Jurnal
Teknologi

Full Paper

Generalization of Mathematical Representation for Tool Path Based on Boundary Representation (BReps) Data Structure

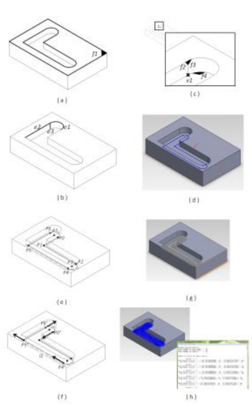
Mohd Azlan Suhaimi, Firdaus Mat Ghani, Jamaludin Mohd Taib*, Masine Md. Tap

Faculty of Mechanical Engineering, Universiti Teknologi Malaysia, 81310 UTM Johor Bahru, Malaysia

Article history
Received
15 April 2014
Received in revised form
24 November 2014
Accepted
15 December 2015

*Corresponding author
jamalt@fkm.utm.my

Graphical abstract



Abstract

A generic representation of linear and curvilinear entities with embedded motion attributes is possible via the integration of a Hermite curve and dynamic of motion. A Hermite curve will form the curve, while a dynamic of motion will position the vertices on the curves with the introduction of a delta distance (Δs). A three-phase algorithm is introduced to examine the applicability of generic representation in tool-path generation. The preprocessing stage will examine the input model prior to Phase I. Phase I will extract the BReps data, and these BReps data will be used in Phase II to develop the generalized mathematical representation of the tool path. Finally, the tool path is drawn in Phase III.

Keywords: Hermite curve, machining, tool path

Abstrak

Satu representasi generik untuk entiti-entiti linear dan lengkuk yang mempunyai sifat gerakan boleh dihasilkan dengan mengintegrasi Lengkuk Hermite dan dinamik gerakan. Lengkuk Hermite akan membentuk lengkung, manakala dinamik gerakan akan meyusunkan mercu pada lengkuk dengan menggunakan 'delta distance' (Δs). Algoritma tiga fasa diperkenalkan untuk memeriksa applikasi representasi ini terhadap laluan mataalat. Peringkat pra-proses membuat pemeriksaan terhadap model sebelum Fasa I. Fasa I akan mengekstrak data BReps dan data ini akan digunakan pada Fasa II untuk tujuan pembentukan formula matematik yang umum. Akhirnya, laluan mataalat dihasilkan pada Fasa III.

Kata kunci: Lengkuk Hermit, proses pemesinan, laluan mataalat

1.0 INTRODUCTION

Computer-aided process planning (CAPP) links computer-aided design (CAD) and computer-aided manufacturing (CAM) through feature recognition. Despite the fact that features can be used to set up the planning and suggest the process, the tool path must still be developed for machining purposes. This shows that these features are the transition stage between CAD and CAM. Thus a, seamless link between CAD and CAM may be achieved if this transition stage can be removed. One approach is to develop a system that is able to skip the feature-recognition stage by embedding the foundation of feature recognition without having to identify the feature.

In this approach, a tool path should be developed directly from the low-level product definition. In a feature extraction system, when a profile is identified, the feature-extraction technique is used to identify the features that emanate from the profile. Based on this information, the machining path can be developed based directly on the profile, without having to identify the feature. The identification of the constituent features can be skipped by replacing this with a generalized mathematical representation of the path that is generated by the various entities that constitute the profile.

Motion attributes can also be embedded in the mathematical representation. This will make the approach more flexible and practical when handling the machining process because this involves both path and speed.

Therefore, this paper will discuss the development and applicability of a system that generalizes a mathematical equation integrating a Hermite curve and dynamic of motion. A Hermite curve will be used to draw the curve, while a dynamic of motion will be used to position the tool based on the tool's assigned velocity.

In Section 2, this paper will review the related literature. Section 3 provides an overview of the overall methodology. Section 4 will discuss the mathematical formulation of the generic mathematical representation, followed by Section 5, which will contain the detailed algorithm adopted by the system to show its applicability. Illustrative examples are then provided in Section 6. Finally, the discussion and conclusion will be provided in Sections 7 and 8, respectively.

2.0 LITERATURE REVIEW

CAD and CAM may seem to be integrated, but in reality, they are not. Currently, feature-based computer-aided process planning (CAPP) is the link between the low-level product definitions in CAD and the down-stream activities in CAM. The feature definition proposed by Gindy [1] considered this issue and took into account the process planning. This taxonomy has become the foundation of feature definition for many feature recognition systems today.

Patil and Pande [2] developed intelligent feature process planning (IFPP) to automate the writing of the CNC coding. First, the feature definition will set up the planning and process required. Then, AutoPlan, which is one of the modules, will create the tool path and set the machining parameter.

