Reaction forces on a milling tool during three-axis milling

by

Daniel E. Shanahan

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Department: Mechanical Engineering

Major: Mechanical Engineering

Major Professor: James H. Oliver

Iowa State University
Ames, Iowa

1996

Graduate College
Iowa State University

This is to certify that the Master's thesis of

Daniel E. Shanahan

has met the thesis requirements of Iowa State University

Signatures have been redacted for privacy

TABLE OF CONTENTS

CHAPTER 1. INTRODUCTION

This thesis discusses a graphical numerically controlled (NC) milling simulation. Graphical simulations give a better feel for what happens during a complicated process, such as NC milling, than does numerical output from a mathematical model. NC milling simulations can be used to verify tool paths, detect collisions, and check the material removal rate, which can be related to force feedback on the milling tool. A graphical simulation should make calculations and update displays as quickly as possible, while still maintaining a reasonable level of accuracy.

The typical alternative to using a simulation is to perform a proofing run, in which the tool path is tested by milling a piece of wood or foam stock before producing the actual part [1]. Collisions and some tool path errors can be detected using this method, but slight misses are hard to detect. Also, the correct cutting forces resulting from material removal cannot be determined.

This thesis presents a real time NC milling simulation technique that incorporates a cutting force model to calculate forces generated between the milling tool and the workpiece. This information can be used to determine if a piece of the machinery could fail due to driving (feeding) the milling tool too rapidly, creating too large a force on the tool shaft or flutes.

1.1 Motivation

Simulation can be used to detect several kinds of milling errors. The tool path could contain errors that would cause the tool to mill the workpiece such that the surface is out of tolerance. A simulation can also be used to predict collisions between the tool and workpiece holding fixtures.

Another possible problem that can be encountered when milling is that the resultant forces on the milling tool from material removal could be too high. This could also be a reason to modify the tool path so that the tool is not cutting too deeply, creating forces it cannot handle. Another method to decrease the force on the milling tool is to decrease the feed rate of the tool. Material removal rates are the second most likely source of problems

during milling, after path error [2].

Although potentially very useful, previous cutting force simulation methods have not been coupled with real-time graphical simulation. Like in all numerical simulation techniques, researchers in tool force simulation are faced with the trade-off between computational speed and simulation accuracy. Most previous research in this field has focused on achieving numerical accuracy (as verified by experimental cutting with machine tools) at the expense of high computational overhead.

This research takes the alternative approach of incorporating a simplified cutting force model with a real-time graphical milling simulation method. By providing a user with qualitative force information (as opposed to precise quantitative results) and immediate visual feedback of the corresponding material removal process, it is hoped that many typical milling errors related to force may be detected and eliminated by an analyst, thus magnifying the benefit of the graphical simulation. Research in the area of spatial partitioning of solid models, previously used for milling simulation and tool path verification, is extended for use in force analysis on the milling tool.

1.2 Previous Research

Voelcker and Hunt approached the problem of NC milling verification and simulation by using direct solid modeling [3]. This approach performs a Boolean difference operation between a model of the workpiece and the volume swept by a 3-dimensional object, the milling tool, during rigid-body motion. With complex tool shapes and swept volumes, this approach typically results in calculations that are too cumbersome for real time simulations. This method generally does not quantify milling error in terms of gouges and misses, but rather produces a solid model of the "as-milled" part. Further additional Boolean operations between this model and one representing the actual desired design would be necessary to characterize milling errors.

In an effort to address the high computational expense of this approach, Wang and Wang [4] introduced image space Boolean operations. This approach scan converts polygonal models of the tool swept volumes and the workpiece and computes the boolean

operations using a z-buffer in image space. Sambandan and Wang [5] generalize this method by introducing a polygonal representation of 5-axis milling tool swept volumes.

Oliver and Goodman [1] and Jerard, et al. [6] developed a method for dimensional verification involving the intersection of swept volumes with vectors projecting from the desired part surface. These techniques generally divide the problem into three tasks: 1) create a distribution of surface points and corresponding vectors to approximate the desired part surface, 2) determine which points and vectors could feasibly intersect each swept volume, and finally, 3) intersect the vectors with the swept volume. These methods are computationally faster than direct solid modeling, but cannot be used for NC milling simulations because they do not produce a model of the milled surface. They are limited to dimensional verification. Narvekar, et al. [7] generalized this approach by incorporating precise 5-axis milling tool swept volumes.

None of these approaches can be used for force analysis because they do not give the detailed information about the location of the tool at each instance in time for a small time step. That information is required for determining forces.

Van Hook developed the *dexel data structure* for use in shaded real time NC milling simulation [8]. Chapter 2 explains this data structure in detail. Van Hook mentions extending the dexel data structure to include information about the type of object and the shape of the object used to create the dexel. Huang [9] uses the same model, and includes information about the type of object used to create the dexel, for simulation and dimensional verification of the tool path. His method does not use swept volumes, but instead subtracts the tool from the workpiece at finite instances of motion. He also implements a method to convert the dexel display back into a form in which it can be viewed from any angle. Such an extension to the dexel display was mentioned by Tim van Hook as an area that needed work. It is not implemented in this thesis. Use of the dexel data structure is the only method mentioned that can be extended for force analysis during real-time NC milling simulation.

A common method used to calculate milling forces is to model chip geometry for part of one flute on the milling tool [10]. The tangential force on the milling tool is related to the axial depth of cut, the chip thickness, and an empirical constant. (As shown in Figure 1,

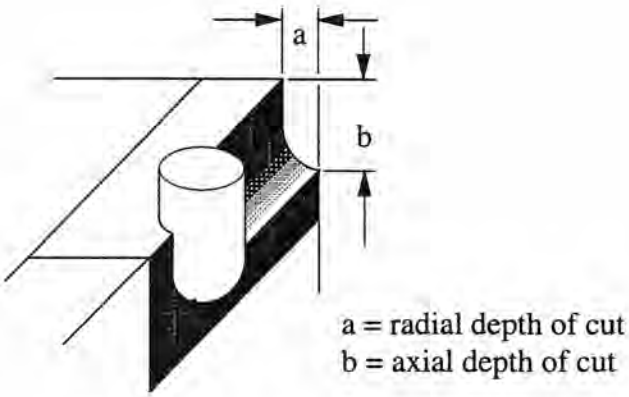


Figure 1 Radial and axial depth of cut

the axial depth of cut, *b,* is the distance the tool penetrates the workpiece in the axial direction.) The radial force is related to the tangential force by a constant determined by experiment.

Tlusty and MacNeil [11] use elemental forms of these equations, dividing them into ranges where the flutes are partially engaged, and a range where they are fully engaged. The forces are determined by integrating the equations over all of these ranges. This method can be used for milling at a constant radial and axial depth of cut, or for *transient milling*, where the tool is entering the workpiece at a perpendicular.

DeVor et al. use the same model and include cutter deflection [12]. Deflection is calculated from cutter geometry and from the cutter material stiffness. Instead of integrating over separate ranges to determine the total force, the tool is split up into layers and the range of cutting angles in each layer is determined. Forces are summed over all layers.

Sutherland and DeVor use the same basic formulas used by Tlusty and MacNeil, which are referred to as a *rigid system*, again splitting the tool up into layers, and further develop a flexible end milling system with cutter runout [13]. They compare the different systems to measured force values for constant spindle speed, feed rate, and radial and axial depth of cut. This comparison shows that the flexible end milling system with cutter runout is an improvement over the rigid system. This is a *static model* because the deflection at each instance depends only on the force at that instance, and does not affect the force calculations [3].

According to Smith and Tlusty, a *dynamic model* must take into consideration not only the feed rate and chip thickness, but also the deflection of the cutter and the surface generated by the previous teeth [3]. Tlusty and Ismail develop such a model, the *Regenerative Force, Dynamic Deflection Model*, with a flexible cutter that has multiple degrees of freedom in two perpendicular axes. The axes lie on the plane perpendicular to the tool axis. Instead of computing deflections, accelerations are computed for this model by considering mass, damping, and stiffness of the cutter, and deflections are then produced by integrating twice. Although this model is more accurate than previous models, it is still an approximation because only past cutter deflections are considered in force calculations, and current deflections need to be considered as well [14].

Sutherland uses the same principals in the development of his dynamic model for calculation of cutting forces [15]. This model incorporates feedback by examining past and future tool deflections and chip thicknesses to determine the cutting force. This model is again shown to be an improvement over the rigid model when compared with experimental data, but again the spindle speed, feed rate, and radial and axial depth of cut are held constant for the comparison.

Another method for determining the average, rigid, static deflection cutting forces uses the metal removal rate to determine the average power requirements for cutting a chip from the workpiece [3][16]. According to Smith and Tlusty, this method does not produce reasonable results because it assumes that from an average force, an average tool deflection can be calculated, and from this, the surface generated by the cutter on the workpiece is calculated [3]. Tool deflection depends on the instantaneous force. The calculated surface will not be correct unless instantaneous deflections are examined. Therefore, this method cannot be extended beyond the rigid model, whereas the model that uses chip geometry can be extended.

Yellowley uses the rigid model, along with values for the average forces and torque, to derive a Fourier series to predict cutting forces [17]. This type of method for predicting forces is not used in this thesis because it does not make use of the instantaneous force information that is available when using dexel models to simulate milling.

1.3 Overview

The algorithm described in this thesis works as follows: The workpiece and milling tool are converted into a specialized geometric representation according to a method previously used by van Hook [8] and Huang [9]. At each tool location, the dexel model gives detailed information indicating areas on the flutes of the tool that are currently cutting material. From this, forces are calculated from a chip load model that is the basis for more accurate models for determining forces developed by Tlusty and McNiel [11], DeVor et al. [12], Sutherland and DeVor [13], Tlusty and Ismail [14], and Sutherland [15].

Chapter 2 briefly describes the geometric technique used to facilitate NC simulation. The dexel data structure is introduced and its relationship to force analysis is described.

Chapter 3 explains how forces are modeled and how the algorithm uses dexel information to determine the forces.

Chapter 4 explains an implementation and demonstrates how the results depend on the implementation. Results from this program are compared with results from a paper published by Tlusty and McNeil [11].

Chapter 5 draws conclusions about the program and discusses future work.

