The International Journal of Advanced Manufacturing Technology (2023) 124:2701–2717
https://doi.org/10.1007/s00170-022-10568-7

ORIGINAL ARTICLE

The use of the rapid unequal interval interpolation method (RUIIM) for a non-RTCP NC program and five-axis machining

Wei-Chih Chen[1,2] · Tzu-Kuan Lin[3] · Hsing-Ming Teng[4] · Chung-ChenTsao[2]

Received: 14 September 2022 / Accepted: 18 November 2022 / Published online: 17 December 2022

Abstract

In industries such as electronics 3C, medical, aviation, and energy, multi-axis machining machines are an important element of modern production systems, especially for post-processing, which necessitates the safe operation of control equipment. Some controller and software for five-axis simultaneous machining only interpolate equidistantly between cutter location (CL) points and do not support post-processing using a rotation tool center point (RTCP) so the machining path of the tool has manufacturing errors (overcutting or undercutting) and the workpiece fails to be machined to the correct size. Depending on the maximum allowable error (δ_{Max}), this study uses rapid unequal interval sub-point interpolation to perform sub-point interpolation on the original path of the five-axis machining machine, and the sub-point values are output in the computer numerical control (CNC) machining program in C# programming language. The tool tip is then moved to the desired machining position. Finally, an example involving helical milling of a curved surface is used to verify the feasibility of the rapid unequal interval interpolation method (RUIIM). The experimental results show that rapid unequal interval sub-point interpolation gives a more accurate original path that is generated in the CAM and a faster processing speed for the entire controller so the operation speed is increased.

Keywords Five-axis machining · Rapid unequal interval interpolation method · Post-processor

1 Introduction

Most product manufacturing and development uses computer-aided design (CAD) and computer-aided manufacturing (CAM) for design and manufacture. Machining is completed by a computer numerically controlled (CNC) machine tool. Five-axis machining is used for the machining of complex components for the high-tech electronics 3C, medical, aviation, and energy industries. However, controlling this equipment to correctly and safely process complex products involves post-processing using CAM software to generate the desired cutter location (CL) points for the tool path and to convert the final numerical control (NC) processing program. This task is often controlled by foreign software companies or is purchased from third-party companies, so there are restrictions on the development of the secondary CAM process.

Shigehiko and Ichiro [1] noted that most five-axis machines are based on traditional three-axis machining machines and are adapted to machine multiple inclined surfaces and complex shapes, so a structure with two rotating axes is added to allow processing using just one clamping operation. Lin et al. [2] proposed a spherical two-circle (STC) method for a five-axis machining machine with any structure. Lee and She [3] used homogeneous coordinate transformation and inverse kinematics to convert orthogonal multi-axis milling machine CL data to NC data. Yoshimi and Takahiro [4] proposed a machining coordinate system (MCS) and a workpiece coordinate system (WCS) for a five-axis machining machine, to allow workpieces with complex

✉ Chung-ChenTsao
aetcc@mail.lhu.edu.tw

[1] Kuantai Technology Company, New Taipei 241062, Taiwan, Republic of China

[2] Department of Mechanical Engineering, Lunghwa University of Science and Technology, Taoyuan 333326, Taiwan, Republic of China

[3] Department of Mechanical Engineering, National Taiwan University of Science and Technology, Taipei 106335, Taiwan, Republic of China

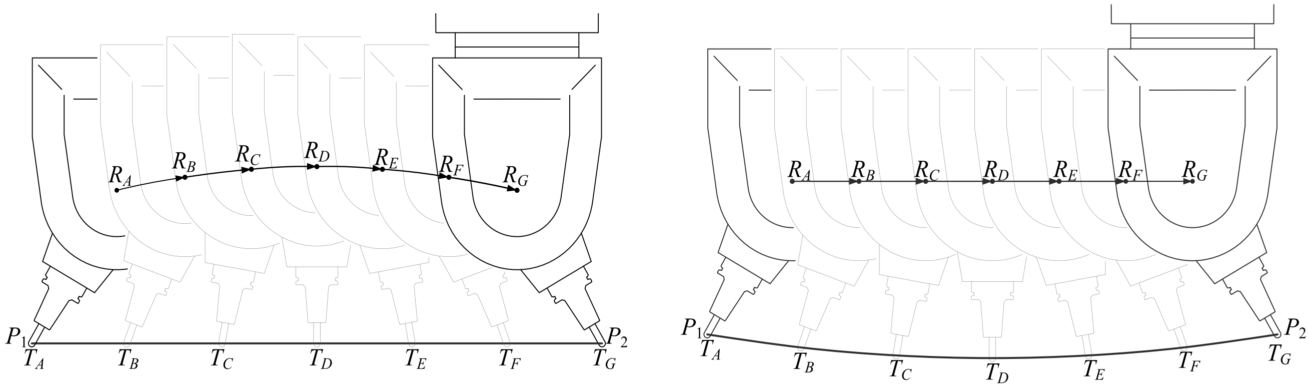
[4] Department of Mechanical Engineering, Asia Eastern University of Science and Technology, New Taipei 220303, Taiwan, Republic of China

shapes to be processed using a five-axis machine. Jung et al. [5] noted that when processing free-form surfaces on a three-axis machine, CL data must be obtained using CAD/CAM software. This data comprises the coordinates that are generated by the definition of the WCS, to allow post-processing. It is converted into the NC machining program for the MCS. However, there is a rotating shaft so errors can occur in the forward and reverse process. This study also showed that the CL data must include the position of the tool relative to the workpiece and the direction of the tool axis and the WCS relative to the MCS is converted into the actual cutting motion for the tool. Two methods were proposed: (1) an interference free algorithm in phase reverse and (2) an algorithm for fewer phase reverse processes. This technique and algorithm is proprietary so it cannot be used by third parties to solve post-processing problems. Vavruska [6] interpolated the tool path using a post-processor to increase the accuracy of multi-axis machining. If the rotation tool center point (RTCP) function is not used, the toolpath tolerances may be violated, so an algorithm in the post-processor dynamically generates new toolpath point coordinates to obey the tolerance and ensure machining accuracy. Some controllers do not support the RTCP function or because of errors in the machine so the post-processing program is modified by the equipment manufacturer. In this case, a technician cannot use the RTCP function. Lai et al. [7] did not interpolate many points using a linearized algorithm. This study recalculated new paths for the NC program to prevent discretization near singularities. When a five-axis machining machine performs linear cutting on a curved surface, the controller does not automatically interpolate data points between two NC blocks in a linearly distributed manner so the path is not a straight line and the tool deviates from the original path. Lin and Lin [8] propose a method to increase the number of CL interpolation points to prevent deviation from the machining path. Yu and Ding [9] determined the relationship between the MCS and the WCS. The tool deviation path is corrected by linearly interpolating points between the two CLs and then outputting to the NC program. Five-axis equal chord-height linear interpolation and the circular interpolation method are similar. If the effect of the five-axis machining on path interpolation is not considered, the final d product may not be cut correctly.

When the RTCP function is not used (non-RTCP), the tool path is interpolated in equal segments or equal fixed lengths to correct deviations so the workpiece can be cut incorrectly. For two-point straight $P_1 \rightarrow P_2$ cutting, the difference between RTCP and non-RTCP is shown in Fig. 1. The tool path trajectory for RTCP cutting is a straight line $P_1 \rightarrow P_2$, because the coordinates are converted by the five-axis machining machine controller on the tool axis. However, non-RTCP has no reference from which to generate toolpaths so only the output results can be used to simulate and adjust the interpolated values. This interpolation method results in equally spaced interpolation of the original deviation path and a greater number of interpolation points at positions that do not require interpolation, so the NC program changes. If this change is too great, the time that is required for calculation and transmission to the controller is increased and more memory is required by the controller, so processing efficiency is compromised. This study uses controllers without RTCP or CAM software that do not output RTCP-specific instructions and uses unequal interval interpolation when path deviation occurs. It also allows optimal segmental interpolation for CL points and outputs these to a correct NC program. The system is then applied to the machining tool path for a five-axis machine tool.

2 Research method

This study concerns non-RTCP, so the operation mode for non-RTCP is defined in terms of a coordinate transformation matrix. For different types of processing machines, the



(a) RTCP

(b) Non-RTCP

Fig. 1 Displacement action of RTCP and non-RTCP

tool axis vectors (i,j,k) are calculated. The machine tool is rotated using the AC, AB, or BC, and the X, Y, and Z coordinates for the NC program are calculated using the rotational angle and the x, y, and z coordinates for the CL point. This study determines the NC interpolation point, so the error between the NC linear interpolation and the CL data points is calculated and then obtained using the calculation of NC to CL, as shown in Fig. 2. The NC program coordinates for the RTCP are derived using the WCS, and the deviation parameters for the rotational axis are set by the controller parameters to generate a NC program to move the machine. However, the NC program for non-RTCP is calculated using the MCS, so if the deviation parameters are not calculated and the coordinates converted, the tool path trajectory produces a workpiece that is incorrect. A table type (TT) five-axis machining machine tool does not swing so it is more rigid. This type of machine is used for heavy cutting and high feed processes. Therefore, this study uses a five-axis machining machine that rotates around the A-axis and C-axis (TATC) for verification, as shown in Fig. 3.

The five-axis NC coordinates include NC point coordinates on the X, Y, and Z axes and rotation angles A, B, or C. The rotation angle is derived from the tool axis vector, so the rotation angle through the tool axis vector is calculated. To calculate the cutting path for a five-axis machine tool, this study uses the STC method, which generates the cutting path for various types of five-axis machine tools. The STC method uses four vectors and two circles, as shown in Fig. 4. For a general model of a TT five-axis machine tool, the motion of the simulated tool axis vector (workpiece) must terminate at the spindle vector and the tool axis vector is aligned with the spindle vector. Therefore, the dynamic point for a TT five-axis machining machine model starts from point P_t and moves around the secondary rotational axis vector (V) and then through the conversion point P_1 (or P_2). The point P_s moves around the main rotation axis vector (U). The angle between the conversion point P_1 (or P_2) and the main axis vector and the tool axis vector is calculated using inverse kinematics to determine the rotational angles for the two rotational axes. These two rotation angles are used to calculate the NC point coordinates for TT five-axis machine tools using forward kinematics and by transforming the coordinates for the post-processor using CAM software.

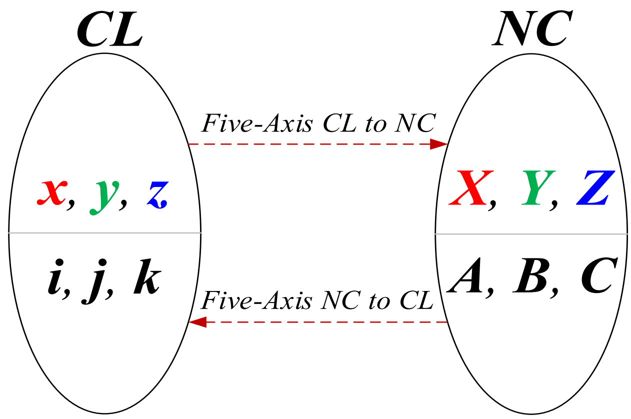


Fig. 2 Relationship between CL and NC

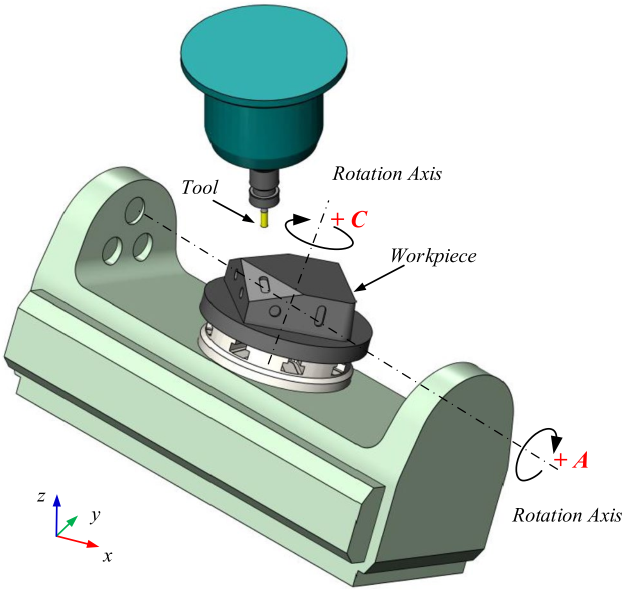


Fig. 3 Table type

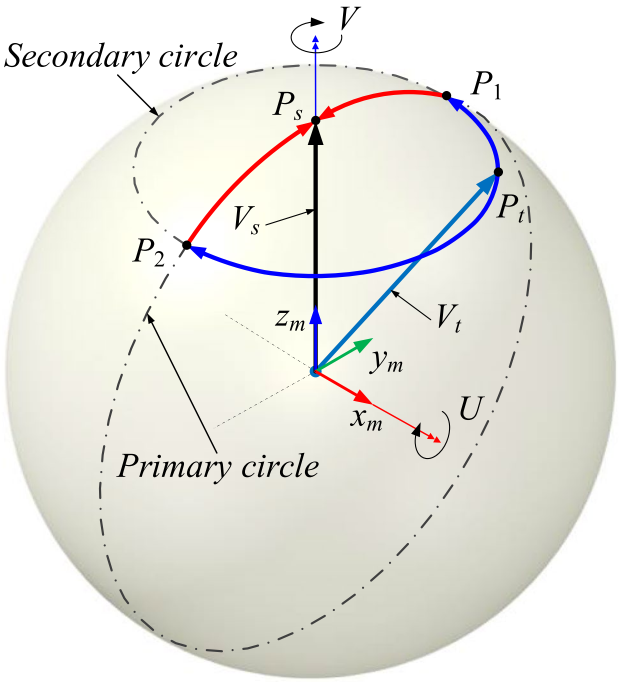


Fig. 4 The general model of the TT five-axis machine tools

The conversion point plays a very important role in generating the cutting path for TT five-axis machine tools.

