envelop are surface convex features.

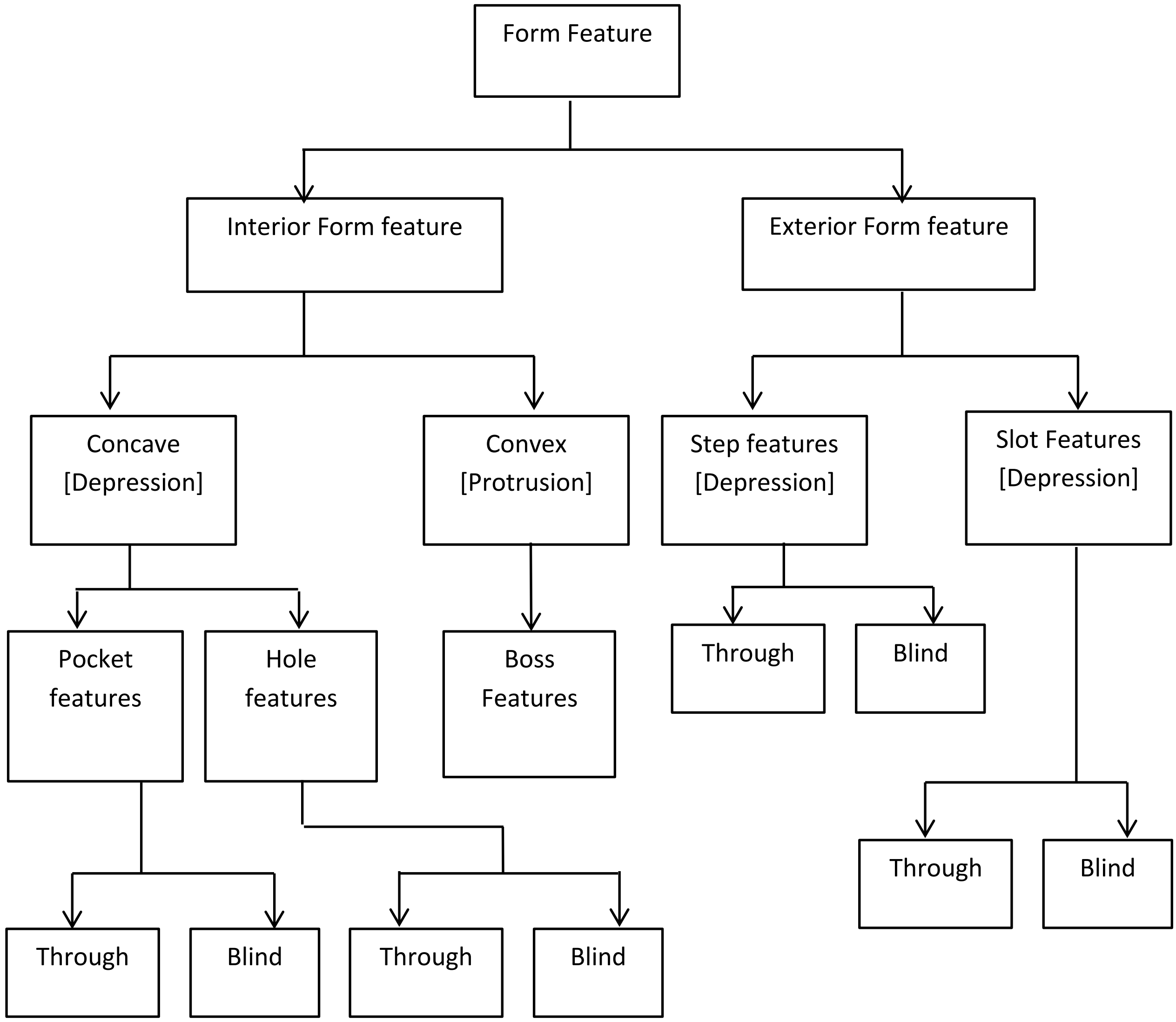


Figure 3.5 Hierarchy of form features

The Feature Recognition process begins by classifying all the edge entities into three categories, namely, concave, convex and tangent. This edge classification is achieved by processing through each edge in a loop. The relationship between adjacent faces is identified to define the type of edge. In the next step all the face entities are processed similarly and classified into four categories, namely concave, convex, and tangent and inclined. As mentioned earlier, a face can be a part of two features, but its edges are part of distinct feature. A face is classified with respect to each edge. This classification helps to find the type of a feature (i.e. protrusion or depression). Once all the faces and edges have been classified and categorized, the process of Feature Recognition is reduced to identifying a selection of faces and edges that share a specified topological relationship as expressed by their classification and categorization.

3.2 Entities

The geometric and topological relations are the source for determining the geometric characteristics which helps with recognition of features. There are two types of entities namely, face and edge.

1 Edge entities

2 Face entities

3.2.1 Edge Entities

Edges in the B-Rep method separates two neighboring faces. Edges are connected together to form a loop. These loops can define outer boundary of a face or inner protrusions within the face. The loops are the seed for features that are associated with the given face. As loops comprise of edges, each edge is first classified as concave, convex and tangent. This classification is done by performing concavity test on each edge in a loop.

Concavity test for Edges

To perform the concavity test on an edge, the angle between the two adjacent faces separated by the edge is found. This angle is the factor used to classify the edge as concave edge or convex edge. To calculate the angle between the adjacent faces, the normal vectors of the two faces are used in the connectivity test to determine the edge classification. The angle between the two faces is determined by the following formula:

$$\textbf{Angle between two faces, } \theta = \cos^{-1}\left(\frac{i_1 i_2 + j_1 j_2 + k_1 k_2}{\sqrt{i_1^2 + j_1^2 + k_1^2}\sqrt{i_2^2 + j_2^2 + k_2^2}}\right) \mathrm{x}\ \frac{\pi}{180} \qquad (3.1)$$

Where, $(i_1\, j_1\, k_1)$ = Components of the normal vector of the first face.

$(i_2\, j_2\, k_2)$ = Components of the normal vector of the second face.

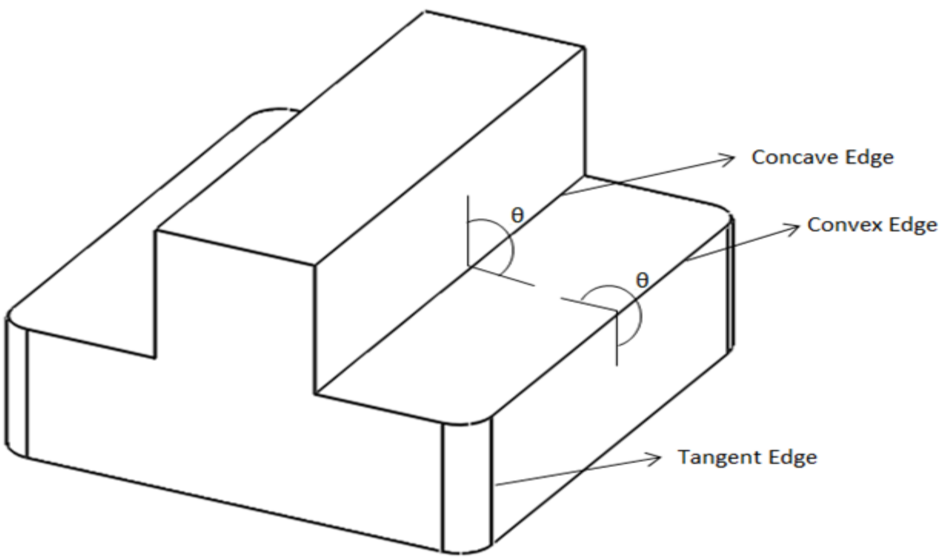


Figure 3.6 Types of edges

If the angle, θ is greater than 180, then the edge can be classified as convex edge and if it is less than 180, then the edge can be classified as concave edge. Both concave edge and convex edge can be classified as sharp or smooth edges based on the curvature of the adjacent surfaces as shown in figure 3.6. If the two adjacent faces are perfect planar surfaces whose normal vector is constant all over the face, then the edge connecting these two faces are defined as sharp edge. If one of the adjacent face is non- planar whose normal vector is not constant, then the edge connecting these two faces are defined as smooth or tangent edge. These attributes are said to be blend surfaces.

Figure 3.7 illustrates the algorithm to find the concavity of an edge entity with respect to its adjacent faces. An edge has two vertices which are used to determine the length of the edge. For example, the edge with start vertex (X_1, Y_1, Z_1) and end vertex (X_2, Y_2, Z_2), will have a length that can be calculated using the following formula,

Length of the line = $\sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2 + (Z_1 - Z_2)^2}$ (3.2)

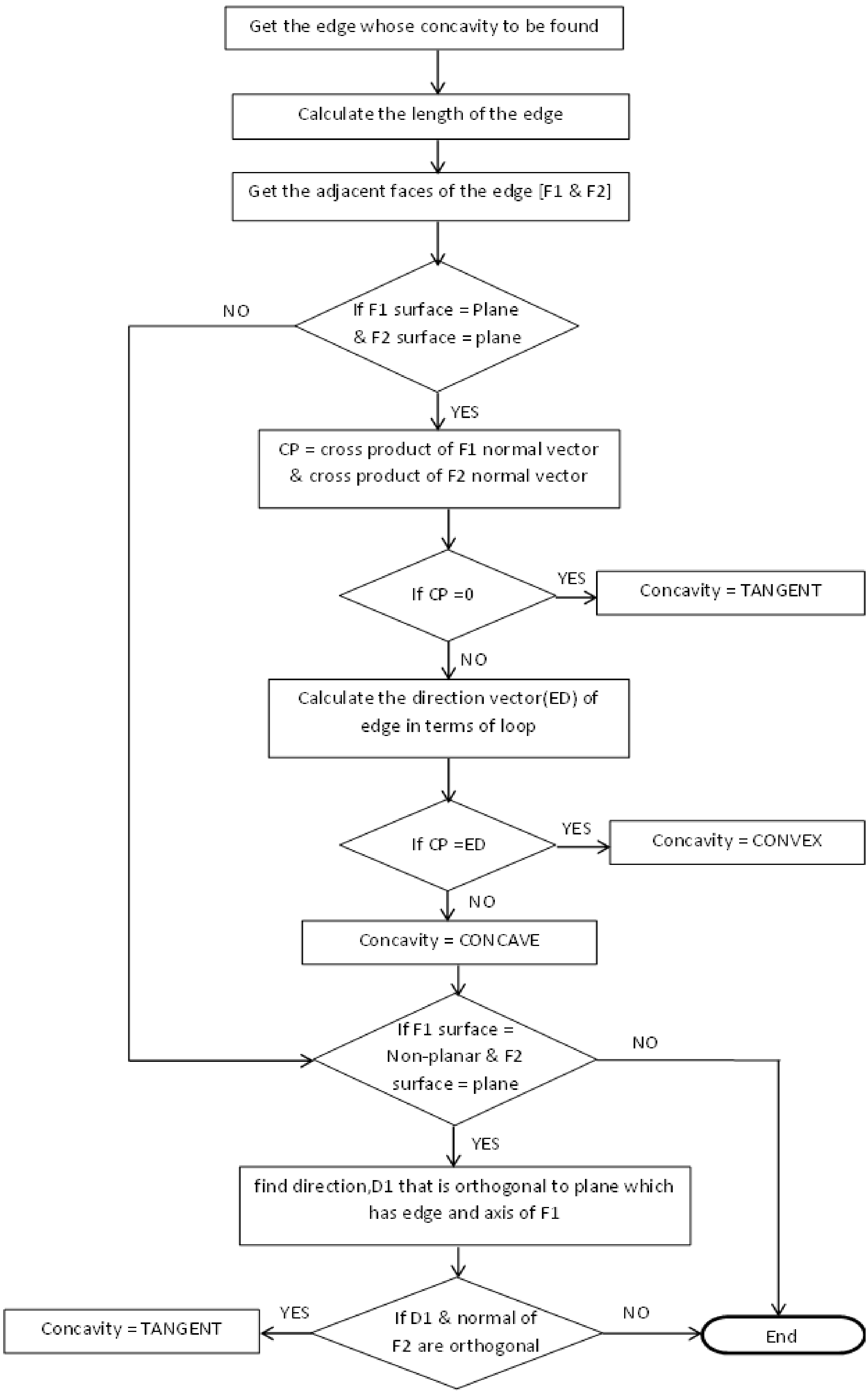


Figure 3.7 Algorithm for concavity test

This method of classification works for planar faces and straight edges. For non-planar faces and non-linear edges such as circle, arc, etc., the normal vector is not constant and it varies along the surface. In such case, the angle of axis is used to determine the type of the edge.

Angle of Axis

Consider a hole on a planar face as shown in figure 3.8, the edge connecting the cylindrical face and the planar face is a circular convex edge. In this case, the normal vector of the cylindrical face is not constant and it varies along the face. Since the concavity test depends on the components of normal vector, it is difficult to apply in this case. However, the axis of the hole remains constant. It can be used to compare with the normal vector of the planar face in order to classify the circular edge.

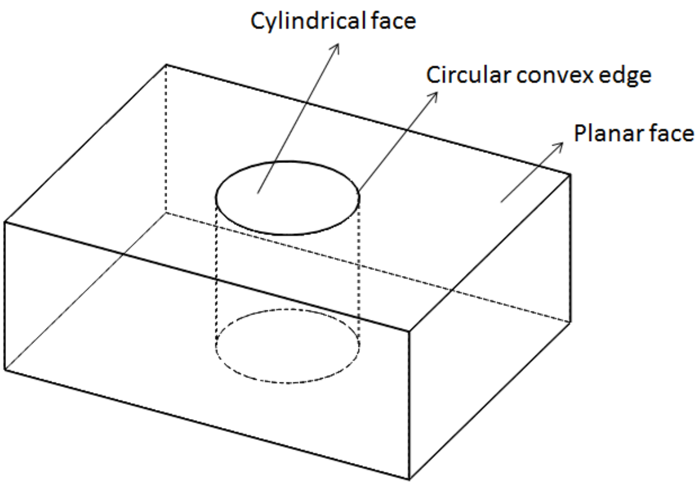


Figure 3.8 Cylindrical hole feature on a planar face

In some cases, if the angle varies constantly along the edge, then the angle of the edge cannot be defined. For example, the hole feature on another cylindrical surface has round edge whose angle varies constantly as shown in figure 3.9. These types of edges cannot be classified and special method such as the one for cylindrical faces must be devised.

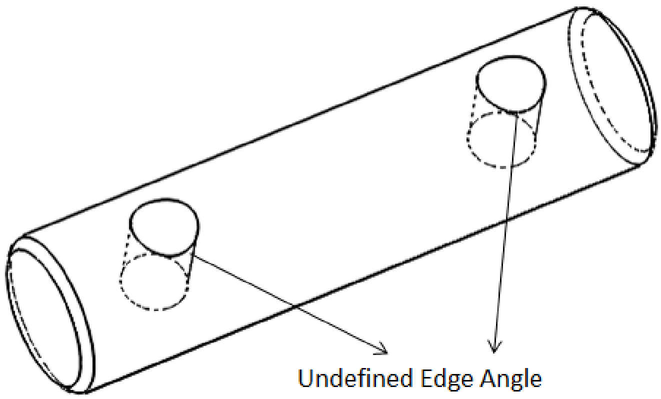


Figure 3.9 Cylindrical hole feature on a cylindrical face

3.2.2 Face Entities

Faces in a B-Rep model describe the bounding volume of the part. The faces are divided into following types, namely, base and side faces, co-defined and coaxial faces, planar and non-planar based on the type of surface.

Base and Side Faces

In this work, the base face is defined from a machining perspective. A face that is perpendicular or inclined at an angle less than 90° to the tool axis is defined as a base face. A base face is a prime candidate for defining the trajectory of the tool as it removes material from the machining feature. The side faces are those which lies parallel to the tool axis, and can be reached by the edge of the tool. Figure 3.10 shows a simple solid model with various base and side faces. The base face lies within a feature and is identified after the feature has been recognized. The base face is used to define the tool path foot print and to develop the tool path.

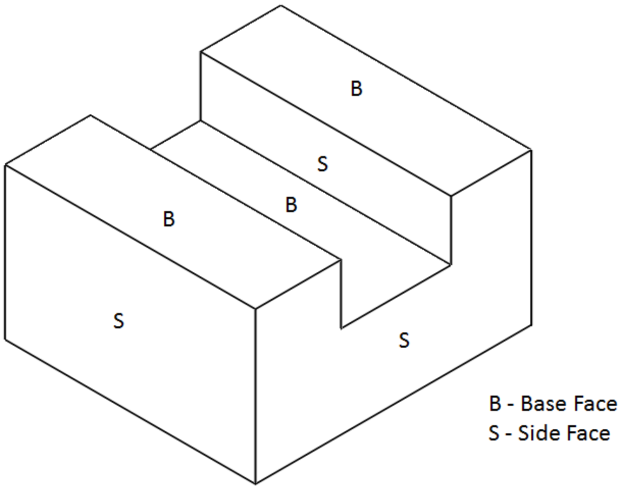


Figure 3.10 Base and Side faces

Co-defined and Coaxial faces

This entity shows the relation between various faces that have similar geometrical characteristics. The co-defined faces have the same underlying geometry but are split topologically. If the co-defined faces entities are also base faces then they are linked to define one base face.

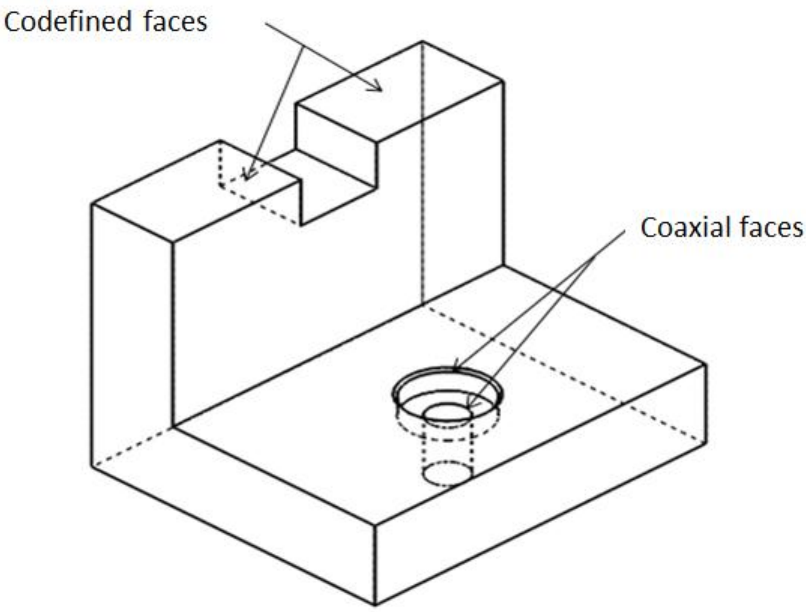


Figure 3.11 Co-defined and Coaxial faces

The attribute which express the relation between two faces sharing same axis is the coaxial faces. The axial symmetric features like countersunk and counter bore holes have a coaxial relation which can be easily represented by using this entity. Identifying co-defined and coaxial faces helps in fast feature recognition and tool path planning and saves time.

Type of surface

The important geometric characteristics of a face are its surface type, which can be planar or non-planar. Though the actual geometry of the surface is the same, sometimes the surface type can be different. The surface should be recognized based on the geometric invariant properties of the surface. For example, if the surface is planar, the surface parameters will be constant all over the face, which makes the tool path generation process as an easy task. Whereas if a surface is non-planar, the surface parameters will not be constant, this makes tool path generation process as a challenging task. In this case, the tool path should be generated according to the varying surface parameters. So, it is necessary to identify the type of the surface before proceeding with tool path generation.

3.3 Feature Recognition

The proposed algorithm uses rule based approach to identify machining features in the part. The Feature Recognition module builds two lists namely, the Prime face list and the Feature list. The Prime face list contains a list of all faces in a part. The Feature list contains all the information about the identified features such as type of feature, list of faces associated with a feature, topological relationship between the faces and type of surface. The features recognized by the Feature Recognition module are first classified into four major categories as Protrusion, depression, slot and step features. The features such as boss, islands etc. are considered as subcategories of Protrusion features. The features such as pockets, simple holes, tapered holes, counter-bore holes, and counter-sunk holes etc. are considered as subcategories of Depression features.

Loops are the key to features. Consider a protrusion as shown in figure 3.12, the protrusion is connected to the rest of the part along a loop of edges that belong to a base face. Similarly the depression is connected to the rest of the part along a loop of edges. Each loop on a face indicates the existence of a feature. However, there are some features that are not connected to the inner loop. For example, a step features as shown in figure 3.12 is connected to the outer loop of the faces, such features can be identified based on topological classification of the face and are discussed later.

