COMPUTER-AIDED DESIGN & APPLICATIONS, 2016
VOL. 13, NO. 6, 863–871
http://dx.doi.org/10.1080/16864360.2016.1168235

NC simulation for adaptive look-ahead interpolator with on-line collision detection

Hong-Tzong Yau, Chia-Hao Chang, Tsan-Jui Yang and Jiun-Ting Lin

National Chung Cheng University

ABSTRACT
This paper proposes a numerical control (NC) machining simulation system with adaptive look-ahead interpolator and on-line collision detection function. Adaptive collision detection between tool and machine can be executed synchronously to avoid interference during the machining process. Traditional multi-axis machines are used to process free-form surfaces. However, determining how to check whether or not the machining path is correct remains a challenge. In path planning, determining how to avoid the wrong path resulting in the waste of materials and machine damage has become an important issue in the industry. Currently, most of planning use pre- processing simulation prior to actual machining aims to eliminate the wrong path that may result in collision. In this study, the efficiency of a developed multi-axis collision detection system is based on the computational complexity of possible collision parts. Hence, during simulation, the number of collision detections is variable and depends on the time interval of each path; this ensures that the most sophisticated collision detection can be done in limited processing time. The NC simulation structure with adaptive look-ahead interpolator and on-line collision detection proposed in this study can be applied to all kinds of processing machines. Through the results of this study, the value of the controller can be greatly enhanced to achieve the purposes of real-time motion planning and adaptive collision detection.

KEYWORDS
Multi-axis machine simulation; look-ahead function; adaptive path interpolation; collision detection

1. Introduction

CNC machining has become a major process for producing precise components for manufacturing industries. As products become more complex in terms of geometry, CNC machining also evolves from 3-axis/5-axis machining to more complex multi-axis turn-mill machining. In addition, to achieve flexible manufacturing and increase automation level, a multi-function CNC machining center further integrates machining, grinding, polishing and even laser sintering (additive manufacturing) into a single CNC machine platform. However, as the complexity and cost of the machine increase, the issue of on-line collision detection and avoidance becomes ever more important.

The issue of collision detection and avoidance is particularly critical for high-speed multi-axis machining. With the advancement of the CNC and machine tool technology, multi-axis machining has become the main processing method for producing everyday products. Compared with three-axis machining, multi-axis machining offers high precision, efficiency and better surface quality in short machining periods. While this process is used for machining complex and contour free-form products, due to the complex tool path, interference or collision may easily occur, with subsequent damage to the machine or parts. Traditionally, off-line simulation is used to detect collisions prior to machining, but this procedure is out-of-date because the demand for real-time simulation of the CNC machining process is increasing, for the aim of saving time and money.

The CNC controller plays the key role in multi-axis machine tools. The function of the controller includes NC code interpretation, look-ahead feedrate scheduling, tool-path interpolation and servo motion control. To maintain feedrate continuity and avoid undesired vibration, the controller has to look ahead in the path and plan the feedrate. Kim [6] applied feedrate planning to produce smoother interpolation processing. Han [4] proposed the look-ahead interpolation of a high-speed machining algorithm; the controller could upgrade the cutting speed without increasing the hardware cost. Yong and Narayanaswami [17] detected the sharp corner in an off-line system to reduce contouring errors. However, none of these approaches has integrated the look-ahead interpolation with the collision detection process.

CONTACT Hong-Tzong Yau imehty@ccu.edu.tw

In a traditional controller, linear and circular NC segments are mostly used. When a sculptured surface is being machined, a great amount of linear segments are generated to approximate the contour geometry. As a result, the processing workload is significantly increased [7] [8]. Koren [9] used parametric spline to represent the machining curve; he solved the problem of using a great amount of short lines to fit the curve. Yeh and Hsu [16] proposed adaptive feedrate interpolation to control the chord error and improve the precision of the parametric interpolation. Lo [10] used iso-feedrate to derive the implicit function and parametric curve interpolation; compared with linear interpolation, it improves the precision in curve interpolation. In recent years, NURBS interpolation has gained much popularity in PC-based controller.

Collision detection is widely used in computer graphics and games. In recent years, it is used in the virtual machine tools to simulate the movement of CNC machines or robotic arms. Cohen[2] used the axis-aligned hexahedral to simulate the complex machine tool model. Gottschalk[3] used the Oriented Bounding Box(OBB) and the separating axis to detect the collision of the models. Ilushin[5] used the ray casting and the symmetry of the tool to subdivide the workspace and detect the collision of the workpiece and tool. It did not detect the collision of the machine tools. Tang[13] used a two-step approach to detect the collision. First, using the sphere to simulate the models and rapidly detect the collision. If the collision is possible, use the octree to subdivide the sphere models. Second, Slip the model to analyze the position of the model. Ahmad[1] used the rectangular enveloped safe and efficient trajectory (RESET) based on snakes and ladders analogy for production trajectory (SLAPT) to detect the collision of turn-mill machines in the two dimension plane.