Five-axis simultaneous machining uses the NC program that is generated by post-processing, and the controller uses this program to drive the five-axis machine to process complex parts and avoid interference. This study uses the five-axis tool path method for Mastercam software to calculate and generate the tool path and determine the coordinates of the tool cutting location (CL) and the tool axis vector. However, in addition to the CL point for the five-axis path, the tool axis vector is also considered. If the five-axis machining path is directly calculated using linear interpolation, the machining path generates a five-axis error. Using the STC method, the rotational angle and the NC point coordinates are first calculated and the five-axis error (δ) between two adjacent CLs is calculated using a midpoint approximate. If the value of δ is within the range of the maximum allowable error (δ_{Max}), the rotational angle and the NC point coordinates are output and written into the NC program. If δ exceeds δ_{Max}, the number of sub-points is calculated using the RUIIM and the calculated dividing points are used to divide the points between two adjacent CLs that exceed δ_{Max}. The value for δ between the two adjacent CLs after division is then calculated. If δ is within the range of δ_{Max}, the rotational angle and the NC point coordinates are output and written into the NC program. If δ exceeds δ_{Max}, the RUIIM is used to calculate the sub-points until the value of δ is within the range of δ_{Max}. Finally, the NC program for the range of δ_{Max} is verified using simulation software such as CIMCO and VERICUT and cutting is initiated using a five-axis machining machine to verify the accuracy of the method for this study. The flow chart for the method and the result for the non-RTCP RUIIM that is proposed is shown in Fig. 5.

3 Five-axis error

CNC tool machining uses both hardware and software. The hardware error affects the assembly of mechanical parts. It is determined by adjusting and calibration. There is also an error in the deviation of the rotating shaft due to poor assembly or wear over a period of time. The software error affects the calculation error when the CAM software and the controller execute the NC program. For three-axis machining, the controller executes a linear motion on *XYZ* plane using linear interpolation. For five-axis machining, there is an original linear motion in the *XYZ* plane and the motion of the two axes of rotation in A, B, and C. If the linear interpolation method is used, the workpiece is cut using an incorrect path. For non-RTCP, the tool cutting path that is generated by the five-axis controller is from NC_1 (X50. Y80. Z100. A30. C90.) to NC_2 (X100. Y120. Z200. A60. C150.). The tool path results are simulated using simulation software, as shown in Fig. 6. The tool path from NC_1 to NC_2 is a straight line, but the tool path that is cut by the five-axis machine is a curve. When the controller executes

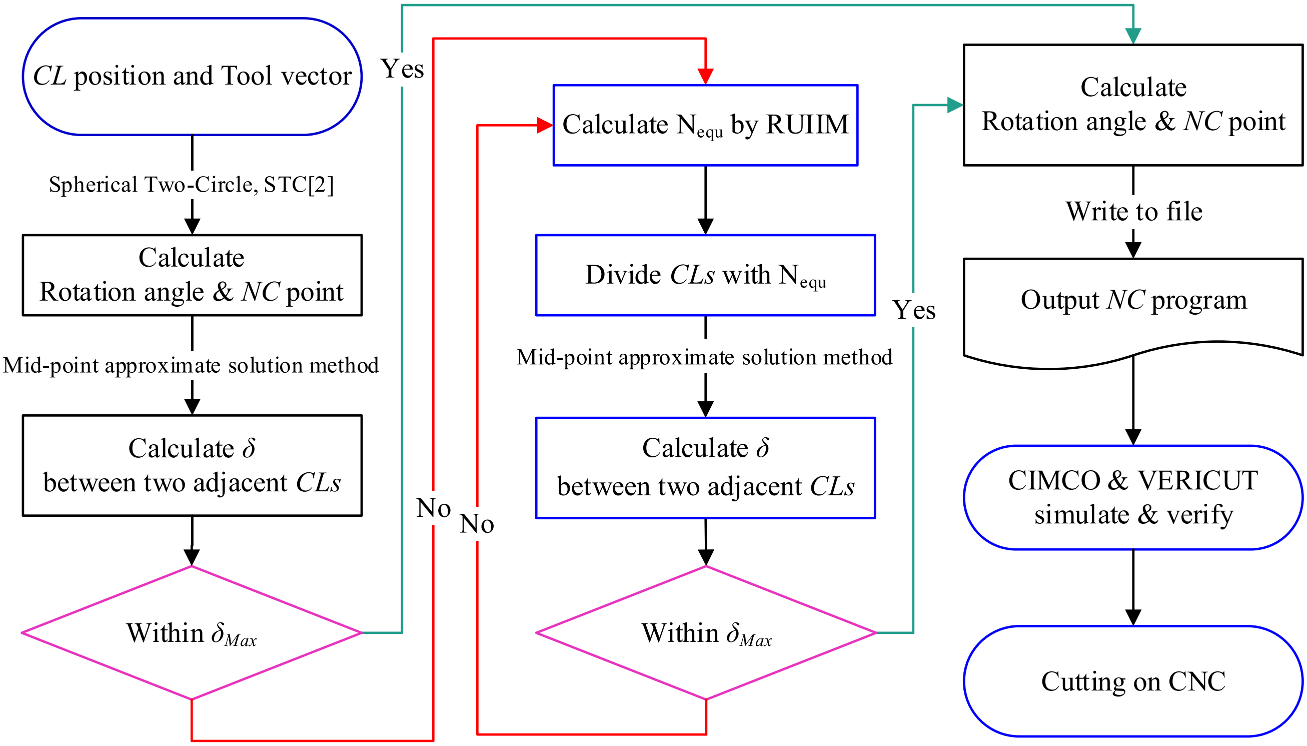
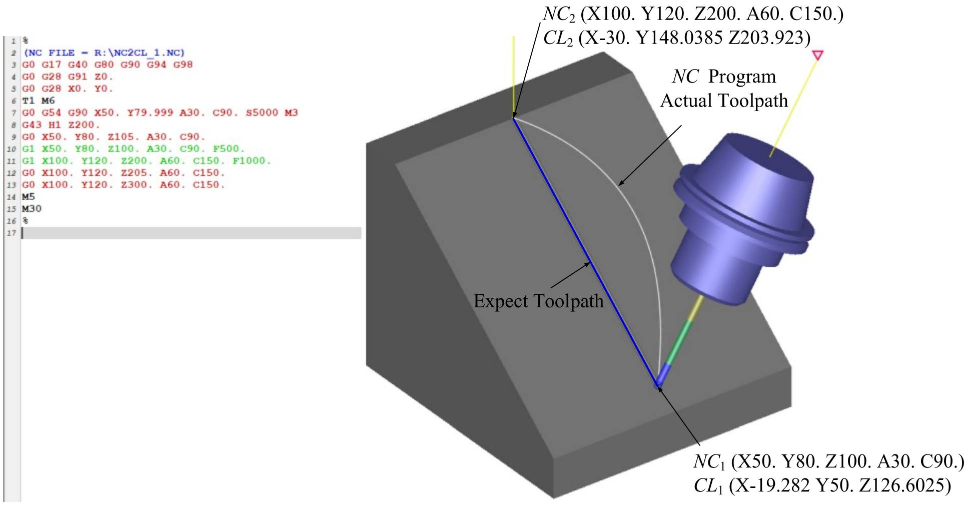

CL position and Tool vector
Spherical Two-Circle, STC[2]
Calculate Rotation angle & NC point
Mid-point approximate solution method
Calculate δ between two adjacent CLs
Within δ_Max
Yes
No
Calculate N_equ by RUIIM
Divide CLs with N_equ
Mid-point approximate solution method
Calculate δ between two adjacent CLs
Within δ_Max
No
Yes
Calculate Rotation angle & NC point
Write to file
Output NC program
CIMCO & VERICUT simulate & verify
Cutting on CNC

Fig. 5 The research method and result flow chart of the RUIIM research for non-tool following five-axis machining

Fig. 6 Simulation screen of two NC points



```
%
(NC FILE - R:\NC2CL_1.NC)
G0 G17 G40 G80 G90 G94 G98
G0 G28 G91 Z0.
G0 G28 X0. Y0.
T1 M6
G0 G54 G90 X50. Y79.999 A30. C90. S5000 M3
G43 H1 Z200.
G0 X50. Y80. Z105. A30. C90.
G1 X50. Y80. Z100. A30. C90. F500.
G1 X100. Y120. Z200. A60. C150. F1000.
G0 X100. Y120. Z205. A60. C150.
G0 X100. Y120. Z300. A60. C150.
M5
M30
%
```

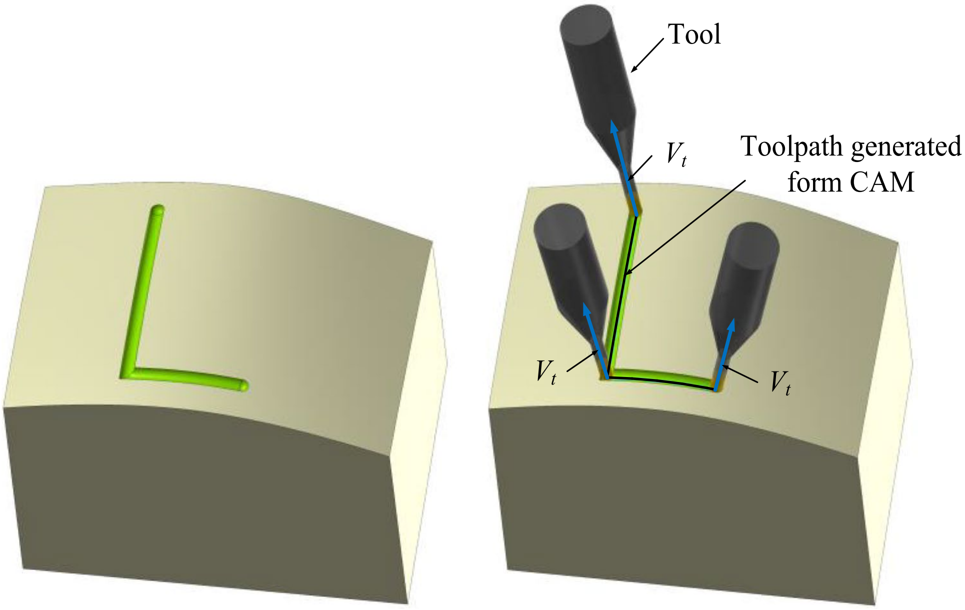
NC_2 (X100. Y120. Z200. A60. C150.)
CL_2 (X-30. Y148.0385 Z203.923)
NC Program
Actual Toolpath
Expect Toolpath
NC_1 (X50. Y80. Z100. A30. C90.)
CL_1 (X-19.282 Y50. Z126.6025)

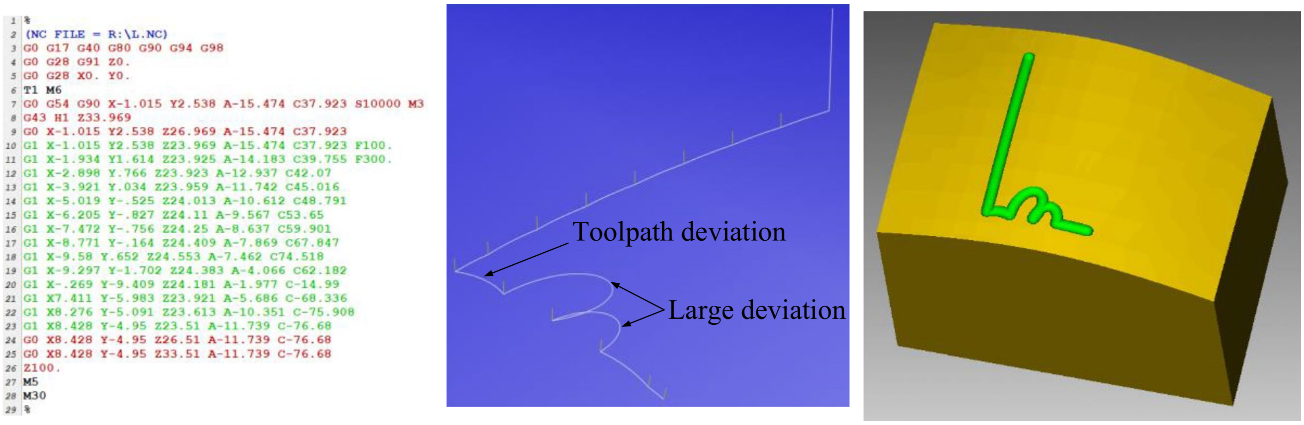
the five-axis program, the CL between the paths is generated by linear interpolation.

