Computer-Aided Design and Applications

http://www.cadanda.com

Curvature Matched Roughing Using Flat End Mills

Edward Red[1], Greg C. Jensen[2] and Michael B. Thompson[3]

[1]Brigham Young University, ered@byu.edu
[2] Brigham Young University, cjensen@byu.edu
[3]SkyWest Airlines, mthompson@skywest.com

ABSTRACT

This paper presents an algorithm that will reduce the overall machining time for parts and surfaces, by focusing on the process of rough machining. The algorithm will incorporate principles of Curvature Matched Machining (CM^2) and planar cutting for rough machining; hence, we introduce the algorithm for Rough Curvature Matched Machining (RCM^2). The algorithm will 'morph' planar machining slices to the semi-roughed surface, allowing the finish pass to be complete in one pass. The RCM^2 roughing algorithm can save up to 60% of the machining time over current roughing techniques.

Keywords: curvature matched machining, rough machining.
DOI: 10.3722/cadaps.2009.181-194

1. INTRODUCTION

Jensen [1] developed a set of algorithms that optimized the machining of surfaces using a flat end mill, by adjusting the orientation of an inclined flat end mill being swept through contact points. Effectively, curvature-matched machining (CM^2) matches the tool's projected curvature to local surface curvatures, resulting in fewer passes per inch, reduced scallop height and improved surface finish. CM^2 technology is more efficient than conventional methods in the machining of surfaces such as a boat impellor, a turbine blisk (bladed-disk stage), automobile hood and door dies, etc. But rapid changes in tool orientation pose additional machining challenges such as surface interference and tool collisions. For example, Ernst [2] developed routines to eliminate tool surface gouging due to local curvature variance, as well as gouging cavity wall when machining within pockets.

Machining is expensive, particularly for parts with complex surfaces such as dies. The average lead time for an American mold and die manufacturer is 20-30 weeks, Fallbohmer [3]. Of that time 60% is actual machining with about half the machining time dedicated to rough machining (7-8 weeks). Even a 10-15% reduction in machining time represents considerable time and costs savings.

Current roughing practices use 3-axis planar cuts. The planar cuts are designed to remove the most material in the shortest amount of time. However, this rough cutting approach results in the stair-stepped features of Fig. 1. Before finish machining the part surface, two semi-rough passes are generally utilized to remove the stairs, adding to the overall machining time.

CM^2 uses 5-axis motion, which theoretically is slower than 3-axis motion because of dynamic tool reorientation. The authors postulate that by combining 5-axis motion (CM^2) with 3-axis planar motion the part stock can be morphed to a surface ready for finish machining faster than conventional 3-axis planar cuts with semi-roughing passes. Combining CM^2 with 3-axis planar cuts, Rough Curvature Matched Machining (RCM^2) is the first method to integrate CM^2 into rough machining.

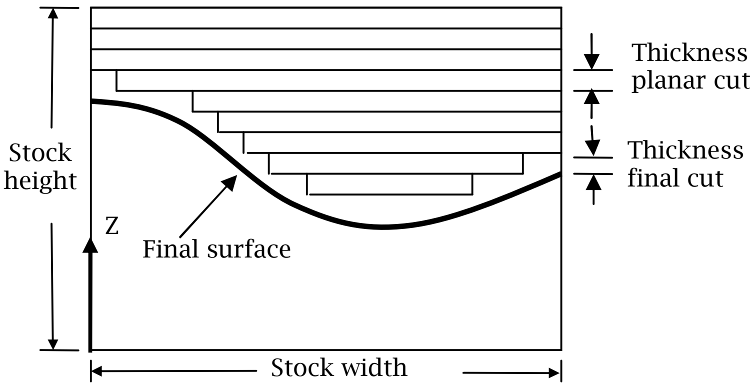

Fig. 1: Stair stepped planar roughing.

2. BACKGROUND

Currently, less than 8% of mold and die manufacturers in the United States use 5-axis machines. Current mold and die research focuses on automating mold and die design and decreasing machining time. One of the challenges is machining undercut features, with recesses or protrusions. Ye [4] developed an algorithm to geometrically recognize an undercut feature. Although mathematically complex, implementing this algorithm can decrease the overall fabrication time for a mold.

Research is also being conducted on die cavity pocketing. Choi [5] developed an algorithm that defined a boundary surface and created tool paths based on 2-D offsets from that boundary path. Choi's earlier research method is computationally intense because it uses optimization methods to search forpaths that minimize cusp height in terms of cutter tilt and yaw angles.

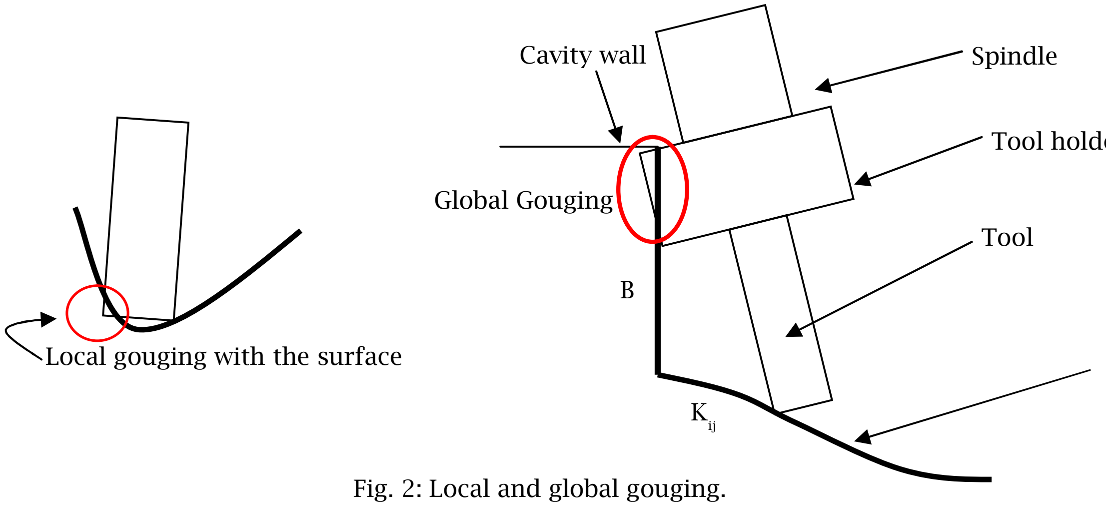

Fig. 2: Local and global gouging.

2.1 Gouging

Gouging is defined as interference by the cutter, tool holder, or spindle with the finished surface or cavity walls. In this paper local gouging is defined as interference with the part surface and global gouging is interference with the cavity walls – see Fig. 2.

There have been several effective methods of determining local gouging, including ray casting [6], colinear normal lines [7], and triangulation [8-9]. In all cases, when creating a five axis tool path, each cutter location and orientation is checked and corrected if there is interference. If an acceptable orientation cannot be determined, the tool position is adjusted.