The aim of this work is to develop an on-line collision detection system that can be integrated with a CNC controller to look ahead in the multi-axis tool path in real-time and stop the machine execution when a possible collision is detected before it happens

2. System structure

Fig. 1 shows the research flowchart proposed by this paper. First of all, the virtual multi-axis machine needs to be modeled by analyzing the kinematics and dynamics of the CNC machine. Using this virtual machine information, the second step is to perform the collision efficiency analysis. We separate the machine moving axes into relative pairs. Those pairs which may not collide with each other are excluded from the collision detection during simulation. This will significantly increase the collision detection efficiency. Moreover, the look-ahead function will be used to calculate the feedrate intelligently by reading a block of data before it is ready to be executed. Then, the relations of feedrate and time can be established in advance by calculating the acceleration and deceleration of machining in path planning. Finally, the machining path can be interpolated by Taylor series according to

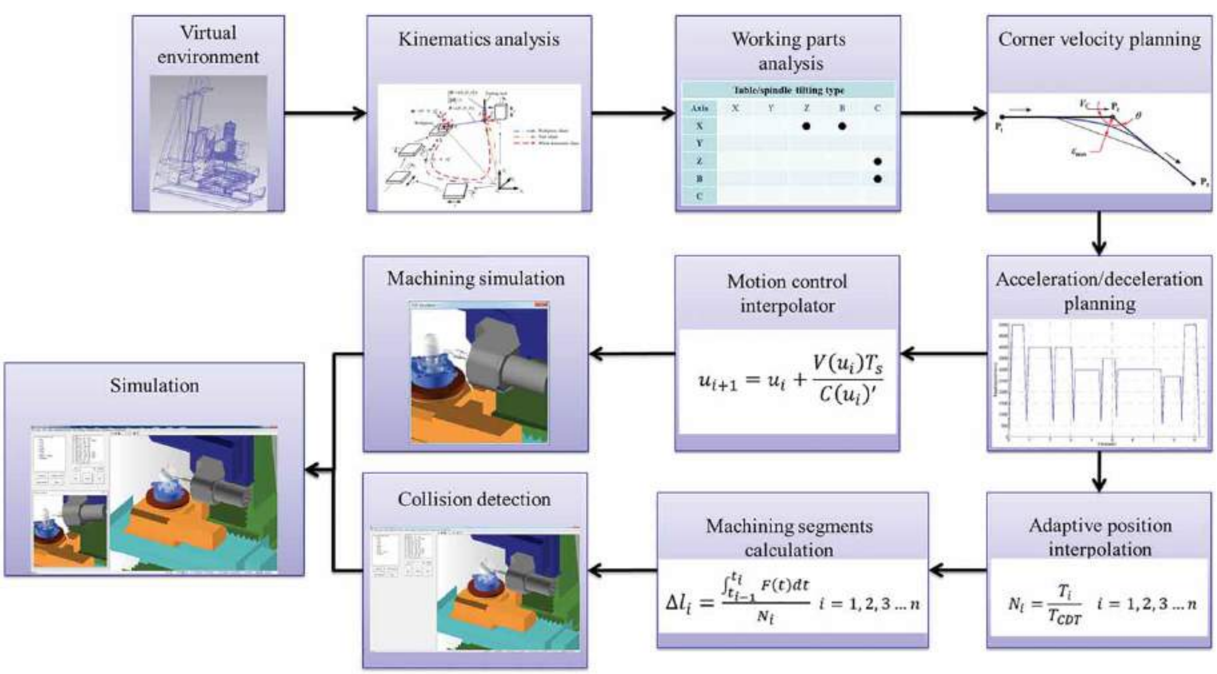

Figure 1. Flow chart of the system.

both the feedrate and time information obtained from the look-ahead function. For each path segment, knowing the speed and distance, the algorithm will calculate the suitable collision detection number (or detection steps) adaptively. The collision detection and simulation will therefore be carried out at a safety distance ahead of the real machining steps. The safety distance is the distance that is at least larger than the minimum distance in which the machine can stop, under the maximum feedrate. This proposed intelligent machining system equipped with adaptive look-ahead function and on-line collision detection function can be integrated with a machine tool controller for on-line simulation and collision detection/avoidance to protect expensive multi-axis machine tools from being damaged by avoidable collisions.

3. The virtual simulation system of machine tools

Before creating the virtual simulation system, it is necessary to analyze the kinematics of machine tools and look ahead on the tool path. In the kinematic analysis, the location of the rotary axes affects the transformation of the cutter location and NC point. The look-ahead system maintains the continuity in the tool path and provides the feedrate and time to the path interpolation.

3.1. Kinematic configuration analyses

DMU 50 eVolution five-axis milling machine is shown in Fig. 2. It is a table-tilting type machine. A special feature of the machine is that the rotary axes (B and C axis) are non-orthogonal. Based on [12], the forward kinematic analysis is used to calculate the CL (x, y, z, i, j, and k) data the machine axis variable (e.g X, Y, Z, B, and C). It can be expressed by Eq.(3.1)-(3.6).