CHAPTER 2. GEOMETRIC REPRESENTATION

In NC milling, a rotating cutter is moved along trajectories in the vicinity of the workpiece to affect material removal. To simulate this behavior using geometric models one must compute the Boolean difference (intersection volume) between the workpiece and the volume swept by the moving tool. This is generally a difficult computation if traditional solid modeling techniques are applied, particularly when considering the real-time requirements of NC simulation. Previous researchers [8][9] have developed a specialized geometric form called the dexel representation which facilitates real-time Boolean operations for NC simulation. The same approach is adopted in this research for the same reasons. The dexel representation is a regularized spacial decomposition of a solid model. It is easy to compute, and it facilitates incorporation of a simple force model.

2.1 Dexel Model Creation

Any geometric model can be converted to a dexel representation given that a robust method for computing the intersection of a ray with the solid exists. The steps for implementation of the conversion method used in this thesis are the same as those used by Huang [9].

First, the objects to be converted are projected onto the viewing plane (Figure 2). Next, a bounding box is created to enclose the projection (Figure 3). At evenly spaced locations inside this bounding box, sight lines aimed in the viewing direction (the negative z-direction) are intersected with the object (Figure 4). Where a sight line intersects a solid object, it creates a dexel (Figure 5). The z-coordinate at the locations where the sight lines enter an object and where they exit are stored in the *dexel data structure* as *near location* and *far location* values, respectively. The surface normals at these locations are used to calculate color map indices, which are stored as *near color* and *far color* values [8].

When the object to dexel conversion is complete, the object will be represented by a neatly stacked pile of rectangular prisms (the dexels), most of which are relatively long in the sight line direction (negative z-direction).

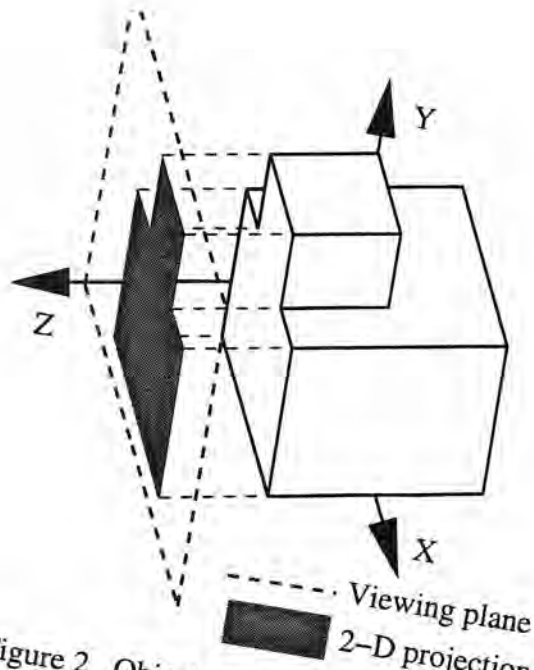


Figure 2 Object projected onto viewing plane

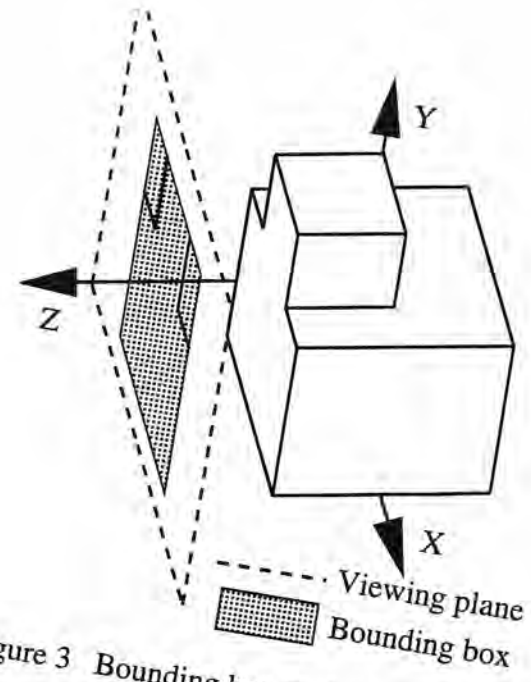


Figure 3 Bounding box encloses projection

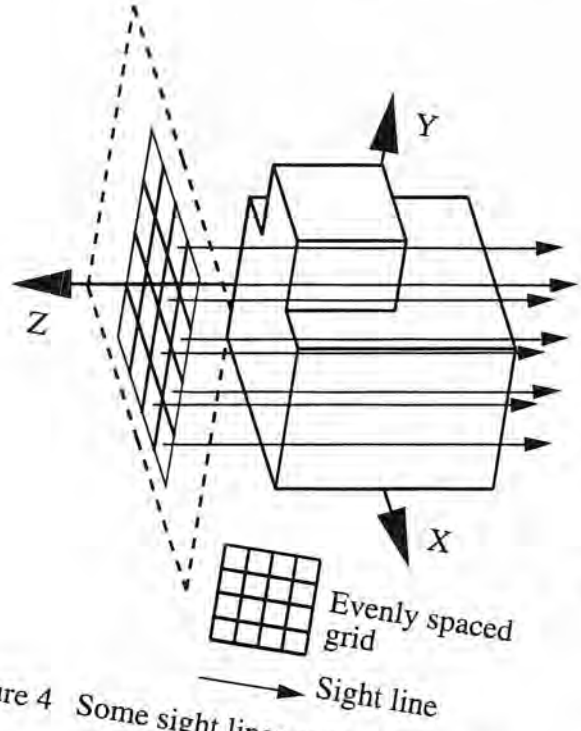


Figure 4 Some sight lines at evenly spaced locations

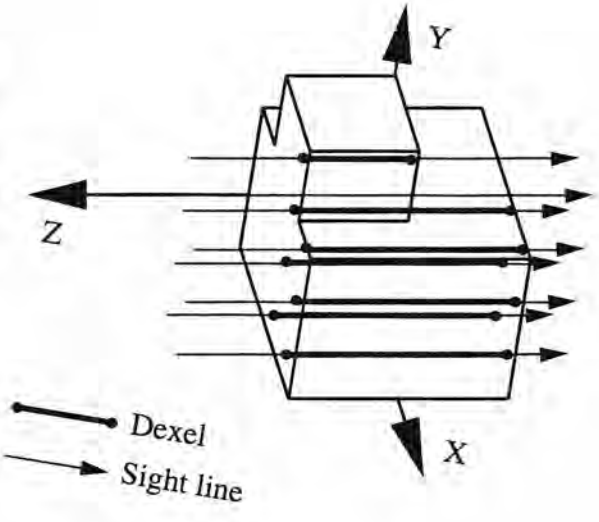


Figure 5 Dexels exist where sight lines intersect the object

The *dexel location* is given by the (x, y) coordinates at the center of the rectangular prism faces that lie in the viewing plane. These square, planar faces are located at the near and far location values in the z-direction.

In an NC milling simulation, dexels are created only from sight lines intersecting with specific shapes. Those shapes are: planar blocks, when sight lines hit the workpiece, or spheres, cylinders, cones and tori, when sight lines hit the tool. Huang describes the details behind the calculation of these intersections quite well [9].

All dexels have the same width (x-dimension) and the same height (y-dimension). In this application, dexel height and width are equal to the length of the edge of one pixel. All dexels are contained in a bounding box (Figure 3). Each row in the bounding box contains the same number of dexels as all other rows, and each column contains the same number of dexels as all other columns. Therefore, each dexel x- and y-location can be determined by its column number and row number respectively, and these are determined by one index in an array. In this application, the row number is equal to the dexel index divided by the number of columns of dexels in the bounding box. The column number is equal to the modulus of the dexel index and the number of columns.

All dexel information is stored in a *dexel data structure* (Figure 6). The data structure contains the *near location* and *far location* of the dexel. The dexel near location is the location where a sight line enters the object in the negative z-direction, and the dexel far location indicates where a sight line exits the object. The data structure contains a color map index for the color at the near location as the *near color*, and one for the color at the far location as the *far color*. The color map index indicates which color to use within a chosen color map. Another variable in the data structure, the *type of dexel,* determines which color map to use when coloring the dexel. The type of object the sight line intersects when creating the dexel determines the type of dexel (for example: tool, workpiece, fixture are possible types). If the sight line does not intersect anything, then the type of dexel is set to "no dexel," or the near location can be set to the farthest allowable z-location: the background. In this case, no other values in the dexel data structure are important. The data structure also contains a pointer to the *next dexel*. The next dexel is another dexel created by

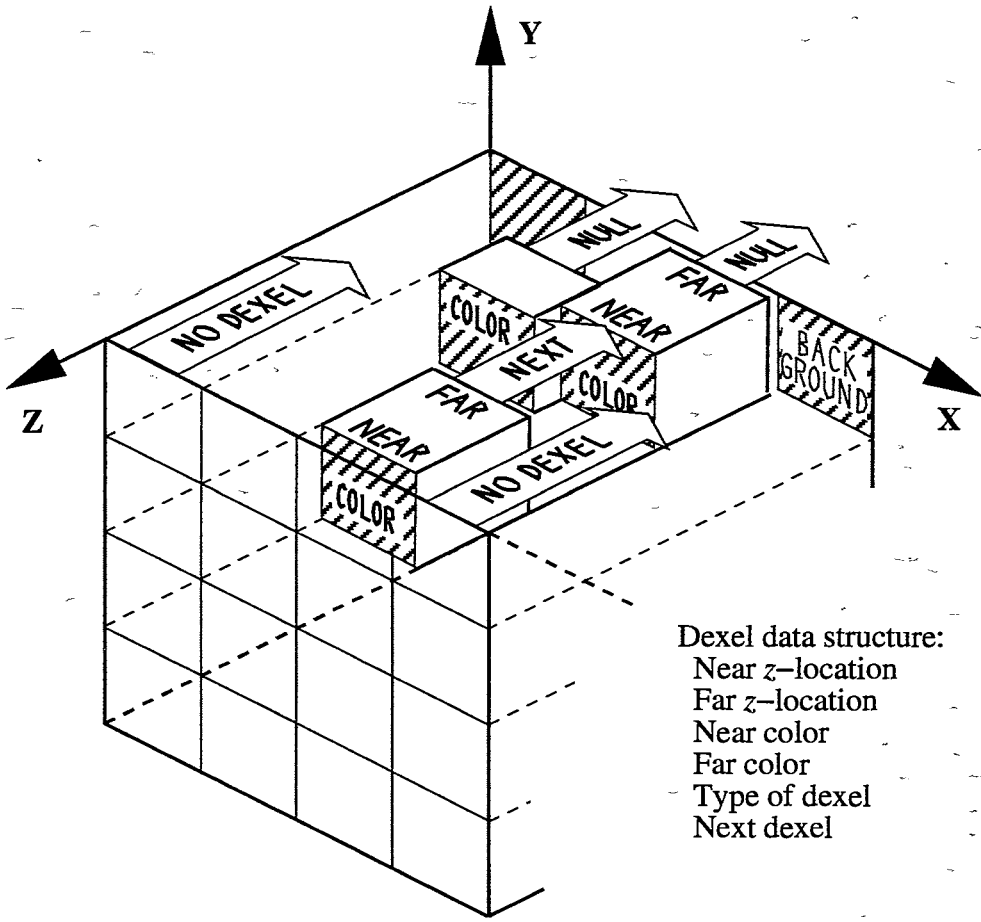


Figure 6 Dexel data structure [8]

the same sight line as the current dexel. If there is no other dexel created by this sight line, the next dexel is a null pointer.