Later in 2005, Woo *et al.* [3] used a different approach to feature definition to solve the problem of process planning. They used a hybrid feature recognition technique that integrates three methods of the feature recognizer. The integration of these methods produced unified hierarchical feature decomposition, which captures the machining precedence relationship.

Hou and Faddis [4] developed a system that integrated the machining features with the tool path. The machining features used here are similar to the STEP-NC features. This integration allows the tool path to be generated in an electronic format, which minimizes human intervention.

Lee *et al*. [5] proposed a system that develops the process planning for simple interacting features based on the feature definition. The process planning is sequenced according to the projective of the features. Khoshneviset *et al.* [6] proposed a new integrated process planning system that is comprised of three modules. The three modules are feature completion, process selection, and process sequencing.

In 2011, Xu *et al.* [7] reviewed CAPP and predicted that STEP and STEP-NC will be the future of process planning. The foundation of CAPP is to develop machining parameters based on the feature definition. However, complications may occur when the features interact.

An alternative approach to recognizing machining features has been proposed by considering the problem of machining to be the conversion of design features [8],[9]. However, the work is limited to identifying machining features. Deja and Siemiatkowski [10] used a part-design feature to search for the optimal process planning. A methodical approach was used to examine all possible process plannings based on part-design data via the development network of features using a feature precedence tree. An optimal solution can be achieved by examining the network. In the latest development, Chen and Du [11] used a 3D design environment to generate process planning via a rule-based system. The machining process ends when there are no more rules that can be matched.

Many CAM researchers have focused on the development tool path of surfaces, such as Bezier, B-Spline, or NURBS. The surfaces are created using Reverse Engineering (RE) by digitizing the part [12],[13]. The link between CAD and CAM can be regarded as nonexistent due to the reverse engineering technique used. Tapie *et al.* [14] had the opportunity to develop the machining path using a low-level production definition. They converted the part into the stl file format and used CAM software to identify the machining area on the stl file. Makhanov [15] reviewed the implementation of a geometric pattern for a five-axis machine.

In total, it seems that the CAPP system acts as the pivotal link between CAD and CAM because the feature definition can provide the planning sequence and process. On the extreme end, the link between the tool path and features for machining purposes within CAM itself seems to be invisible. Even though machining in CAM caters to the need of RE, machining is also required for any CAD part.

This paper will propose a system that can develop generic mathematical representations of tool paths from a BReps data structure. The advantage of this mathematical representation is that it is the most compact format for generating vertices for machining. Because this is an initial study, the study will focus on the development of a generic mathematical representation for tool path on the inner-profile machining area. Moreover the system will link CAD and CAM directly from the low-level product definition. Thus, the gap between CAD and CAM will become smaller, without having to use feature as a means of linking CAD and CAM.

3.0 SYSTEM ARCHITECTURE

The system developed here adopts a three-phase procedure, as shown in Figure 1. The procedures are as follows:

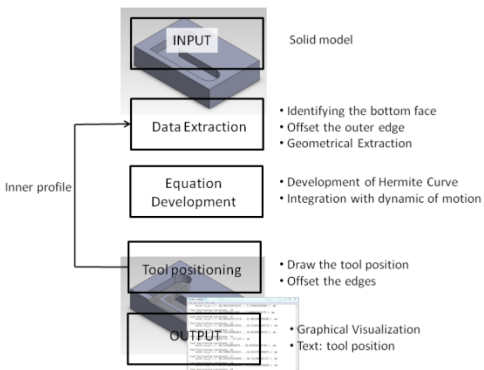


Figure 1 System Architecture

3.1 Phase 1: Data Extraction

This phase will examine the BReps data of the part to identify the bottom face. When the bottom face is identified, the system uses the built-in offset command to offset the outer edges of the bottom face. The offset distance is based on the tool radius. The offset entities are, in fact, the tool path used to machine the part.

3.2 Phase 2: Development of the Generic Mathematical Representation

Phase 1 provides the tool path. In phase 2, the system will examine the topological and geometrical data of the offset path. For each entity that forms the tool path, the system will retrieve information on the entity type and the coordinates of the important points on the curves. Using the coordinates of the points and the entity type, all four parameters of the Hermite curve are now known. The velocity of the tool is used, and the length of the entity will be the input for the dynamic of motion to create the generic mathematical representation. At the end of this phase, the generic mathematical representation is developed.

3.3 Phase 3: Tool Positioning

Finally, after the generic mathematical representation is developed, the vertices on the curves are, in fact, the position of the tool at the specified time. Interfacing with a built-in graphical application programming interface (API) enables the tool path to be presented as circles, and the position of the tool can be visualized. In this paper, the time interval of display has been set to 0.1 seconds. Then, the system will return to Phase 1 and create another inner path. The creation of the inner path ends when the system is alerted by the API command that the inner loop cannot be formed, and the final path will be one straight edge that goes through the middle of the machining area.