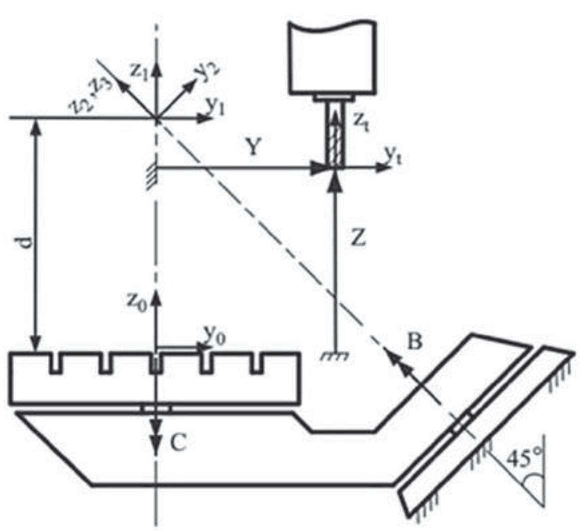

Figure 2. The configuration of DMU 50 eVolution five-axis milling machine [9].

$$i = \frac{1}{2}\left(\sqrt{2}\cos C \sin B - \sin C \cos B + \sin C\right) \quad (3.1)$$

$$j = \frac{1}{2}\left(\sqrt{2}\sin C \sin B - \cos C \cos B + \cos C\right) \quad (3.2)$$

$$k = \frac{1}{2} + \frac{1}{2}\cos B \quad (3.3)$$

$$x = \frac{\sqrt{2}}{2}[X \sin C + (Y + Z - d)\cos C]\sin B + \frac{1}{2}[-Y + Z - d - (Y + Z - d)\cos B]\sin C + X \cos C \cos B \quad (3.4)$$

$$y = \frac{\sqrt{2}}{2}[(Y + Z - d)\sin C - X\cos C]\sin B + X \sin C \cos B + \frac{1}{2}[(Y + Z - d)\cos C + (Y - Z + d)]\cos B \quad (3.5)$$

$$z = \frac{1}{2}\left[-\sqrt{2}X \sin B + (Y + Z - d)\cos B - Y + Z + d\right] \quad (3.6)$$

The inverse kinematic analysis can be expressed by Eq. (3.7) (3.11):

$$B = \cos^{-1}(2k - 1) \quad (3.7)$$

$$C = \tan^{-1}\left\{\left[(1 - k)i + (\sqrt{2(k - k^2})j\right] / \left[\sqrt{2(k - k^2)})i + (k - 1)j\right]\right\} \quad (3.8)$$

$$X = \left[-y\left(\sqrt{2(k - k^2)}\right) - x + 2xk\right]\cos C + \left[x\left(\sqrt{2(k - k^2)}\right) + 2yk - y\right]\sin C + (d - z)\left(\sqrt{2(k - k^2)}\right) \quad (3.9)$$

$$Y = \left[x\left(\sqrt{2(k - k^2)}\right) + yk\right]\cos c + \left[y\left(\sqrt{2(k - k^2)}\right) - xk\right]\sin C - z + d - dk + zk \quad (3.10)$$

$$Z = \left[x\left(\sqrt{2(k - k^2)}\right) + yk - y\right]\cos C + \left[y\left(\sqrt{2(k - k^2)}\right) - xk + x\right]\sin C + d - dk + zk \quad (3.11)$$

3.2. Sharp-corner and curve feedrate detection

In the sharp-corner and curvature areas, the feedrate has to slow down to reduce the chord error. The controller uses the NC information to plan the feedrate and avoid the machine shark in the machining process. The shark will increase the loading of the motor.

In the sharp corner, shown in Fig. 3, the feedrate is calculated by the variation of the angle (θ), the position gate (K_p), feedforward gain (K_f) and the maximum chord error ($\varepsilon_{\max}$) in [14]. The feedrate solution is expressed by Eq. (3.12):

$$V_c = \frac{K_p}{1-K_f} \cdot \frac{\varepsilon_{\max}}{\cos\left(\frac{\theta}{2}\right)} \tag{3.12}$$

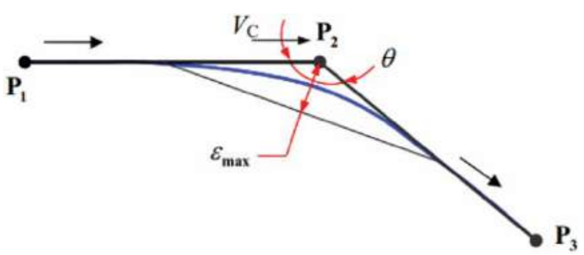

Figure 3. The error of the sharp corner [11].

Yau [15] proposed the shrinkage of the circle caused by the servo control and the feedrate planning; it could estimate the feedrate in the curve if the controller is a feedforward and ladder feedrate control system. The solution of the curve feedrate is expressed by Eq. (3.13):

$$V = \sqrt{a_{\max} \cdot \rho} \tag{3.13}$$

where ρ is the curvature of the tool path; $a_{\max}$ is the maximum acceleration of the controller. Using the inner parameters, $a_{\max}$ is expressed by Eq. (3.14):

$$a_{\max} = \Delta R / \left(\frac{(1-{K_f}^2)}{2{K_p}^2} + \frac{T^2}{24} \right) \tag{3.14}$$

where T is the constant of the feedrate time.