2.2 Motivation for Use of Dexels

For each dexel, the *x*- and *y*-dimensions of the face of the dexel are the same. Only the *z*-dimension, the depth of the dexel, varies. Therefore, they can be considered one-dimensional line segments, parallel to the *z*-axis [8]. Dexels exist only at discrete, evenly spaced *x*- and *y*-locations.

During milling simulation, the intersection between the milling tool trajectories and the workpiece must be calculated precisely. Dexels are used to speed up intersection calculations. When objects have been converted to dexels, intersection calculation

algorithms do not have to deal with all the special data structures and cases resulting from intersecting two solids of general shapes. Instead, groups of dexels are intersected. Intersecting two dexels is the same as intersecting two parallel line segments. Only if the x- and y-values for the dexels are the same can they intersect. Therefore the algorithm only has to compare z-values to determine the intersection. There are only 8 possible types of situations to consider (Figure 7). The method for determining tool and part intersection, using van Hook's dexel data structure, is as follows [8][9]:

```
tool_dexel_index = 0                                    start with the 1st tool dexel
while tool_dexel_index < number_of_tool_dexels          search through each tool dexel
    if  workpiece_near_z = background                   case 1: no workpiece dexel
                                                        therefore, no intersection
    else if tool_near_z = background                    case 2: no tool dexel
                                                        therefore, no intersection
    else if tool_near_z < workpiece_near_z
        if tool_near_z < workpiece_far_z                case 3: tool behind workpiece
                                                        therefore, no intersection
        else if tool_far_z < workpiece_far_z            case 4: front of tool hits back of part
            workpiece_far_z = tool_near_z               update workpiece
        else tool_far_z > workpiece_far_z               case 5: tool splits workpiece
            workpiece_far_z = tool_near_z               update workpiece
            new workpiece_near_z = tool_far_z           create new workpiece dexel
            new workpiece_far_z = workpiece_far_z
    else tool_near_z > workpiece_near_z
        if tool_far_z > workpiece_near_z                case 6: tool in front of workpiece
                                                        therefore, no intersection
        else if tool_far_z > workpiece_far_z            case 7: back of tool hits front of part
            workpiece_near_z = tool_far_z               update workpiece
        else tool_far_z < workpiece_far_z
            workpiece_near_z = background               case 8: tool deletes workpiece
    increment tool_dexel_index                          go to the next tool dexel location
```

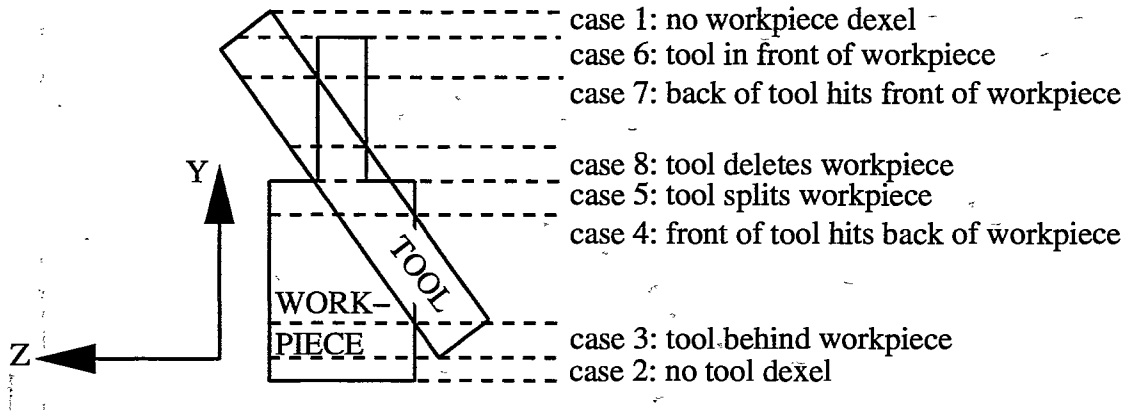


Figure 7 Eight types of intersections [8]

Huang does not use swept volumes to calculate these intersections. Instead intersections are calculated at each tool location as the tool moves through minute motion instances, approximating a swept volume [9].

For force analysis, dexels are convenient because at each instance in time, they provide information about where the simplified tool model and the workpiece are in contact with each other. This is not true for swept volumes, used by Narvekar, et al. [7]. The contact information is needed for determining how the forces are applied to the tool flutes, as described in the next chapter.

2.3 Additional Dexel Information Required for Force Analysis

In this simulation, milling can be done with a flat-end tool or a ball-end tool. The simplified ball end tool model used for determining intersections consists of a cylinder and a half sphere. Although flutes do not need to be modeled for determining intersections or for the milling simulation, they are required for determining the magnitude and direction of the tool forces. If the area of the flute being examined is on the cylindrical part of the tool, then the distance from the central axis of the tool to the cutting edge of the tool flute is equal to the tool radius. If it is on the spherically shaped part of the tool, then this distance varies. If there is contact between the bottom end of a flat end tool and the workpiece, this contact is ignored for force analysis. When creating dexels to represent the milling tool, the local shape must be recorded in the dexel data structure so that the cutting edge can be located, and forces computed. This shape information is included in the *type of dexel* field of the data structure.

CHAPTER 3. FORCE ANALYSIS

In milling, the cutter shank can break due to an excessive force or bending moment on the tool. If the instances of highest force or moment on the tool can be predicted before milling, then the tool path, spindle speed, or feed rate can be modified to avoid forces that are too high. The computer program discussed in this thesis attempts to make an approximation of the reaction forces during three-axis milling by using a model that approximates the chips of material removed by the milling tool. To keep calculations simple, deflections and vibrations in the cutter are neglected.

3.1 Chip Load Model

The overall goal of this graphical NC simulation is to be as close to real time as possible. Therefore, force calculations are kept simple. The formulas for computing tangential and radial cutting forces on a small part of one flute are approximated by:

$$F_t = K_t b t_c \tag{1}$$

$$F_r = K_r F_t \tag{2}$$

where F_t is the tangential force, F_r is the radial force, b is the axial depth of a small part of the cut, t_c is the chip thickness (Figure 8), and K_t and K_r are constants that are determined by experiment [11]. K_t is dependent on the material of the workpiece, the tool shape, and the chip thickness. Tlusty and MacNiel use $K_r = 0.3$ [11]. DeVor et al. show that K_r depends on cut geometry [12]. There is also an axial force on the tool, but according to Sutherland, it is relatively small and does not contribute much to the bending moment [15]. Chip thickness in

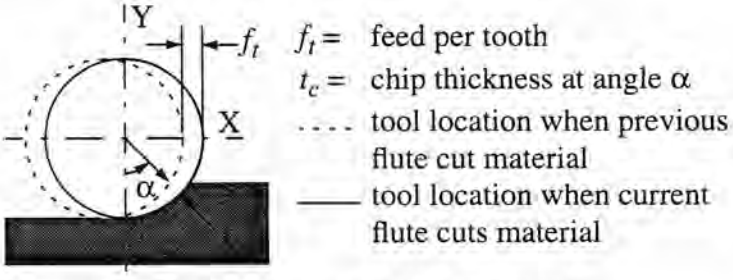


Figure 8 Chip thickness and feed per tooth

equation (1) is approximated by:

$$t_c = f_t \sin\alpha \tag{3}$$

where f_t is the feed per tooth (Figure 8), and α is the angle to the flute of interest. This angle is measured about the tool axis, starting from the perpendicular to the tool feed direction [18]. and Figure 9 show how α is measured. Equation (3) is an approximation because it assumes the flutes move in a circular path. This is a good approximation when the feed rate per tooth is much smaller than the radius [13]. The feed per tooth is calculated by:

$$f_t = \frac{V_f}{(N_s N_f)} \tag{4}$$

where V_f is the tool feed rate, N_s is the angular rotation or spindle speed, and N_f is the number of flutes (Figure 9).

3.1.1 Computing Forces by Summation

Equations (1) and (2) can be written in terms of a fixed, tool centered coordinate system,

$$dF_x(i,j) = \sum_{k=1}^{N_f} -dF_r(i,j,k)\sin\alpha(i,j,k) + dF_t(i,j,k)\cos\alpha(i,j,k) \tag{5}$$

$$dF_y(i,j) = \sum_{k=1}^{N_f} dF_t(i,j,k)\cos\alpha(i,j,k) + dF_r(i,j,k)\sin\alpha(i,j,k) \tag{6}$$

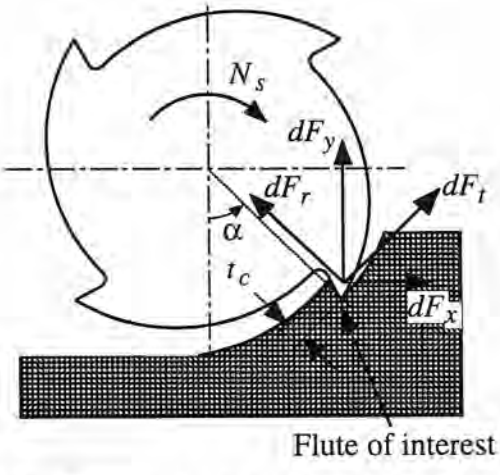


Figure 9 Elemental cutting forces on a flute [15]

where (*i, j, k*) refers to i^{th} axial element of the tool, at the j^{th} position or instant in time, for the k^{th} flute. For the program implemented in this research, total forces are calculated by summing equations (5) and (6) over *i* and *k*, for each instant in time, *j*.

