AUTOMATIC SPOT WELDING FEATURE RECOGNITION FROM STEP DATA

By

NS Muhammad Ali Kiani

00000170904

MS-16

Submitted to the Department of Mechanical Engineering in Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE

In

MECHANICAL ENGINEERING

Thesis Supervisor

Hasan Aftab Saeed, PhD

College of Electrical & Mechanical Engineering

National University of Sciences & Technology

2019

AUTOMATIC SPOT WELDING FEATURE RECOGNITION FROM STEP DATA

Author

Muhammad Ali kiani

00000170904

A thesis submitted in partial fulfillment of the requirements for the degree of

MS Mechanical Engineering

Thesis Supervisor:

Dr. Hasan Aftab Saeed

Thesis Supervisor's Signature: ____

DEPARTMENT OF MECHANICAL ENGINEERING

COLLEGE OF ELECTRICAL & MECHANICAL ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY, ISLAMABAD

July, 2019

بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ

In the name of Allah, the most Beneficent and the most Merciful

Declaration

I hereby affirm that this thesis Title "**AUTOMATIC SPOT WELDING FEATURE RECOGNITION FROM STEP DATA”** is absolutely based upon my own personal hard work under the valuable guidance of my supervisor Dr Hasan Aftab Saeed. The Contents have not been plagiarized and sources used are cited. No part of work presented in thesis has been submitted in favour of any application of other degree of qualification to this or any other university or institute of learning.

………………………

Muhammad Ali Kiani

LANGUAGE CORRECTNESS CERTIFICATE

This thesis has been read by an English expert and is free of typing, syntax, semantic, grammatical and spelling mistakes. Thesis is also according to the format given by the university.

Signature of Student
NS Muhammad Ali Kiani
00000170904

Signature of Supervisor
(Dr Hasan Aftab Saeed)

PLAGIARISM CERTIFICATE (TURNITIN REPORT)

This thesis has been checked by plagiarism in turnitin. The report countersigned by respective supervisor is attached along with.

Signature of student
Muhammad Ali kiani
00000170904

Signature of supervisor
Dr Hasan Aftab Saeed

Copyright Statement

ACKNOWLEDGEMENT

First and foremost, I would like to thank my Almighty Allah for all His countless blessings and benevolence which embraced me to complete this incredible task in terms of integrity and completeness.

Secondly, there is long list of people without whom I would have not been able to complete this assignment successfully. I would like to communicate special recognition and gratitude to my advisor,

Dr. Hasan Aftab Saeed for his extended support throughout my thesis tenure. His untiring efforts and advice helped me to move around every predicament during my research work. Secondly, I would also like to pay special thanks to my co advisor **Dr. Zahid faraz** whose guidance served as beacon of light and inspiration throughout my research work.

I would also like to thanks Assistant Professor **Dr. sajidullah Butt** and **Dr Naveed Akmal Din**, for serving as my Guidance and Evaluation committee even at hardship. I obliged for letting my defense as enjoyable moments and for brilliant comments and suggestions.

I would also like to thanks my all friends who help me in this regard, specially Umar Qureshi whose tireless effort will be a continuous source of inspiration for me in the entire thesis.

Special thanks to my **family**; **my mother**, **father** and siblings for all the sacrifices that they made and who were always in my support in dire moments. Conclusively, I would like to thank the entire honorable faculty of Department of Mechanical Engineering, whose professional approach and vision groomed me as a sound person both technically and morally.

Muhammad Ali Kiani , 2016

Dedication

- I dedicate my thesis to my family and friends. A special feeling of gratitude to my loving parents, whose words of encouragement were always there in my moral support.

- I would place sincere thanks to Dr. Hasan Aftab Saeed for his guidance and support throughout this study, and specially his confidence in me.

- I pray ALLAH, for always being kind by way His countless blessings.

Abstract

In design and manufacturing many systems are used to manage technical product data. Each system has its own data formats so the same information has to be entered multiple times into multiple systems leading to redundancy and error. Generic format is needed to overcome this misunderstanding. Geometric models with complete design data details are created through CAD software and translated into an equivalent boundary representation(B-Rep) format in the form of neutral file. These neutral files can be transported for further analysis or process planning. Automatic feature recognition is the process of significant information extraction from neutral file and identifying the welding features without any human intervention. Automatic feature recognition is an important aspect for CAPP and it is an important task between CAD and CAPP. CAPP is the bridging gap between CAD and CAM. Automated feature recognition is considered a critical node for integration of CAD/CAPP/CAM. Research in feature recognition has received significant attention, however, the majority of feature recognition systems are for machining features.

Present work is an attempt to develop a method for automatic feature recognition of spot welding from neutral file, STEP (AP203). which is an international standard for geometric and non-geometric data transfer between heterogeneous systems. This study explains the software Development in Python language for feature recognition which work as an integration between CAD and CAPP. The system developed uses a simplified and generalized methodology of extracting spot welding feature information from STEP files. Three dimensional Assembly with multiple spots weld created by using Creo 3.0 software to recognize features. Three main features of Spot welding are highlighted (1) Total Number of spot welds (2) Coordinates and Diameter of each spot weld (3) plane of weld. Rule based technique is used for recognizing features. The software has been implemented and tested on several components. The results are satisfactory and successful. An illustrative case studies is presented to test and validate the method.

Table of Contents

List of Figures

List of tables

LIST OF ACRONYMS

STEP	standard for the product Data exchange
CAD	Computer Aided Design
CAM	Computer Aided Manufacturing
CAPP	Computer Aided Process Planning
AP	Application Protocols
B-Rep	Boundary representation
AFR	Automatic feature recognition
WFR	Welding feature Reader
CSG	(constructive solid geometry)

Chapter 1

INTRODUCTION

Decades of research have been done on recognizing features from solid model. Features play a significant role to build relationship between CAD/CAM. Explanation of features by Pratt and Wilson [1] in 1985 gave a pretty wider and unique definition; 'A feature is a section of importance on the part surface'. Features can be categorized into different types i.e welding features, machined features, assembly features, fixturing features and the analysis features depending on the application. Informally, features are a part's generic forms or other characteristics with which engineers can combine helpful understanding to reason about the part. Many researches have been done on recognizing the machining features based on the CAD model. Whereas automatic recognition of welding feature are recognized as separate islands and it is a grey zone in field of feature recognition. The notion of characteristics has been a significant challenge in the automation of welding devices and remains a crucial study task. For many years, Automatic Data extraction has been important region of parametric modeling studies and is regarded as a vital job for CAD / CAM automation.

Welding automations are the biggest research area right now in the feature recognition Welding automation offers improved productivity and welding consistency to reduce manufacturing, labor and material expenses. Applications will most benefit from automation if the weld's quality or function is critical; if repetitive welds have to be made on identical components; or if the components have accumulated significant value before welding. Four primary benefits are offered by automated welding systems: enhanced welding quality, increases. In this research extraction of Welding features has been done to make the process plane more efficiently for automatic welding

1.1 Aim and Goal

Nowadays the shortage of expert employees in manufacturing industry is a common problem .It is also impossible to achieve completely automated systems without preventing human intervention .In an integration of CAD / CAM system, a straight linkage is created between design and all the manufacturing pprocesses.

It incorporates product data which is produced during process of design, for example, geometry information, material details, bill of materials, tool selection, tolerance, assemble sequence and so on., just as additional data needed for manufacturing. The capability to inevitably perceive welding features from file used for exchanging information is significant approach for CAD/CAM coordination due to significance of CAD/CAM integration the significant data of welding has gotten impressive consideration todays.

Author believes that Welding is a grey areas regarding data extraction from STEP file. In this study work has been done on the feature recognition of spot welding.

These are as follows:

1-Though automation of Welding machine has concerned, it remains mainly vendor-specific. There is no vendor-neutral approach for process planning welding.

2-Many software's or algorithms are used today to extract the features from the STEP file but all of them are vendor specific and all are related to machining features. we work on the data extraction from STEP file for spot welding and make a generic plat form for manufacturer to extract the welding data and make process plan more efficiently. with the help of this study we easily recognize the welding coordinates from the STEP file where the spot weld occurs and also extract the diameter of the weld and send this information to the any automated welding machine to start welding on the specific point on the surface and at a specific plane. Welding will be technology independent.

3- Data incompatibility is also reduced by using STEP neutral format for data extraction.

In design and manufacturing , numerous frameworks remain utilized to cope specialized technical information. Every framework has its own information centers so a similar data must be entered on different occasions into different systems prompting repetition and blunders. The issue isn't remarkable to assembling however progressively intense in light of the fact that structure information is unpredictable and 3D prompting expanded degree for mistakes and mistaken assumptions between administrators. The National Institute of Standards (NIST) has assessed that information inconsistency is a 90-billion-dollar issue for the industry which manufacture mechanical parts.

4- To demonstrate the framework by taking curve shaped having three spots weld with different diameter on different coordinates axis and planes.

For many years, automatic feature recognition (AFR) is dynamic area of studies in solid modeling and is regarded a stern job for integration between CAD / CAM.

In this research a generalized method is proposed to extract the significant information from the vendor neutral data exchange format.

1.2 FEATURE BASED CAD/CAM INCORPORATION

A research has been started enthusiastically in the recent twenty years (20) for the advancement of workstations and different software's has been made so that we can recognize features having genuine significance from 2D pictures or 3D configuration prototypes. Actually the data extraction include acknowledgment that can consequently recognize Geometry characteristics for some assembly or additional reason. The data could be collected as information of assembly, welding information, gathering features and material information. Different kinds of mix information can be present in neutral format in the form of topological and geometrical from. Assembly features tells us where the actuall part is present in the assembly and material features gives us the infromation of the material used. Whereas welding features gives the information of welding coordinates of weld, diameter of weld and also tells the plane of weld.

In 3D designing two primary exemplification arrangements, one of them is Boundary representation (B-Rep) and the other is Constructive Solid Geometry (CSG) . In current thesis we take B-Rep shape where we extract or isolated information as CAD features and CAM features. CAD structures are explicit spatial bits That's certain view point related to there utility perspective, though Production characteristics are volume extraction or joining the metal with help of welding from the original inventory to the final product.

To make system automated we used numerous neutral CAD/CAM formats list of them are , Stereo Lithography (STL), Drawing Exchange Format (DXF), Initial Graphics Exchange specification (IGES), Standard for Exchange of Product (STEP) and other neutral fromats. IGES and STEP both are the popular and prevalent vendor neutral files for the data exchange. STEP contain complete information regarding product and IGES contains information regarding the geometry of the product. Storage space needed for STEP is less than IGES.

Different kinds of information can be present in the part, when we modelled them which we used at later stages in the industry.

In the present work, the parts is modelled in Creo 3.0 and than translate the data into STEP file from where we can extract the features data of welding whose named is “STEP to Feature recognition” which we called welding feature reader.

1.3 MINIMUM HARDWARE AND SOFTWARE SPECIFICATIONS TO INSTALL AND EFFECTIVELY OPERATE

Table 1.1 provides prerequisite of Workstation to run the developed computer program.

Table 1.1: Workstation Limitation for the execution of program

Hardware	Software
Intel(R) Core(TM) i7-3470QM 2.70 GHz PC or above processor	python
8 GB RAM or above	Creo 3.0
Hard disk space 500GB or above	

1.4 OUTLINE OF THE PROPOSED SYSTEM

Our aim goal is to make a computer program to detect spot welding information from neutral format. This algorithm is generated by using ‘python’ linguistic. Three main features of Spot welding are highlighted in this software (1) Total Number of spot welds (2) Coordinates and Diameter of each spot weld (3) plane of weld

Figure 1.1 shows the structure of computer program

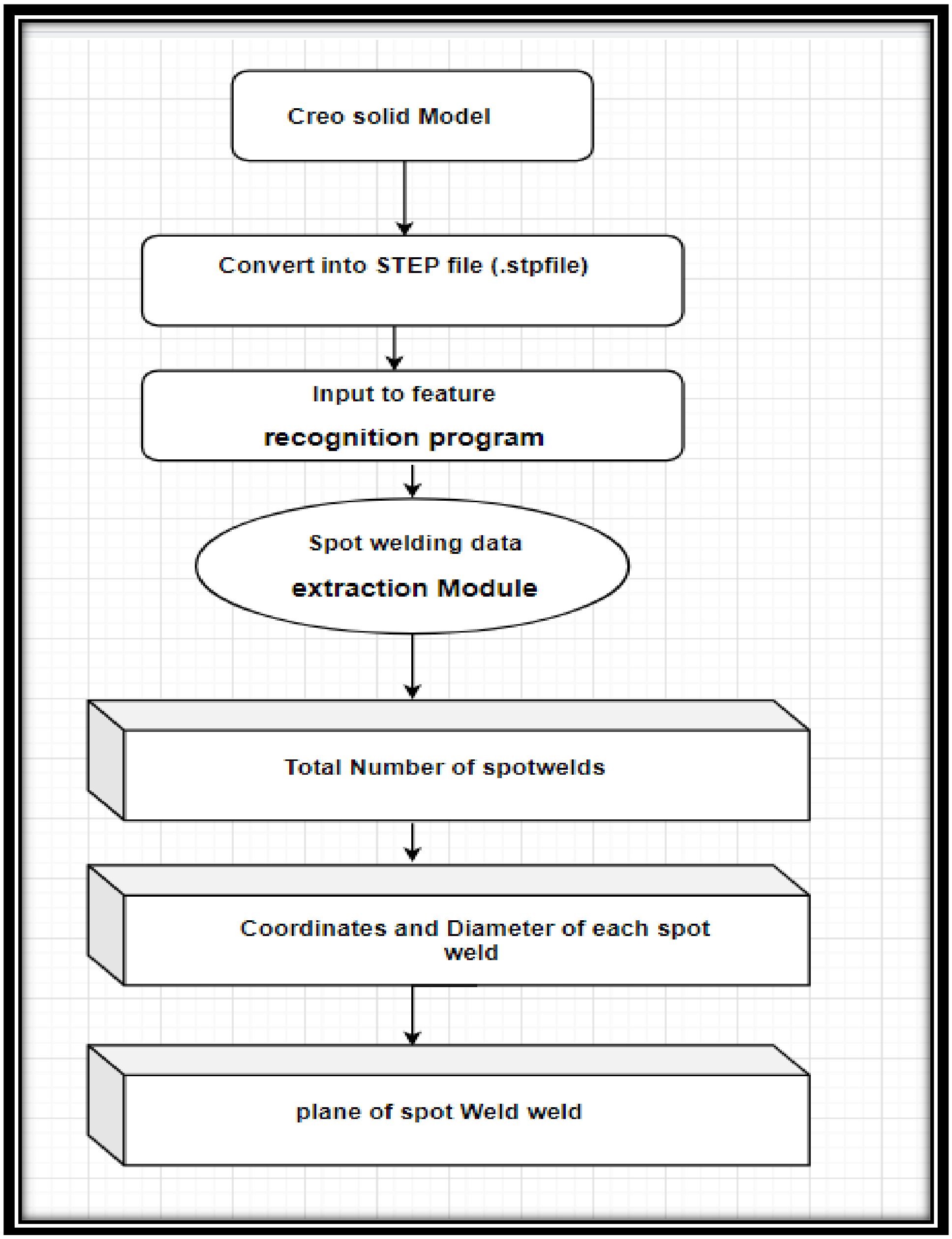

Fig. 1.1: The general characteristics identification of system structure

1.5 STRUCTURE OF THE THESIS

There are six chapters in the current thesis. The introductory section begins with a debate on the overall implementation of strong model feature recognition and design and production CAD / CAM integration. Requirements for computer with its accessories and also an outline of the scheme created

Chapter 2 current chapter is an analysis of the journal papers on the recognition of features. Critical observations are indicated on the basis of objective literature review and research scope.

Chapter 3 Presents the methodology used for welding feature Extraction and algorithm to extract welding data from STEP file explains the different Automatic feature recognition techniques detailed overview of STEP neutral file is also added in this chapter.

Chapter 4 Presents extraction procedure for special welding features like number of spot welds, diameter of spot weld, coordinates of weld as well as plane on which spot welding takes place.

Chapter 5 Presents implementation of source code and check them on different case studies than deals with results and discussion.

Chapter 6 Explains the end results/conclusions and scope of current work in future.

A graphical view is shown in Figure 1-1 to visually communicate the deposition of thesis.

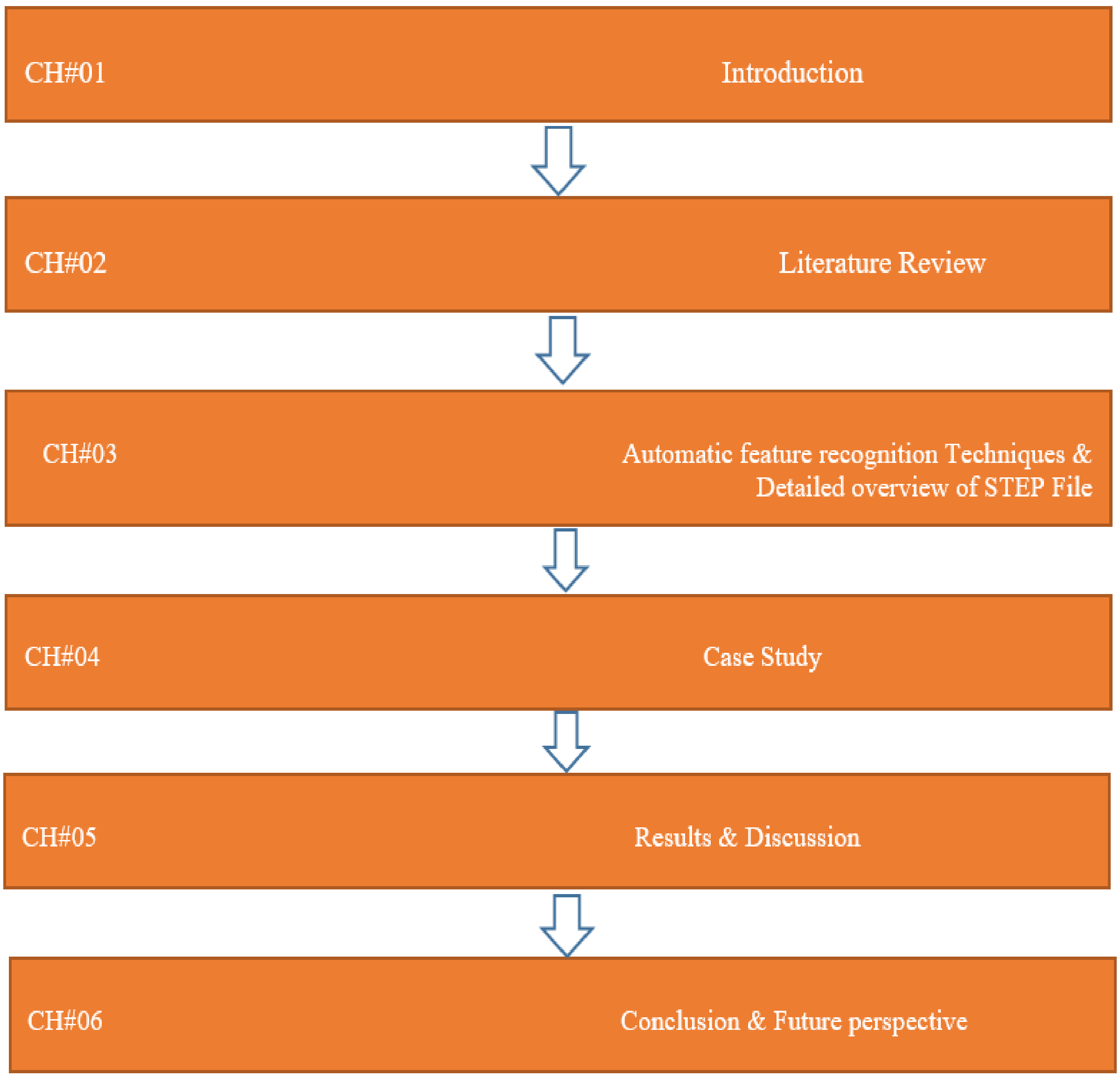

Figure 1-2 Deposition of Thesis Report

At the end of the dissertation, a list of references and appendices is added

Chapter - 2

LITERATURE REVIEW

In this chapter a base is formed in a sophisticated way that reader can easily understand the research related to feature recognition from STEP file. Before going into detail, reader must know some of the knowledge about basic terminologies used in this work to understand the research carried out in this field .In the beginning of this chapter, we explained the definition of feature that are used by many researchers and then explained the different Automatic feature recognition techniques which are hot topic for Data extraction .Brief introduction, STEP file is also explained from which we extract the data and then also explain the welding and shed some light on spot welding which author want to curb in this thesis. literature is reviewed on the basis of neutral files and feature recognition techniques and lastly introduction of some research carried out on different manufacturing features which are extracted from STEP file

An in depth review of present work is in this section and find research gap in the current research.

2.0 INTRODUCTION

There were almost two centuries have been spending on the analysis of extracting the significant characteristics from 3D model considering the research of Kyprianou [2] in 1988. Features play a essential role in the development of CAD / CAM. Among the explanations of features, Pratt and Wilson in 1985 provided a unique definition; 'Any part on the surface having significance importance is called feature'. Form characteristics can be classified directly into welding features, machined features, assembly features, fixturing features and analysis features, dependent upon their intended applications . .In present research we take welding features which are mostly related to joining of the work pieces. Informally we say that, **features are generic styles or other features relating to a part that often allows engineers to combine data that is helpful for reasoning about the part**. It is generally increasingly obvious that you need a part of a large automated design analysis implementation as a subsystem to recognize manufacturing characteristics from a strong type instantly. Recognition of features is a fresh discipline in which extract the features of the slid model and execution of techniques to detect CAM information such as welding features , diameters, holes, slots, etc.. Computer aided process planning (CAPP) is seen since a communication agent among computer aided design (CAD) and computer aided production (CAM) . Given a part's CAD data (a component of an item to be produced), CAPP's specific objective will be to produce a sequenced set of instructions used to produce the specific part. To do this, CAPP requires to interpret the portion as to its characteristicshe specific process-planning scheme now requires characteristics that are either specified directly in a specific design interface or generally acquired from a feature-recognition scheme.

