

An Integrated Modular Approach For Features Extraction And Generation Of Optimal Process Plans For Prismatic Parts

B.V. Sudheer Kumar*, C.S.P. Rao**

**Research Scholar, Dept. of Mechanical Engineering, NIT Warangal, A.P, India-506021*

***Professor, Dept. of Mechanical Engineering, NIT Warangal, A.P, India-506021*

Abstract

In the present work, a novel approach has been developed to extract the features and to generate optimal process plans automatically, from the CAD solid models. This methodology utilises four modules to perform the said task. The first module contains three stages. In the first stage the list of features are extracted from the VRML file. These list of features are converted to sequence of individual operations with machining time and cost in the next stage. Further stage three of the module1 converts the sequence of operations into XML file format. This XML file provides the necessary input, to the second module, which generates all possible process plans. The optimal process plans will be identified with the help of third module. The module one is developed in visual C++ environment, where as P3 and Integrated Net Analyser (INA) are used as modules two and three, respectively. The developed methodology has been tested on various parts and successfully generated optimal process plans.

Key words— Feature Extraction, Optimal Process Plans, VRML.

1. Introduction

The term feature implies various meanings in different engineering disciplines. This has resulted in several ambiguous definitions for feature. A feature, in Computer Aided Design (CAD) software, can be called a region of a part with some interesting geometric or topological patterns. This meaning can refer to all sorts of information, such as for example, shape, functional or manufacturing information. Although many types of features have been investigated, the most common type of feature is the form feature, which contains both shape information and parametric information. Examples of form features common in many shape models are round holes, slots, steps, bosses and

pockets. Features can also be used to represent manufacturing information of the part. Different manufacturing domains require different feature representations. Some of the properties that need to be encoded by features are assembly method, manufacturing process and tolerances. A manufacturing feature can be defined as a form feature, but not necessarily vice versa. Among manufacturing features, the machining features have received extensive attention. A machining feature can be considered as, the volume swept by a cutting tool. In this sense, it is always a negative (subtracted) volume, in contrast with form features that are sometimes positive (added) volumes.

Feature data in a CAD model can be represented either as a collection of surfaces or volumes. Surface features are naturally used for example to describe manufacturing tolerances or locating surfaces in fixture design. Volumetric features on the other hand, are used in process planning since manufacturing information (particularly in machining) is better portrayed volumetrically.

Most of the parts can be interpreted in terms of machining features i.e., holes, steps, slots and pockets. Computer Aided Process Planning (CAPP) will use these features to generate manufacturing instructions to produce the part. Jung Hyun Han [1] made a survey on feature recognition and merits of several algorithms of feature recognition i.e. graph pattern matching, cell based decomposition, convex hull decomposition and Hint based reasoning etc. Mike Pratt and William C. Regli [2] gave an overview on the three major algorithmic approaches for feature recognition and mentioned several drawbacks of them. In the graph-based algorithms, the part is represented by a graph data structure, and is searched for particular patterns for features. The volumetric decomposition approach decomposes the volume of the part to be manufactured into a set of intermediate volumes and then manipulates the volumes to produce features. The hint-based

reasoning starts from a minimal indispensable portion of a feature's boundary which should be present in the part, and performs extensive geometric reasoning. Joshi and Chang [3] developed a graph named the Attribute Adjacency Graph (AAG) to represent features in which each face of the part is represented as a node, and each edge or face adjacency is represented as an arc. Sashikumar Venkataraman [4] presented a graph based frame work for feature recognition. The feature recognition step involved finding similar sub graphs in the part graph. The novelty of this framework lied in the usage of a rich set of attributes to recognize a wide range of features efficiently. W.F. Lu [5] gave an approach to recognize features from a data exchanged part model. A litany of algorithms for the identification of design and machining features are proposed. Emad S. Abouel Nasr [6] discussed a methodology for extracting manufacturing features from CAD system. The system takes a neutral file in Initial Graphics Exchange Specification (IGES) format as input and translates the information in the file to manufacturing information. The boundary (B-rep) geometrical information of the part is then analyzed by a feature recognition program that is created specifically to extract the features from the geometrical information based on a geometric reasoning approach.