The selected local gouge detection and correction method of this research was first developed by Ernst [2] because it is less computationally intense and programmatically easier to define than triangulation.

Our global gouging routine bounds the tool, tool holder, and spindle with cylinders of appropriate size. We let B define a boundary and K_{ij} be the final surface. At every point along the tool path the cylinders are checked for interference with K_{ij} and B. If any cylinder is found to gouge, then the tool is reoriented by incrementing the tool tilt angle α to remove the interference.

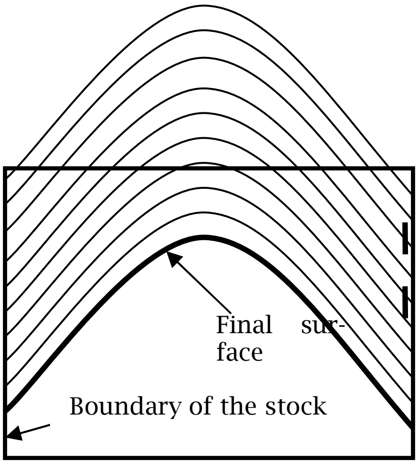

Fig. 3: Offset surface machining.

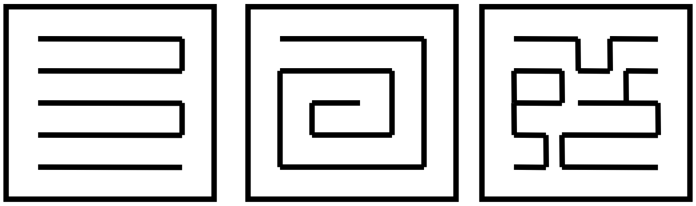

Fig. 4: Zig-zag, offset and random patterns.

2.2 Rough Machining

Rough machining practices are defined by two types of tool movement: tool path generation algorithms and tool path patterns. Historically, roughing algorithms began with offset curves and have transitioned to the contour map approach.

The most basic method of rough machining uses a 3-axis flat end mill to cut the surface in planar cuts. The planes are bounded by the minimum amount of material, i.e. the finished depth of cut. One deviation of the planar cuts approach is the use of offset machining. The final surface is offset and then machined at a constant thickness, as illustrated in Fig. 3. This approach has several obvious problems. Ultimately, these deficiencies led to the contour map approach, a method that incorporates both planar and offset cuts.

Contour mapping - The contour map approach proposed by Vickers [10], while currently gaining popularity as a viable method of rough machining free-form surfaces, is still not regarded as the industry roughing standard. Contour mapping consists of setting up an optimization equation that details the total production time for machining a part, considering the cutting time, tool approach and return time, part loading and unloading, tool change time, and tool life. All of the variables are incorporated into an optimization loop to determine the number of cutting planes that minimize the total cutting time within the given constraints of feed, speed, and depth of cut. In addition, the contour map approach introduces the concept of rough machining around the geometric features of the stock to produce a surface similar to the final, without the inefficiencies of the pure offset approach.

There are several differences between the contour map approach and RCM^2. First, the contour map approach uses a 3-axis motion to rough cut the parts while RCM^2 combines 3-axis and 5-axis motion. And second, the contour map approach makes no adjustment for differences in surface curvature, nor does it make any attempt at tool orientation or tool offset machining. Thus, the two concepts, while based on similar principles have different implementations.

Cutting patterns - Another important area of roughing is the cutting pattern involved in the creation of tool paths. There are three fundamental tool patterns: 1) zigzag (or staircase, direction parallel) approach used in the RCM^2 research described in this paper, 2) offset (or window frame, contour parallel milling) and 3) random as shown in Fig. 4. Most CAD/CAM packages use either one directional cutting or a simple zig-zag pattern. Other methods and patterns have been proposed [11] but generally have fallen short of providing significance savings.

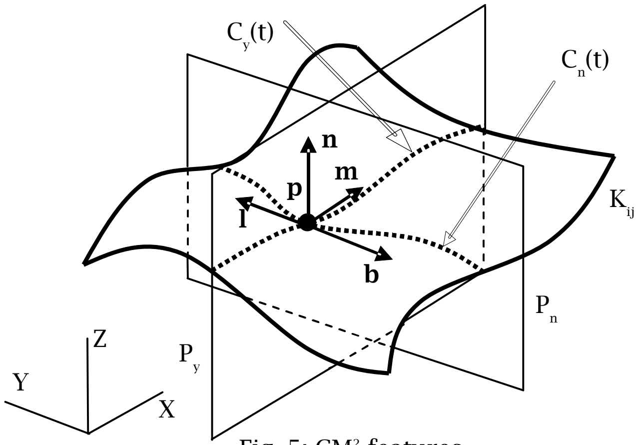

Fig. 5: CM² features.

Additional research - Vafaeesefat and El-Margahy [12] developed an approach to pocket roughing that creates offset boundary conditions for sculptured surfaces. Their work uses a 2-D Z-map array that stores the Z-values of grid points on the X-Y plane. Huang [13] considered image detection to create NC tool paths, by breaking the roughing procedure into two parts, an initial rough cut and a finish rough cut, based on the complexity of the geometry. This approach adds additional time for tool changeovers since Huang allowed for different size tools.

3. RCM² METHOD

The RCM² method of rough machining is presented along with its benchmarking process. An empirical model is proposed as a measure of projecting cutting times. The algorithms and equations were implemented using Visual C++ and the Siemens NX Application Programming Interface (API).

3.1 Surface Definition

Consider the finish machined surface patch defined by K_{ij} in Fig. 5 where i = 1,2, … n and j = 1,2,..m. Surface K_{ij} is offset by a vertical distance Δ to create surface $P_{ij} = K_{ij} + \Delta$ (not shown). Δ will change depending on the specific implementation (semi-rough, rough or finish).

To understand curvature matched machining, consider surface K_{ij}, and a plane perpendicular to the y axis, P_y. $C_y(t)$ is the curve resulting from the intersection of surface K_{ij} and plane P_y. Let **p** be a point on $C_y(t)$ and **m** be the unit vector tangent to $C_y(t)$ at point **p**. P_n is a plane normal to **m** and $C_n(t)$ is the resulting curve created by intersecting surface K_{ij} and P_n. The curvature at point **p** on curve $C_n(t)$ is the normal curvature. **l** is the unit vector tangent to $C_n(t)$ at point **p**. The surface unit normal **n** of point **p** on surface K_{ij} is created by the cross product of $\mathbf{m} \otimes \mathbf{l}$. The bi-normal of point **p**, **b** is $\mathbf{m} \otimes \mathbf{n}$, where the tool is tilted about **m** in the direction of **b** by α.

