Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 133 (2018) 823–830

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Robotics and Smart Manufacturing (RoSMa2018)

ACAM: A CNC Simulation Software for Effective Learning

Praveen R. Nair, Harshit Khokhawat, Rajeevlochana G. Chittawadigi*

Department of Mechanical Engineering, Amrita School of Engineering, Bengaluru, Amrita Vishwa Vidyapeetham, India

Abstract

With an ever increasing need for products with better surface finish and exceptional dimensional accuracies, CNC machines have become an integral part of all manufacturing processes. Hence CNC programming has been an integral part of engineering education. In this scenario it only seems reasonable, there be a basic visualization tool which aids learning process of interested candidates and seamlessly integrates into an environment they are already familiar with, so that clarity in concepts can be attained without unwanted jargon. To provide effective learning of CNC, the authors have developed ACAM (Amrita Computer Aided Manufacturing), a software which smoothly integrates into the very familiar environment of Autodesk Inventor in the form of an addin. In the current form, ACAM can perform simulation for drilling, milling and turning operations. It is proposed to be used as a tool to help teachers explain different manufacturing steps and later on allow students to write CNC codes and visualize the resulting motion of the tools and/or workpiece.

Peer-review under responsibility of the scientific committee of the International Conference on Robotics and Smart Materials.

Keywords: CNC Simulation; Computer Numerical Controlled; Virtual Manufacturing;

1. Introduction

Closer control over dimensional tolerances and surface finish during manufacturing promises an increased efficiency in performance and better durability of the product. Hence, CNC (Computer Numerical Controlled) machines are used extensively in the manufacturing industries. In addition, the reliability of CNC machines is much better than that of conventional manufacturing tools. These machines are operated by writing CNC program or code. Due to the demand for these machines and advancement in technologies, several courses are included in Mechanical Engineering curriculum in graduate and post-graduate levels which deal with CNC programming.

* Corresponding author. Tel.: +91-80-25183700; fax: +91-80-28440092.
E-mail address: rg_chittawadigi@blr.amrita.edu

Peer-review under responsibility of the scientific committee of the International Conference on Robotics and Smart Materials.
10.1016/j.procs.2018.07.113

CNC machines are usually expensive and not all institutes would possess them for the students to learn CNC programming. The few institutes which have CNC machines may not be in a position to accommodate the operations being performed by students while learning. Hence, there is a genuine need of a simulation software that can be used to learn the CNC programming and try out the program developed by the students. Also, in a classroom environment, a teacher can demonstrate motion of tool or workpiece for different possible scenarios.

Several CNC simulation software exist that can be accessed online by using internet connection as reported in [1-3]. These require constant internet connection between the client computer and a server located remotely. Though many methods like using an applet on the client side to create and store workpiece [2] have been implemented, the quality and reliability of output still depends upon the internet connectivity available in the area.

Similar solutions are also available as desktop applications that run without any internet connectivity. Real-time simulation of 3 axis milling using isometric projection is reported in [4], where it uses z-map structure to show the generated 3D model of the workpiece. There are many open-source CAM simulation software such as CAMotics [5] which have a dedicated 3D rendering environment. The authors feel that developing a CAM simulation software from scratch using OpenGL or Direct3D technologies require deeper understanding of the graphics implementation and hence more development efforts. There are instances where a CAD software has been customized to develop an addin or plugin for simulation of CNC programs. CATIA, a well known CAD software has been used and reported in [6]. CATIA is an expensive software and not all institutes, at least in India, have access to it. On the other hand, Autodesk Inventor, a full fledged 3D modeling CAD software is available for free for "Academic Usage" and is also made available through AICTE. Most of the engineering institutes in India already use Autodesk Inventor and naturally the students are already familiar with the software. Hence, an addin or plugin developed for Autodesk Inventor using its Application Programming Interface (API) for CNC or CAM simulation is a win-win proposal for both developers and the end-users.

