journal homepage: www.elsevier.com/locate/jmatprotec

A solid model-based off-line adaptive controller for feed rate scheduling for milling process

K.P. Karunakaran, R. Shringi[*]

Computer Graphics Laboratory, Department of Mechanical Engineering, Indian Institute of Technology Bombay, Powai, Mumbai 400076, India

ARTICLE INFO

Article history:
Received 18 May 2006
Received in revised form
5 November 2007
Accepted 7 November 2007

Keywords:
Solid modeling
Octree representation
Boundary representation
Image space representation
NC verification
Volumetric NC simulation
MRR

ABSTRACT

Most CAM packages available today provide NC simulation systems that generally use inexact solid representation as they are computationally intensive. Popular inexact volumetric NC simulation packages use either a *voxel based 2.5D approach* (particularly *Image-space modeling*) or *Polyhedral approximations* of 3D objects. The former does not degenerate with time but has limited resolution whereas the latter has a selectable resolution but degenerates rapidly with time. As against these, the authors' use octree solid representation in their volumetric NC simulation system. This hierarchical space decomposition (HSD) technique combines the benefits of both the earlier approaches, i.e., it has a selectable resolution and does not degenerate with time.

The basic input to the octree-based NC simulation system is the NC cutter path either in cutter location (CL) format or *G/M-code* format. After the CL data are interpreted, the swept volume of the cutter is intersected with the blank at every small sampling interval along the tool path and the intersection is considered as the undeformed chip. The chip parameters necessary for predicting the milling force are extracted and using the material removal rate (MRR) model and cutting force is predicted. According to the simulation result the feed rate is adjusted based on force calculation to generate optimized CL File. The improvement can be seen from the experimental results presented.

Such a simulation system will be useful, in addition to the visual check, for automatic verification of safety of machining and conformance of the produced part to design specifications as well as for the optimization of the feed rate and spindle speed.

1. Introduction

CNC machining is widely used for the precision manufacture of automotive, aerospace and heavy engineering parts, dies and molds. Generating the optimal process plan, viz., the NC programs, to produce the required geometry involves determination of the optimal cutter paths and machining parameters. The common practice is to verify the cutter path based on purely geometric simulation and the machining parameters such as feed rate is set to constant value for the number of cutter paths, based on worst case cut geometry using machinability handbooks and experience of skilled machinist. The selected parameters are often so conservative that efficiency is very low during the large part of the machining. This paper presents a virtual machining system that optimizes NC programs based on physical simulation (in contrast to geometric simulation) to improve the efficiency of cutting process.

[*] *Corresponding author*. Tel.: +91 22 2576 7530; fax: +91 22 2572 3480.
E-mail addresses: karuna@iitb.ac.in (K.P. Karunakaran), rshringi@iitb.ac.in (R. Shringi).

0924-0136/$ – see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.jmatprotec.2007.11.092

Nomenclature

a	angle of segment OP measured counter clockwise from the X-axis
b	angle of segment QS measured clockwise from the Z axis
CL	cutter location
CL_i	ith cutter location data giving the next position (x_i, y_i, z_i) and orientation (i_i, j_i, k_i) of cutter
d	diameter of cutter measured from point R
(e,f)	coordinates of point A, the centre of fillet arc
$f_{\mathrm{r\_current}}$	current feed rate
$f_{\mathrm{r\_opt}}$	optimum feed rate
$F_{\mathrm{max\_limit}}$	upper force limit
$F_{\mathrm{predicted}}$	predicted force
F_r	resultant force
F_t	tangential cutting force
F_th	thrust force
$\boldsymbol{g}(u)$	parametric definition of two dimensional profile of cutter in terms of u
h	non parallel height of cutter
h_1	shank length of cutter
MRR	material removal rate
n	number of tool paths
N	spindle rotating speed
OP	first straight line segment corresponding to the bottom cutting edge of cutter
$p(u, v)$	biparametric definition of the surface of the cutter
P_avg	average power
PQ	fillet arc of cutter
P_sp	specific power
QS	second straight line segment corresponding to the side cutting edge of the cutter
r	radius of circular segment PQ of cutter
R	point of intersection of first line segment OP and third line segment QS
s_1	length of first straight line segment OP
s_2	length of arc segment PQ
s_3	length of second straight line segment QS
s_4	length of third straight line segment ST
ST	straight line shank corresponding to the shank of the cutter
v	parameter defining the rotation of $\boldsymbol{g}(u)$ about the Z-axis
V_c	cutting speed

Based on the literature review (Anderson, 1978; Wang and Wang, 1986; Van Hook, 1986; Atherton et al., 1987; Jerard et al., 1990; CGTech, 2003; Chappel Ian, 1983; Oliver and Goodman, 1990a; Drysdale et al., 1989; Jerard et al., 1989; Sungertekin and Voelcker, 1986; Spence and Altinas, 1991; Spence and Altinas, 1994; Roth et al., 2003; Feng et al., 1995; Mounayri et al., 1998; Spence and Li, 2001; Fleisig and Spence, 2005; Wastra et al., 1994; Mortenson, 1985; Ayala et al., 1985; Carlbom et al., 1985; Brunet and Navazo, 1990; Chonglin et al., 1996; Kondo, 1994; Roy and Xu, 1999) the various approaches used for geometric NC simulation and verification can be classified into (i) Image space (ii) Vector based (iii) Object space based NC simulation. The Image space (also called as z-map) approach (Anderson, 1978; Wang and Wang, 1986; Van Hook, 1986; Atherton et al., 1987) is the most widely used for NC simulation. Anderson (Anderson, 1978) introduced z-map method to simulate three axis machining for collision detection. Subsequent investigations were conducted by Wang and Wang (1986), Van Hook (1986) and Atherton et al. (1987) to enhance the capability and simulation efficiency of z-map approach. The image-based systems can give a 3D view of the cutting process and the final part from the original viewing direction, but a true solid model is not available as only a z-buffer image is maintained. Further the errors not visible in the chosen viewing direction are undetected, and generating another view of the part requires restarting the entire simulation (Jerard et al., 1990). Image space approach allows for fast computation and approximate calculation of volume removal rates but its computation time and memory consumption increases drastically if the accuracy is to be enhanced. Commercially reported NC simulator VERICUT (CGTech, 2003) uses image space approach.

Several investigators (Chappel Ian, 1983; Oliver and Goodman, 1990a; Drysdale et al., 1989; Jerard et al., 1989) have used vector-based approach for NC simulation. Chappel Ian (1983) proposed NC simulation method based on the vector clipping approach. Oliver and Goodman (1990a) developed a system similar to Chappel's, which uses a computer graphics image of the desired surface to select the points. This image space is then used as the basis for the simulation. Jerard et al. (1989, 1990) and Drysdale et al. (1989) used an approach that shares characteristics of the methods of Chappel and Oliver, but it also contains features that improve efficiency and allow the user to make tradeoffs between the accuracy of the approximate simulation and the CPU time. The major limitation of vector-based methods is that it does not directly generate a solid model for the machined part in the simulation process and is not suitable for cases when the normal vector directions of a surface at some locations on the part model dramatically change during the simulation process.

The geometric simulation based on 3D object space solid modeling includes constructive solid geometry (CSG), boundary representation (BRep) and spatial decomposition approaches. Constructive solid geometry method is popular because this method can complete Boolean operation of any 3D part model relatively easily and accurately. Sungertekin and Voelcker (1986) have developed a simulation system for milling using PADL-2 CSG modeling system for maintaining a 3D model of the stock-in-progress. Spence and Altinas (1991, 1994) developed a 2–1/2 axis milling process simulation system using CSG for part representation. The limitation of CSG based approach is high computational expense when the complex NC program might contain ten thousand movements, making the computation intractable. The cost of simulation is reported to be $O(n^4)$, where n is the number of tool movements (Roth et al., 2003).