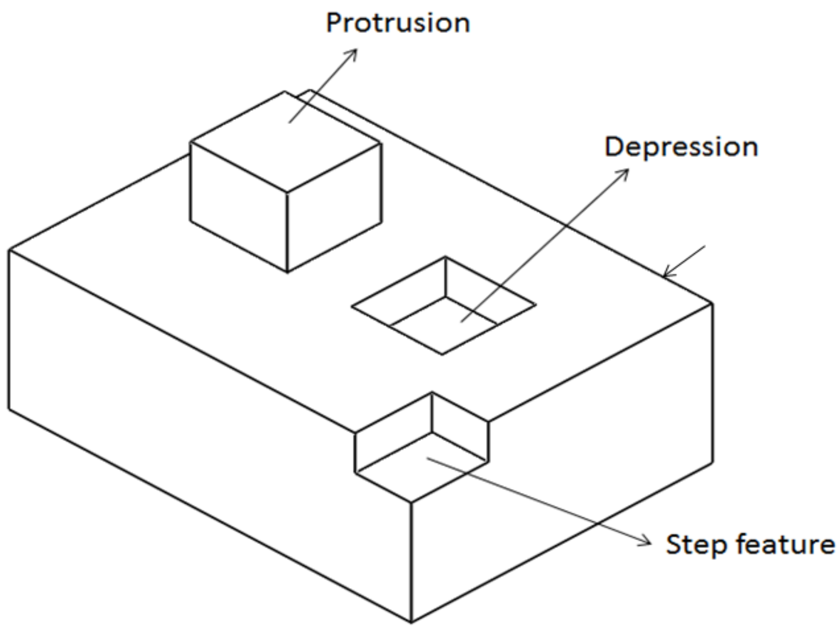


Figure 3.12 Classifications of features

The Feature Recognition starts with edge and face classification as mentioned earlier. Feature recognition module begins its task by scanning all the faces in the part and classifies its edges and type of the surface. The faces of the part are extracted and the Prime face list is generated. The first face in the prime face list is selected and it is checked for existence of inner loops because as explained earlier each inner loop indicates the existence of a feature. If there are no inner loops in the selected face or after processing the inner loops, the next face in the prime face list is selected. If a face contains an inner loop, then the loop is processed to identify the feature it generates. The edges in the loop are examined based on the classifications (edge and face) done earlier. This examination helps to determine the type of the feature such as protrusion or depression. For example, if all the edges in an inner loop are concave edges, the resulting feature is considered as protrusion. Similarly, if all the edges in an inner loop are convex edges, then the resulting feature is considered as a depression feature. If the inner loop has a mixture of concave and convex edges it is considered as an interaction between a protruding feature and a depression feature and is considered under interacting features. All the faces associated with the inner loop are extracted and grouped together to form a feature. These faces are then removed from the prime face list and added into the feature list under respective feature categories.

In some cases, a feature may exist within another feature. Such features are defined as sub-features. The faces removed from the prime face list are processed and checked for their inner loops and outer loops in order to identify the sub-feature. This check is similar to the one described above. Since features can be embedded within features, this process continues recursively until all the features and sub-features on features etc. have been identified. The structure of the identified features is then compared with the predefined rules to identify the type of feature described by the faces as discussed later.

Once the Feature list is populated with features built from the inner loops of all faces, the remaining faces in the prime list are selected and checked for outer loop features. The outer loops in a face may describe features like slots, steps etc. Consider an outer boundary of a face that has all convex edges, such a face does not form any feature. But existence of one or more concave edges on the outer loop of face indicates the existence of feature as shown in the figure 3.13.

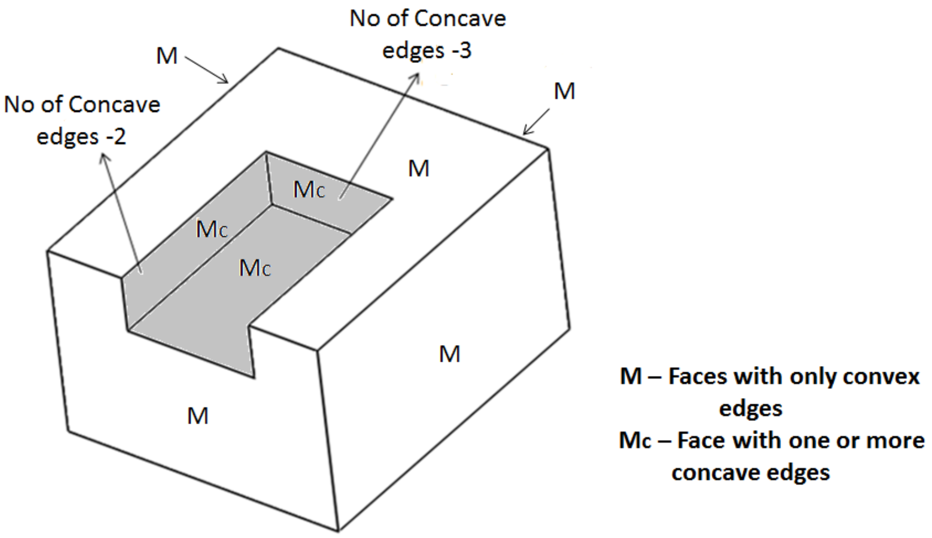
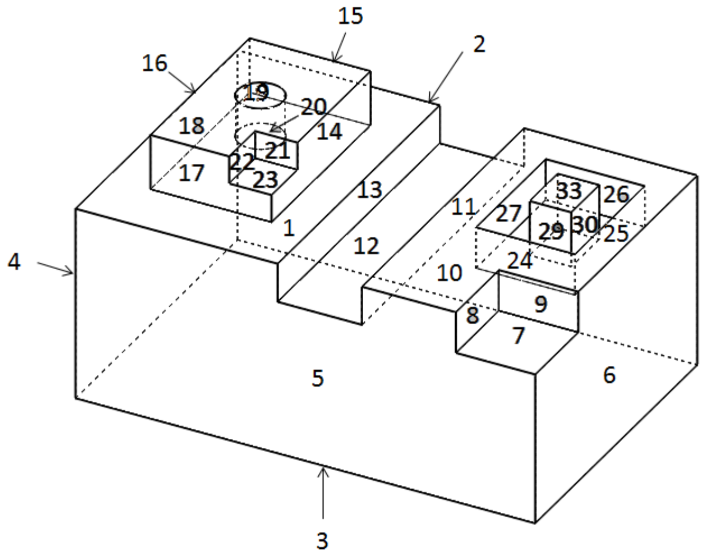


Figure 3.13 Main faces with concave and convex edges

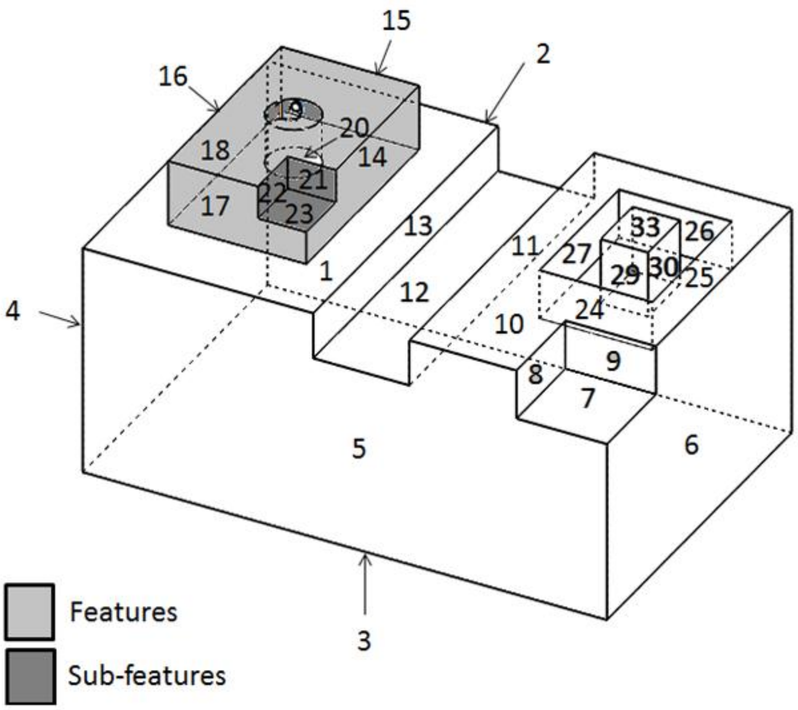
Each edge in the outer loops is checked for existence of concave edge. The adjacent faces of the concave edges are grouped together to form a feature and all the information such as angle between the faces, type of the surface etc. is extracted. All the extracted faces are added to the Feature list and the structure of the feature is compared with the predefined rules in order to identify the type of feature. The prime face list still contains some faces such as the faces which do not form any feature. Such faces most commonly include the side and bottom faces of the part. For example, figure 3.14 shows a part with many features in it and its correponding Prime face list with list of all faces. The following figures illustrates the various steps involved in Feature Recognition and its corresponding Feature list at various stages.



Prime Face List	
Face 1	Face 18
Face 2	Face 19
Face 3	Face 20
Face 4	Face 21
Face 5	Face 22
Face 6	Face 23
Face 7	Face 24
Face 8	Face 25
Face 9	Face 26
Face 10	Face 27
Face 11	Face 28
Face 12	Face 29
Face 13	Face 30
Face 14	Face 31
Face 15	Face 32
Face 16	Face 33
Face 17	

Figure 3.14 Part with many faces and prime face list

The first face (Face 1) in the prime face list is selected and the faces associated with its inner loop ([14, 15, 16, 17, 18, 19, 20, 21, 22 and 23]) is identified and extracted as shown in Figure 3.15. All those faces is removed from the prime face list. Then each removed face is traversed to identify the sub-features([19, 20] and [21, 22, 23]) formed by the inner loops and outer loops.



Prime Face List	
Face 1	
Face 2	
Face 3	
Face 4	
Face 5	
Face 6	
Face 7	Face 24
Face 8	Face 25
Face 9	Face 26
Face 10	Face 27
Face 11	Face 28
Face 12	Face 29
Face 13	Face 30
	Face 31
	Face 32
	Face 33

Figure 3.15 Identification of inner loops features from first face in prime face list

Each group of faces that forms a feature is extracted and added to the feature list under respective feature categories as shown in figure 3.16. For example, the face 19 and 20 is part of a hole feature. These faces are added under depression category and then compared with predefined rules to identify the type of feature as Hole. Similarly all the groups of faces are added to the feature list and its type is identified.

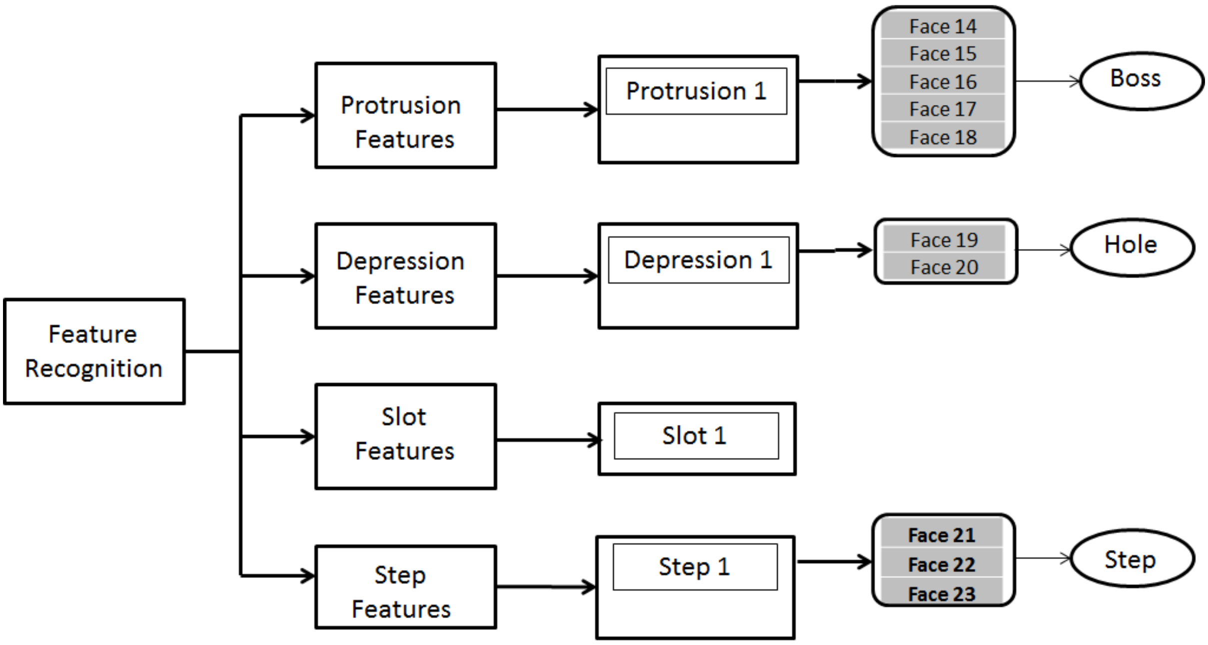


Figure 3.16 Corresponding feature list

As a next step, the Feature Recognition module traverses through the faces available in the prime face list until a face with inner loop is found. As shown in figure 3.17, the next face in the prime face list that has inner loop is Face 10. The process continues similarly as mentioned above. The inner loop faces of Face 10 such as ([24, 25, 26, 27, 28, 29, 30, 31, 32, 33]) are removed from the prime face list and each face is traversed in order to find the sub-feature. The depression feature ([24, 25, 26, 27, 28]) and its sub-feature ([29, 30, 31, 32, 33]) is added to the feature list and its type is identified as shown in figure 3.18.

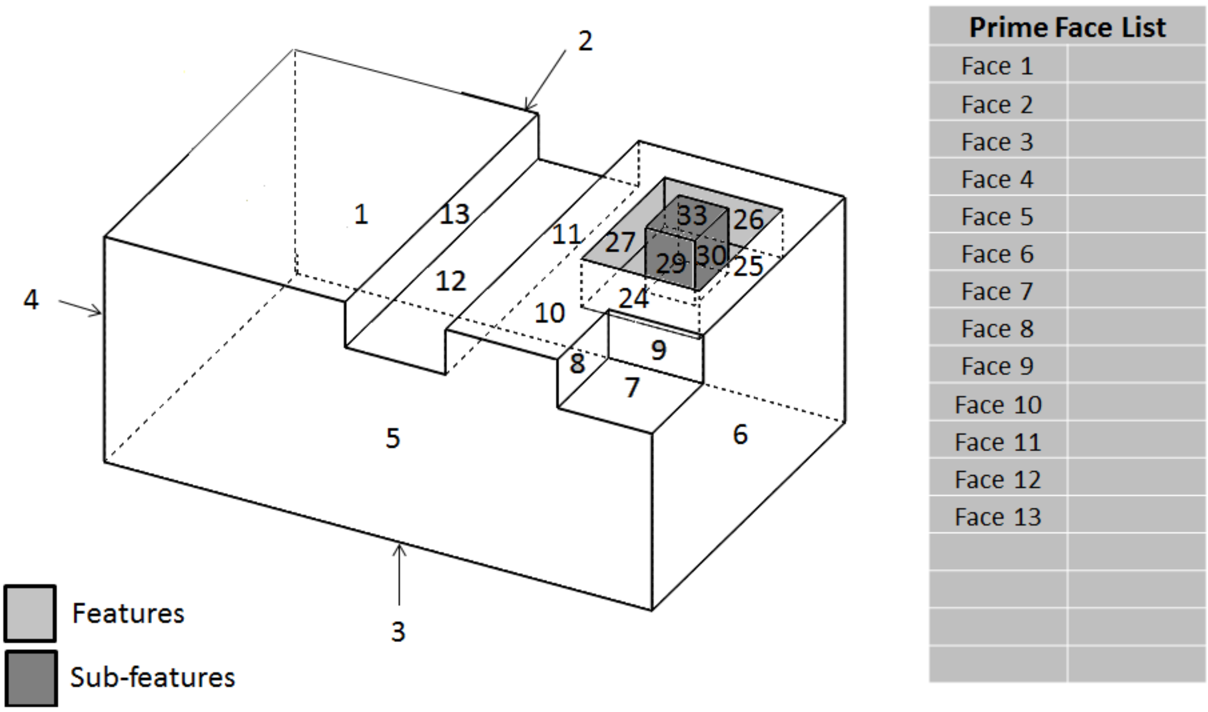


Figure 3.17 Identification of inner loops features from next face in prime face list

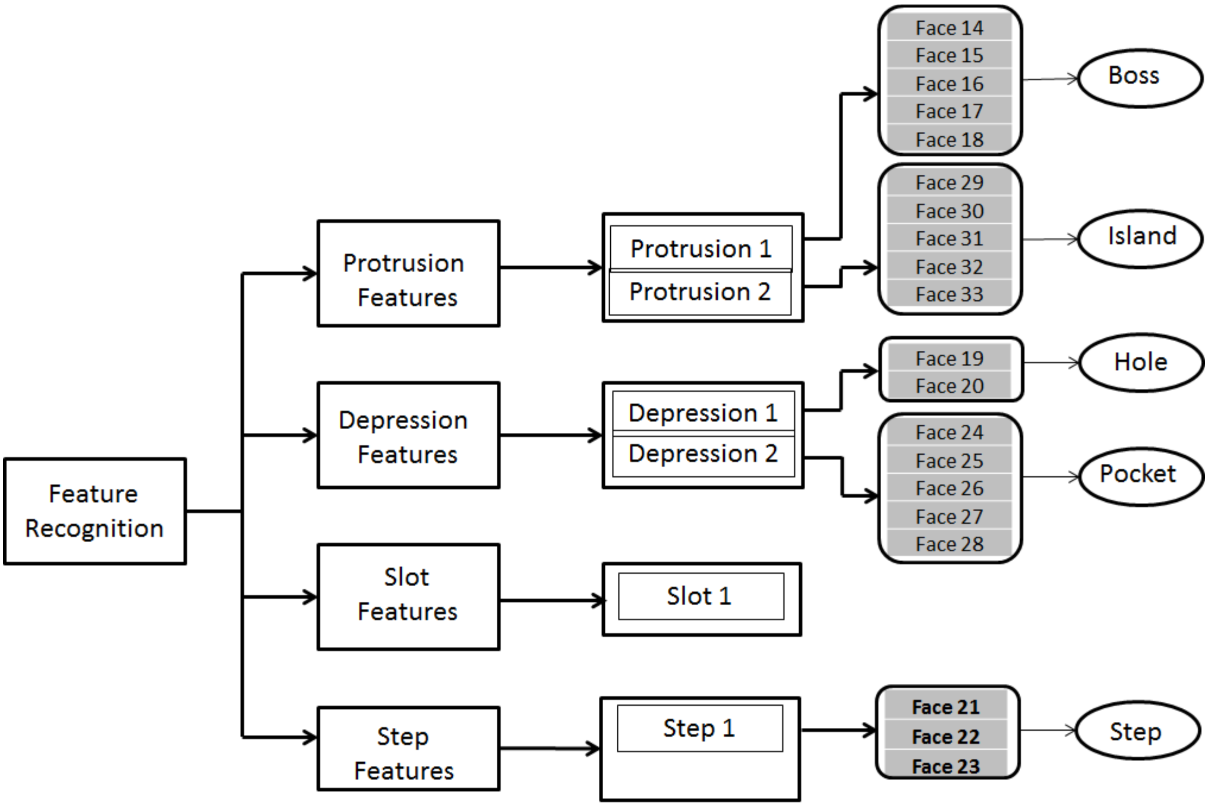
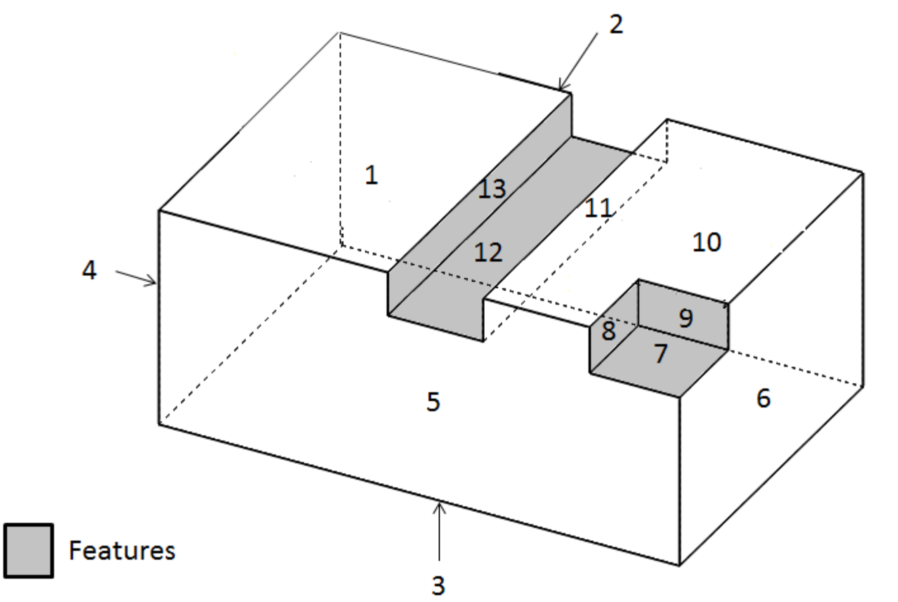


Figure 3.18 Corresponding feature list

When no face in the prime face list has unprocessed inner loops, then the outer loop of each face is checked for the existence of concave edges. Because as mentioned earlier, the concave edge in the outer loop of a face is the key for identifying features such as steps and slots. The adjacent faces of each concave edge is selected and removed from the prime face list. In this part, the faces ([7, 8, 9, 11, 12, 13]) contains convex edges in its outer loop which are removed from the prime face list as shown in figure 3.19.



Prime Face List	
Face 1	
Face 2	
Face 3	
Face 4	
Face 5	
Face 6	
Face 10	

Figure 3.19 Identification of outer loops features from next face in prime face list

Each removed face is traversed and grouped based on its topological relationship and compared with the predefined rules to define the type of the feature. This group of faces which are linked to each other such as ([7, 8, 9]) and ([11, 12, 13]) are compared with the predefined rules and added to the feature list under respective category as shown in figure 3.20. The figure 3.21 shows the part with all feature identified with no feature faces in it and its respective prime face list. The remaining faces in the prime face list is the side faces and bottom face of the part and the faces which does not form the feature through its outer loop edges.