A typical format for neutral document exchange is needed for data exchange between

CAD and CAM. Current CAM tools provide data user interface for IGES, STEP, and major native CAD types and generate NC rules. Communication protocols such as neutral files like IGES (Initial Graphics Exchange Specification), STEP (Standard for the Exchange of Product model data), and various other standards have been created to facilitate this inter-tool communication. The international standard, ISO 10303, informally called STEP has one of the considerable departures from existing standards is that a elegant information building technology was adopted to represent data in the standard, rather than using set file formats. STEP AP 203 is still the most widely supported standard in commercial products and can be used extensively. an unambiguous, neutral, computer-interpretable electronic digital representation to effectively connect among dissimilar CAD/CAM systems and produce timely, budget-friendly manufacture of quality products .

At present, the depiction of welded data as manufacturing data in a neutral format is not only in academia, but also by sectors, a positive research area. While important progress has been made on many study fronts, no fully automated welding feature recognition systems appear to exist at current.

2.1 FEATURE RECOGNITION

Feature representation can be performed either by a developer interactively or by a computer automatically. In internet feature recognition, a developer generates characteristics by choosing topological components of a geometric prototype that form a feature interactively. Recognizing interactive features has the advantage of being flexible enough, as the designer can set any combination of topological features as a feature. A very serious disadvantage, however, is that the manual selection of all features of the product model is a real problem in which errors can occur. In programmed feature recognition, the computer software automatically generates a feature model based on a geometric model and a description of most possible features. This thesis deals with the automatic recognition of features. Automatic feature recognition (AFR) is an integral part of knowledge-based engineering (KBM). It is used to examine a model, identify existing features, and extract features for further processing. Using the Principles (KBM) to integrate the processing brains into the CAM software, the software can automatically select welded processes. Automatic character recognition has been an active area of solid state model research for some time and is considered an integral part of CAD / CAM integration. The automatic use of feature recognition minimizes the need to manually inspect and manipulate the workpiece to organize it for welding. By automatically recognizing the CAD and CAM mixing function, the information contained in the solid model can be easily and reliably transferred from the customer's CAD system to the CAM system. Once a volume model is filled in the CAM system, the user can analyze the topology of the model to improve or automate the production process. Currently, the two most commonly used reliable modeling schemes are CSG (Solid Construction Geometry) and B-rep (Boundary Representation). Typically, the feature recognition module generally accepts a representation of the boundaries of the components and is therefore general enough to interact with most currently available CAD systems. The present work also assumes robust modeling schemes for the representation of boundaries.

2.2 STEP NEUTRAL FILE

Recent design and manufacturing developments have required the ability to present product information in a unique, neutral format that provides clear information that simplifies the various phases of the product lifecycle and that anyone can understand. CNC design and CAD manufacturing technologies. The data exchange consists of the transfer of information from the software to another medium via a medium representing the status of the data at a particular time. This snapshot of the information is numeric encoded, usually in ASCII or binary form. While this is not enough for an important exchange of data between computer systems, it is essential to display and send information. Interpretation of the data. Data logging provides a single source of rational information when multiple software systems can be accessed. The features that characterize the data exchange of a data exchange are the centrality of its data and the ownership of that data. In the replacement model, the application system stores the backup of the main data and exports a summary of the data to other users.. This usage has no explicit control over how changes to internal data are updated with the version of the paging file, and vice versa. Any other software system that imports the paging file has actually acquired ownership of the data. IGES (Initial Graphics Exchange System) provided a first practical solution for exchanging CAD data with a file format exchange. The IGES-neutral file format is today by far the most widely used method for exchanging files between heterogeneous CAD systems. However, replacing CAD models with IGES is very difficult using traditional methods. The neutral IGES files have some shortcomings:

IGES contained several ways to capture the same information, so that the correct interpretation depended largely on pre- and post-processing. IGES requires large files that require hours or even days to analyze. The average processing power available at this point loses information as information is transferred between two CAD systems with inherently different capabilities. Due to the lack of common tools, a common neutral file exchange format is required.

ISO 10303 is an international standard for computer readable presentation and product data exchange. The goal is to provide a neutral mechanism that can describe product data throughout a product's lifecycle, regardless of a particular system. For mechanical parts, the description of the product data has been standardized by ISO 10303. ISO 10303-21, informally called STEP, describes an ASCII representation, does not provide status information, and is only a snapshot. Due to the nature of this description, STEP is not only suitable for the neutral exchange of files, but also for the implementation of common product databases. This provides the opportunity to use standard data throughout the process chain of the manufacturing process. The integrated product team, made up of engineers from a variety of engineering disciplines, can develop the design in one part of the world, save that design as STEP data, and then send the ASCII file to the manufacturing site where the same data can be retrieved. Requirements are visualized. All subsequent activities (manufacturing, assembly, etc.) now become views of the same standard data relevant to that particular application. STEP also enables dynamic data exchange between different systems through the standard SDAI data access interface. STEP is a set of international standards based on an integrated Application Protocol (AP) architecture and integrated generic resources. The key difference with IGES is that STEP not only focuses on the type of data, but may also ask for the average of the data and the relationship between your data. The STEP opportunity includes all product data for each phase of the product life cycle in each area. STEP supports the complete and unambiguous exchange of product data between the systems of the program. STEP must support the exchange of product data between application systems. STEP needs to be more reliable and more efficient than other standards. The ACTION standards are divided into Application Protocols (AP). This is a unique set of agencies selected for a specific product, process, or industry. Exactly this common format as well as the data access components would like to provide STEP Neutral Record. Currently, a number of STEP application protocols are active, but here the work relates to the configuration designs of mechanical assemblies and 3D parts controlled by AP203.

The STEP Intermediate CAD file format in manufacturing machines for reading welding functions.

The survey is based on several methods for identifying features, such as: B.: Trick-based, graph-based, syntax-based, rule-based, volume-driven, and hybrid approaches. The survey also examines separately the work done on the Interact properties, working with a neutral STEP file, and working with other methods.

2.2.1 HINT BASED

The Generate and Test strategy is used during the recognition process to formulate hypotheses / indications of the presence of features in a part. Then, a validation procedure based on additional geometric and topological constraints is performed to confirm or ignore the generated displays.

During the modeling of the design features, it is very important to store the information of the features in an extended boundary representation model (B-rep). Geometric thinking or index-based methods recognize features in the search for a key, i. e. Something that is specific to an entity class in the schema of a part and then creates the largest feature that refers to that particular index, Hint-based methods have several good Conventional feature recognition methodologies often consider only the part's geometry and topology. Function classes can be personalized by users. Hint -based techniques can be employed to identify features from a variety of manufacturing domains. This particular approach is most ideal for components for which the feature instances them are easy, but there may be in huge numbers.

2.2.2 GRAPH-BASED

Numerous researchers have investigated graph-based methods. Since a B-rep(Boundary Representation) model can be seen as a graph framework, feature recognition could be considered as a graph-matching issue. Some well-known methods within Graph-based methods are utilized to recognize a function in a geometric design by using a graph-matching technique to match. In the graphics-based approach, an entity class is first modeled by a graphical structure that describes the particular topological and geometric constraints that are required to determine an entity. Then the structure of the particular graph can be encoded in a variety of calculation types.

The surfaces of an entity are represented by the nodes of a graph, and the approximation information between these surfaces is represented by the arcs connecting the nodes. You can also add additional information to the chart, such as: B. the orientation of the face. During the recognition process, the B-Rep model of a part is translated into a graph, which is then searched by the isomorphism of the subgraph to find assignments with predefined characteristic graphics

Graph-based algorithms have proven to be very useful in extracting certain types of features. These methods are divided into two categories: those for the search for graphics and those for the equivalent of the model. A new general difficulty for the graphics-based approach is that it is difficult to translate full-model graphics-based representations to the complex angles and topologies observed in practical operations. The elimination method isolates an attribute as a subgraph identifying the intersection nodes in the object graph. A new intersection node in the graphic object means that the function is assigned to only one side of the element.

The diagrams are used for different representations of entities. A new graphical grammar can be used to define features. The reputation of the features is based on the corresponding graphics. However, functions are not interactively defined and only identified to recognize them. The approach works face-to-face and is therefore not necessarily capable of dealing with complex features that involve interactions between confrontations. The characteristics of the graphics-based reputation are as follows: (1) It applies to many websites, not just machine. (2) allows the user to add new kinds of functions without changing the code; (3) It is ideal for modeling incremental elements. and (4) can effectively recognize the isolated properties. On the negative side, the interaction between features and multiple interpretations of features in graphical recognition can not be handled correctly.

2.2.3 SYNTACTIC-PATTERN

Design a model than write the design data into some descriptive language than parse this input language to some grammatical rules (each grammatical rule defines some specific model) If the syntax matches the grammar, the description can be classified in the corresponding model

Heart analysis and ECG measurements are a good example. The ECG curves can be approximated with diagonal and vertical detection segments. If normal and unhealthy waveforms can be described as elegant grammar, the measured ECG signal can be classified as healthy or unhealthy by first describing it in even segments and then trying to parse descriptions based on grammar.

The method is limited to simple polyhedral components.

2.2.4 RULE-BASED

A set of necessary and sufficient conditions for the patterns found in features is defined. **During the recognition process**, these rules are applied on data stored in the solid model of a part.

Within rule-based approach a arranged of heuristic rules can be used to describe the functional definition for a category of features. For example, by the observation of the slot, one can suggest the following set of heuristic rules to explain a slot feature. The slot is composed of three faces F1, F2, F3, Encounter F1, is adjacent to face F2, Face F2 is next to face F3, Face F1, and F3 are parallel, Face F1, forms a 90 level angle with F2, Encounter F3 forms a ninety degree angle with F2. Such heuristic rules for describing features are employed as tools for developing expert systems. Rule-based approaches therefore are also regarded as a specialist system approach.. The particular extracted features are then recognized utilizing a rule-based system.

2.2.5 VOLUME DECOMPOSITION

Volume decomposition technique is used to dissemble the difficult 3D model into sub parts these sub parts are used as a primitives to create the solid model. Different techniques are used to make the solid model simple by dividing into sub parts from which the whole part is formed .Many of the features are present in the zone of machining for the recognition. These types of methods were used to divide the delta sizes, the volume difference between the workpiece and the workpiece of the machined components into volumes of simple machining functions. Volume decomposition methods generally subdivide the geometric model into a series of intermediate sizes, which are then modified to produce properties.

The particular method of cell-based decomposition breaks a delta volume into minimal cells by enlarging and intersecting all average surfaces or surfaces of the delta amount. Then the cells are combined to create editing functions. This method is prone to combinatorial difficulties and it is difficult to control non-polyhedral designs.

2.2.6 HYBRID

In an automatic feature recognition technique hybrid approach is used. In which many hybrid techniques have been created that combine aspects of the fundamental techniques. Gao and Shah have already reported a hybrid approach that combines facets of graph-based and volumetric approach.

2.2.7 Neural network-based approach

In neural network approach user defined his own features and set some features examples. The values are extracted from each example of features and stored in a vector that is input to the neural network. The neural network can divide the features into four different classes (square, rectangle, parallelogram, and slits). During the recognition phase, some values of each property are extracted. and stored in a vector that has entered the neural network. During the recognition phase, the extracted features are encoded in vectors that have entered the network to be classified into one of four features

2.3 STEP BASED FEATURE RECOGNITION

STEP uses Boundary representation as a basis for defining solid model, hence review here is based on the methods which take information from B-Rep as input and use of geometric and topological boundary relationships. The B-Rep tree is checked and the tree framework is compared with the structure needed for a function identifies the features, but the extracted features do not make it easy for Agile Manufacturing to exchange data as warranted by its needs. Also, the characteristics are not stored. Then feature like form features were recognized using neutral format of the model generated in CAD like IGES as input, which also use B-Rep as one of the basis of solid representation.

The IGES format could not be recognized as international standard, the new area is meant to use STEP file as input (based on B-Rep) for feature recognition.

In general, almost all feature recognition techniques are not sufficient to detect all interactive features because a new interactive feature can still be proposed so that an existing feature recognition technique detects its lack of robustness. ,

In the field of feature recognition, many articles have been published that address this complex issue. This work should not give a global overview of all possible detection techniques, developed systems and their problems

2.4 Resistance Spot Welding Process

Large scale and high rate of production of components using sheet metal has been possible only due to the resistance welding processes which have the dual advantage of making joints quickly and with minimum distortion. You will find at least eight important resistance welding processes including spot welding, seam welded, projection welding, flash welded, percussion welding, upset bottom welding, high frequency level of resistance welding and high rate of recurrence induction welding. However, in phrases of business practice mainly the very first three procedures have found wide software.

Resistance welding processes vary from other welding procedures in that no fluxes are used, metallic fillers are rarely used, and sore joints are often lap form. The specific quantity of heat produced in the job product will rely on the magnitude of the present, the strength of the current conducting route, and the moment for which the current will flow. In sentences of joule heating, this is stated.

According to Ohm’s legislation, the voltage (V) required for a current circulation (I) is given by the connection V=IR, where L is the resistance proposed by the work piece to the flow of current. The particular heat created is, thus, given by the method,

$$
\begin{aligned}
H &= IVt \\
&= I\,(IR)\,t \\
&= I^2Rt
\end{aligned}
$$

Where,

H = Heat generated, joules

I = current, amperes

R = resistance, ohms

t = time of current flow, seconds

At the interface of the two surfaces forming the lap joint is the point of greatest resistance, it is also the point of greatest heat. A low voltage high ampere current flows from one adjoining place to the other until the metal at the interface is heated to a high enough temperature to trigger localized fusion that pinches the molten from both components under the applied pressure to a homogeneous mass called the weld nugget as shown in Fig.2.1.

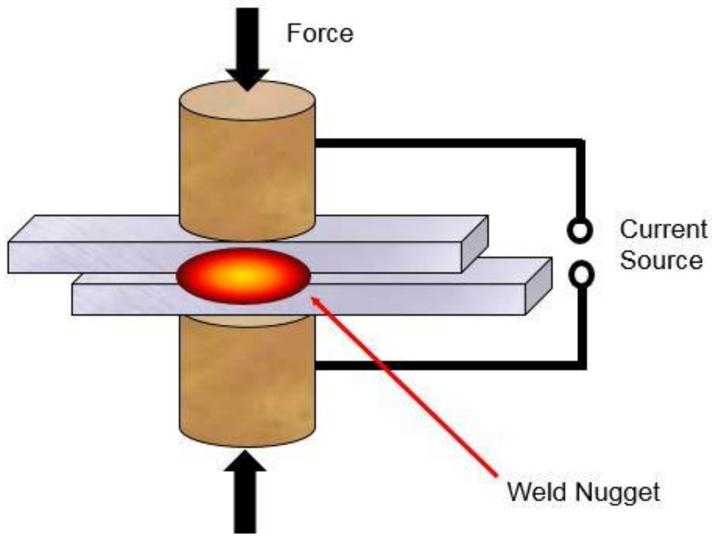

Figure.2.1 Resistance Spot Welding Process Details

2.5 BRIEF REVIEW ON THE GEOMTERIC FEATURE RECOGNITION

Numerous scholars have the idea of obtaining geometric features of various models and CAD standard formats such as the format specification for the exchange of output images (IGES), the format for information exchange (DXF) and STEP (standard format studies product exchange). The special extraction of geometrical properties in a data source design has been focused in the 1970s in which considerable attention by the scholor Woo [1] who developed a technique for extraction of geometrical feature of a constructive solid geometry model having Dimension of workpieces are 2.5D. They also mention a technique of "alternating sum of volume" in 1982. Pande and Prabhus [2] presented the way and performance of feature recognition from DXF files and allotment of tools for rotating component automata. Kruth and. to [3] he presented information about properties of the output definable by the user of the CAD systems and CAD systems dependent properties, a method for defining and retrieving. Researcher proposed the method to CAD and one Linear Graphics document formats are used to extract the information on geometric and technical features. Zuraini and Habibollah [4] featured work in the near future to categorize a Formula Features DXF file and map the algorithm's properties to technical parameters. The code identifies geometric features that can be used to identify the significant information of three dimensional object. The features generated from the program code sorting program provide the processing parameter Information via the one to one algorithm correspondence.

The unique mapping algorithm relies on the machinery dimension to the operations perfomed on milling, namely the dimensional counter, Half of the angle plane diameter. The editing functions are optimized with the scalable method. Kumar and Saha [5] dveleoped a technique to read the neutral STEP file of CAD model and find the important data in it. These individuals have made the "STEP READER" software to automatically screening of the STEP neutral file according to the programmed sent for the Data extraction. Srikanth and ing. [6] have proposed various softwares for extracting style data stored as documents in the information exchange format (DXF) as well as for properly storing data. Mohamad [7] proposed an algorithm which extract the design information from the CAD model. The CAD data (Computer Aided Design) is automatic feature recognition from Design model and contains the manufacturer's product data information. The Design data contains geometric and topological (non-geometric) data. Data related to geometry includes lines, vertices and curves. While non-geometric data includes text, colors, and layers. The extracted design data is must produce which is helpful for making the object according to the CAD Model.

2.6 REVIEW OF RESEARCH ON FEATURE RECOGNITION

For more than two centuries, the automatic identification of production characteristics and the transformation of function designs were active study fields. Various methods have been created for automatic character recognition (AFR), such as heuristics, syntactic models, rules, indices, graphs, and neural networks. There are so many techniques, the one which got attention is the technique based on the STEP file, this technique attracted the devotion of the author and also depends on the author's capacity to acknowledge various features based techniques depending on its dimensions.

Function-Based Processing System with STEP By Fredrick Proctor and Thomas R. Kramer [3] In this document, the authors have proposed a prototype processing system in which the product and process data replace the NC programs at the same time execution time

This porotype developed on the basis of manufacturing features and creats a data formats which is consistent with the STEP where the STEP is not exist

This protype built in the express language and all are in the 21 file format, same language which is used by the STEP and this prototype is made by NIST (national institute of standard and technology)

[4]Lau, H.C.W in 2004 used the STEP design file as a information source for recognition of features limited to only prismatic parts. [5]*Bojan babic and nenad nesic* To generate an automated Capp, we used an AFR technique the purpose of this is to review the major AFR techniques for different problems one is extraction of geometric primitives and second is the feature patterns .Three different types of geometry model are used one is boundary representation geometry in which features comprises of vertices, edges, faces

Second is the constructive solid geometry which is used sphere and cylinder

Whereas wire frame geometry we used vertices and edges

In this case the application protocol which is used for AP224 which contains the complete information regarding to manufacture the part

1- The first approach which is the syntactic pattern based approach is used for 2-D which is the short coming of this approach

2- Graph based approach we need extensive preprocessing is required to construct each part the main problem of graph based technique is that it will not solved the interacting features.Rule base technique is used in this case.The major short comings which we faced in al the AFR technique is that we can access only limited feature and these features are not enough to produce a proper production plan.[6] T. R. Kannan & M. S. Shunmugam in his paper explains the feature recognition system which act as a input for feature reasoning system.The previous suveys reveals that most of the researchers have been proposed the cuts and bent features from the step file as well as features specific to the dimensions means when dimension is changed the feature will be changed and mostly these types of features are expressed not the features which are related to the shapes features are also changed when there is changed in shape occurs

This paper will focus on cuts ,bent as well as the drawing and stretching(it means when you stretch a sheet the thickness of the sheet will changed) features , extracted the drawing(means that the shape wiil be changed but the thickness remains the same)[in actual the thickness will take as uniform throughout the process] and stretching features from the STEP file by using the rule based approach

First of all draw a 3 D model in the CAD software any, than translate the 3d product data into the STEP AP 203 format than recognize the features of the relevant DATA and generate the central 3D plane. this central 3D plane is considered as the sheet plate with zero thickness. many faces are attached to this plane which is separated as inner and outer faces the remaining faces made the main structure which is further processed to recognized the manufacturing features

Data which is classified on the basis of faces depends on four different types

1- Planar surface (planar is just a fency word for plane e.g all six sides of the cube are planar surface it is also 2-d surface)

2- Spherical surface (perfectly round three dimensional object)

3- Cylindrical surface(three dimensional shape whose boths ends are round shapes)

4- Toroidal surface (surface looks like doughnut)

The manufacturing features which is recognized can be further classified into four types cuts, stretched, bent and drawing . the major difference between the stretching and drawing is that thickness of sheet will have changed whereas in drawing the thickness of the sheet will not changed in bents pure bending will be occur whereas in cutting pure shear will be occur. In actual we take a sheet of uniform thickness. {In drawing tensile force is applied to convert the metal into desired shape by keeps them on thinner}.[7]

Suziyanti Marjudi & Mohd Fahmi Mohamad Amran compare the IGES and STEP data files. Reached on a conclusion in his paper that IGES and STEP are standards and popular in product data exchange between Heterogeneous CAD systems.[8] c.w keong and yusri yousaf in his paper extracts the manufacturing information. The main aim of this paper is to recognize a system and data bases which extracts a manufacturing information from the from CAD in this process we need a concurrent engineering across all the manufacturing activities take ap 203 as input than applies the manufacturing features and data extracting and convert them into ap 224 which is used for manufacturing process in this paper generate link between AP 203 format and Ap 224 format for concurrent design because manufacturing process needs app 224 format to manufacture the part . All automatic feature recognition consist of two basic features one is to define the feature to be extracted and the second one is to used the type feature extraction technique.[9] S. Sivakumar and V. Dhanalakshmi in his paper is extracted the information for cylindrical features components to developed the relationship between CAD/CAM and to generate NC codes. Many researcher have been proposed the methodologies and algorithms to extract the DATA . some of them extract the DATA directly from the CAD model some of them extract data from the neutral file to generate the NC codes. In this paper they extract the manufacturing information from the STEP file for cylindrical parts and generate the NC codes from the extracting features. They neither used the CAD model nor the neutral file to generate the NC codes they used the extracting DATA for generating the NC codes which will be directly executed by the CNC machine. Actually the proposed program is worked as an interface between the CAD and CAM for machining. after generating the NC codes, then machine a part on CNC lath than validate the result by inspecting them by CMM with the data extracted from the step file which is a reference data file in text format. The part which is to modelled can be six different types of features right hand tapered, left hand tapered, right hand convex cylinder, straight cylinder right hand concave cylinder and hemisphere. Dimensional features are also taken from the reference extracted features which is extracted from the STEP file which is used by all the downstream activities after extracting all the features and stored in the from of text file after gerating NC codes than start turning to make a part { a part is manufactured of 25 mm thickness material aluminium xultran lath machine is used tool which is used is made up of carbide tipped tool FUNAC control system is used}.[10] V. N. Malleswari, Dr. P. M.Valli and Dr. M. M. M. Sarcar The identification of features is one of the most effective approaches to solving the CAD and CAPP interface issue. One of the most challenging functions in simultaneous design and manufacturing is to create a connection between CAD and CAPP. It is difficult to create a process plan effectively without a correct interface between CAD and CAPP. Recognition of features is essential to achieving this goal. Automatic recognition between can Reinforce the connection in them for cad and capp when manual processes are used to translate the CAD geometry language into a greater CAPP language the main aim of this paper is to make an interface between CAD and Capp which reduces the error because CAPP do not understand the 3D geometry .feature recognition can be both plat form dependent and plat form independent .In plat form dependent features can be extracted from the the design directly where as plat form independent features can be extracted from the neutral file .we used platform independent case . many systems have been developed to extract the straight features of holes and blind slots means when straight hole as are present than w extract the features but no one extract the features when the interaction of features are exist just as cross holes in this paper we extract the features of the cross holes and inclined holes in the rotational parts .The program is coded in java language and input for the system is STEP file the software recognizes the various strings and interpret them in the form of manufacturing and recognizing the features for the cross holes and cylindrical holes.[11] "oussama jaider , abdelilah elmesbahi and rechia ahmed in his paper AUTOMATIC FEATURE RECOGNITION FOR ROTATIONAL PARTS explain that CAPP plays a very important role between the CAD and CA Programs are used to make an efficient process plan but for this process we need a to make an efficient process plan we need automatic feature recognizers most of the automatic feature recognisers can extract data for the isolated features if the features are interacting ng with each other it is very difficult to extract data from it.Many researches have been done on finding the isolated features in this paper We find the isolated and interacting features for rotational parts to make CAPP plane efficiently when features are interacting with each other more number of process plane are generated these process planes are reduced by taking into account the manufacturing rules as well metrial removals principles.[12] Bitla Venua, Venkateswara Rao Kommab, Deepanshu Srivastavac in his paper develop a methodology to recognize manufacturing features of prismatic parts from STEP AP 203 file (i.e. ISO 10303 file), which will recognize all common types of manufacturing features of prismatic parts. In this work a new concept called Object Slicing Approach (OSA) in which an imaginary plane with a specific feature definition will be moved along different axes directions to identify all the manufacturing features of the component. [13] Leila Zehtabann, Omar Elazhary, Dieter Roller , The main objective of this paper is to present a framework to support similarity and retrieval system that would assist to search for and retrieve similar models from a given CAD model. [14]Bitla Venua, Venkateswara Rao Kommab, Deepanshu Srivastavac in his work, STEP-based feature recognition system is developed to recognize B-spline surface features and its sub-types from the 3D CAD model represented in AP203 neutral file format .