Boundary representation is the dominant choice for commercial modelers. Currently a number of systems based on the spatial Technology Inc. *ACIS* solid modeler kernel have been marketed (AutoCAD, SolidEdge). Feng et al. (1995) used commercial CAD/CAM software CATIA to determine the boundary of engaged surface between tool and work piece for 3-axis chip geometry calculation. Using a BRep polyhedral-based solid modeler ball-end milling simulation was reported in Mounayri et al. (1998). The primary disadvantage of BRep modelers for machining simulation is long running time. Parallel processing and BRep topology to reduce overall wall clock running time of BRep based machining simulations for rough machining of the 2–1/2 D pockets has been reported in literature (Spence and Li, 2001; Fleisig and Spence, 2005). It was shown that, for a total of n tool movements, the BRep growth rate is estimated to be $O(n^{1.5})$.

Wastra et al. (1994) developed a simulation system based on the uniform spatial decomposition (USD) using voxels. The voxel model has simple data structure which enables to generate fast updating of a part model. However to increase the accuracy of cell decomposition in the model, the size of the voxels needs to be reduced. This results in huge memory space requirement for storing the model. Several hierarchical space decomposition (HSD) based modeling approaches have been reported in literature (Mortenson, 1985; Ayala et al., 1985; Carlbom et al., 1985; Brunet and Navazo, 1990; Chonglin et al., 1996; Kondo, 1994; Roy and Xu, 1999) to reduce the storage requirements. A HSD based model is similar to a USD in that it represents a solid as an aggregate of hexahedra, but it reduces the memory requirement considerably by dividing the space adaptively. HSD models may be of different types such as Octree, bintree or polytree etc. In an octree representation (ORep) the universe (a cube that contains the object) is subdivided into eight parts recursively. Each cube is one-eighth of its parent cube in size and is called an octant. Furthermore, all the octants can be represented as the nodes of a tree in which every node has eight branches (Fig. 1). This tree is called Octree (Mortenson, 1985).

Roy and Xu (1999) applied the extended octree modeling technique to machining simulation. However, Roy's model search cutting area using quadtree in an envelope projected on a xy plane. It is thus difficult to say when an octree structure was used for machining simulation. No commercial HSD based NC simulation system is available though these have been reported in literature (Roy and Xu, 1999).

Based on the review of the popular geometric representation schemes for NC simulation system, a point wise comparison is given in Table 1. In table the '+' sign denotes that the factor is favorable from the point of view of NC simulation system and '−' sign shows that the factor is not favorable.

Although ORep is an inexact representation, one can choose the desired resolution optimally i.e., one can use a poor resolution for quick visual simulation and a high resolution for dimensional verification. The ORep is ideally suited for NC simulation as the size of ORep is independent of the size of the NC program and boolean operations are very trivial since they involve only tree traversals performing binary boolean operations on the corresponding leaf nodes. Further, all the computations take place in binary or unsigned integer modes and this minimizes space and time complexity of the algorithms.

In this paper the authors have presented an octree solid modeling based off-line adaptive controller (Oct–OAC). With this system the physical model of machining can be integrated with the geometric modeling methods to determine if the cutting conditions are safe. With a model of the machined

Table 1 – Comparison of representation schemes for NC simulation

Feature	Representation scheme		
	Image-space	BRep	HSD
Model size as simulation proceeds	(+) remains constant	(−) linearly increases	(+) remains constant (significantly less than voxel-rep)
Simulation time per NC block as simulation proceeds	(+) remains constant	(−) factorially increases	(+) remains constant
Resolution selection	(−) the required resolution value cannot be directly specified; it is decided by the viewing conditions	(+) the required resolution value can be directly specified	(+) the required resolution value can be directly specified
Accuracy	(−) poorer in directions other than the simulation direction	(+) same in all directions	(+) same in all directions
Isotropy	(−) not isotropic	(+) isotropic	(+) isotropic

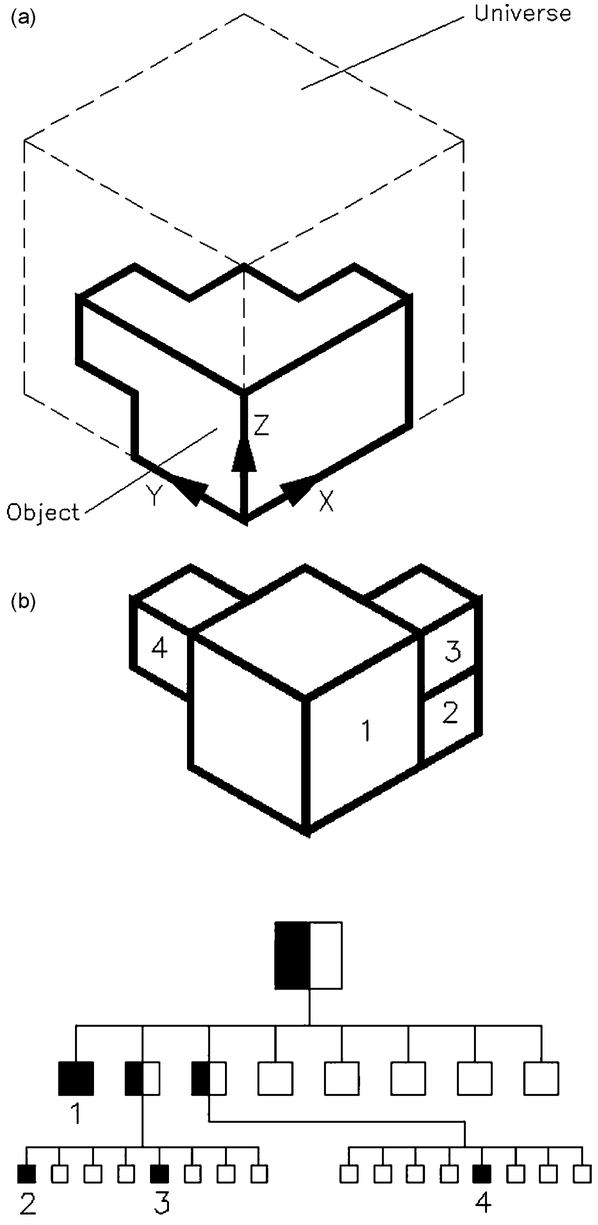

Fig. 1 – Octree representation of an object (a) object and universe. (b) Octree of the object.

object always present during the simulation it can be used to calculate the material removal rate and from that optimum feed rates can be determined. It is shown that optimization of cutting conditions during NC machining using Oct–OAC can improve the productivity and efficiency of CNC machines.

2. Architecture of octree-based off-line adaptive control NC simulation system

Using octree-based solid modeling approach, Oct–OAC i.e., octree-based off-line adaptive control NC simulation system is being developed by authors. The architecture of the Oct–OAC Simulation system is shown in Fig. 2. It consists of the seven internal modules namely; (i) CSG modeler (ii) NC program file manager and translator (iii) octree modeler (iv) octree to BRep converter (v) generic cutter modeler (vi) module for swept volume calculation (vii) optimization module. These modules are discussed below.

2.1. CSG modeler

The *CSG modeler* has the basic primitives of block, sphere, cone, cylinder and torus and Boolean operations of union, subtraction and intersection. The user can create blank, fixture, clamps and tool holders using this modeler. The blank provides the initial geometry defined as solid. The fixtures also defined as solid can be used for checking possible collisions with the tool. The ideal part design can be used for checking finish cut with tolerances. Information gathered from this CSG modeler is used to create octree model of the object, which is stored in the system for further computations. This also has a STL interface for accepting geometries from other CAD systems.

2.2. NC program file manager and translator

NC program file manager and translator has two sub-modules namely CL file interpreter and Generic inverse processor. CL file contains machine independent list of cutter movements and auxiliary machine control information. The basic input to the simulation system is the CL file which is interpreted by the CL file interpreter. Based on the type of movement (rapid, linear or circular) and the number of simultaneous axes moving, it calls the appropriate routine for calculating the swept volume. This module first opens the given CL file and reads it into a buffer. A line at a time is scanned from this buffer and stored in a character array called current block. The keyword is then extracted from this current block. The keyword may be MSYS, TLDATA, FEDRAT, RAPID, GOTO, GODLTA, CIRCLE etc. Separate routines handle different keywords. Position of the cutter is stored in a static variable after each block. This is required to know the start point of the next statement. Sometimes it may be necessary to read the next line for the current line to make sense, for example the RAPID and the CIRCLE statements. In such cases the next line specifies the target point of the cutter motion.