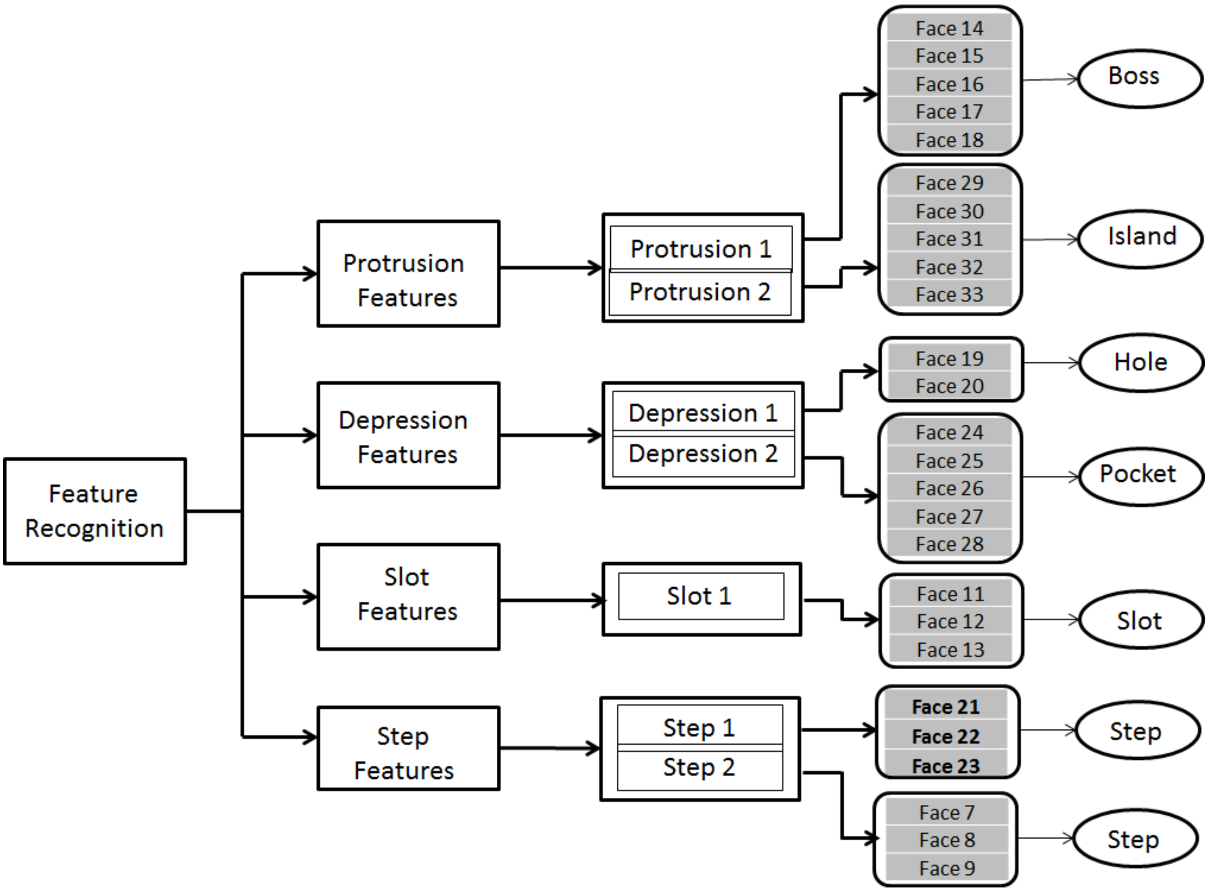
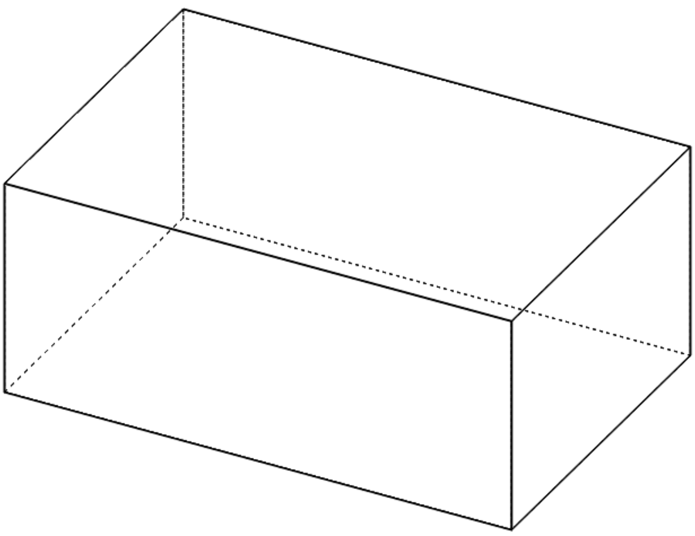


Figure 3.20 Corresponding feature list



Prime Face List	
Face 1	
Face 2	
Face 3	
Face 4	
Face 5	
Face 6	
Face 10	

Figure 3.21 Part and prime face list after extracting all feature faces

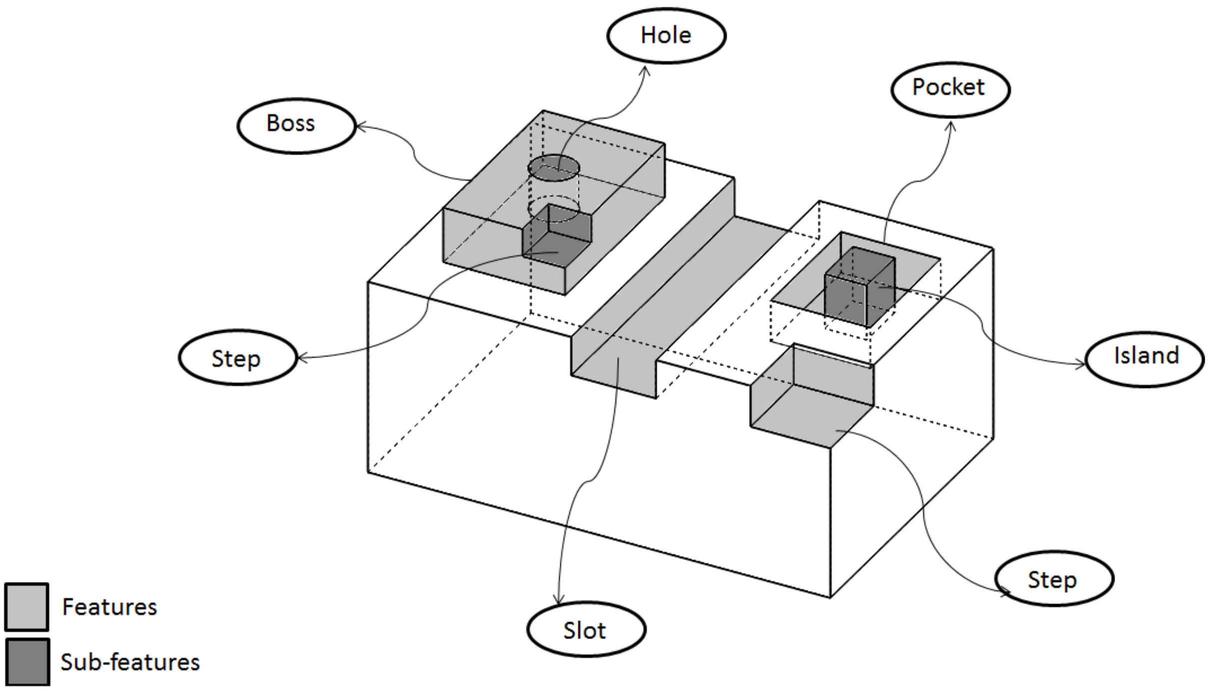


Figure 3.22 Part with recognized features and sub-features

Thus all the features in the part are recognized and its type is identified as shown in figure 3.22. The Feature Recognition module follows the above mentioned steps to recognize the direct features from the part. However, in some cases multiple features may interact each other. In such case, the Feature faces cannot be added to the feature list without knowing the category of the feature. The Feature Recognizing module follows few steps in order the split the interacting features according to their category which is presented below.

3.4 Interaction among features

Recognizing features with well-defined topologies was explained in the previous section. This section is devoted to interacting features. Such interactions are hard to predefine using rules. Feature interaction distorts the relationship between face entities in the part and also result in various topological variations. Under normal condition the feature interaction just splits the edges and faces, and in more complex conditions it may remove the edges or faces or both. The interacting features can be classified into four types [11] as discussed below.

3.4.1 Nested features

When a feature lies inside another feature, it is represented by the presence of inner loops. These types of interactions are said to be nested features. For example, the counter bore holes contains hole inside another hole which is considered as nested feature. This type of interacting features can be identified by the proposed Feature Recognition module.

3.4.2 Concurrent and Virtual Interaction

When a feature interacts with other feature along the outer boundary of a feature, the interaction can be termed as concurrent interaction [11]. For example, interaction between a slot and an open pocket along the boundary as shown in figure 3.23 (a). Since the interaction takes place along the boundary, this type of interaction can also be identified by the Feature Recognition module.

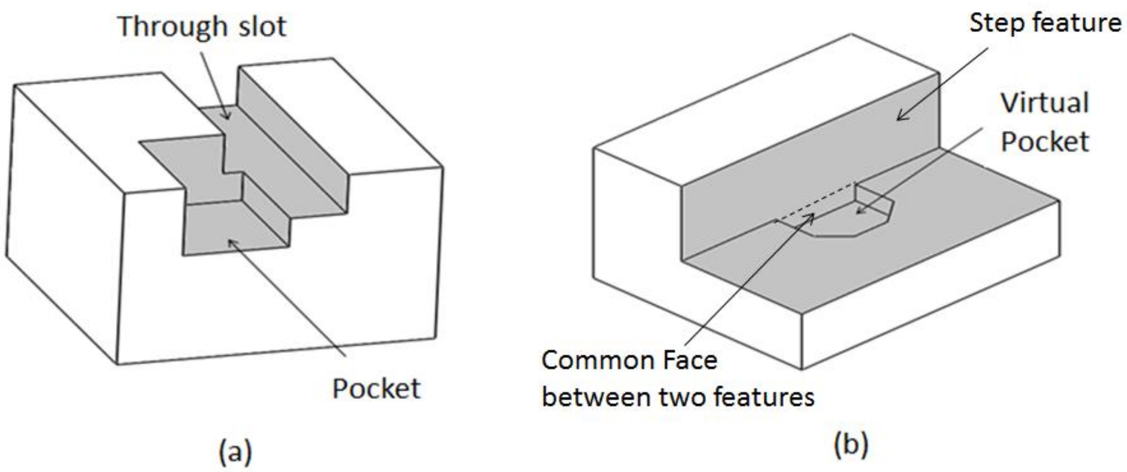


Figure 3.23 Concurrent interaction and Virtual interaction shown in (a) and (b) respectively

When faces of a feature merges with other faces of the part, such interaction are termed as virtual interaction. For example, side face of the pocket merges with the face of a step feature as shown in figure 3.23 (b). Though such interaction removes the edge and shares the face, this feature is formed by the outer loops of various faces. These edges and faces can be identified by the Feature Recognition module and used to recognize its existence.

3.4.3 Inner Loop Interactions

In some cases the interacting features may exist in the inner loop of a face as shown in figure 3.24. As discussed earlier, if the inner loop of a face contains mixture of concave and convex edges, it indicates the interaction between a protruding feature and depression feature. In such case, the algorithm creates a plane on the face containing the inner loop and virtual edges are created by intersecting this plane with one or more face that are common to the two or more features. All the faces associated with the inner loop edges intersect with this plane.

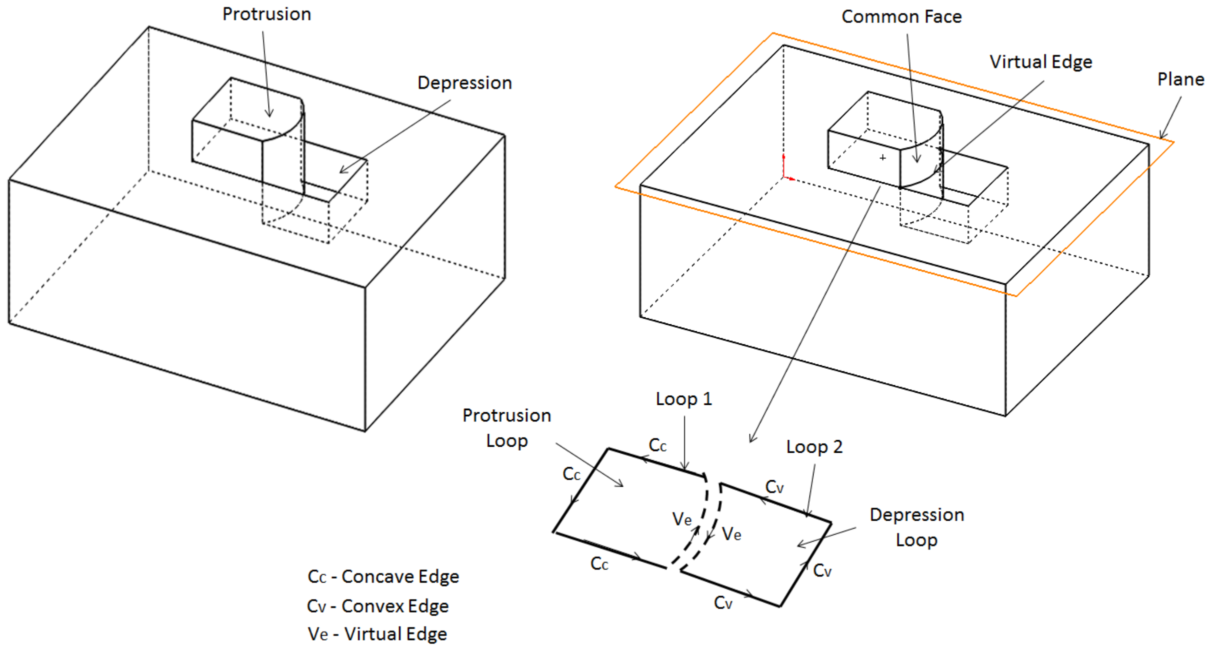


Figure 3.24 Intersecting features inside inner loop

A group of curves are generated on these faces with respect to the intersecting plane. Many generated curves intersect along edges already in the inner loop. Such curves are discarded and the remaining curves are included in the inner loop. This inclusion may require the loop to be divided into two loops with shared edges as shown in figure 3.24. These edges are considered as virtual edges connecting the loop edges. The one or more shared virtual edges with a group of concave edges in a loop determine a protrusion feature. Similarly, the virtual edges with a group of convex edges are determined as depression feature as shown in figure 3.24. This virtual edge helps to split the protrusion and depression feature separately which are then identified as different features.

3.5 Transition Features

In order to improve the strength and aesthetics of the part, sometimes sharp edges and vertices of the part are filleted or chamfered as shown in figure 3.25. The filleted and chamfered faces are determined as the transition features. If the adjacent face of an edge is a tangential face, then it is termed as fillet. Similarly, if the adjacent face of an edge is an inclined planar face, it is termed as chamfer. In this work, all the inclined faces are considered as transition feature.

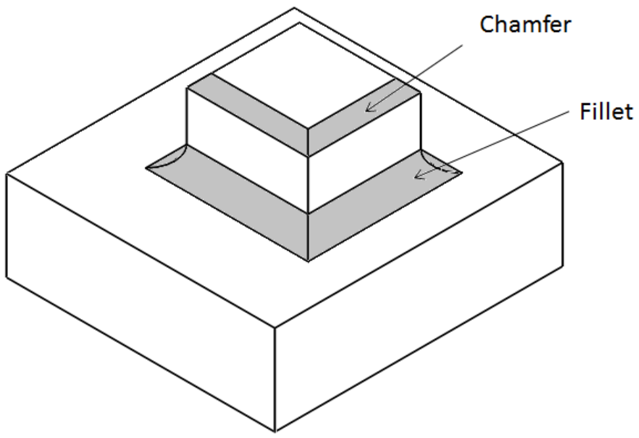


Figure 3.25 Transition features

The transition feature may exist on both convex and concave edge. The chamfer or fillet created on a convex edge is termed as convex chamfer or convex fillet respectively. Similarly, the chamfer or fillet created on a concave edge is termed as concave chamfer or concave fillet respectively. Both chamfer and fillet replace a single convex or concave edge into two parallel or alternate convex or concave edges (highlighted) respectively in the same direction as shown in figure 3.26.

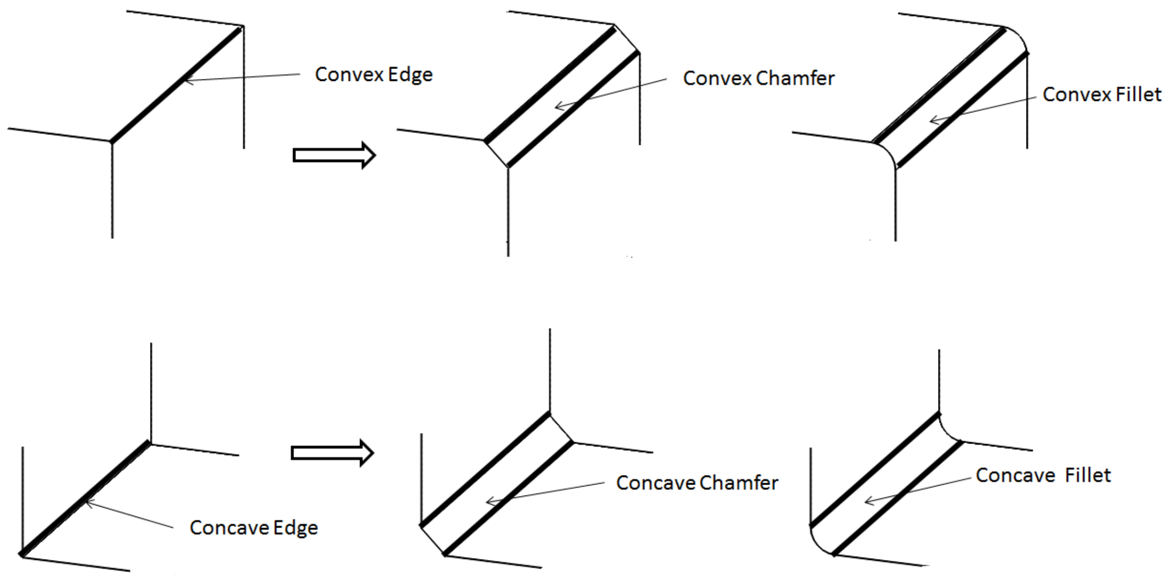


Figure 3.26 Classifications of transition features

The convex fillet or convex chamfer may be part of a feature or a separate feature. All the convex fillets or convex chamfers that lie inside an inner loop feature (Protrusion or depression) are considered as the part of a feature. These transition features are grouped along with the other face of the feature. The convex fillets or convex chamfer which lies along the outer loop edges of a face are considered as separate feature.

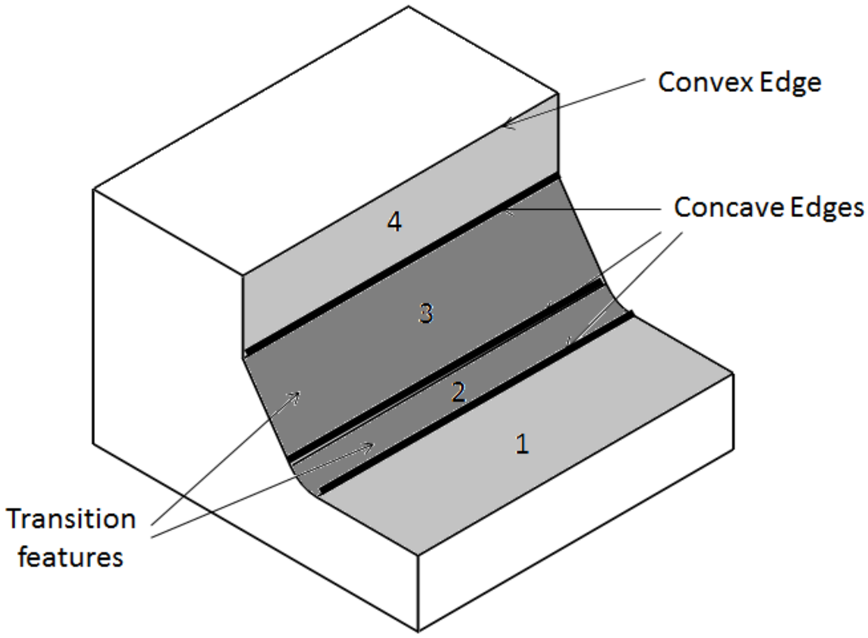


Figure 3.27 Identification of features w.r.t transition features

As discussed earlier, the presence of concave edge on a face indicates the existence of a feature. Whenever, a concave edge on a face contains an adjacent face as either tangential or inclined, it indicates the existence of a concave fillet or concave chamfer. In such case, the parallel or alternate concave edge in the concave transition feature is selected. The adjacent faces of these two concave edges are grouped together along with the intermediate concave fillet or concave chamfer to form a feature as shown in figure 3.27. This process continues recursively by grouping a series of faces until the parallel or alternate edge in the adjacent face is a convex edge. After identifying all the feature faces, the transition features are extracted separately.

3.6 Base face Identification

The next step is to determine the base face and side faces of a feature. This step is essential in order to extract machining parameters and generate cutter contact path. When all the features have been identified, the faces of each feature are scanned to identify the base face and side faces. In the scan the outer loop of edges is traversed in sequence. If the neighboring edges have different classification i.e. concave and convex, then this is considered as a transition. This scan is based on number of concave and convex edges and transitions that exists in those faces. A feature may contain a mixed sequence of concave and convex edges along the boundary of its faces. The identification of sum of all transitions between concave edge and convex edge and between convex edge and concave edge helps in recognizing the type of the each face. For example, the top face of boss and the base face of a pocket will not have any transitions.