Hsu-Pin Wang and Ching-Ann Lin [15] creates two software's based on 2 Dimensional CAD models. The initial software detects the properties of the surface of the rotated sub-assemblies and in the next portion a software is developed to generate the program of the NC part. Srinivasa Kumar [16] Were using the data sharing format of IGES for automated feature recognition and partial data extraction from the computer aided design. Wang and Wong [17] developed a system to recognize the feature of with an o-o(0bject-oriented) method and used the necessary and sufficient condition for confirm or discard the information in the data base to make an efficient process plan , Sabourin and Villeneuve [18] proposed an efficient process plan by using professional system that uses the CATIA STEP file to recognize the features . Jung and Lee [19] proposed an automated data extraction algorithm that used the IGES format as the input file for the algorithm to extract the symmetry information. Fuh [20] used a solid model and a technical design to capture geometric and non-geometric definitions for feature recognition scheduled a plan which is working that can be developed to meet manufacturing and fastening necessities. All this work requires suitable direction tool so that tool is known in advance. In addition, they can only manage the existing category of the features.

Yang and Lee [21] developed a framework for modifying features based on decomposition and geometry-oriented configuration to create alternative process plans. The method found by the authors is effective in the wave and profile features that require tool changes. The functions are broken down and subdivided into editing functions that can be edited with fewer tools. Gao [22] introduced the system for detecting features of hybrid processing by leveraging the Positive capacity of different strategies to enhance effectiveness and to detect features of computing system capacity.

Jiang et al. [23] presented a framework for extracting and identifying features based on group technology, using an expert system with database to generate the optimal process plan. Bhandarkar and Nagi [24] developed a system for extracting feature-oriented standard forms, which converts the level of the lower geometric and topological data specified in a STEP file into corresponding manufacturing data. The smooth edges for recognizing the entities they contain are not classified by them. The taxonomy of features at a proposed level does not provide information related to the fabrication. The intersection of entities and their relationships are not investigated.

Saravanan et al. [25] introduced Genetic Algorithm (GA) and Simulated Annealing(SA) strategies applied to solve the continuous machining profile problem. Simulated Annealing performs barely higher than Genetic Algorithm. Ong et al. [26] developed a manufacturing feature consciousness system from a design via function model. Their feature recognition processor converts the format points in a CSG tree into manufacturing features by way of traversing from its root to the best leaf. They carried out a unique characteristic relationship analysis, which is in the main based on geometric and placement traits of the features. They additionally developed a manufacturability evaluation module.

Developed an autonomous platform based on STEP for recognition of design and production features. The system can comprehend characteristics from the reduced point of geometry and topology available in the ACTION file as well as design and production. Table 2.1 presents the multiple methods used by scientists to extract information from STEP relatively neutral file Ap 203

Name	***year***	***Features***
Frederick M. Proctor and Thomas R. Kramer	**1998**	Design a software by using the EXPRESS Language and part-21 file format,Generate a STEP format where STEP does not
Lau, H.C.W	**2004**	Use the STEP design file as a source of data to recognize characteristics that are restricted to prism components only
Bojan babic & Zoran miljkovic	**2008**	Review of various approaches for solving major **AFR** problem(i)Extraction of geometric primitives (ii) feature pattern
Kannan, T.R and Shunmugam	**2009**	Used **3D** model information in **STEP AP-203** type as a known input with different characteristics Constrained for sheet metal parts
Suziyanti Marjudi & Mohd Fahmi Mohamad Amran	**2010**	**IGES** and **STEP** are standards and popular in product data exchange between **Heterogeneous CAD systems**

C W Keong & Yusri Yusof	**2012**	In this paper they extract the Geometrical features from Drawings and identify manufacturing features from STEP Input is **AP-203** whereas as **Ap-224** is output file
Sivakumar & Dhanalakshmi	**2012**	extracting manufacturing features from STEP AP203 file for cylindrical parts
V. N MALLESWARI	**2013**	recognizing the **cylindrical** and **cross hole** features and inclined holes in a rotational parts
OUSSAMA JAIDER	**2014**	recognize features for **rotational parts**, and on the other hand, to generate optimal combinations of **interacting features** based on **features elimination**
Bitla Venua, Venkateswara Rao Kommab, Deepanshu Srivastavac	**2015**	Develop a methodology to recognize all common types of manufacturing features of prismatic parts from STEP AP 203 file by Using a new method Object Slicing Approach (OSA)
Leila Zehtabann, Omar Elazhary, Dieter Roller	**2016**	The main objective of this paper is to present a framework to support similarity and retrieval system that would assist to search for and retrieve similar models from a given CAD model
Bitla Venua, Venkateswara Rao Kommab, Deepanshu Srivastavac	**2018**	In this work, a STEP-based feature recognition system is developed to recognize B-spline surface features and its sub-types from the 3D CAD model represented in AP203 neutral file format

Table 2.1 Various manufacturing features recognition

2.7 CRITICAL OBSERVATIONS

The following are the literature observations:

1-Many scientists have obtained geometric information from conventional formats such as Codes IGES, DXF, STEP, etc. There is no documentation on how to do it extract welding information, especially spot welding, from STEP file.

2-Literature on manufacturing features of a part is available, whereas literature to recognize welding feature specially spot welding is not available

3- Most of the researchers are considered the machining features. There is no literature available for the recognition of spot welding.

4-No author has considered the spot welding feature recognition.

2.8 OBJECTIVES AND SCOPE OF THE STUDY

Based on the evaluation of current literature and critical observations as stated in chapter 2.3, the goal of the current study job is recognized. The primary goal of this study is to create software to extract data from the STEP file to determine the characteristics of spot welding. It is an effort to standardize the recognition of features from "STEP to welding Feature" (STWF), which in turn decreases the time spent on feature recognition by enhancing productivity and minimizing production costs. .

2.8.1 OBJECTIVE OF THE PRESENT WORK

1-To study the different methods used for automatic feature recognition of machining features and develop a new method of feature recognition for spot welding from boundary representations of solid models.

2- To create algorithms to extract data from the STEP file to evaluate spot welding characteristics.

3-Develop a software which extracts three main features of Spot welding that are highlighted as

(i) Total Number of spot welds

(ii) Coordinates and Diameter of each spot weld

(iii) plane of weld

4- To develop a device to facilitate the assimilation of computer-aided format and computer-aided manufacturing for Computer aided technique planning(CAPP) for automatic spot welding.

5- To obtain welding data of recognized feature for high-level process planning.

2.8.2 SCOPE OF THE PRESENT WORK

Towards this goal, the scope of the research work is outlined as follows:

1.STEP Application Protocol 203 (AP 203), neutral file is selected for data exchange which supports most commonly used solid representation i.e. boundary representation(B-rep).

2.The modeler selected is commercially available parametric feature based CAD system, the Creo 3.0. It works on principles of Boundary Representation (B-Rep). This software is used for CAD environment to develop solid model.

3.The product CAD data that is used is obtained from a neutral STEP file format generated using the PRO/STEP interface module.

4-STEP file format has abundant raw data, hence software named "**WELDING FEATURE READER**" is developed to scan this raw data and extract the data as per B-Rep structure and as required for feature recognition.

5. Develop a program to extract all the salient spot welding part coordinates from the STEP file and store them in arrays and display them in the output file.

6.Development of a program for the recognition to recognioze the welding data characteristics from extracted information.

Chapter - 3

Welding Data Extraction from STEP Files (WDE)

CHAPTER - III

WELDING DATA EXTRACTION FROM STEP FILES (WDE)

There are many two exclusive impartial archives reachable in CAD software. A few of the neutral archives are IGES, STEP, DXF, STL archives etc. STEP and IGES are most popular. STAGE is meant for product facts exchange, whereas IGES is perfect for geometry information trade. STEP is worldwide recounted impartial file layout of almost all industrial CAD software.

ISO 10303 is an ISO standard for the pc interpretable portrayal and exchange of product manufacturing information. Its diagnosed title is: Automation systems and integration - Item information illustration and trade. It in reality is known informally as "STEP", which stands for "Standard for the Trade of Product model data". The description of product data for mechanical components has been standardized with the aid of ISO10303 and special protocols are accessible in STEP. A few of the Application Protocols are AP203, AP214, AP224 and many others used for exclusive applications. AP203 is Configuration dealt with THREE DIMENSIONAL sorts of mechanical parts and assemblies. ISO-10303-21 is the starting keyword of STAGE file and END-ISO-10303-21 is the ending key-word of the STEP file. This is based on the B-Rep(Boundary Representation).

3.1 HIERARCHY OF WELDING STRINGS

Initially, the plan is drawn in Creo 3 0 and converted into a STEP file. The extension of the STEP file is. stp. This file is given as enter to the developed program. The developed characteristic consciousness application begins searching the STEP file with a string Open_SHELL and it ends at a string CARTESIAN_POINT. In between more than a few strings such as ADVANCED_FACE, FACE_OUTER_BOUND, CARTESIAN POINT, PLANE, and so forth . are searched in a hierarchal manner. The hierarchies of Welding string is shown in Figure 3.1

.In the given case study the number of spots welds are three so the three open _shells are found each open shell have different entities attached to it. Another method to find the spot weld by searching the line 82 with hash tag #85 ,tells us i.e,

#85=SHELL_BASED_SURFACE_MODEL('',(#28,#56,#84));

The number of pointer’s present in the manifold surface model tells us the number of spots weld in the geometry. The SPOT Welding string is OPEN_SHELL and entity numbers are (#28, #56, #84)

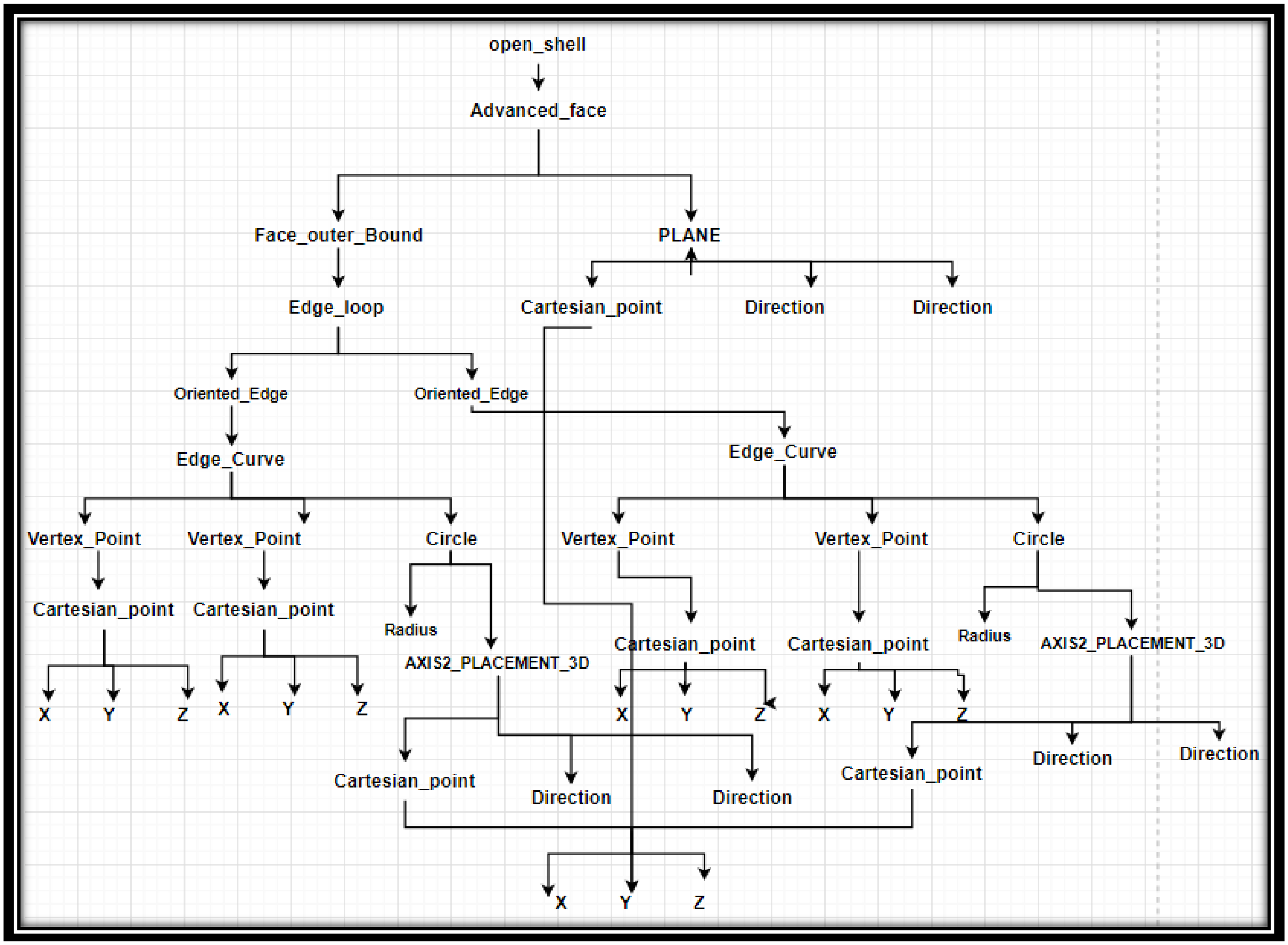

Fig. 3.1: Hierarchy of Spot Welding strings

3.2 SPOT WELDING DATA EXTRACTION FROM STEP FILES

The input for function consciousness is extracted Spot welding data from STEP files. STEP archives are saved in a textual content primarily based ASCII format. Every line starts offevolved with # (hash) and observed by way of a positive number, this is called entity. Every line ends with a semicolon (;). Extraction of facts begins with parsing the STEP file

For the reason of cognizance of feature, assessment of part curve construction, circle centers, radius of the circles, a variety of surfaces and floor radius are required. For the clarification of above aspects, awareness of a cylindrical feature is described in element in the following sections.

Totol number of Spot weld

For every STEP file the algorithm parse the step file and tells us how many spots welds are there in the given file .the number of pointers present in the SHELL_BASED_SURFACE_MODEL('',(#28,#56,#84)); will tells us the number of spots welds . in this three pointers are present so the totol number of spots welds are three. Another Alternate method may be used to recognize to detect the number of spots welds which is by calculating the number of OPEN_SHELL s present in the file .

We will take the first logic that is the number of pointers present in the SHELL_BASED_SURFACE_MODEL is equal to the number of the spots welds.

3.2.1. Edge Curve Construction

For the edge construction for the spot three OPEN_SHELLs are found (#28,#56,#84) .Each Advanced face consist of only one ADVANCED_FACE .Just like #28 (OPEN_SHELL) consist of advanced face having hash tag (#27) similarly #56 (OPEN_SHELL) consist of advanced face having hash tag (#55) same for the third spot weld #84 (OPEN_SHELL) consist of advanced face having hash tag (#83)

Each ADVANCED_FACE is certain by quantity of edges which is termed as FACE_OUTER_BOUND). Each FACE_OUTER_BOUND is formed with edges as a loop which is termed as EDGE_LOOP ((#25,#53,#81). Further each EDGE_LOOP is formed by ORIENTED_EDGEs (#21,#23,#49,#51,#77,#79) termed as EDGE_CURVE(#14,#15, #42,#43,#70,#71). Each EDGE_CURVE is shaped with exclusive geometric shapes like CIRCLEs (#5,#10,#33,#38,#61,#66) etc.

3.3 First spot weld

In this paragraph I will take the features of first spot weld like coordinates of weld, diameter of weld and the plane of weld

3.3.1 Finding the coordinates of the First spot weld

In continuation from EDGE_CURVE construction the last hashtag of each edge curve will gives us the entity number of circle whereas first two will gives us a number of vertex point which ends with a Cartesian point. Circle is a type of curve, which is formed with two edge curves.

AXIS2 PLACEMENT 3D is used for each circle command. In turn, AXIS2 PLACEMENT 3D is a CARTESIAN POINT circle center. The last three CARTESIAN POINT hashless numbers show spot weld coordinates.

```
#2=CARTESIAN_POINT('',(-3.482E1,4.721586427335E1,7.85E1));
#3=DIRECTION('',(-8.421193301994E-1,-5.392912327346E-1,0.E0));
#4=DIRECTION('',(-5.392912327346E-1,8.421193301994E-1,0.E0));
#5=AXIS2_PLACEMENT_3D('',#2,#3,#4);
#7=CARTESIAN_POINT('',(-3.482E1,4.721586427335E1,7.85E1));
#8=DIRECTION('',(-8.421193301994E-1,-5.392912327346E-1,0.E0));
#9=DIRECTION('',(5.392912327346E-1,-8.421193301994E-1,0.E0));
#10=AXIS2_PLACEMENT_3D('',#7,#8,#9);
#12=CARTESIAN_POINT('',(-2.807885959082E1,3.668937264586E1,7.85E1));
#13=CARTESIAN_POINT('',(-4.156114040918E1,5.774235590084E1,7.85E1));
#14=VERTEX_POINT('',#12);
#15=VERTEX_POINT('',#13);
#16=CARTESIAN_POINT('',(-3.482E1,4.721586427335E1,7.85E1));
#17=DIRECTION('',(8.421193301994E-1,5.392912327346E-1,0.E0));
#18=DIRECTION('',(5.392912327346E-1,-8.421193301994E-1,0.E0));
#19=AXIS2_PLACEMENT_3D('',#16,#17,#18);
#20=PLANE('',#19);
#22=ORIENTED_EDGE('',*,*,#21,.T.);
#24=ORIENTED_EDGE('',*,*,#23,.T.);
#25=EDGE_LOOP('',(#22,#24));
#26=FACE_OUTER_BOUND('',#25,.F.);
#27=ADVANCED_FACE('',(#26),#20,.T.);
#28=OPEN_SHELL('',(#27));
```

Figure 3.2 Coordinates of first spot weld in step file

3.3.2 Finding Diameter of the First spot weld

In continuation from EDGE_CURVE construction the last hash tag number will tells us the pointer of the circle whereas the CIRCLE's last number without hash (#) indicates the circle radius i.e 1.25E1 E means the exponent and one shows the power of exponent i.e. 10 power one.Radius of the first circle is 12.5 from where we calculate the diameter of the spot weld i.e 25

```
#6=CIRCLE('',#5,1.25E1);
#11=CIRCLE('',#10,1.25E1);
#21=EDGE_CURVE('',#15,#14,#6,.T.);
#23=EDGE_CURVE('',#14,#15,#11,.T.);
```

Figure 3.3 Diameter of first spot weld in step file

3.3.3 Finding PLANE of the First spot weld

Pointers in the OPEN_SHELL tells us the number of advanced faces; means the number of surfaces it consists of. Different kind of surfaces are there e.g. CYLINDRICAL_SURFACE, CONICAL_SURFACE, TOROIDAL_SURFACE, SPHERICAL_SURFACE etc. in our case the surface is cylindrical.