4.0 DEVELOPMENT OF THE MATHEMATICAL FORMULATION

This section will discuss the integration of the Hermite curve and dynamic of motion to develop the generic mathematical representation. Then, it discusses the generic characteristics of the mathematical representation that covers the entity types and path types.

4.1 Integration of Hermite Curve and Dynamic of Motion

The integration of the Hermite curve is carried out via the introduction of delta distance Δs, which replaces the free parameter of the Hermite curve. Therefore, the generic equation of curve P with the condition of delta distance Δs ∈ (0,1) is shown in Equation 1 [16].

$$P(\Delta s) = \begin{bmatrix} \Delta s^3 & \Delta s^2 & \Delta s & 1 \end{bmatrix} \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} P0 \\ P1 \\ P0' \\ P1' \end{bmatrix} \quad (1)$$

where P0: start point
P1: end point
P0': tangent of start point
P1': tangent of end point

Delta distance Δs is the ratio of the distance travelled at time t and the total distance.

$$\Delta s = \frac{s(t)}{s_{total}} \quad (2)$$

Because the tool velocity is constant, the formula for distance travelled at time t will be

$$s(t) = ut \quad (3)$$

where u : velocity of the tool
t : time

4.2 Generic Mathematical Representation

The mathematical representation is generic enough to cover various entities and paths. For this research, the entities will include linear and curvilinear entities and paths. The generic characteristics of the mathematical representation will be discussed in the following sections.

4.2.1 Linear Properties

Straight lines are considered to be linear entities. The start point (P0) and end point (P1) are the start and end points of the curve, respectively. The tangent of the end points (P0' and P1') can be calculated using the following formula:

$$P0'.x = P1'.x = \frac{P1.x - P0.x}{\sqrt{(P1.x - P0.x)^2 + (P1.y - P0.y)^2}} \quad (4)$$

$$P0'.y = P1'.y = \frac{P1.y - P0.y}{\sqrt{(P1.x - P0.x)^2 + (P1.y - P0.y)^2}} \quad (5)$$

The end points and their tangents are the input for the drawing of the path. Then, the delta distance Δs will place the vertices of the curve.

4.2.2 Curvilinear Properties

Various curves can be created by changing the tangents of both the start and end points, without changing the position of the start and end points. Figure 2a shows some examples of curvilinear curves, such as a line, arc, and semi-circle, which are formed by varying the direction of the vector. A curve can also be manipulated without changing the position of the points or the directions of vector. This is done by manipulating the magnitude of the vector, as shown in Figure 2b. By increasing the magnitude of the vector, the curvature of the curve will become larger. If the magnitude of the vector is equal to the radius of the arc, a quarter-circle is formed.

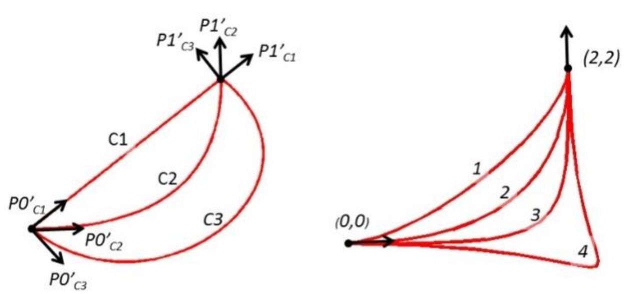


Figure 2 Curves generated by varying the (a) tangents and (b) magnitudes of equal vectors

4.2.3 Motion Properties

The Hermite curve will draw the curve, while the dynamic of motion will place the vertices on the curve. Regardless of whether there is linear or curvilinear motion, linear motion is used as the input. If the tool path is curved, the angular characteristics of the motion will be handled by the Hermite curve. Figure 3 shows a number of vertices plotted using a generalized mathematical equation and the angle of motion equation. The crossed (+) vertices are developed using Equation (1), whilst the circular (o) vertices are developed using the angular motion equation, as in Equation (6), using a constant time increment. Because the Hermite curve will handle the curvilinear attribute of the motion, even though the input uses linear motion, the position of the vertices from the generalized mathematical equation are at the same position as the vertices using the angular motion.