In addition to the CAM simulation software listed above, many commercial and free software are also available. Some of these include addins/plugins developed for Autodesk Inventor such as FeatureCAM, Inventor HSM, PowerMill, etc. These are available for a fee. On the other hand, learning a standalone application will always have a steeper learning curve and require more efforts than using an already familiar software. In this paper, ACAM (Amrita Computer Aided Manufacturing), a CNC simulation software developed as an addin inside Autodesk Inventor software is presented. An overview of parametric modeling to create features in Autodesk Inventor is explained in Section 2, followed by how those features can be programmatically modeled using its API in Section 3. Section 4 briefly describes the interface of the proposed software for simulation of drilling, milling and turning operations, followed by future scope and conclusions.

2. Parametric Modeling in Autodesk Inventor

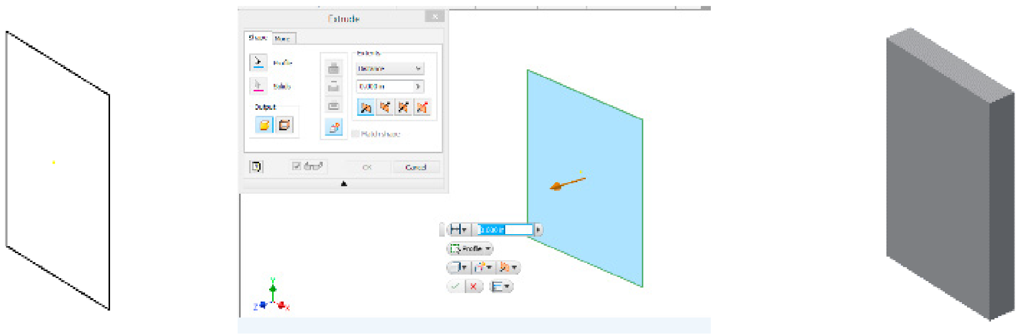
"Parametric" refers to information regarding the geometry of the design such as dimensions which can be subjected to variation at any instant in the design process. Parametric modeling is accomplished by identifying and creating the key features of the design with the aid of a computer software. The design variables, described in the sketches and described as parametric relations, can then be used to modify the design. Parameters are not only used in the sketches but also in all the features such as Extrude, Revolve, Sweep, etc., which exist in a feature based CAD software such as Autodesk Inventor, SolidWorks, CATIA, Pro-E, etc. Hence, understanding how parameters are related to a particular shape is a key to make designs that can be modified based on any particular need or criteria. Another category of CAD software use direct modeling technology where the current form of the shape of the solid is stored in the file/memory and any modifications of the shape updates the information. Here, the history of operations performed to reach the current form is not saved. Examples of direct modeling CAD software are SpaceClaim, PTC Creo, Autodesk Inventor Fusion, etc. For a CAM simulation software, the former category of feature based CAD software is preferred as it will be easy to modify any feature programmatically. For the reasons listed down in the previous section, Autodesk Inventor was chosen as the CAD software for ACAM.

In this section, a few of the standard features in Autodesk Inventor are explained, which are used in ACAM. These features require a set of 2-dimensional (2D) lines or curves in a sketch, referred to as a profile. Profile for these features can be of any shape but have to be closed, i.e., they have to be water-tight. The parameters used in the profile can be accessed and edited manually or programmatically using the API. When these parameters are modified and updated, the sketch and the corresponding features created using the profile are also updated.

2.1. Extrude

Extrude is one of the most commonly used feature while modeling a 3D object. It involves the following steps:

- Create a 2D sketch in any plane. The profile created in the sketch should be closed. An example of a rectangular profile is shown in Fig. 1(a).
- Extrude dialog is shown in Fig. 1(b), which lets the user select the profile and input the length of extrusion.
- Extruded part is shown in Fig. 1(c).