The second entity number with hash tag in the advanced face tells us the information of PLANE (#20). In which AXIS2_PLACEMENT_3D(#19) is present which contains three pointers(#16,#17,#18) the first pointer will tells us the Cartesian point , shows the location through which planes passes and the other two tells us the directions , at which directions the plane passes.

```
  #16=CARTESIAN_POINT('',(-3.482E1,4.721586427335E1,7.85E1));
#17=DIRECTION('',(8.421193301994E-1,5.392912327346E-1,0.E0));
#18=DIRECTION('',(5.392912327346E-1,-8.421193301994E-1,0.E0));
#19=AXIS2_PLACEMENT_3D('',#16,#17,#18);
#20=PLANE('',#19);
#27=ADVANCED_FACE('',(#26),#20,.T.);
#28=OPEN_SHELL('',(#27));
```

Figure 3.4 partial step file for plane of first spot weld

3.4 Second spot weld

We will take Different features of the second spot weld like coordinates of weld, diameter of weld and the plane of weld

3.4.1 Finding the coordinates of the Second spot weld

In continuation from EDGE_CURVE construction the last hashtag of each edge curve will gives us the entity number of circle whereas first two will gives us a number of vertex point which ends with a Cartesian point. circle contains first hash tag which is AXIS2_PLACEMENT_3D having three pointers (#30,#31,#32) .The first pointers will tells us the Cartesian point which indicates the coordinates of a circle, which is spot weld .Circle is a type of curve, which is formed with two edge curves in creo.

```
  #30=CARTESIAN_POINT('',(-7.088E1,1.079158279640E2,7.263E1));
#31=DIRECTION('',(-8.470233271991E-1,-5.315557197327E-1,0.E0));
#32=DIRECTION('',(-5.315557197327E-1,8.470233271991E-1,0.E0));
#33=AXIS2_PLACEMENT_3D('',#30,#31,#32);
#35=CARTESIAN_POINT('',(-7.088E1,1.079158279640E2,7.263E1));
#36=DIRECTION('',(-8.470233271991E-1,-5.315557197327E-1,0.E0));
#37=DIRECTION('',(5.315557197327E-1,-8.470233271991E-1,0.E0));
#38=AXIS2_PLACEMENT_3D('',#35,#36,#37);
#40=CARTESIAN_POINT('',(-6.157777490468E1,9.309291973806E1,7.263E1));
#41=CARTESIAN_POINT('',(-8.018222509532E1,1.227387361900E2,7.263E1));
#42=VERTEX_POINT('',#40);
#43=VERTEX_POINT('',#41);
#44=CARTESIAN_POINT('',(-7.088E1,1.079158279640E2,7.263E1));
#45=DIRECTION('',(8.470233271991E-1,5.315557197327E-1,0.E0));
#46=DIRECTION('',(5.315557197327E-1,-8.470233271991E-1,0.E0));
#47=AXIS2_PLACEMENT_3D('',#44,#45,#46);
#48=PLANE('',#47);
#50=ORIENTED_EDGE('',*,*,#49,.T.);
#52=ORIENTED_EDGE('',*,*,#51,.T.);
#53=EDGE_LOOP('',(#50,#52));
#54=FACE_OUTER_BOUND('',#53,.F.);
#55=ADVANCED_FACE('',(#54),#48,.T.);
#56=OPEN_SHELL('',(#55));
```

Figure 3.5 Coordinates of second spot weld in step file

3.4.2 Finding Diameter of the Second spot weld

Three pointers are presents in the EDGE_CURVE construction the last number with the hash tag tells us the pointer of the circle whereas The last number in the CIRCLE without a hash tag (#) indicates the circle radius, i.e. 17.5E1 E means the exponent and one shows the power of exponent i.e. 10 power one. Radius of the first circle is 15mm, from where we calculate the diameter of the spot weld i.e. 35mm.

```
#34=CIRCLE('',#33,1.75E1);
#39=CIRCLE('',#38,1.75E1);
#49=EDGE_CURVE('',#43,#42,#34,.T.);
#51=EDGE_CURVE('',#42,#43,#39,.T.);
```

Figure 3.6 Diameter of second spot weld in step file

.

3.4.3 Finding PLANE of the Second spot weld

OPEN SHELL with hash tag number #56 tells us the second spot weld , which contains pointer having hash tag #55 tells us the Advanced face , the number of surfaces present in it . spot weld contains only one surface which is drawn in the form of curves which meets at two different vertex points. PLANE number is #48 contains AXIS2_PLACEMENT_3D with (#47) in which three different pointers are present (#44,#45,#46) in which #44 tells us the Cartesian through which the plane passes in the direction of other two pointers #45 and #46.

```
#44=CARTESIAN_POINT('',(-7.088E1,1.079158279640E2,7.263E1));
#45=DIRECTION('',(8.470233271991E-1,5.315557197327E-1,0.E0));
#46=DIRECTION('',(5.315557197327E-1,-8.470233271991E-1,0.E0));
#47=AXIS2_PLACEMENT_3D('',#44,#45,#46);
#48=PLANE('',#47);
#55=ADVANCED_FACE('',(#54),#48,.T.);
#56=OPEN_SHELL('',(#55));
```

Figure 3.7 partial step file for plane of second spot weld

3.5 Third spot weld

Third open shell in a STEP file represents the third spot weld on the geometry. We will discuss some properties of the third spot weld like coordinates of weld, weld diameter and plane of weld.

3.5.1 Finding the coordinates of the Third spot weld

As shown in the figure below the line number in the Line 77 with hash tag number #84 which is the open_shell in STEP file and it is the third spot weld. It contains (#83) which tell us the advanced face and same phenomenon of pointers repetition is used for the third spots weld by only changing the pointers values. CARTESIAN_POINT with hash tag (#58) tell us the coordinates of the third spot welds i.e. (-50.82,73.85,146.15)

```
#58=CARTESIAN_POINT('',(-5.082E1,7.385022755610E1,1.4615E2));
#59=DIRECTION('',(-8.658522188003E-1,-5.002998452914E-1,0.E0));
#60=DIRECTION('',(-5.002998452914E-1,8.658522188003E-1,0.E0));
#61=AXIS2_PLACEMENT_3D('',#58,#59,#60);
#63=CARTESIAN_POINT('',(-5.082E1,7.385022755610E1,1.4615E2));
#64=DIRECTION('',(-8.658522188003E-1,-5.002998452914E-1,0.E0));
#65=DIRECTION('',(5.002998452914E-1,-8.658522188003E-1,0.E0));
#66=AXIS2_PLACEMENT_3D('',#63,#64,#65);
#68=CARTESIAN_POINT('',(-4.331550232063E1,6.086244427410E1,1.4615E2));
#69=CARTESIAN_POINT('',(-5.832449767937E1,8.683801083810E1,1.4615E2));
#70=VERTEX_POINT('',#68);
#71=VERTEX_POINT('',#69);
#72=CARTESIAN_POINT('',(-5.082E1,7.385022755610E1,1.4615E2));
#73=DIRECTION('',(8.658522188003E-1,5.002998452914E-1,0.E0));
#74=DIRECTION('',(5.002998452914E-1,-8.658522188003E-1,0.E0));
#75=AXIS2_PLACEMENT_3D('',#72,#73,#74);
#76=PLANE('',#75);
#78=ORIENTED_EDGE('',*,*,#77,.T.);
#80=ORIENTED_EDGE('',*,*,#79,.T.);
#81=EDGE_LOOP('',(#78,#80));
#82=FACE_OUTER_BOUND('',#81,.F.);
#83=ADVANCED_FACE('',(#82),#76,.T.);
#84=OPEN_SHELL('',(#83));
```

Figure 3.8 Coordinates of Third spot weld in step file

3.5.2 Finding Diameter of the Third spot weld

In continuation with Edge_curve. The Line 743 the second value in the bracket of circle command having hash number is #62 will gives us the radius of the circle . which is 1.5 mm from where we calculate the diameter by multiplying with 2 , which becomes 30mm.
As illustrated in the STEP file related to Diameter of the weld

```
#62=CIRCLE('',#61,1.5E1);
#67=CIRCLE('',#66,1.5E1);
#77=EDGE_CURVE('',#71,#70,#62,.T.);
#79=EDGE_CURVE('',#70,#71,#67,.T.);
```

Figure 3.9 Diameter of third spot weld in step file

3.5.3 Finding PLANE of the Third spot weld

AS shown in the figure 3.10 the entity number # 76 tells us the plane information which AXIS2 PLACEMENT 3D is linked in turn. In this entity # 75 the Cartesian point is indicated with #72 which tells us the information thorgh which PLANE passes, In the direction of other two entitines named #73 and #74.

```
#72=CARTESIAN_POINT('',(-5.082E1,7.385022755610E1,1.4615E2));
#73=DIRECTION('',(8.658522188003E-1,5.002998452914E-1,0.E0));
#74=DIRECTION('',(5.002998452914E-1,-8.658522188003E-1,0.E0));
#75=AXIS2_PLACEMENT_3D('',#72,#73,#74);
#76=PLANE('',#75);
```

Figure 3.10 partial step file of plane for third spot weld

3.6 Methodology

In the advanced STEP file reading software, the data extraction from the STEP neutral file as mentioned in chapter 3.1 is introduced WELDING FEATURE READER as described in the flow chart below:

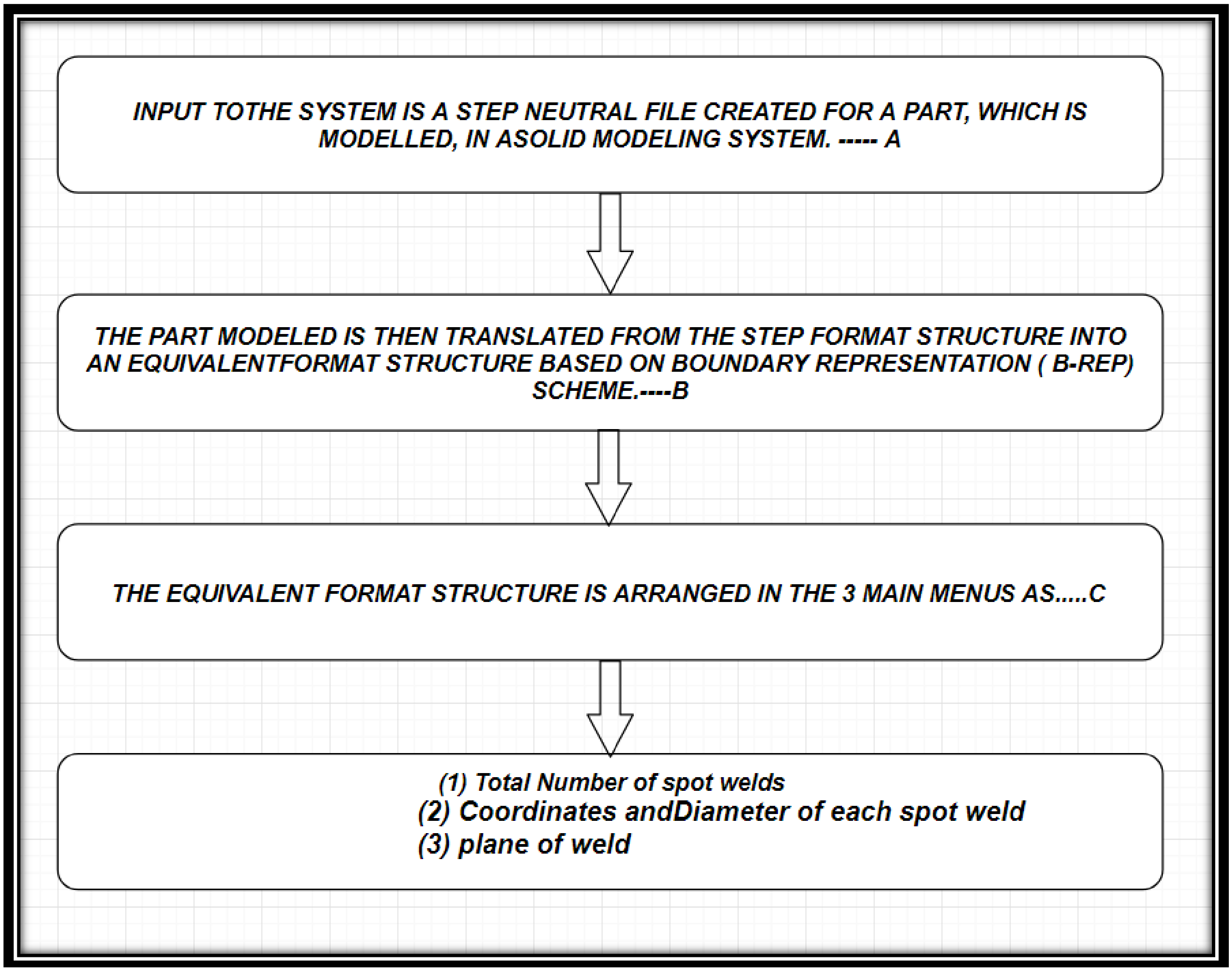

Figure 3.11 Flow chart of welding STEP reader

A ---- The software is designed to accept as input AP203 files(STEP file), which has the file extension as the.stp extension. The STEP file can be produced in any of the strong models such as PRO / E, CATIA etc. that have the STEP file export.

B--- Than translate the data into an equivalent Boundary representation (B-Rep).

C—Equivalent structure can be arranged in three main menu , means .Three main features of Spot welding are highlighted (1) Total Number of spot welds (2) Coordinates and Diameter of each spot weld (3) plane of weld.

3.7 Algorithm Steps

Different steps have been used in parsing STEP file to reach the extracted data from the design step data.

Step#1 First of all we read the STEP file in read only mode and then save the path of the given file. Save the file in local work space.at the end closes the file. whatever we done now will be in locally saved file.

Step#2 Split the file by semi colon and save in the form of array.

Step#3 Check step file validity, if we found cofig controlled design or not.

Step#4 Check the shell based surface model if it's true than parse this line number to find hash tag. And then count the number of hash tag in that line. From this we can find the total number of spots welds in design.

Step#5 Find open shell and then we get the lines of that token. And we got the information of advanced face by these tokens. Than match that token by parsing the file and reached the Advanced face of the open shell.

Step#6 Calculate the hash tag numbers in advanced face which tells us the information of face outer bound and PLANE. In the next step used hash tag of face outer bound will be used by the edge loop. Whereas plane information can be taken as it is.

Step#7 In the next step hash tags of ege loop will be scanned and then used these hash tags to find in oriented edges, these hash tags will give us the information of edge curve.

Step#8 Edge curve contains three hash tags; the last hash tag will tell us the information of circle. In circle the value without hash tag will gives us the radius of the spot weld whereas hash tags value will give us the coordinates of the spot weld.

Chapter - 4

CASE STUDY

4.0 CASE STUDY

This chapter deals with the case study of feature recognition of the Spot Welding features considered for this work. The case study here takes a part drawing drawn in Creo 3.0 for study. The part is drawn by using B spline curve and offset by 10mm after offsetting, project the curve so that the two overlapping curves are created with the upper curve which is rectangular in shape and slightly projected outward. The software developed in this work as WELDING FEATURE READER which generates the result and tells us the spot welding information by reading the Export file of the part considered. The Export file considered here is STEP Ap 203 neutral file.
Part is modeled in different Steps. Each step of case study is explained by its dimension and their geometry.

Geometry is drawn as Assembly in creo by Assemble the the two new parts in session, assemble. Than set them as fully constrained in the Assembly and also enables tree filters to fully understand each and every step.

4.1 B spline curve

B-spline equations are strong Bezier curve generalization. By using a unique set of mixing features that provide local impact, they provide local control of the curve form as opposed to worldwide control. They also enable control points to be added without raising the degree of the curve. As shown in the figure 14. B spline curve is used with knots to draw the curved shape. The distance taken from the origin for one half is equal to 150 mm same used for the other. The distance between two specific points on the curve is equal to 180 mm.

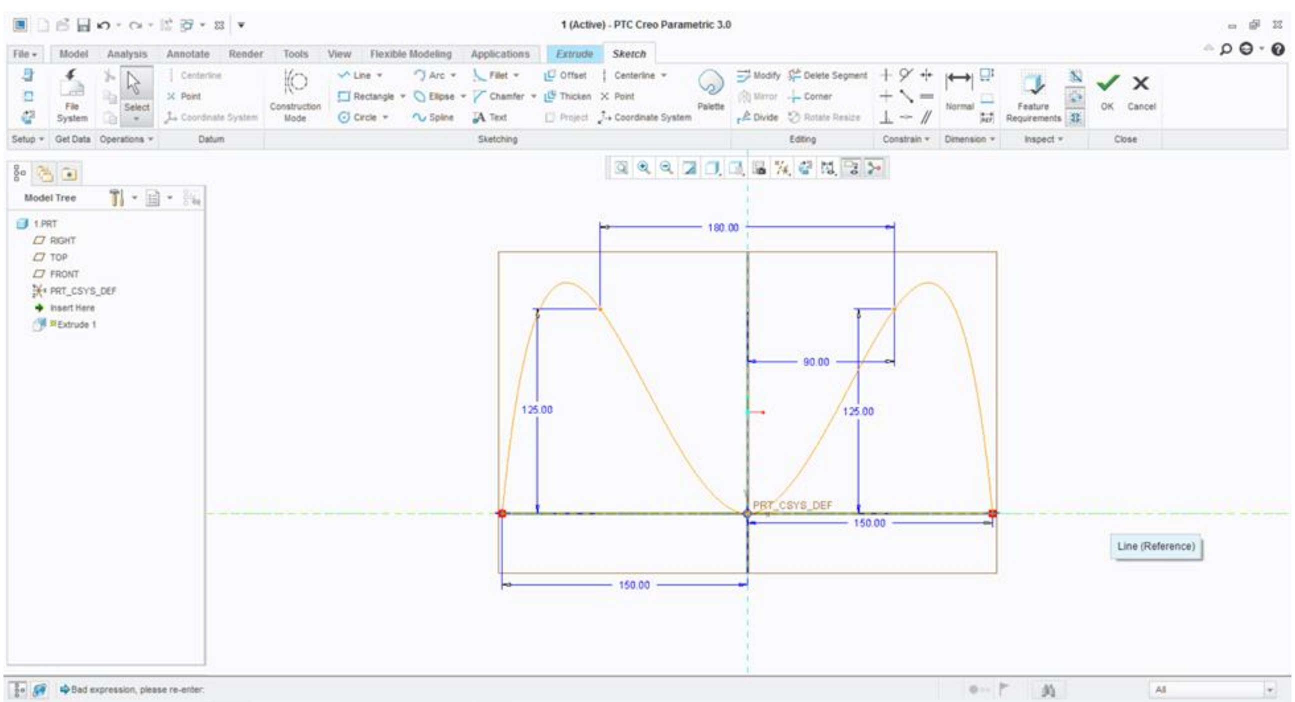

Figure 4.1 B spline curve with multiple knots

Than offset(Use the Offset tool to create a new feature by offsetting either a surface or a curve with a constant or variable distance) the Bspline curve by 10 mm . and than extrude by 200 mm in the z direction.
Shown in the figure below

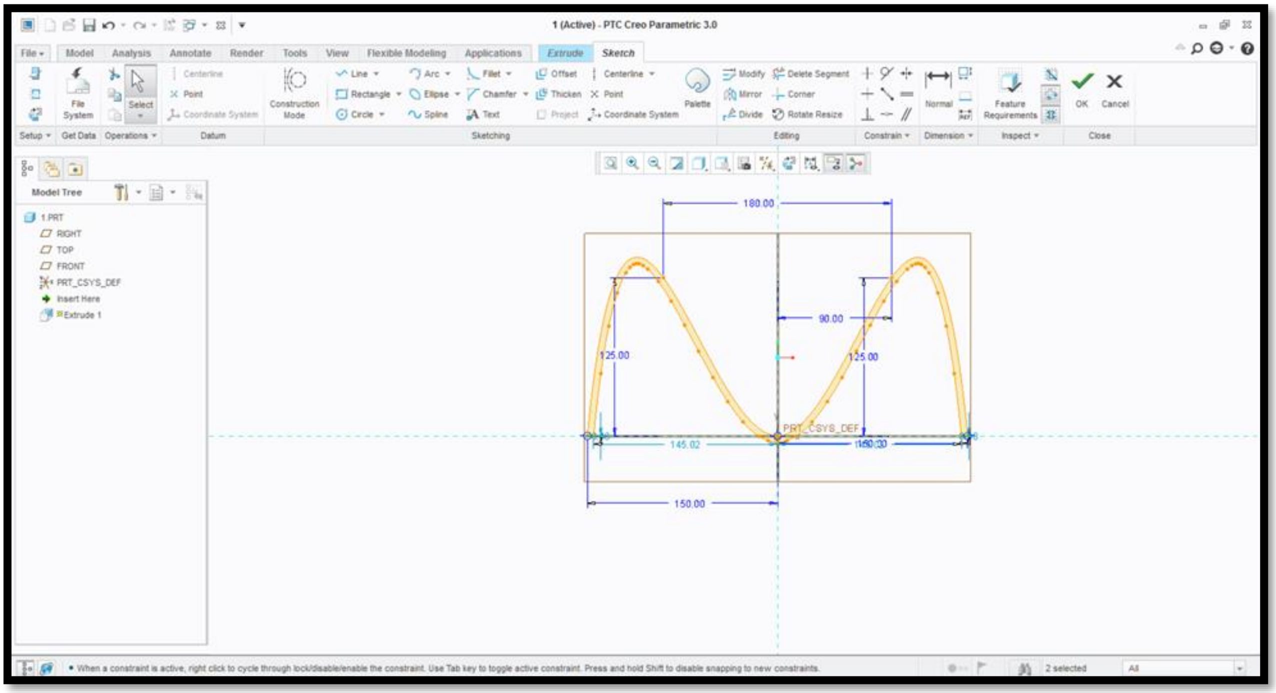

Figure 4.2 Offset tool used in B spline curve

Draw another curve by projecting the first curve by using the project command (Use the **Project** command to create Geometry by projection onto the sketching surface of chosen design curves or corners. The scheme aligns the entity's endpoints with the border endpoints. The formed organization has the sign "~" and and than offset by the same dimension. than closed the surface so that the geometry will be extruded.

4.2 First datum point for the First Spot weld

As shown in the figure, the first datum points on which first spot weld is used . the dimension of the datum points can be taken from the assembly axis because the parts are drawn in the assembly. The first point having name point 0 taken from the Assembly right plane with dimension of 34.82 mm and the second reference of this point can be taken from assembly front plane having dimension 78.50 mm. than draw the spot weld of 35mm.