$$\theta_f = \theta_o + \omega t + \frac{1}{2}\alpha t^2 \quad (6)$$

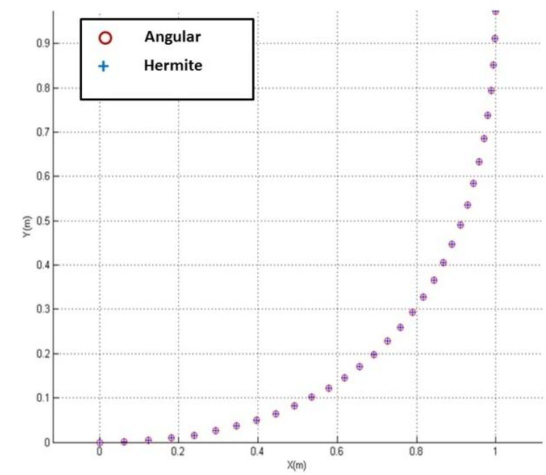


Figure 3 Curvilinear motion using an angular dynamic of motion and generic mathematical equation

5.0 SYSTEM DEVELOPMENT

This section discusses, in detail, each step in the development of the system.

5.1 Data Extraction

This phase will extract the BReps data of the part. The aim of the data extraction is to identify the bottom face of the feature. Based on the bottom face of the feature, the outer edges of the face will be offset by the tool radius to form the tool path. Finally, the geometrical data of the entities that forms the tool path are extracted.

5.1.1 Identification of Bottom Face of the Feature

Because the feature is limited to the pocket, the identification of the bottom face is based on the detection of the profile that emanates the features. In this case, the profiles that emanate from the features are inner loops. Inner loops cause two or more loops on the faces. Using this as the basis for detection, the system retrieves the number of loops on the face, and if the number of loops is two or more, the system will proceed to detect the bottom face of the feature by first identifying the inner loop.

For each inner loop, the system will proceed to detect one of the vertices. Once one of the vertices is detected, the system will search three enumerated edges. Because two of the edges belong to the inner loops, the edge that does not belong to the inner loop will be the edge with the endpoint belonging to the bottom face, as shown in Figure 4. When the vertex is known, the system will search the incident faces of the vertex. The face that does not have edges will be the bottom face.

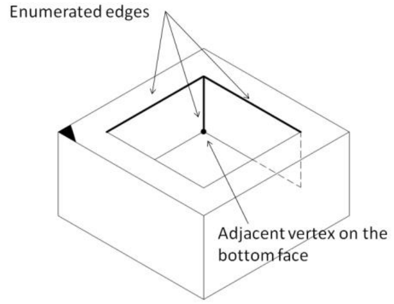


Figure 4 Adjacent vertex on the bottom face

5.1.2 Offsetting the Outer Edges of the Bottom Faces

When the bottom face is detected, the outer loop of this face can be used as the reference tool path. The outer loop is, in fact, the final edge after machining. When a tool is used to machine, the position of the tool must be offset inward by the radius of the tool. Therefore, the tool path will be the offset inward from the outer edges by the radius of the tool. Hence, when a tool is selected, the system will allow the user to input the tool diameter using a user-input dialog box.

The input on the tool diameter will be cross-checked with the tool size available for the machine. Therefore, a tool diameter database should be created. If a tool with the specified diameter is not available, the system will choose another tool with a smaller diameter from the database.

5.1.3 Extraction of the Geometrical Data

Currently, the tool path, which is the offset of the outer edges, is comprised of a number of entities. Because the development of the generic equation relies on the geometrical properties of the entities, the extraction of the geometrical properties for each compound entity must be carried out, and these data will be passed to the next stage.

To extract the geometrical data, the behavior of the vertices on the tool path should be understood first. The behavior of the vertex arrangement is shown in Figure 5. The loop starts from the first vertex and travels in a clockwise direction. In the case of geometrical properties, a line is described by its start and end points, while an arc is defined by three vertices: the start, center and end points.

As far as the arrangement of the geometrical properties are concerned, the geometrical properties of the entities are stored separately. This is because the entities are wireframe representations, which causes each entity to have its own topology and geometrical data. However, Solidworks provides the arrangement of the vertices that form the loop and the geometrical properties of the entities to be arranged according to their positions in the loops.

As far as the representation of the vertices is concerned, the coordinates of the vertices are stored in a structured format as *s_ver*, and the vertex is stored as *(s_ver.x, s_ver.y, s_ver.z)*