The basic role of the CAD is to precisely define the geometry of the product, as it is critical to all the subsequent activities in the product life cycle. In most of the cases parts are designed in separate CAD environments, which have no direct link with manufacturing. Different CAD packages use different types of database structures to store the information of the part in a CAD file. In this paper CATIA V5R16 is used for modeling the part. However any other CAD soft ware, which has an option to convert CAD data, into Virtual Reality Modeling Language (VRML) file format, can also be used.

After developing the CAD model, the VRML file will be generated and used as input to the module1, which produces an output file (text file) with list of features and the sequence of operations. The output of module1 is the input for module2 and the output in the form of XML file will be obtained. The output of module2 will be used as input for module3 (software named P3) to get a Petri net diagram to indicate various process plans. The module4 (software named INA) has been used to identify the optimal process plans

VRML, sometimes pronounced vermal, is an

acronym for the Virtual Reality Modeling Language. Technically speaking, VRML is neither virtual reality nor a modeling language. Virtual reality typically implies an immersive 3D experience (such as a head-mounted display) and 3D input devices (such as digital gloves). VRML neither requires nor precludes immersion. Furthermore, a true modeling language would contain much richer geometric modeling primitives and mechanisms. VRML provides a bare minimum of geometric modeling features and contains numerous features far beyond the scope of a modeling language [17].

VRML can most easily be seen as a 3D interchange format that supports common features such as hierarchical transformations, light sources, geometry, animations, visual effects, material properties and texture mapping. VRML has been designed to be an analog to the commonly used HTML, in that it is a multi-platform language for publishing web content in a relatively simple and straight-forward way.[18]

2. Development of Automatic Process Plans Generation methodology.

In the present paper automatic process plan generation has been done with the help of four different modules. The following subsections explain the developed modules.

2.1 Module1: The part modelled using the CAD software forms the basis for the extraction of the features. In this paper CATIA V5R16 is used for modeling the component. The input required for module1 is generated in the form of VRML format. Software required to perform the said task has been developed using the Visual C++, which can extract the geometrical entities like surfaces, edges and vertices.

Then the automatic feature extraction algorithm extracts all the details of the prismatic features present. The details include type of feature, location of the feature and necessary dimensions related to the feature. The features considered in this paper are holes, steps, slots, islands and pockets. Every feature present in the part is associated with sequence of operations. The module1 consists of three stages. Stage1 extracts the list of features from the VRML file. Stage2 converts these lists of features into a sequence of operations with machining time and cost of individual operations. Then stage3 converts the sequence of operations into XML file which serves as

input to module2.

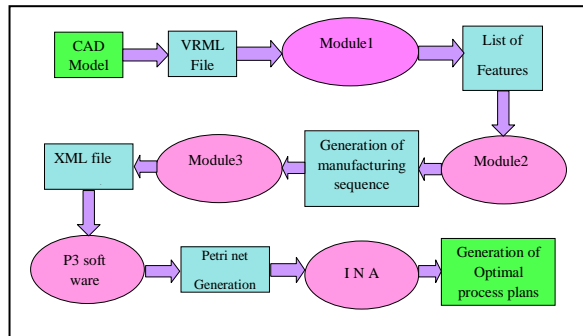


Fig 1. Flow chart showing the proposed modular approach.

From the VRML file, various features of the prismatic part are extracted by utilizing the following procedure.

Step1: All the lines (edges) and points (vertices) of the given part are identified, from the VRML file.

Step 2: All the lines, indicating the full edges of the rectangular block are removed from the set of all lines.

Step 3: Separate all lines into two sets, such that one set contains lines with two points and the other set having all the lines with more than two points. Then the two point line sets will be further sub divided into horizontal (the y-coordinate of both start and end points are same and x- and z- coordinates may differ) and vertical (the x- and z- coordinates are same with different y-coordinate) lines.