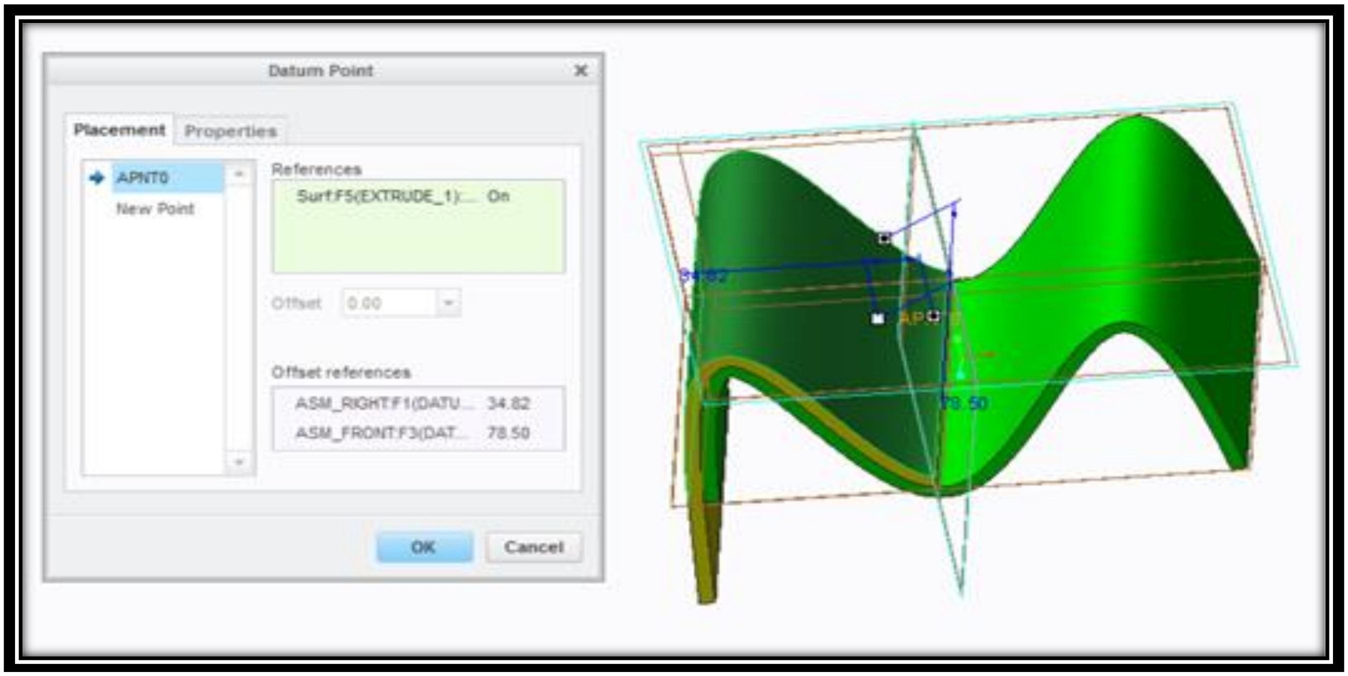

Figure 4.3 selection of first datum point

4.3 Second datum point for the Second Spot weld

As shown in the figure the Second datum point with its dimesnionn whose name is shown as point 1 which is taken from the assembly front plane of 72.63mm and assembly right plane at 70.88 mm .This point is drawn on the extrude surface f5.
Than spot a weld of 30 mm.

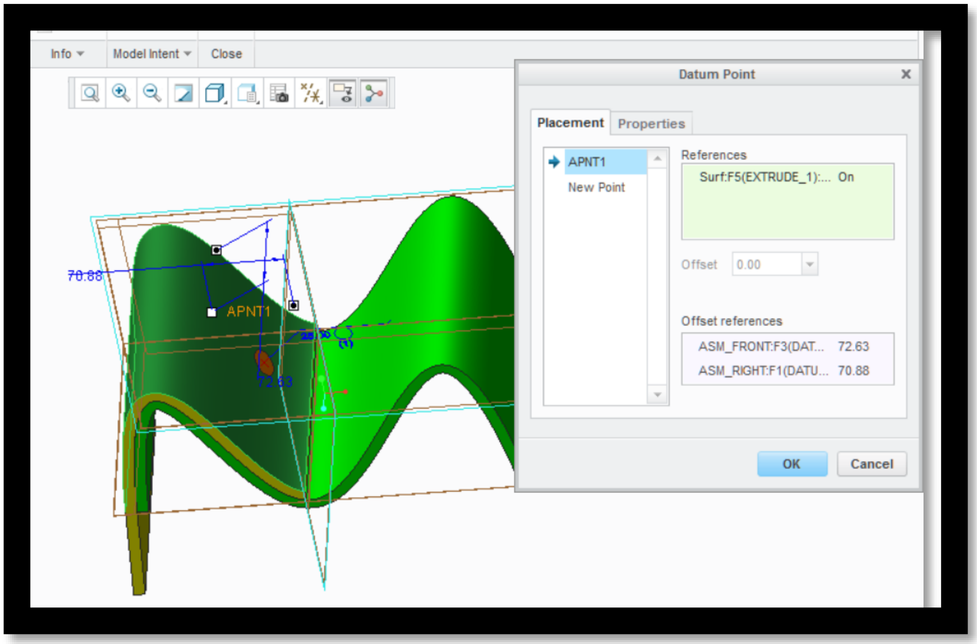

Figure 4.4 Second datum point with its Dimensions

4.4 Third datum point for the Third Spot weld

As shown in the figure the third point with a name APNT2 with its dimension.its aslo drawn on the f5 extrude surface . the dimension of the point from the assembly front plane is 146.15 mm where as the the dimension of the point from the assembly right plane is 50.82mm.third spot weld have dimeter is 25 mm.

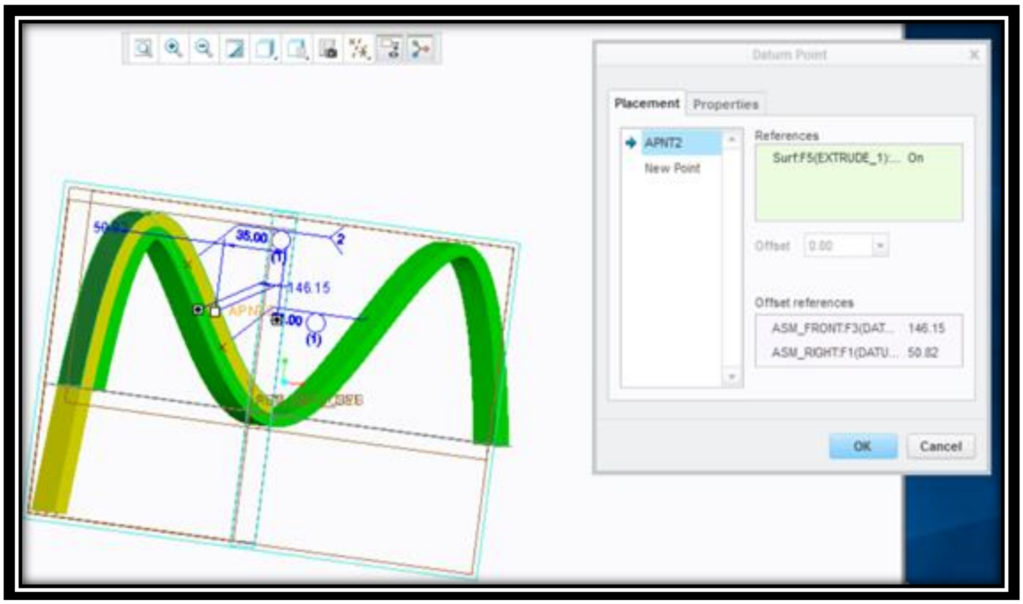

Figure 4.5 Third datum point with on Assembly plane

4.5 SPOT WELDs can be shown with different Orientations

Spots welds can be shown in different orientations to understand the case study in an illustrative manner. Different types are used one is used as wireframe view other is used as solid modelling view. All three welds are easily shown having different diameters on a curved shape.

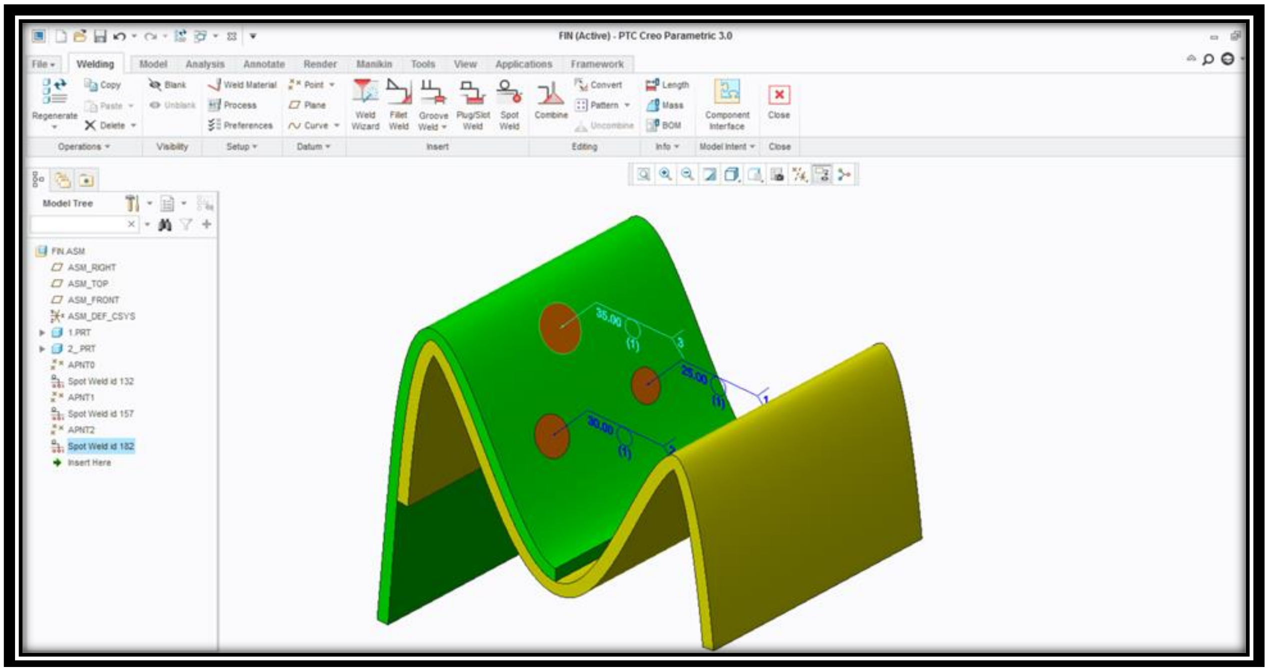

Figure 4.6 All the three spots weld are shown with its diameters

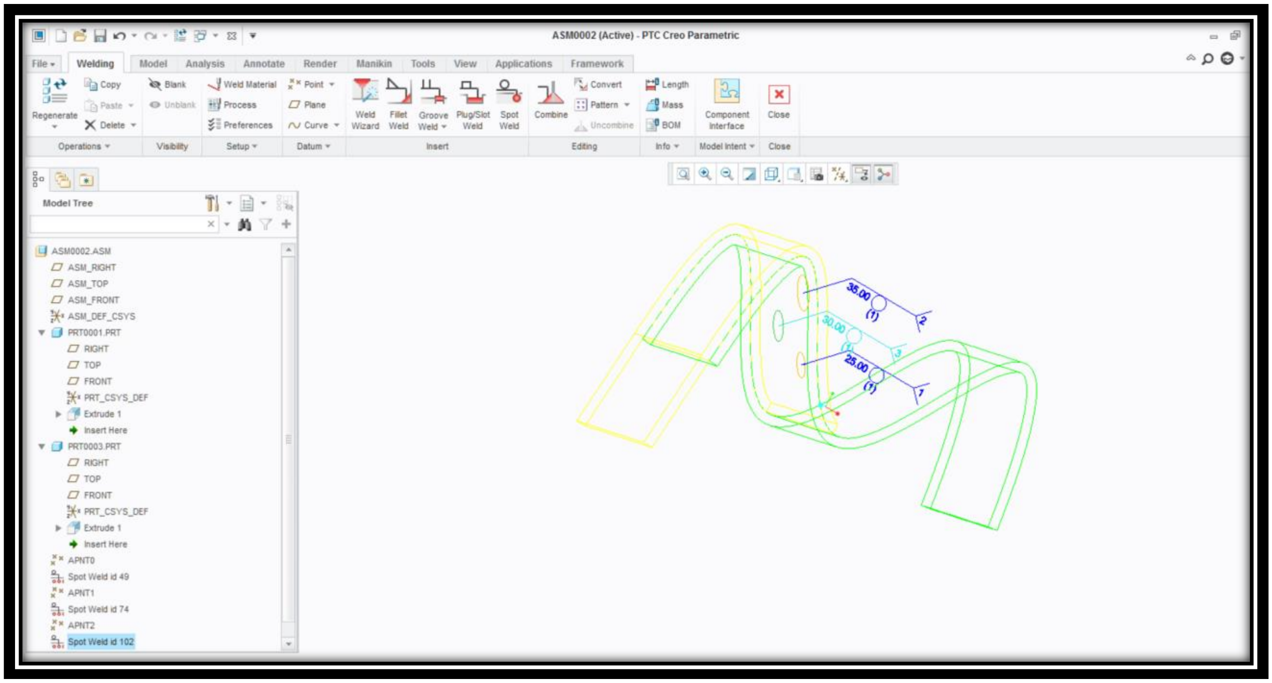

Figure4.7 :Wire frame view of the three spots welds

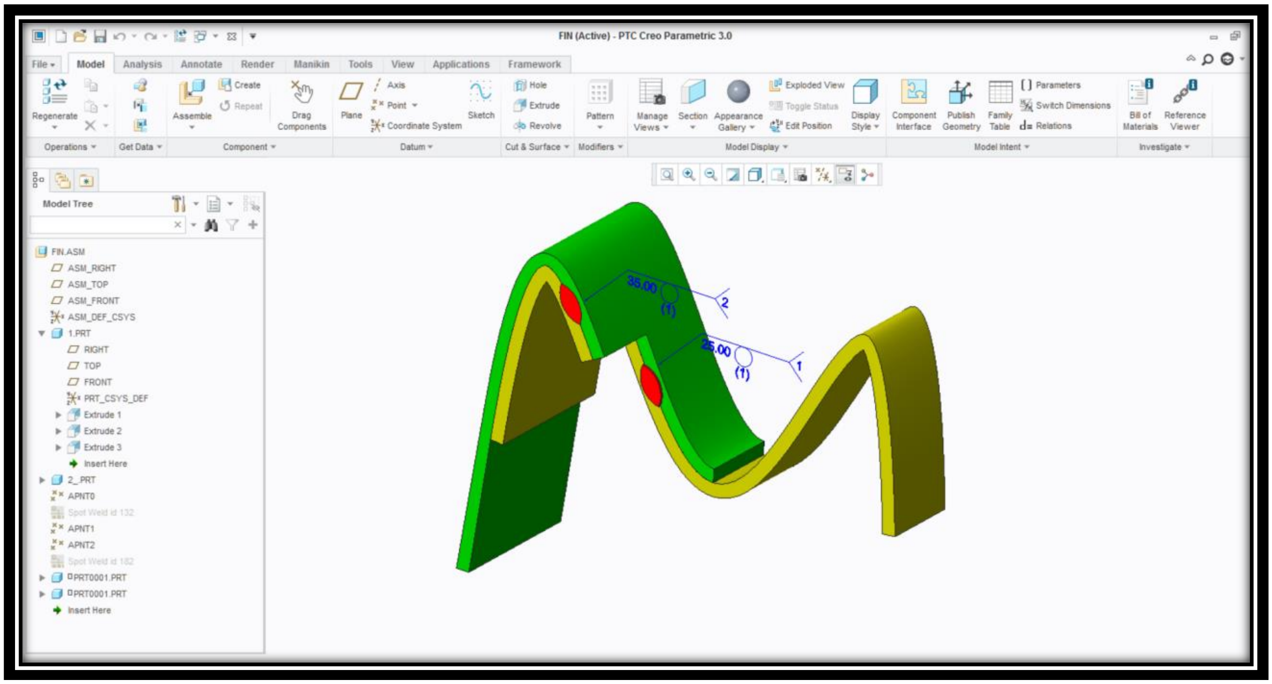

Figure 4.8: Sectionized view of the Spots Welds

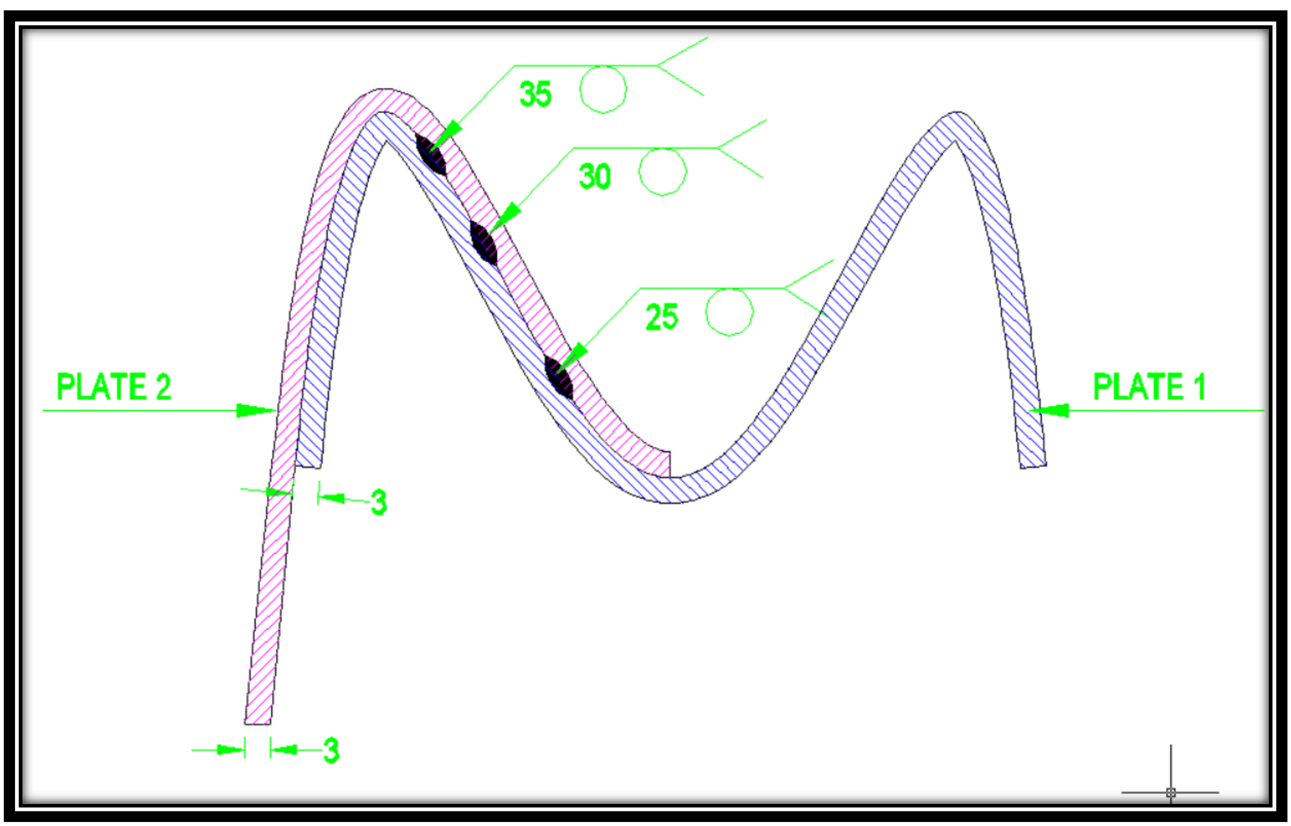

Figure 4.9: AutoCAD Drawing of the Case study

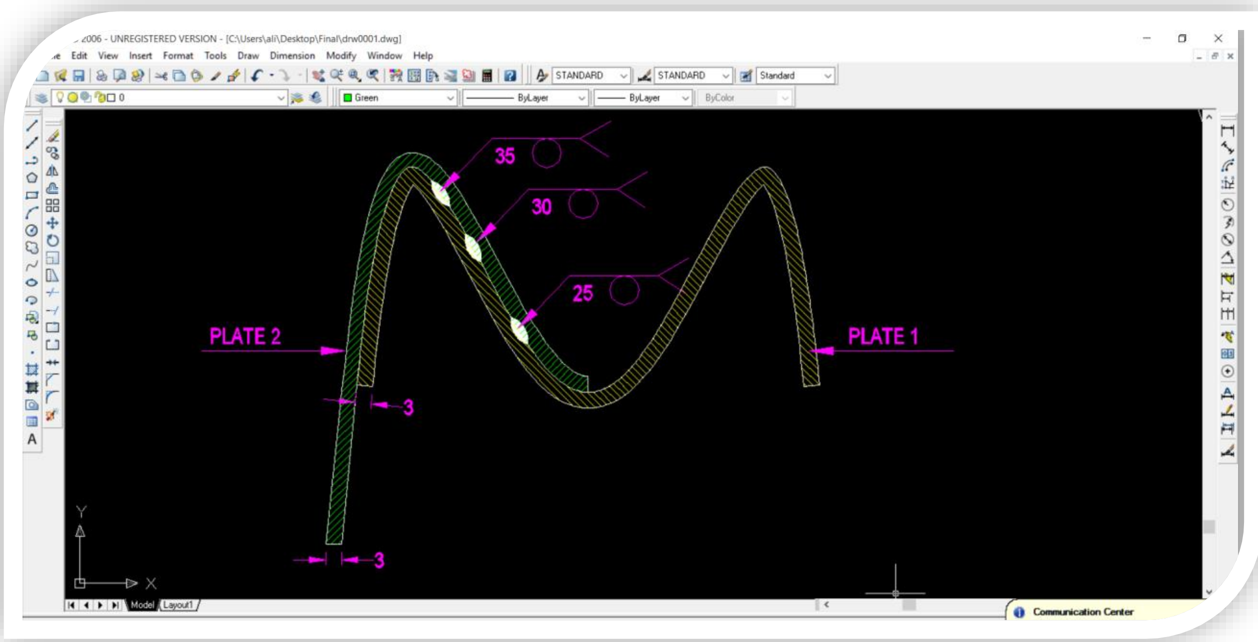

Figure 4.10 AutoCAD Drawing of three spot welds with dimensions

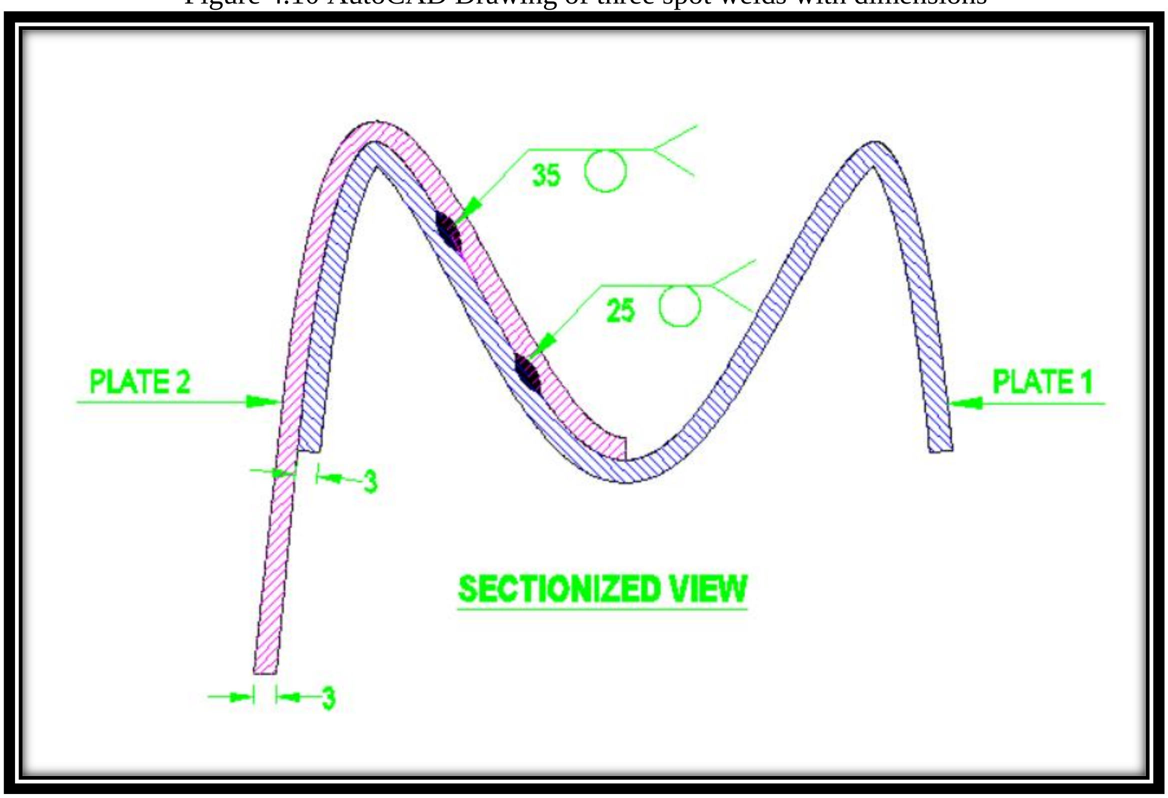

Figure 4.11 sectionized view of Weld

4.6 STEP file of the The case study

Partial STEP data can be displayed in this section before added the step file some of the preliminary knowledge has been needed to understand the step file data. Few terminologies has been added in this section to easily understand the STEP data .In addition to Definitions of the technical terms step file written text can aslo be shown.