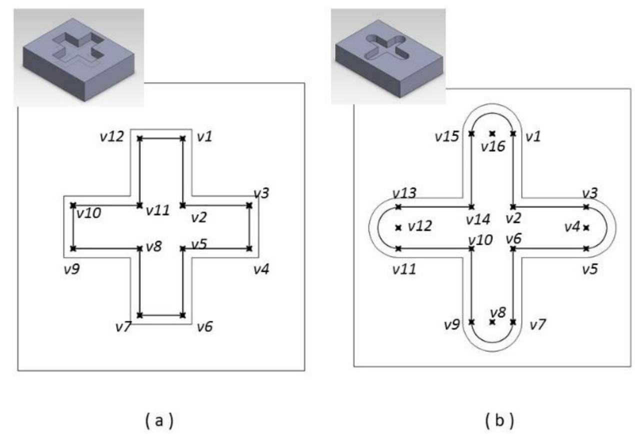


Figure 5 Sequence of vertices

In the case of lines, arcs and circles, their information is stored in a variable comprised of 16 arrays *(s_var)*. The first six sub-variables *(s_var(0) to s_var(5))* will store the attributes of the entities, such as colour, layer, etc. Then, the remaining variables *(s_var(6) to s_var(15))* store the geometrical properties according to the type of entities. The arrangement of the geometrical properties for each entity is as follows:

Line: start point (x1, y1, z1) = *(s_var(6), s_var(7), s_var(8))*
End point (x2,y2, z2) = *(s_var(9), s_var(10), s_var(11))*
Arc: start point (x1, y1, z1) = *(s_var(6), s_var(7), s_var(8))*
End point (x2,y2, z2) = *(s_var(9), s_var(10), s_var(11))*
Center point (xc, yc, zc) = *(s_var(12), s_var(13), s_var(14))*

5.2 Development of the Generic Equation

The foundation of the integration is that a Hermite curve will draw the curve, while the dynamic of motion will position the vertices on the curves.

5.2.1 Development of the Hermite Curve

Currently, the system has the information for all the entities that form the tool path. The following shows the determination of the four parameters for the Hermite curve.

Straight Line

Take a line with start point P0 (x0, y0) and end point P1(x1, y1). Both points can be used as the input in the development of the Hermite curve. However, two more inputs are required: the tangents of the points, which can be calculated based on these points using equation (7). Figure 6a shows the development of the Hermite curve using the coordinates and the tangents of both points.

$$P0' = P1' = \frac{x1 - x0}{\|P\|}\hat{i} + \frac{y1 - y0}{\|P\|}\hat{j} \tag{7}$$

Arc

In the case of an arc with start point P0(x0,y0), center P1(x1, y1) and end point P2(x2,y2), the coordinates of the start and end points can be used directly as the input for the start and end points of the Hermite curve. However, the other two inputs, which are the tangents of the start and end points, must be calculated from the coordinates of the start, center, and end points.

To calculate the tangents of the start and end points, the normal vector and the dot product of the tangent (v) and normal (u) are used to form the basic simultaneous equations used to solve the tangent of the start point. To demonstrate the method used to calculate the tangent, a point on arc P (xp, yp) and a center point of arc PC (xc, yc) are used.

The normal of the point can be calculated using equation (8).

$$u = \frac{x_C - x_P}{\|R\|}\hat{i} + \frac{y_C - y_P}{\|R\|}\hat{j} \tag{8}$$

The tangent of the point can be determined using equation (9).

$$v = \frac{V_x}{\|R\|}\hat{i} + \frac{V_y}{\|R\|}\hat{j} \tag{9}$$

Because both vectors are perpendicular, the dot product of the tangent and normal will be as follows:

$$u \cdot v = \cos(90^\circ)$$

$$u.v = 0 \tag{10}$$

Based on Equations (8) and (10), the tangent of the start point will be as follows:

$$V_x = -(y_P - y_c) \tag{11}$$

$$V_y = x_P - x_C \tag{12}$$

Replacing the point on the arc with P0 and P2 will lead to calculating the tangents of the start and end points, respectively, as illustrated in Figure 6b. When the tangents of the start and end point are known, the Hermite curve for curvilinear entities can be drawn.

5.2.2 Integration with Dynamic of Motion

When the Hermite curve has been drawn, the next step is to integrate it with the dynamic of motion. First, the user must input the velocity of the tool in meters per minute. Because the tool has a constant velocity, the initial velocity at the start point will be equal to the final velocity at end point of the curve.

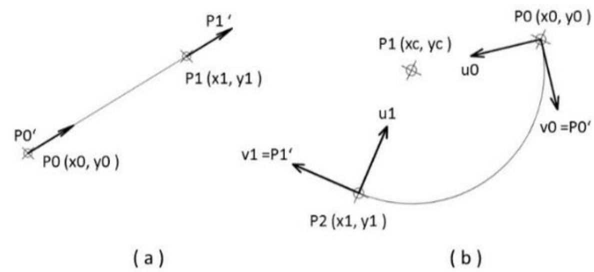


Figure 6 Point and tangent definition for Hermite Curve

As previously mentioned, the delta distance (Δs), as in Equation (2), will integrate the Hermite curve with the dynamic of motion. The delta distance is the ratio of the distance travelled at time t and the total distance travelled. Because the tool travels at a constant speed, equation (3) will calculate the distance as a function of time. The next step is to calculate the total distance traveled. The total distance travelled can be calculated using the Hermite curve, as in Equation (1), but the delta distance is replaced by free parameter t. By setting the value of the free parameter as 0 to 1, with an increment of 0.1, the total distance can be calculated using the total distance of the individual distance calculated from one vertex to the next vertex on the curve. The equation used to calculate the total distance is shown in Equation (10).