Often, the user may be interested to simulate the NC program that is in *G/M-code* format. For this purpose, a generic inverse processor is used that takes both the NC program and the machine tool data file as input and generates the corresponding CL data file as output. The machine tool file contains the details of process, kinematics and formats for the particular CNC machine.

2.3. Octree modeler

Octree modeler creates the octree model of the blank, clamps, and fixtures etc. Octree modeler also creates octree model of the cutter and the swept volume generated by the cutter motions. This modeler also provides Boolean operators (Union, Subtraction and Intersection) to operate on different solids. Boolean operations of the swept volume of the cutter with the clamps, fixtures and non cutting parts are required for collision detection (Drysdale et al., 1989; Oliver and Goodman,

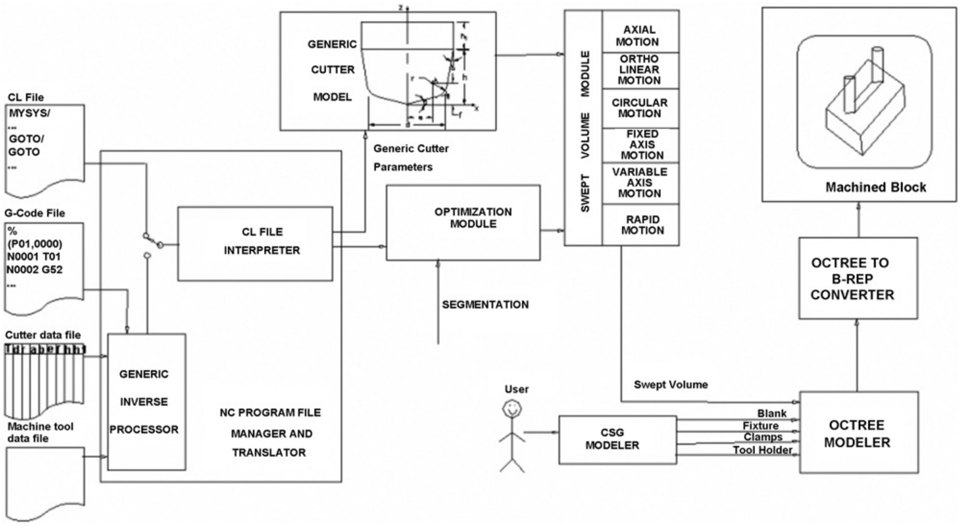

Fig. 2 – Architecture of octree-based off-line adaptive NC simulation system.

1990b). An algorithm for octree creation from an STL file is also provided.

2.4. *Octree to BRep converter*

The isometric rendering of octree can be achieved by simple tree traversal with trivial geometric computation involved. But, it makes use of only three colors and hence is unable to provide adequate contrast required to depict edges. Furthermore it will be difficult to provide different colors to different objects and hidden line removal when multiple objects are present. Moreover for detailed visual verification, one requires to rotate and zoom the object at arbitrary angles and scales. In order to achieve these zoom, rotate and rendering features for visual verification and measurements required for dimensional verification, it is required to convert octree to BRep. Furthermore the designed component is generally available as a BRep model and in order to identify the dimensional deviations and depict them, it is required to convert octree of the machined component to BRep. The octree to BRep converter module takes care of this. Once the BRep model is obtained, OpenGL (Jackie et al., 1993) displays the model. Further as the conversion of octree of the machined component into BRep is much simpler and faster than the reverse, the blank is always stored as octree and the user can convert it into BRep for visual verification as often as required and for dimensional verification at the end (Karunakaran and Shringi, 2007). The present system supports two type of display techniques, one is an isometric display obtained by tree traversal and the other is an OpenGL based BRep display of the object. Conversion of octree to BRep is required in case of OpenGL display.

2.5. *Generic cutter modeler*

In order to handle a variety of cutter shapes in a unified manner, the present work uses a parametric model of a generic cutter. The generic cutter is an axisymmetric bi-parametric surface obtained by revolving a planar profile (generatrix) about its axis (directrix). It is a piecewise continuous planar curve consisting of three straight-line segments and a circular fillet between the root flank and the side flank. The generatrix profile is defined by means of eight parameters, viz., d, r, e, f, a, b, h and h_1 as shown in the Fig. 3. By appropriately choosing these parameters, a variety of cutter shapes such as flat end mill, dome end mill, ball-end mill, angle cutters, and face mill can be obtained. Further details of this generic cutter formulation may be seen in Dhande and Karunakaran (1994).

From the eight parameters of generic cutter as shown in Fig. 3, the coordinates of points O, P, Q, S, T, R and A, and the arc lengths (s_1, s_2, s_3 and s_4) are calculated as shown in Appendix A. The parametric equation of the two-dimensional profile of the cutter in terms of u can be defined as:

$$g(u) = [x_g(u) \quad y_g(u) \quad z_g(u)] \quad \text{where } 0 \le u \le s_1 + s_2 + s_3 + s_4.$$

Also, the bi-parametric equation of the axisymmetric surface of the cutter in terms of u and v where v is the parameter

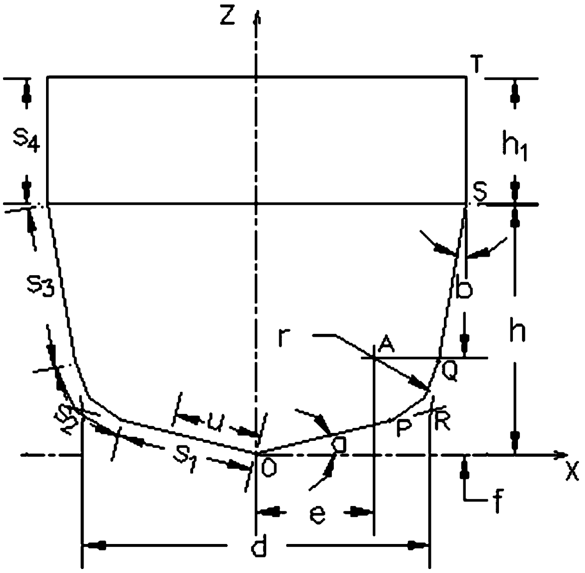

Fig. 3 – Parametric profile of generic cutter.

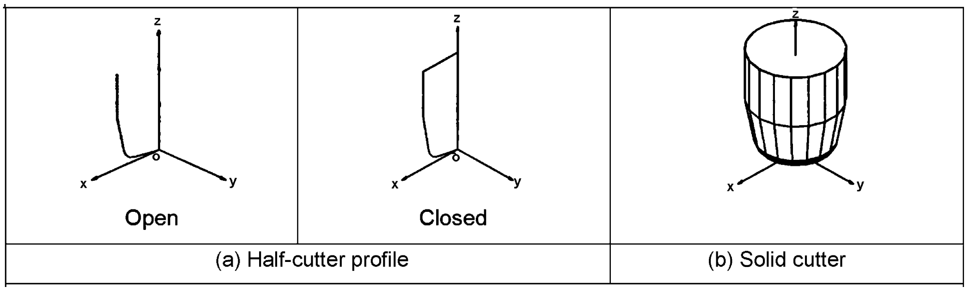

Fig. 4 – Half cutter profile of the generic cutter. (a) Half-cutter profile, (b) solid cutter.

defining the rotation of $g(u)$ about z axis can be written as (see Fig. 4b):

$$\boldsymbol{p}(u, v) = [x_g(u)\cos v \quad x_g(u)\sin v \quad z_g(u)]$$

where $0 \leq u \leq s_1 + s_2 + s_3 + s_4$ and $0 \leq v \leq 2\pi$.

There are eight parameters that define one half of the planar profile $g(u)$ of the cutter. This profile will be referred as *half cutter profile*. Fig. 4a shows two types of the half cutter profiles used in swept volume algorithms. If the half cutter profile is revolved by π radian, one can obtain the half cutter surface. Further if the half cutter profile is revolved by 2π radian, one can obtain the solid cutter (Fig. 4b). Whenever "TLDATA", the tool change instruction in CL file is read, all the cutter parameters and the geometries of cutter shown in Fig. 4 are calculated and stored at origin. To calculate the swept volume of any motion command, it is required to copy the appropriate cutter geometry and position and orient it at $\mathrm{CL}_i = (x_i, y_i, z_i, i_i, j_i, k_i)$. To achieve this, we determine the homogeneous transformation matrix for CL_i. This transformation matrix is the concatenation of the following transformations:

(i) rotation of the geometry by an angle $-\theta$ about X axis where $\sin\theta = j_i$ and $\cos\theta = \sqrt{k_i^2 + i_i^2}$
(ii) rotation of the geometry by an angle φ about Y axis where $\sin\varphi = i_i/\sqrt{k_i^2 + i_i^2}$ and $\cos\varphi = k_i/\sqrt{k_i^2 + i_i^2}$
(iii) translation of the geometry by (x_i, y_i, z_i).