3.2 Machining Parameters

The creation of an accurate, affordable milled part is a careful balance of many machining parameters. These parameters, i.e. inputs to the tool path algorithm, include such things as finished scallop height or tool step over, depth of cut, width of cut, tool radii, plunge and retract feed rates, roughing feed rate, finish feed rate, spindle speed, etc. Some of these parameters are not simple factors but are relationships controlled by yet other factors and relationships. For example, surface finish is a function of controlling both the waviness of the scallops, i.e. tool step over, as well as the roughness of the nominal profile, i.e. how sharp the tool is, the feed rate and spindle speed, the hardness of the material, etc. A first glance the reader may assume that depth of cut, which refers to the distance (in Z direction) between two consecutive machining layers, is a simple parameter to calculate or choose some random default value. However, due to the multitude of factors and relationships that a seasoned NC part programmer must consider, the depth of cut value is generally left as an input to the tool path algorithm to ensure rapid and cost effective machining.

3.3 Benchmark
The benchmark method was designed to match current industry machining practices. Current rough-

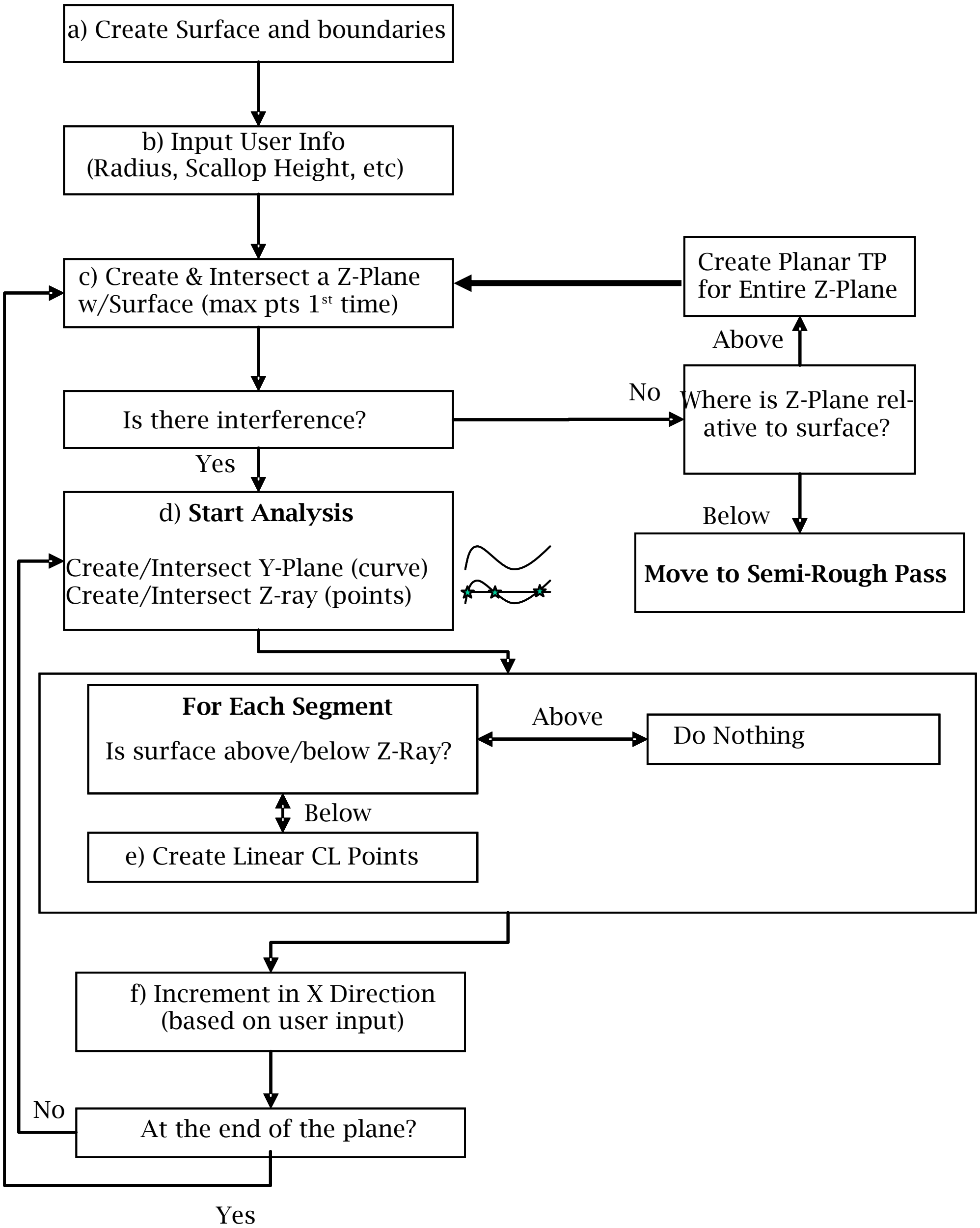

Fig. 6: Benchmark algorithm.

ing practice involves first, rough machining using a flat end mill, second, semi-rough paths using a ball end mill, and finally a finish path using a ball end mill. This paper focuses on isolating the roughing and semi-roughing practices.

Roughing - The benchmark roughing technique utilizes planar flat end mill roughing. CL points are created such that the flat end mill removes the most material without gouging the final surface. The tool paths are created by traversing the surface in either the x or y horizontal direction and machining the surface planar steps, Fig. 1. The specific algorithm used to create the benchmark roughing tool paths is outlined in Fig. 6. After defining the surface and boundary conditions (Fig. 6a) and retrieving the user information (Fig. 6b) the surface is analyzed (Fig. 6c). Consider the offset surface P_{ij} defined earlier. Now define n_{cp} as the number of horizontal x-y cutting planes $P_{zk}(k = 1, 2,\ldots, n_{cp})$ and D_c as the depth of cut where n_{cp} is determined from the vertical cutting range:

$$n_{cp} = (Z_{max} - Z_{min})/D_c \tag{3.1}$$

If $P_{ij} \cap P_{zk}$ is false and P_{zk} is below the surface the algorithm will move to a semi-rough pass. If $P_{ij} \cap P_{zk}$ is false and P_{zk} is above the surface the algorithm can create planar zig-zag tool paths for P_{zk} without fear of gouging the surface. If $P_{ij} \cap P_{zk}$ is true, the surface data (Fig. 6d) must be analyzed further.
Consider surface P_{ij} and a vertical slicing plane perpendicular to the Y axis, P_y. Curve $C_y(t)$ is the curve created by intersecting P_{ij} and P_y: $C_y(t) = P_{ij} \cap P_y$, Fig. 7. Intersecting $C_y(t)$ with P_{zk} and $P_{z(k-1)}$ yields the segmented curve as shown in Fig. 8. Each segment in the curve is analyzed and CL data points are created

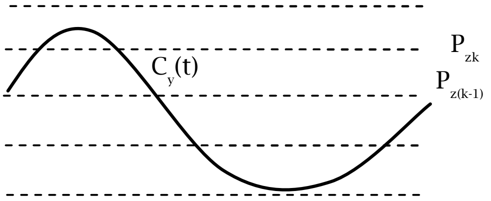

Fig. 7: Machining layers.