The *axial elements* are referred to as *tool layers*. In this application, the tool layer size used for determination of the forces is based on the tool orientation and the dexel size, described in section 3.2.3. The tool is converted into dexels before milling begins, and it is only during milling that forces are calculated.

Tlusty and McNeil calculate forces by integrating equations (5) and (6) [11]. They are able to use integration because their examples have a constant radial and axial depth of cut, and constant feed per tooth. Therefore, as the tool is rotating, the only variable is the angle to the flute of interest, α. For this research, none of these is required to be constant.

3.1.2 Computing Forces by Integration

The method presented in this section is used by Tlusty and McNeil [11]. Although their method is used to generate a total of four sets of equations used to calculate forces on the tool for any angle α, only one particular set of their formulas is presented. The set of formulas chosen for discussion is the set that is used when the flute is completely engaged with the workpiece, producing the maximum forces on the tool. This set was chosen because the maximum forces are of most interest when attempting to prevent tool failure.

If there is a range of angular orientations for the tool, about the tool axis, for which the entire flute embedded in the material is cutting over the entire axial depth of cut (Figure 1), then the maximum force occurs at the largest angle α (Figure 9). This situation always occurs for some value of α when the engagement angle, δ, is smaller than the difference between the angles where flutes enter and exit the material (Figure 10). In Figure 10, this difference is (ϕ - 0), or ϕ. When δ is larger than the difference, this situation cannot occur.

The formulas used by Tlusty and McNeil [11] to determine the force components when δ is smaller than the difference, determined by integrating equations (5) and (6), are:

$$F_x = -F_u\left[\sin^2\alpha - \sin\ (\alpha-\delta)\ + 0.3\delta + 0.15\sin 2\,(\alpha-\delta) - 0.15\sin 2\alpha\right] \qquad (7)$$

$$F_y = F_u\left[0.3\sin\ (\alpha-\delta) - 0.3\sin^2\alpha + \delta + 0.5\sin 2\,(\alpha-\delta) - 0.5\sin 2\alpha\right] \qquad (8)$$

where F_x and F_y are the force components in the x- and y-directions, and F_u is defined by:

$$F_u = 0.5K_t s_t \frac{r}{\tan\beta} \tag{9}$$

where s_t is the feed per tooth, r is the tool cutter radius, β is the helix angle, and a value of K_r=0.3 has been used. Similar equations for the cases in which a flute has partially entered or partially exited the material, or the engagement angle, δ, is greater than the difference between the angle where flutes enter and exit the material, are all described in [11].

Figure 10 shows the range of angles where flutes are cutting. Measured from the negative y-axis, the angles where areas of flutes can cut the material range from 0 to ϕ. It is easier to see what parts of the flutes are cutting if the tool is unrolled, as in Figure 10c. In this figure, it can be seen that the angular measurements to areas on the cutting flute depend on the axial location on the tool. These angles range from (α - δ) at the bottom of the tool, to α at highest part of the tool that is cutting material, which is located at a distance b from the bottom of the tool.

3.1.3 Model Limitations

Sutherland and DeVor compared the chip load model used for this research, and a flexible end mill model that he proposed as an improvement on the chip load model, to experimental data [13]. Maximum forces in his improved model correlated more closely with experimental data. The simple chip load model does not take torsional and lateral cutter deflections and vibrations into account. Because of this simplification, the model is not accurate for long cutters. Sutherland showed that for long cutters (5.250 inches long, and 0.75 inch diameter in his example), the maximum force may be overestimated in excess of 100 percent when using the simple chip load model of Tlusty and MacNiel [11] used in this research.

3.2 Force Calculation

To get the forces on the tool, the areas on the tool where the flutes are cutting material and the angles to these flutes must be calculated, along with the feed rate per tooth. Other variables, such as K_t and K_r from equations (1) and (2) and the total axial depth of cut, b, are predetermined.

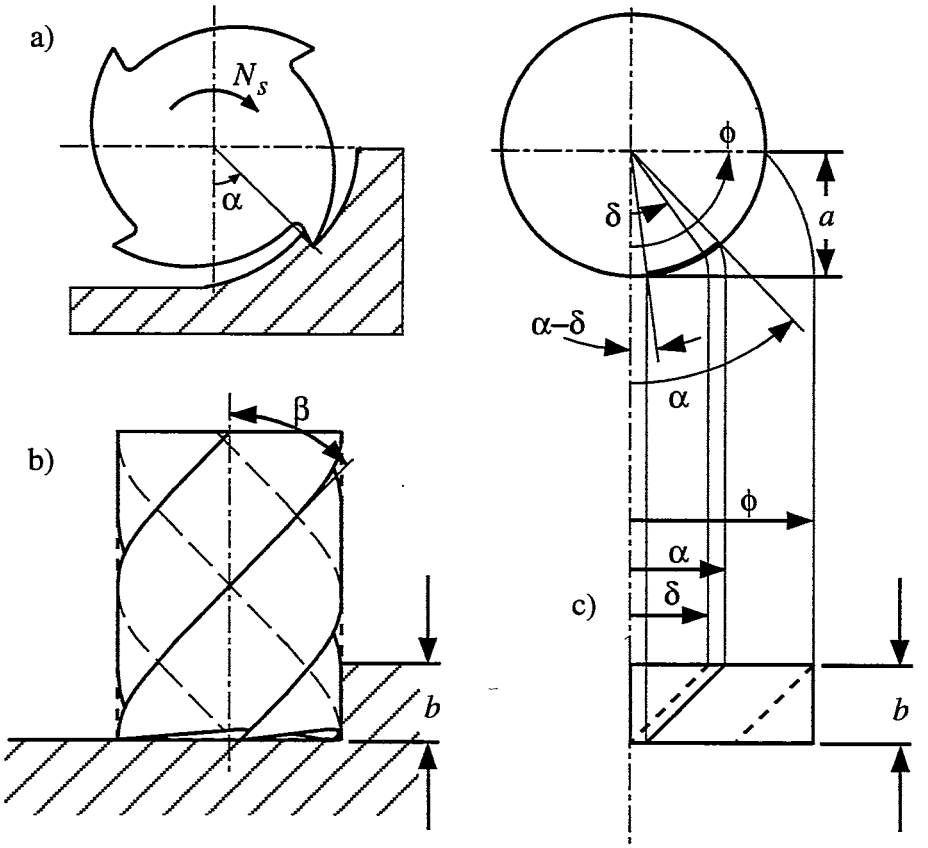


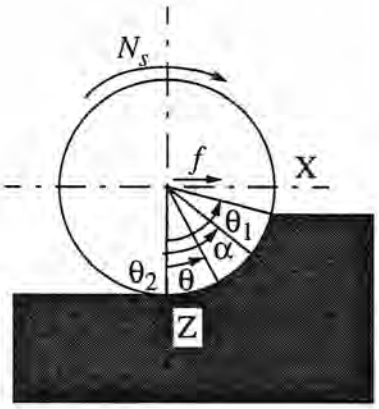
Figure 10 Angles used in equations (7), (8), and (9); a) top view of tool; b) side view of tool; c) unrolled section of tool where flutes may be cutting the workpiece [11].

3.2.1 Determining Cutting Angles

To use equations (5) and (6), the algorithm must determine which areas on which flutes are currently cutting. This is done by comparing the angular range where the tool contacts the workpiece in each layer of the tool (Figure 11 angle θ_1 to θ_2) to the angles to the flute locations in the same layer (Figure 9 and Figure 11; angle α).

Dexel information can be used to determine approximately what parts of the flutes are cutting at any instant in time. In this research, a new instance occurs when the screen is updated with a new tool location. During the milling simulation, the tool is subtracted from the part at each instance. The near and/or far locations (depending on which part of the tool dexel did the cutting) are stored in Cartesian coordinates. These coordinates are used to determine an angle θ (Figure 11) according to:

$$\theta = \text{atan}\left(\frac{\Delta x}{\Delta z}\right) \tag{10}$$



y-axis is perpendicular to the page
f = feed rate and direction
N_s = spindle speed and direction
α = flute cutting angle
θ = angle to a dexel near or far location contacting workpiece
θ_1 = angle where flute begins contact with workpiece
θ_2 = angle where flute ends contact with workpiece

Figure 11 Angle θ used to determine where tool contacts part

This angular location, θ, measured about the tool axis to the near or far location of a tool dexel that has cut a workpiece dexel, is called a *cutting angle*. In equation (10), Δx and Δz are determined by:

$$\Delta x = x_d - x_t(y_d) \tag{11}$$

$$\Delta z = z_d - z_t(y_d) \tag{12}$$

where x_d, y_d, and z_d are the Cartesian coordinates of a tool dexel near or far location, and $x_t(y_d)$ and $z_t(y_d)$ are the corresponding x- and z-Cartesian coordinates of the tool axis at the same y_d value where x_d and z_d are located (Figure 12a). From this information, all angles where the tool is cutting the part are determined. These angles are compared to the flute angles, α, for all layers of the tool that are cutting the workpiece. The tool is assumed to be aligned this the y-axis in the derivation of equations (10) through (12).

3.2.2 Orienting the Tool

Up to this point in this thesis it has been assumed that the tool axis is aligned with the y-axis in order to simplify equations and figures. In general, the tool axis may be aligned with either the x- or y-axis to perform any of these calculations, but the equations must be adjusted accordingly.