Step 4: Based on the y-coordinate value, segregate the horizontal lines, into number of subsets that are having same y-coordinate value.

Step 5: From the base plane, identify the nearest y-plane subset and identify the number of closed loops. This plane forms the base plane for the feature to be identified.

Step 6: Identify the nested and non-nested loops and number them.

Step 7: Consider the loop number one as non-nested and number the plane as plane1. This plane1 indicates the bottom/top plane of the feature. Identify all the vertices of loop1 in plane1. Identify the other plane where all the vertices with same x and z-coordinates with different y values exists and name it as plane2. This plane2 indicates the bottom/top plane of the feature.

Then attach the vertical lines at vertices of the considered loop in plane1. Then close the join lines with suitable line segments from plane2 to complete the faces (that is, vertical loop).

Step 8: Count the number of join lines above and below the loop.

Then the following six options, such as 0&4, 4&0, 1&3, 3&1, 2&2 and 2&2 arise in order to decide the feature. From the above, 0&4 indicates no lines above and four lines below the closed loop of plane 1, respectively. Moreover, the difference between 0&4 and 4&0 is that they imply feature on above and below the closed loop of plane 1, respectively. It is important to note that other options also follow the similar meaning.

Slot

No join lines below and two vertical faces with four join lines above the rectangular loop, indicates a slot on top face. Similarly, no join lines above and two vertical faces with four join lines below the rectangular loop, indicates a slot on bottom face.

Blind Slot

No join lines below and three vertical faces with four join lines above a rectangular loop, indicates a blind slot on top face. Similarly, three vertical faces with four join lines below and no join lines above the rectangular loop, indicates a blind slot on bottom face.

Blind Step

One join line below and three join lines with two vertical faces above a rectangular loop, indicates a blind step on top face. Similarly, one join line above and three join lines with two vertical faces below a rectangular loop, indicates a blind step on bottom face.

Step

Two join lines with one vertical face below and two join lines with one vertical face above a rectangular loop, indicates a full step. Areas of plane1 and plane 2 are compared to decide the step location. If area of plane2 is lesser then the step is on top face, else the step is on bottom face.

Step 9: Consider the loop number two as nested and number the plane as plane1. Identify all the vertices of loop1 in plane1. Identify the other plane where all the vertices with same x and z-coordinates with different y

values exits and name it as plane2. To the outer closed loop attach join lines (vertical lines) at vertices. Then close the join lines with suitable line segments from plane2 to complete the faces.

Pocket in Island

The outer loop which is nested, will have four join lines with four faces above it. The inner loop will have four join lines with four faces above it. These conditions indicate the presence of a pocket in island.

Island in Pocket

The outer loop without any join lines or faces is present on plane1. The inner loop will have four join lines with four faces above it. These conditions indicate the presence of an island in a pocket.

Repeat the procedure mentioned above for different y-planes identified in Step 4.

Then the following procedure is used in identification of cylindrical pockets.

Consider the line sets with more than two points obtained from step3.

Segregate the multi point line sets with same y values and sets with different y values. Consider the sets with same y value.

Find the values of A and B for the first and second points from the following equations.

$$X(i+1) = AX_i - BY_i$$

$$Y(i+1) = AY_i + BX_i$$

Find the values of A and B for the second and third points from the above equations.

Verify the value of A and B obtained in both cases. If they are same, the curve is considered as a part of the circle.

Then the circle diameter and centre coordinates will be calculated as follows.

From the same set of points Maximum x coordinate minus minimum x coordinate gives the diameter.

Centre x coordinate can be obtained as maximum x plus minimum x divided by two.

Centre z coordinate can be obtained as either z coordinate associated with maximum x or with minimum x.

If not a message "unidentified curve" will be displayed.