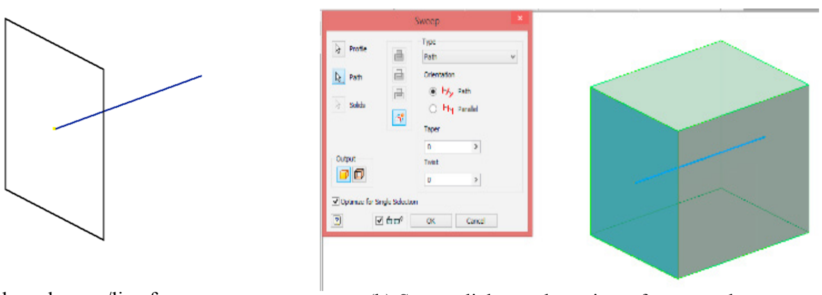
(a) Rectangular profile (b) Extrude dialog and selection of profile (c) Extruded part

Fig. 1. Extrude feature in Autodesk Inventor

2.2. Sweep

Sweep feature is similar to Extrude feature. But the profile can be swept across a curve or a set of curves. Sweep along a straight line results in Extrude feature. Sweep feature requires the below steps to be followed:

- Create a profile and a curve (a set of curves) for the sweep to be carried out on, as illustrated in Fig. 2(a).
- Sweep dialog is shown in Fig. 2(b), which lets the user select the profile and sweep curve and the preview is also shown. When the dialog is executed, the swept solid is created.



(a) Rectangular profile and curve/line for sweep (b) Sweep dialog and preview of swept volume

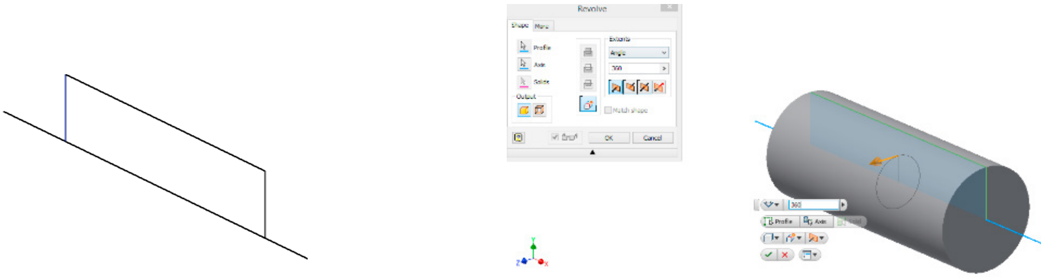
Fig. 2. Sweep feature in Autodesk Inventor

2.3. Revolve

Revolve is applicable for parts that have an axis of symmetry. Examples of such shapes are cylinder, cone, sphere, etc. The steps are:

- Create a profile and a line to be used as axis of revolution, as shown in Fig. 3(a).

- Revolve dialog lets the user select profile, axis of revolution and angle. If angle is set as 360°, a full revolution is performed. An example revolution of rectangular profile resulting into a cylinder is shown in Fig. 3(b).

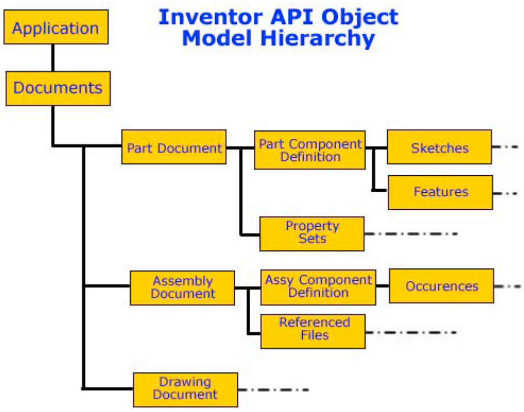


(a) Rectangular profile and revolution axis (b) Revolve dialog and preview or revolution