3.3. Feedrate planning

In multi-axis machine tools, the processing of interpretation and interpolation is more complex. It generally requires more computing time and resources. To keep the system in real time, the controller uses the pre-processing in interpretation, feedrate planning and interpolation. The method is called "Look-Ahead".

The feedrate planning of the look-ahead system is shown in Fig. 4. The lengths of each block in the tool path are integrated by the feedrate and time, expressed by Eq. (3.15):

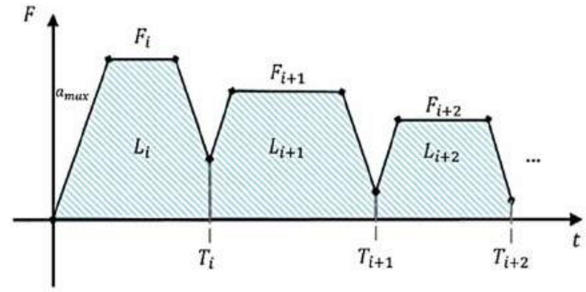

Figure 4. The feedrate planning.

$$L_i = \int_{T_{i-1}}^{T_i} F(t)dt \tag{3.15}$$

where $F(t)$ is the feedrate of the each block; L_i is the length of the each block.

4. Multi-axis simulation and collision detection

There are two interpolators in this paper: Motion interpolator and Adaptive interpolator. Motion interpolator provides the information to the multi-axis simulation, and adaptive interpolator provides the information to the collision detection.

4.1. Interpolator

The interpolator is proposed to subdivide the tool path to fit the shape of the surface in the model. When the accuracy on the surface is better, the subdivision of the tool is more detailed. The shape of workpiece is closer to the original design.

4.1.1. Motion interpolator

Based on the feedrate planning and one order Taylor expansion, the motion interpolator provides the information on the step to simulate the machine's movement. The function is expressed by Eq. (4.1):

$$u_{i+1} = u_i + \frac{V(u_i)}{|C'(u_i)|} T_s \tag{4.1}$$

where $V(u_i)$ is the velocity of the machine; $C'(u_i)$ is the curvature of the path; and T_s is the sample time.

4.1.2. Adaptive interpolator

After the feedrate planning, the execution time of each block is obtained. Based on the efficiency of the collision detection in the virtual environment, the count of the interpolator is computed by:

$$N_i = T_i / T_{CDT} \quad (4.2)$$

where T_i is the execution time of each block; T_{CDT} is the collision detection time. The length of each block is subdivided into N_i sections; each step has the same length:

$$\Delta L_i = L_i / N_i = \left(\int_{T_{i-1}}^{T_i} F(t)dt \right) / N_i \quad (4.3)$$

The system can adaptively adjust the length of the interpolator to detect the collision in the machining time. Using the path to test the adaptive interpolator, shown in Fig. 5, the setting of the system is: $\varepsilon_{max} = 0.01$ mm, $V_{max} = 5000$ mm/min, $A_{max} = 500$ mm/s^2, $K_p = 80$ and $K_f = 0.52$.

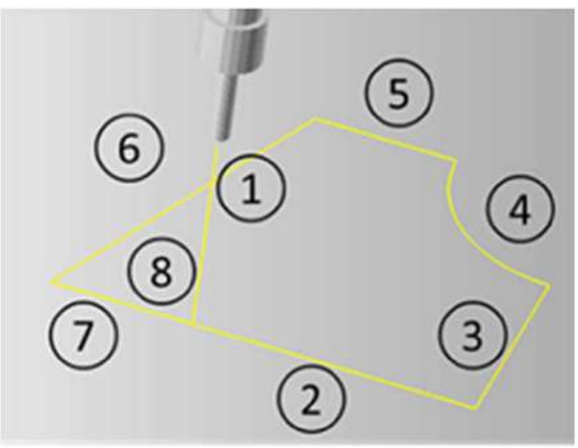

Figure 5. The tool path to test adaptive interpolator.

Table 1 shows the adaptive information. The step length of the adaptive interpolator is based on the execution time, the length of each block and the collision detection time. When changing the feedrate and the chord error (0.01 mm → 0.05 mm), the execution time and the step length change, as shown in Figs. 6 and 7. When the chord error increases from 0.01 mm to 0.05 mm, the execution time of each block decreases and the step of the interpolation is shorter than the original condition. On the other hand, when the feedrate of the NC data decreases, the execution time increases and the step of interpolation is longer than the original condition.

4.2. Collision detection

In recent years, collision detection has been widely used in industry, usually in the virtual reality in machine tools and robot arm. The collision detection uses the comparison of the STL model. When the complexity of the STL model increases, the computing time will also increase.