In stage 2 of module 1, the corresponding manufacturing operations will be mapped with the features identified. Cutter selection for every operation

will be done automatically with the help of knowledge base. Here, it is assumed that the cutters and machines are always available. The maximum diameter of the cutter is selected to minimize the machining time. The setup will be decided based on the feature location, that is, on which face the feature was present. For clamping purposes, magnetic base is selected. Finally, in stage 3 these sequences of operations have been converted to set of instructions in the form of XML file. These instructions can be given as input to the P3 software. The said task has been performed with the help of software developed using visual C++.

2.3 P3 software (Module 2)

The XML file generated by module 1 is given as input to module 2 which is a public domain software (P3 software). This P3 software generates Petri net diagram in the form of places, transitions and arcs, which indicates all possible process plans and can be used to physically verify the petri net diagram.

2.4 INA software (Module 3)

The manufacturing time, cost and process plans obtained from the P3 software will be used as input to the INA software. The optimal process plans in terms of manufacturing time and cost will be generated as output.

3. Results & Discussions

In this section, the optimal process plans for producing a prismatic part using modular approach has been developed. The functions performed by various modules are explained with the help of an example (refer fig. 2) in the following sections.

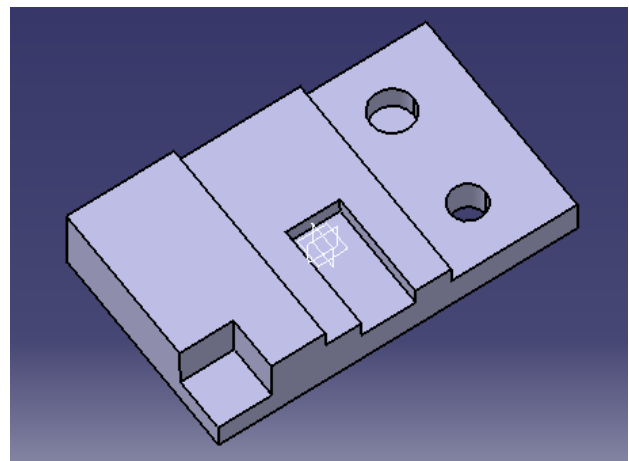


Fig. 2 Schematic diagram showing the example part.

3.1. Module 1

Input & Output of Feature Recognition

Module:

Due to the large size of input and output files, only small parts of the files are shown. A few fields of the VRML file generated are shown below.

```

geometry IndexedFaceSet {
  solid FALSE
  coord Coordinate {
    point [
      40 30 60,
      40 40 60,
      40 30 -60,
      40 40 -60,
    ] }
  coordIndex [
    0,1,2,-1,
    1,3,2,-1,
  ] }
    
```

Table1: Geometry Indexed Face Set field of VRML.

The output of feature extraction module for two features namely blind slot and through hole is shown below.

```

Feature Id : 3
TYPE OF FEATURE IS A BLIND SLOT
Feature Boundary :: (x=10.00, -20.00), (y=20.00,
30.00), (z=-60.00, 0.00)
Lines of Feature on y=20.00 plane are 4
Ln 50:: 10.00 : 20.00 : 0.00 --> 10.00 : 20.00 : -
60.00
Ln 45:: 10.00 : 20.00 : -60.00 --> -20.00 : 20.00 : -
60.00
Ln 41:: -20.00 : 20.00 : 0.00 --> 10.00 : 20.00 :
0.00
Ln 49:: -20.00 : 20.00 : -60.00 --> -20.00 : 20.00
: 0.00
Join Lines of Feature are 4
Ln 48:: -20.00 : 20.00 : 0.00 --> -20.00 : 30.00 :
0.00
Ln 47:: -20.00 : 20.00 : -60.00 --> -20.00 :
    
```

Table2: Result for blind slot

Feature Id : 1
Cx : 60.00, Cz : -30.00 , Diameter : 20.00
Max y : 20.00, Min y : 0.00

Table3: Result for through hole

Input & Output of Manufacturing

Operations Module:

The input for this module is the output of the previous module i.e., F E sub module.