Before making the comparison between flute angles and angles where tool and workpiece dexels contact each other, one other problem has to be resolved. When the tool is

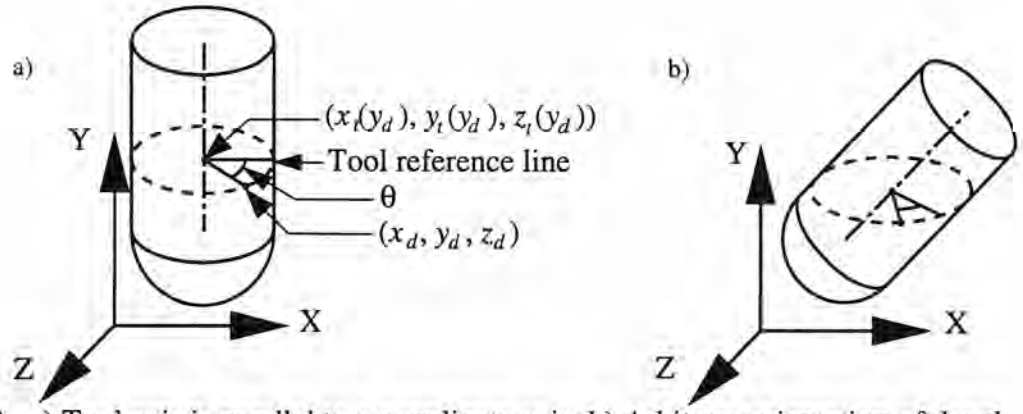


Figure 12 a) Tool axis is parallel to a coordinate axis; b) Arbitrary orientation of dexel representation of the tool

converted to dexels, the tool axis can be at any arbitrary angle with respect to the planes containing the sight lines used to create the dexels (Figure 12b). The sight lines are always in the viewing direction, which is the negative z-direction. For the cutting angles, θ, and the flute angles, α, to have any meaning with respect to each other, they must be measured on the same plane, which must be perpendicular to the tool axis because that is the plane on which the chips are created, and the chip thickness, t_c, in equations (1) and (3), is measured. This is the alignment shown in Figure 12a.

Flute angles are known: an initial value is assigned at some position along the tool axis, the angles at other positions are determined by the helix angle. These values are incremented when the screen is updated based on tool spindle speed.

It is easiest to determine the cutting angle, θ, by using equation (10). This requires that the tool be oriented so the axis is parallel to a coordinate axis (Figure 12a) and θ can be calculated about the global y-axis (or x-axis) using two Cartesian coordinates, as in equation (10).

For three-axis milling, the transformation process to orient the tool is simple. In the computer program discussed in this thesis, the tool and workpiece are initially generated so the tool axis is in alignment with the z-axis (Figure 12a). Before the milling simulation begins, the user may orient the tool by performing scaling operations, rotations about any coordinate axis, and translations in the x- and y-directions. The product of all of these

transformations is stored as the *orientation matrix*. This transformation can change the tool axis direction and the feed direction. While the tool is in this orientation, the conversion to dexels takes place. In this orientation, equations (11) and (12) cannot be used to determine Δx and Δz because the tool axis is not perpendicular to a coordinate axis: the cutting angle θ determined in this orientation is meaningless (Figure 12b). To position the tool in a useful orientation, it is necessary to realign the tool with a coordinate axis. To accomplish this, the inverse of the orientation matrix is used. During milling, the tool axis direction never changes, so the orientation matrix is only calculated once.

3.2.3 Discrete Tool Layers

When the tool is converted to dexels, the dexel near and far locations lie along an evenly spaced grid of rows and columns, but the tool may be in an orientation such that the tool axis is not parallel to a coordinate axis. After the transformation to properly orient the tool axis for determination of the cutting angles, θ, the transformed dexel near and far locations will no longer lie on an evenly spaced grid with layers parallel to the planes containing the sight lines (The evenly spaced grid is shown in Figures 4 and 6) These transformed values must be separated into layers so a beginning and ending cutting angle can be found for each tool layer. Figure 13 displays the relationship between dexel discretization and tool layer discretization.

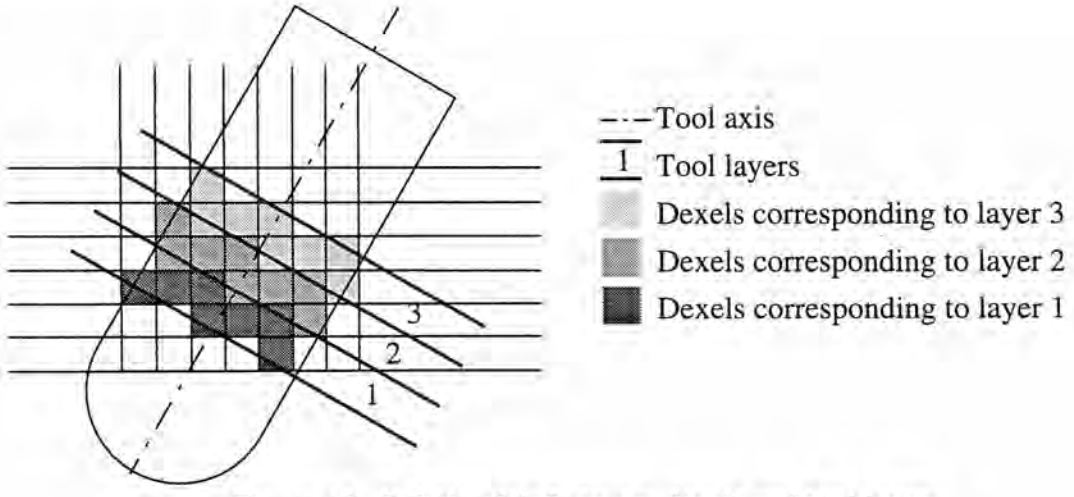


Figure 13 Relationship between dexels and tool layers

Figure 14 shows how the layer size is determined after the tool is properly oriented for using equations (11) and (12) to calculate θ. In Figure 14a, the tool is shown in its position during dexel creation. Figure 14b illustrates how the layer size depends on tool orientation during dexel creation. The thickness of these layers is related to the width of a dexel according to:

$$T = \frac{w_d}{\cos\theta_d} \tag{13}$$

where T is the thickness of a tool layer, w_d is the width of a dexel, and θ_d is the angle between the tool axis direction and the coordinate axis with which it makes the smallest angle. In Figure 15a, θ_d is measured from the y-axis, and in (b) it is measured from the x-axis. The smaller angle is used in order to create a more finely discretized layering of the tool, which will result in more accurate axial depth of cut calculations.

All cutting angles created by dexels that fall in the same layer are grouped together.

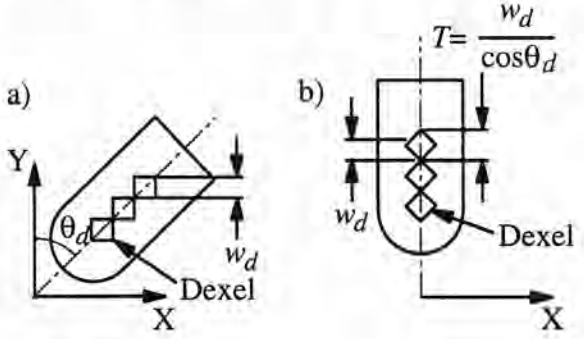


Figure 14 Determining layer size

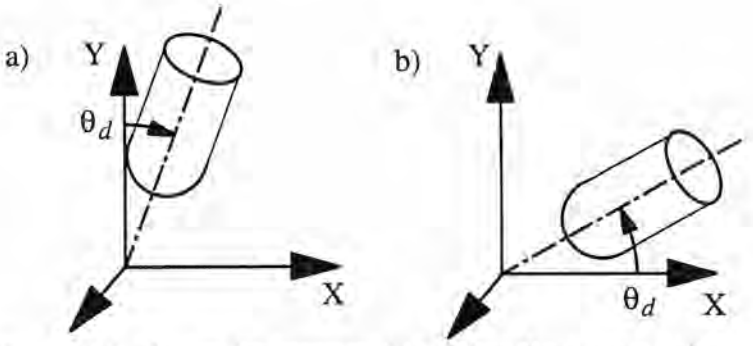


Figure 15 θ_d can be measured to either the x- or y-axis

From these groups, the beginning and ending cutting angles are found for the tool layer. Then the flute angles, α, in each tool layer can be compared with the beginning and ending cutting angles, θ_1 and θ_2, in each layer, to determine whether or not they are cutting the workpiece.

3.2.4 Adjusting Beginning and Ending Cutting Angles

The beginning and ending cutting angles are determined from the angles measured about the tool axis to the near and far dexel locations, using equation (10). In most cases these angles are a close approximation to the actual beginning and ending cutting angles because dexels are small with respect to the tool. Figure 16a shows a case where the beginning cutting angle, θ_1, needs adjustment. In this case the orientation of the dexels does

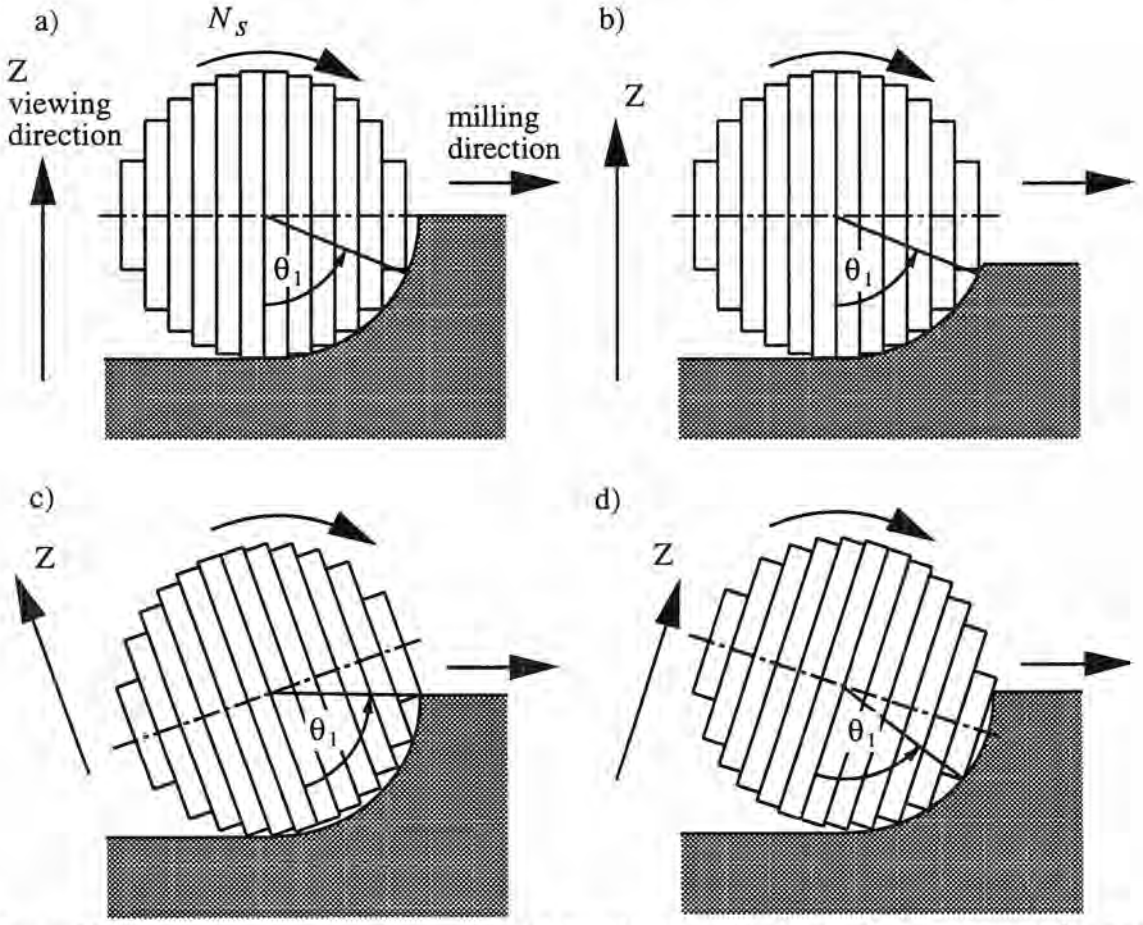


Figure 16 Examples showing layers of tool dexels where the beginning cutting angles need different amounts of adjustment