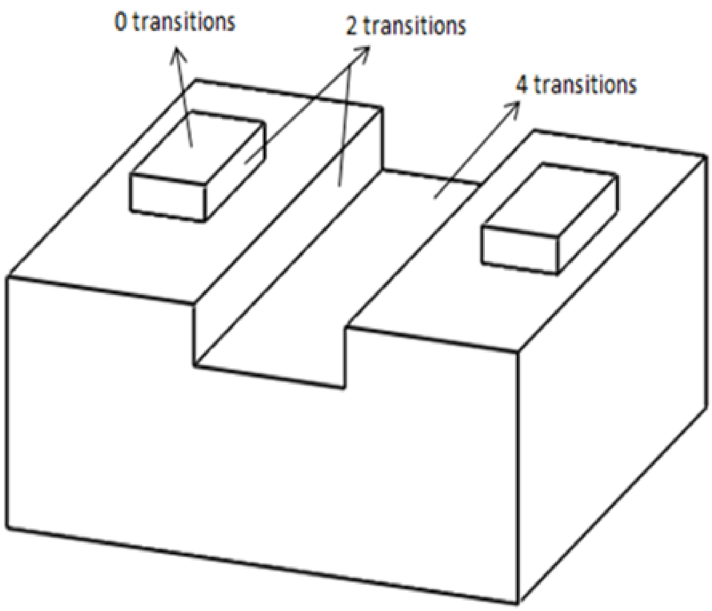


Figure 3.28 Transitions attributes of boss and slot

So if a feature has a face with zero transitions, then that face can be defined as a base face and all other faces as side faces. If a face has four alternate transitions (two convex and two concave), it indicates the base face of a slot feature as shown in figure 3.28. However, in some cases all the faces in a feature may have unplanned number of transitions. In such a case, the identification of base face becomes a difficult task. For example, the figure 3.29 shows a part with blind slot in it. In this slot feature, the base face and one of the side faces has same number of transitions between concave and convex edges. It is difficult to determine the base face among the feature faces. In such cases, the normal of these faces is used to determine the base face. The feature face which has its normal aligned with the tool axis is selected as base face. To do this, the face with maximum number of concave or convex edges is identified and its normal is compared with tool axis in order to define the base face.

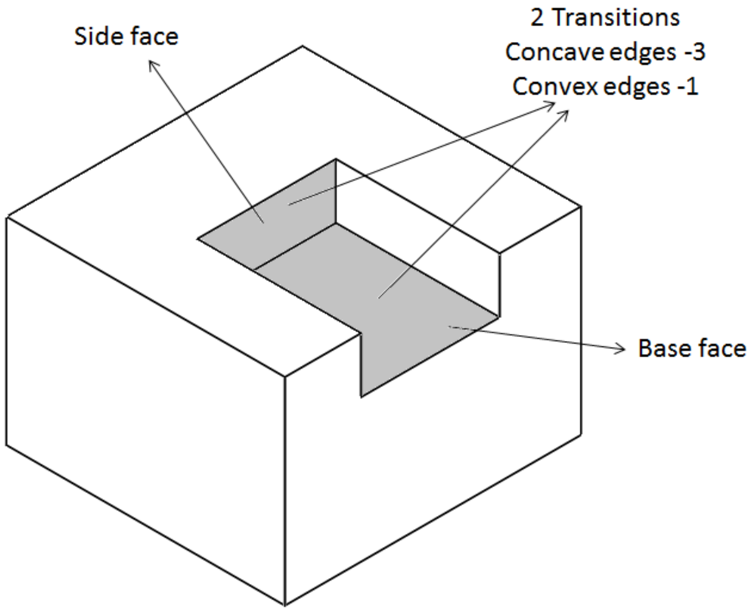


Figure 3.29 Difference between base face and side face

In figure 3.29, the normal vector of the feature faces lies in different direction. So the normal is used to select the base face. This method of using transitions to identify base faces is useful for inclined planar faces.

3.7 Feature Definitions

In this work, the feature recognizer module recognizes various machining features such as boss, blind pockets, through pockets, fillets, chamfers, filleted and chamfered pockets, through slots, blind slots, filleted and chamfered slots, boss or islands in pockets and slots, simple holes, tapered holes, counter-bore holes, counter-sunk holes, stepped coaxial holes. The basic rule that governs the feature recognizer module in this work is discussed below for some common features.

3.7.1 Step feature

There are two types of step feature, through step and blind step. The through step is a feature that connects two planar faces (F_1 and F_2) with a concave edge (e_1) as shown in figure 3.30 (a). The blind step is a feature that contains three or more planar faces (F_1, F_2 and F_3) which are connected to each other with a concave edges (e_3 and e_4) as shown in figure 3.30 (b). In some cases, a through step and a blind step can interact with each other as shown in 3.30 (c). In all these step features the base face is perpendicular to the tool axis and contains minimum two convex edges. This base face contains only two transitions between concave and convex edges. Identifying these two transitions is the key for recognizing a step feature. In figure 3.30 (a) and (b), the transitions are determined from the loops formed by the base face outer loop e_1, e_2, e_3 and e_4. The classification of the edges in the loop is given across the arrow representing it. 'C_c' represents a concave edge and 'C_V' represents a convex edge. While traversing through the edges in the loop, there are two transitions (circles with dashed lines) for all the step types. It must be noted in figure 3.30, that the transition edges of through step shown in bold will be topologically parallel to each other as shown in figure 3.30 (a) and the transition edges of a blind step are always at an angle to each other and when extended, it forms a corner as shown in figure 3.30 (b). This aspect is used to identify the type of step once the step feature has been recognized.

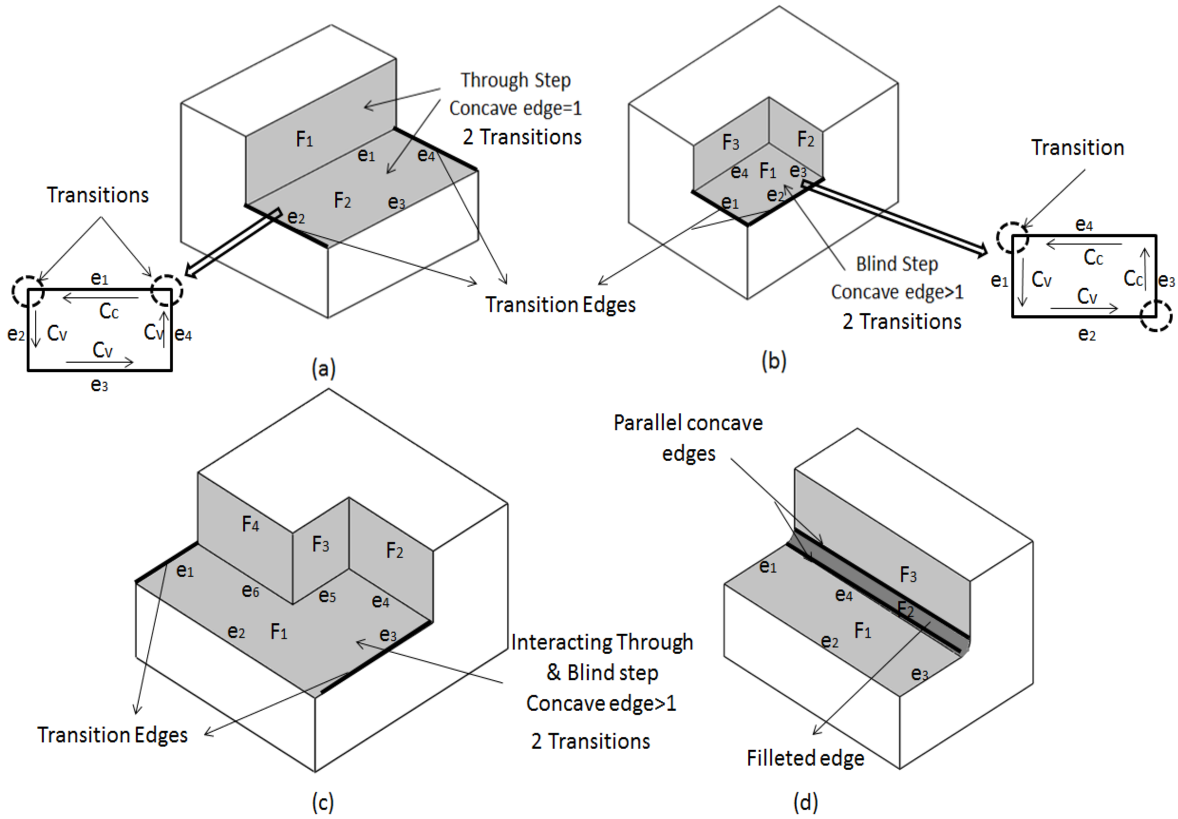


Figure 3.30 Recognition of step feature (a) Through step (b) Blind step (c) Interacting step feature and (d) Step with filleted edges

In some cases the concave edges of the step feature may be chamfered or filleted as shown in figure 3.30 (d). In such case, as discussed earlier each filleted or chamfered face may contains two parallel concave edges (shown in bold in figure 3.30 (d)) which are processed as discussed earlier and the series of adjacent faces that forms the feature is selected and grouped together.

The following procedure illustrates the brief steps for recognizing the step feature,

For each concave edge in the edge list

If the concave edge has two common planar face and orthogonal to each other then

if the outer loop of base faces has two transitions then

Feature is THROUGH STEP

End For

For each concave edge in the edge list

If the concave edge has two common planar face and orthogonal to each other then

if the outer loop of all faces has more than one concave edge then

if the outer loop of base faces has two transitions then

Feature is BLIND STEP

End For

3.7.2 Slot feature

There are two types of slot feature, through slot and as a blind slot. The through slot contains three or more planar faces with two faces perpendicular or at an angle to the base face. The through slot has the largest number of transitions in its base face. There are four or more transitions in the base face of a slot. The through slot shown in the figure 3.31 (a) has three mutual perpendicular or near perpendicular faces (F_1, F_2 and F_3) connected by the concave edges (e_1 and e_3). The four alternating transitions in the base face are shown in the figure 3.28 (a). Identification of number of transitions in the base face is the key for determining the type of the slot. The blind slot would have four or more planar faces with three faces perpendicular or at an angle to the base face. The blind slot shown in figure 3.31 (b) has four faces (F_1, F_2, F_3 and F_4) connected by the concave edges (e_1, e_3 and e_4). The base face in the blind slot feature will have only two transitions and contains three or more concave edges as shown in figure 3.31 (b). In figure 3.31 (a) and (b), the transitions are determined from the loops formed by the base face edges e_1, e_2, e_3 and e_4. The classification of the edge is given across the arrow representing it. While traversing through the edges in the loop, there are four transitions in the through slot and only two transitions in the blind slot. This aspect of the transition is used to recognize these features.

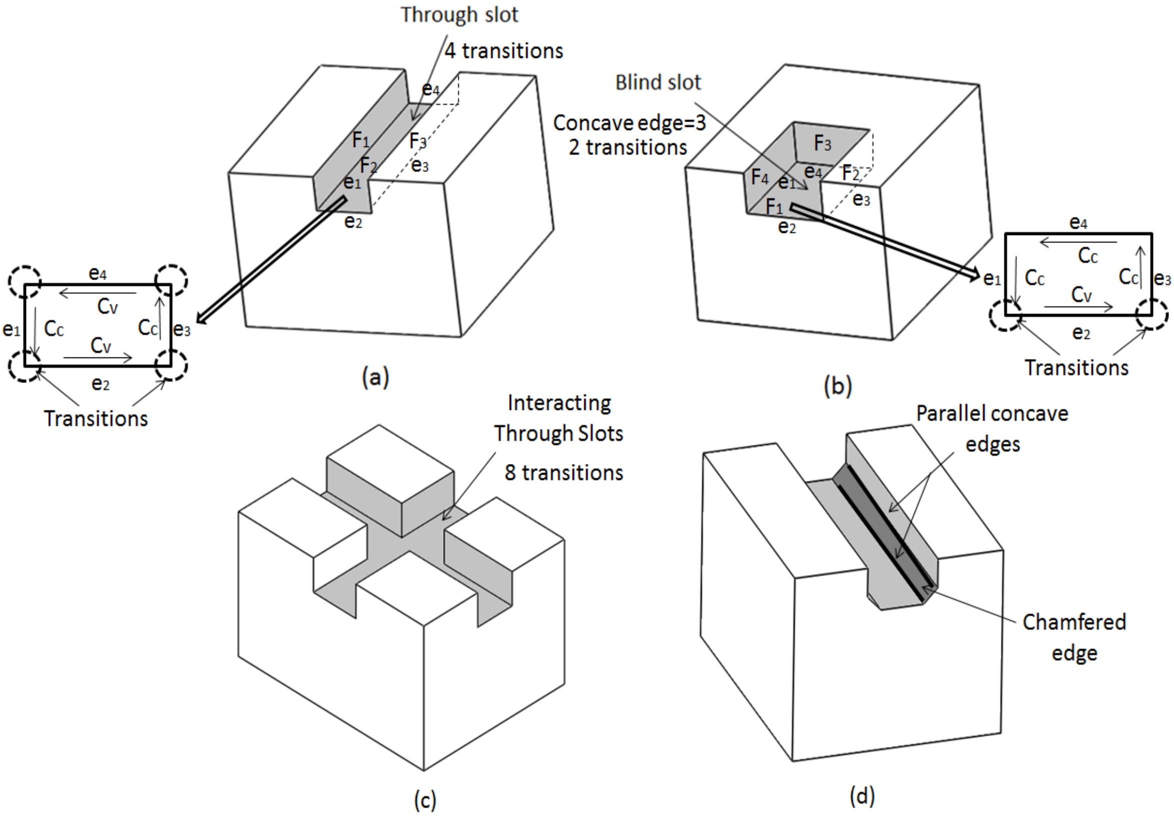


Figure 3.31 Recognition of slot feature (a) Through slot (b) Blind slot (c) Interacting slot feature and (d) Slot with chamfered edges

In some cases, multiple slots may interact with each other. In such case, the number of transitions increases accordingly. For example, the figure 3.31 (c) shows two through slots interacting each other which contain eight transitions in its base face. In some cases, the concave edges in the slot feature may be chamfered or filleted as shown in figure 3.31 (d). As explained earlier, the parallel concave edges in the fillet or chamfer is processed and the series of adjacent faces that forms the feature are grouped together to identify the slot feature.

The following procedure illustrates the brief steps for recognizing the slot feature,

For each concave edge in the edge list

if the outer loop of base face has two or more concave edges then

if the base face has four alternate transitions then

Feature is THROUGH SLOT

if the outer loop of base face has three or more concave edges then

if the concave edges are continuous and if the base face has two transitions then

Feature is BLIND SLOT

End slot rule

3.7.3 Boss feature

A boss is a feature which protrudes in the positive direction from the base face. Normally the boss is determined by inner loop edges of the base face, with all the edges as concave edges. Occasionally the boss feature may have filleted or chamfered edges, which form a tangent edge. The tangent direction and its adjacent faces combined together determine the existence of boss with round edges.

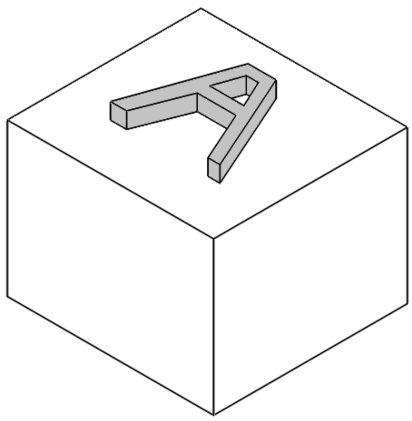


Figure 3.32 Recognition of Boss feature

The following procedure illustrates the brief steps for recognizing the boss feature,

Find the inner loop on base face, and its edge type

Perform concavity test on all edges of the loop

If edge count of loop equal to number of concave edge

Feature is BOSS

Get the end face or top face

End boss rule

3.7.4 Pocket feature

A pocket is a feature with one base face which is more accessible for machining process and with a group of side faces. The pocket is defined by the inner loop of the face that contains all convex edges. This group of convex edges in the inner loop denotes the existence of depression feature and all the edges of its base face are concave edges as shown in figure 3.33 (b). This pocket with a base face is defined as a blind pocket. If a pocket does not contain any base face, then the pocket is said to be through pocket as shown in figure 3.33 (a).

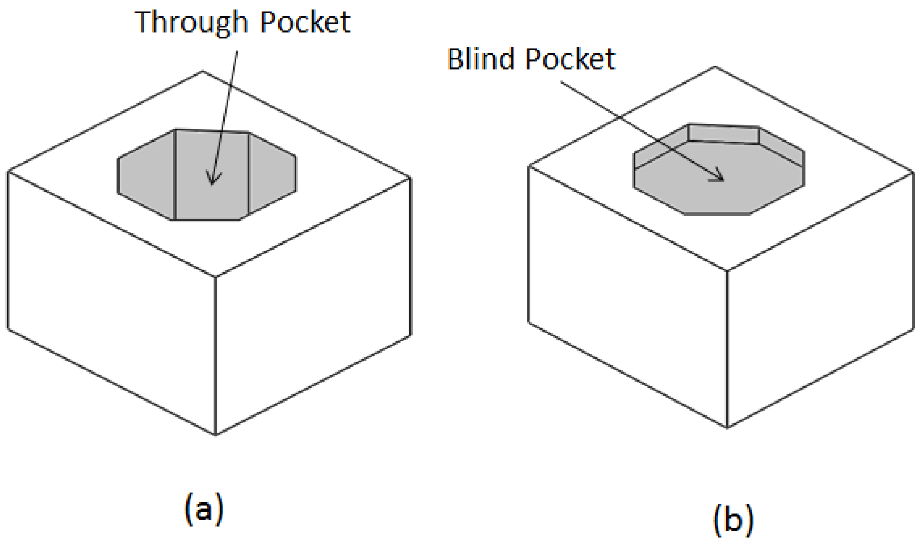


Figure 3.33 Recognition of Through and Blind POCKET as shown in (a) and (b) respectively

The following procedure illustrates the brief steps for recognizing the pocket feature,

Find the inner loop on base face, and its edge type

Perform concavity test on all edges of the loop

If all edges are convex edges then

If end face or base face is present then

Feature is BLIND POCKET

Else

Feature is THROUGH POCKET

End pocket rule

3.7.5 Simple, Counterbore and Countersunk holes

The simple hole or stepped hole is recognized based on the inner loop that lies on the base face which contains the hole feature in it. The stepped hole is a series of coaxial holes such as counter bore and counter sunk hole. The feature type depends on the number of cylindrical, planar and conical faces which forms the hole feature type and also few characteristics like inner loops of the planar face, top and bottom radius of the conical face. All the faces of the hole features contains the same axis, which is the important property used to identify the group of faces that forms simple or stepped hole.

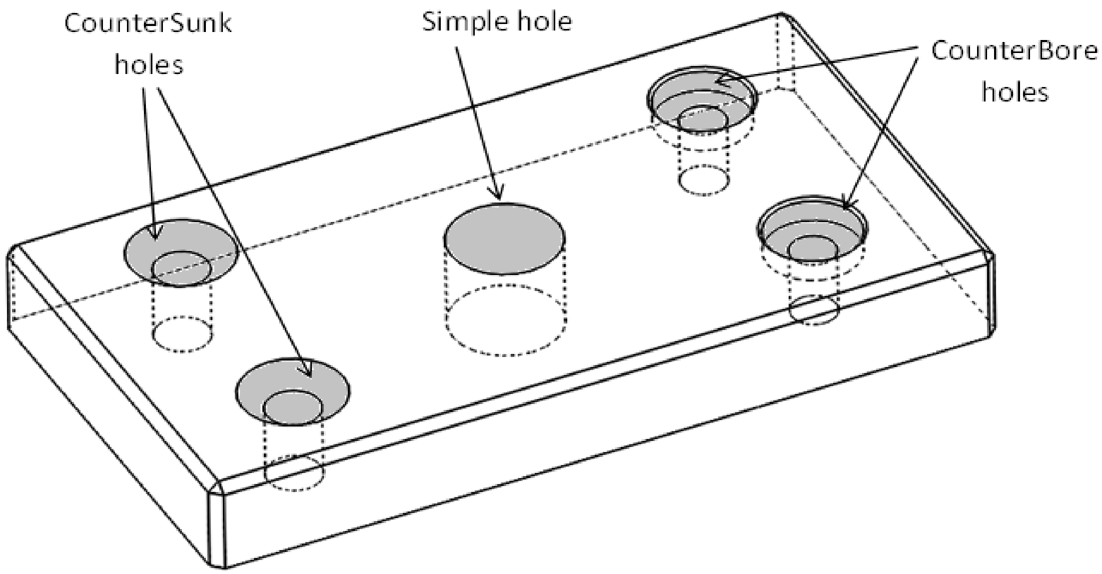


Figure 3.34 Recognition of HOLES

The following procedure illustrates the brief steps for recognizing the simple and stepped hole,

Find the inner loop on the base face, which has one edge and edge type-Circle

Calculate the angle between the hole axis and face normal

Get the adjacent face of the edge

If the adjacent face is Cylinder

Get the end face of the cylindrical face

If there is no end face on the other side

Feature is THROUGH HOLE

Else

Feature is BLIND HOLE

If the end face is planar with no inner loops then

Feature is SIMPLE HOLE with flat end

If the end face is conical then

Feature is SIMPLE HOLE with conical end

If the end face is planar with inner loops then

Get the edge of the inner loop and its adjacent cylindrical face

Get the end face of cylindrical face

If (no end face) or (planar end face with no inner loops or conical end face) then

Feature is COUNTERBORE HOLE

If its planar with inner loops then

Feature is STEPPED coaxial HOLE

End Hole rule

When all the features has been categorized and identified, the portion of the part to be machined is known. The cutter contact path can be generated on the base faces of each feature in order to machine the feature. The type of the tool and size of the tool to be used depends on the type of feature and its parameters. Before proceeding with machining process, it's necessary to extracts these parameters from the feature. The extraction of various parameters from the feature is discussed below.

3.8 Feature parameterization

The feature parameterization is the next step after feature recognition for machining application. The Feature Recognizer includes an algorithm to extract the necessary parameters for machining process from the geometry of the topological entities. This process automatically extracts the basic parameters of the feature. The parameters obtained by the feature recognizer module are distance parameters with respect to the reference geometry, sketch parameters, angle parameter and radius. The distance parameters can be defined as the distance between two reference planes such as planar faces, planar edges and also distance between a reference plane and a point. In order to machine a feature, the tool has to travel through certain depth to remove the material from the part. This tool travelling distance in Z-direction is determined by the depth parameter which can be identified by calculating the distance between the top reference and bottom reference of a feature.