4.2.1. Oriented bounding box

To increase the efficiency of the collision detection, the bounding box and data structure of the model needs to be built. Gottschalk [3] proposed the Oriented Bounding Box (OBB) to enhance the efficiency of the collision detection. The axis of the OBB can along any vector. As shown in Fig. 8, the level of the OBB tree increases, the bounding box closes the shape of the object. The bounding box can achieve the closest shape of the model. This paper builds the bounding box of the parts of machine tools and the separating axis overlap test is employed to detect machine collisions.

4.2.2. The collision pair analyses

Tang [13] proposed the method of the machine collision pair. Based on the type of machine and the limit of each axis, the method analyzes the collision pair of the machine and builds the collision table to exclude the redundant collision pair, thereby increasing the efficiency of the collision detection.

Before analyzing the collision pair of the DMU 50 eVolution five-axis milling machine, the self-collision pair and repeat collision pair have to be excluded. The X, Y and Z axes are the serial configuration in this machine, so the collision pair of X-Y and Y-Z is excluded. The B-C remains the same. Based on the limit of each axis, the collision pair of X-Z, X-B, X-C, Y-B and Z-B are excluded. The only collision pair, the Z-C pair, has to detect the machine collision. In Table 2, ※ is the self-collision pair;

Table 1. The adaptive information of the testing tool path.

Block	NC Code	Initial Velocity (mm/min)	Maximum Velocity (mm/min)	Final Velocity (mm/min)	Path Length (mm)	Time (s)	Count of Interpolate	Step of Interpolate (mm)
1	G00	0.000	5000.000	141.421	60.000	0.882	882	0.068
2	G01	141.421	4000.000	141.421	80.000	1.324	1324	0.060
3	G01	141.421	4000.000	141.421	60.000	1.024	1024	0.059
4	G03	141.421	2989.122	141.421	62.832	1.351	1351	0.047
5	G01	141.421	3500.000	178.371	40.000	0.792	0.792	0.051
6	G01	178.371	3000.000	120.763	107.403	2.244	2244	0.048
7	G01	120.763	2700.000	141.421	40.000	0.970	970	0.041
8	G00	141.421	5000.000	0	60.000	0.882	882	0.068

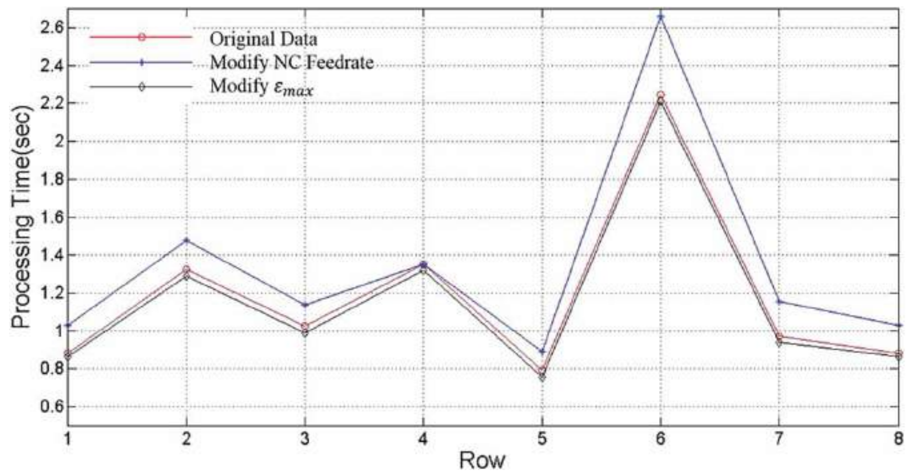

Figure 6. The execution time.

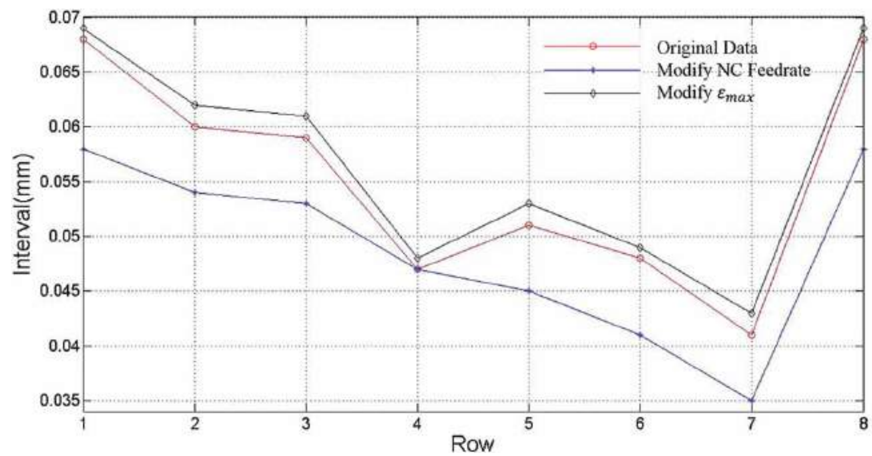

Figure 7. The step length of each block.