Fig. 3. Revolve feature in Autodesk Inventor

3. Customization in Autodesk Inventor

Almost all standard CAD software expose their API (Application Programming Interface) to extend their core functionality so that end-users can develop plugins or addins to increase their productivity. These extension are generally known as customization. Autodesk Inventor exposes its API in the form of a COM (Component Object Model). A high level representation of Autodesk Inventor Object Model is shown in Fig. 4(a), in which Application is the entry point to the CAD data. Representative code in Visual C# programming language to connect an addin to Inventor is given in Fig. 4(b), represented by the variable "invApp". The 3D CAD data can be accessed through invApp object and modified from external applications. Once the modifications are made, the 3D CAD data is updated in Inventor. In this paper, the Application object is used as the gateway to make all modifications in the inventor parts and also to move the parts inside the 3D CAD environment. A good overview on Inventor customization for beginners is given in [7]. Representative code to read and modify data required for the simulation of CAM operations are also provided so that the readers can use it as a reference to develop similar simulation software. Few researchers have exploited the advantages of Autodesk Inventor API to develop simulation and visualization software. An example of such usage for simulation and motion planning of industrial robots is reported in [8].



(a) Object model hierarchy [7]

```
Inventor.Application invApp =
(Inventor.Application)System.Runtime.InteropServices.Marshal.GetActiveObject("Inventor.Application");
```

(b) C# code to connect to Inventor

Fig. 4. CAD customization using Autodesk Inventor API

In the proposed ACAM software, the features which are explained from an end-user point of view in Section 2 have been implemented as a program. The explanation of those features is given in this section.

3.1. Extrude program

The end-user steps mentioned in Section 2.1 can be executed through a C# program. Representative code is given in Fig. 5(a). "partDoc" is the variable which refers to a new Inventor part file created. The "partCompDef" variable refers to the Component Definition of the newly created part, using which the program can add new sketch with a rectangular profile and also create Extrude feature, based on the value of the "distance" variable.

Since "profile" and "distance" variables are being used in the "extrudeFeature", any modifications to these variables is reflected in the 3D model of the extrusion.

```
PartDocument partDoc = (PartDocument)invApp.Documents.Add(DocumentTypeEnum.kPartDocumentObject,
invApp.FileManager.GetTemplateFile(DocumentTypeEnum.kPartDocumentObject));
PartComponentDefinition partCompDef = partDoc.ComponentDefinition;
PlanarSketch sketch = partCompDef.Sketches.Add(partCompDef.WorkPlanes[3]);
TransientGeometry transGeo = invApp.TransientGeometry;
SketchEntitiesEnumerator rectangleLines =
sketch.SketchLines.AddAsTwoPointRectangle(transGeo.CreatePoint2d(0,0), transGeo.CreatePoint2d(4,3));
Profile profile = sketch.Profiles.AddForSolid();
ExtrudeDefinition extrudeDef = partCompDef.Features.ExtrudeFeatures.CreateExtrudeDefinition(profile,
PartFeatureOperationEnum.kJoinOperation);
extrudeDef.SetDistanceExtent(distance, PartFeatureExtentDirectionEnum.kNegativeExtentDirection);
ExtrudeFeature extrudeFeature = partCompDef.Features.ExtrudeFeatures.Add(extrudeDef);
```

(a) C# code for Extrude feature

```
WorkPoint[] workPoints = new WorkPoint[2];
            workPoints[1] = partCompDef.WorkPoints.AddFixed(transGeo.CreatePoint(0, 0, 0));
            workPoints[2] = partCompDef.WorkPoints.AddFixed(transGeo.CreatePoint(3, 0, 0));
Sketch3D sketch3D = partDoc.ComponentDefinition.Sketches3D.Add();
SketchLine3D line = sketch3D.SketchLines3D.AddByTwoPoints(workPoints[1], workPoints[2], true, 1);
WorkPlane workPlane = partCompDef.WorkPlanes.AddByNormalToCurve(line, line.StartSketchPoint);
Path path = partCompDef.Features.CreatePath(line);
PlanarSketch sketch = partCompDef.Sketches.Add(partCompDef.WorkPlanes[2]);
sketch.SketchLines.AddAsTwoPointRectangle(transGeo.CreatePoint2d(0, 0), transGeo.CreatePoint2d(4,3));
SweepFeature sweep = partCompDef.Features.SweepFeatures.AddUsingPath(profile, path,
PartFeatureOperationEnum.kJoinOperation);
sweep.Path = path;
```