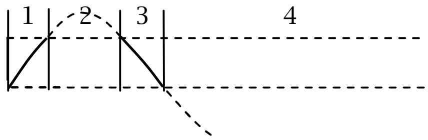

Fig. 8: Analysis of machining layer.

(Fig. 6e). P_y is then incremented (Fig. 6f) using a commonly applied value of $(2R_f - D_c)$, where R_f is the flat end mill radius and D_c is the depth of cut. Once the roughing pass is complete, the ball end mill semi-rough pass must be completed.
Semi-Roughing - The surface is offset twice using multiple semi-rough passes to maintain a fairly constant load on the cutter, avoiding surface irregularities and possible tool failure. The first iteration removes the stair-stepped features created by the roughing planar pass. The second iteration smoothes the features in preparation for the finish pass.
To remove the stair-stepped feature, the first semi-rough pass uses a depth equal to one half of the depth of cut to compensate for differences in load: $\Delta = D_c/2$. Δ for the second semi-rough pass is the depth of cut required by the finish pass. The result will be a surface ready to be finish machined.
The curve is tessellated into a series of incremental linear movements, as a function of a specified tolerance, where L_x and L_y bound the curve's second derivative, and ε is the allowed path deviation.

$$q \geq \sqrt{\frac{\sqrt{L_x^2 + L_y^2}}{8\varepsilon}} \tag{3.2}$$

Step Over - Incrementing plane P_y for step over requires that we approximate the geometry left by ball end mill machining. The step over is a function of scallop height S_h in (3.3), where $S_{o(BEM)}$ is the ball end mill step over and R_b is the radius of the ball end mill. It is determined as twice the horizontal projection of the tool radius inclined to the scallop height:

$$S_{o(BEM)} = 2\sqrt{R_b^2 - (R_b - S_h)^2} \tag{3.3}$$

Benchmark Finish - Ball end mill finish points are created using the same mathematics and method as the semi-rough paths.

3.4 RCM²

RCM² CL-points are derived using the algorithm of Fig. 9, similar to the roughing algorithm for the benchmark tool paths. However, the main difference is in the handling of the CM² segments, step over, transitions and gouge detection.

CM^2 segments are represented by incremental linear movements defined by (3.2) and CL data points are created. Planar segments are handled in a similar fashion to those of the roughing algorithm.

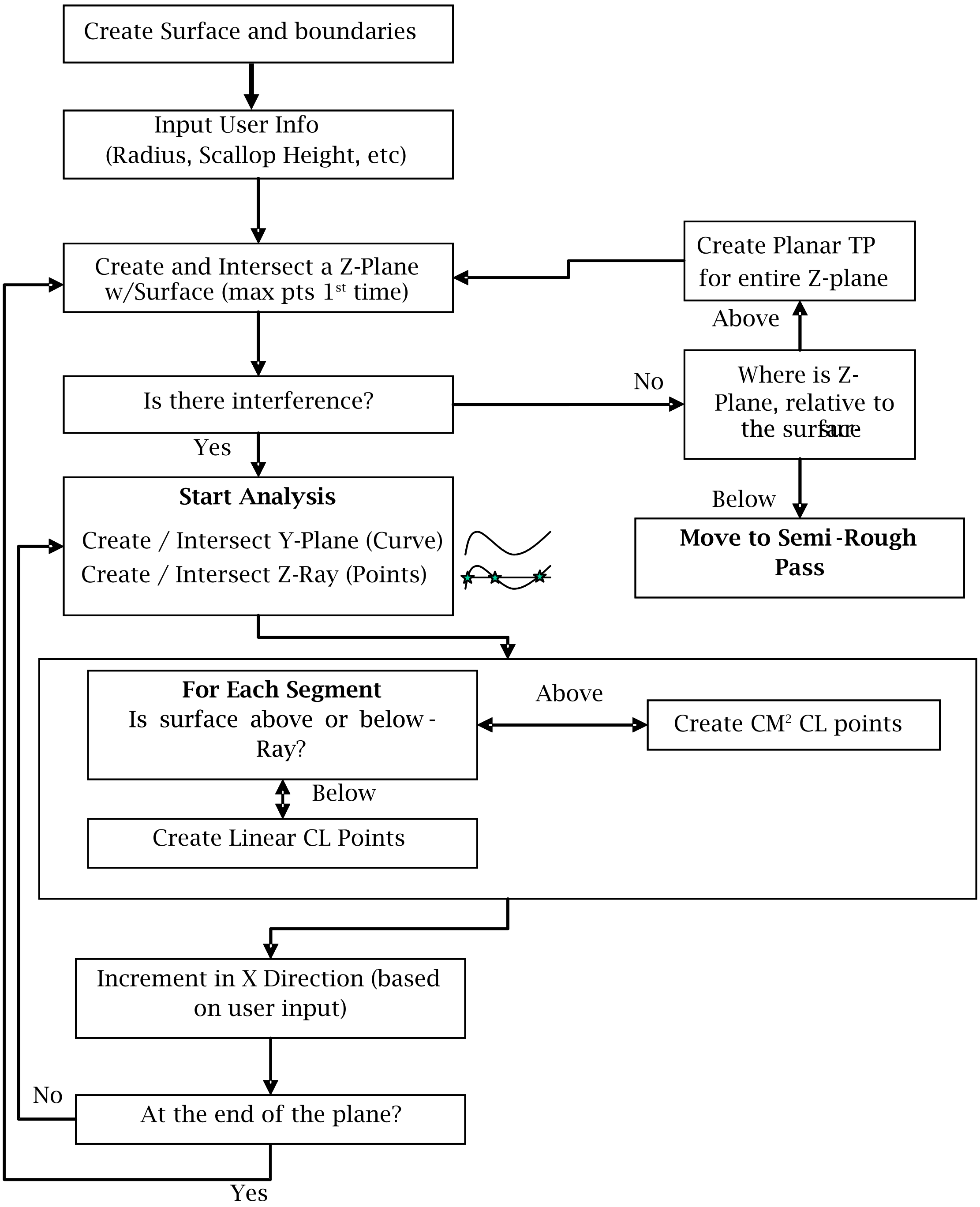

Fig. 9: RCM^2 algorithm outline.

RCM^2 Step Over - The RCM^2 tool step over directly affects the surface finish and drives the surface scallop height. When semi-rough machining a surface it is desirable to maintain control over the maximum surface scallop height. Since the surface will require a finish pass, it is not necessary to have a completely uniform scallop height.