The intersecting edges or sketches formed when a reference plane intersects with one or more faces are defined as sketch parameters. This parameter can also be used to estimate the distance between two reference geometry. The edge can be used to estimate the angle between its two adjacent faces which is termed as angle parameters. The recognizer also calculates the radius of circular edges or arcs, for example the radius of the hole can be determined by identifying the radius of its circular edge.

The parameter extraction from the feature can differ according to the type and structure of the feature. The figure 3.35 shows the necessary parameter extraction from various features. In the step feature, if all the faces are planar and if all its outer edges are linear, then the top edge of the side face can be used as the reference as shown in figure 3.35 (a). The distance between this reference edge and the base face gives the depth of the step feature.

If a feature has multiple side faces and if those faces are in different heights, the face with maximum height is considered and used as the reference. For example, a blind step may have two or more side faces, and if their heights do not match, the face with maximum height is used as the reference. This process is common for slot features, boss and pockets as shown in figure 3.35 (b), (c) and (d) respectively. But for some depression features such as hole, additional parameters has to be extracted. For example, to drill a hole the radius and type of the hole must be known. If the hole is a simple hole, the distance between its two circular edges of the cylindrical face determines the depth of the hole. The radius of the circular edge gives the radius of the hole.

The counter-bore holes are similar to the simple holes, but is contains two radius and two depth parameters as shown in figure 3.35 (e). If the hole is a counter-sunk hole, it contains an additional angle parameter between the conical face and the hole axis as shown in figure 3.35 (e).

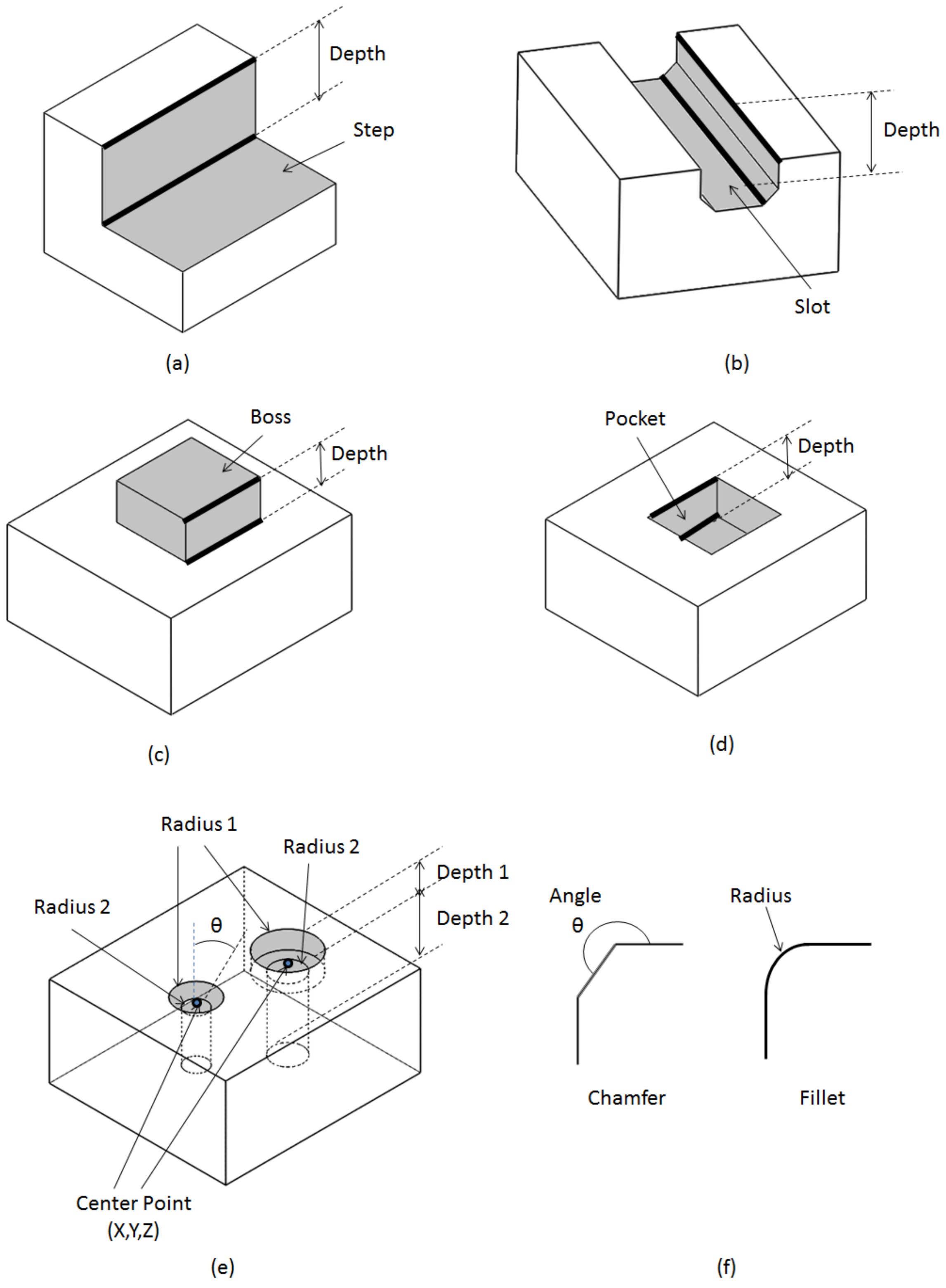


Figure 3.35 Parameters of various features

The module also builds the reference geometry in terms of topological entities which is used for many applications. For example, if a feature exists on non-planar or non-parametric surface as shown in figure 3.36, then the inner loop edges which defines the features may not have linear edges.

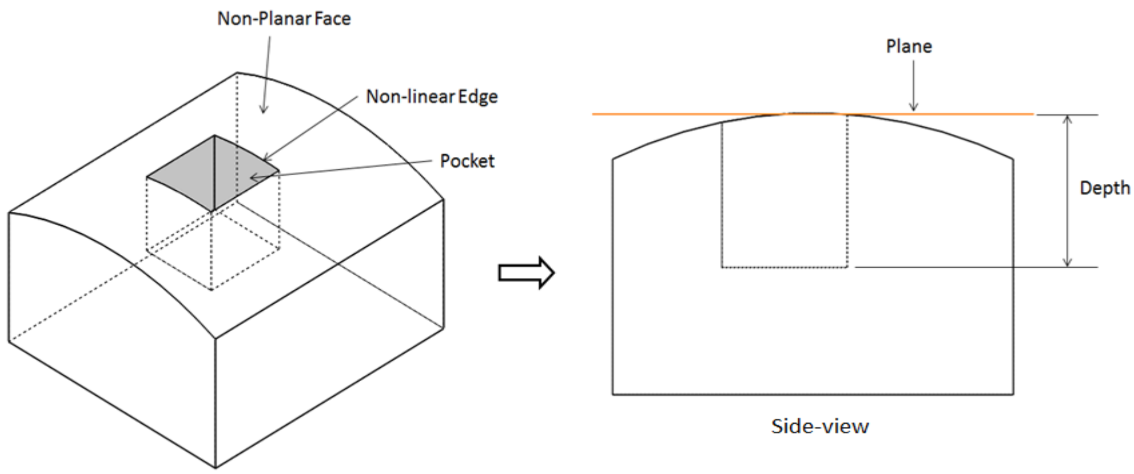


Figure 3.36 Reference Geometry for Features on Non-Planar surfaces

In such case, a reference plane is created on the top of the non-planar surface which is used as reference for calculating the depth of the feature. The distance between this plane and the base face of the feature determines the depth parameter. Similarly, if the base face of a feature has non-planar or non-parametric surface, a reference plane is created on it and distance between the two planes gives the depth.

The feature recognizer module discussed in this work can be used in automatic recognition of features, which recognizes design and machining features by using geometry and topological entities. This Feature Recognizing module can recognize the common features such as boss, pockets, holes, slots and steps. The time taken by the module to recognize features depends on the number of features and the size of the model. Though the module recognizes all the faces in a feature, it mainly focuses on the base face of the feature which is the main entity in machining process. These entity and parameters extracted by the feature recognizer module are used effectively for tool path planning which is discussed in next chapter.

Chapter 4

Tool-Path planning

Following the feature recognition on a part as discussed in previous chapter, the tool path used to machine the part is built. The generic process of tool path planning was discussed in chapter 2. As explained earlier, the entire tool path planning process is based on geometry and topology. The positioning of the tool, the movement of tool etc. are all determined using geometry and topology of the part. This makes tool positioning a prime candidate for algorithmic tool path planning.

Automatic tool positioning and planning method have been researched and implemented [13][17][33][34][42][43]. These methods use triangulated model for tool path planning on parts with specific geometry and fixed topology. Fixed types of tools are used in the process. Overall the cited works confirm that automatic tool path planning is viable but is limited in the type of geometries and topologies that it can cover. Furthermore, as the B-Rep model is converted to triangulated model, the geometry and topology information embedded in B-Rep models is lost. This loss of information necessities an additional step of gouge detection and checking at all cutter contact points before tool path planning. Furthermore, these methods do not have a robust gouge avoidance algorithm and this limits their application to specific family of parts.

Automatic tool path generation has also been used to machine features as well. The feature topology is used to select the tool path topology and the tool type. These parameters, then coupled with the part geometry are used to determine the tool path. The methods [33][34] work well when the part is composed of simple feature. Each feature is machined separately with a tool path designed for its geometry. However, when features interact, the tool paths can lead to gouging, see figure 3.20. The failure of the process can be traced to the difficulty in merging two trajectories into one. This difficulty arises because the tool positioning and trajectory planning are inter-twined with one another.

Patel [43] proposed a method to de-link the tool positioning and trajectory planning elements. In the approach used by Patel, first the foot prints is generated and discretized into points. Next, the tool is positioned at each point while ensuring that gouging does not happen. Patel's work is limited to a part family and is not easily extendable. In this work, a modified version of this concept has been adopted. The tool path planning is broken into two parts: Cutter Contact Path Planning and Cutter Location Path Planning based on the tool geometry.

The concept used in this work is centered on the use of base surfaces. A Base surface is a part of the feature and defines the plane and its boundaries within which the feature is scoped. When viewed from the tool axis direction all base faces in a part (with no occlusions) do not overlap each other. If cutter contact paths are generated within the bounds of the base faces the tool paths will not intersect. Considering this as a main advantage, the base faces are used to generate the Cutter Contact Path. This method is generic and flexible, and can be easily adapted to any type of tool. The trajectory of the tool defined relative to the base plane is the Cutter Contact Path. The Cutter Contact Path and Cutter Location Path are same for flat-end tool. The flat-end tool can move along the Cutter Contact Path without gouging, as the tool center and contact point is same. But in the case of ball-end and toroidal tool, the tool center and contact point is different which results in Cutter Location Path to be different from Cutter Contact Path. Although the use of base faces eliminate intersection of tool paths, gouging can still happen when two paths are close to one another. The second step is that of determining the gouge-free Cutter Location Points. In Cutter Location Path generation the tool is positioned and checked for gouging with the whole part. If gouging occurs the cutter avoids machining the point depending on the type of feature being machined. This strategy is feature dependent and is discussed later. Parts with occlusions cannot be machined using 3-Axis CNC machines and are thus not a restricting factor.

The proposed method directly uses the B-Rep model for tool path planning. Since the B-Rep is embedded with all geometry and topology information, the location and parameters of each entity are known. The Cutter Contact Paths are determined independently using the base face for each feature that has been identified in the part. The geometry of the base faces are used to generate the Cutter Contact Paths. If any feature does not contain base face, it is considered as a through feature i.e. through hole or through pocket. Though the features interact, the base faces are non-intersecting.

4.1 Tool Path Planning

After all the features are recognized from the part by the Feature Recognition module, they are used by the Tool Path Planning and Process Planning modules. The Tool Path Planning module and Process Planning module are inter-connected to each other. The Tool Path Planning module splits its operation in two steps, roughing and finishing. Before proceeding with roughing and finishing, the Process Planning module creates a bounding box for the solid model and sequences the features to be machined based on its location. The process of bounding box creation and sequencing the features are discussed below.

4.1.1 Oriented Bounding box

The Bounding Box is a simple, convex, rectangular box that encloses the solid model to determine the size of the part and it is assumed as the stock for the part to be machined. The bounding box with edges parallel to the coordinate axes is commonly used to bind a constructive solid geometry models. The minimum and maximum coordinates of the solid model becomes the corners of the bounding box, which are connected to each other to form a rectangular box. The volume of the stock required for machining the part can be estimated from the parameters of rectangle box such as length, width and height. The bounding box which is aligned with the origin is considered to be more effective due as its coordinate system matches that of the machining workspace and simplifies tool path planning. The proposed work is implemented using Solidworks- Application Programming Interface. The inbuilt tools in the Solidworks-API create the minimum bounding box of the solid model as shown in figure 4.1. This bounding box is oriented with the parts coordinate system, resulting in axis aligned bounding box.

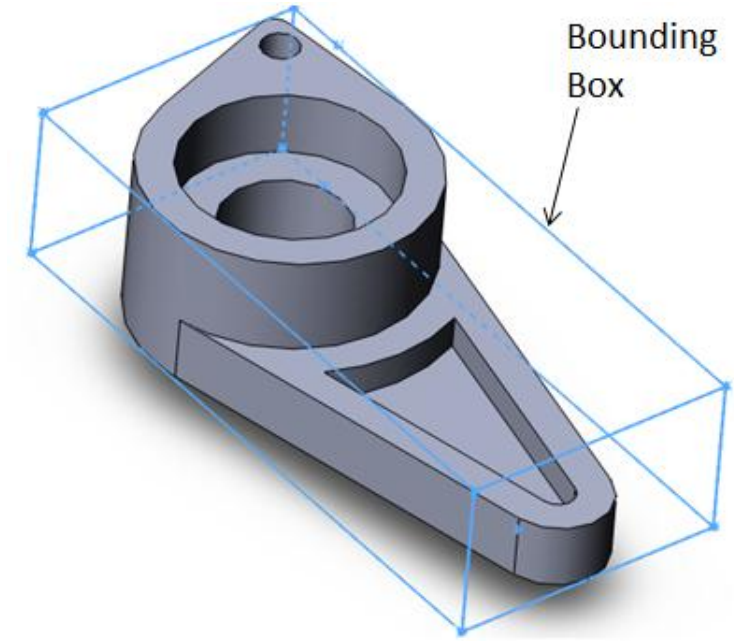


Figure 4.1 Bounding box

4.1.2 Sequencing features for machining

The machining operation for making a part from its solid model are planned as a sequence with respect to the list of features recognized by the Feature Recognition module. The Process Planning module sequences the features based on its location from the top plane of the bounding box. The steps involved are:

1. The top face of the stock (bounding box) is extracted and considered as the starting face. The contour parallel tool path (described later) is generated on the top plane of the bounding box with small offset in z+ direction which leaves thin layer of material for finishing process.
2. The Feature Recognizing module determines the base face of each feature. In order to allow the tool to reach the features of the model quickly, the excess material must be removed. As the bounding box determines the stock size, the portion that lies above the feature is considered to be excess material. The Tool Path Planning module generates the contour parallel tool paths above the base face of the feature. The Process Planning

module sequences the features based on the depth of base face of the features below the starting face. The top-most base face is the first feature and the deepest base face is the last feature in the sequence. Since the material to be removed may be large it is machined in layers. These layers are equally spaced between the base face and the top of the stock at the time of machining. A sequence of machining steps is built as a list. This machining steps list contains a feature and raised base face at a specified depth from the top face. A feature may occur a number of times depending on the layers of material above it. Each layer is at a specific depth from the top face and should be machined separately until the base face of the feature is reached by the tool. Hence each layer is added to the sequence list under the respective feature. After the roughing steps have been added, the finishing steps are added to the list. Among the finishing steps, the feature with the highest base face relative to the top face is finish machined first. The other features are finish machined in sequence determined by the depth from the top face. The features and the depths are added to the machining step list. After the machining step list is populated, the Tool Path Planning module generates tool paths for each step in the list sequentially.

4.1.3 Roughing

The purpose of rough cutting is to remove as much material as possible and bring the stock shape close to the final geometry. During the roughing process, the surface tolerance and surface quality is considered as secondary and the tool of maximum size is used to aggressively remove unwanted material. For each feature in the machining step list, the Tool Path Planning module choose the type of tool and tool path based on the type of feature being machined. Next the Cutter Contact Paths are generated on the raised base plane whose height is determined from the machining step list. The Paths for the various features vary and are discussed later. The different types of Cutter Contact Paths used in this work are one-way, zig-zag and contour parallel. These are used for roughing process according to the type of the feature. After the Cutter Contact Paths have been built, they are divided into small and equal intervals, and their coordinates are extracted to determine the gouge-free Cutter Location Points which is discussed later in section 4.4. Generally the cutter used in roughing process is a flat end mill cutter for all planar surfaces, and ball end nose cutter for free form parametric surfaces. The gouge free tool paths are determined for all features in sequence.

4.1.4 Finishing

The purpose of finishing process is to remove the thin layer of material on the surface in order to maintain the tolerance and quality of the product. The feature sequencing is extracted from the machining step list and starts with upper and outermost feature and moves inwards. For each feature in the machining step list, the Tool Path Planning module chooses the type of tool and tool path based on the type of feature being machined. Next the Cutter Contact Paths are generated on the raised base plane whose height is obtained from the machining step list. The Paths for the various features vary and are discussed later. The different types of Cutter Contact Paths used in this work are one-way, zig-zag and contour parallel. These are used for finishing process according to the type of the feature. The generated Cutter Contact Paths are divided into small and equal intervals, and their coordinates are extracted to determine the gouge-free Cutter Location Points. Generally the same cutter is used to machine the entire feature during the finishing process. The gouge-free tool paths generated are determined at smaller step distance and path intervals for better surface finish. The finishing operation in this work uses the same tool path strategy as the roughing process, but the feed rate, step size, path intervals and depth intervals differs from those used in roughing process. The type of tool paths generated for roughing and finishing varies with the type of feature. The different tool paths types used for machining various types of features are discussed below.

4.2 Feature based tool path types

The type of tool path ideal for machining the feature type is embedded in it definition and can be selected without any user interaction. The tool path planning module automatically selects the defined tool path type for machining the feature. The common types of tool paths used for machining parts are,

1. One way direction parallel tool paths
2. Zig-Zag direction parallel tool paths
3. Contour parallel tool paths

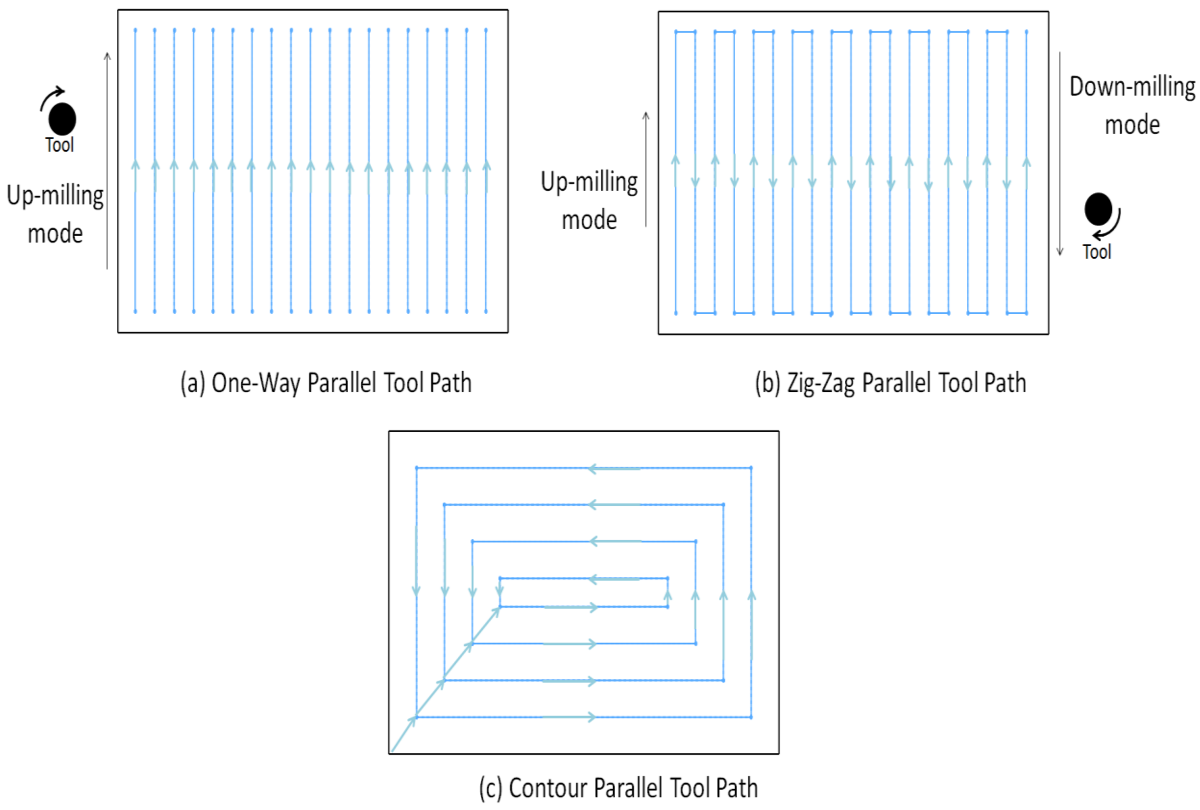


Figure 4.2 Types of tool paths

In one way direction parallel tool paths the cutting of material takes place only in up-milling mode as shown in figure 4.2(a). When the tool encounters down milling it lifts and moves to the next point on the trajectory. In zig-zag direction parallel tool paths and contour parallel tool paths the cutting of material takes place in both up-milling and down-milling mode. In zig-zag parallel tool path, the tool moves continuously on a surface in a zig-zag manner as shown in figure 4.2(b). In contour parallel tool path, the tool moves along the group of parallel closed loop paths as shown in figure 4.2(c). These tool paths are used to machine a variety of feature which is discussed next. This discussion is divided into three parts as follows,

1. Machining Pockets/Slots/Steps with no Islands
2. Machining Pockets/Slots/Steps with Islands
3. Machining Freeform Parametric Surfaces

1. Machining Pockets/Slots/Steps with no Islands: The proposed method uses the contour parallel type tool path for machining all types of pocket features, step features and slot features whose base face do not contain inner loop Protrusion features. The contour parallel cutter contact path is generated in the raised base face of the feature by linking all the edges in a loop and offsetting the loop by a side step distance that is proportional to the tool radius. The path segments are linked based on the tool travelling direction. This type of tool path is a collision free tool path, since it is used only with features having planar base face with no protrusions features on it.