The manufacturing operations (including alternatives) will be mapped with the features with the support of machining data knowledge. Cutter selection for every operation will be done automatically with the help of knowledge base. Here it is assumed that the cutter is always available and the suitable cutter with maximum diameter is selected to minimize the machining time. The setup will be decided based on the feature location i.e., on which face the feature was present. For clamping purposes a magnetic base was selected. The raw material is assumed as a rectangular block. Hence six facing operations are assumed on all six faces to obtain a dimensionally correct rectangular block.

A small scale production unit is considered where three machines are present in the shop floor as follows.

Machine Code	Type of Machine Tool	Cost of machining (Rs. Per minute)

M1	Vertical milling machine	20
M2	Jig boring machine – vertical	25
M3	Radial drilling machine	15

Table1: The machining cost on various machine tools

The list of cutters used on the four machine tools is as follows.

S.No	Cutter description	Cutter code
1	Flat end mill, 15mm diameter, 2 inserts	C1
2	Slot drill, 20mm diameter	C2
3	Slot drill, 24mm diameter	C3
4	Face mill, 50mm diameter and 4 inserts	C4
5	Drill bit, 20mm diameter	C5
6	Drill bit, 24mm diameter	C6

Table2: list of available cutters of all machine tools

Assumptions:

- 1 The machine tool change cost from any machine tool to any machine tool is same and is equal to Rs. 300 per change.
- 2 The cutter change cost on any machine tool is same and is equal to Rs. 200 per change.
- 3 The setup change on any machine tool is same and is equal to Rs.100 per change.

The table showing the full details of the operations , necessary machine tools , setups and cutters along with machining cost is shown in tabular form in Appendix-A

XML Based Interfacing Module: the manufacturing operations sequence generated in the above sub module will be converted to XML file format containing places and transitions connected with arcs, which can be viewed in the P3 software. The transitions indicate operations and places indicate machine tool availability, cutter availability and completion of work holding in the necessary setup etc.

```
<?xml version="1.0" encoding="UTF-8" ?>
pnml
xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="upn
pnml.xsd">
net id="n1" type="UPNPML">
place id="p1">
graphics>
name>
value>START</value>
</name>
initialMarking>
```

Table3: A part of generated XML file

Process sequence optimization module : In this module two tools are used named P3 and INA. The XML file generated in the previous module can be viewed by using the P3 software. The Petri net model developed for the example part is shown in the figure below

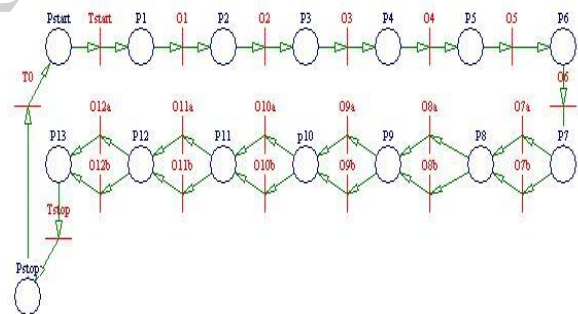


Figure3: Petri net model for the example part

The necessary manufacturing operations to produce the given part which are in the form of Petri net is the input for the INA. The input should be given in the prescribed format suitable for INA. The transition times i.e., operation times and transition values i.e., cost of operations should be entered to obtain the necessary optimal sequence of operations. For the part shown above the minimum machining time and minimum manufacturing cost is obtained for the same sequence of operations.

Optimal Sequence Result:

Circuit nr. 64:

n1<-s21:0--n15<-s20:0--n14<-s19:45--n13<-
s17:30--n12<-s15:450--n11<-s13:1260--n10<-
s11:2520--n9<-s9:564--n8<-s7:180--n7<-s6:180--
n6<-s5:108--n5<-s4:108--n4

<-s3:300--n3<-s2:300--n2<-s1:0--n1

Corresponding step-invariant:

1 1 1 1 1 1 1 0 1 0 1 0 1 0 1 0 1 1 1 1

Livelocked transitions:

64: 9, 11, 13, 15, 17, 19,

Cycle time = 6045 value = 2010 efficiency = 0,3.