not allow for close approximation of the cutting angles because the tangent to the tool surface at the point of cutter contact is parallel or nearly parallel to the direction of the sight lines used to create the dexels. This is the case with the dexels on the left and right edge of the tool in Figure 16a. In Figure 16a, the cutting angle is underestimated. The required adjustment depends on workpiece geometry, sight line direction for dexel creation with respect to tool orientation, and tool movement direction as well. Figure 16b shows a case where θ_1 does not need adjustment because the workpiece geometry is different. Figure 16c shows a case where θ_1 does not need adjustment because the direction of the sight lines used to create the dexels is different. Figure 16d shows a case where θ_1 needs more adjustment than (a) because the direction of the sight lines is different.

Figure 17a shows a slice of the tool in the arbitrary orientation of dexel creation. The figure displays the linear dimensions used to determine cutting angles. The precise amount of adjustment required for the beginning and ending cutting angles cannot be determined from the available information, but the maximum amount of error without adjustment can be determined. The maximum amount of error is given by $2\theta_a$, where θ_a, as shown in Figure 17b, is determined by:

$$\theta_a = \operatorname{atan}\left(\frac{\left(\frac{1}{2}d_d\right)(\sin\theta_{dz})}{\left(r - \frac{1}{2}w_d\right)(\sin\theta_{dx})} \right) \tag{14}$$

where w_d is the dexel width, d_d is the dexel depth, θ_{dx} is the angle from the projection of the tool axis onto the xy-plane to the x-axis, and θ_{dz} is the angle from the projection of the tool axis onto the yz-plane to the z-axis. The dexel depth is determined from the Pythagorean Theorem, using the tool radius, r, and w_d, as follows:

$$\left(r - \frac{1}{2}w_d\right)^2 + \left(\frac{1}{2}d_d\right)^2 = r^2 \tag{15}$$

Notice that θ_a shown in (a) cannot be used for the same reason θ shown in Figure 12b cannot be determined in the tool orientation shown: θ_a must be determined on a plane perpendicular to the tool axis. This has been taken into account in equation (14) by using θ_{dx} and θ_{dz}.

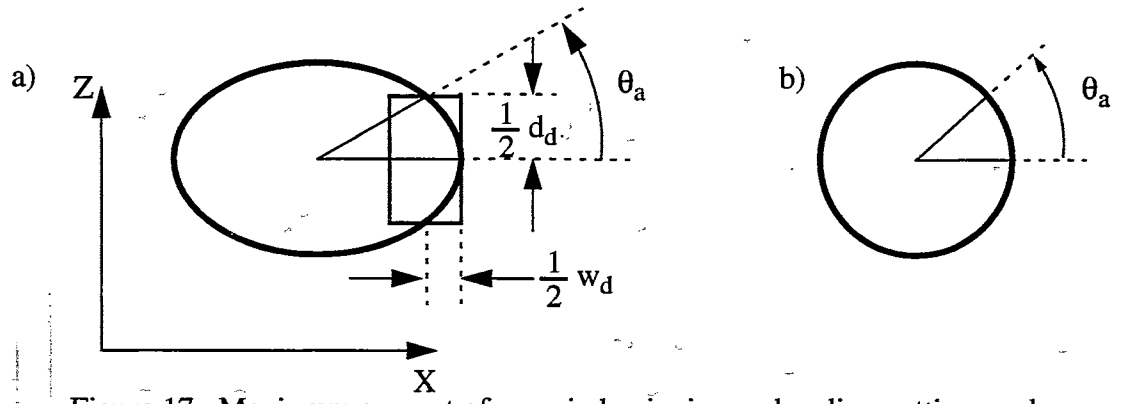


Figure 17 Maximum amount of error in beginning and ending cutting angles

In all the cases tested in Chapter 4, adding or subtracting θ_a from the beginning or ending cutting angles made negligible difference in the force calculations. However, this error could become significant if the dexel to tool size ratio is large.

3.2.5 Determining the Forces and Moments

Equations (5) and (6) are used to determine the forces on the tool. These forces can be used to determine the moments in regions where the tool is most likely to fail.

One likely mode of failure is the tool shaft shearing about its axis. Checking for this form of failure involves calculating the moment about the tool axis (the z-axis). An elemental form of the moment is calculated by:

$$dT_z = rdF_x + rdF_y \tag{16}$$

where dT_z is the moment about the z-axis, and r is the tool radius.

Another possible type of failure is due to bending at the location where the tool is held in the milling machine. Checking for this involves calculating the moment at this location, perpendicular to the tool axis (about an axis in the xy-plane). The elemental form for this moment is:

$$dT_{xy} = (l-z)\,dF_x + (l-z)\,dF_y \tag{17}$$

where dT_{xy} is the moment about an axis in the xy-plane, l is the tool length measured along the tool axis from the tip of the tool to an axis in the xy-plane, and z is the distance along the tool axis from the tip of the tool to the location where the elemental forces, dF_x and dF_y, are being applied.

CHAPTER 4. IMPLEMENTATION AND RESULTS

4.1 Implementation

To view a milling simulation, data describing the tool and workpiece geometry, and the tool path must first be loaded into the simulation program. This information is contained in a setup file. The number of instances of force calculation to be calculated and plotted per instance of tool movement displayed during the milling simulation is also specified in the setup file. The more instances, the smoother the plot and the more likely the force calculation will coincide with the maximum force values for each tool revolution. If the number of instances is too high, simulation speed decreases. Once a setup file is loaded, the user can orient the workpiece prior to dexel conversion.

4.2 Results

The results of force calculations produced from dexel information are compared with the model developed by Tlusty and McNeil [11]. The model developed by Tlusty and McNeil, described in Section 3.1.2, determines the force by integration over the range of flute angles that are cutting the workpiece. Tlusty and McNeil compute plots for three examples, varying either the radial depth of cut, a, or axial depth of cut, b (Figure 1). The three examples have been implemented for this research. Table 1 contains the maximum force on the tool in each example. The model implemented in this research corresponds closely with Tlusty and McNeil's model at the peak force values. In the column entitled "layers embedded in the workpiece," the different values are due to different tool orientation and scaling, not different values of b. The table indicates that the percent error varies depending on the number of slices currently cutting material. For example, when a=1.5875 mm, b=7.0 mm, and the scaling is such that there are 10 tool layers embedded in the workpiece, and the tool is rotated about its axis such that more than 4 layers are cutting, error is less than 25 percent.

The tool used for these calculations is a flat end tool with two flutes and a helix angle of 30 degrees.

Table 1: Comparison of results of summation of tool layers to integration over range of flute cutting angles to determine forces for three examples

a (mm)	*b* (mm)	layers embedded in the workpiece	layers cutting at max. force	max. force sum. (N)	max. force int. (N)	% error at max. force	max. slices 10% error	max. slices 25% error
1.5875	7.0	10	8	9.32	8.97	3.8	8	4
1.5875	7.0	20	17	9.21	8.97	2.6	10	3
1.5875	7.0	40	33	9.09	8.97	1.3	9	3
1.5875	7.0	50	41	9.11	8.97	1.5	7	4
1.5875	14.0	10	4	10.30	8.97	15.9	4	4
1.5875	14.0	20	9	9.89	8.97	10.0	9	4
1.5875	14.0	40	17	9.44	8.97	5.1	9	3
1.5875	14.0	50	21	9.14	8.97	2.0	9	4
3.175	7.0	10	10	15.93	16.69	4.5	9	4
3.175	7.0	20	20	16.61	16.69	0.4	10	3
3.175	7.0	40	40	16.37	16.69	1.9	9	3
3.175	7.0	50	50	16.43	16.69	1.5	7	4

4.2.1 Selecting an Orientation

The results of force calculations are view dependent, as Figure 16 indicates. They depend on tool size with respect to dexel size and on the orientation that the user chooses. Error is likely to increase when tool size decreases with respect to dexel size and when the tool axis direction approaches the *z*-direction.

For best results, a view should be selected so that the situation shown in Figure 16a is avoided, i.e., in which the viewing direction is nearly parallel to the tool surface cutting material. It is possible to select such an orientation if the radial depth of cut, *a*, in Figure 1, is constant. If the radial depth of cut is not constant, such an orientation cannot necessarily be selected. A similar situation exists when selecting an optimum orientation so that the axial depth of cut, determined by the number of dexels currently cutting material, corresponds to

the axial depth of cut determined by the number of tool layers currently cutting: it can only be accomplished if the axial depth of cut, b, is constant.

4.2.2 Sources of Error

By comparing results from the method used in this thesis, which sums force values for all layers of the tool that are cutting material, with equations (7) and (8) and the rest of Tlusty and McNeil's equations from [11], it is clear that this program can accurately reproduce Tlusty and McNeil's results at the peak force values. The error induced from summation of discrete flute cutting angles as opposed to integration over the range of cutting angles in generally around 2-5 percent (Table 1). The comparison shows that if there are more than ten layers cutting, there is generally less than ten percent error due to summation instead of integration.