The resulting homogeneous transformation matrix [T] is

$$[T] = \begin{bmatrix} \dfrac{k_i}{\sqrt{k_i^2 + i_i^2}} & 0 & -\dfrac{i_i}{\sqrt{k_i^2 + i_i^2}} & 0 \\ -\dfrac{i_i j_i}{\sqrt{k_i^2 + i_i^2}} & \sqrt{k_i^2 + i_i^2} & -\dfrac{j_i k_i}{\sqrt{k_i^2 + i_i^2}} & 0 \\ i_i & j_i & k_i & 0 \\ x_i & y_i & z_i & 1 \end{bmatrix}$$

When this matrix operates on the geometry such as the ones shown in Fig. 4b defined in terms of vertices defined as homogeneous row vectors, i.e., a matrix of size [1 × 4], it gets transformed from origin to CL_i.

2.6. *Module for swept volume calculation*

This is the core modeler for creating the swept volumes. The swept volume is organized into the six types. (i) axial motion (ii) orthogonal linear motion (iii) circular motion (iv) fixed-axis motion (v) variable-axis motion (vi) rapid motion. This classification is driven by the geometry of the cutter path and its relation with the cutter axis (Karunakaran et al., 2000). Each has a separate algorithm for calculation of swept volume. The swept volume algorithms used take into account all variations of cutter shapes, motion types and machine kinematics. Once the swept volume is calculated, boolean subtraction of the swept volume is done from the blank to obtain the instantaneous geometry of the blank.

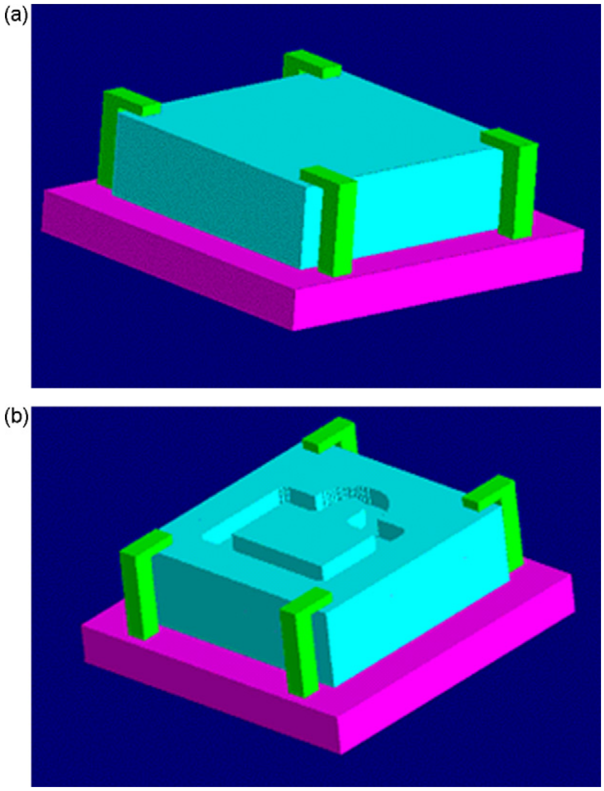

Fig. 5 – Groove milling: example of 2.5-axis simulation. (a) Blank, (b) output from the octree-based NC simulation system of the authors.

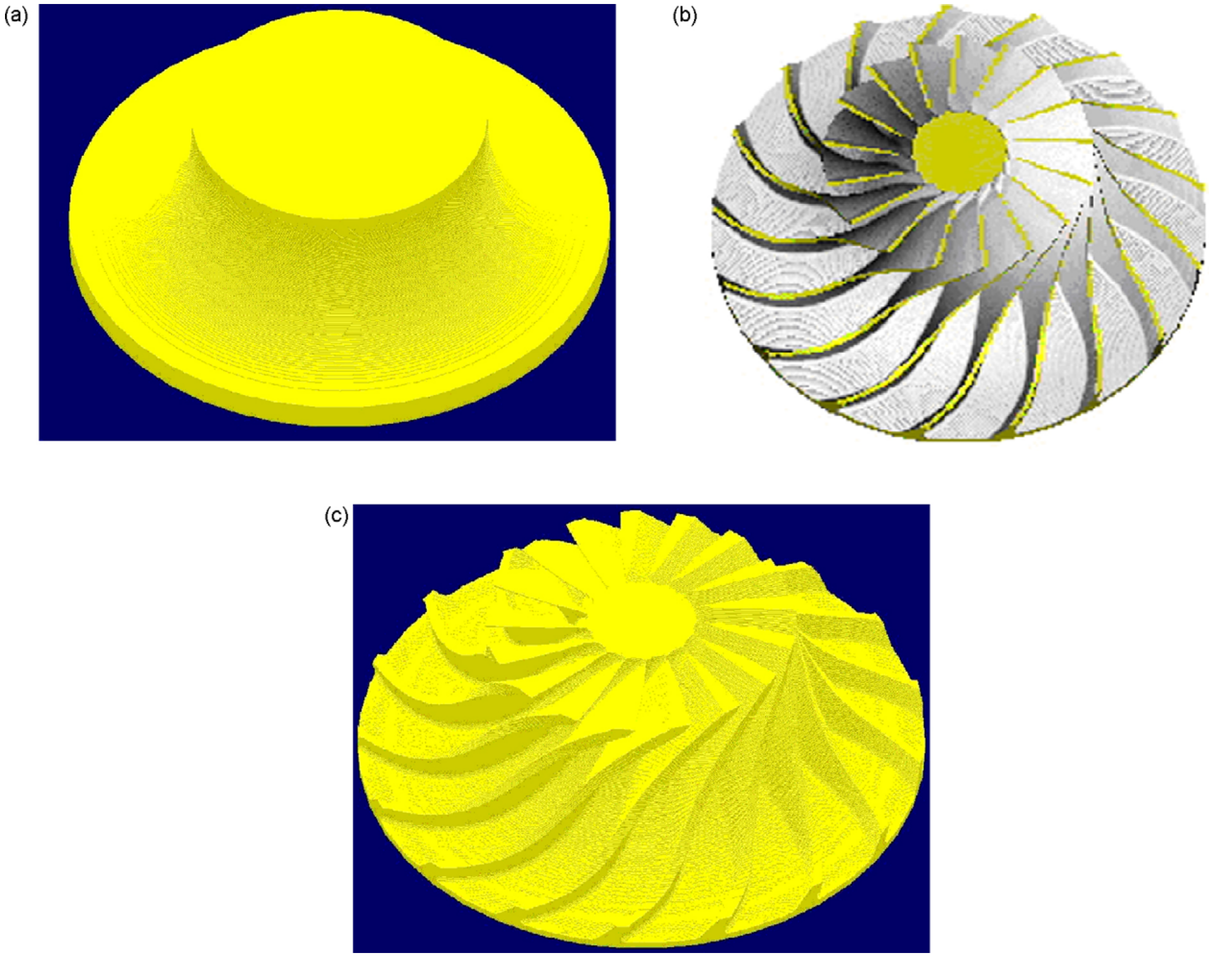

Fig. 6 – Impeller blades: example of 5-axis simulation. (a) Blank of impeller, (b) vericut output, (c) output from the octree-based NC simulation system of the authors.

3. Illustrations of geometric simulation using Oct–OAC

Octree-based NC simulation system is able to simulate large files up to 5-axis motions. For 2.5-axis machining a groove-milling program is chosen for simulation. Geometry of the blank after simulation is as shown in Fig. 5. The virtual machining of an impeller machined in 5-axis mode is shown in Fig. 6. Fig. 6a is its blank, Fig. 6b and 6c are respectively the simulation results obtained from VERICUT and Octree-based NC simulation i.e., Oct–OAC developed by authors.