$$s_{total} = \sum_{i=1}^{i} \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \tag{10}$$

When the delta distance is calculated, it will be inserted into Equation (1), and the curve created by the integration of the Cubic spline and dynamic of motion is plotted. A vertex is plotted every 0.1 seconds until time is equal to the maximum tool travel from the start point to the end point. A similar process is repeated for other entities and ends when all entities are examined.

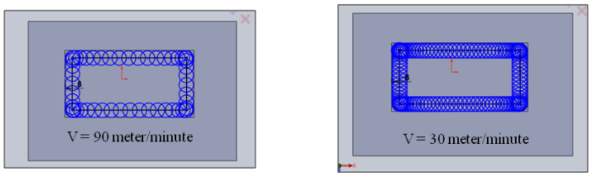


Figure 7 Tool position based on tool velocity

5.3 Tool Positioning

The following section will discuss the detailed procedure for the tool-positioning stage.

5.3.1 Drawing the Tool Based on the Position

The drawn vertices from the previous stage are, in fact, the position of the tool at the specified time. Using built-in API function, a circle with the tool diameter is drawn to represent the tool position. Figure 7 shows the tool position of the same inner profile with different tool velocities. A slower tool velocity will cause tool positions to be closer together than the tool positions at a higher tool velocity.

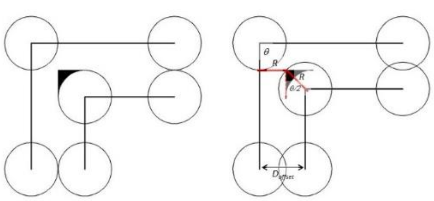
5.3.2 Offsetting the Inner Path

The next stage is to create the inner tool path. This is carried out by offsetting the tool path. Figure 8 shows the method used to calculate the offset distance to ensure the accuracy of the cut profile. When the offset distance is set to the tool diameter, some part, which has been highlighted in Figure 8a, will not be machined. Therefore, the tool must cover this area, as shown in Figure 8b. Therefore the maximum distance for offsetting the tool path is as in Equation (11).

$$D_{offset} = R_{tool} + R_{tool}\cos(\theta/2) \tag{11}$$

When θ = 90º,

$$D_{offset} = 1.7071R_{tool}$$



(a) (b)

Figure 8 Maximum offset distance

6.0 ILLUSTRATIVE EXAMPLES

The discussion in this section is illustrated by three illustrative examples. The focus of the first example is to discuss the details of the procedure, while the discussion of the second and the third illustrative examples will focus on the capabilities of the system.

Figure 9 shows the features of a part that will be used as an example to demonstrate the system. The part is comprised of a pocket and inner loop comprised of linear and curvilinear entities. During the first stage, which is data extraction, the system will search all the faces and identify those faces that have more than one loop. In this case, face *f1*, as shown in Figure 10a, is the only face that has more than one loop. Then, the system searches the inner loop and examines one of the vertices on the inner loop. Based on the vertex of the inner loop, the system searches the incident edges (*e1*, *e2*, and *e3*), as shown in Figure 10b. Then, the system looks for the vertex on the incident vertex that does not belong to the inner loop. In this example, vertex *v1* is connected to the edges of the bottom face. However, there are three incident faces (*f2*, *f3*, and *f4*) of the vertex, as shown in Figure 10c. The bottom face (*f4*) is differentiated when none of the edges of the faces are enumerated edges (*e1*, *e2*, and *e3*).

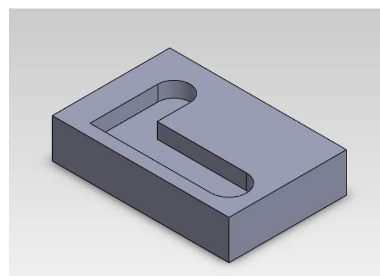


Figure 9 Feature for Example I

When the bottom face is differentiated, the system will offset the outer edge by the tool diameter. The tool diameter is set by the user using a user-input pop-up menu, and in this example, 10 mm is set as the input. The system will now-cross check with the tool database to ensure that the requested 10 mm tool is available. Then, the outer edges of the face are offset by 5 mm, which is the radius of the tool (refer to Figure 10d).