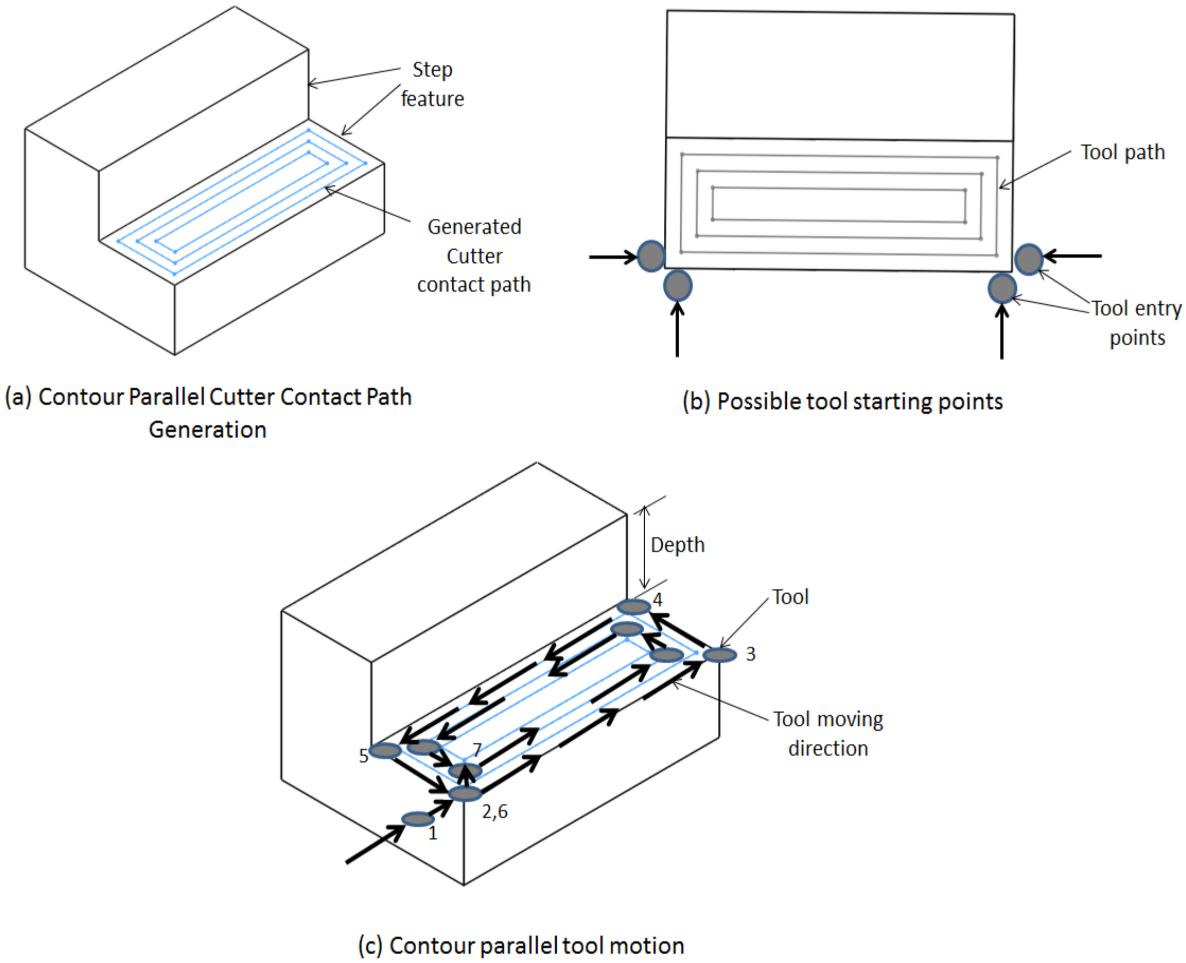


Figure 4.3 Contour parallel tool path generation for features with no islands

The following steps are used to implement this method,

1. The outer loop of the base face which contains a set of edges linked together is selected and a contour parallel cutter contact path is built with an offset of side step distance proportional to tool radius. This process is repeated until the contour parallel cutter contact paths are generated all over the base face as shown in figure 4.3 (a). All the paths generated form closed loop path. If the tool path intersect, the intersection point is detected and the tool path is broken into closed loops. The loops that result in undercuts are discarded. The process continues with the remaining closed loops.

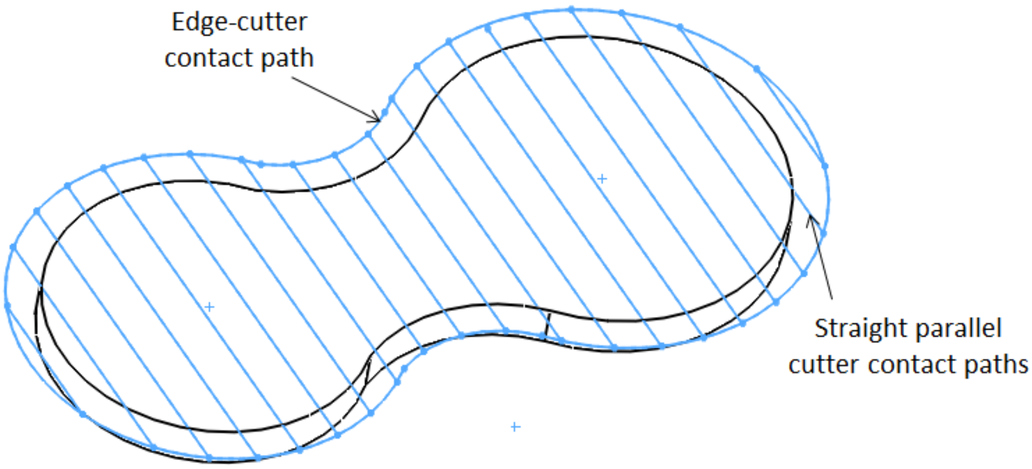
2. Each segment in the cutter contact paths is discretized into equal intervals according to tool radius and its coordinates are extracted. These coordinates denotes the Cutter Contact Points from which the gouge-free Cutter Location Points are calculated with a uniform offset of tool radius along the tool axis. These Cutter Location Points are extracted to generate the Cutter Location Path as shown in figure 4.3 (b) which is discussed later.
3. The above steps are repeated for each parallel contour in order to build a machining tool path for a particular feature.

2. Machining Pockets/Slots/Steps with Islands: The one way and zig-zag tool paths are used for machining the features like pockets, steps, and slot with islands or protrusion in them and also for machining features whose base faces are non-planar or sculptured surfaces. The creation of one way and zig-zag tool path differs according to the base surface type. In planar base surfaces the tool path is generated from the Cutter Locations Points in the same manner as the contour parallel tool path. In non-planar base surfaces, the contour parallel Paths cannot be generated by offsetting the edge profile because the surface parameters are not constant and the normal of the surface varies continuously at each location. The method for generating Cutter Contact Path on non-planar surfaces is explained in the freeform parametric surface section.

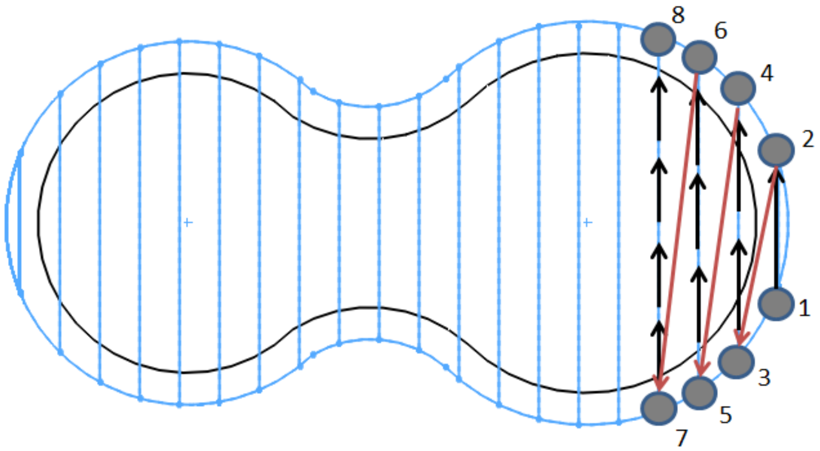
In planar base surfaces, the surface normal is constant at all locations and so Cutter Contact Path can be generated easily. Both one-way and zig-zag parallel tool path are created by generating Cutter Contact Paths parallel to either x–plane or y-plane all over the surface at equal intervals based on tool radius. In one-way parallel tool path, the path segments are not linked to each other. Each segment is divided into equal intervals and sequenced in order.

When the tool reaches the end point of the first path segment, it travels in z+ direction to a safe height where no other feature exists and retreats to the start of the next path segment. The tool then moves down to the next pass with a displacement in z- direction and reaches the starting point of the next pass. An example of one way tool path movement is illustrated in figure 4.4. The following steps are used to implement the one-way parallel tool path method,

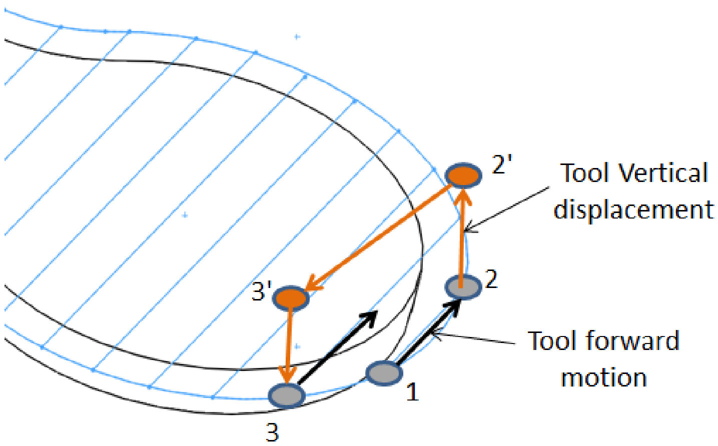
1. The outer loop of the base face which contains a set of edges linked together is selected and a Cutter Contact Path is built with an offset of tool radius. This outer boundary Cutter Contact Path is said to be Edge-Cutter Contact Path. If all the edges in the base face are convex, then the Edge-Cutter Contact Path is created outwards as shown in figure 4.4 (a) or else the Edge-Cutter Contact Path is created inwards.
2. A group of straight Cutter Contact Paths parallel to x-plane or y-plane is created throughout the surface at equal intervals with respect to the tool radius as shown in figure 4.4 (a). The portion of these paths which lies outside the Edge-Cutter Contact Path are removed.
3. All the path segments are divided into equal intervals whose coordinates (Cutter Contact Points) are extracted. The gouge-free Cutter Location Points are calculated from the Cutter Contact Points in order to generate the Cutter Location Path. As shown in figure 4.4 (c), when the end point (2) of the first segment is reached, the tool moves to (2’) in z+ direction with and at same level moves to (3’) before reaching starting point (3) of the next segment. This step is repeated until the final point of the last path segment is reached.



(a) One way tool path generation



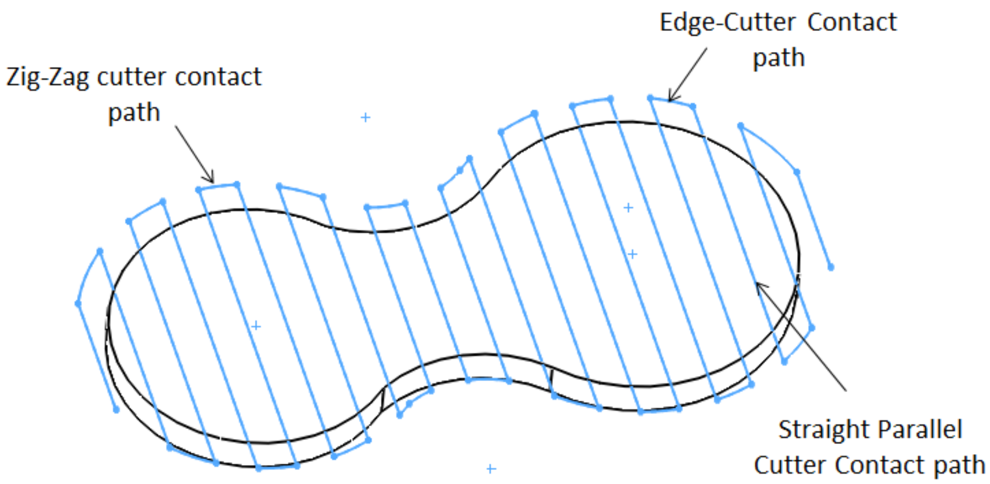
(b) One way tool motion



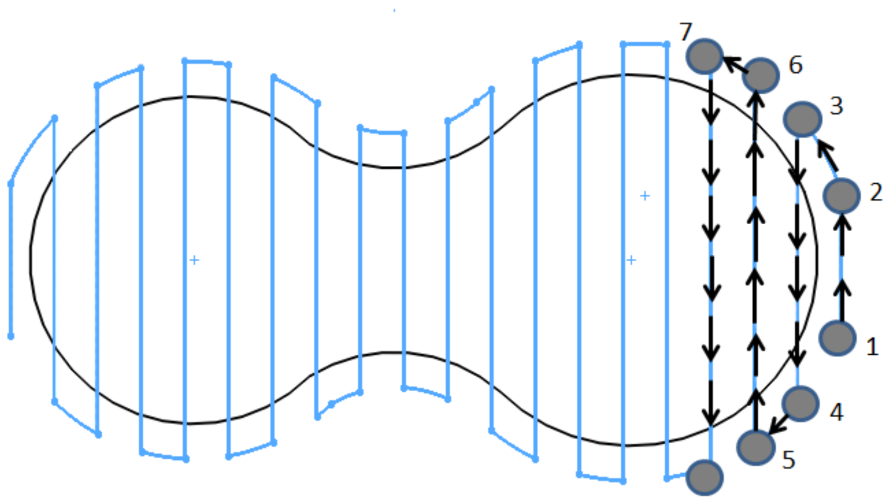
(c) Tool displacement from point to point

Figure 4.4 General one way tool path strategies shown in (a),(b) and (c)

In zig-zag parallel tool path, the Cutter Contact Paths are created in a manner similar to the one-way parallel tool path. The straight Cutter Contact Paths which are generated parallel to x-plane or y-plane are linked to the Edge-Cutter Contact Path as shown in figure 4.5 (a). The path segments are connected in zig-zag manner in order to create continuous tool movement as shown in figure 4.5 (b) with less air cutting. The generated tool paths are built at the depth parameter attained from machining step list.



(a) Zig-Zag tool path generation



(b) Zig-Zag way tool motion

Figure 4.5 General zig-zag tool path

The proposed method uses the zig-zag type tool path for machining all type of pocket features, step features and slot features which contain inner loops with boss or islands in it. The types of pocket features that accept this tool path type includes the pocket whose base face is planar and may have inner loops with both protrusion and depression features.

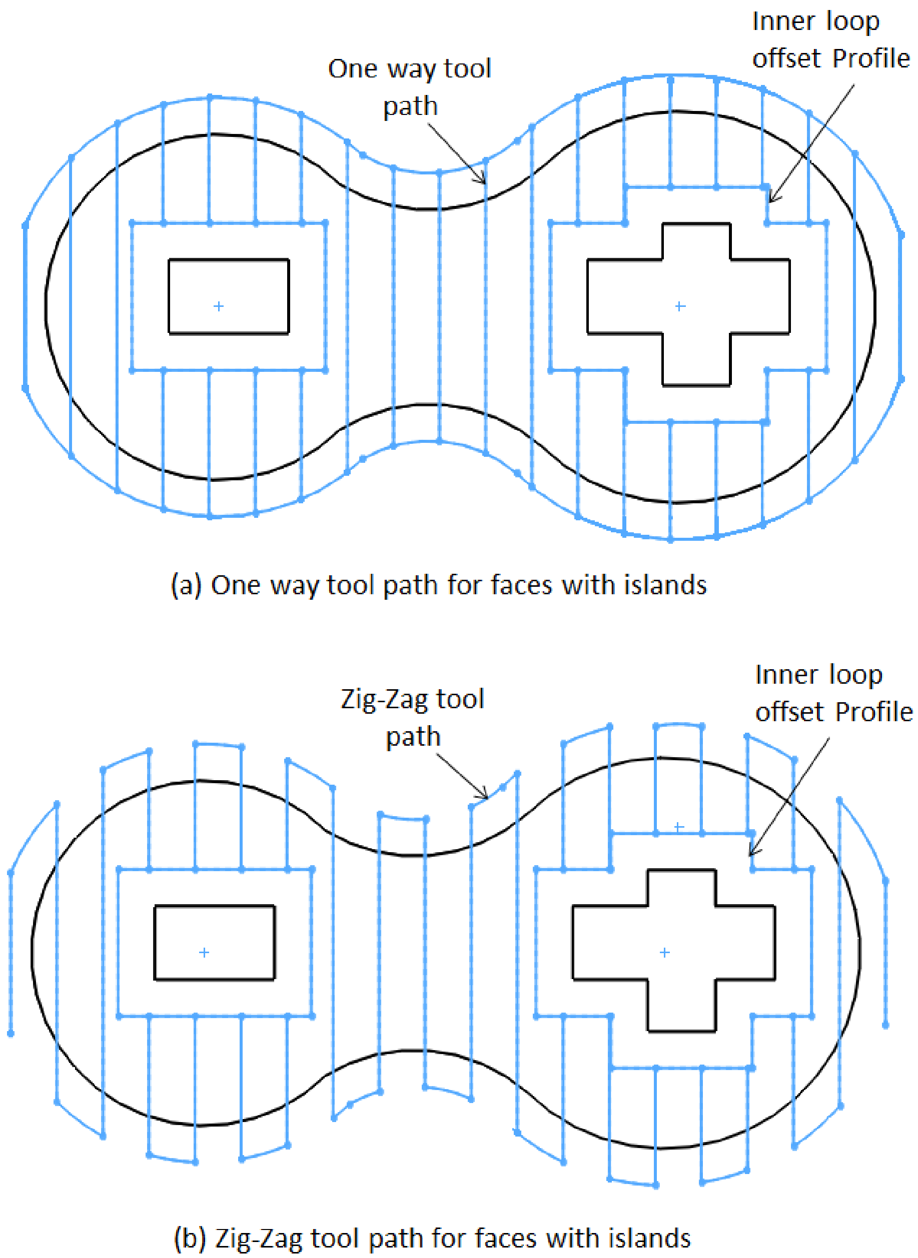


Figure 4.6 Tool paths for features with islands

If the planar base surfaces contain inner loop protrusion features, then the edges of the inner loops are extracted separately. An offset loop (Cutter Contact Path) is created with respect to tool radius by offsetting these inner loops that exist on the base face. The algorithm compares the generated one way or zig-zag Cutter Contact Path with the newly created inner loop offset profile and eliminates the path segments that lie inside the inner loop offset profile as shown in figure 4.6 (a) and (b)**.** Depression feature lies below the base surface and are not considered.

3. Machining Freeform Parametric Surfaces: The free form surfaces are widely used in building complex surfaces such as aerodynamic surfaces, molds, dies, etc. These surfaces are usually machined with ball-end cutters in three-axis machining by choosing a small incremental isoparametric curves as the tool path. Usually the parametric surfaces are represented using two parameters (u, v) and by keeping one parameter either u or v as constant, the isoparametric curves on a free-form surfaces, S (u, v) can be obtained. The surface parameters are extracted at fine and equal intervals and the coordinate at each location is also extracted. The Cutter Contact Path is generated by keeping the surface parameter, *v* as constant and the Cutter Contact points are extracted through the direction of parameter, *u*. when the extraction reaches the end parameter of '*u*' in one direction, the extraction of Cutter Contact Points continues in the opposite *u*-direction with a small increment in *v*-direction in order to generate a zig-zag parallel Cutter Contact Path. The Cutter Location Points are calculated from the Cutter Contact Points in order to generate the tool path.