CAM software was used to produce an English letter L on a curved surface using a five-axis path method. The tool axis vector V_t for the tool path is normal to the surface, the cutting tolerance is 0.025 mm, and the step (the length of the points) is 2 mm. The resulting tool path is shown in Fig. 7. The NC program is generated using the post-processor in the CAM software, and the tool path is simulated using CIMCO simulation software. Some tool paths feature large deviations and many tool paths that deviate significantly, as shown in Fig. 8.

The NC program for the CAM software generates deviating tool path so a 0.1 mm step size (dividing point length) is used to generate a new tool path and the NC program is generated using the post-processor in CAM software. CIMCO simulation software is used to simulate the tool path, and the simulation results are shown in Fig. 9. The tool path deviation in Fig. 8 is resolved, but the CAM software performs more equal-spaced interpolation on the original deviation path. This adds CL points where no interpolation is required. A comparison of Figs. 8 and 9 shows that the NC program increases from 28 to 248 blocks and the single block rate for the increase is nearly 8 times that of the original NC program. The CAM software segments and interpolates all of the tool paths and to increase the number of CL points so there are too many interpolation points in the tool path for the NC program and the number of single blocks for the NC program increases. This is a common method for correcting path deviation, but the problem of generating path deviation is not resolved.

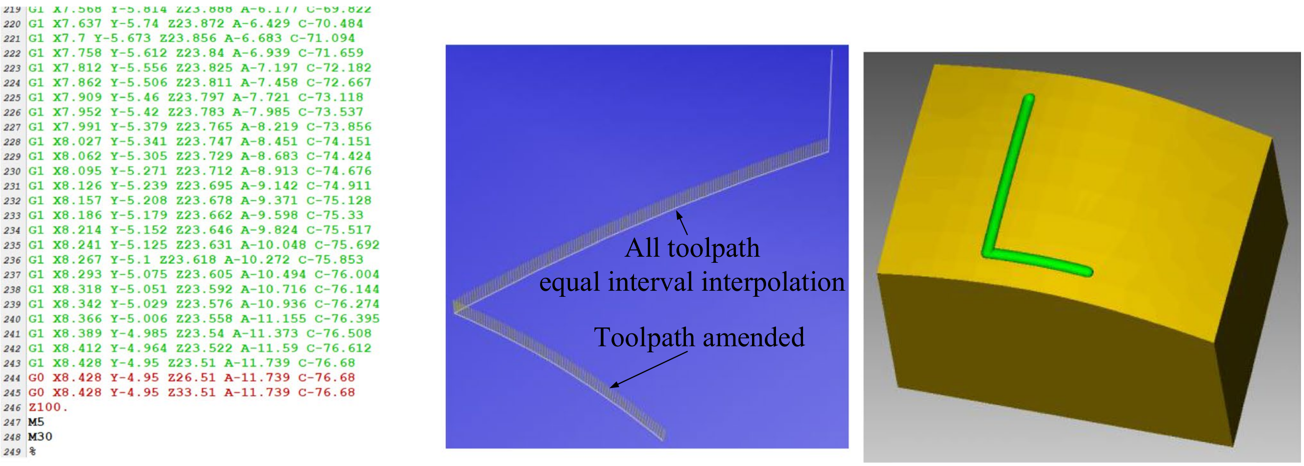
Fig. 7 Letter L for five-axis surface machining





```
%
(NC FILE = R:\L.NC)
G0 G17 G40 G80 G90 G94 G98
G0 G28 G91 Z0.
G0 G28 X0. Y0.
T1 M6
G0 G54 G90 X-1.015 Y2.538 A-15.474 C37.923 S10000 M3
G43 H1 Z33.969
G0 X-1.015 Y2.538 Z26.969 A-15.474 C37.923
G1 X-1.015 Y2.538 Z23.969 A-15.474 C37.923 F100.
G1 X-1.934 Y1.614 Z23.925 A-14.183 C39.755 F300.
G1 X-2.898 Y.766 Z23.923 A-12.937 C42.07
G1 X-3.921 Y.034 Z23.959 A-11.742 C45.016
G1 X-5.019 Y-.525 Z24.013 A-10.612 C48.791
G1 X-6.205 Y-.827 Z24.11 A-9.567 C53.65
G1 X-7.472 Y-.756 Z24.25 A-8.637 C59.901
G1 X-8.771 Y-.164 Z24.409 A-7.869 C67.847
G1 X-9.58 Y.652 Z24.553 A-7.462 C74.518
G1 X-9.297 Y-1.702 Z24.383 A-4.066 C62.182
G1 X-.269 Y-9.409 Z24.181 A-1.977 C-14.99
G1 X7.411 Y-5.983 Z23.921 A-5.686 C-68.336
G1 X8.276 Y-5.091 Z23.613 A-10.351 C-75.908
G1 X8.428 Y-4.95 Z23.51 A-11.739 C-76.68
G0 X8.428 Y-4.95 Z26.51 A-11.739 C-76.68
G0 X8.428 Y-4.95 Z33.51 A-11.739 C-76.68
Z100.
M5
M30
%
```

Fig. 8 NC program and simulation screen generated by CAM software



```
G1 X7.568 Y-5.814 Z23.888 A-6.177 C-69.822
G1 X7.637 Y-5.74 Z23.872 A-6.429 C-70.484
G1 X7.7 Y-5.673 Z23.856 A-6.683 C-71.094
G1 X7.758 Y-5.612 Z23.84 A-6.939 C-71.659
G1 X7.812 Y-5.556 Z23.825 A-7.197 C-72.182
G1 X7.862 Y-5.506 Z23.811 A-7.458 C-72.667
G1 X7.909 Y-5.46 Z23.797 A-7.721 C-73.118
G1 X7.952 Y-5.42 Z23.783 A-7.985 C-73.537
G1 X7.991 Y-5.379 Z23.765 A-8.219 C-73.856
G1 X8.027 Y-5.341 Z23.747 A-8.451 C-74.151
G1 X8.062 Y-5.305 Z23.729 A-8.683 C-74.424
G1 X8.095 Y-5.271 Z23.712 A-8.913 C-74.676
G1 X8.126 Y-5.239 Z23.695 A-9.142 C-74.911
G1 X8.157 Y-5.208 Z23.678 A-9.371 C-75.128
G1 X8.186 Y-5.179 Z23.662 A-9.598 C-75.33
G1 X8.214 Y-5.152 Z23.646 A-9.824 C-75.517
G1 X8.241 Y-5.125 Z23.631 A-10.048 C-75.692
G1 X8.267 Y-5.1 Z23.618 A-10.272 C-75.853
G1 X8.293 Y-5.075 Z23.605 A-10.494 C-76.004
G1 X8.318 Y-5.051 Z23.592 A-10.716 C-76.144
G1 X8.342 Y-5.029 Z23.576 A-10.936 C-76.274
G1 X8.366 Y-5.006 Z23.558 A-11.155 C-76.395
G1 X8.389 Y-4.985 Z23.54 A-11.373 C-76.508
G1 X8.412 Y-4.964 Z23.522 A-11.59 C-76.612
G1 X8.428 Y-4.95 Z23.51 A-11.739 C-76.68
G0 X8.428 Y-4.95 Z26.51 A-11.739 C-76.68
G0 X8.428 Y-4.95 Z33.51 A-11.739 C-76.68
Z100.
M5
M30
%
```

Fig. 9 NC and simulation screen after equidistant interpolation of CL points by CAM software

In the non-RTCP state, the NC program uses linear interpolation so there is path deviation phenomenon between the designed path and the expected tool path, which is δ. To resolve a five-axis error between two CL points, isometric interpolation is performed between the CL points of the tool path. If all of the CL points are interpolated using the same condition, then all paths are interpolated equidistantly. If there are insufficient equidistant interpolation points, deviation in the local position point is not corrected. If there are too many interpolation points, the position points that do not undergo equidistant interpolation are also interpolated, which affects the overall processing efficiency. To resolve the problem of redundant equidistant interpolation position points for this study, the calculation of δ must be understood. This study uses the approximate solution method to solve δ. The flow chart for error calculation is shown in Fig. 10. The coordinates of the two CL points CL_1 and CL_2 are converted to NC$_1$ and NC$_2$, and NC$_1$ and NC$_2$ are used to determine the intermediate value CL_{NCm}. The distance between the intermediate value CL_m is calculated for CL_1 and CL_2 and the path deviation for CL_{NCm} is δ.

4 Unequal interval CL interpolation

If too many interpolation points are generated by equidistant interpolation five-axis errors, this study uses five-axis unequal spacing CL interpolation to accelerate the operation. The value of δ is used to determine the equally spaced points. The first division point for CL is calculated using the set expected error. If the value of δ for the CL output NC for the interpolation points is greater than the expected error, a second interpolation point is used. If there is a third interpolation point, the five-axis toolpath errors reach the convergence level so

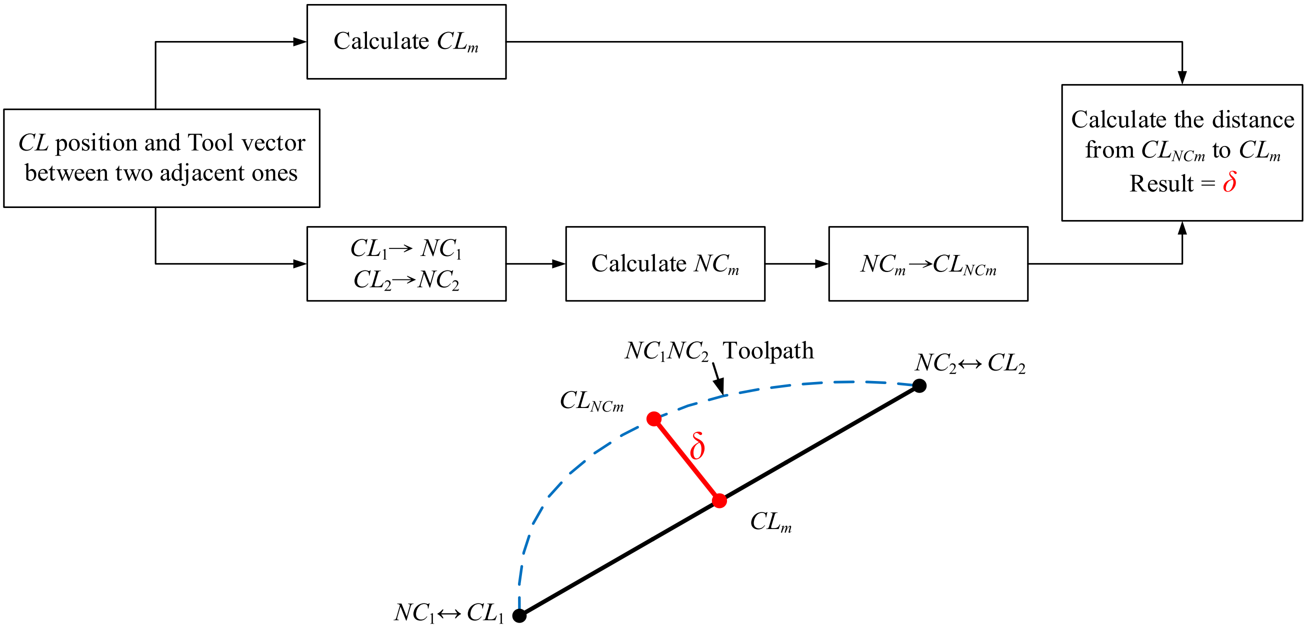

Calculate CL_m
CL position and Tool vector between two adjacent ones
Calculate the distance from CL_{NCm} to CL_m Result = δ
$CL_1 \rightarrow NC_1$
$CL_2 \rightarrow NC_2$
Calculate NC_m
$NC_m \rightarrow CL_{NCm}$
NC_1NC_2 Toolpath
$NC_2 \leftrightarrow CL_2$
CL_{NCm}
δ
CL_m
$NC_1 \leftrightarrow CL_1$

Fig. 10 Flow chart of δ calculation

the unequal interval interpolation points are obtained. The CL points that are output by CAM software are used to optimize the piecewise interpolation. The five-axis unequal-spaced CL interpolation procedure is described in detail later.