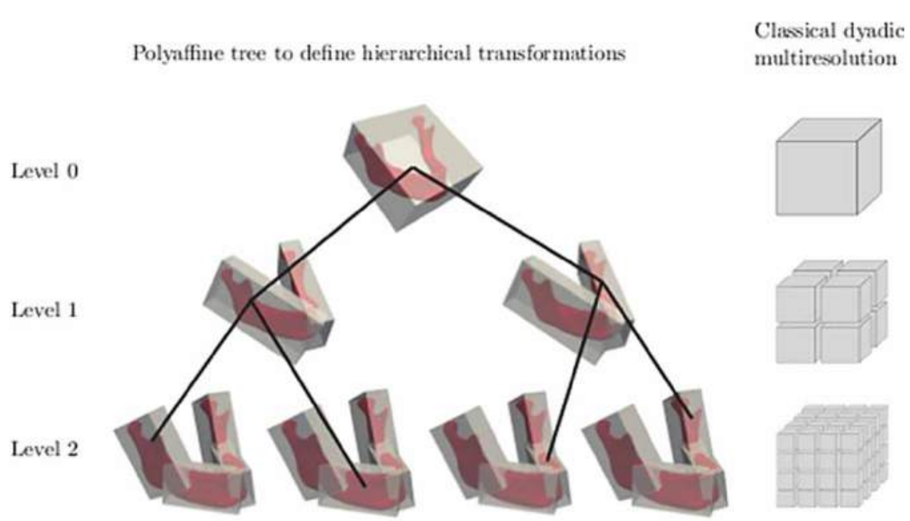

Figure 8. The level of the OBB tree [8].

Table 2. The collision table of DMU 50 eVolution.

Axis	X	Y	Z	B	C
X	※				
Y	•	※			
Z	•	•	※		■
B	•	•	•	※	
C	•	•	•	•	※

• is the repeat collision pair; ■ is the pair needing to detect the collision.

5. Results

The purpose of this paper was to establish a reliable machining simulation system with adaptive collision detection. The proposed system can be classified into two categories: Multi-axis machining simulation and Adaptive collision detection module. The hardware specification in this paper is listed in Table 3.

5.1. Multi-axis machining simulation

For the feedrate planning, the related parameters are defined as follows: $\varepsilon_{max} = 0.01$ mm, $V_{max} = 5000$ mm/ min and $A_{max} = 500$ mm/s^2. K_p and K_f are set 80 and 0.52, respectively, for the use of the simulation. Fig. 8 shows the relation between the feedrate and time. The feedrate planning shown in Fig. 8 can be obtained by the processing tool path with look-ahead function.

The real tool path can be interpolated by the look-ahead function and applied for simulation. Fig. 9 is the final machining path calculated by the abovementioned method. The color of the tool path presents the density of interpolated points.

The machine used in this paper is composed of 8 working parts. The total triangle number is 40875. The efficiency of collision detection concerning the optimization issue is listed in Table 4. With the same size of the model, the number of the detection does not affect the efficiency of the collision detection. Each collision detection time of DMU 50 eVolution is 3 ms. The value can

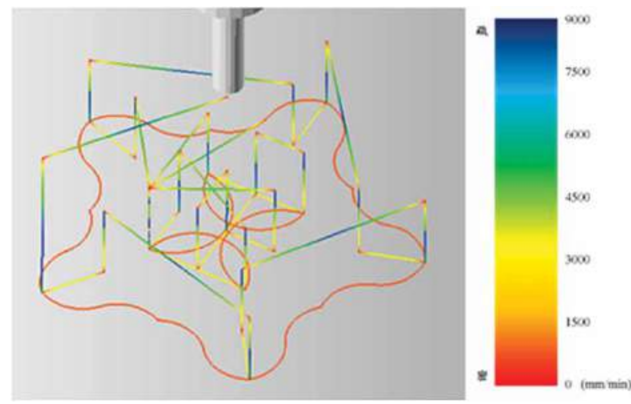

Figure 10. The color map of the interpolator.

Table 3. Hardware spec.

Hardware	Spec
OS	Windows7
CPU	AMD Phenom(tm)IIX4 965 Processor 3.40GHz
GPU	NVIDIA GeForce GTX460
Memory	4.00GB

Table 4. The collision detection analysis of DMU 50 eVolution.

The number of detection	Total detection time (*Sec*)	Average (*ms*)
329	0.95488	2.90
425	1.26761	2.98
781	2.32199	2.97
1261	3.71135	2.94
5847	17.30790	2.96

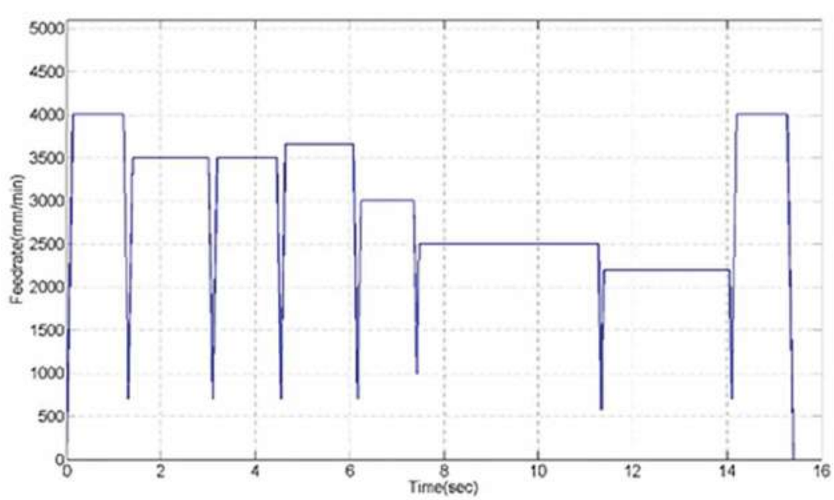

Figure 9. Feedrate planning of simulated tool path.