Kitchen [14] & Hill [15] developed the cut width for an inclined CM^2 tool. The cut width function uses the curvature of the surface to approximate a circle. The circle is then intersected with the ellipse formed by the bottom of the cutter. The cut width equation developed from that relationship is shown in (3.4):

$$Cut\_Width = 2\sqrt{\left|\frac{1}{2}\left(\frac{-2\frac{R_f}{\sin(\alpha)} + 2\kappa_n - \psi}{-\frac{1}{\sin^2(\alpha)} + 1}\right)\right|} \tag{3.4}$$

where

$$\psi = \frac{\sqrt{8\kappa_n(S_h\cos^2(\alpha) - R_f\sin(\alpha)) + 4(R_f^2 + \sin^2(\alpha)\kappa_n^2 - -S_h^2\cos^2(\alpha))}}{\sin(\alpha)} \tag{3.5}$$

Equation (3.4) provides the cut width for a point machined by CM^2. As a curve is analyzed, the minimum cut width for a given pass would become the step over for that curve; thus, passes would have a variable step over depending on the surface curvature. Using the scallop height as the surface tolerance assures that the created pass doesn't result in the machining of a scallop that exceeds the maximum scallop height.

Tool Transitions - One important element in the RCM^2 method is the tool transition. As the tool is moved along a cutting plane it is required to machine both planar and five-axis points. As the tool approaches and exits a CM^2 cut there are two factors: 1) first, the tool must not gouge the final surface; 2) to avoid overloading the tool the tool must not cut more than the allowable width or depth. Thus, when making a transition the tool should not machine below its current cutting plane. In addition, the tool needs to be able to transition from planar to CM^2 (and vice versa), curve – curve, and plane – plane motion. Appropriate transitions are essential to tool paths. Transitions refer to the motion of the tool within and between curves as well as between machining planes. Successful transitions do not gouge the finished surface or any other defined boundaries and they do not result in excess movement.

Since the RCM^2 implementation requires planar to CM^2 transitions, it is necessary to develop a method for transitioning a tool between three and five axis motion, considering inner curve transitions (movement of the tool within the curve, i.e. transition from planar to CM^2) and curve – curve transitions. Transitions between planes are handled as curve – curve transitions.

Inner Curve Transitions - Consider several intersection curves i of Fig. 7, $C_{yi}(t)$, where i = 1,2…n and n is an unknown number which depends on the step over, defined by the intersection of a plane perpendicular to the y axis, P_{yi} and surface P_{ij}. As the layers are machined the tool must be able to transition from planar motion to CM^2 motion and vice versa along curve $C_{yi}(t)$ without plunging into the layer below. If the tool were allowed to plunge into the layer below it would exceed the specified depth of cut and could result in tool deflection or tool failure.

Consider the tool motion as a result of machining one of the layers of curve $C_{yi}(t)$ as seen in Fig. 10. The curvature of the surface requires the tool to plunge into the layer below on the transition from planar motion to CM^2 motion. In order to prevent potential tool deflection, the tool is not allowed to machine a CL point lower than the current plane (P_{zk}). Points along the tool edge are checked for their location relative to the current maximum depth. If a CL point caused the tool to plunge into the layer below, the tool is rotated along an axis perpendicular to the normal vector **b** of the point on Curve $C_{yi}(t)$ until it no longer plunged, Fig. 11. If the tool must be rotated > 45° and is still below the maximum depth of cut, the tool is rotated to a vertical orientation and transitioned to a three axis move. Tool rotation is capped at 45° to prevent the tool from rotating underneath the surface and creating an undesirable CL point. All CL points are checked for interference with the final surface and boundaries according to the gouge detection algorithms outlined earlier.

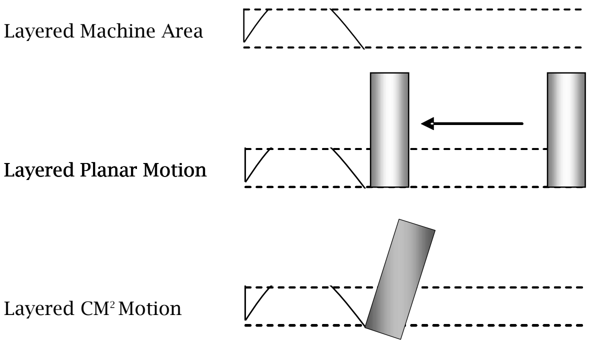

Fig. 10: CM^2 without adjusting plunging effect.

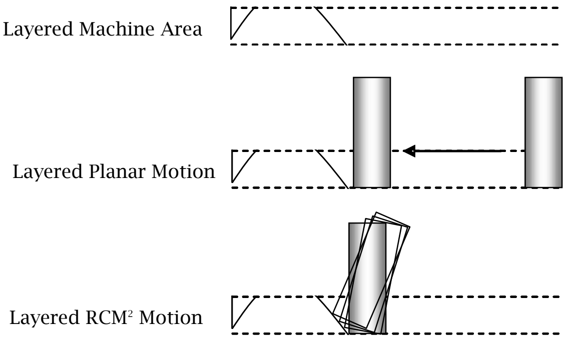

Fig. 11: RCM^2 tool transition: planar to CM^2.

Curve–Curve Transitions - After the feasible areas of curve $C_{yi}(t)$ have been machined the tool transitions to the next curve ($C_{y(i+1)}(t)$). The distance between these two curves is determined by the step over equations outlined earlier. Consider two curves, $C_{yi}(t)$ and $C_{y(i+1)}(t)$, both created by intersecting planes P_{yi} and $P_{y(i+1)}$ respectively with surface P_{ij}. P_{yi+1}, the plane used to create curve $C_{yi+1}(t)$, is offset a distance equal to the step over from plane P_{yi}. Fig. 12 is an analysis of the curves and the resulting CL points. Red points indicate that machining the point is not feasible. The tool needs to transition from the last feasible point (**pl**) on $C_{yi}(t)$ to the first feasible point (**pf**) on $C_{y(i+1)}(t)$.
To ensure no gouging occurs, the area between $C_{yi}(t)$ and $C_{y(i+1)}(t)$ is analyzed. Consider plane P_b created from three points $\mathbf{p}_{li}$, $\mathbf{p}_{f(i+1)}$ and $\mathbf{p}_{f(i+1)} + [0,0,\delta]$; δ need only be large enough to differentiate point $\mathbf{p}_{f(i+1)}$, so the actual value of δ is irrelevant. Curve $C_b(t) = P_c \cap P_{ij}$ is analyzed between $\mathbf{p}_{li}$ and $\mathbf{p}_{f(i+1)}$ to determine the maximum Z value for transitioning the tool between curve $C_{yi}(t)$ and $C_{y(i+1)}(t)$.