When the offset of the outer edge is created, the system will traverse the loop and, at the same time, store the entity types and the vertices of the entities. Figure 10e shows the counter-clockwise direction of the loop and the vertices of the entities.

When the entities and their vertices are identified, the system will proceed to the second stage, which is the generation of the generic equation. Using the formulae discussed previously, all four parameters for the Hermite curve can be determined. For instance, line *l1* will have *P4*, *P5*, *P4'*, and *P5'* and the semi-circular arc *a1* will have *P6*, *P0*, *P6'*, and *P0'*. The tangents (*P4'*, *P5'*, *P6'*, and *P0'*) are shown in Figure 10f. With the velocity of 30 m/min, the tool is drawn for every 0.1-second increment and, the tool path is shown in Figure 10g.

Stage 3 will draw the circle representing the tool position at the specified time, and another inner tool profile will be created. The process ends when all the tool paths are drawn, as shown in Figure 10h.

Figures 12a and 12b show two more examples. The first example (Figure 11a) shows similar method that can be used for a non-orthogonal inner profile. The second example (Figure 11b) illustrates how the system can be used for the inner profiles of multiple planes.

7.0 DISCUSSION

The proposed system developed the tool path using the integration of the Hermite curve and the dynamic of motion, whereby the delta distance Δs replaces the free parameter of the Hermite curve. The Hermite curve will draw the curve, while the dynamic of motion will place the vertices on the curve based on the velocity of the tool. The development of the generalized mathematical representation has the following advantages.

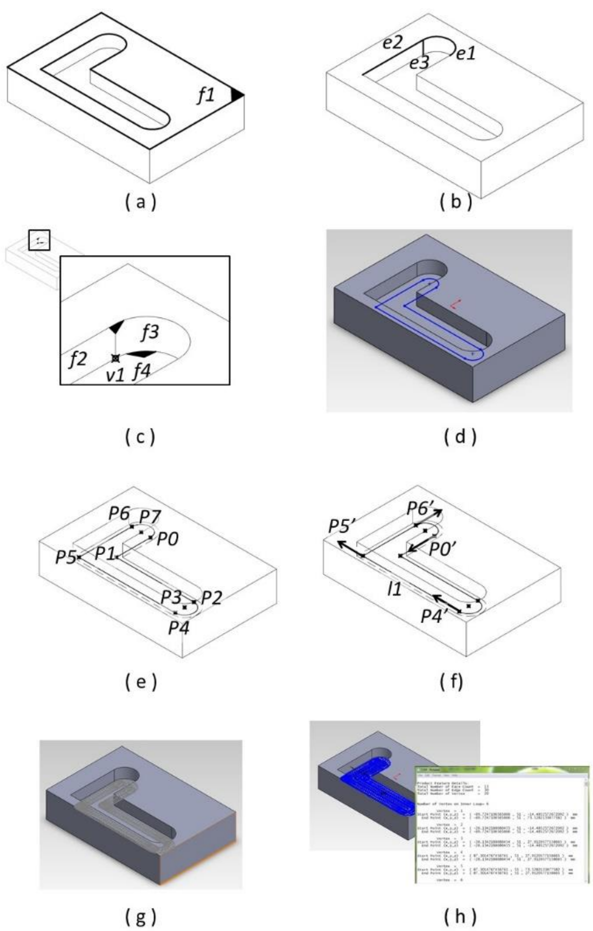
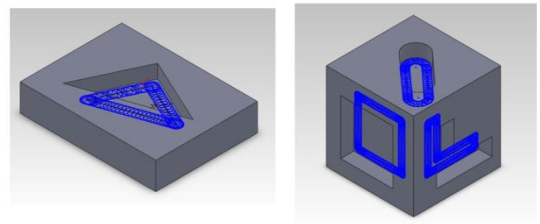


Figure 10 Illustrative Example I



(a) (b)

Figure 11 Illustrative Examples II

Better Integration of CAD and CAM

The purpose of the system developed is to achieve closer integration between CAD and CAM. By embedding the feature-recognition principle without having to recognize the feature, the machining process can be carried out directly using the low-level product definition. Therefore, better integration of CAD and CAM can be achieved.

Generic Mathematical Representation

The mathematical representation developed here is generic enough to cover a number of curves, such as those shown in this study. The generic representation can be applied to all lines and arcs, regardless of their orthogonality. Furthermore, the generic representation will handle both linear and angular motion as linear problems because the Hermite curve will transform the linear motion into curvilinear motion.

Mathematical Representation With Embedded Velocity

The dynamic of motion will determine the position of the tool at a specific time. Therefore, by setting different tool velocities, the visualization of time taken by the machine can be translated as the density of the circles on the tool path. The density of the circles can be used as an inference to show the time required to machine the part.