(b) C# code for Sweep feature

```
RevolveFeature revolve = partCompDef.Features.RevolveFeatures.AddByAngle(profile,
partCompDef.WorkAxes[3], 2*Math.PI,PartFeatureExtentDirectionEnum.kNegativeExtentDirection,
PartFeatureOperationEnum.kJoinOperation);
```

(c) C# code for Revolve feature

Fig. 5. Features created programmatically using Autodesk Inventor API

3.2. Sweep program

The end-user steps mentioned in Section 2.2 can be executed to obtain a Sweep feature. Stripped down code, which uses some parts from Fig. 5(a), is shown in Fig. 5(b). An array of WorkPoints is created to be used in a Sketch3D, which in turn is used as the "path" along which the "profile" has to be swept. By modifying the "path" variable, the swept volume can be updated programmatically.

3.3. Revolve program

Revolve feature mentioned in Section 2.3 can be programmatically executed similar to Extrude and Sweep features. The corresponding C# code, which uses an axis of revolution is given in Fig. 5(c). The "profile" of the revolve feature can be modified through the API and the model gets updated.

3.4. Movement of component

An assembly file in Autodesk Inventor can consist of one or multiple parts or sub-assemblies, referred to as components. Each of the component has a Coordinate system, known as Part Origin, with respect to which its shape is defined. The assembly file also has a Coordinate system, known as Assembly Origin. The position and orientation of the Part Origin of a component with respect to the Assembly Origin is defined using a 4x4 Homogenous Transformation Matrix (HTM).

To move a component in an assembly, the HTM of the Part Origin can be modified programmatically. To achieve a smooth animation of the intended motion, an interpolated motion can be given to the component. Due to the space constraints, representative code for modifying the Placement of a component is not explained.

4. ACAM Software

ACAM (Amrita Computer Aided Manufacturing) has been developed as a CAM simulation software, primarily aimed at educating students on how CNC programs work. It is developed using Visual C# programming language in Visual Studio IDE (Integrated Development Environment). In the current form, it has interface for end-user to do the following operations individually.

- Drilling
- Milling
- Turning

These operations are the building blocks for G code simulation. Hence, understanding how these primitive operations can be programmatically created and modified to show the animation was the main challenge that has been the focus in this paper. Due to space constraints, detailed implementation details have not been included.