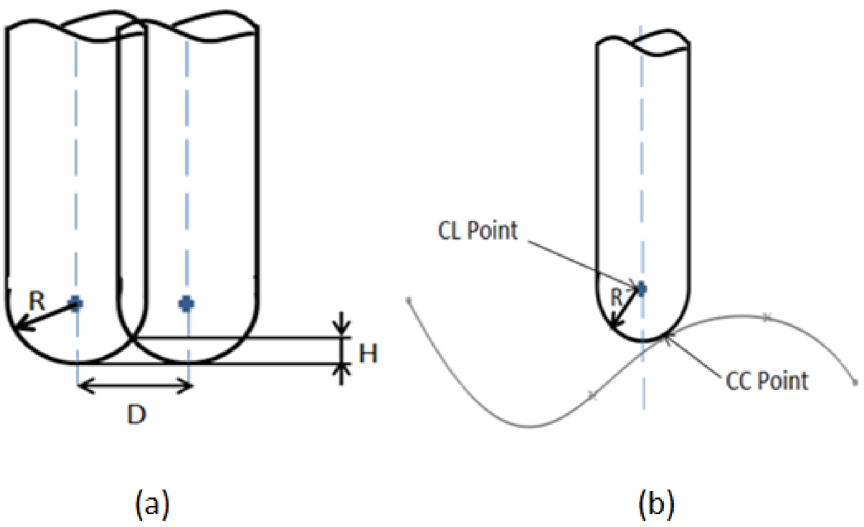


Figure 4.7 Tool path Interval and cutter contact point is shown in (a) and (b) respectively

To reduce the non-predictable scallop height and in order to maintain the surface accuracy, the smallest tool path interval must be used as the offset distance to define the constant isoparametric offset. When the cutter is following the parallel tool paths, scallops are created on the surface. The distance between these parallel tool paths depends on the cutter size, surface curvature and allowable scallop height after machining process and this interval can be termed as cutter contact path interval. By offsetting the Cutter Contact Path with the distance equal to the cutter radius, in the direction of the surface normal, the actual tool path interval can be attained. The precise tool path interval can be attained by setting the direction of tool path interval orthogonal to the tool path.

The increment in the *v*-direction depends on the tool path interval, D which is calculated by specified tolerance, H, the tool radius, R, and the radius of curvature of the surface as given in equation 5.11,

$$D = 2\sqrt{(2R - H)H} \qquad (4.1)$$

The procedure is repeated until the tool path for entire surface is generated. Thus the zig-zag tool path is implemented for roughing free form surfaces.

4.3 Cutter Contact Path to Cutter Location Path (Tool Path)

As discussed above, the various types of Cutter Contact Paths are generated on the base face. These paths are discretized into fine interval points and their coordinates are extracted. To describe the tool motion the Cutter Location Point is required. The Cutter location Point is the trajectory followed by the tool center. The Cutter Location Point in planar surfaces is identified with the uniform offset of tool radius from the Cutter Contact Point in the direction of tool axis. In non-planar surfaces, a point offset by the tool radius along the surface normal from the Cutter Contact Point becomes the Cutter Location Point. When series of Cutter Location Points are connected together forms a Cutter Location Path or tool path. Though the generated Cutter Contact paths are gouge-free from inner loop features or islands, there may exists some unpredictable tool gouge with the whole part. This gouging depends on the geometric structure of the feature and so it is said to be Feature-based Gouging. It is necessary identify this gouging and modify the tool path to generate gouge-free Cutter Location path. This is achieved by using the techniques discussed below.

The proposed overall work of feature recognition from the CAD model and feature-based tool path generation followed by machining operation sequence planning have been implemented using Solidworks API – VBA with an assembly model. This assembly model contains the given part and sample tool geometry as shown in figure 4.8.

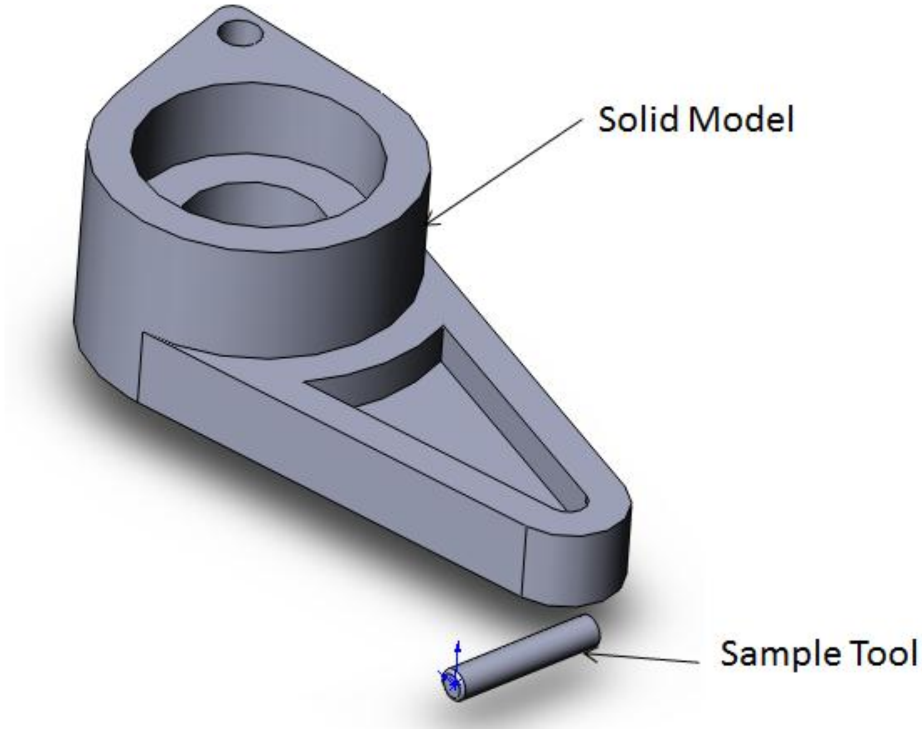


Figure 4.8 Assembly model with solid model and sample tool geometry

As the tool is positioned at the Cutter Location Point, it may gouge the part. The sample tool moves along the series of Cutter Location Points for gouge-detection. An inbuilt gouge detection tool in the solidworks API is used to detect the gouge between the tool and the part. If the gouge exists, the Cutter Location Point is tagged. The Tool Path Planning module collects and removes all the Cutter Location points that have been tagged. For example, assume a part with pocket feature on its top face as shown in figure 4.9. The contour parallel Cutter Contact Paths are generated on the base face of the pocket based on tool radius which results in overlap of paths as shown in figure 4.9(a).

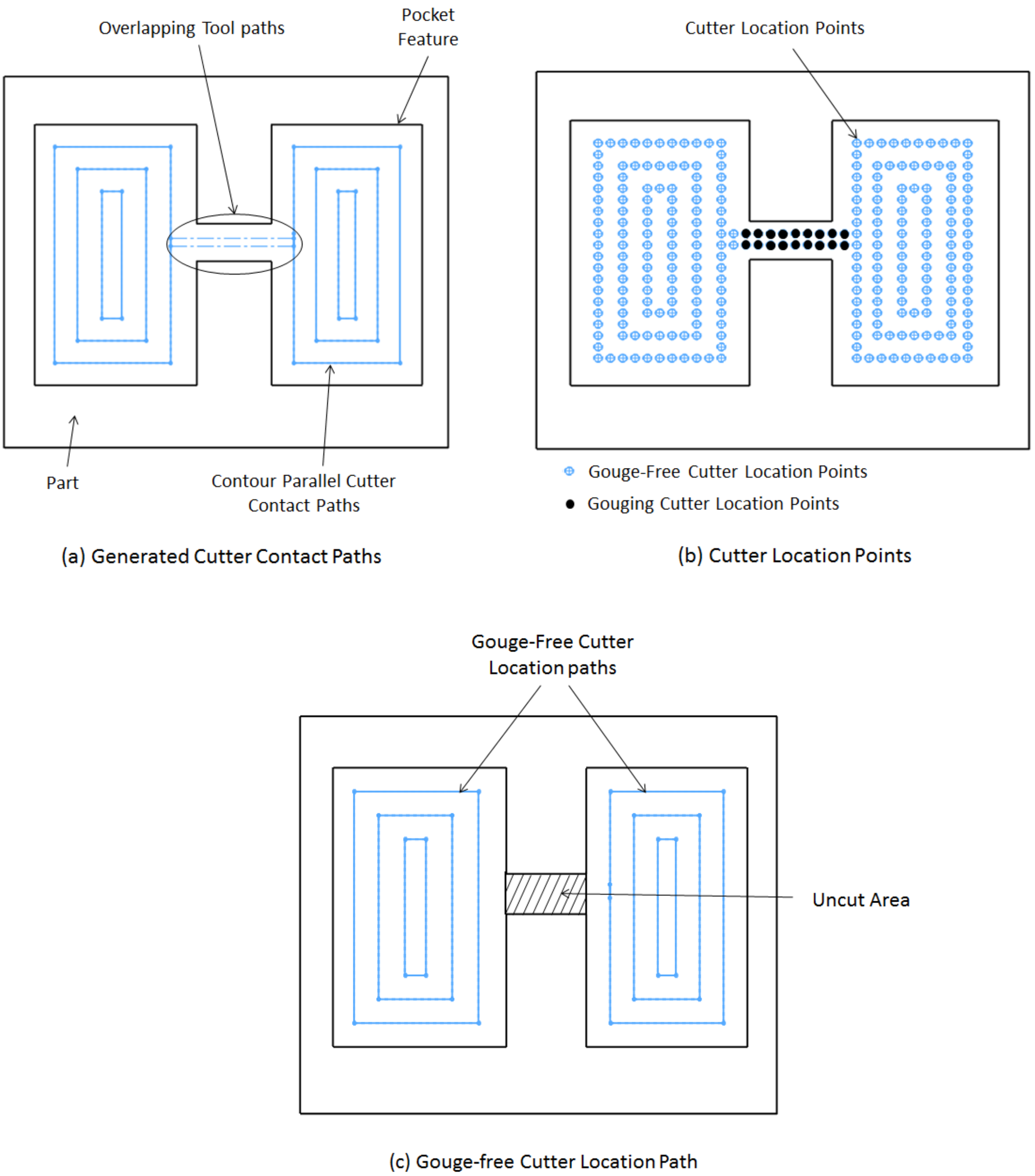


Figure 4.9 Gouge-free cutter location paths generation

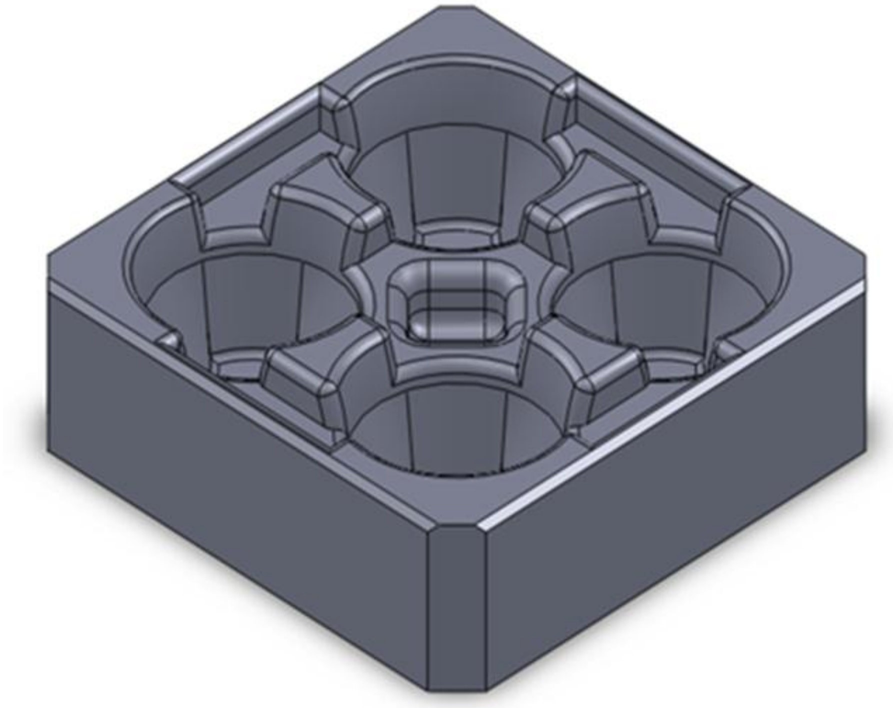
When the sample tool moves along this path, it detects and tags all the gouging Cutter Location Points as shown in figure 4.9 (b). At these tagged point the tool is raised to a safe height where no feature exists and moves to the start of the next gouge-free Cutter Location point. This process is the key for generating gouge-free tool path.

4.4 Implementation and results

As mentioned earlier, the proposed work is entirely implemented using Solidworks-API. The evolution of CAD progressed from wireframe modeling to surface modeling to solid modeling. Solid modellers build a database of geometry and topology associated with the part. This allows the users to access the geometry and topology through an algorithm interface and use the results of these geometric queries to develop automatic design and reasoning algorithm. This ability of solid modeling systems are availed through an enhanced application programming interface (API). An API allows automated design and design based on rules. An API also offers designers and manufacturers to algorithmize the commonly performed steps using a CAD system and with these algorithms the steps can be repeated automatically.

The proposed work is implemented and tested on various solid models. The figure 4.10 shows two sample solid models used to test the algorithm. The 'solid model 1' in figure 4.10(a) shows a model with simple features with planar base faces and the 'solid model 2' in figure 4.10(b) shows a model with complex features including non-planar surfaces.

(a) Solid Model 1



(b) Solid Model 2

Figure 4.10 Sample solid models

The various sequential operations that take place during efficient tool path planning by the proposed modules for the ‘solid model1’ and ‘solid model 2’ are shown in the figure 4.11 and figure 4.12 respectively. The figure 4.11 (a) shows the various features identified from the ‘solid model 1’ by the Feature Recognition module and it is segregated according to the machining operation i.e. milling or drilling. Next a minimum bounding box is created by the Process Planning module as shown in figure 4.11(b). It is followed by generation of contour parallel Cutter Location Paths or tool paths above the bounding box as shown in figure 4.11(c) in order to remove the excess material from the stock. Then for each feature in the machining step list, gouge-free Cutter Location paths are generated as shown in figure 4.11 (d), (e), (f) and (g). Finally a series of contour parallel tool paths are generated along the depth of the model for removing material along the boundary as shown in figure 4.11(h).

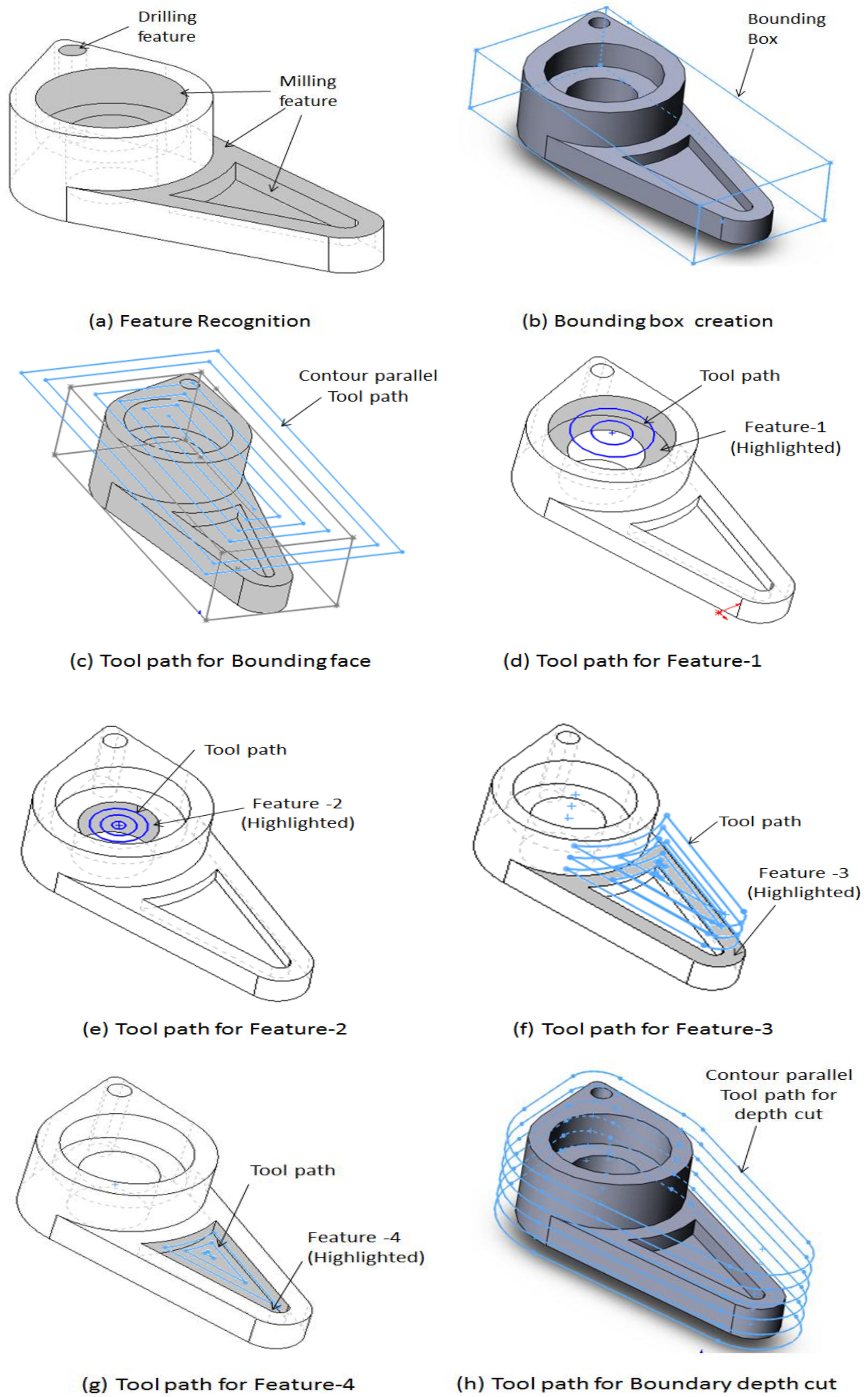


Figure 4.11 Sequential operation for Solid model-1

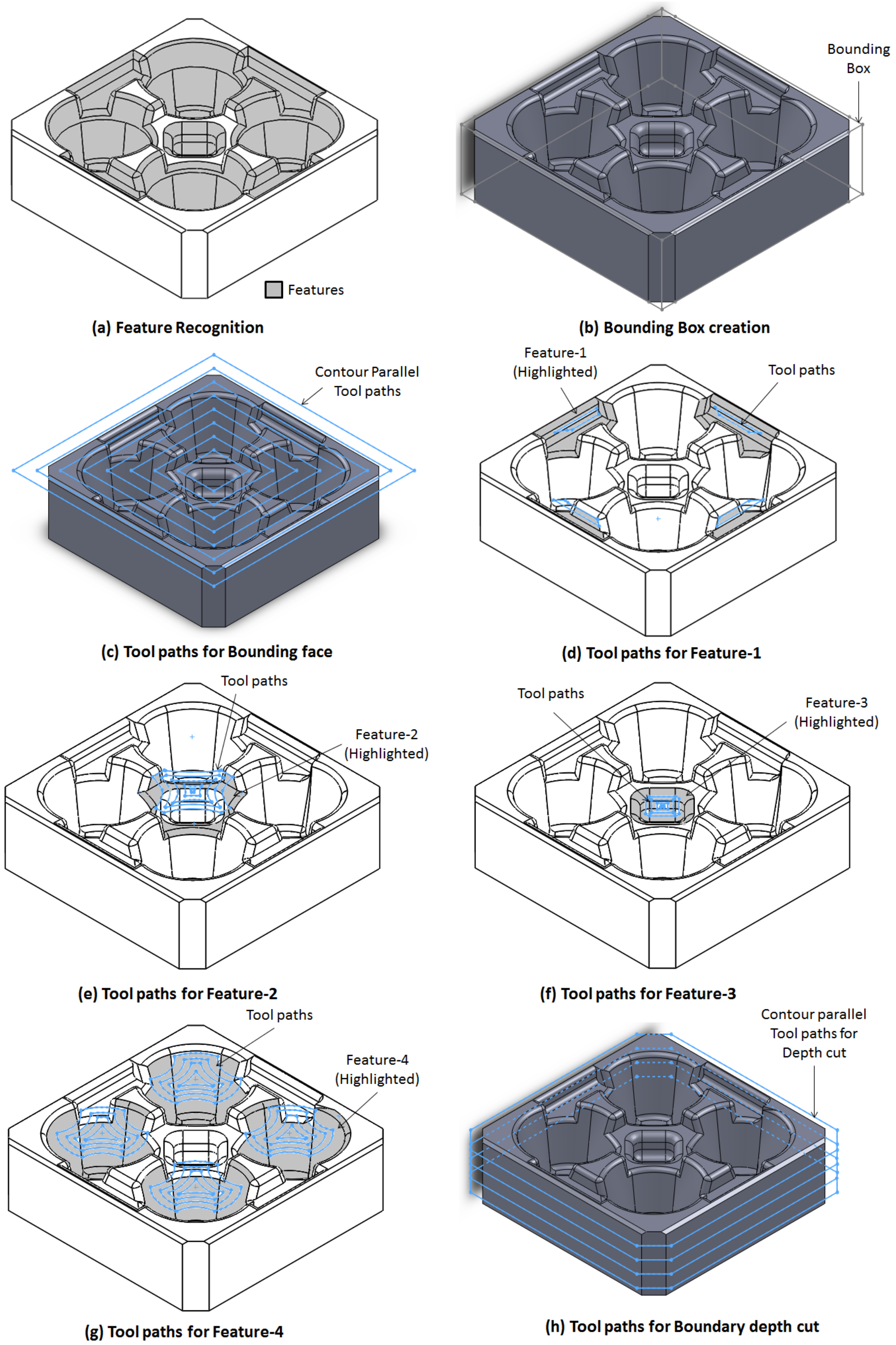


Figure 4.12 Sequential operation for Solid model-2

The figure 4.12 (a) shows the various features identified from the ‘solid model-2’ by the Feature Recognition module. Next a minimum bounding box is created by the Process Planning module as shown in figure 4.12(b). It is followed by generation of contour parallel Cutter Location Paths or tool paths above the bounding box as shown in figure 4.12(c) in order to remove the excess material from the stock. Then for the each feature in the sequence list, gouge-free Cutter Location paths are generated as shown in figure 4.12 (d), (e), (f) and (g). All the filleted faces in a feature are machined individually during machining the feature. Finally a series of contour parallel tool paths are generated along the depth of the model for removing material along the boundary as shown in figure 4.12 (h).