Circuit nr. 64 has minimal cycle time = 6045.

Circuit nr. 64 has minimal value = 2010.

Optimal Sequence is o1-o2-o3-o4-o5-o6-o7a-
o8b-o9b-o10a-o11b-o12a

7. CONCLUSIONS:

- [1]An attempt is made successfully to integrate CAD and CAPP for prismatic parts and tested practically for variety of prismatic parts.
- [2]The work comprise feature extraction module, manufacturing operations module, XML based interfacing module and process sequence optimization module.
- [3]Feature extraction module contains several algorithms being developed for feature extraction from CAD models.
VRML file of any CAD model contain geometric information of the part being modeled, which is exploited in the present work. An algorithm is developed as a part of feature extraction module for parsing the data and provides the full details of the geometry like points (vertices), lines

(edges), closed loops (faces) including the connectivity, which forms the basis for feature extraction algorithms.

- [4]A Prismatic Feature identification algorithm is developed and implemented for identifying different milling features like slot, blind slot, step, blind step, pocket and island in pocket etc.

Cylindrical Feature Identification algorithm is also developed and implemented, which has the ability to identify cylindrical features such as through hole, blind hole and stepped hole.

- [5]Manufacturing operations module will map individual manufacturing operations with corresponding features and identify the required machine tool and corresponding cutter from the knowledge base which contains the information regarding machine tools, cutters, machinability data and machining cost. The system generates a sequence of operations including alternatives. Finally it generates an XML file compatible for modeling the manufacturing sequence problem in the form of a Petri net using P3 soft ware. Thus a Petri net model of a manufacturing sequence problem is developed automatically.

- [6] The author has used a public domain soft ware tool named INA (Integrated Net Analyzer) which reads the Petri net and execute it to generate all possible process plans with the individual sequence machining times and manufacturing costs. At the end of the output file along with the data of all possible sequences, the sequence with minimum machining time, maximum machining time, minimum manufacturing cost and maximum manufacturing cost will also be calculated. Thus the final objective of the present work is achieved.

A number of case studies collected from the industry and literature were studied successfully with ease. Thus the attempt to

develop seamless integration of CAD and CAPP to generate optimal process plans for prismatic parts is successful.