1. The interpolation points in the original equal interval between CLs are added, and if the interpolated δ is less than or equal to the set δ_{Max}, the next point is calculated using the value for δ_{Max} to calculate the required number of points. The points between CL points are then interpolated.
2. If the CL_u point is inserted linearly between CL_1 and CL_2, then using the calculation method of the flow chart in Fig. 10, the difference in the position of the CL_u point between CL_1 and CL_2 also affects the value of δ. If the distance between CL_1 and CL_u increases, the corrected five-axis error δ_{u1} also increases, as shown in Fig. 11a. If the distance between CL_1 and CL_u decreases, the corrected five-axis error δ_{u1} increases. The error δ_{u2} is smaller, as shown in Fig. 11b. The results in Fig. 11 show that the distance between CL_1 and CL_u is closely related to the corrected value for δ. The position of the next point CL_u that gives a value of δ that is within δ_{Max} is very important.

u is the position of the CL_u point on $CL_1 \rightarrow CL_2$, so the value of u is calculated before the coordinate value of the CL_u point is calculated. The CL_u point is the position for which δ is within δ_{Max}. Therefore, the distance between CL_1 and CL_2 is set as a unit length with a value of 1. Its relationship is shown in Eq. (1):

Fig. 11 δ of the next point CL_u

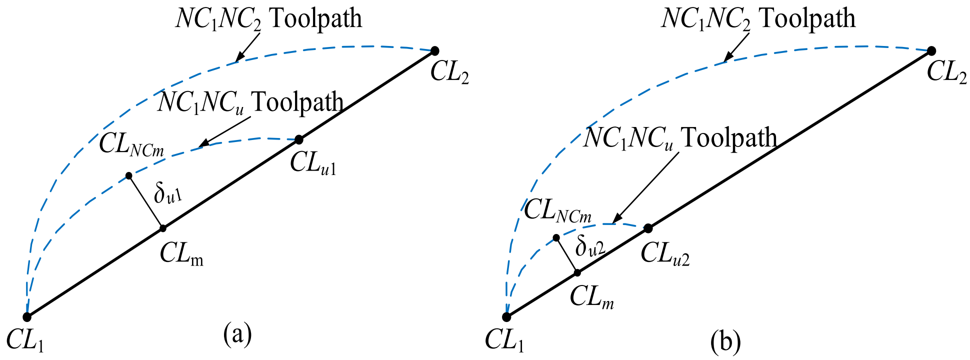


Table 1 Two CL points and their tool axis vectors

	CL_x	CL_y	CL_z	i	j	k
CL_1	-19.2820	50.0000	126.6025	0.5000	0.0000	0.8660
CL_2	-30.0000	-148.0385	203.9230	0.4330	0.7500	0.5000

Table 2 CL_1, CL_u, and tool axis vector

	CL_x	CL_y	CL_z	i	j	k
CL_1	-19.2820	50.0000	126.6025	0.5000	0.0000	0.8660
CL_u	-19.5461	52.4157	128.5077	0.5026	0.0186	0.0864

Table 3 Double-solution NC program coordinates of interpolated CL_u points

		X	Y	Z	A	C
$CL_u \rightarrow NC_u$	Pos.	53.1062	79.8263	102.2310	30.1928	92.1308
	Neg.	-53.1062	-79.8263	102.2310	-30.1928	-87.8692

$$
\begin{aligned}
&CL(u)\\
&CL(0) = CL_1\\
&CL(1) = CL_2
\end{aligned}
\tag{1}
$$

After linear interpolation of point CL_u on each axis (X, Y, Z), $(CL_{ux}, CL_{uy}, CL_{uz})$ is calculated using Eq. (2):

$$
CL(u) = \left(CL_{1x}(1-u) + CL_{2x}(u),\ CL_{1y}(1-u) + CL_{2y}(u),\ CL_{1z}(1-u) + CL_{2z}(u)\right)
\tag{2}
$$

The value that is calculated using Eq. (2) is the scalar value for the five-axis toolpath, so the same linear interpolation is performed on the original tool axis vector (i, j, k) for point CL_u. The axial components of point CL_u after linear interpolation are $(CL_{uit}, CL_{ujt}, CL_{ukt})$, as shown in Eq. (3):

$$
\begin{aligned}
CL_{uit} &= CL_{1i}(1-u) + CL_{2i}(u)\\
CL_{ujt} &= CL_{1j}(1-u) + CL_{2j}(u)\\
CL_{ukt} &= CL_{1k}(1-u) + CL_{2k}(u)
\end{aligned}
\tag{3}
$$

Each axial component is unitized into $(CL_{ui}, CL_{uj}, CL_{uk})$ after linear interpolation, which is the tool axis vector (i, j, k) of CL_u, as shown in Eq. (4):

$$
\begin{aligned}
CL_{ui} &= CL_{uit} / \sqrt{{CL_{uit}}^2 + {CL_{ujt}}^2 + {CL_{ukt}}^2}\\
CL_{uj} &= CL_{ujt} / \sqrt{{CL_{uit}}^2 + {CL_{ujt}}^2 + {CL_{ukt}}^2}\\
CL_{uk} &= CL_{ukt} / \sqrt{{CL_{uit}}^2 + {CL_{ujt}}^2 + {CL_{ukt}}^2}
\end{aligned}
\tag{4}
$$

For the two CL points in Table 1, δ_{Max} is 0.025 mm. Starting at CL_1, according to Fig. 11, the u value on $CL_1 \rightarrow CL_2$ is calculated to determine the position of point CL_u. If the δ between $CL_1 \rightarrow CL_u$ is within the range of δ_{Max} of 0.025 mm and u_{Max}, which is the maximum value, this is the best interpolation point position. The inverse matrix solution method is used, starting from 0, substituting the u value with micro-steps and repeatedly testing and calculating. If the δ between $CL_1 \rightarrow CL_u$ is also within the range of δ_{Max}, the u_{Max} value is 0.0246 mm. Using Eqs. (2), (3), and (4), the coordinates $(CL_{ux}, CL_{uy}, CL_{uz})$ for the interpolated CL_u and the tool axis vectors (i, j, k) are calculated, as shown in Table 2. The double-solution NC program coordinates of the interpolated point CL_u are calculated as shown in Table 3.

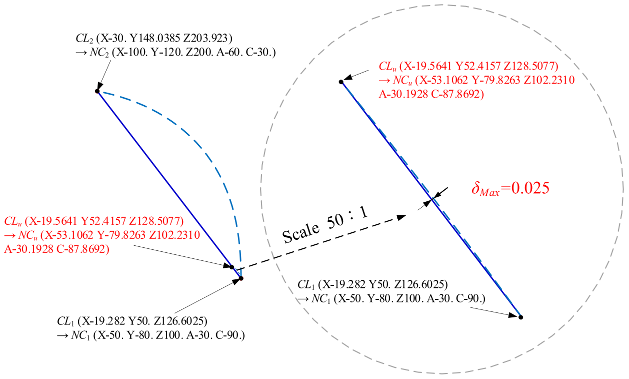

CL_2 (X-30. Y148.0385 Z203.923)
→ NC_2 (X-100. Y-120. Z200. A-60. C-30.)
CL_u (X-19.5641 Y52.4157 Z128.5077)
→ NC_u (X-53.1062 Y-79.8263 Z102.2310
A-30.1928 C-87.8692)
δ_{Max}=0.025
CL_u (X-19.5641 Y52.4157 Z128.5077)
→ NC_u (X-53.1062 Y-79.8263 Z102.2310
A-30.1928 C-87.8692)
Scale 50 : 1
CL_1 (X-19.282 Y50. Z126.6025)
→ NC_1 (X-50. Y-80. Z100. A-30. C-90.)
CL_1 (X-19.282 Y50. Z126.6025)
→ NC_1 (X-50. Y-80. Z100. A-30. C-90.)

Fig. 12 Interpolate CL_u points according to δ_{Max}

The drawing results for the two negative (Neg.) solutions of CL_1 and CL_u are used to confirm that δ_{Max} is 0.025 mm, which is within the error range for this study, so CL_u is the point that is calculated for this study, as shown in Fig. 12. If δ_{Max} is known, the CL_{u1} position of the first interpolation point for CL_1 can be obtained. CL_{u1} is then used to calculate the next CL_{u2} until the relationship between CL_{uN} and CL_2 is also within δ_{Max}, as shown in Fig. 13. There are N interpolation points between two CLs. The value of N depends on the δ between each two CL points so there is no need to perform linear interpolation with the same equal spacing between all CLs. If δ_{Max} is used to determine the interpolation method for the next CL_u point, N interpolation points are not required between two CL points to correct the path

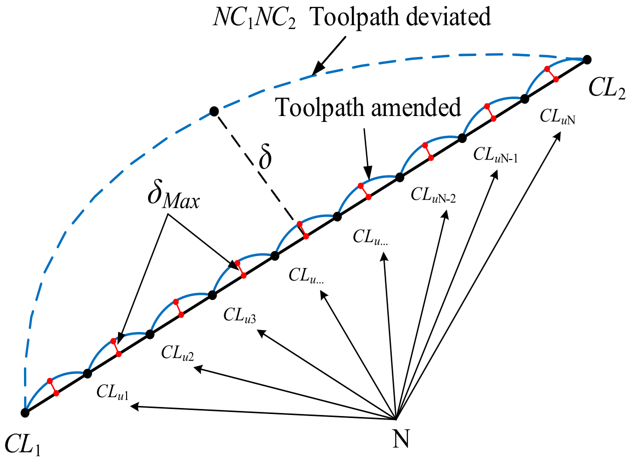


Fig. 13 Calculated interpolated CL_u point according to the δ_{Max}

Table 4 Two CL points and their tool axis vectors

	CL_x	CL_y	CL_z	i	j	k
CL_1	-20.	-10.	0.	0.44721	0.00000	0.89443
CL_2	10.	20.	30.	0.03161	0.03161	0.99900

Table 5 δ_{Max} and the number of interpolation points N

δ_{Max}	0.001	0.002	0.003	0.004	0.005
N	106	75	61	53	48
δ_{Max}	0.01	0.02	0.03	0.04	0.05
N	34	24	20	17	15
δ_{Max}	0.06	0.07	0.08	0.08	0.1
N	14	13	12	12	11

deviation. For two CL points that are generated by CAM software, for CL_1 point $(-20, -10, 0)$, the tool axis unit vector is (0.44721, 0, 0.89443), and for CL_2 point (10, 20, 30), the tool axis unit vector is (0.03161, 0.03161, 0.999) and d_{oc} is (0, 0, 0), and d_{ca} is (0, 0, 0), as shown in Table 4.

The approximate solution value for the maximum δ for the two negative (Neg.) solutions of the path deviations for CL_1 and CL_2 is calculated as $\delta = 11.24403$ mm. The number N of CL_u points to be interpolated using different values for δ_{Max} is calculated as shown in Table 5. If δ_{Max} is smaller, the number of interpolation points N that is required increases significantly. If δ_{Max} is less than 0.01 mm, the required number of interpolation points N is significantly increased. The data in Table 5 is used to plot the relationship between δ_{Max} and the number of interpolation points N, as shown in Fig. 14.