4. Optimization module

The optimization of NC program using Oct–OAC is done in two steps:

(i) geometric simulation,
(ii) physical simulation.

4.1. *Geometric simulation of cutting process*

The concept of off-line adaptive control is based on the use of current cutting conditions as the basis of monitoring the process. One of the important indications of the current cutting conditions is the geometry of material removed when the cutter moves along the specified path. The geometric simulation of machining calculates the instantaneous contact area between the cutter and work piece at any time. The optimization module reads the NC tool path file and divides the motion into a number of smaller segments based on user specified sampling interval. Given the shape of the material removed obtained by boolean intersection of the cutter and in-process blank (Fig. 7d), the octree model of the contact area between the cutter and the blank at every sampling interval along the cutter path is determined (Fig. 7e). From the octree of the contact area (considered as undeformed chip) the geometrical parameters necessary for the physical simulation are calculated. For example to calculate the cutting force the geometrical parameters needed may include chip volume, axial and radial depth of cut, cutting thickness etc and this data are extracted from the octree of the contact area as shown in Fig. 7e. By analyzing the geometry of this contact area and by

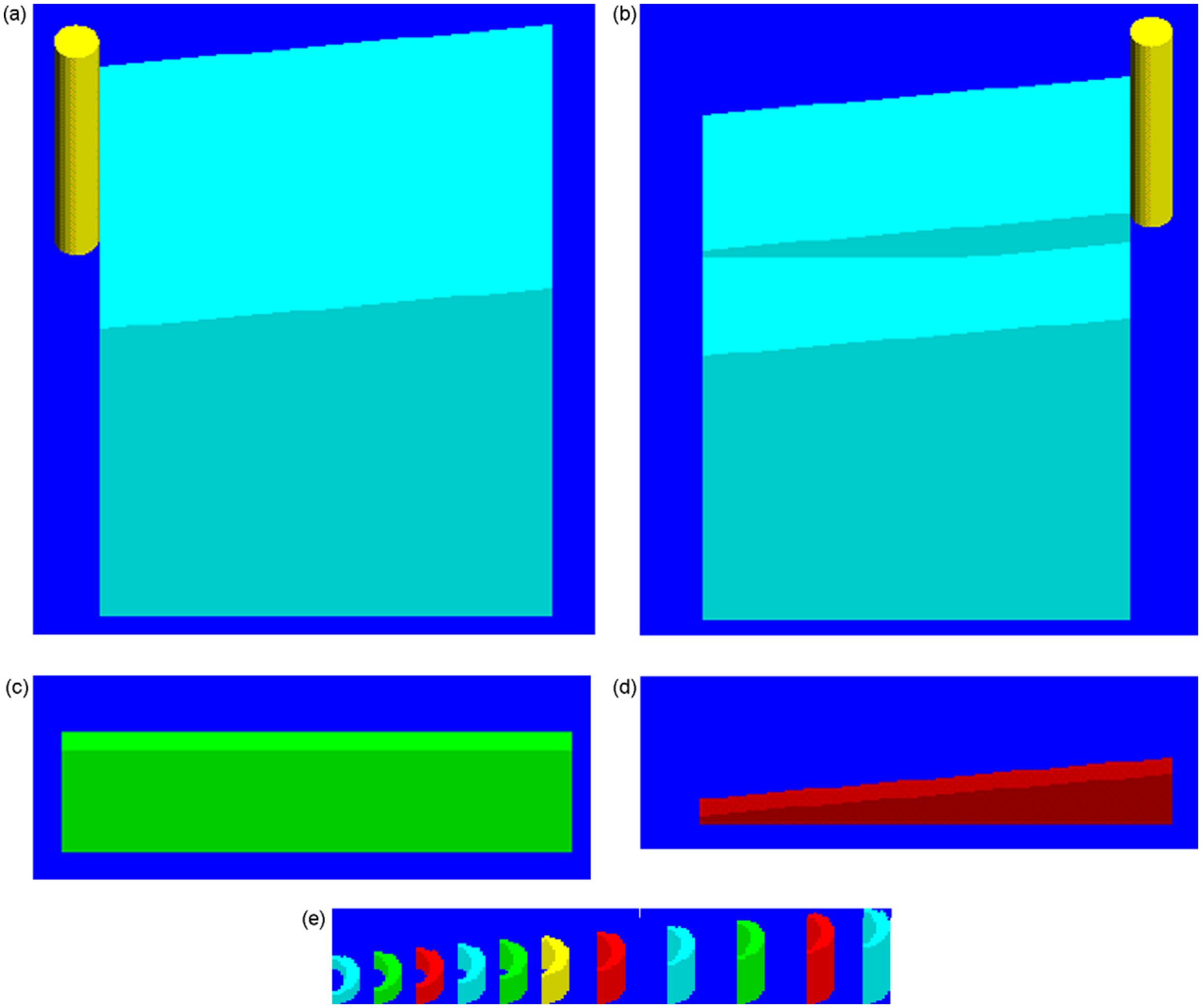

Fig. 7 – Principle of off-line adaptive control using Oct–OAC. (a) Cutter at the beginning of motion, (b) cutter at the end of motion, (c) swept volume of cutter, (d) shape of material removed, (e) contact area between cutter and blank in octree.

calculating the cutting forces the optimal feed rate for every sampling interval can be found out (Fig. 8).

4.2. *Physical simulation of cutting process for feed rate scheduling*

The simulation of physical cutting process is required for selecting cutting conditions that are both safe and efficient. The goal is to cut parts as quickly as possible while maintaining the safe cutting conditions and high part quality. This step deals with the simulation of the end milling process for prediction of the cutting forces during the cutting process. The physical parameters such as depth of cut, feed rate and cutting speed have the greatest effect on the success of a machining operation. Depth of cut is usually predetermined by the work piece geometry and operation sequence. It is recommended to machine the features with the required depth in one pass to keep machining time and cost low, when possible. Therefore, the problem of determining machining parameters is reduced to determining the proper cutting speed and feed rate combination. Here, it is assumed that the cutter path (axial depth of cut and radial width) is pre-selected and the spindle speed is chosen using machinability data. As the part geometry varies along the cutter path, the feed rate is scheduled by the Oct–OAC based process simulation.

For Optimization currently Oct–OAC adopts the material removal rate (MRR) based on average force calculations as the basic physical model for physical simulation. The predicted cutting force (tangential and radial forces) using the MRR model is used as feed back from the machining process to optimize the feed rate. Three approaches for optimization are used; (i) maximum force limitation (ii) force range limitation (iii) minimum force limitation. The maximum force limitation method is used to adjust cutting force to be as large as possible but less than the upper specified limit and the force range limitation method is to make the force fall into the desired specified range. The minimum force limitation method makes cutting force as less as possible but greater than the lower specified limit. The feedrate is adjusted to reach optimization objectives according to the predicted cutting force.

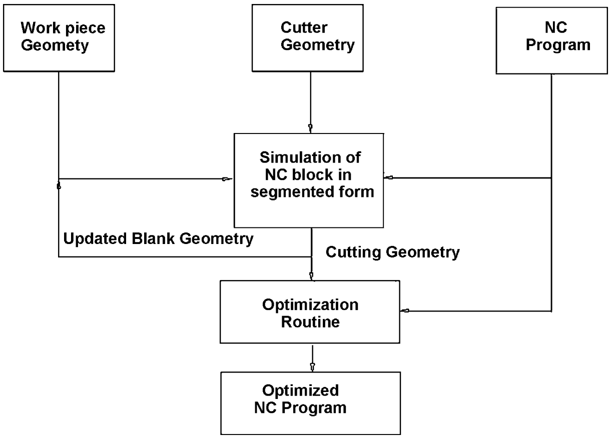

Fig. 8 – Block diagram of NC program simulation and optimization.