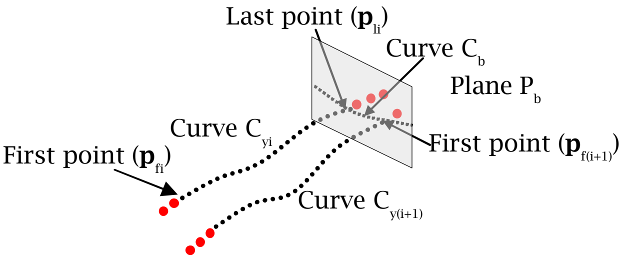

Fig. 12: Creation of curve $C_{y(i+1)}$.

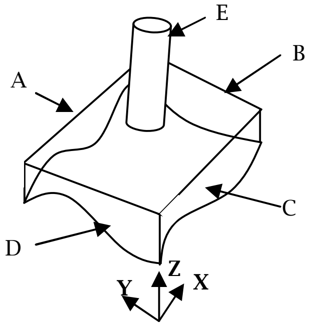

Fig. 13: Walled cavity definition.

Gouge Detection - Gouging is interference between the tool and any unwanted object, including the finished surface and cavity walls shown as Fig. 13. Local and global gouging is detected and corrected using the algorithm in Fig. 14. Both the local and global gouge detection routines for the RCM^2 algorithm are variations from those discussed in Ernst [2].
Only flat end mills will be used for RCM^2. Therefore, there is no need to discuss the implications of filleted or ball end mills with regard to gouge detection. Since a flat end mill is rectangular in size it is appropriate to represent the tool as a cylinder located at a certain orientation and position. Let the surface of interest be represented by P_{ij} and E be the geometry defined as a cylinder representing the tool.
E is checked for interference with surface P_{ij}. If $E \cap P_{ij}$ is false, then the tool doesn't gouge the surface. However, if $E \cap P_{ij}$ is true, adjustments to the tool orientation need to be made. Once E is checked for interference with P_{ij}, E is also checked for interference with the walled surfaces representing the cavity, surfaces A, B, C, and D.

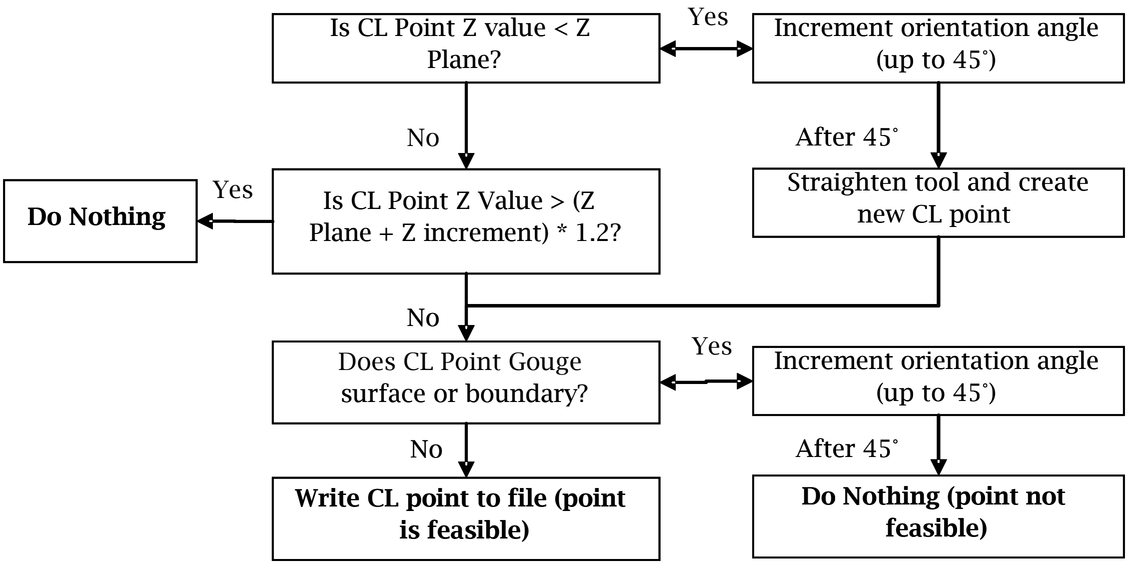

Fig. 14: RCM^2 CL data point algorithm.

RCM^2 Finish - The finish algorithm for RCM^2 is identical to that used to create the benchmark finish tool paths, leading to similar tool paths, with the exception of the initial tool location.

3.5 Empirical Testing Model

There are two important elements of the empirical testing model. The first piece is the development of a mathematical model to predict machining times. The second is a simple prediction algorithm that doesn't rely on an analysis of the final tool paths to determine which method is faster.

Machining Time Mathematical Model - Vickers et al. [10] set up machining time as a function of cutting time, tool approach and return time, part loading and unloading time, tool changeover time and tool life. The author assumed that all times, except cutting time, would be held constant by machining with the same surfaces, the same tools and similar size work pieces. The empirical testing will look only at one element of the total machining time and that is the cutting time. Cutting time is estimated as $C_t = T_l / F_r$ where C_t is the total cutting time, F_r is the feed rate and T_l is the total tool path length.

The equation to project machine cut times in five axis motion is more complicated and involves an analysis of each tool segment. Consider a segment of tool path motion required to move from TP_{i-1} to TP_i, where TP_i is designated by (X,Y,Z,A,B). The segment time is the maximum time required to either transverse the (X,Y,Z) distance or rotate the A or B axis, whichever is greater. The maximum segment times is summed to create the total machining time:

$$C_{t(5\_axis)} = \sum_0^n max(F_r * T_l, F_A * D_A, F_B * D_B) \tag{3.6}$$

where F_A, F_B are the angular feed rates and D_A, D_B are the rotation angles for the A and B axes respectively.

Prediction Algorithm - While the mathematical model to predict machining time is beneficial, it requires the tool paths for both alternatives to be created. It is not always feasible to develop the entire tool paths for both alternatives and compare the resulting times. The following algorithm provides a method for estimating whether RCM^2 is faster than conventional machining by analyzing the surface features and simple machining parameters.

Conventional machining tool path length is a function of the roughing flat end mill tool paths and the semi-rough and finish ball end mill tool paths. Since both experiments exclude the effects of finish machining, the finish ball end mill paths can be assumed to be constant. RCM^2 tool path length is a function of the flat end mill CM^2 and flat end mill planar tool paths.

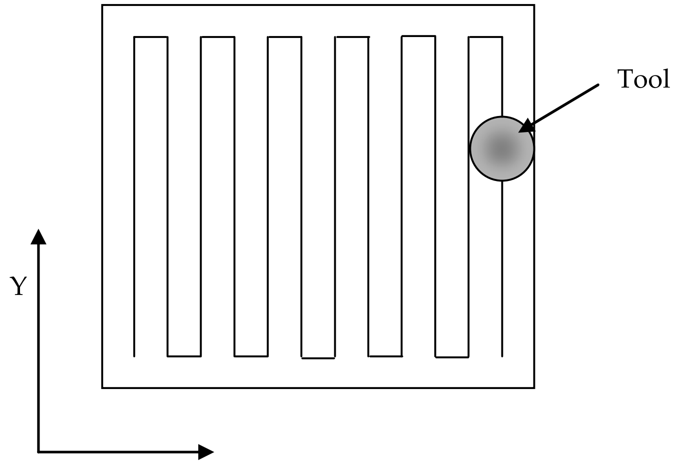

Fig. 15: Tool motion.