The results in Table 5 show that the relationship between δ between two CL points, the number of interpolation points N, and δ_{Max} is expressed using Eq. (5), so this study is defined as the RUIIM. The required number of interpolation points N_{equ} is unconditionally rounded up to the nearest integer.

$$N_{equ} = \sqrt{\delta / \delta_{Max}} \tag{5}$$

where N_{equ} is the number of interpolation points, δ is the five-axis error between two CL points, and δ_{Max} is the maximum error. A comparison of the number of interpolation points N_{equ} that is calculated using the RUIIM and the number of N points that are interpolated as calculated using δ_{Max} in Table 5 shows that the number of interpolation points N_{equ} and N is almost the same, as shown in Table 6. In Table 6, there is an error between N_{equ} and N, but this is within the range of one interpolation point. The time that is required to calculate the point and processing efficiency are improved by using the RUIIM to divide the CL points that are output by the CAM software into unequal interval interpolation points to achieve optimal segmental interpolation. The RUIIM is used to calculate the required number of interpolation points N_{equ} between the two CL points to correct the δ for the path deviation. The two CL points in Table 4 are interpolated

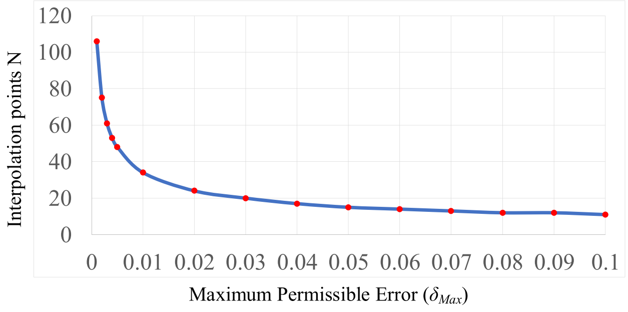


Fig. 14 Curve of δ_{Max} and the number of interpolation points N

Table 6 δ_{Max} interpolation point number N and RUIIM solution interpolation point number N_{equ}

δ_{Max}	0.001	0.002	0.003	0.004	0.005
N	106	75	61	53	48
N_{equ}	107	75	62	54	48
δ_{Max}	0.01	0.02	0.03	0.04	0.05
N	34	24	20	17	15
N_{equ}	34	24	20	17	15
δ_{Max}	0.06	0.07	0.08	0.09	0.1
N	14	13	12	12	11
N_{equ}	14	13	12	12	11

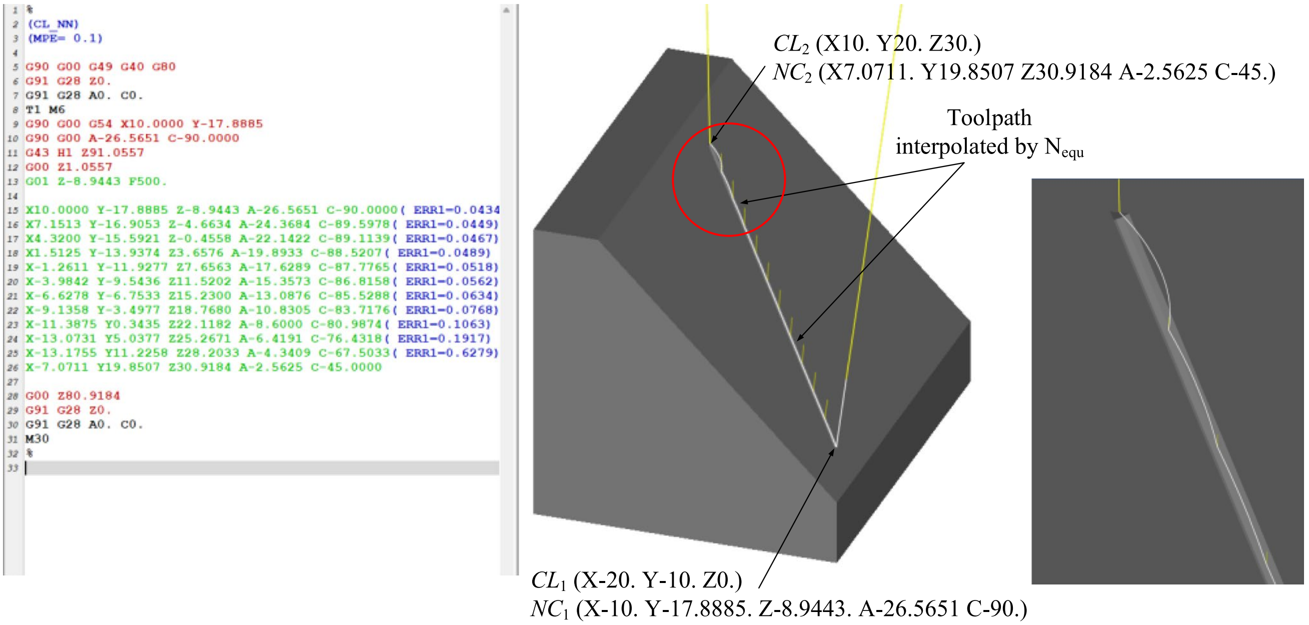
using the RUIIM to achieve unequal interval CL interpolation between the five axes.

In Table 4, the two CL points that are generated by the CAM software are CL_1 $(-20, -10, 0)$, for which the tool axis unit vector is (0.44721, 0, 0.89443), and CL_2 (10, 20, 30), for which the tool axis unit vector is (0.03161, 0.03161, 0.999). The value of d_{oc} is (0, 0, 0) and of d_{ca} is (0, 0, 0). If $\delta_{Max} = 0.1$ mm, as shown in Table 6, the number of interpolation points that is required to correct the δ for the two CL points is $N = N_{equ} = 11$. The number of interpolation points between CL_1 and CL_2 is linearly interpolated by 11 equal divisions. The NC simulation screen with N_{equ} as the first interpolation point is shown in Fig. 15.

Figure 15 shows that the path offset correction between two CLs $(CL_1 \rightarrow CL_2)$ is significantly improved using RUIIM. However, the movement between some interpolation points is not completely corrected to less than δ_{Max}. Five-axis machining involves the movement of three linear axes $(X, Y$, and $Z)$ and the motion of any two rotational axes (such as AB, AC, or BC) so some motion paths between interpolation points are not within δ_{Max}. If the δ value between the two CL points is not within δ_{Max}, the two adjacent CL points are solved using the RUIIM again until all the calculated values for δ for the interpolation point are within the δ_{Max} value. In Fig. 15, using the first RUIIM, there are still some motions between interpolation points that are greater than δ_{Max} so a second RUIIM is used to produce the interpolation points. The RUIIM continues to interpolate points that have a range that is greater than δ_{Max}. The NC program adds the interpolated single-block program here. The NC simulation screen for N_{equ} for the second interpolation point is shown in Fig. 16.

Figure 16 shows that if only the first points are interpolated for the path deviation between two CL points, some CL interpolation points do not have a range within δ_{Max} and a second interpolation point is required. After the second interpolation point is produced for the relevant path deviation interval, there are still some motions between the interpolation points that are greater than δ_{Max}. The RUIIM that is proposed by this study is used repeatedly to interpolate the path deviation interval until the path deviation interval is within δ_{Max}. The path deviation correction between the two CL points $(CL_1 \rightarrow CL_2)$ is complete. The NC simulation screen for N_{equ} for the third interpolation point is shown in Fig. 17.

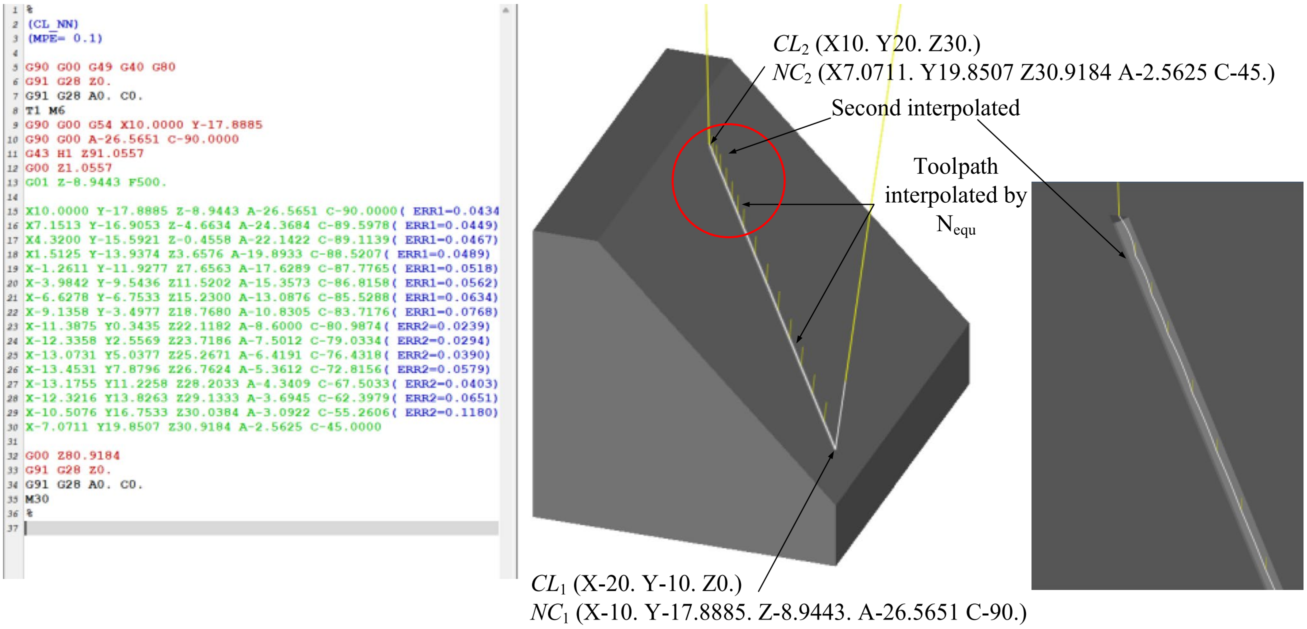
The optimal method for interpolation points is to define δ_{Max}. The RUIIM is used to move the CL points that are generated by the CAM software, and the deviation paths are processed separately. In Table 4, for $\delta_{Max} = 0.01$ mm, the original two CL points are interpolated to interpolate CL with an unequal interval between the five axes, as shown in Fig. 18. To add interpolation points at equal intervals between all CL points under the same conditions, multiple interpolation points are produced using the RUIIM. As shown in Fig. 19, for the path deviation problem in Fig. 9, the optimal result is obtained by interpolating points with unequal spacing CL using the RUIIM.



```
%
(CL_NN)
(MPE= 0.1)

G90 G00 G49 G40 G80
G91 G28 Z0.
G91 G28 A0. C0.
T1 M6
G90 G00 G54 X10.0000 Y-17.8885
G90 G00 A-26.5651 C-90.0000
G43 H1 Z91.0557
G00 Z1.0557
G01 Z-8.9443 F500.

X10.0000 Y-17.8885 Z-8.9443 A-26.5651 C-90.0000( ERR1=0.0434
X7.1513 Y-16.9053 Z-4.6634 A-24.3684 C-89.5978( ERR1=0.0449)
X4.3200 Y-15.5921 Z-0.4558 A-22.1422 C-89.1139( ERR1=0.0467)
X1.5125 Y-13.9374 Z3.6576 A-19.8933 C-88.5207( ERR1=0.0489)
X-1.2611 Y-11.9277 Z7.6563 A-17.6289 C-87.7765( ERR1=0.0518)
X-3.9842 Y-9.5436 Z11.5202 A-15.3573 C-86.8158( ERR1=0.0562)
X-6.6278 Y-6.7533 Z15.2300 A-13.0876 C-85.5288( ERR1=0.0634)
X-9.1358 Y-3.4977 Z18.7680 A-10.8305 C-83.7176( ERR1=0.0768)
X-11.3875 Y0.3435 Z22.1182 A-8.6000 C-80.9874( ERR1=0.1063)
X-13.0731 Y5.0377 Z25.2671 A-6.4191 C-76.4318( ERR1=0.1917)
X-13.1755 Y11.2258 Z28.2033 A-4.3409 C-67.5033( ERR1=0.6279)
X-7.0711 Y19.8507 Z30.9184 A-2.5625 C-45.0000

G00 Z80.9184
G91 G28 Z0.
G91 G28 A0. C0.
M30
%
```

CL_2 (X10. Y20. Z30.)
NC_2 (X7.0711. Y19.8507 Z30.9184 A-2.5625 C-45.)
Toolpath interpolated by N_{equ}
CL_1 (X-20. Y-10. Z0.)
NC_1 (X-10. Y-17.8885. Z-8.9443. A-26.5651 C-90.)