(1) CARTESIAN Cordinates of a point in 3D

Specify the location of a point in **3D** With respect to **x,y,z**

Address of a point in Cartesian space. is defined by coordinate(x,y,z); direction is defined by direction ratios

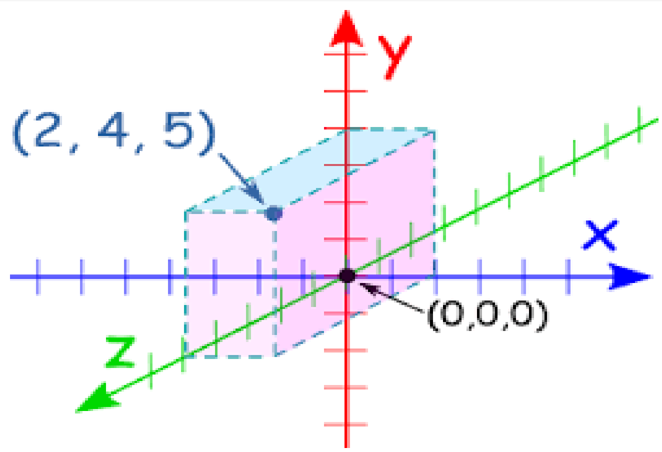

Figure 4.12 cartesian point in 3D

(2) Vector

A **vector** is a mathematical object that has a size, called the magnitude, and a direction

orientation : direction; and defined by magnitude : length_measure

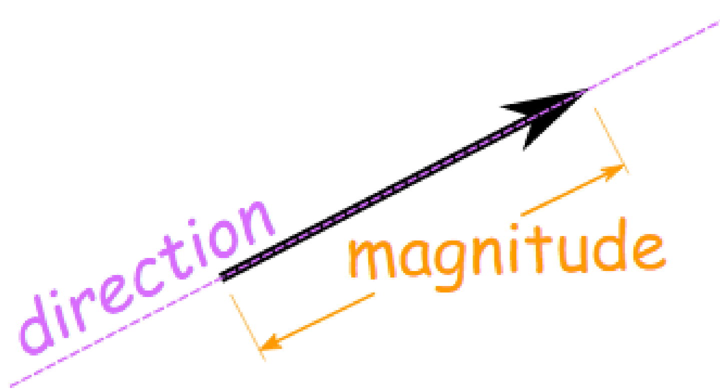

Figure 4.13 vector with its magnitude and direction

(3) DIRECTION RATIOS

Any numbers that are proportional to the direction cosines are called direction ratios

direction is defined by direction ratios in STEP

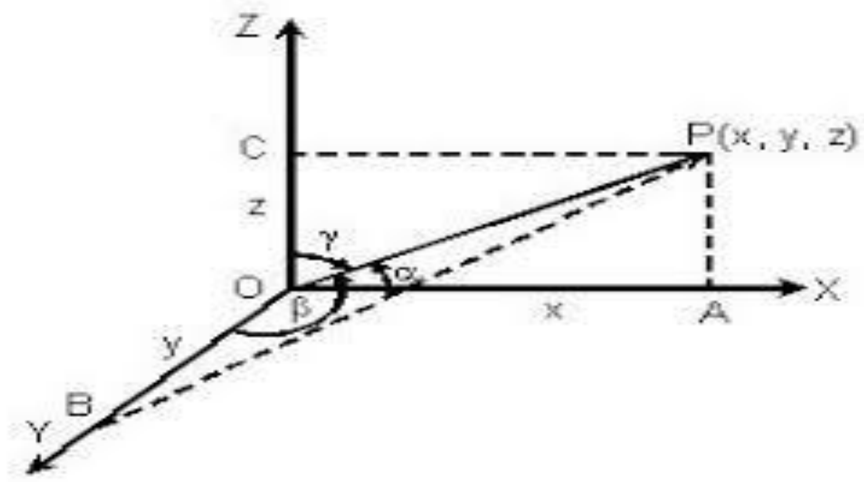

Figure 4.14 Direction ratios shown in 3D

(4) LINE

A row is a straight one-dimensional image without density in both directions and expanding indefinitely.

points : Cartesian point; and defined by dir : vector

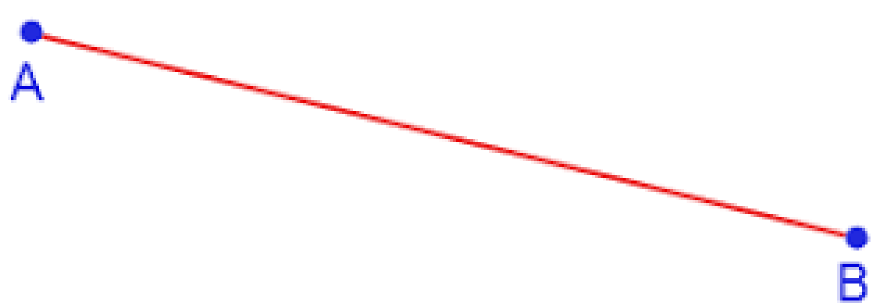

Figure 4.15 Line with two different points

(5) ORIENTED_EDGE

A edge which is made up of another **edge**

An edge constructed from another (original) edge and containing the direction (orientation) information is also defined by edge_start : vertex; edge_end : vertex;

(6) VERTEX_POINT

In geometry, a **vertex** (plural: **vertices** or vertexes) is a **point** where two or more curves, lines, or edges meet.

Vertex is defined by vertex_point and
vertex_geometry:point
vertex_point :A point defining the geometry of a vertex.

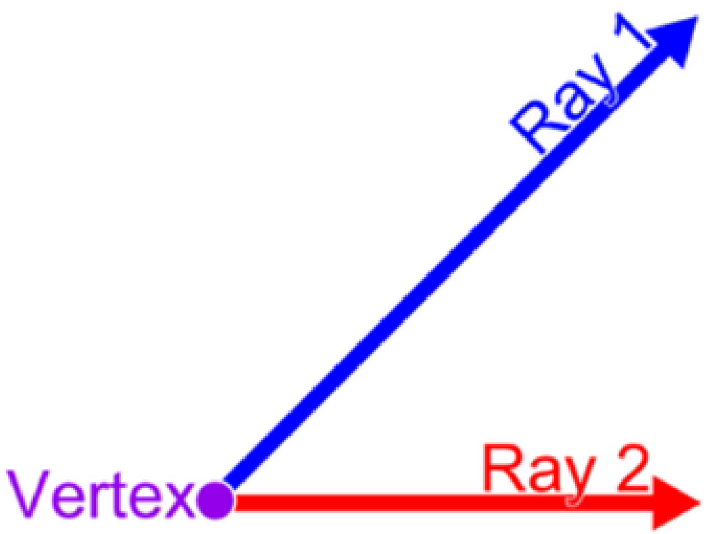

Figure 4.16 vertex point joining ray 1 and ray 2

(7) AXIS2_PLACEMENT_3D

(8)

The location of some specific point or plane etc

Placement is defined by axis2_placement_3d , which is defined by Cartesian point

(8) PLANE

A *plane* is a flat, two-dimensional surface that extends infinitely far

The PLANE pointer leads to AXIS2_PLACEMENT_3D

The AXIS2_PLACEMENT_3D contains three pointer ,one of the pointer leads to CARTESIAN-POINT , this is the point in x,y,z through which the plane crosses. The second and third pointer leads to the two Directions of the plane in x,y,z

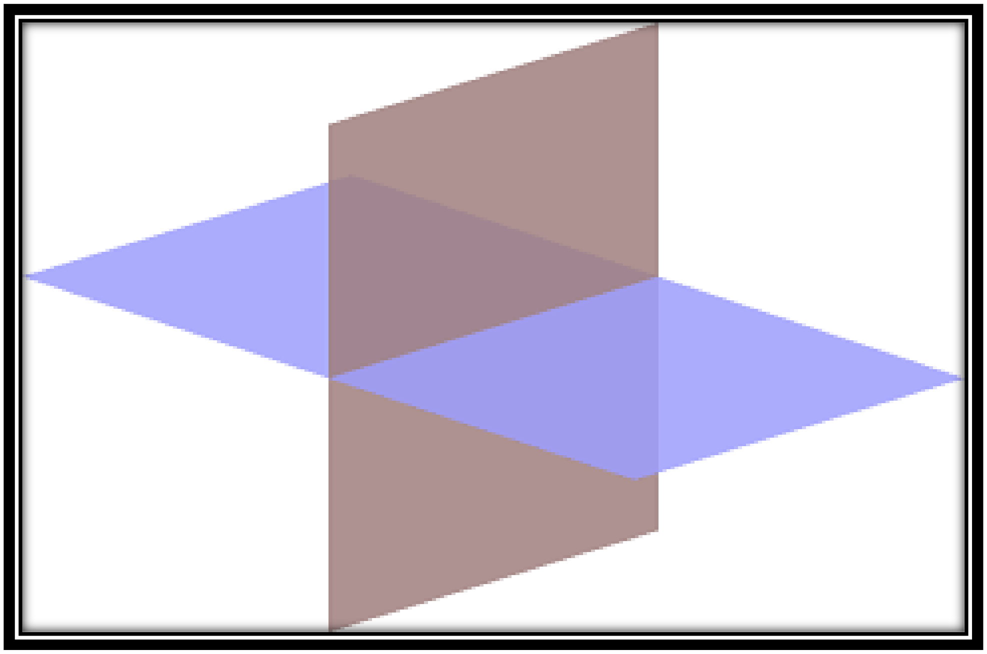

Figure 4.17 Diagrame of Plane

(9) EDGE_LOOP

A path in which the start and end vertices are the same

(9) face_outer_bound

face_bound A loop used for bounding a face is defined by
bound:loop; and orientation : BOOLEAN

(11) Advanced_face

FACE_SURFACE: A type of face in which the geometry is defined by the associated surface, boundary and vertices.

(12) MANIFOLD_SOLID_BREP

TYPE OF (solid_model); defined by outer : closed_shell

(13) CLOSED_SHELL

A collection of one or more faces, which bounds a region in three-dimensional space and divides the space into two regions, one finite and the other infinite.

(14) SHELL_BASED_SURFACE_MODEL

TYPE OF (surface_model); defined by outer : Open_shell

(15) OPEN_SHELL

It consist of Only One face,drawn in the form of Circle with the help of edge curve having two curves meet at the two vertex point

The area bound by two edge curve which meets at two vertex point in the form of circle

(16) EDGE_CURVE

Provides the information of circle

(17) Circle

A **circle** is a simple closed shape. It is the set of all points in a plane that are at a given distance from a given point, the centre; equivalently it is the curve traced out by a point that moves in a plane so that its distance from a given point is constant

Position axis2placement and radius:positive legth_measure

4.7 PARTIAL STEP FILE RELATED TO SPOT WELD

```
#2=CARTESIAN_POINT('',(-3.482E1,4.721586427335E1,7.85E1));
#3=DIRECTION('',(-8.421193301994E-1,-5.392912327346E-1,0.E0));
#4=DIRECTION('',(-5.392912327346E-1,8.421193301994E-1,0.E0));
#5=AXIS2_PLACEMENT_3D('',#2,#3,#4);
#7=CARTESIAN_POINT('',(-3.482E1,4.721586427335E1,7.85E1));
#8=DIRECTION('',(-8.421193301994E-1,-5.392912327346E-1,0.E0));
#9=DIRECTION('',(5.392912327346E-1,-8.421193301994E-1,0.E0));
#10=AXIS2_PLACEMENT_3D('',#7,#8,#9);
#12=CARTESIAN_POINT('',(-2.807885959082E1,3.668937264586E1,7.85E1));
#13=CARTESIAN_POINT('',(-4.156114040918E1,5.774235590084E1,7.85E1));
#14=VERTEX_POINT('',#12);
#15=VERTEX_POINT('',#13);
#16=CARTESIAN_POINT('',(-3.482E1,4.721586427335E1,7.85E1));
#17=DIRECTION('',(8.421193301994E-1,5.392912327346E-1,0.E0));
#18=DIRECTION('',(5.392912327346E-1,-8.421193301994E-1,0.E0));
#19=AXIS2_PLACEMENT_3D('',#16,#17,#18);
#20=PLANE('',#19);
#22=ORIENTED_EDGE('',*,*,#21,.T.);
#24=ORIENTED_EDGE('',*,*,#23,.T.);
#25=EDGE_LOOP('',(#22,#24));
#26=FACE_OUTER_BOUND('',#25,.F.);
#27=ADVANCED_FACE('',(#26),#20,.T.);
#28=OPEN_SHELL('',(#27));
#30=CARTESIAN_POINT('',(-7.088E1,1.079158279640E2,7.263E1));
#31=DIRECTION('',(-8.470233271991E-1,-5.315557197327E-1,0.E0));
#32=DIRECTION('',(-5.315557197327E-1,8.470233271991E-1,0.E0));
#33=AXIS2_PLACEMENT_3D('',#30,#31,#32);
#35=CARTESIAN_POINT('',(-7.088E1,1.079158279640E2,7.263E1));
#36=DIRECTION('',(-8.470233271991E-1,-5.315557197327E-1,0.E0));
#37=DIRECTION('',(5.315557197327E-1,-8.470233271991E-1,0.E0));
#38=AXIS2_PLACEMENT_3D('',#35,#36,#37);
#40=CARTESIAN_POINT('',(-6.157777490468E1,9.309291973806E1,7.263E1));
#41=CARTESIAN_POINT('',(-8.018222509532E1,1.227387361900E2,7.263E1));
#42=VERTEX_POINT('',#40);
#43=VERTEX_POINT('',#41);
#44=CARTESIAN_POINT('',(-7.088E1,1.079158279640E2,7.263E1));
#45=DIRECTION('',(8.470233271991E-1,5.315557197327E-1,0.E0));
#46=DIRECTION('',(5.315557197327E-1,-8.470233271991E-1,0.E0));
#47=AXIS2_PLACEMENT_3D('',#44,#45,#46);
#48=PLANE('',#47);
#50=ORIENTED_EDGE('',*,*,#49,.T.);
#52=ORIENTED_EDGE('',*,*,#51,.T.);
#53=EDGE_LOOP('',(#50,#52));
#54=FACE_OUTER_BOUND('',#53,.F.);
#55=ADVANCED_FACE('',(#54),#48,.T.);
#56=OPEN_SHELL('',(#55));
#58=CARTESIAN_POINT('',(-5.082E1,7.385022755610E1,1.4615E2));
#59=DIRECTION('',(-8.658522188003E-1,-5.002998452914E-1,0.E0));
#60=DIRECTION('',(-5.002998452914E-1,8.658522188003E-1,0.E0));
#61=AXIS2_PLACEMENT_3D('',#58,#59,#60);
#63=CARTESIAN_POINT('',(-5.082E1,7.385022755610E1,1.4615E2));
#64=DIRECTION('',(-8.658522188003E-1,-5.002998452914E-1,0.E0));
#65=DIRECTION('',(5.002998452914E-1,-8.658522188003E-1,0.E0));
#66=AXIS2_PLACEMENT_3D('',#63,#64,#65);
#68=CARTESIAN_POINT('',(-4.331550232063E1,6.086244427410E1,1.4615E2));
#69=CARTESIAN_POINT('',(-5.832449767937E1,8.683801083810E1,1.4615E2));
#70=VERTEX_POINT('',#68);
#71=VERTEX_POINT('',#69);
#72=CARTESIAN_POINT('',(-5.082E1,7.385022755610E1,1.4615E2));
#73=DIRECTION('',(8.658522188003E-1,5.002998452914E-1,0.E0));
#74=DIRECTION('',(5.002998452914E-1,-8.658522188003E-1,0.E0));
#75=AXIS2_PLACEMENT_3D('',#72,#73,#74);
#76=PLANE('',#75);
#78=ORIENTED_EDGE('',*,*,#77,.T.);
#80=ORIENTED_EDGE('',*,*,#79,.T.);
#81=EDGE_LOOP('',(#78,#80));
#82=FACE_OUTER_BOUND('',#81,.F.);
#83=ADVANCED_FACE('',(#82),#76,.T.);
#84=OPEN_SHELL('',(#83));
#85=SHELL_BASED_SURFACE_MODEL('',(#28,#56,#84))
```

Figure 4.18 Partial step file related to spot weld

Chapter - 5

RESULTS AND DISCUSSIONS

CHAPTER - V

RESULTS AND DISCUSSIONS

This section describes the legitimacy of the software package that has been created. It information how the customer uses the package for particular apps related to automatic feature recognition for spot welding . The software developed and coded has highlighted Three main features of Spot welding which are (1) Total Number of spot welds (2) Coordinates and Diameter of each spot weld (3) plane of weld. Case study has been taken to illustrate and describe how to recognize spot welding characteristics. The remaining sketches of the experiment element are shown in Appendix A.

5.1 PYTHON PROGRAM EXECUTION

User-friendly systems are intended and created in this bundle when running a python software a window closes as shown in Figure 5.1 and asks for a file to be opened. This software Only accept the filename with .stp extension or STEP file .As shown in the figure 5.1

When Entered and clicked OPEN button then a window opened which shows the totol number of spot weld on the main and each spot weld one by one in a column form. as shown in Figure 5.2.

Clicked on each spot weld one by one than it will gives us all the information about the spot weld