8.0 CONCLUSION AND FUTURE WORK

In this study, a generalized mathematical representation for the development of tool paths from BReps is used for lines and arcs. Even though the entities are limited, the main focus of the study is to show the applicability of the generic mathematical representation to create tool paths, with the aim of improving the integration between CAD and CAM. The system developed in this study directly creates the tool path by embedding the foundation of feature recognition. This will eventually reduce the dependency of feature recognition when linking CAD and CAM.

Because the focus of the study is to show the applicability of the generic mathematical representation to bring CAD and CAM closer together, the study was carried out on two main entities (lines and arcs). However, this study can be extended to other types of entities or curves using a cubic spline. A cubic spline is, in fact, the combination of a number of Hermite curves. A cubic spline will solve the continuity of each Hermite curve to ensure that both Hermite curves at the joint will have equal curvatures and share the same center of the radius.

The research can also be extended to a non-orthogonal bottom face. This can be done by adding another Z component of the coordinate. Because the basic representation of the curve is parametric, the Z component can be inserted directly, with few changes to the current representation. This, in fact, increases the applicability of the representation from a 3-axis to a 5-axis CNC machine.

Acknowledgement

The authors would like to thank the Ministry of Education, Malaysia, and the Research Management Center, Universiti Teknologi Malaysia, for the FRGS grant (Vote 4F335) to conduct this research.

References

[1] Gindy, N. N. Z. 1989. A Hierarchical Structure for Form Features. *International Journal of Production Research*. 27(12): 2089-2103.

[2] Patil, L. and S. S. Pande. 2002. An Intelligent Feature-Based Process Planning System for Prismatic Parts. *International Journal of Production Research*. 40(17): 4431-4447.

[3] Woo, Y., E. Wang, Y. S. Kim, and H. M. Rho. 2005. A Hybrid Feature Recognizer for Machining Process Planning Systems. *CIRP Annals-Manufacturing Technology*. 54(1): 397-400.

[4] Hou, M. and T. N. Faddis. 2006. Automatic Tool Path Generation of a Feature-Based CAD/CAPP/CAM Integrated System. *International Journal of Computer Integrated Manufacturing*. 19(4): 350-358.

[5] Lee, H. C., W. C. Jhee, and H. S. Park. 2007. Generative CAPP through Projective Feature Recognition. *Computers & Industrial Engineering*. 53(2): 241-246.

[6] Khoshnevis, B., D. N. Sormaz, and J. Y. Park. 1999. An Integrated Process Planning System Using Feature Reasoning and Space Search-Based Optimization. *IIE Transactions.* 31(7): 597-616.

[7] Xu, X., L. Wang, and S. T. Newman. 2011. Computer-Aided Process Planning–A Critical Review of Recent Developments and Future Trends. *International Journal of Computer Integrated Manufacturing*. 24(1): 1-31.

[8] Sormaz, D.N. and C. Tennety. 2010. Recognition of Interacting Volumetric Features Using 2D Hints. *Assembly Automation*. 30(2): 131-141

[9] Zheng, Y., J. Mohd Taib, and M. Md Tap. 2012. Decomposition of Interacting Machining Features Based on the Reasoning on the Design Features. *International Journal of Advanced Manufacturing Technology*. 58(1): 359-377.

[10] Deja M., and M. S. Siemiatkowski. 2013. Feature-Based Generation of Machining Process Plans for Optimised Parts Manufacture. *Journal of Intelligent Manufacturing.* 24(4): 831-846.

[11] Chen H., and X. Du. 2014. Integrated Process Planning Method for 3D Designing Environment. *Applied Mechanics and Materials.* 496-500: 2316-2319.

[12] Yin, Z. and S. Jiang. 2003. Automatic Segmentation and Approximation of Digitized Data for Reverse Engineering. *International Journal of Production Research*. 41(13): 3045-3058.

[13] Chui, K. L., K. M. Yu, and T. C. Lee. 2002. Direct Tool-Path Generation from Massive Point Input. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*. 216: 199-206.

[14] Tapie L, B. Mawussi, and A. Bernard. 2012. Topological Model for Machining of Parts with Complex Shapes. *Computers in Industry.* 63(5): 528-541.

[15] Makhanov S. S. 2010. Adaptable Geometric Patterns for Five-Axis Machining: A Survey. *International Journal of Advanced Manufacturing Technology*. 47(9): 1167-1208.

[16] Fouladinejad, N. 2011. Intelligent Traffic Flow Simulation System on a Small Scale Environment. *M. Eng Thesis*. Universiti Teknologi Malaysia.