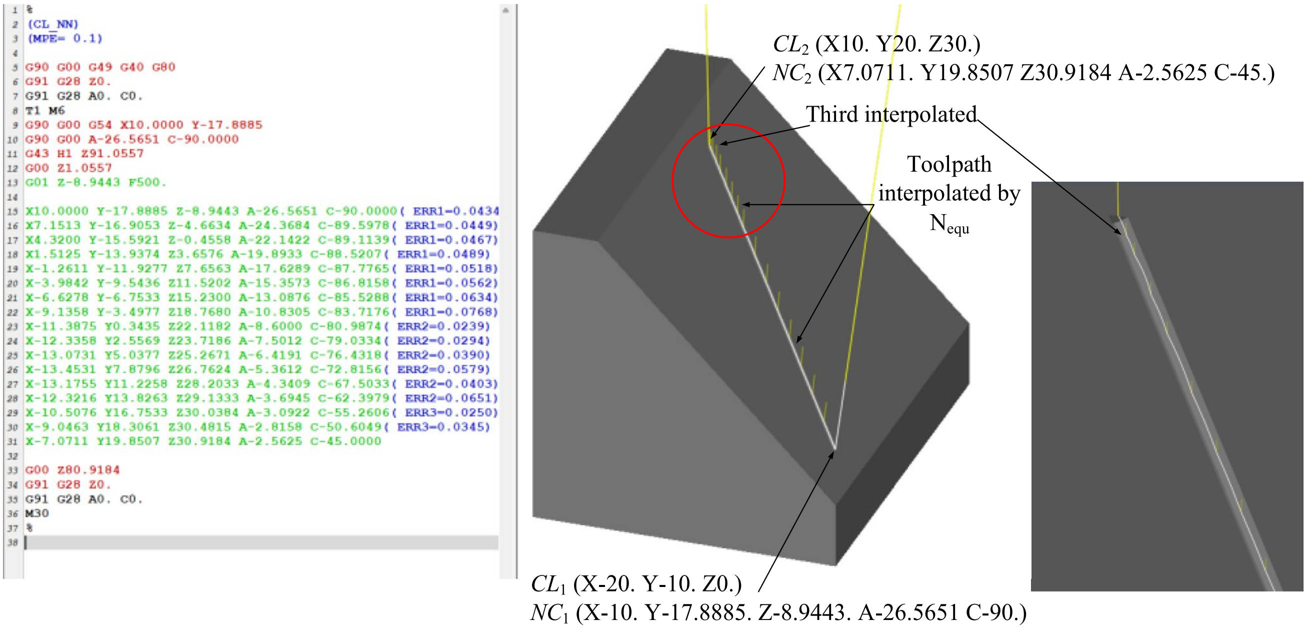
Fig. 15 NC simulation screen with N_{equ} as the first interpolation point



```
%
(CL_NN)
(MPE= 0.1)

G90 G00 G49 G40 G80
G91 G28 Z0.
G91 G28 A0. C0.
T1 M6
G90 G00 G54 X10.0000 Y-17.8885
G90 G00 A-26.5651 C-90.0000
G43 H1 Z91.0557
G00 Z1.0557
G01 Z-8.9443 F500.

X10.0000 Y-17.8885 Z-8.9443 A-26.5651 C-90.0000( ERR1=0.0434
X7.1513 Y-16.9053 Z-4.6634 A-24.3684 C-89.5978( ERR1=0.0449)
X4.3200 Y-15.5921 Z-0.4558 A-22.1422 C-89.1139( ERR1=0.0467)
X1.5125 Y-13.9374 Z3.6576 A-19.8933 C-88.5207( ERR1=0.0489)
X-1.2611 Y-11.9277 Z7.6563 A-17.6289 C-87.7765( ERR1=0.0518)
X-3.9842 Y-9.5436 Z11.5202 A-15.3573 C-86.8158( ERR1=0.0562)
X-6.6278 Y-6.7533 Z15.2300 A-13.0876 C-85.5288( ERR1=0.0634)
X-9.1358 Y-3.4977 Z18.7680 A-10.8305 C-83.7176( ERR1=0.0768)
X-11.3875 Y0.3435 Z22.1182 A-8.6000 C-80.9874( ERR2=0.0239)
X-12.3358 Y2.5569 Z23.7186 A-7.5012 C-79.0334( ERR2=0.0294)
X-13.0731 Y5.0377 Z25.2671 A-6.4191 C-76.4318( ERR2=0.0390)
X-13.4531 Y7.8796 Z26.7624 A-5.3612 C-72.8156( ERR2=0.0579)
X-13.1755 Y11.2258 Z28.2033 A-4.3409 C-67.5033( ERR2=0.0403)
X-12.3216 Y13.8263 Z29.1333 A-3.6945 C-62.3979( ERR2=0.0651)
X-10.5076 Y16.7533 Z30.0384 A-3.0922 C-55.2606( ERR2=0.1180)
X-7.0711 Y19.8507 Z30.9184 A-2.5625 C-45.0000

G00 Z80.9184
G91 G28 Z0.
G91 G28 A0. C0.
M30
%
```

CL_2 (X10. Y20. Z30.)
NC_2 (X7.0711. Y19.8507 Z30.9184 A-2.5625 C-45.)
Second interpolated
Toolpath interpolated by N_{equ}
CL_1 (X-20. Y-10. Z0.)
NC_1 (X-10. Y-17.8885. Z-8.9443. A-26.5651 C-90.)

Fig. 16 NC simulation screen with N_{equ} as the second interpolation point



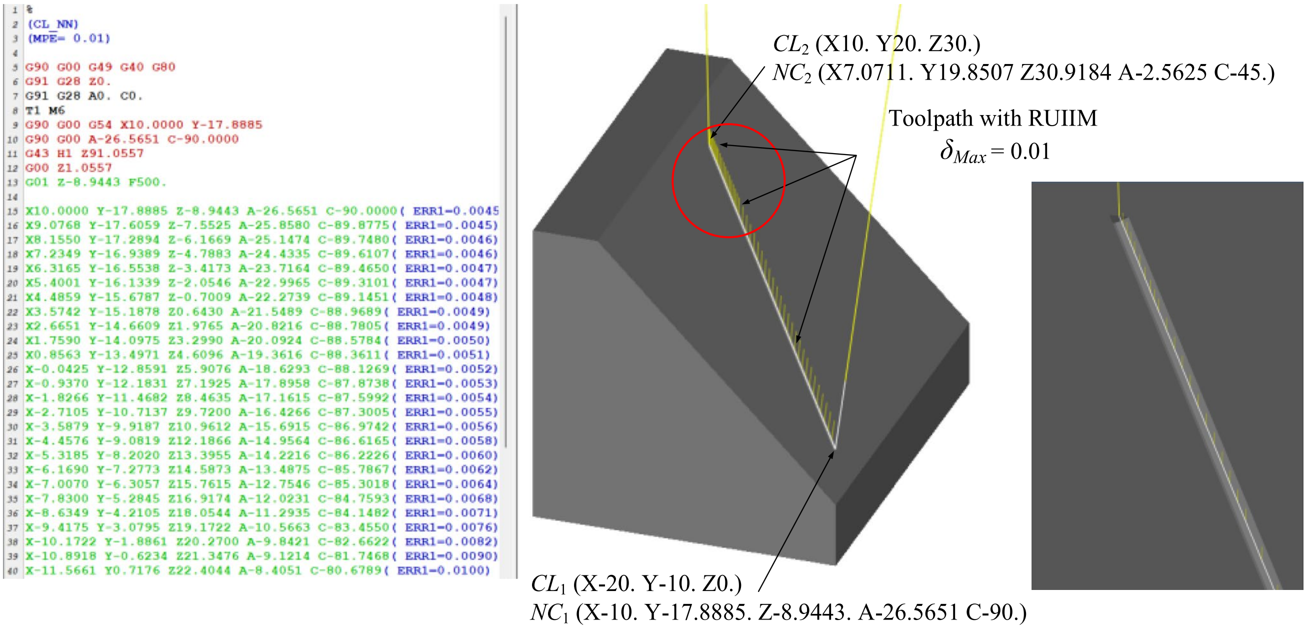
```
%
(CL_NN)
(MPE= 0.1)

G90 G00 G49 G40 G80
G91 G28 Z0.
G91 G28 A0. C0.
T1 M6
G90 G00 G54 X10.0000 Y-17.8885
G90 G00 A-26.5651 C-90.0000
G43 H1 Z91.0557
G00 Z1.0557
G01 Z-8.9443 F500.

X10.0000 Y-17.8885 Z-8.9443 A-26.5651 C-90.0000( ERR1=0.0434
X7.1513 Y-16.9053 Z-4.6634 A-24.3684 C-89.5978( ERR1=0.0449)
X4.3200 Y-15.5921 Z-0.4558 A-22.1422 C-89.1139( ERR1=0.0467)
X1.5125 Y-13.9374 Z3.6576 A-19.8933 C-88.5207( ERR1=0.0489)
X-1.2611 Y-11.9277 Z7.6563 A-17.6289 C-87.7765( ERR1=0.0518)
X-3.9842 Y-9.5436 Z11.5202 A-15.3573 C-86.8158( ERR1=0.0562)
X-6.6278 Y-6.7533 Z15.2300 A-13.0876 C-85.5288( ERR1=0.0634)
X-9.1358 Y-3.4977 Z18.7680 A-10.8305 C-83.7176( ERR1=0.0768)
X-11.3875 Y0.3435 Z22.1182 A-8.6000 C-80.9874( ERR2=0.0239)
X-12.3358 Y2.5569 Z23.7186 A-7.5012 C-79.0334( ERR2=0.0294)
X-13.0731 Y5.0377 Z25.2671 A-6.4191 C-76.4318( ERR2=0.0390)
X-13.4531 Y7.8796 Z26.7624 A-5.3612 C-72.8156( ERR2=0.0579)
X-13.1755 Y11.2258 Z28.2033 A-4.3409 C-67.5033( ERR2=0.0403)
X-12.3216 Y13.8263 Z29.1333 A-3.6945 C-62.3979( ERR2=0.0651)
X-10.5076 Y16.7533 Z30.0384 A-3.0922 C-55.2606( ERR3=0.0250)
X-9.0463 Y18.3061 Z30.4815 A-2.8158 C-50.6049( ERR3=0.0345)
X-7.0711 Y19.8507 Z30.9184 A-2.5625 C-45.0000

G00 Z80.9184
G91 G28 Z0.
G91 G28 A0. C0.
M30
%
```

CL_2 (X10. Y20. Z30.)
NC_2 (X7.0711. Y19.8507 Z30.9184 A-2.5625 C-45.)
Third interpolated
Toolpath interpolated by N_{equ}
CL_1 (X-20. Y-10. Z0.)
NC_1 (X-10. Y-17.8885. Z-8.9443. A-26.5651 C-90.)

Fig. 17 NC simulation screen with N_{equ} as the third interpolation point

5 Verification experiment

A five-axis toolpath was created using Mastercam software to generate CL data, and the RUIIM was used to correct the path deviation when the CL point is directly converted into an NC program. CIMCO software was used to simulate the path trajectory for the NC program, and VERICUT was used to confirm the result before processing. CIMCO and VERICUT are currently the most commonly used NC program software for verification before five-axis machining in industry and ensure the safety of five-axis machining. For this study, the tool axis is normal



```
%
(CL_NN)
(MPE= 0.01)

G90 G00 G49 G40 G80
G91 G28 Z0.
G91 G28 A0. C0.
T1 M6
G90 G00 G54 X10.0000 Y-17.8885
G90 G00 A-26.5651 C-90.0000
G43 H1 Z91.0557
G00 Z1.0557
G01 Z-8.9443 F500.

X10.0000 Y-17.8885 Z-8.9443 A-26.5651 C-90.0000( ERR1=0.0045
X9.0768 Y-17.6059 Z-7.5525 A-25.8580 C-89.8775( ERR1=0.0045)
X8.1550 Y-17.2894 Z-6.1669 A-25.1474 C-89.7480( ERR1=0.0046)
X7.2349 Y-16.9389 Z-4.7883 A-24.4335 C-89.6107( ERR1=0.0046)
X6.3165 Y-16.5538 Z-3.4173 A-23.7164 C-89.4650( ERR1=0.0047)
X5.4001 Y-16.1339 Z-2.0546 A-22.9965 C-89.3101( ERR1=0.0047)
X4.4859 Y-15.6787 Z-0.7009 A-22.2739 C-89.1451( ERR1=0.0048)
X3.5742 Y-15.1878 Z0.6430 A-21.5489 C-88.9689( ERR1=0.0049)
X2.6651 Y-14.6609 Z1.9765 A-20.8216 C-88.7805( ERR1=0.0049)
X1.7590 Y-14.0975 Z3.2990 A-20.0924 C-88.5784( ERR1=0.0050)
X0.8563 Y-13.4971 Z4.6096 A-19.3616 C-88.3611( ERR1=0.0051)
X-0.0425 Y-12.8591 Z5.9076 A-18.6293 C-88.1269( ERR1=0.0052)
X-0.9370 Y-12.1831 Z7.1925 A-17.8958 C-87.8738( ERR1=0.0053)
X-1.8266 Y-11.4682 Z8.4635 A-17.1615 C-87.5992( ERR1=0.0054)
X-2.7105 Y-10.7137 Z9.7200 A-16.4266 C-87.3005( ERR1=0.0055)
X-3.5879 Y-9.9187 Z10.9612 A-15.6915 C-86.9742( ERR1=0.0056)
X-4.4576 Y-9.0819 Z12.1866 A-14.9564 C-86.6165( ERR1=0.0058)
X-5.3185 Y-8.2020 Z13.3955 A-14.2216 C-86.2226( ERR1=0.0060)
X-6.1690 Y-7.2773 Z14.5873 A-13.4875 C-85.7867( ERR1=0.0062)
X-7.0070 Y-6.3057 Z15.7615 A-12.7546 C-85.3018( ERR1=0.0064)
X-7.8300 Y-5.2845 Z16.9174 A-12.0231 C-84.7593( ERR1=0.0068)
X-8.6349 Y-4.2105 Z18.0544 A-11.2935 C-84.1482( ERR1=0.0071)
X-9.4175 Y-3.0795 Z19.1722 A-10.5663 C-83.4550( ERR1=0.0076)
X-10.1722 Y-1.8861 Z20.2700 A-9.8421 C-82.6622( ERR1=0.0082)
X-10.8918 Y-0.6234 Z21.3476 A-9.1214 C-81.7468( ERR1=0.0090)
X-11.5661 Y0.7176 Z22.4044 A-8.4051 C-80.6789( ERR1=0.0100)
```

CL_2 (X10. Y20. Z30.)
NC_2 (X7.0711. Y19.8507 Z30.9184 A-2.5625 C-45.)
Toolpath with RUIIM
$\delta_{Max} = 0.01$
CL_1 (X-20. Y-10. Z0.)
NC_1 (X-10. Y-17.8885. Z-8.9443. A-26.5651 C-90.)