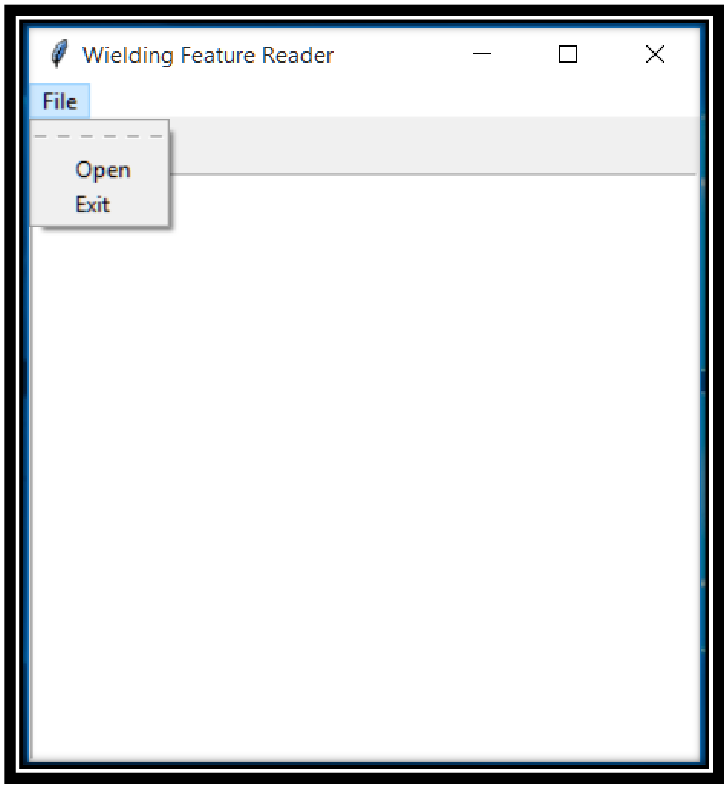

Fig. 5.1: First window asking for a step file

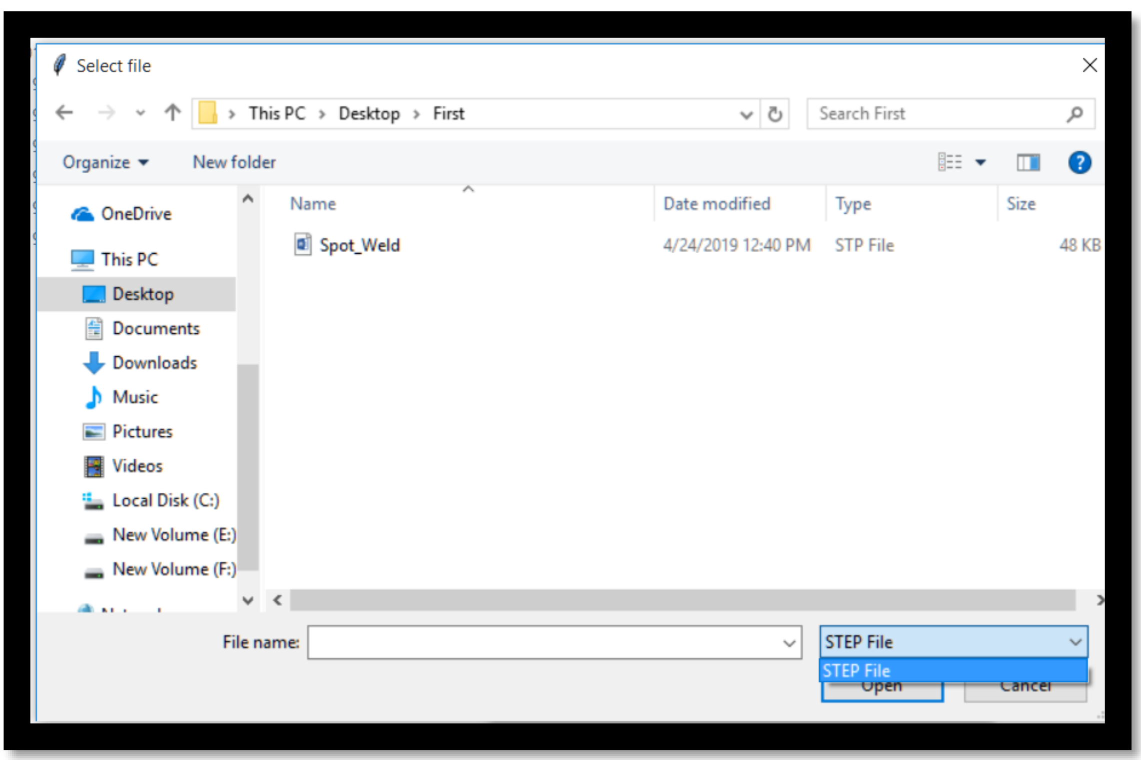

Fig. 5.2: Second window which shows it only Accept the file with .stp extension

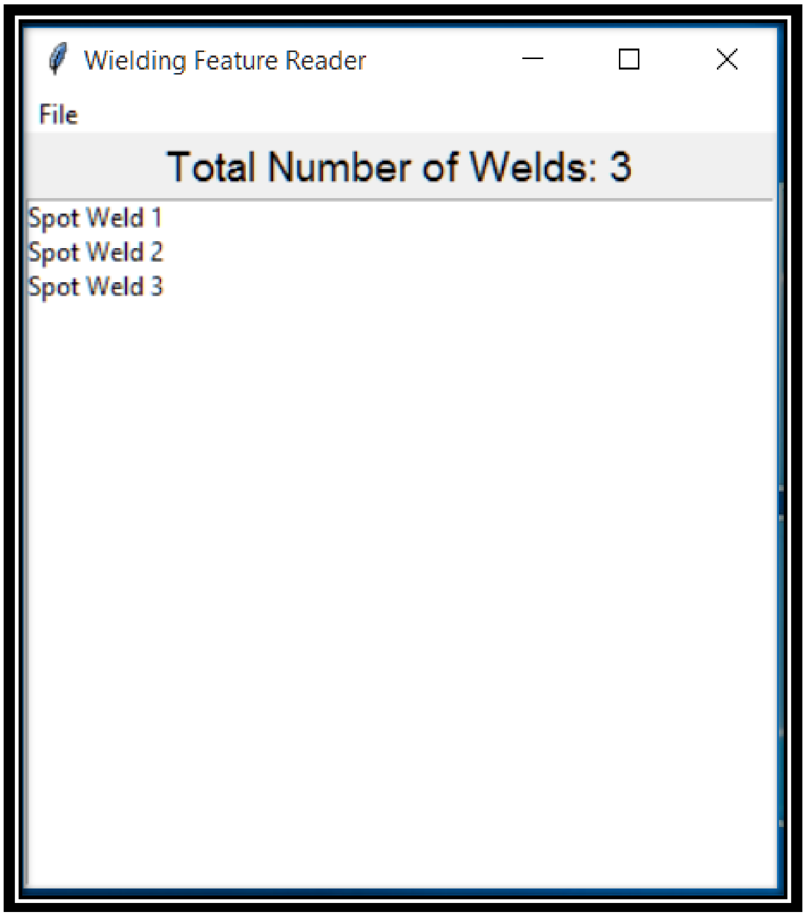

Fig. 5.3 After opened the STEP File

5.2 CAD MODEL

Figure 5.4 shows can be used as the Drawing of the case study. As illustrated in figure it consist of three spots welds with three different diameters having different coordinates All three spots welds are acknowledged with the assistance of the developed PYTHON program.

The characteristics of the problem like coordinates of spots weld , plane of spots welds and diameter of spot weld are obtained. The case study yield is shown in figure.

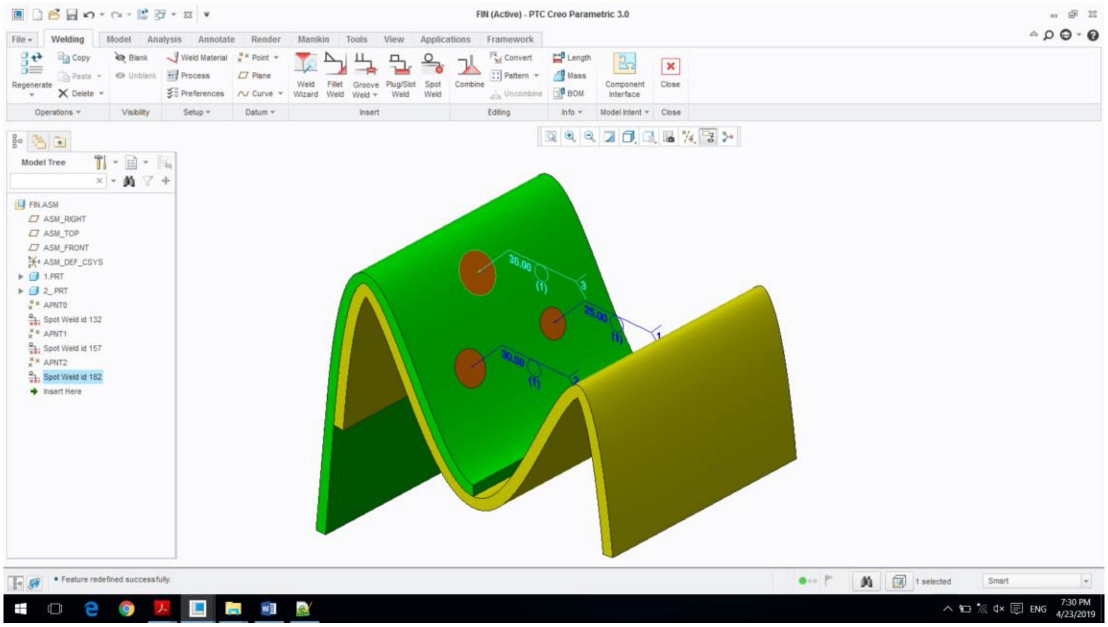

Figure 5.4 :Case study drawing with three spots welds

5.2 Results of the given case study

5.2.1 Results of Spot weld number one

Features of the first spot weld has been shown in the figure. The diameter of the first spot weld is 25 mm coordinates of the weld are (-34.820,47.216,78.500) whereas the information of plane has also been added point at which the plane passes is ((-34.820,47.216,78.500). direction cosines are also shown in which they are passes,
One thing should be focused at that point the the Cartesian point of weld and coordinates of spot weld are same.if these informations are same than the information which we get is the information of the spot weld.

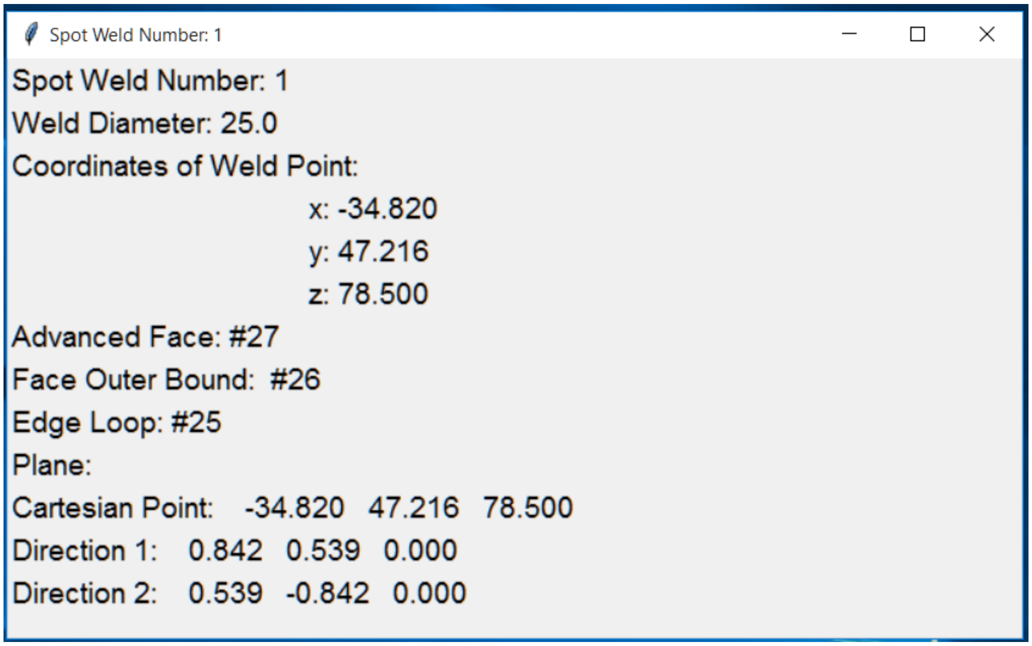

Figure 5.5 First spot weld number with its features

5.2.2 Results of Spot weld number 2

The diameter of the weld is 35 mm. coordinates of the weld are (-70.880,107.916,72.630). surface attached to them is also shown by advanced face hash tag number. Information of plane has also been shown cartseain point at which plane passes (-70.880,107.916,72.630)
Direction ratios 1 is(0.847,0.532,0) direction 2 (,0.532, -0.847,0)

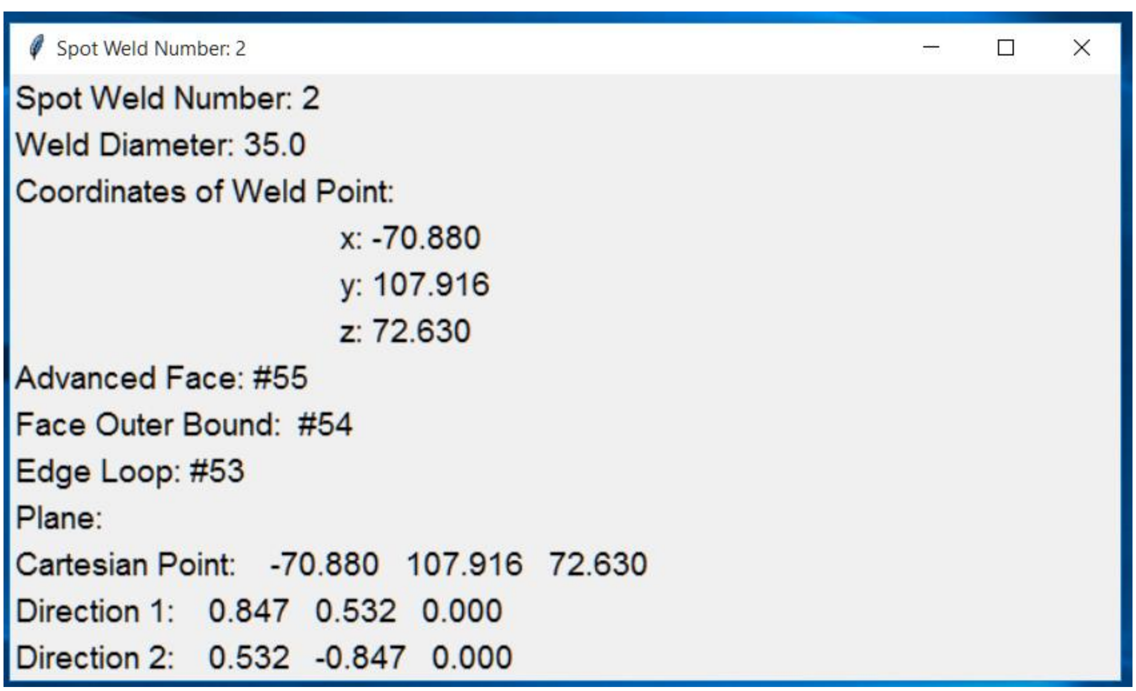

Figure 5.6 Second spot weld number with its features

5.2.3 Results of Spot weld number 3

Coordinates of the third spot weld are extracted as (-50.820,73.850,146.150) there weld diameter is also shown in figure 5.6 as 30 mm. hash tags of edge loop, face outer bound, and Advanced face is also shown which are (#81, #82, #83)

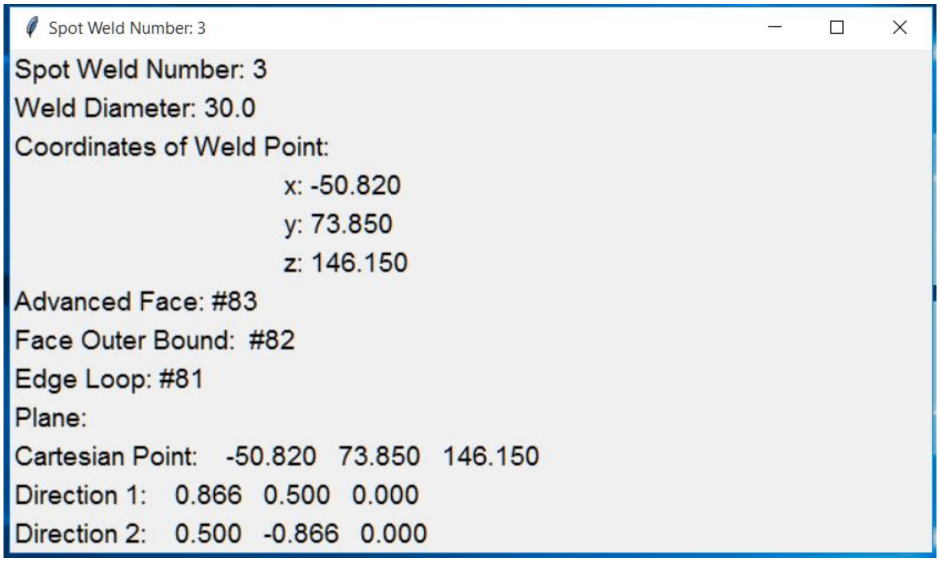

Figure 5.7 Third spot weld number with its features

5.3 Summarize the results in a table

All the results are shown in one table in which we show the different diameters of each spots welds and there coordinates axis with the PLANE in which they weld.

Spots Weld Features

		Coordinates Of Weld		
Spots Weld Numbers	Diameter (mm)	X	Y	Z
1	25	-34.820	47.216	78.500
2	35	-70.880	107.916	72.630
3	30	-50.820	73.850	146.150

Plane Information's

	Cartesian Points			Direction Ratios		
1	-34.820	47.216	78.500	0.842	0.539	0.000
				0.539	-0.842	0.000
2	-70.880	107.916	72.630	0.847	0.532	0.000
				0.539	-0.847	0.000

3	-50.820	73.850	146.150	0.866	0.500	0.00
				0.500	-0.866	0.00

Table 5.1 Summary of the complete results

Chapter - 6

CONCLUSIONS AND FUTURE SCOPE OF WORK

CHAPTER - VI

CONCLUSIONS AND FUTURE SCOPE OF WORK

A thorough analysis of the current literature was conducted to define the purpose and breadth of the current studies. The literature review points out that there has been no important quantity of job performed to obtain characteristics for welding from the STEP file.

6.1 CONCLUSIONS

An attempt has been made in the present investigation to design and develop a software capable of extracting spot welding data from STEP file and parameter recognition features. The program created can be implemented to agricultural auxiliary components such as thin curved sheets mostly used in the automobiles industries. Different types of welding processes are used in industry like fillet, groove or v butt, plug, slot or a spot weld for joining the material but This software facilitates the extraction of spot welding attributes. To fully create an automated Welded scheme from a STEP file to a identification function without human interference,

Within The following are the limitations of the current job general conclusion can be drawn.

1) 1) A theoretical analysis of the STEP neutral file was conducted
2) 2) Data removal from the STEP neutral folder is carried out using the WEDLING FEATURE READER software created.
3) The data prepared from the STEP file by the developed software is arranged as COORDINATES, DIAMETER, and PLANE

three modules are developed – namely 1) coordinates of the spot weld 2) total number of spots weld on the part 3) diameter of the weld and plane of the weld. The suggested algorithms are python-coded and can be performed on any IBM or compatible machines.

In brief, an effort was made to introduce a fresh technology, "STEP to welding feature" to the manufacturing globe. The main concept here for CAD / CAM inclusion is to use STEP file as the driving constraint in the identification of features and thus to give it as an input to Computer Aided Process Planning (CAPP).

6.2 SCOPE FOR FUTURE RESEARCH

- This dissertation is for place welding only. For fillet, it can be expanded, groove weld (including its types), plug/slot weld making the program far more complete.
- In advance research, the welding function reader can be upgraded to acknowledge characteristics such as material used for welding, pressure applied by the electrodes and current flowing from the material.
- Variation in the Thickness of the sheets can also be observed.
- It is possible to prepare effective process schedules using different characteristics from the welding function viewer.
- Process schedules can also be optimized.

REFERENCES

1-Pratt. M. J. , “Application of feature recognition in the Integrated Manufacturing,” in International Journal of Computer Aided Manufacturing, VoL 6(1 & 2),1993

2-Kyprianou L.K . “Shape classification in computer aided design.” PhD thesis, University of Cambridge, UK, 1980.

3-Proctor ,F. M. and Kramer,T. R.(1998) .“A feature Based machining system using STEP ‘’. National institute of standards and technology(NIST), Maryland:Gaithersburg, MD 20899-0001.

4-Lau, H.C.W., Lee, C.K.M., Jiang, B., Hui, I.K. and Pun, K.F. (2004) .“Development of a computer-integrated system to support CAD to CAPP”. International Journal of Advanced Manufacturing Technology.

5- Bojan ,B.and Zoran,M.(April, 2008). A review of automated feature recognition with rule-based pattern recognition, Computers in Industry.

6- Kannan, T.R. and Shunmugam,(2009) M.S. “Processing of 3D sheet metal components in STEP AP-203 format Part I: feature recognition system”. International Journal of Production Research ,47(4), 941-964 ,.

7- Suziyanti,M and Mohd F.M. A.(2010) “A review and Comparison og IGES and STEP”. World Academy of science and technology.

8- keong,c.w and yusri, y .(2012). “Developing A Step-Compliant Database System In Supporting Product Life Cycle Applications.Applied Mechanics and Materials (Volumes 229-231)

9-S. Sivakumar and V. Dhanalakshmi, . “A feature-based system for CAD/CAM integration through STEP file for cylindrical parts”, Indian Journal of Engineering & Material Sciences, vol. 20, February 2013, pp. 21-26.

10-V. N. Malleswari, Dr. P. M.Valli and Dr. M. M. M. Sarcar,(March ,2013) . "Automatic Recognition of Machining Features using STEP Files", International Journal of Engineering Research & Technology (IJERT), ISSN: 2278-0181, Vol. 2 Issue 3,.

.11- Jaider,O , Elmesbahi,A and Ahmed,R.(2014) "automatic
feature recognition for rotational parts"World Congress on Computational Mechanics (11)

12-Venua, B, Kommab,V.R, Srivastavac,D (2015). "A Methodology to Recognize Manufacturing Features of Prismatic Parts from STEP AP 203 File". International Conference on Materials Processing and Characterization(4)

13-Zehtaban,L, Elazhary,O, Roller,D (2016), "A framework for similarity recognition of CAD models". Journal of Computational Design and Engineering (3) pp, 274–285

14-Venua, B, Kommab,V.R, Srivastavac,D (2018). "STEP-based Feature Recognition System for B-spline Surface Features". International Journal of Automation and Computing,pp.(500-512).

15-Hsu-Pin, Wang and Ching Ann, Lin, 1987. Automated Generation of NC part programs for turned parts based on 2D drawing files, International Journal of Advanced Manufacturing Technology, 2(3), 23-35.

16-Srinivasa kumar, S., Madurai and Li Lin, 1992. Ruled based automatic part feature extraction and recognition from CAD data, Computers Ind. Engineering, 22(1), 49-62.

17-Wang, T.N. and Wong, K.W. 1995. A feature based design system for computer aided process planning. Journal of Material Processing Technology, 52, 122-132.

18-Sabourin, L. and Villeneuve, F., 1996, Omega, an expert CAPP system, Advances in Engineering Software, 25, 51-59

19-Jung, M.Y., and Lee, K.H., 1996, A CAD/CAPP interface for complex rotationally symmetric parts, International Journal of Production Research, 34(1), 227-251.

20- Fuh, J.Y.H., Chang, C.H. and Melkanoff, M.A., 1996, The development of an integrated and intelligent CAD/CAPP/CAFP environment using logic based reasoning, Computer Aided Design, 28,217-232.

21- Yang, M.Y. and Lee, S.C., 1998. A feature modification framework for the generation of alternative process plans, International Journal of Production research. 36(7), 1825-1840.

22- Gao, S and Shah, J.J., 1998, Automatic recognition of interacting machining features based on minimal condition sub graph, Computer Aided Design, 30, 727-739.

23-.Jiang, B., Lau, H., Chan, F.T.S and Jiang, H., 1999, An automatic process planning system for the quick generation of manufacturing process plans directly from CAD drawings, Journal of Material Process Technology, 87, 97-106.

24- Bhandarkar, M.P and Nagi, R., 2000. STEP based feature extraction from STEP geometry for agile manufacturing, Computers in Industry, 41, 3-24.

25- Saravanan,R., Siva Sankar,R. Asokan,P. Vijayakumar,K and Prabharan, G., 2005. Optimization of cutting conditions during continuous finished profile machining using non-traditional techniques, International Journal of Advanced Manufacturing Technology, 26, 30-40.