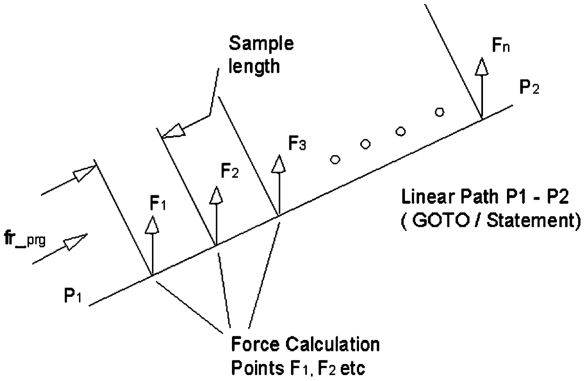

Fig. 9 – Block diagram of feed rate optimization.

The feed rate programmed by the NC programmer is the reference input to the system. The program starts processing the cutter paths after positioning moves of the cutter in RAPID mode (G00 in NC data). It checks whether initial tool, spindle speed and feed rate are defined. Then for each NC pass, i.e., APT GOTO statement for user specified sampling interval, using the geometric simulation the chip volume is calculated at every sampling interval, from the octree of the contact area representing the amount of material removed. Once the chip volume is known the MRR can be calculated and the resultant cutting force is predicted at each sampling interval as shown in Fig. 9.

Further at the user specified sampling intervals, the optimal feed rate is calculated and the segments of the cutter path between successive intervals are output as NC block with optimum feed rates (fr_opt). Although there is no simple relationship between cutting force and feed rate, here it is assumed that modifying the feed rate, based on a linear relationship between the cutting force and the feed rate, gives reasonable results as a first approximation (Fussell and Srinivasan, 1989). The adjusted feed rate, is calculated from:

$$f_{\text{r\_opt}} = f_{\text{r\_current}} \times \left[\frac{F_{\text{max\_limit}}}{F_{\text{predicted}}}\right] \qquad (1)$$

The average force is calculated using following procedure:

- Calculate the length of the NC block i.e., path between two end points of block. For user specified sampling lengths determine the number of segments for the given NC block. for ∀ segments:
- Calculate locally next point which is the end point of the current segment and create swept volume for this segment.
- Store the initial shape of the blank and call it temp_blank and carry out the boolean subtraction of the swept volume from the octree of the blank.
- Transform the cutter to the next point and do the boolean subtraction of the cutter from the octree of the blank and subtract the octree of the blank from temp_blank to generate the cut volume representing the amount of material removed and its volume.
- The MRR is then:

$$\text{MRR} = \frac{\text{Volume removed}}{\text{time}} = \frac{\text{Volume removed}}{(\text{segment length})/\text{feed rate}}$$

- Given the value of specific power (P_{sp}) for different material and cutter combinations from handbook (MCSD, 1980) and MRR calculated, the average power (P_{avg}) consumed is given by:

$$P_{avg} = P_{sp} \times \text{MRR} \qquad (2)$$

- The tangential cutting force (F_t) can be calculated as:

$$F_t = 60000 \times \frac{P_{avg}}{V_c}$$

where cutting speed V_c is given by π DN.

- Over a wide range of materials and cutting conditions thrust force (F_{th}) is approximately equal to 0.5 F_t (Tlusty and Smith, 1991).

$$F_{th} = 0.5 \times F_t$$

- The resultant force (F_r) can be obtained from F_t and F_{th} using:

$$F_r = \sqrt{F_t^2 + F_{th}^2} \qquad (3)$$

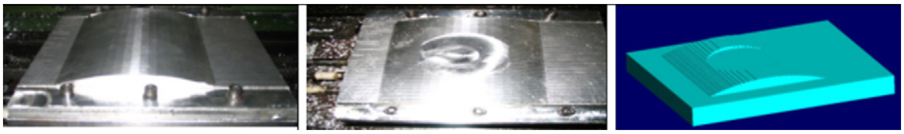

Fig. 10 – View of the circular profile milling. (a) Workpiece before machining, (b) workpiece after machining, (c) output from Oct–OAC.

- This resultant force is compared with the specified limiting force and optimum feed rate for the current segments using Eq. (1).
- Output the optimized CL file with the modified feed rate for the current segment and delete the stored temp_blank.
- Update the current point with the next point locally and repeat iteration for all segments similarly.

5. Experimental verification

To evaluate the validity of Oct–OAC cutting experiments were performed on 2½ axis CNC milling machine (Deckel FP4A Maho). For measurement of cutting forces three component milling dynamometer (Kistler 9257A), charge amplifier (Type 5051A) and data acquisition system were used. The dynamometer has three component piezoelectric force transducers (quartz crystal) which measures forces in three orthogonal directions. The Kistler charge amplifier receives charge signal from the piezoelectric sensors and converts it into a proportional voltage. The charge amplifier has RS232C port which allows transferring the data to PC. The voltage output of the charge amplifier is connected to a data logger which acquires data by converting analog data into digital signals. A windows based c++ program is used to read the digital data and store it into a file and simultaneously display it on screen.

As shown in Fig. 10, a circular hole was milled on a cylindrical surface. The cutting path includes a series of concentric circles and several lines. Along the path in NC block of the cutter movement, the cutter comes across variable axial depth of cut hence the feed rate optimization becomes advantageous. The overall size of the aluminum work piece (Al6061 grade) is 153 mm wide by 180 mm long by 25 mm thick. In this example, cutter used is HSS, 20 mm diameter; six fluted flat end mill with helix angle of 30°. The spindle speed used was 500 rpm.

The simulation of the forces was performed using a 512 MHz Intel PIV based microcomputer. Using the unoptimized CL file given in Table 2, the forces from the MRR model is predicted and compared with the measured forces (Fig. 11). The values along X axis show the linear distance of cutter movement. For the circular portion of the cutter path the arc length is taken. During the experiment, the force data was collected at intervals of 0.01 s. Fig. 11 shows that the force values predicted from the MRR model using Oct–OAC and the measured force values are in close agreement. The cutting time from un-optimized file was found out to be 251 s. During milling of circular hole along the cutting path, the material to be removed and the geometrical cutting parameters change with the cutter positions and therefore the cutting forces also changes with the cutter positions. It can be seen from Fig. 11 that the change in the cutting force matches with the work piece shape change when MRR model is used to predict the cutting forces. The force increases when the metal removal rises and decreases when the removed material volume declines.

Table 2 – Comparison of un-optimized and optimized CL files

Un-optimized CL file

```
MSYS/75.0000,76.500,50.0000,1.0000000,0.0000000,0.0000000,
0.0000000,1.0000000,0.000000
   TLDATA/MILL,20.000,0.000,0.000,0.000,0.000,10.000,0.00
   FEDRAT/MMPM,50
   GOTO/75,76.5,22
   GOTO/65,76.5,22
   CIRCLE/75,76.5,22,0.00,0.00,1.00,10.00
   GOTO/85,76.5,22
   CIRCLE/75,76.5,22,0.00,0.00,1.00,10.00
   GOTO/65,76.5,22
   GOTO/55,76.5,22
   CIRCLE/75,76.5,22,0.00,0.00,1.00,20.00
   GOTO/95,76.5,22
   CIRCLE/75,76.5,22,0.00,0.00,1.00,20.00
   GOTO/55,76.5,22
   GOTO/55,76.5,50
   SPINDLE/OFF
   FINI
```

MRR model (optimized CL File)

```
MSYS/75.000000,76.500000,50.000000,1.000000,0.000000,0.000000,
0.000000,1.000000,0.000000

TLDATA/MILL,20.000000,0.000000,0.000000,0.000000,0.000000,
10.000000,0.000000
   FEDRAT/MMPM,50.00
   GOTO/75.000000,76.500000,22.000000
   FEDRAT/MMPM,79.83
   GOTO/72.500000,76.500000,22.000000
   FEDRAT/MMPM,68.87
   GOTO/70.000000,76.500000,22.000000
   FEDRAT/MMPM,79.83
   GOTO/67.500000,76.500000,22.000000
   FEDRAT/MMPM,83.63
   GOTO/65.000000,76.500000,22.000000
   FEDRAT/MMPM,99.08
   CIR-
CLE/75.000000,76.500000,22.000000,0.000000,0.000000,1.000000,
10.000000
   GOTO/65.405070,79.317326,22.000000...
```

Table 3 – Comparison of model predicted cutting time, maximum and average forces for un-optimized and optimized feed rates

	Un-optimized case		Optimized case	
	Measured	Predicted	Measured	Predicted
Cutting time (s)	251	251	46	46
Maximum force (N)	234.9	201.8	350.2	300
Average force (N)	105.8	90.2	244.3	270.1

The optimized CL file as shown in Table 2 is generated by the segmentation of NC block at 3 mm interval using MRR based force model using Oct–OAC. Fig. 12 shows the comparison of forces predicted by the optimization module of Oct–OAC and experimentally obtained resultant forces for optimized CL file. Here, the upper bound of 300 N maximum force values is used for adjusting the feed rate. The average force values moves close to maximum force mark (300 N) in both cases. The cutting time for optimized cutting i.e., variable feed rate cutting (Fig. 13) came out to be 46 s from MRR model.