Fig. 18 The NC simulation screen of the unequal interval CL interpolation points among the five axes

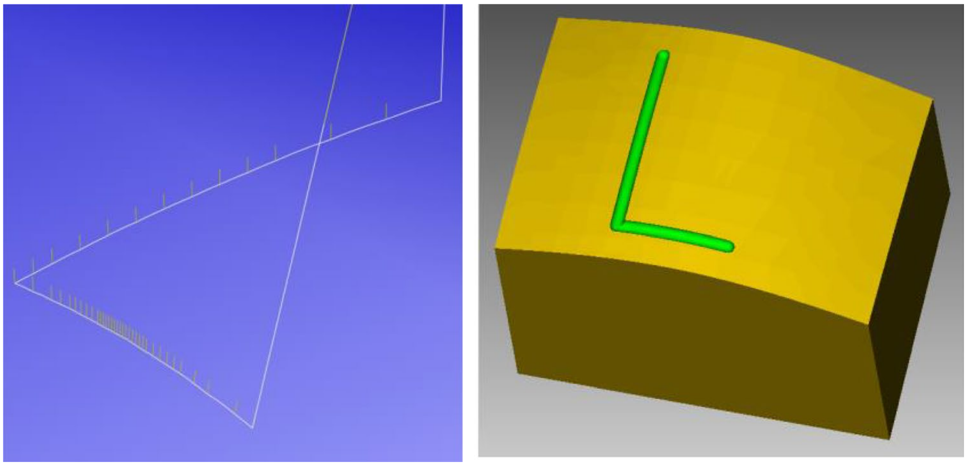


Fig. 19 NC simulation screen of RUIIM

to the surface of the workpiece. Helical pattern milling is used to verify that if the RUIIM is used after the CAM software is corrected to generate the CL point and the NC program is directly output to the CL point, there is deviation from the toolpath and an accurate NC program is obtained for five-axis machining. The NC program that is corrected using the RUIIM is then used for cutting on a TATC five-axis machining machine to verify the method of this study. The cutting tolerance for the Mastercam software is 0.025 mm, as shown in Fig. 20. The inclination angle of the tool axis vector V_t varies with the position of the machined surface.

Using the CL point that is calculated by the CAM software, the output NC program is directly converted and the path trajectory is simulated using CIMCO (Fig. 21a). VERICUT is used to verify solid cutting (Fig. 21b). Figure 21 shows that many toolpaths exhibit deviation, so the cutting paths are different to the required cutting path, which produces partial overcuts. The CL point that is generated by the CAM software is set to δ_{Max} which is 0.025 mm. Using the proposed RUIIM, CIMCO simulates the path trajectory (Fig. 22a) and VERICUT verifies the solid cutting (Fig. 22b). The results in Fig. 22 show that the part for which the toolpath deviates is corrected to the correct one by the interpolation using the RUIIM.

For helical pattern milling, the RUIIM that is proposed in this study corrects the NC program that originally generates the toolpath deviation and the range of deviation is less than

Fig. 20 Five-axis helical milling path and tool axis vector V_t

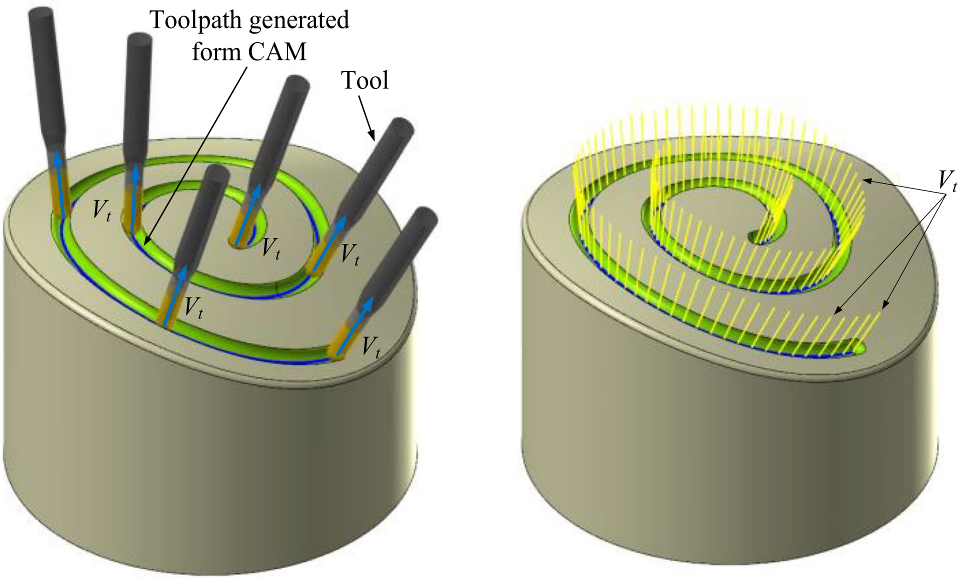
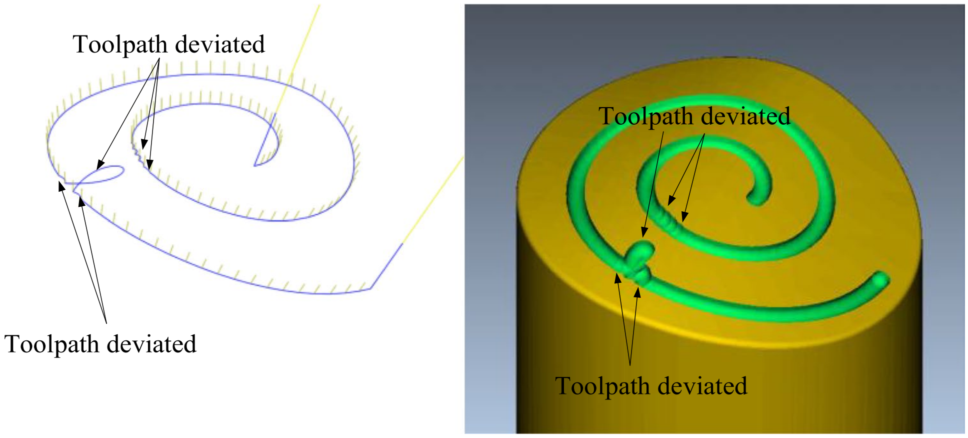


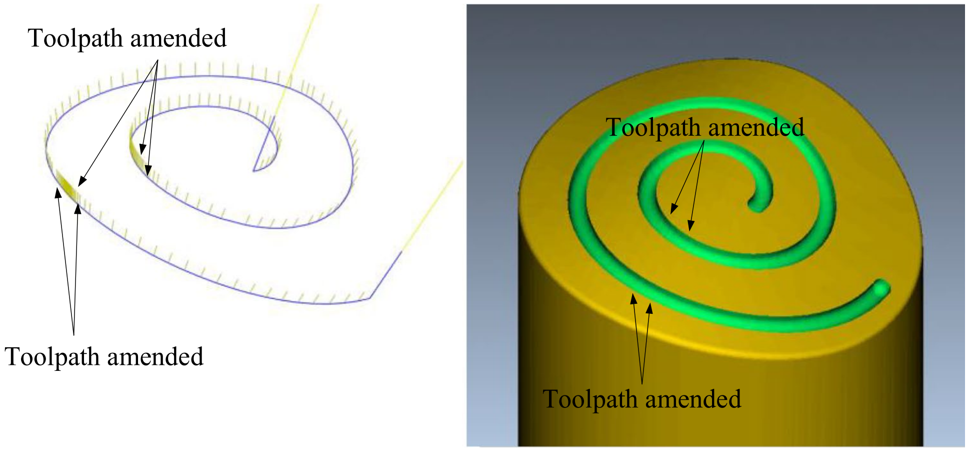
Fig. 21 Five-axis path helical milling path simulation (pure linear interpolation)



(a) Simulated by CIMCO

(b) Simulated by VERICUT

Fig. 22 Five-axis path helix path simulation (RUIIM)



(a) Simulated by CIMCO

(b) Simulated by VERICUT

δ_{Max}. CIMCO and VERICUT software show that processing is accurate before the actual five-axis processing. For the NC program for helical milling, the results for the RUIIM are compared with those for the equal-spaced interpolation method. If the path length for one cutting pass is 212.2 mm and the equal spacing is 0.1 mm, there are 2120 blocks for one pass. For a value of $\delta_{Max} = 0.025$ mm, there are 296 blocks, so 6.16 times more blocks are used for the equal-spaced interpolation method than for the RUIIM, as shown in Table 7. This result is for one pass. If there are multiple passes used for layered cutting, the number of blocks for the NC program will depend on the number of layers for one pass. If the equidistant interpolation distance is increased to 0.5 mm, the NC program requires 426 blocks but the deviation of the toolpath is not completely corrected, as shown in Fig. 23.

In order to verify that the RUIIM corrects the NC program that originally generates the toolpath deviation to within δ_{Max}, this study uses a five-axis machining machine that is produced by Baide Machinery for machine cutting verification. The model is a MF400U, the controller uses HEIDENHAIN TNC 640, and the axial strokes are X-axis 410 mm, Y-axis 610 mm, Z-axis 510 mm, A-axis + 30° ~ − 120°, and C-axis 360° (continuous). The working turntable is 400 mm in diameter and the maximum spindle speed is 12,000 rpm. Before the verification process, it is necessary to perform a five-axis rotation center correction to confirm the offset values for d_{oc} and d_{ca}. This is an extremely important parameter for non-RTCP. The outer diameter of the test piece and the datum plane was first turned using a CNC lathe to give a workpiece for verification with dimensions of Ø65 mm and 95 mm in length, which was clamped in a concentric vise. Figure 24 shows the processing machine, schematic diagrams of each axial direction, and specimen clamping.

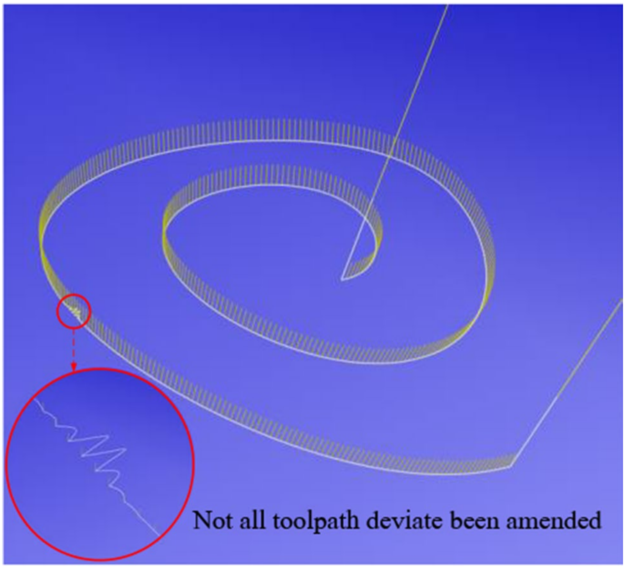
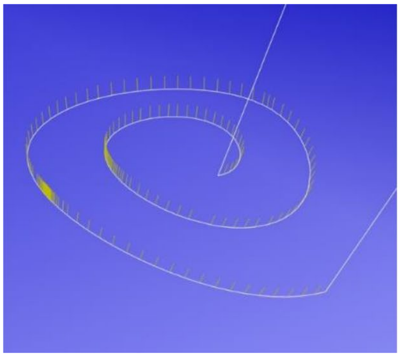
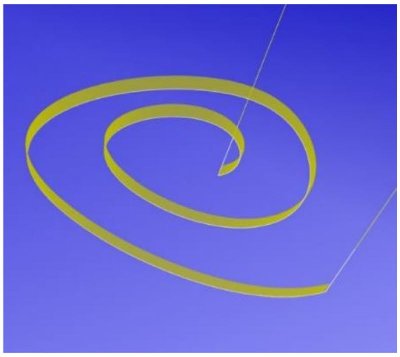


Fig. 23 Helical milling with equal interval interpolation 0.5 mm

A three-axis surface path was used for dynamic roughing to ensure surface optimization using a Ø10 mm flat cutter.