Consider the generic motion of a tool following a tool path along a surface as seen in Fig. 15. The tool machines a path along the surface in the Y direction and steps over in the X direction. Generic total tool path length is a function of the number of times the tool traverses the surface along the Y axis.
The semi-rough ball end mill tool paths must cover the entire surface twice regardless of surface features. The number of Y curve crossings (Y_x) is specified in (3.7), where C_v is the concavity factor, $S_{O(FEMR)}$ is the flat end mill roughing step over, and $S_{o(BEM)}$ is the ball end mill step over:

$$Y_x = \frac{2(X_{\max} - X_{\min})}{S_{o(BEM)}} + \sum_{i=0}^{n_{cp}-1} \frac{\left(\frac{i}{n_{cp}}\right)^2 (X_{\max} - X_{\min})}{S_{o(FEMR)}} C_v + \frac{(n_{cp}-1)(X_{\max} - X_{\min})}{S_{o(FEMR)}}(1 - C_v) \quad (3.7)$$

Concave surface areas shorten the machining time for the conventional roughing algorithm, by producing a shorter tool path length for flat end mills, Fig. 16. RCM^2 is unaffected. Area #1 and Area #2 would be machined by RCM^2, but not the flat end mill roughing resulting in a shorter tool path for conventional flat end mill roughing.

To account for concavity consider a cone bound by Y_{max}, Y_{min}, with the tip at Z_{min} on the concave surface as seen in Fig. 17 with dashed lines representing cutting planes. The number of Y_x is found by reducing the crossing to a fraction of the total.
To determine the widths of the cone at each subsequent level simply recognize the similar triangles created by the cutting planes, Fig. 18. The width of the triangle (Y direction) is indirectly related to the number of cutting planes. The result is the concavity portion of the equation which is multiplied by the concavity factor. The concavity factor is simply the percentage of the surface that is concave, when viewed from the Z direction. A C_v of 1 represents a surface that is entirely concave, 0 represents no concavity and so forth.
RCM^2 crossings are determined by (3.8). Cut width for the surface is derived through the use of a simple algorithm that queries the surface for an average normal radius of curvature. Based on the information gathered, an average cut width for the entire surface can be determined.

$$Y_x = \frac{n_{cp}(X_{\max} - X_{\min})}{S_{o(RCM^2)}} \quad (3.8)$$

where $S_{o(RCM^2)}$ is the average RCM^2 step over determined from (3.4). The assumptions for RCM^2 point step over are valid for the use with the surface average tool orientation and surface average radius of curvature. The alternative with the fewest number of crossings will result in the shortest tool paths

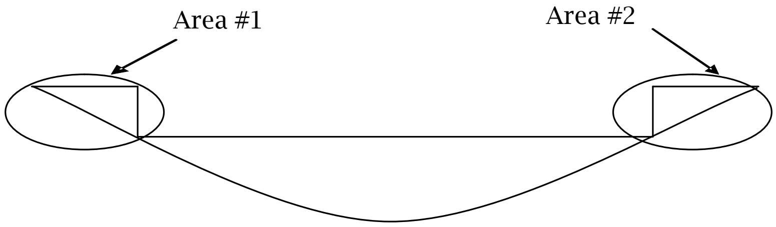

Fig. 16: Detail of concave curve.

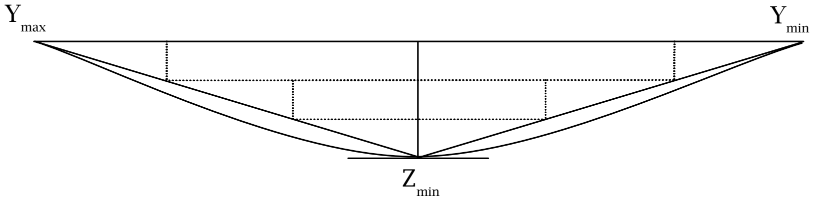

Fig. 17: Concave curve analysis.

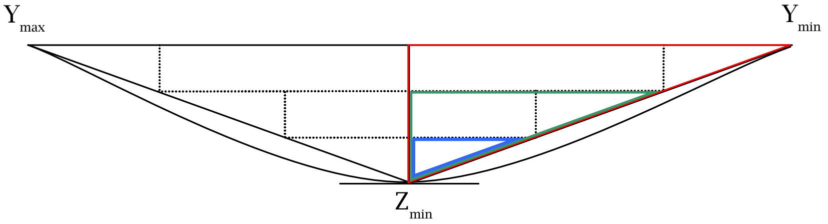

Fig. 18: Similar triangles.

and thus the shortest machining time. This simple comparison will allow users to determine the quickest alternative without computing tool paths for the entire surface.

3.6 Physical Testing

Since the early 90's when Jensen developed CM^2 algorithms, the typical machining test cases have focused on surfaces of the following classifications: concave, convex, saddle, and multiple curvatures. Four surfaces 5"x7"x3" will be machined using two different roughing methods, i.e. conventional planar passes followed by two semi-finishing (roughing) ball-end mill passes to remove the stair steps prior to a finish tool path verses RCM^2.

4. RESULTS

Conventional rough milling paths for the four test surfaces ranged from a low of $Y_x = 145$ predicted crossing of the saddle surface to a maximum of $Y_x = 230$ crossing of the multiple curvature. The predicted RCM^2 for these two surfaces were respectively $Y_x = 114$ and $Y_x = 106$. The concave surface had the fewest RCM^2 crossing $Y_x = 70$ and the convex surface had the most crossing at $Y_x = 127$ as contrasted with conventional crossings of $Y_x = 165$ and $Y_x = 141$, respectively.

These predicted crossing were a good measure of the actual tool path lengths that resulted in overall longer roughing tool motion and longer roughing cutting times for the conventionally mill surfaces. Machining parameters of spindle speed, feed rate, depth of cut, scallop height, etc. were set the same for both the RCM^2 and Conventional processes. Using our current implementation of RCM^2, where the curvature matched 5-axis tool motion sweeps from the edge of P_{ij} to the intersection of P_{zi}, at which point the cutter transitions into a 3-axis orientation and sweeps across P_{zi} until it reaches and intersection with a P_{ij} patch where it again begins sweeping across this patch in a curvature matched orientation until it reaches the outer edge, provided saving in the range of 20-30%. A superior algorithm is being worked on that will limit the 5-axis curvature matched machining to only those wedge shaped region of each planar pass as opposed to re-machining all of the previously exposed area of P_{ij}.