Table 3 shows the comparison of models predicted cutting time, average and maximum forces for un-optimized and optimized cuts. It can be observed that the measured value of average force from experiments is in closer agreement with the average predicted force value from the MRR model using Oct–OAC. Further, the predicted force shows fairly good agreement with the measured force. Improvements can be seen from the contrast of the cutting force curves of the un-optimized and optimized cases (Figs. 11 and 12). In maximum force optimization using the MRR model using Oct–OAC, the cutting time is shortened to almost one fifth of the cutting time without optimization.

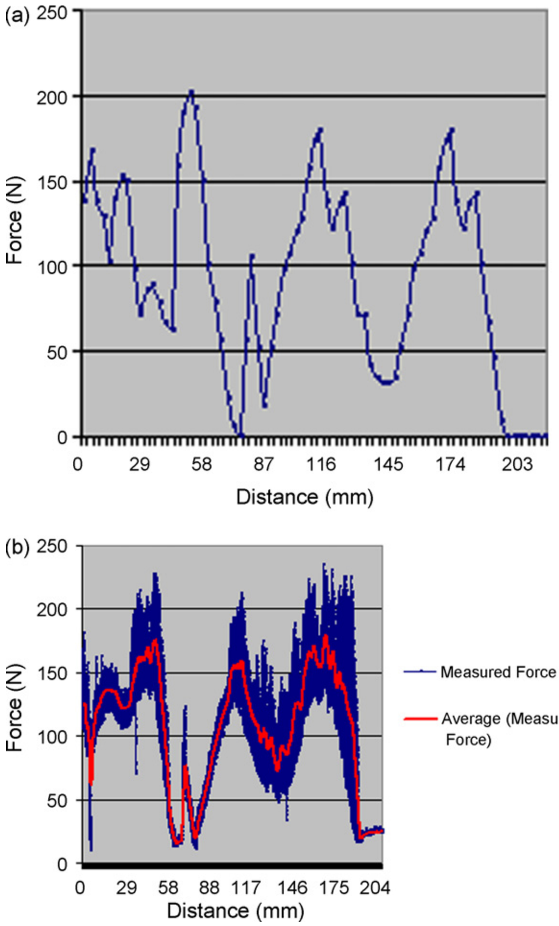

Fig. 11 – Resultant force plot in un-optimized cutting using constant feed rate. (a) Predicted forces (b) Measured forces.

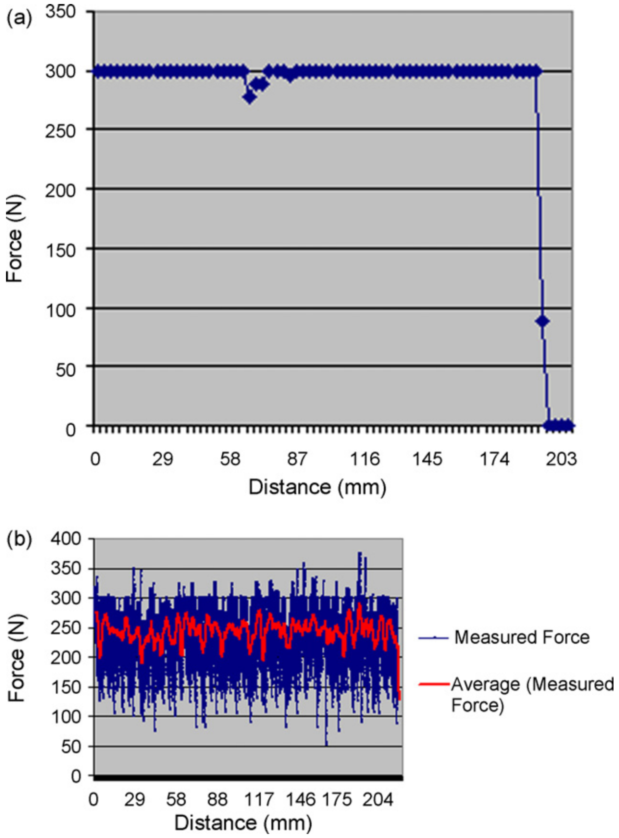

Fig. 12 – Resultant force plot in optimized cutting using variable feed rate. (a) Predicted forces (b) Measured forces.

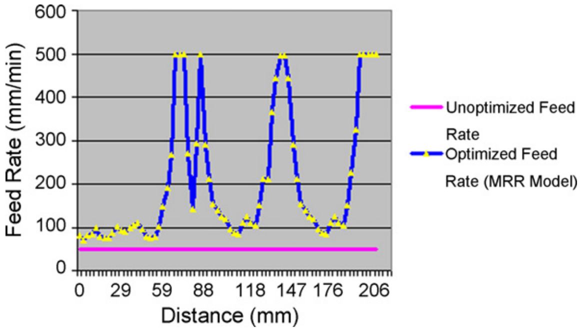

Fig. 13 – Variation of feed rate along the cutter path.

6. Conclusions

In this paper, a solid model-based integrated volumetric NC simulation system using octree solid modeling system Oct–OAC to simulate cutting process and optimize the cutting parameters has been presented. Presently the NC program optimization is effective using the MRR model. By analyzing the simulated results it is shown that the estimated cutting force agrees well with the experimental result; and its optimization function can effectively adjust the cutting parameters to satisfy the cutting force constraints.

Although the time complexity of the developed Oct–OAC is higher than the existing Image space based systems like Vericut but it may be possible to develop more accurate cutting process optimization algorithm for machining using developed simulation system. Further research is continuing to improve the prediction of the cutting force using advanced physical models for simulation to be used in industrial environment.

Appendix A. Calculation for generic cutter

Calculation of the coordinates of points O, P, Q, S, T, R and A, and the arc lengths (s_1, s_2, s_3 and s_4) from the eight parameters (d, r, e, f, a, b, h and h_1) of the generic cutter as shown in Fig. 3 are calculated as follows:

- *Step 1*: Calculate Point O: It is assumed that bottom centre point of the cutter is located at (0, 0, 0). Therefore the coordinates for the point O (x_o, y_o, z_o) will be (0, 0, 0).
- *Step 2*: Calculate point P (x_p, y_p, z_p):

$$\begin{aligned} x_p &= (e + f\tan a)\cos^2 a \\ &\quad - \sqrt{[(e + f\tan a)\cos^2 a]^2 - (e^2 + f^2 - r^2)\cos^2 a} \\ y_p &= 0 \\ z_p &= x_p \tan a \end{aligned}$$

- Step 3: Calculate point Q (x_q, y_q, z_q):

$$\begin{aligned} z_q &= \left\{f + [e + \frac{d}{2}(\tan a \tan b - 1)]\tan b\right\}\cos^2 b \\ &\quad + \sqrt{\begin{array}{l}\left[\left\{f + \left[e + \dfrac{d}{2}(\tan a \tan b - 1)\right]\tan b\right\}\cos^2 b\right]^2 \\ -\left\{f^2 + \left[e + \dfrac{d}{2}(\tan a \tan b - 1)\right]^2 - r^2\right\}\cos b\end{array}} \\ y_q &= 0 \\ x_q &= z_q \tan b - \frac{d}{2}(\tan a \tan b - 1) \end{aligned}$$

- *Step 4*: Calculate point S (x_s, y_s, z_s):

$$\begin{aligned} x_s &= h\tan b - \frac{d}{2}\tan a \tan b + \frac{d}{2} \\ y_s &= 0 \\ z_s &= h \end{aligned}$$