Limitations

Though the proposed work is generic and flexible for various types of parts, it has some limitations as the proposed work concentrates only on 3-axis machining. This method works efficiently only on features whose faces can be reached by the tool. For example, the figure 4.13(a) shows the base face with Cutter Location Path on it. The tool follows this path to remove material without any gouging. But in the case of features with occluded base faces as shown in figure 4.13(c), gouging may occur. As mentioned above, the algorithm detects the gouging Cutter location Points and generates new gouge-free Cutter Location Path. For example, the figure 4.13(e) shows that the tool follows the new gouge-free Cutter Location Path and stops at point ‘P’ in order to avoid gouging. The part of base face that lies under the inclined face as shown in figure 4.13(f) cannot be reached by the tool in 3–axis machining which remains as an uncut area.

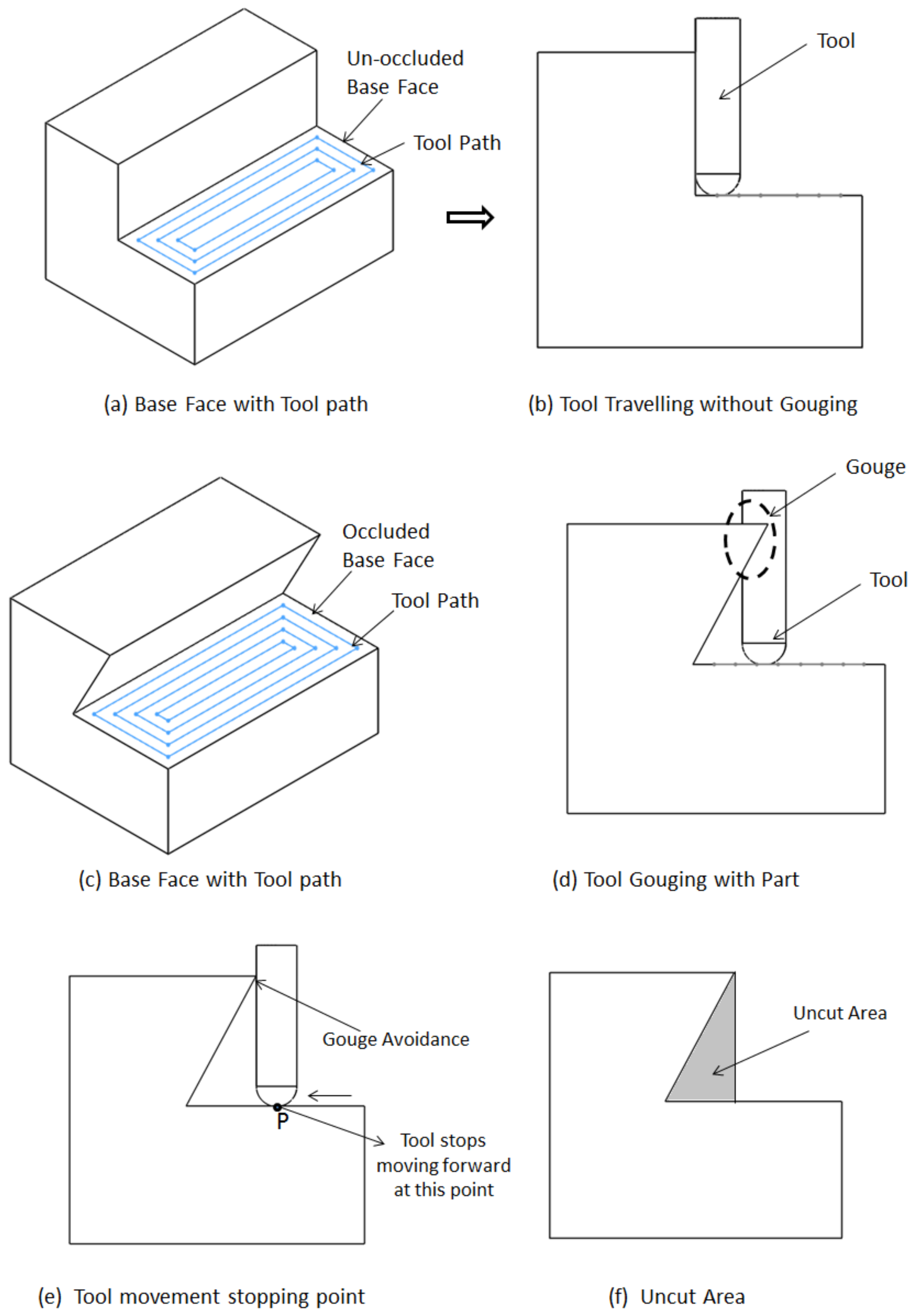
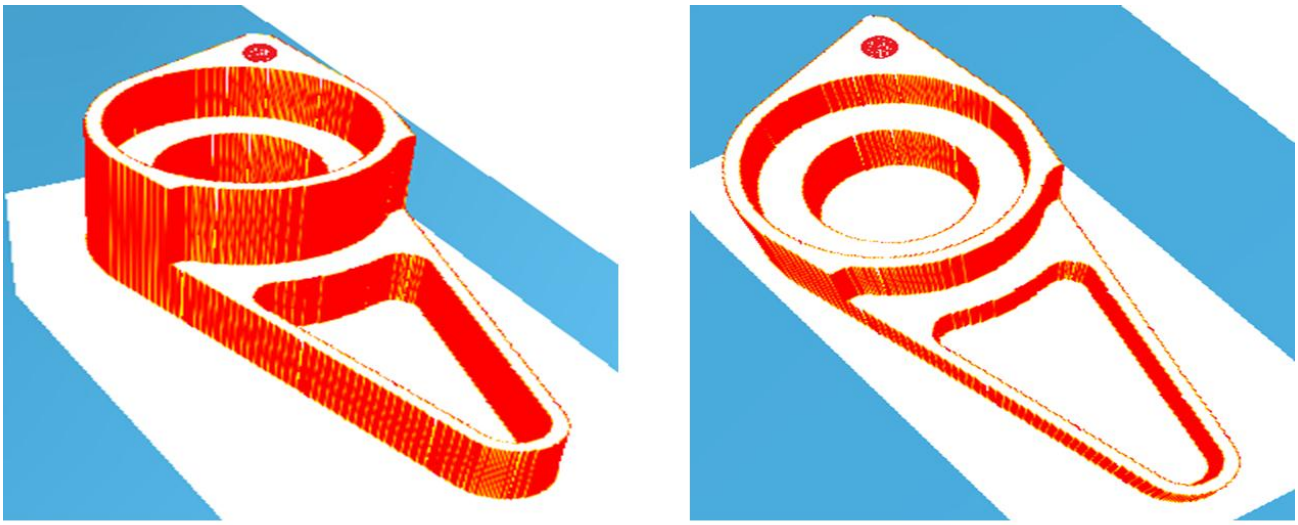


Figure 4.13 Limitations of proposed tool path strategy

4.5 Simulation results

The proposed feature-based tool path planning generates an accurate tool path for both roughing and finishing operation, which is simulated using a machining simulator to verify the effectiveness of the manufacturing process and the tool path. Many manufacturing engineers use CNC simulators to verify the tool path before using in the CNC machine in order to reduce the cost and to minimize the errors that occurs during machining. In this work, the simulation process were carried out using a custom machining simulator **ToolSim** developed by Israeli **[20].** ToolSim is a machine model with a graphical interface which behaves much similar to an actual CNC milling machine. ToolSim was used to simulate the machining process for the generated tool path, and the simulation error seems to be within limits as specified in Mann et al **[21].**

The tool path generated by the proposed module for the example solid models shown in figure 4.11 (a) and (b) are tested using ToolSim. The results obtained from the ToolSim simulator is shown in figure 4.14 and 4.15 respectively.



(a) (b)

Figure 4.14 Simulation results for Solid model-1

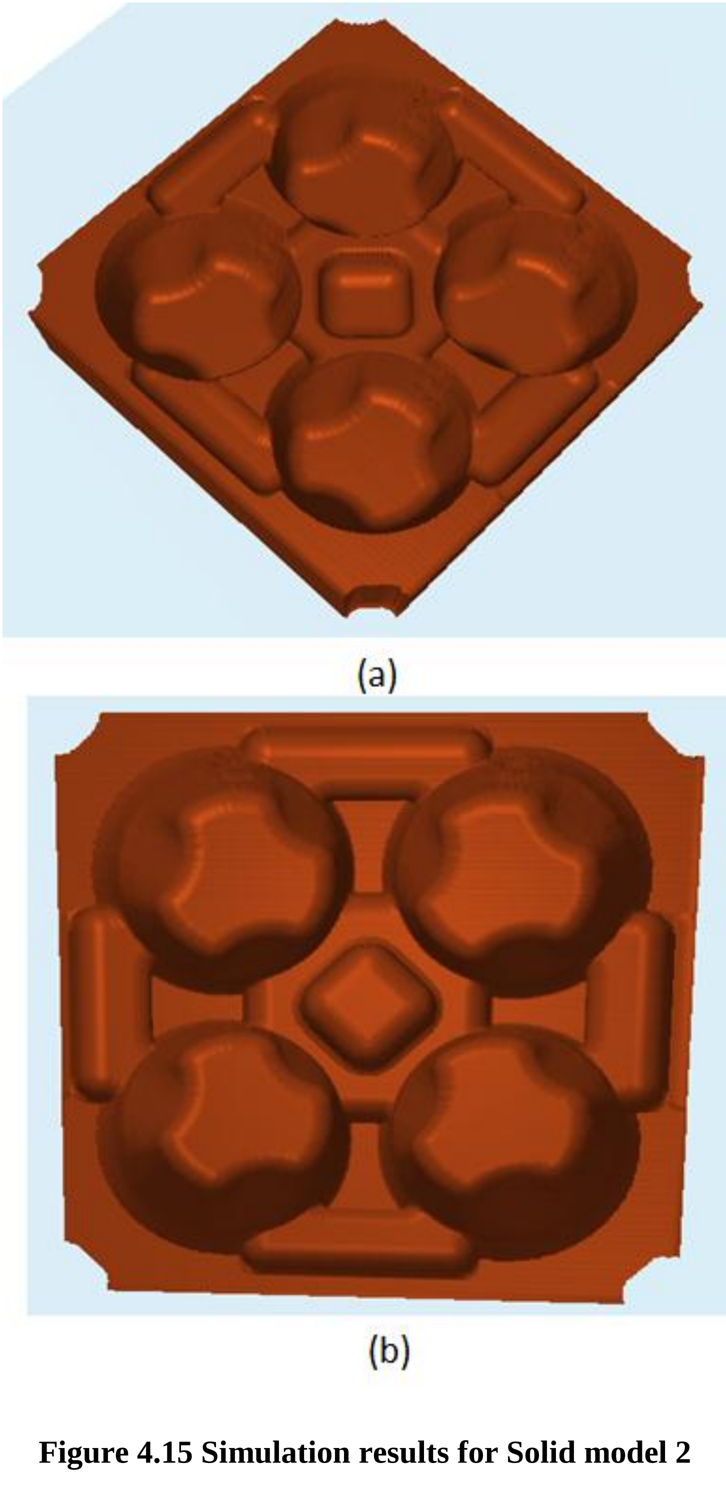


Figure 4.15 Simulation results for Solid model 2

The tool path generated in solidworks is checked in ToolSim in order to verify the effectiveness of the tool path and machining sequence. The simulation results obtained from the ToolSim seems to be same and accurate when compared with the input CAD model.

Chapter 5

Conclusion

A technique for feature-based automatic tool path planning is proposed, implemented and tested in this work. Feature Recognition and Feature-based Machining has been an active research area for past few decades. It plays a vital role in the integration of design and manufacturing process. Researchers have used various methods for Feature Recognition and Tool Path Planning. All these methods work on triangulated models. The use of triangulated models amounts to throwing away the topology and geometry information stored in a B-rep model. This loss of information has caused difficulty in identifying complex features such as interacting features and transition features such as chamfers, filleted edges etc. The parameter extraction for algorithmic reasoning such as in tool path planning and process planning from the features has also become a challenging task. Planning machining sequence is not easy unless the feature parameters and their locations on the part and geometry are known. The method proposed in this work uses B-Rep model which is embedded with geometry and topology information. This information helps and extends the capability of identifying features from the solid model. The proposed Feature Recognition module recognizes various types of features including transition features and interacting features from the solid model. The module also extracts the parameters of all the identified features. This advantage of B-Rep models paves efficient way for planning machining sequence based on the extracted parameters and the location of the feature.

The main goal of this work is to show that B-Rep models, which include the part geometry and topology, can be used to algorithmically add Tool Path Planning and Process Planning information to a part design, thereby introducing efficiency in the manufacturing process. Most of the studies discussed in the literature only present in concept methods to integrate design with machining. Some works have implemented feature recognition concepts in software. Those who have implemented ideas have focused either in feature recognition or on tool path planning. The separation of focus between feature recognition and tool path planning has made it harder to do tool path planning for parts which have interacting features and transition features like chamfers and fillets. In addition, most of the research on automatic tool path planning applies to triangulated models only. Planning gouge-free tool path for the interacting features still remains a challenging task. Though few works [33][43] have proposed methods for planning gouge or interference free tool path, these works are confined to specific family of parts and cannot be scaled. In this work, the part definition is not simplified into a triangulated model before use in Feature Recognition. The choice to recognize features from the B-Rep definition gives the proposed method flexibility to deal with transition feature like fillets and chamfers. Example, section-3.5 in chapter-3 demonstrates this capability.

In addition, the proposed method introduces the concept of Base planes. Base planes separate out the regions of machining. This separation allows different tool to be used in different area to take advantage of the feature shape and tool ability. The base planes help in sequencing the machining process. The sequencing ensures that the material covering a feature is removed before the feature can be machined. This method was demonstrated on two parts in chapter-4. In machining of features, the base planes separate the tool paths for the various features. This separation allows different features to be machined with different tool paths such as contour parallel, one-way parallel and zig-zag parallel tool path. The proposed process also divides the Tool Path Planning into Cutter Contact Path and Cutter Location Path. This division helps to detect and avoid any gouge that may occur when two features are close to one another. This strategy was discussed in chapter-4. Overall this work shows that automatic algorithmic extraction of machining information is possible and can lead to improve efficiency in the manufacturing process.

5.1 Future Consideration

The results obtained from this research are encouraging, but confined to only 3- axis machining operations. Future research will include ways in which these methodologies can be integrated successfully into 5-axis machining process. Further development could be focused on:

1. Optimization of feature recognition module for recognizing more complex features.
2. Generate new tool path and process planning strategies for 5-axis machining process.

Bibliography

[1] Paolo Di Stefano, Francesco Bianconi, Luca Di Angelo. An approach for feature semantics recognition in geometric models. Computer-Aided Design, 2004: 993~1009

[2] Madurai, S. S., & Lin, L. (1992). Rule-based automatic part feature extraction and recognition from CAD data. Computers and Industrial Engineering, 22(1), 49–62.

[3] Fu, M. W., Lu, W. F., Ong, S. K., Lee, I. B. H., & Nee, A. Y. C. (2003). An approach to identify design and manufacturing features from a data exchanged part model. Computer Aided Design, 35(11), 979–993.

[4] Lin, A. C., Lin, S. Y., Lin & Cheng, S. B. (1997). Extraction of manufacturing features from a feature-based design model. International Journal of Production Research, 35(12), 3249–3288.

[5] Shah, J., Anderson, D., Kim, Y., Joshi, S., (2001). *A Discourse on Geometric Feature Recognition from CAD Models*. Journal of Computing and Information Science in Engineering, Vol. 1, pp. 41-51.

[6] Nasr, E. S. A.; Kamrani, A. K.: A new methodology for extracting manufacturing features from CAD system, Computers & Industrial Engineering, 51, 2006, 389–415.

[7] Bedi S., Ismail F., Mahjoob M. J., and Chen Y. Toroidal versus ball nose and at bottom end mills. International journal of Advanced Manufacturing Technology, 13:326-332, 1997.

[8] Li, W.D., Ong, S.K. and Nee, A.Y.C. (2006) *Integrated and Collaborative Product Development Environment - Technologies and Implementation*, World Scientific Publisher.

[9] Lin, A. C., & Lin, S-Y. (1998). Volume decomposition approach to process planning for prismatic parts with depression and protrusion design features. International Journal of Computer Integrated Manufacturing, 11(6), 548–563.

[10] Kao, C-Y., Kumara, S. R. T., & Kasturi, R. (1995). Extraction of 3D object features from cad boundary representation using super relation graph method. IEEE Transaction on Pattern Analysis and Machine Intelligence, 17(12), 1228–1233.

[11] Joshi, S., & Chang, T. C. (1988). Graph-based heuristics for recognition of machined features from 3D solid model. Computer Aided Design, 20(2), 58–66.

[12] Kang, T. S., & Nnaji, B. O. (1993). Feature representation and classification for automatic process planning systems. Journal of Manufacturing System, 12(2), 133–145.

[13] Kayacan, M. C., & Celik, S. A. (2003). Process planning system for prismatic parts. Integrated Manufacturing Systems, 14(2), 75–86.

[14] Das, D., Gupta, S.K. and Nau, D.S., Estimation of setup time for machined parts: accounting for work holding constraints using a vise. Proc. ASME Computers in Engineering Conference, Boston, MA, 1995, 619–631.

[15] Hirode, K. and Shah, J.J., Metrics for Evaluating Machining Process Plans, DETC99-DFM-8931. Proc. of the 1999 Design Engineering Technical Conference, 1999 Las Vegas, NV.

[16] Y.-S.Lee and H. Ji (1997) Surface interrogation and machining strip evaluation for 5-axis CNC die and mold machining. International Journal of Production Research, 35,1, 225-252.

[17] Han,J., & Han,I. (1999). Manufacturable feature recognition and its integration with process planning. Proceedings of the Symposium on Solid Modeling and Applications, 108–118.

[18] Harun, W. A., & Case, K. (2000). Feature-based representation for manufacturing planning. International Journal of Production Research,38(17), 4285–4300.

[19] Choi, B. and Jerrard, R. (1998). Sculptured Surface Machining: Theory and Applications. Kluwer Academic Publishers, The Netherlands, First edition.

[20] Israeli Gilad. Software simulation of numerically controlled machining. Master's thesis, University of Waterloo, 2006. 81

[21] Mann S., Bedi S., Israeli G., and Zhou X. Machine models and tool motions for simulating five-axis machining. Computer-Aided Design, 42:231-237, March 2010. 81,82

[22] Subrahmanyam, S., & Wozny, M. (1995). An overview of automatic feature recognition techniques for computer-aided process planning.Computers in Industry, 26(1), 1–21.

[23] Roucoules, L., Salomons, O., & Paris, H. (2003). Process planning as an integration of knowledge in the detailed design phase. International Journal of Computer Integrated Manufacturing, 16(1), 25–37.

[24] Venkataraman, S., Sohoni, M., & Kulkarni, V. (2001). A graph-based framework for feature recognition. Proceedings of the Symposium on Solid Modeling and Applications, Ann Arbor, MI, pp. 194–205.

[25] Rozenfeld, H., & Kerry, H. T. (1999). Automated process planning for parametric parts. International Journal of Production Research, 37(17), 3981–3993.

[26] Nagaraj, H. S., & Gurumoorthy, B. (2002). Machinable volume extraction for automatic process planning. IIE Transactions, 34(4), 393–410.

[27] Zhao, Y., Ridgway, K., & Al-Ahmari, A. M. A. (2002). Integration of CAD and a cutting tool selection system. Computers and Industrial Engineering, 42(1), 17–34.

[28] C.-S. Jun, D.-S. Kim, S.Park, A new curve-based approach to polyhedral machining, Computer-Aided Design 34 (5) (2002) 379-389.

[29] G.Elber, “Freeform surface region optimization for 3-axis and 5-axis milling”, Computer-Aided Design, Volume 27, No.6, 1995, pp, 465-469

[30] J.Hwang, T.-C.Chang, Three- axis machining of compound surfaces using flat and filleted endmills, Computer-Aided Design 30 (8) (1998) 641-647.

[31] Choi, B K ‘CAD/CAM compatible tool oriented process planning for machining centers’ PhD Thesis Purdue University, West Lafayette, IN, USA (1982)

[32] K. I. Kim and K. Kim, “A new machine strategy for sculptured surfaces using offset surface”, International Journal of Production Research, 33(6), pp. 1683-1697, 1995.

[33] A. Hatna, R. J. Grieve and P.Broomhead, “offsetting 3D contourson parametric surfaces”, International Journal of Advanced manufacturing Technology, 16, pp. 189-195, 2000.

[34] J. S. Hwang, “Interference-free tool-path generation in the NC machining of parametric compound surfaces”, Computer-Aided Design, 24(12), pp.667-676, 1992.

[35] R.-S. Lin and Y. Korean, “Efficient tool-path planning and machining for free-form surface”, Journal of Manufacturing Science and Engineering, 118, pp. 20-28, February 1996.

[36] Misra, Debananda; Sundararajan, V.; Wright, Paul K.: ZigZag Tool Path Generation for Sculptured Surface Finishing, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, 2003

[37] G.Farin. (1992) Curves and surfaces for CAGD. Academics press

[38] Y.-S.Lee and H. Ji (1997) Surface interrogation and machining strip evaluation for 5-axis CNC die and mold machining. International Journal of production research,35,1, 225-252

[39] G.Salas Bolanos, S.Mann, and S.Bedi. Targeted webcad. Computer Aided Design and Applications, 2009

[40] T.Kandala, Implementation of User-defined features in web-based CAD applications. Master's thesis, University of Waterloo, 2009.

[41] K.Patel, G.Salas Bolanos, R.Bassi, and S.Bedi, Optimal tool selection. pp.19-64, 2010

[42] K.B.Mawussi, & L.Tapie,. A knowledge based model for complex forging die machining. Computers and Industrial Engineering, 61(2011), 84–97.

[43] K.patel, Web Based Automatic Tool Path Planning Strategy for Complex Sculptured Surfaces. Master's thesis, University of Waterloo, 2010.