Figures 19 and 20 show the roughed surface after two roughing passes. Note the large step-over possible using RCM^2. Now compare this to the finished surface using ball end-mill in the left side of Figure 21, where the number of required passes is much higher.

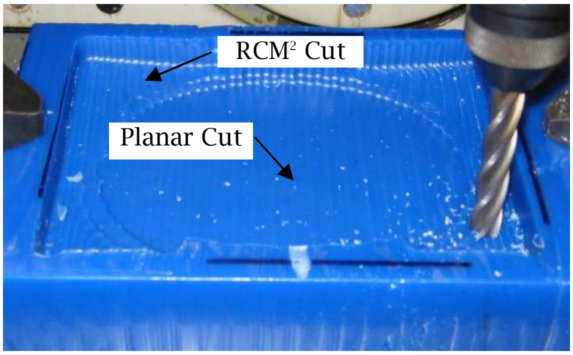

Fig. 19: Surface 1, RCM^2 plane 1.

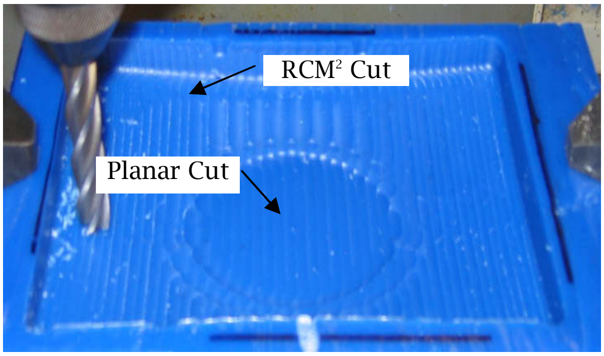

Fig. 20: Surface 1, RCM^2 plane 2.

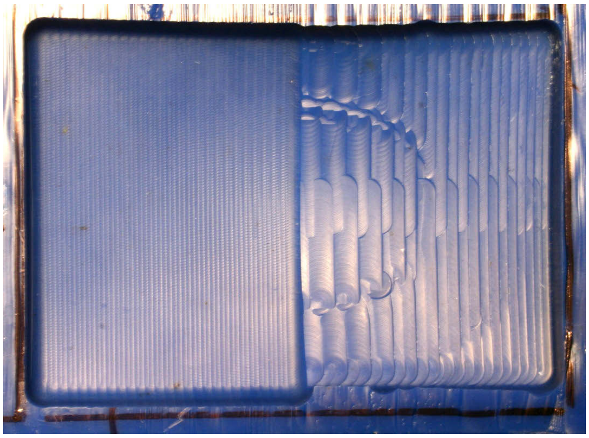

Fig. 21: Surface 1, with ball end-mill finished surface on left half.

5. CONCLUSIONS

Machining time is one of the most important aspects of the manufacturing process. The development of algorithms that machine surfaces in a more efficient manner is of benefit to a variety of industries. Enclosed surfaces present specific challenges, such as gouging into boundary walls and roughing multiple layers. This paper integrated CM^2 and planar roughing. The resulting algorithm, RCM^2, is a viable alternative to conventional roughing practices. The benefit of such an algorithm is a reduction in the overall machining time of a part or surface, depending on the surface features and distributed curvatures. The reduction in time comes specifically from the RCM^2 implementation.

Additional savings will result when an optimized RCM^2 algorithm for limiting the curvature matched motion to just the removal of the wedge shaped steps is implemented. It is projected that time savings for these four test cases could be as high as 60% over conventional roughing methods. The benefit of future refinement and research to the RCM^2 algorithm and CM^2 technology will bring about the commercialization of faster and more efficient machining technologies. Through the use of a prediction algorithm developed here, a machinist can determine when RCM^2 is more efficient than conventional practices.

Finishing path tolerance errors due to tool flexibility have not been addressed in this paper, although it is easily argued that curvature matched cutter diameters are generally much larger than ball end mill cutters for the same scallop height. Even though our proposed finishing depths are greater, our larger CM tools will reduce tolerance error due to tool flexing.

6. REFERENCES

[1] Jensen, C. G.: Analysis and synthesis of multi-axis sculptured surface machining, Ph.D. Thesis, Purdue University, 1993.

[2] Ernst, C.: Development and Implementation of Global Gouge Detection in Curvature Matched Machining, M.S. Thesis, Brigham Young University, Dec., 2000.

[3] Fallböhmer, P.; Altan, T.; Tönshoff, H.-K.; Nakagawa, T.: Survey of the die and mold manufacturing industry, Journal of Materials Processing Technology, 59, 1996, 158-168.

[4] Ye, X. G.; Fuh, J. Y. H.; Lee, K. S.: A hybrid method for recognition of undercut features from moulded parts, Computer-Aided Design, 33, 2001, 1023-1034.

[5] Choi, B. K.; Park, J. W.; Jun, C. S.: Cutter-location data optimization in 5-axis surface machining, Computer-Aided Design, 25(6), 1993, 377-386.

[6] Nishita, T.; Sederberg, T. W.; Kakimoto, M.: Ray tracing trimmed rational surface patches, International Conference on Computer Graphics and Interactive Techniques, 24(4), 1990, 337-345.

[7] Sederberg, T. W.; Christensen, H. N.; Katz, S.: Improved test for closed loops in surface intersections, Computer-Aided Design 21(8), 1989, 505-508.

[8] Pi, J.: Automatic Tool Selection and Tool Path Generation for Five-Axis Surface Machining, Ph.D. Dissertation, Brigham Young University, 1996.

[9] Li, S. X.; Jerard, R. B.: 5-axis machining of sculptured surfaces with a flat-end cutter, Computer-Aided Design, 26(3), 1994, 165-178.

[10] Vickers, G. W.; Dong, Z.; Li, H.: Optimal rough machining of sculptured parts on a CNC milling machine, Journal of Engineering for Industry, Transactions of the ASME, 115(4), 1993, 424-432.

[11] Suh, S. H.; Shin, Y. S.: Neural network modeling for tool path planning of the rough cut in complex pocket milling, Journal of Manufacturing Systems, 15(5), 1996, 295-304.

[12] Vafaeesefat, A.; El-Margahy, H. A.: Rough pocketing of multi-sculptured surface cavities, Proceedings of the Institute of Mechanical Engineers, 215(6), 2001, 745-753.

[13] Huang, Y. S.; Webster, P. D.; Dean, T. A.: An image detection approach to NC rough-cut milling from solid models, International Journal of Machine Tools and Manufacture, 36(12), 1996, 1321-1333.

[14] Kitchen, P. M.: Application of Curvature Matched Machining to Turbo Machinery, M.S. Thesis, Brigham Young University, August, 1996.

[15] Hill, P. M.: Curvature Matched Machining Applied to Real Part Surfaces and Materials Compared to Current Accepted Practices, M.S. Thesis, Brigham Young University, April, 2001.