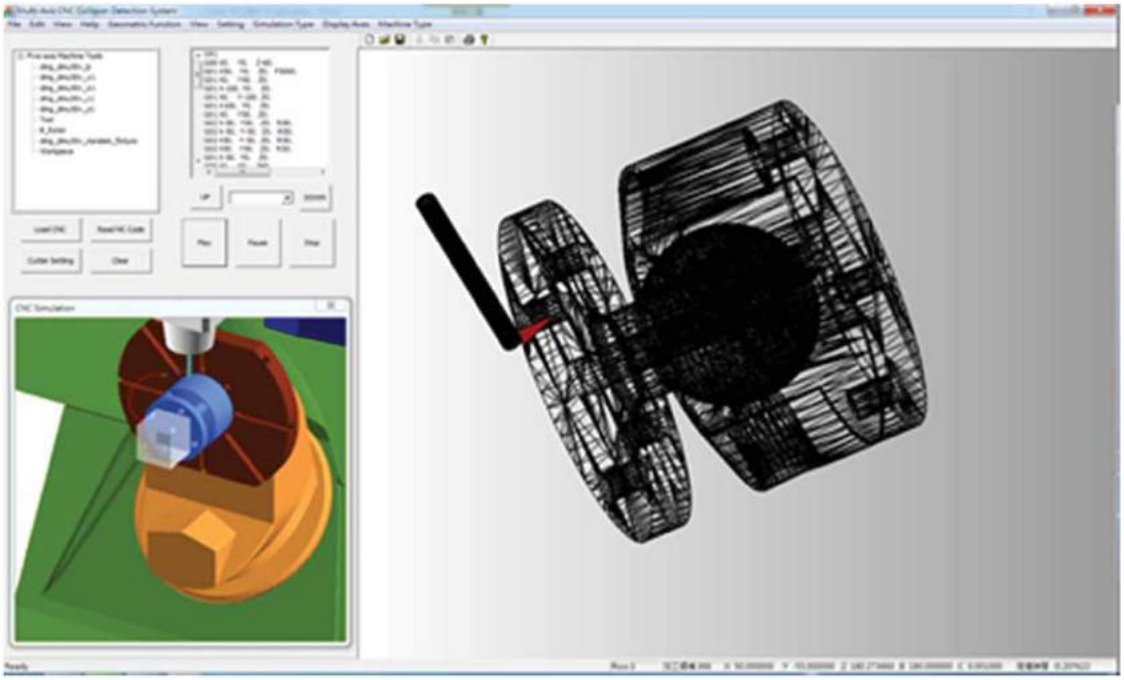

Figure 11. Collision of the machine.

be imported into the interpolator to obtain the adaptive collision detection number for the current tool path. This allows the simulation to execute the most precise collision detection during a limited time interval.

5.2. Multi-axis machining simulation with adaptive collision detection

Multi-axis machining simulation can be achieved by integrating feedrate planning and adaptive tool path interpolation. In this section, a multi-axis machining simulation system is developed to verify the reliability of the adaptive theory proposed in this paper. Fig. 11 shows the simulation environment used in this research. The right view shows the machining processing with collision detection. During simulation, the calculation of interference between working parts can be executed; it can provide an alert message if a collision is about to happen. The left view is the tool path calculated by the motion control interpolator module. Before simulation, the target model first needs to be selected to estimate the efficiency of its environmental collision detection and import it into the virtual simulation system. Fig. 10 is the result of the machining simulation of DMU 50 eVolution. During simulation, the collision detection between working parts can be executed in advance. If a collision occurs, the motion of the machine can be stopped and the exact location of the collision can be marked, as shown in Fig. 10.

6. Conclusion

We developed an NC machining simulation system with adaptive look-ahead interpolator and on-line collision detection function. The interferences between working parts can be detected by looking ahead at a safety distance during machining. In order to provide reliable safety, the machining can be stopped before the collision takes place to avoid machine or parts being damaged. The simulation system proposed in this paper is described as follows:

(1) Look-ahead function integration: The adaptive feedrate can be calculated according to tool path information and feedrate planning. The trapezoidal or S-curve acceleration and deceleration principle can be adopted to enhance the machining simulation's accuracy.
(2) Motion control interpolator: The natural machining simulation can be achieved by integrating look-ahead velocity planning with machine dynamic modeling.
(3) Adaptive collision detection: The collision detection number or steps can be calculated automatically by analyzing the processing path interval and detection efficiency.