References

- [1] Nafis Ahmad, and A.F.M. Anwarul Haque., "Manufacturing feature recognition of parts Using DXF file", 4th International Conference on Mechanical Engineering, December 2001, Dhaka, Bangladesh. pp. 111-115.
- [2] T. Dereli, and H. I. Filiz, "Optimization of Process Planning functions by Genetic algorithm", Computers and Industrial Engineering, 1999, Vol. 36, pp. 281-308.
- [3] Mike Pratt and William C. Regli, "Manufacturing Feature Recognition from Solid Models: A Status Report", IEEE transactions on robotics and automation, 2000, vol. 16, no. 6, pp. 782-796.
- [4] Jung Hyun Han, "Survey on Feature Research, Institute for Robotics and Intelligent systems", USC, USA, 1996. (Technical report IRIS-96-346)
- [5] S. Joshi, and T. C. Chang, "Graph Based Heuristics for Recognition of Machined Features from 3D Solid Model", Computer Aided Design, 1988, Vol. 20, pp. 58-66.
- [6] Sashikumar Venkataraman, "A Graph based Framework for Feature recognition", ACM 2001 1-58113-366-9/01/06, 2001.
- [7] W.F. Lu, "An approach to identify design and manufacturing features from data exchanged part model", Computer-Aided Design, 2003, Vol. 35, pp. 979-993.
- [8] Emad S. Abouel Nasr, "A new methodology for extracting manufacturing features from CAD system", Computers & Industrial Engineering, 2006, Vol. 51, pp. 389-415.
- [9] Murata Tadao, "Petri Nets: Properties, Analysis and applications". IEEE Transactions on Electrical Engineering and Computer Science, 1989.
- [10] K. Srihari, and C. R. Emerson, "Petri nets in dynamic process planning", Computers & Industrial Engineering, 1990, Vol. 19, pp. 447-451.
- [11] J. A. Cecil, C. R. Emerson, K. Srihari, "A review of Petri net applications in process planning", International Journal of Advanced Manufacturing Technology, 1992, Vol. 7, pp. 168-177.
- [12] P. Xirouchakis and D. Kiritsis, "A generic Petri net model for disassembly planning processes", presented at TMCE'96, Budapest, 1996
- [13] D. Kiritsis and M. Porchet, "A generic Petri net model for dynamic process planning and sequence optimization". Adv. Engineering Software, 1996, Vol. 25, No. 1, pp. 61-71.
- [14] D. Kiritsis, K. P. Neuendorf, and P. Xirouchakis, "Petri net techniques for process planning cost estimation", Advances in Engineering Software, 1999, Vol. 30, pp. 375-387.
- [15] J. M. Usher, and R. O. Bowden, "The application of genetic algorithms to operation sequencing for use in computer-aided process planning", Computers & Industrial Engineering, 1996, Vol. 30, No.4, pp. 999-1013.
- [16] H.B.Marri, A. Gunasekaran, and R. J. Grieve, "Computer aided process planning: A state of art survey", International journal of Manufacturing Technology, 1998, Vol. 14, pp. 261-268.
- [17] Ramesh Naidu Ande, "Integrated VRML, JAVA, XML and HTML in a web based tool", SCSC, 2004.
- [18] Don Brutzman, "The Virtual Reality Modeling Language and Java, Communications of the ACM", 1998, vol. 41 no. 6, pp. 57-64.
- [19] Kian-Huat Tan, Tze Leong Yew, and Kurt Gramoll, "Understanding Machine Operations and Manufacturing using VRML", Am. Soc. of Engineering Educ. (ASEE) 1999 Conf., June 20 - 23, Charlotte, NC, 1999.
- [20] Anil Sawhney, Andre Mund, and Jennifer Marble, "Simulation of the Structural Steel Erection Process", Proceedings of the 31st Winter Simulation Conference, 1999.
- [21] WD Version 0.9.0, "Software and Systems Engineering - High-level Petri Nets Part 2: Transfer Format", International Standard ISO/IEC 15909-2 June 23, 2005.

APPENDIX A: List of operations along with set up, machine tools and cutters used

S. No	Name & Description of Manufacturing Operations	Machine tool used	Type of Cutter	Setup used	Machining time in Sec	Machining cost in Rs.
1	O1[Face milling-bottom face]	M1	C4	S2	300	100
2	O2[Face milling-top face]	M1	C4	S1	300	100
3	O3[Face milling-side face1]	M1	C4	S3	108	36
4	O4[Face milling-side face2]	M1	C4	S4	108	36
5	O5[Face milling-side face3]	M1	C4	S5	180	60
6	O6[Face milling-side face4]	M1	C4	S6	180	60
7	O7a[Milling Blind step]	M1	C1	S1	564	188
	O7b[Blind step on Jig-Boring]	M2	C1	S1	600	250
8	O8a [on Jig-Boring M/c]	M2	C4	S1	2580	1075
	O8b[Milling through step1]	M1	C4	S1	2520	840
9	O9a[on Jig-Boring M/c]	M2	C4	S1	1320	550
	O9b[Milling through step2]	M1	C4	S1	1260	420
10	O10a[Milling Blind slot]	M1	C1	S1	450	150
	O10b[on Jig-Boring M/c]	M2	C1	S1	480	200
11	O11a[on Milling M/c]	M1	C3	S1	36	12
	O11b[Drilling Blind hole 24Dia]	M3	C6	S1	30	8
12	O12a[Drilling Through hole 20Dia]	M3	C5	S1	45	12
	O12b[on Milling M/c]	M1	C2	S1	54	18