- *Step 5*: Calculate point A (x_a, y_a, z_a):

$$\begin{aligned} x_a &= e \\ y_a &= 0 \\ z_a &= f \end{aligned}$$

- *Step 6*: Calculate arc length s_1, s_2, s_3 and s_4:

$$\begin{aligned} s_1 &= \sqrt{(x_p)^2 + (z_p)^2} \\ s_2 &= r\left\{\tan^{-1}\left(\frac{z_q - f}{x_q - e}\right) - \tan^{-1}\left(\frac{z_p - f}{x_p - e}\right)\right\} \\ s_3 &= \sqrt{(x_s - x_q)^2 + (z_s - z_q)^2} \\ s_4 &= h_1 \end{aligned}$$

Define the parametric equation of the two-dimensional profile of the cutter in terms of u as:

$$\boldsymbol{g}(u) = [x_g(u) \quad y_g(u) \quad z_g(u)] \quad \text{where } 0 \le u \le s_1 + s_2 + s_3 + s_4.$$

For the interval $0 \le u < s_1$

$$\begin{aligned} x_g(u) &= u\cos a \\ y_g(u) &= 0 \\ z_g(u) &= u\sin a \end{aligned}$$

For the interval $s_1 \le u < s_1 + s_2$

$$\begin{aligned} x_g(u) &= e + abs(r)\cos\left\{\tan^{-1}\left(\frac{s_1\sin a - f}{s_1\cos a - e}\right) + \left(\frac{u - s_1}{r}\right)\right\} \\ y_g(u) &= 0 \\ z_g(u) &= f + abs(r)\sin\left\{\tan^{-1}\left(\frac{s_1\sin a - f}{s_1\cos a - e}\right) + \left(\frac{u - s_1}{r}\right)\right\} \end{aligned}$$

For the interval $s_1 + s_2 \le u < s_1 + s_2 + s_3$

$$\begin{aligned} x_g(u) &= [u - (s_1 + s_2 + s_3)]\sin b + x_s \\ y_g(u) &= 0 \\ z_g(u) &= [u - (s_1 + s_2 + s_3)]\cos b + z_s \end{aligned}$$

For the interval $s_1 + s_2 + s_3 \le u \le s_1 + s_2 + s_3 + s_4$

$$\begin{aligned} x_g(u) &= x_s \\ y_g(u) &= 0 \\ z_g(u) &= [u_1 - (s_1 + s_2 + s_3)] + z_s \end{aligned}$$

REFERENCES

Anderson, A.O., 1978. Detecting and elimination collisions in NC machining. Comput. Aided Des. 10 (4), 231–237.

Atherton, P.R., Earl, C., Fred, C., 1987. A graphical simulation system for dynamic five-axis NC verification. In: Proceedings of Autofact, SME, Dearborn, MI, November 1987, pp. 2.1–2.12.

Ayala, D., Brunet, P., Juan, R., Navazo, I., 1985. Object representation by means of nonminimal division quadtrees and octrees. ACM Trans. Graphics 4 (1), 41–59.

Brunet, P., Navazo, I., 1990. Solid representation and operation using extended octrees. ACM Trans. Graphics 9 (2), 170–197.

Carlbom, I., Chakravarty, I., Vanderschel, D., 1985. A hierarchical data structure for representing the spatial decomposition of 3D Object. IEEE Comput. Graphics Appl., 25–31.

http://www.cgtech.com: 2003 Website of CGTech, USA, the developers NC simulation package VERICUT.

Chappel Ian, T., 1983. The use of vectors to simulate material removed by numerical controlled milling. Comput. Aided Des. 15 (3), 156–158.

Chonglin, L., Donald, M.E., Josep, F., Eric, M., 1996. Dimensional verification of NC machining profiles using extended quadtrees. Comput. Aided Des. 28 (11), 845–852.

Dhande, S.G., Karunakaran, K.P., 1994. Symbolic and computational conjugate geometry of a generic cutter and machine tool assembly. J. Des. Manuf. 4, 167–186.

Drysdale, R.L., Jerard, R.B., Schaudt, B., Hauck, K., 1989. Discrete simulation of NC machining. Algorithmica 4 (1), 33–60.

Feng, H.-Y., Menq, L.M., Chai, H., Hang, Z.L., 1995. The prediction of dimensional error for sculptured surface production using the ball-end milling process. Int. J. Mach. Tools Manuf. 35 (8), 1149–1169.

Fleisig, R.V., Spence, A.D., 2005. Techniques for accelerating B-rep based parallel machining simulation. Comput. Aided Des. 37, 1229–1240.

Fussell, B.K., Srinivasan, K., 1989. Online identification of end milling process parameters. Trans. ASME 111, 322–330.

Jackie, N., Tom, D., Mason, W., 1993. OpenGL Programming Guide. Addison-Wesley Inc.

Jerard, R.B., Hussaini, S.Z., Drysdale, R.L., Schaudt, B., 1989. Approximate methods for simulation and verification of numerically controlled machining programs. Visual Comput. 5, 329–348.

Jerard, R.B., Angleton, J.M., Drysdale, R.L., Su, P., 1990. The use of surface points sets for generation, simulation, verification and automatic correction of NC machining programs. In: SME Proceedings of NSF Design and Manufacturing Systems Conference, Tempe, AZ, January 8–12, pp. 143–148.

Karunakaran, K.P., Shanmuganathan, P.V., Gupta, N., Issac, M., 2000. Swept volume of a generic cutter. Proc. I MECH E Part B J. Eng. Manuf. 214 (10), 915–938.

Karunakaran, K.P., Shringi, R., 2007. Octree-to-BRep conversion for volumetric NC simulation. Int. J. Adv. Manuf. Technol. 32 (1–2), 116–131.

Kondo, M., 1994. Decomposition of complex geometry for a manufacturing application. Comput. Aided Des. 26 (3), 244–252.

Metal Carboloy Systems Department, 1980. Milling Handbook of High Efficiency Metal Cutting. General Electric Company.

Mortenson, M.E., 1985. Geometric Modeling. John Wiley & Sons.

Mounayri, H.E.I., Spence, A.D., Elbestawi, M.A., 1998. Milling process simulation – a generic solid modeller based paradigm. J. Manuf. Sci. Eng. Trans. ASME 120, 213–221.

Oliver, J.H., Goodman, E.D., 1990a. Direct dimensional NC verification. Comput. Aided Des. 22 (1), 3–10.

Oliver, J.H., Goodman, E.D., 1990b. Direct dimensional NC verification. Comput. Aided Des. 22 (1), 3–9.

Roth, D., Ismail, F., Bedi, S., 2003. Mechanistic modeling of the milling process using an adaptive depth buffer. Comput. Aided Des. 30 (8), 1–17.

Roy, U., Xu, Y., 1999. Computation of geometric model of a machined part from its NC machining program. Comput. Aided Des. 31, 401–411.

Spence, A., Altinas, Y., 1991. End milling force algorithms for cad systems. Ann. CIRP 40, 31–34.

Spence, A.D., Altinas, Y., 1994. A solid modeler based milling process simulation and planning system. Trans. ASME 116, 61–69.

Spence, A.D., Li, Z., 2001. Parallel processing for 2-1/2D machining simulation. In: Proceedings of the Sixth ACM Symposium on Solid Modeling and Applications. ACM Press, New York, pp. 140–148.

Sungertekin, V.A., Voelcker, H., 1986. Graphical simulation and automatic verification of NC machining programmes. In: International Conference on Robotics and Automation, pp. 156–165.

Tlusty, J., Smith, S., 1991. An overview of modeling and simulation of the milling process. Trans. ASME 113, 169–175.

Van Hook, T., 1986. Real time shaded NC milling display. Comput. Graphics (Proc. SIGGRAPH) 20 (4), 15–20.

Wang, W.P., Wang, K.K., 1986. Real time verification of multiaxis NC programs with raster graphics. In: Proceedings of IEEE, International Conference on Robotics and Automation, San Francisco, 7–10 April, pp. 166–171.

Wastra, W.H., Bronsvoort, W.F., Vergeest, J.S.M., 1994. Interactive simulation of robot milling for rapid shape prototyping. Comput. Graphics 18 (6), 861–871.