Table 7 Comparison of helical milling NC programs

Cutting length 212.2mm (calculated by cutting 1 pass)	
RUIIM δ_{Max}=0.025mm	Equal interval interpolation method Interval 0.1mm
	
The number of NC cutting programs is 296 blocks	The number of NC cutting programs is 2,120 blocks
	Block increase ratio = 616%

A Ø10 mm ball cutter was then used to perform equidistant wrapping finishing, as shown in Fig. 25. A Ø3 mm ball cutter was used to process the helix on the curved surface. The tool axis is normal to the surface, and the machining depth is 1.5 mm to generate the CL point of the toolpath. The CAM software was then simulated and verified, as shown in Fig. 26.

The NC program that is directly output from the CL point (Fig. 27a) and the RUIIM were then used (Fig. 27b). The value for δ_{Max} is 0.025 mm. After correcting the path deviation for the CL point, the output NC program is compared with the actual machining. Using a simulation in CIMCO and VERICUT software, the results for pure linear interpolation and RUIIM are consistent, as shown in Fig. 27. Some of the helical cutting positions are incorrect because of the δ relationship so the cutting path is incorrect, as shown in Fig. 27a. When the deviation is corrected, the overcut positions that were not correct using the original cutting path are corrected, as shown in Fig. 27b. The machining results for milling a helical pattern milling show that the RUIIM corrects the deviation in the original tool path for the NC program to less than the value of δ_{Max}.

6 Conclusion

Many five-axis machines currently use RTCP controllers but some controllers do not feature RTCP commands or post-processing using CAM software, so they do not output NC programs that support RTCP. This study allows controllers that do not support RTCP instructions or have CAM software that does not support post-processing output RTCP to calculate the interpolation points for a toolpath, in order to prevent overcutting of the workpiece due to an error in the cutting path. In the non-RTCP mode, five-axis machining features linear motion in the original *XYZ* plane and motion around two rotational axes (such as AB, AC, or BC). If three-axis linear interpolation is also used for processing, five-axis errors result in incorrect or overcut paths. Some current CAM software can only perform equidistant

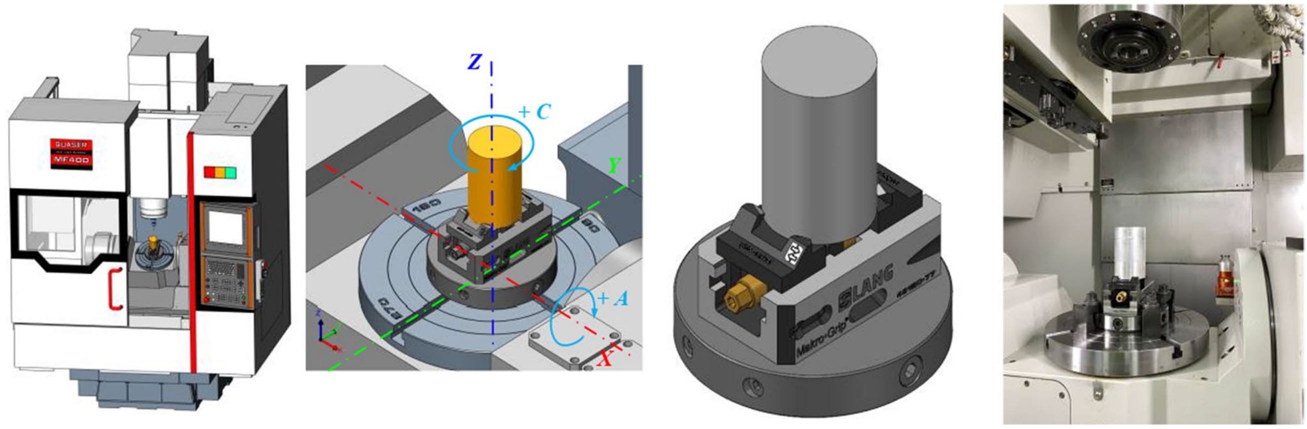


Fig. 24 Appearance, schematic diagram of each axis, and specimen clamping of five-axis machine MF400U

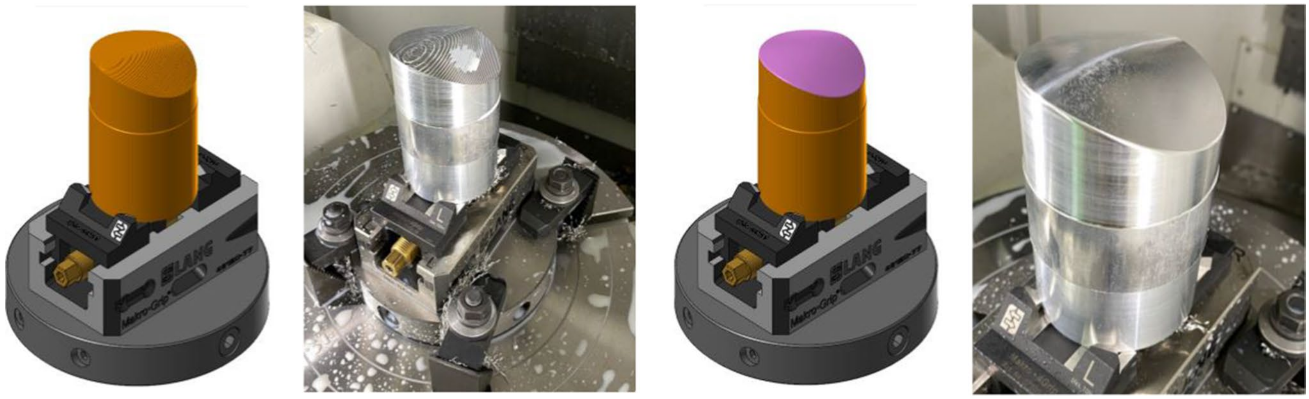


Fig. 25 Pre-machined specimen before helical milling validation

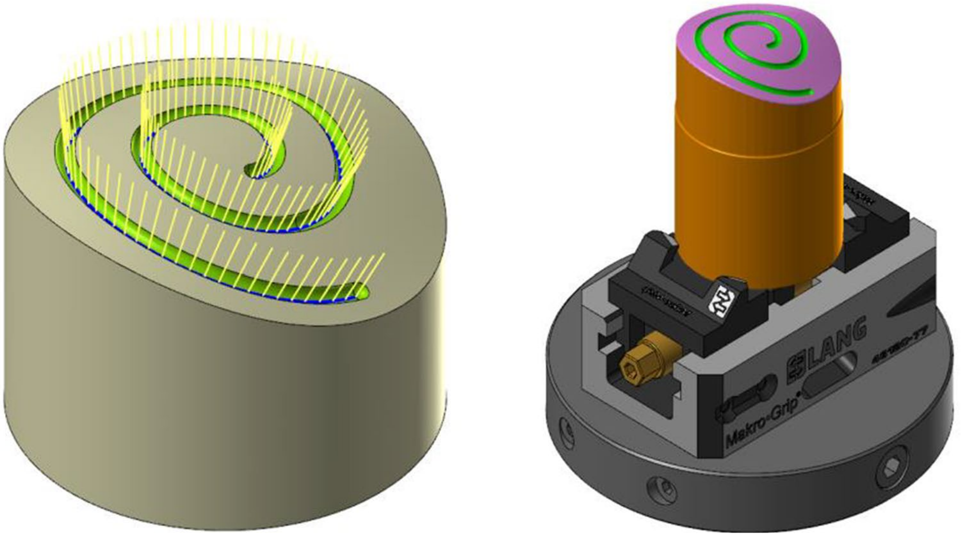
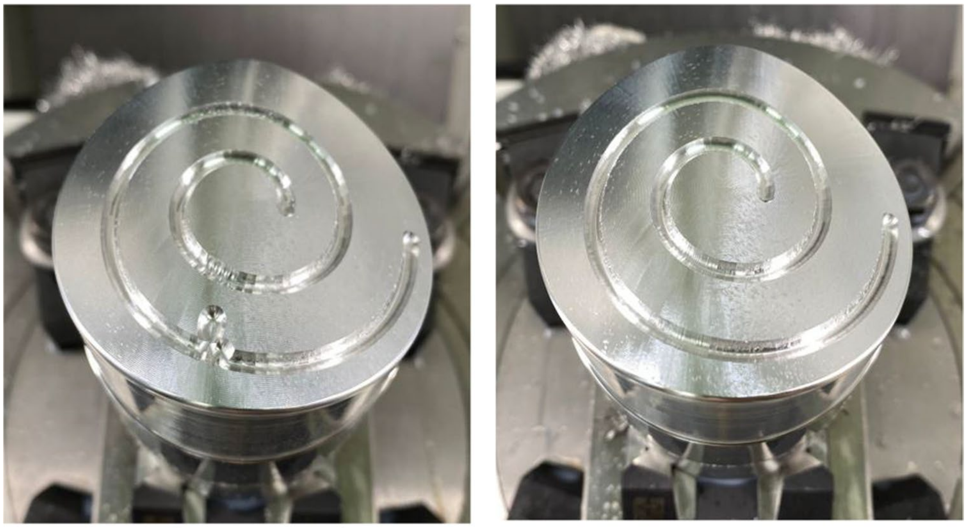


Fig. 26 Simulation and verification of helical milling in CAM software



(a) Pure linear interpolation

(b) RUIIM

Fig. 27 Comparison of actual machining of helical milling

interpolation between CL points to ensure that δ_u is within tolerance so too many interpolation points are generated. This results in a significant increase in the number of blocks that the NC program uses, which also affects the processing speed and processing efficiency of the entire controller.

This study proposes a RUIIM that uses the C# programming language to quickly calculate the CL point coordinates (CL_x, CL_y, CL_z) and the tool axis vector $V_t\,(i,j,k)$. This system corrects the value for δ to less than the range δ_{Max} and corrects the δ value for the original toolpath more quickly. If the output CL point coordinates (CL_x, CL_y, CL_z) and the planned tool axis vector $V_t\,(i,j,k)$ are calculated using CAM software, the CL point can be optimized by segment interpolation and this data is output to a correct NC program to generate the machining toolpath.

Author contribution W-CC: conceptualization, conduct experimental process, data curation, characterization, validation, analysis, and writing—original draft. T-KL: advisor and methodology. H-MT: conduct experimental process and data curation. C-CT: conceptualization, project administration, funding acquisition, revised grammar, and writing—review and editing draft. All authors have read and agreed to the published version of the manuscript.

Funding The authors appreciate the financial support of the Ministry of Science and Technology of the Republic of China (MOST 108–2221-E-262–002-MY2 and MOST 110–2622-E-262–004).

Data availability All necessary data are shown in the figures and tables within the document. The raw data can be made available upon request.

Declarations

Ethics approval Not applicable.

Consent to participate Not applicable.

Consent for publication Not applicable.

Conflict of interest The authors declare no competing interests.

References

1. Sakamoto S, Inasaki I (1993) Analysis of generating motion for five-axis machining centers. The Japan Society of Mechanical Engineers 59(561):1553–1559
2. Lin AC, Lin TK, Lin TD (2014) Derivinggeneric transformation matrices for multi-axis milling machine. International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering 8(7):2625–2629
3. Lee RS, She CH (1997) Developing a postprocessor for three types of five-axis machine tools. Int J Adv Manuf Technol 13:658–665
4. Takeuchi Y, Watanabe T (1992) Generation of 5-axis control collision-free tool path and postprocessing for NC data. CIRP Ann 41:539–542
5. Jung YH, Lee DW, Kim JS, Mok HS (2002) NC post-processor for 5-axis milling machine of table-rotating or tilting type. J Mater Process Technol 130–131:641–646
6. Vavruska P (2016) Interpolation of toolpath by a postprocessor for increased accuracy in multi-axis machining. Procedia CIRP 46:336–339
7. Lai YL, Liao CC, Chao ZG (2018) Inverse kinematics for a novel hybrid parallel–serial five-axis machine tool. Robotics and Computer-Integrated Manufacturing 50:63–79
8. Lin TD, Lin AC (2014) Using interpolation points to rectify cutter paths of five-axis machining. Appl Mech Mater 275–277:2625–2629
9. Yu DY, Ding Z (2019) Post-processing algorithm of a five-axis machine tool with dual rotary tables based on the TCS method. The International Journal of Advanced Manufacturing Technology 102:3937–3944

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.