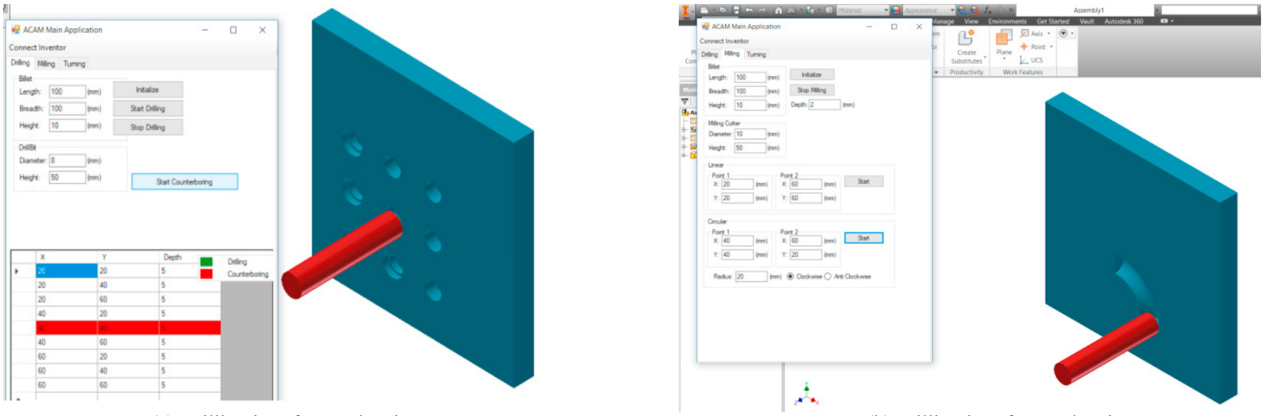
4.1. Drilling

Drilling operations correspond to the movement of a drillbit tool in an assembly and removal of material from a rectangular billet workpiece. The interface is shown in Fig. 6(a) which lets user connect to Autodesk Inventor application. Once connected, the billet and drillbit are created programmatically using the corresponding input data as Extrude features, explained in Section 3.1. A table of X and Y coordinates for the hole center is provided for the user to enter. These coordinates are used to plan the motion of the tool (drillbit) as a linear interpolation of the position between two consecutive points. When the tool moves to a hole center, motion to imitate the drilling operations is given to the tool and the material in billet is removed to the extent the tool has moved, again as Extrude feature in billet with a constant circular profile but by varying the distance of extrusion. If multiple holes of different diameters are provided for same X and Y coordinates, counter boring operations can be performed.

4.2. Milling

Milling operations involves movement of a rotating milling cutter tool on the surface of a rectangular billet to produce grooves on the surface. The movement could be along straight lines or radial arcs. The interface for milling in ACAM is shown in Fig. 6(b). Similar to Drilling, it has inputs for milling cutter diameter and the dimensions of the billet. Thereafter, the user can give linear motion or circular motion (both clockwise and anti-clockwise).

If the linear motion is selected, a straight line is to be used whose start point is fixed but the end point can be varied to get linear interpolation. This line segment is used to sweep a circular profile drawn in a plane perpendicular to the line segment and passing through the start point. Hence, as the tool moves, the material is removed on the billet surface as a Sweep feature through the API. Similarly for the circular arcs, interpolation from the start point along the curve is implemented to achieve both clockwise and anti-clockwise motion of the tool.



(a) Drilling interface and action (b) Milling interface and action

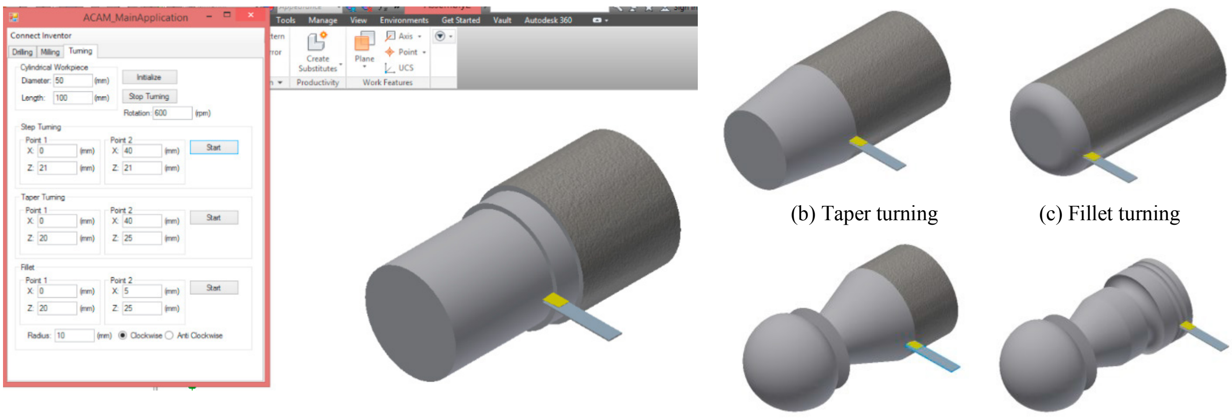
Fig. 6. Drilling and milling operations in ACAM

4.3. Turning

A cylindrical workpiece is generally fixed to a chuck of a CNC lathe machine which rotates at a high speed and a cutting tool is moved along the surface of the cylindrical workpiece to remove material from it. ACAM has a dedicated interface for turning which lets user enter the dimensions of the workpiece. Revolve feature discussed in Section 3.3 is used to model a cylinder. The material removal due to turning can be broadly classified as