Another source of error is that the axial depth of cut, b in Figure 1, tends to be over calculated due to the layering of the tool. Section 4.2.1 explains how error can be minimized with certain orientations.

According to equation (13), the thickness of a tool slice is greater than or equal to the width of a dexel. Therefore, since every dexel must be contained in a tool layer, it is likely that the axial depth of cut determined by summing tool slices will be larger than the axial depth of cut determined by stacking dexels. This can be seen in the following relationships:

$$W_l = w_l N_l \tag{18}$$

$$W_d = w_d N_d \tag{19}$$

$$N_l w_l \geq (N_d - 0.5)\, w_d \tag{20}$$

where W_l is the axial depth of cut for all tool layers cutting, N_l is the number of tool layers cutting, w_l is the width of a tool layer, W_d is the axial depth of cut for all dexels cutting, N_d is the number of dexels cutting, and w_d is the width of a dexel. Equation (20) is an inequality because the number of layers is an integer, and there must be enough layers to hold all the dexels. For the average case, equation (20) becomes:

$$N_l = \frac{(N_d - 0.5)\, w_d}{w_l} + 0.5 \tag{21}$$

Therefore, since w_l is larger than w_d, W_l in equation (18) is, on average, larger than W_d in equation (19).

The third and final source of error is miscalculation of the beginning and ending cutting angles, described in Section 3.2.4. However, experiments indicate that the error induced by this effect is minimal.

4.2.3 Examples

Results for different orientations for the example from Table 1 where a equals 1.5875 mm and b equals 7 mm are tabulated in Table 2 and shown in Figures 18 and 19. For the simulation with a equal to 1.5875 mm and b equal to 14 mm, rotated -77 degrees about the x-axis, and 9 degrees about the y-axis, milling simulation and plot output are displayed in Figure 20. Figure 21 shows the results with a equal to 3.175 mm and b equal to 7 mm, rotated -77 degrees about the x-axis, and 33 degrees about the y-axis.

Table 2: Maximum force on the tool in different orientations

1st rotation (degrees)	2nd rotation	3rd rotation	force magnitude (N)	figure number
-86 about x	9 about y		9.00 (best result)	18
-86 about x			8.43	19a
-34 about x			8.60	19b
-34 about x	69 about y		8.86	19c
-34 about x	69 about y	51 about z	9.32 to 8.89	19d
77 about z	-69 about x		9.57 to 8.78	19e
-77 about x	17 about y		9.07	19f

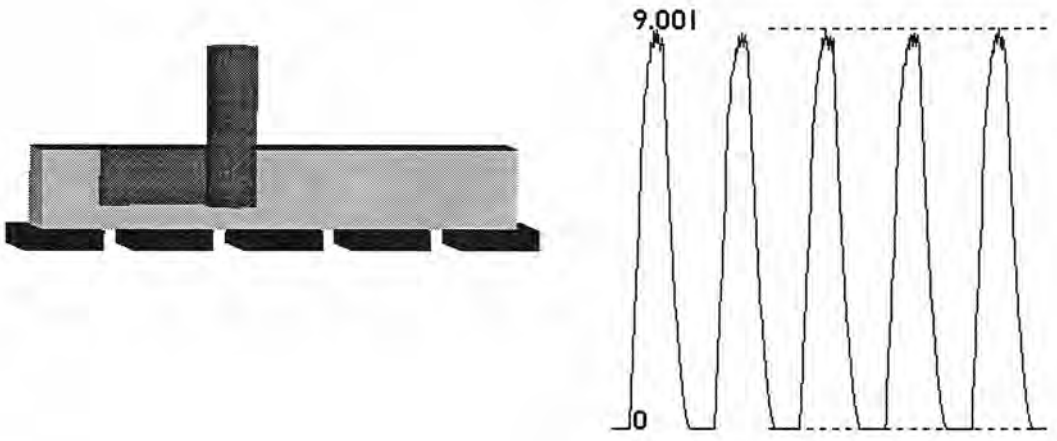


Figure 18 Snapshot of milling simulation and corresponding force plot

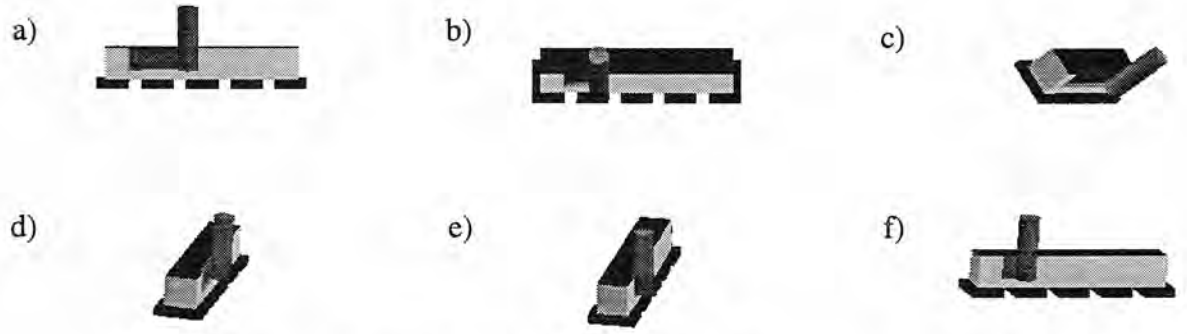


Figure 19 Milling simulation snapshots

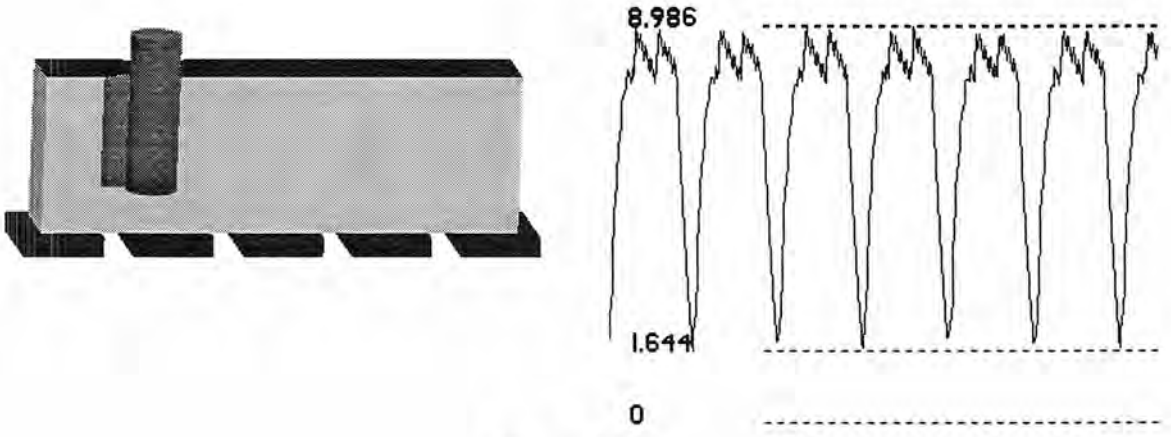


Figure 20

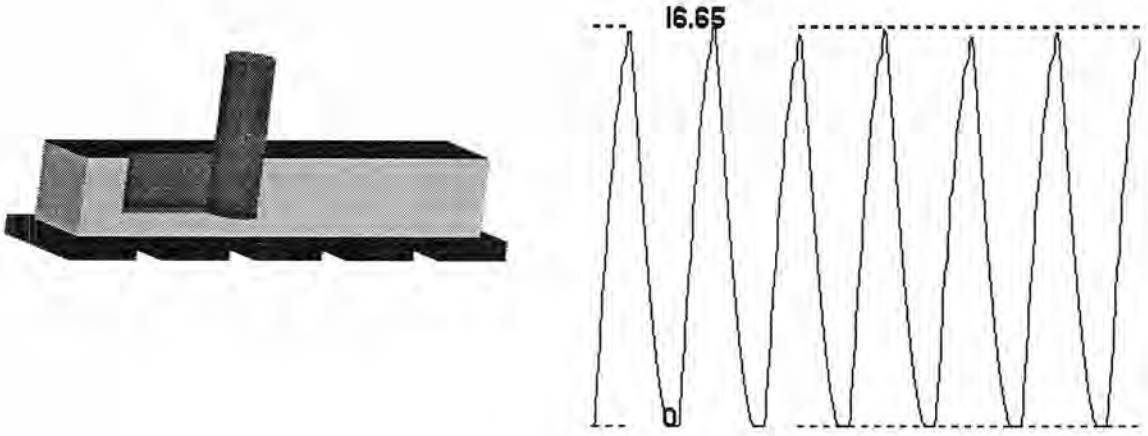


Figure 21

Figure 22 shows how scale affects the force calculations. In both plots, the tool orientation was the same as that shown in Figure 18. In Figure 22a, the scale was 75 percent of the default value. In Figure 22b, the scale was 50 percent. In all other examples in this section, the default scale is used.