The purpose is to perform the best possible precise collision detection during a limited time. Furthermore, the efficiency of collision detection is closely dependent on the triangle number of the machine and part models in the virtual environment. Therefore, parallel computing using GPU is adopted to enhance the collision detection efficiency. Currently, material removal simulation has not been integrated into this system. Further integration will make the system more applicable in real application and improve system versatility. In the future, the adaptive collision detection simulation system proposed in this paper can be integrated with real machines to validate the feasibility and improve machining safety.

ORCID

Hong-Tzong Yau http://orcid.org/0000-0001-8823-8520
Chia-Hao Chang http://orcid.org/0000-0002-1434-1063
Tsan-Jui Yang http://orcid.org/0000-0002-9326-929X
Jiun-Ting Lin http://orcid.org/0000-0002-0872-8923

References

[1] Ahmad, R.; Tichadou, S.; Hascoet, J. Y.: New computer vision based Snakes and Ladders algorithm for the safe trajectory of two axis CNC machines, *Computer-Aided Design*, 44, 2012, pp. 355–366. http://dx.doi.org/10.1016/j.cad.2011.12.008

[2] Cohen, J.D.; Lin, M. C.; Mamcocha, D.; Ponamgi, M.: I-COLLIDE: An interactive and exact collision detection system for large-scale environments, in Proceedings of the symposium on Interactive 3D graphics, 1995, 189-ff

[3] Gottschalk, S.; Lin, M. C.; Manocha, D.: OBBTree: a hierarchical structure for rapid interference detection, in Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, 1996, 171–180.

[4] Han, G. C.; Kim, D. I.; Kim, H. G.; Nam, K.; Choi, B. K.; Kim, S. K.: A high speed machining algorithm for CNC machine tools, in Industrial Electronics Society. IECON'99 Proceedings. The 25th Annual Conference of the IEEE, 1999, 1493–1497.

[5] Ilushin, G.; Elber, G.; Halperin, D.; Wein, R.; Kim, M.S.: Precise global collision detection in multi-axis NC-machining, *Computer-Aided Design*, 37, 2005, 909–920. http://dx.doi.org/10.1016/j.cad.2004.09.018

[6] Kim, D. I.: Study on interpolation algorithms of CNC machine tools, in Industry Applications Conference. Thirtieth IAS Annual Meeting, IAS'95, Conference Record of the IEEE, 1995, 1930–1937.

[7] Koren, Y.: *Computer control of manufacturing systems*, McGraw Hill, New York, NY, 1983.

[8] Koren, Y.: Control of machine tools, *Journal of Manufacturing Science and Engineering*, 119(4B), 1997, 749–755. http://dx.doi.org/10.1115/1.2836820

[9] Koren, Y.; Lo, C. C.; Shpitalni, M.: CNC interpolators: algorithms and analysis, *ASME Manufact Sci Engng*, 64, 1993, 83–92.

[10] Lo, C. C.; Hsiao, C. Y.: CNC machine tool interpolator with path compensation for repeated contour machining, *Computer-Aided Design*, 30(1), 1998, 55–62. http://dx.doi.org/10.1016/S0010-4485(97)00053-5

[11] Seiler, C; Pennec, X.; Reyes, M: Geometry-Aware Multiscale Image Registration via OBBTree-Based Polyaffine Log-Demons, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2011 Lecture Notes in Computer Science*, 6892, 2011, 631–638.

[12] Sørby, K.: Inverse kinematics of five-axis machines near singular configurations, *International Journal of Machine Tools and Manufacture*, 47(2), 2007, 299–306. http://dx.doi.org/10.1016/j.ijmachtools.2006.03.011

[13] Tang T.; Bohez, E. L.; Koomsap, P.: The sweep plane algorithm for global collision detection with workpiece geometry update for five-axis NC machining, *Computer-Aided Design*, 39(11), 2007, 1012–1024. http://dx.doi.org/10.1016/j.cad.2007.06.004

[14] Wang, J. B.; Yau, H. T.: Universal real-time NURBS interpolator on a PC-based controller, *The International Journal of Advanced Manufacturing Technology*, 71(1–4), 2014, 497–507. http://dx.doi.org/10.1007/s00170-013-5460-5

[15] Yau, H.T.; Kuo, M.J.: NURBS Machining and Feed Rate Adjustment for High-Speed Cutting of Complex Sculptured Surface, *International Journal of Production Research*, 39(1), 2001, 21–41. http://dx.doi.org/10.1080/00207540010002360

[16] Yeh, S. S.; Hsu P. L.: Adaptive-feedrate interpolation for parametric curves with a confined chord error, *Computer-Aided Design*, 34(3), 2002, 229–237. http://dx.doi.org/10.1016/S0010-4485(01)00082-3

[17] Yong, T.; Narayanaswami R.: A parametric interpolator with confined chord errors, acceleration and deceleration for NC machining, *Computer-Aided Design*, 35(13), 2003, 1249–1259. http://dx.doi.org/10.1016/S0010-4485(03)00043-5