- Step Turning:
  The tool is moved parallel to the axis of the workpiece. To simulate step turning, a rectangular profile is created from the start position and a revolve cut is performed programmatically to remove material corresponding to the initial rectangle. The width of the rectangle is modified in a linear interpolation till it finally reaches the end position. The Turning dialog and the 3D model of the workpiece after a step turning process is shown in Fig. 7(a). To demonstrate the motion of the tool, the position of the tool object is modified accordingly.
- Taper Turning:
  It is similar to step turning. Here, the tool moves at an angle with respect to the cylindrical axis, thus creating a taper on the cylindrical surface. Instead of a rectangular profile, a trapezoid profile is used in taper turning from the start position towards the end position. However, when the tool reaches the end position, the trapezoid profile degenerates to a triangle, as shown in Fig. 7(b).
- Fillet:
  Fillet refers to rounding off a circular edge on the workpiece to smoothen it. The motion of the tool to result in fillet requires interpolation along a circular arc centered at a point. The coordinates of the center of the arc are determined depending on the start and end positions and radius of curvature. It also depends whether desired circular motion is clockwise or anticlockwise. A profile with two orthogonal lines and a curve connecting them is used for revolve cut, programmatically. An example fillet is illustrated in Fig. 7(c).

A combination of step, taper and fillet turning to produce a complicated shape are shown in Fig. 7(d, e). Note that the turning operations primarily uses Revolve feature where the profile of revolution is changed through the API.



(b) Taper turning (c) Fillet turning

(a) Turning interface and 'Step turning' (d) Sequence of operations (e) Example of Chess pawn

Fig. 7. Turning operations in ACAM

5. Conclusions

ACAM is presented as an effective educational tool available for teaching and learning. It helps in understanding the basic concepts of Computer-Aided Manufacturing (CAM) without the hassle of buying another software or learning the interface of another software. ACAM has been developed as an addin or plugin inside Autodesk Inventor, a CAD software already familiar in the engineering institutes in India. In its current form, simulation for drilling, milling and turning operations are implemented where tool moves in the software and removes material based on the user input. In future, more advanced features like user input to CNC code and simulation of CNC code will be included. The user will be able to perform drilling, milling and turning as explained in Section 4 using the interface which has text boxes and buttons. A functionality to output the resultant CNC code will be developed. This will help in understanding CNC code from the sample data that is already known to the user. It can be useful in the initial stages of learning CNC programming.

References

[1] Ong, S. K., Jaing, L., and Nee, A. Y. C. (2002) "An internet-based virtual CNC milling system." *The International Journal of Advanced Manufacturing Technology*, **20 (1)**: 20-30.

[2] Seo Yoonho, Dae-Young Kim, and Suk-Hwan Suh. (2006) "Development of web-based CAM system." *The International Journal of Advanced Manufacturing Technology*, **28 (1-2)**: 101-108.

[3] Topcu Okan, and Ersan Aslan. (2011) "Web-based Simulation of a Lathe using Java 3D API." *Proceedings of 2nd International Symposium on Computing in Science & Engineering*.

[4] Hsu, P.L., and Yang, W.T., (1993) "Realtime 3D simulation of 3-axis milling using isometric projection." *Computer-Aided Design*, **25 (4)**, 215-224.

[5] CAMotics, http://www.camotics.org, Accessed in April 2018.

[6] Dubovska Rozmarina, Jaroslav Jambor, and Jozef Majerik. (2014) "Implementation of CAD/CAM system CATIA V5 in Simulation of CNC Machining Process." *Procedia Engineering* **69**: 638-645.

[7] Rajeevlochana, C. G. (2008) "Introduction to Autodesk Inventor API and Customization for Dummies." *Technical Document, AUGI India*.

[8] Neto Pedro and Nuno Mendes. (2013) "Direct off-line robot programming via a common CAD package." *Robotics and Autonomous Systems* **61 (8)**: 896-910.