26- Ong, S.K. and Li, W.D. and Nee, A.Y.C., 2003. STEP based integration of feature recognition and design by feature for manufacturing applications in a concurrent engineering environment, International Journal of Computer Applications in Technology, 18(1-4), 78-92

27-Julian Fowler : *STEP for Data Management,Exchange and Sharing.*

28. *https://www.steptools.com/*

APPENDIX - A

PYTHON CODE FOR (Graphical User Interface) GUI

```
"""
  -- ttk.Entry
  -- Using its GET and SET methods to get/set an Entry's information
     (as opposed to using a CONTROL VARIABLE as in a subsequent module)
"""

import tkinter
from tkinter import ttk
from tkinter import *
from tkinter.filedialog import askopenfilename
import tkinter.messagebox
import tkinter as tk

from spot_manager import SpotManager
from spot_model import SpotModel
from stp_parser import STEPFileParser

# Creating object of STEP File Parser
stp_file_parser = STEPFileParser()
stp_manager = SpotManager()
list_box = None


##################################################################################
#####

class STPReader(object):
    list_box = None
    error = None

    @staticmethod
    def locate_file_and_process(file_path):

        # Load the file and find spots
        if stp_file_parser.file_reader(file_path):

            if stp_file_parser.check_step_file_validity():
                # Finds all spots in the file
                stp_file_parser.spots_found =
stp_file_parser.find_shell_based_surface_model()

            else:
                STPReader.error = "Unable to validate STP file. "
        else:
            STPReader.error = "Unable to read file."

        # If spots are found and there are no errors
        if stp_file_parser.spots_found and STPReader.error is None:

            # Pass the spots to spot_model object
            for idx, item in enumerate(stp_file_parser.spots_found):
                # Calculate the spot and all its properties using the hashed
number begining
                spot_obj = SpotModel(item, stp_file_parser.lines_list)

                # Add the calculated spot in the manager
                stp_manager.add_spot(spot_obj)

    @staticmethod
    def OpenFile(list_box, total_welds_label):
        """
        Sets the window to open only ".stp" file.

        :param list_box:
```

```
        :return:
        """
        file_path = askopenfilename(
            title="Select file",
            filetypes=(("STEP File", "*.stp"),)
            # ("all files","*.*"))
        )

        # main function
        STPReader.locate_file_and_process(file_path)

        if stp_manager.all_spot.__len__() > 0:
            for index, item in enumerate(stp_manager.all_spot):
                list_box.insert(index, "Spot Weld " + str(index + 1))

            total_welds_label.config(text="Total Number of Welds: " +
str(stp_manager.all_spot.__len__()))
        else:
            tk.messagebox.showerror("Error", STPReader.error)

    @staticmethod
    def open_window(weld_index):

        # def create_window():
        #     window = tk.Toplevel(root)
        # b = tk.Button(root, text="Create new window", command=create_window)
        # b.pack()

        spot = stp_manager.all_spot[weld_index]

        root = tk.Tk()
        frame = tk.Frame(master=root, width=250, height=200, )
        frame.pack()

        s = tk.Label(frame, width=60, text="Spot Weld Number: " + str(1 +
weld_index), anchor="w")
        s.config(font=("Arial", 14))
        s.pack()

        root.title("Spot Weld Number: " + str(1 + weld_index))

        w = tk.Label(frame, width=60, text="Weld Diameter: " +
str(spot.circle_radius), anchor="w")
        w.config(font=("Arial", 14))
        w.pack()

        x = tk.Label(frame, width=60, text="Coordinates of Weld Point: ",
anchor="w")
        x.config(font=("Arial", 14))
        x.pack()

        a = tk.Label(frame, width=60, text="
" + str(spot.display_x()), anchor="w")
        a.config(font=("Arial", 14))
        a.pack()

        b = tk.Label(frame, width=60, text="
" + str(spot.display_y()), anchor="w")
        b.config(font=("Arial", 14))
        b.pack()

        c = tk.Label(frame, width=60, text="
" + str(spot.display_z()), anchor="w")
        c.config(font=("Arial", 14))
```

```
        c.pack()

        e = tk.Label(frame, width=60, text="Advanced Face: " +
str(spot.display_face_outer_bound()), anchor="w")
        e.config(font=("Arial", 14))
        e.pack()

        d = tk.Label(frame, width=60, text="Face Outer Bound:  " +
str(spot.display_advanced_face()), anchor="w")
        d.config(font=("Arial", 14))
        d.pack()

        f = tk.Label(frame, width=60, text="Edge Loop: " +
str(spot.display_edge_loop()), anchor="w")
        f.config(font=("Arial", 14))
        f.pack()

        g = tk.Label(frame, width=60, text="Plane:", anchor="w")
        g.config(font=("Arial", 14))
        g.pack()

        h = tk.Label(frame, width=60, text="Cartesian Point: " +
str(spot.display_aaa()), anchor="w")
        h.config(font=("Arial", 14))
        h.pack()

        j = tk.Label(frame, width=60, text="Direction 1: " +
str(spot.display_bbb()), anchor="w")
        j.config(font=("Arial", 14))
        j.pack()

        k = tk.Label(frame, width=60, text="Direction 2: " +
str(spot.display_ccc()), anchor="w")
        k.config(font=("Arial", 14))
        k.pack()

        root.mainloop()


##################################################################################
###########

def main():
    # Data object to hold information needed for callbacks.

    # Root window and Frame on it.
    root = tk.Tk()
    root.geometry("350x350")

    # frame = tk.Frame(root, padding=400)
    frame = tk.Frame(master=root, width=400, height=450, )
    frame.pack()

    # Total Welds Label
    total_welds_label = Label(frame, text="")
    total_welds_label.config(font=("Arial", 14))
    total_welds_label.pack()

    # ListBox
    list_box = Listbox(frame, width=350, height=250)

    # list_box = Listbox(frame)

    # Label Welcome Message
```

```
    # label = tk.Label(frame, width=10, height = 20, text="Welcome to STP Reader",
font=("Helvetica", 16))
    # label.pack()
    # list_box.insert(1, "Python")
    # list_box.insert(2, "Perl")
    # list_box.insert(3, "C")
    # list_box.insert(4, "PHP")
    # list_box.insert(5, "JSP")
    # list_box.insert(6, "Ruby")

    # List Box Event
    # This function is executed whenever a user clicks on the list box.
    def onselect(evt):
        # Note here that Tkinter passes an event object to onselect()
        w = evt.widget
        index = int(w.curselection()[0])
        value = w.get(index)
        print('You selected item %d: "%s"' % (index, value))

        # Clicked spot well index is passed as parameter
        STPReader.open_window(index)

    # lb = Listbox(frame, name='lb')
    list_box.bind('<<ListboxSelect>>', onselect)
    list_box.pack()

    # Menu Bar
    menu = Menu(root)
    root.config(menu=menu)
    file = Menu(menu)
    # Attaching a open file function to the "Open", passing list_box as parameter
to populate the list box
    file.add_command(label='Open', command=lambda: STPReader.OpenFile(list_box,
total_welds_label))
    file.add_command(label='Exit', command=lambda: exit())
    menu.add_cascade(label='File', menu=file)

    # Title
    root.title("Wielding Feature Reader")

    # Center
    # root.minsize(width=100, height=100)
    # root.geometry('800x650+50+50')
    # w = 400
    # h = 450
    # ws = root.winfo_screenwidth()
    # hs = root.winfo_screenheight()
    # x = (ws / 2) - (w / 2)
    # y = (hs / 2) - (h / 2)
    # root.geometry('%dx%d+%d+%d' % (w, h, x, y))

    # Built in function to start the gui
    root.mainloop()


# ----------------------------------------------------------------------
# Calls  main  to start the ball rolling.
# ----------------------------------------------------------------------
main()
```

APPENDIX - B

STP_PARSER CODE

```
#classes generated in python code

class STEPFileParser():

    def __init__(self):
        self.lines_list = None
        self.file_path = None
        self.complete_file = None
        # Number of initial points to detect
        self.spots_found = None


        # temp path
        #self.path = "C:/Users/Shuayb/Desktop/Parser/1spot.stp"

    def file_reader(self,file_path):
        """
        :return: True if successfully able to read file
        """

        # A try catch block to check if file actually exists or not
        try:
            # Open file in read mode, read the complete file
            stp_file = open(file_path, 'r')

            # Save file path
            self.file_path = file_path

            # Read complete file
            self.complete_file = stp_file.read()

            # Close the file, this releases the file from the memory of the
program.
            stp_file.close()

            # Split the file by ';'
            self.lines_list = self.complete_file.split(';')

            # Strip all from the lines_list
            # map(str.strip, self.lines_list)
            self.lines_list = list(map(str.strip, self.lines_list))
            return True
        except FileNotFoundError:
            return False

    def check_step_file_validity(self):
        """
        :return: True if successfully able to find "CONFIG_CONTROL_DESIGN"
        """

        characters_to_find = "CONFIG_CONTROL_DESIGN"

        for line in self.lines_list:
            if characters_to_find in line:
                return True
        return False

    def find_shell_based_surface_model(self):
        characters_to_find = "SHELL_BASED_SURFACE_MODEL"

        for line in self.lines_list:
            if characters_to_find in line:
                return STEPFileParser.find_hashed_numbers(line)
```

```
    @staticmethod
    def find_hashed_numbers(str):
        """
        A simple function that finds hashed numbers (generic function)

        :param str: Takes in a string containing brackets ()
        :return: Points between brackets
        """
        find_beginning_bracket_index = str.index('(')
        find_ending_bracket_index = str.index(')')

        substring_with_hashed_numbers = ""

        final_list = list()

        # separate a string between ( )
        for index, item in enumerate(str):
            if index > find_beginning_bracket_index and index <
find_ending_bracket_index:
                substring_with_hashed_numbers += item

        # print(substring_with_hashed_numbers)
        hashes_index = list()
        if substring_with_hashed_numbers.__len__() > 0:

            # Find index of hashes in the substring that is between "(" and ")"
            for index, item in enumerate(substring_with_hashed_numbers):
                if item == "#":
                    hashes_index.append(index)

            if hashes_index.__len__() > 0:

                for index, item in enumerate(hashes_index):
                    sequence_to_locate = ""

                    count_index = item + 1
                    while True:
                        try:
                            if
(substring_with_hashed_numbers[count_index].isnumeric()):
                                sequence_to_locate +=
substring_with_hashed_numbers[count_index]
                            else:
                                final_list.append(sequence_to_locate)
                                break
                            count_index += 1
                        except IndexError:
                            final_list.append(sequence_to_locate)
                            break

        # Adding a prefix # to the found points
        for index, item in enumerate(final_list):
            item = "#" + item
            final_list[index] = item

        return final_list
```

APPENDIX - C

SPOT _MODEL CODE

```
import math

from stp_parser import STEPFileParser


class SpotModel:
    spot_beginning = None
    lines_list = None

    # Related to OPEN_SHELL
    open_shell_line = None
    open_shell_brackets_list = None

    # Related to ADVANCED_FACE
    advanced_face_line = None
    advanced_face_brackets_list = None

    # Related to FACE_OUTER_BOUND
    face_out_bound_line = None
    face_out_bound_line_brackets_list = None

    # Related to PLANE
    plane_line = None
    plane_brackets_list = None

    # Related to axis2_placement_3d
    axis2_placement_3d_line = None
    axis2_placement_3d_brackets_list = None

    # Related to EDGE_LOOP
    edge_loop_line = None
    edge_loop_brackets_list = None

    # Related to ORIENTED_EDGE
    oriented_edge_line = None
    oriented_edge_brackets_list = None

    # Related to EDGE_CURVE
    edge_curve_line = None
    edge_curve_brackets_list = None

    # Related to CIRCLE
    circle_line = None
    circle_brackets_list = None
    circle_radius = None

    # Related to AXIS_PLACEMENT
    axis_placement_line = None
    axis_placement_brackets_list = None

    # Related to CARTESIAN_POINT
    cartesian_point_line = None
    cartesian_point_brackets_list = None
    cartesian_points = list()

    #
    axis2_placement_3d_direction_2_points = None
    axis2_placement_3d_direction_1_points = None
    axis2_placement_3d_cartesian_points = None

    # parameterized constructor
    def __init__(self, spot_beginning, lines_list):
        self.spot_beginning = spot_beginning
        self.lines_list = lines_list
```

```
        self.calculate()

    def calculate(self):

        self.calculate_open_shell()
        self.calculate_advanced_face()
        self.calculate_face_outer_bound()
        self.calculate_plane()
        self.calculate_axis2_placement_3d()
        self.calculate_edge_loop()
        self.calculate_oriented_edge()
        self.calculate_edge_curve()
        self.calculate_circle()
        self.calculate_axis_placement()
        self.calculate_cartesian_point()

    def calculate_open_shell(self):

        characters_to_find = "OPEN_SHELL"

        for line in self.lines_list:
            if characters_to_find in line and self.spot_beginning in line:
                self.open_shell_line = line
                self.open_shell_brackets_list =
STEPFileParser.find_hashed_numbers(line)
                break

    def calculate_advanced_face(self):

        characters_to_find = "ADVANCED_FACE"

        for line in self.lines_list:
            if characters_to_find in line:
                item = self.open_shell_brackets_list[0]
                new_item = str(item) + "="

                if characters_to_find in line and new_item in line:

                    self.advanced_face_line = line
                    # self.advanced_face_brackets_list =
STEPFileParser.find_hashed_numbers(line)

                    # For plane
                    import re
                    extracted_list = re.findall('\d+', self.advanced_face_line)
                    # Adding a prefix # to the found points

                    for index, item in enumerate(extracted_list):
                        item = "#" + item
                        extracted_list[index] = item

                    # remove via index
                    del extracted_list[0]
                    self.advanced_face_brackets_list = extracted_list
                    # For plane - End
                    break


    def calculate_plane(self):
        characters_to_find = "PLANE"

        for line in self.lines_list:
            if characters_to_find in line:
                item = self.advanced_face_brackets_list[1]
```

```
                new_item = str(item) + "="
                if characters_to_find in line and new_item in line:
                    self.plane_line = line
                    self.plane_brackets_list =
STEPFileParser.find_hashed_numbers(line)
                    break

    def calculate_axis2_placement_3d(self):
        characters_to_find = "AXIS2_PLACEMENT_3D"

        for line in self.lines_list:
            if characters_to_find in line:
                item = self.plane_brackets_list[0]
                new_item = str(item) + "="
                if characters_to_find in line and new_item in line:
                    self.axis2_placement_3d_line = line
                    self.axis2_placement_3d_brackets_list =
STEPFileParser.find_hashed_numbers(line)
                    break

        ###
        def cartesian_points(line):
            my_str = line

            find_beginning_bracket_index = my_str.index('(')
            find_ending_bracket_index = my_str.index(')')

            substring_with_hashed_numbers = ""

            # separate a string that begins with ( and ends with )
            for index, item in enumerate(my_str):
                if index > find_beginning_bracket_index and index <
find_ending_bracket_index:
                    substring_with_hashed_numbers += item

            final_list = list()

            # Only allowing number, dot, E to be in the list
            # , also for easy splitting
            for index, item in enumerate(substring_with_hashed_numbers):
                if item.isnumeric() or \
                        item == "." or \
                        item == "E" or \
                        item == "-" or \
                        item == ",":
                    final_list.append(item)

            # Joining to make a simple string for further processing
            makeitastring = ''.join(map(str, final_list))
            final_numbers_list = makeitastring.split(",")

            # Removing empty items in list
            while ("" in final_numbers_list):
                final_numbers_list.remove("")

            # Converting exponents to floats
            float_list = list()

            for index, item in enumerate(final_numbers_list):
                float_list.append(format(float(item), '.3f'))

            return float_list
        ###
```

```
        if self.axis2_placement_3d_brackets_list.__len__() > 0:

            # CARTESIAN_POINT

            characters_to_find_CARTESIAN_POINT = "CARTESIAN_POINT"
            axis2_placement_3d_CARTESIAN_POINT =
str(self.axis2_placement_3d_brackets_list[0]) + "="

            characters_to_find_DIRECTION = "DIRECTION"
            axis2_placement_3d_DIRECTION_1 =
str(self.axis2_placement_3d_brackets_list[1]) + "="
            axis2_placement_3d_DIRECTION_2 =
str(self.axis2_placement_3d_brackets_list[2]) + "="


            for line in self.lines_list:

                if characters_to_find_CARTESIAN_POINT in line and
axis2_placement_3d_CARTESIAN_POINT in line:

                    self.axis2_placement_3d_cartesian_points =
cartesian_points(line)

                if characters_to_find_DIRECTION in line and
axis2_placement_3d_DIRECTION_1 in line:

                    self.axis2_placement_3d_direction_1_points =
cartesian_points(line)

                if characters_to_find_DIRECTION in line and
axis2_placement_3d_DIRECTION_2 in line:

                    self.axis2_placement_3d_direction_2_points =
cartesian_points(line)



    def calculate_face_outer_bound(self):

        characters_to_find = "FACE_OUTER_BOUND"

        for line in self.lines_list:
            if characters_to_find in line:
                item = self.advanced_face_brackets_list[0]
                if characters_to_find in line and item in line:
                    self.face_out_bound_line = line
                    self.face_out_bound_brackets_list =
STEPFileParser.find_hashed_numbers(line)
                    break

    def calculate_edge_loop(self):

        characters_to_find = "EDGE_LOOP"

        for line in self.lines_list:
            if characters_to_find in line:
                item = self.face_out_bound_brackets_list[0]
                new_item = str(item) + "="
                if characters_to_find in line and new_item in line:
                    self.edge_loop_line = line
                    self.edge_loop_line_brackets_list =
STEPFileParser.find_hashed_numbers(line)
                    break
```

```
    def calculate_oriented_edge(self):

        characters_to_find = "ORIENTED_EDGE"

        for line in self.lines_list:
            if characters_to_find in line:
                item = self.edge_loop_line_brackets_list[0]
                new_item = str(item) + "="
                if characters_to_find in line and new_item in line:
                    self.oriented_edge_line = line
                    self.oriented_edge_line_brackets_list =
STEPFileParser.find_hashed_numbers(line)
                    break

    def calculate_edge_curve(self):

        characters_to_find = "EDGE_CURVE"

        for line in self.lines_list:
            if characters_to_find in line:
                item = self.oriented_edge_line_brackets_list[0]
                if characters_to_find in line and item in line:
                    self.edge_curve_line = line
                    self.edge_curve_brackets_list =
STEPFileParser.find_hashed_numbers(line)
                    break

    def calculate_circle(self):

        characters_to_find = "CIRCLE"

        for line in self.lines_list:
            if characters_to_find in line:
                for index, item in enumerate(self.edge_curve_brackets_list):
                    new_item = str(item) + "="
                    if characters_to_find in line and new_item in line:
                        self.circle_line = line
                        self.circle_brackets_list =
STEPFileParser.find_hashed_numbers(line)
                        self.__calculate_radius()
                        break

    def calculate_axis_placement(self):

        characters_to_find = "AXIS2_PLACEMENT_3D"

        for line in self.lines_list:
            if characters_to_find in line:
                for index, item in enumerate(self.circle_brackets_list):
                    new_item = str(item) + "="
                    if characters_to_find in line and new_item in line:
                        self.axis_placement_line = line
                        self.axis_placement_brackets_list =
STEPFileParser.find_hashed_numbers(line)
                        break

    def calculate_cartesian_point(self):

        characters_to_find = "CARTESIAN_POINT"

        for line in self.lines_list:
            if characters_to_find in line:
                for index, item in enumerate(self.axis_placement_brackets_list):
                    new_item = str(item) + "="
```

```
                    if characters_to_find in line and new_item in line:
                        self.cartesian_point_line = line
                        self.__convert_cartesian_points()
                        break

    def __convert_cartesian_points(self):
        """
        :return: cartesian points in proper list
        """

        if self.cartesian_point_line:

            my_str = self.cartesian_point_line

            find_beginning_bracket_index = my_str.index('(')
            find_ending_bracket_index = my_str.index(')')

            substring_with_hashed_numbers = ""

            final_list = list()

            # separate a string that begins with ( and ends with )
            for index, item in enumerate(my_str):
                if index > find_beginning_bracket_index and index <
find_ending_bracket_index:
                    substring_with_hashed_numbers += item

            final_list = list()

            # Only allowing number, dot, E to be in the list
            # , also for easy splitting
            for index, item in enumerate(substring_with_hashed_numbers):
                if item.isnumeric() or \
                        item == "." or \
                        item == "E" or \
                        item == "-" or \
                        item == ",":
                    final_list.append(item)

            # Joining to make a simple string for further processing
            makeitastring = ''.join(map(str, final_list))
            final_numbers_list = makeitastring.split(",")

            # Removing empty items in list
            while ("" in final_numbers_list):
                final_numbers_list.remove("")

            # Converting exponents to floats
            float_list = list()

            for index, item in enumerate(final_numbers_list):
                float_list.append(format(float(item), '.3f'))

            self.cartesian_points = float_list

    # def display_cartesian_points(self):
    #
    #     simple_string = ""
    #
    #     for index, item in enumerate(self.cartesian_points):
    #         if index == 0:
    #             simple_string = "x: " + str(item) + "\n "
    #         elif index == 1:
    #             simple_string += "y: " + str(item) + "\n "
```

```
#         elif index == 2:
#             simple_string += "z: " + str(item) + "\n"
#
#     return simple_string

def display_x(self):
    return "x: " + str(self.cartesian_points[0])

def display_y(self):
    return "y: " + str(self.cartesian_points[1])

def display_z(self):
    return "z: " + str(self.cartesian_points[2])

def display_face_outer_bound(self):
    return str(self.open_shell_brackets_list[0])

def display_advanced_face(self):
    return str(self.advanced_face_brackets_list[0])

def display_edge_loop(self):
    return str(self.face_out_bound_brackets_list[0])

def display_aaa(self):

    final_str = ""

    for item in self.axis2_placement_3d_cartesian_points:
        final_str += "    " + item

    return str(final_str)

def display_bbb(self):

    final_str = ""

    for item in self.axis2_placement_3d_direction_1_points:
        final_str += "    " + item

    return str(final_str)

def display_ccc(self):

    final_str = ""

    for item in self.axis2_placement_3d_direction_2_points:
        final_str += "    " + item

    return str(final_str)

def __calculate_radius(self):
    """
    :return: radius in proper list
    """

    if self.circle_line:

        my_str = self.circle_line

        find_beginning_bracket_index = my_str.index('(')
        find_ending_bracket_index = my_str.index(')')

        substring_with_hashed_numbers = ""
```

```
            # separate a string that begins with ( and ends with )
            for index, item in enumerate(my_str):
                if index > find_beginning_bracket_index and index <
find_ending_bracket_index:
                    substring_with_hashed_numbers += item

            # Removing empty items in list
            for index, item in
enumerate(substring_with_hashed_numbers.split(",")):
                try:
                    if not math.isnan(float(item)):
                        self.circle_radius = float(item) * 2
                        break
                except TypeError:
                    pass
                except ValueError:
                    pass
```

CERTIFICATE OF COMPLETENESS

It is hereby certified that the dissertation submitted by **NS Muhammad Ali kiani**, Reg No. **00000170904**, Titled: ***AUTOMATIC SPOT WELDING FEATURE RECOGNITION FROM STEP DATA*** has been checked/reviewed and its contents are complete in all respects.

Supervisor's Name: **Dr. Hasan Aftab Saeed** Signature: ______________________

Date: ______________________