Figure 23 shows how the force plots depend on the number of instances of force calculation plotted per instance of tool movement. The tool orientation was again the same as that shown in Figure 18. Figure 23a shows a plot where the number of instances is four times the number of flutes on the tool multiplied by the number of tool revolutions between instances of tool movement. This is the minimum number of instances the program will

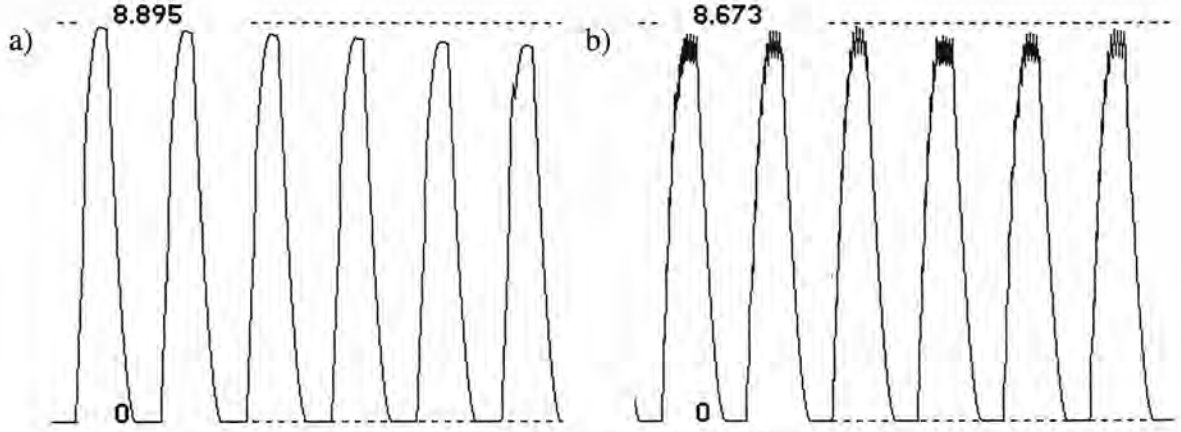


Figure 22 Scale effects the force calculations

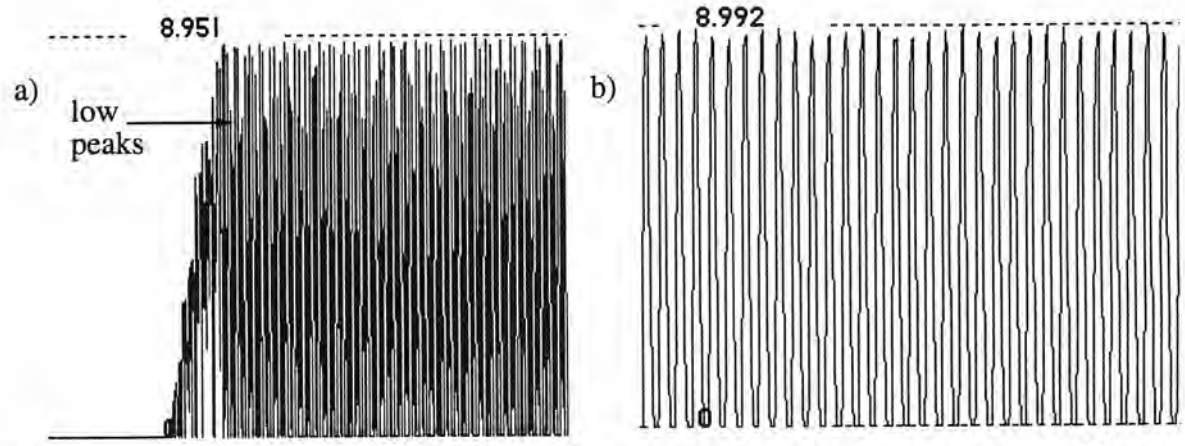


Figure 23 Different force plots resulting from varying the number of instances of force calculation to be plotted per instance of tool movement shown in the milling simulation

allow the user to input. It can be seen that the maximum force values per revolution are often missed when the number of instances is set to this small value. There are many peaks well below the maximum. The peaks are low at the left end of this plot because the tool has not completely entered the workpiece. Figure 23b contains four times as many instances of force calculation as (a). With this number of instances, the peaks are nearly level. Unless otherwise specified, for all other examples in this section, the number of instances of force calculation plotted per instance of tool movement shown in the simulation is equal to 80 multiplied by the number of flutes on the tool multiplied by the number of tool revolutions between instances of tool movement.

Figure 24 shows a case where the axial depth of cut decreases from 7 mm to 2 mm. The tool in Figure 24a is again in the same orientation as Figure 18. Figure 24b shows a plot as the axial depth is decreasing, and (c) shows the force plot at the minimum axial depth of cut. The number of instances is 16 multiplied by the number of flutes on the tool multiplied by the number of tool revolutions between instances of tool movement.

Forty to fifty cutting layers were in contact with the workpiece at the maximum force values in all examples except those corresponding to Figure 19b and (c). Those two used less than forty layers. In examples where the radial and axial depth of cut are not constant, fewer than forty layers may be in contact at some instances.

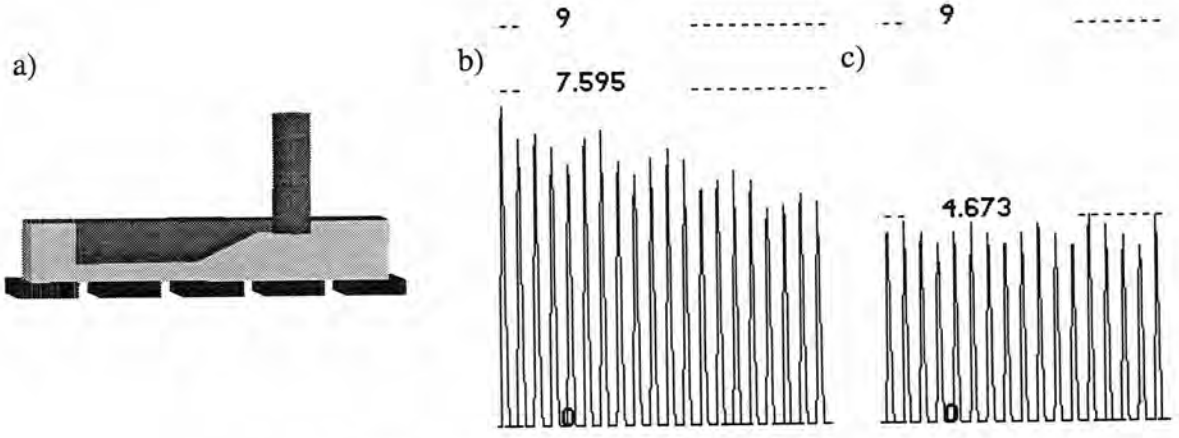


Figure 24 Decreasing axial depth of cut

CHAPTER 5. CONCLUSIONS AND FUTURE RESEARCH

A basic model for force analysis can be added to a milling simulation that performs dimensional verification without greatly affecting the simulation time. If a more realistic model such as the one developed by Tlusty and Ismail [14] is used, the calculations at each step are more complicated, involving integrations, whereas the model used in this thesis only requires multiplication.

Despite the detailed information about tool and workpiece contact that the use of dexels makes available, the angles where flutes enter and exit the workpiece cannot always be accurately calculated. Certain orientations give better results. The viewing angle and the scale also affect force calculations. If the radial and axial depth of cut are constant, optimum orientations can be found for viewing the simulation to give the best force calculations that this program can compute.

Implementation of this algorithm allows the user of an NC milling dimensional verification program to get an estimation of the forces during the milling process while running a material removal simulation. If unexpected peak forces appear during the simulation, the tool path programmer knows immediately that the tool path needs adjustment.

Future work involves making both the cutting angles determined from dexel information more accurate, and possibly using a more accurate force model discussed in the introduction. Currently the latter cannot be done without sacrificing the speed of the simulation.

REFERENCES

[1] Oliver, J. H. and Goodman, E. D., "Direct dimensional NC verification" *Computer Aided Design*, Volume 22, Number 1, 1990, pp. 3-9

[2] Le Maistre, Christopher, and El-Sawy, Ahmed, *Computer Integrated Manufacturing, a Systems Approach*, UNIPUB/Krause International Publications, White Plains, New York, 1987

[3] Voelcker, H. B., and Hunt, W. A., "The Role of Solid Modeling in Machine-Process Modeling and NC Verification," *SAE Technical Paper* No 810195, February 1981

[4] Wang, W. P. and Wang, K. K., "Real-Time Verification of Multiaxis NC Programs with Raster Graphics," *Proceedings of IEEE International Conference on Robotics and Automation*, San Francisco, California, 1986

[5] Sambandan, K. and Wang, K. K., "Five-axis Swept Volumes for Graphic NC Simulation and Verification," *ASME Advances in Design Automation*, B. Ravani, ed., DE-Volume 19-1, 1989, pp. 143-150

[6] Jerard, R. B., Drysdale, R. L., Lauck, K., Schaudt, B., and Madgewick, J. "Methods for detecting errors in numerically controlled machining of sculptured surfaces," *IEEE Computer Graphics and Applications*, Volume 9, Number 1, 1989

[7] Narvekar, A., Huang, Y., and Oliver, J. H., "Intersection of Rays with Parametric Envelope Surfaces Representing Five-Axis NC Milling Tool Swept Volumes," *Proceedings of ASME Advances in Design Automation*, Volume 2, D. A. Hoeltzel, ed., 1992, pp. 223-230

[8] van Hook, T., "Real-Time Shaded NC Milling Display," *Computer Graphics, Proceedings of SIGGRAPH '86*, Volume 20, Number 4, 1986, pp. 15-20

[9] Huang, Y., Ph. D. Dissertation, *Dimensional Verification and Correction of Five-Axis Numerically Controlled Milling Tool Paths*, Iowa State University, IA, 1993

[10] Smith, S. and Tlusty, J. "An Overview of Modeling and Simulation of the Milling Process," *Journal of Engineering for Industry*, Volume 113, May 1991, pp. 169-175

[11] Tlusty, J. and MacNiel, P., "Dynamics of Cutting Forces in End Milling," *Annals of the CIRP*, Volume 24, 1975, pp. 21-25

[12] DeVor, R. E., Kline, W. A., and Zdeblick, W. J., "A Mechanistic Model for the Force System in End Milling with Application to Machining Airframe Structures," *8th NAMRC Proceedings*, 1980, pp. 297-303

[13] Sutherland, J. W. and DeVor, R. E., "An Improved Method for Cutting Force and Surface Error Prediction in Flexible End Milling Systems," *ASME Journal of Engineering for Industry*, Volume 108, November 1986, pp. 269-279

[14] Tlusty, J. and Ismail, F., "Special Aspects of Chatter in Milling," *ASME, Journal of*

Vibration, Acoustics, Stress, and Reliability in Design, Volume 105, January, 1983, pp. 24-32

[15] Sutherland, J. W., "A Dynamic Model for the Cutting Force System in the End Milling Process," *Sensors and Controls for Manufacturing*, 1988, E. Kannatey-Asibu, et al. eds., ASME PED Volume 33, December 1988, pp. 53-62

[16] Wang, W. P., "Solid Modelling for Optimizing Metal Removal of Three-Dimensional End-Milling," *Journal of Manufacturing Systems*, Volume 7, Number 1, 1988, pp. 57-66

[17] Yellowley, I., "A Note on the Significance of the Quasi-Mean Resultant Force and the Modelling of Instantaneous Torque and Forces in Peripheral Milling Operations," *Transactions of the ASME*, Volume 110, August 1988, pp. 300-303

[18] Martelloti, M., "An Analysis of the Milling Process, Part II-Down Milling," *Transactions of the ASME*, Volume 67, 1945, pp. 233-251