Thuật toán tuyến tính hóa đường chạy dao và xây dựng bộ hậu xử lý cho máy CNC 5 trục

Tool path linearization algorithm and postprocessor for five-axis CNC machine

Tran Duc Tang, Nguyen Phu Thuy, Vuong Si Kong
Le Quy Don Technical University
e-Mail: tangtd@gmail.com

Tóm tắt
Khi tạo dữ liệu đường chạy dao (CL data) các hệ thống CAM giả định là đường chạy dao giữa hai điểm mũi dao liên tiếp là đường thẳng. Tuy nhiên điều này chỉ đúng đối với máy có ba trục chuyển động thẳng đồng thời; đối với máy năm trục, do có thêm hai trục quay nên đường chạy dao giữa hai điểm liên tiếp trong chương trình NC sẽ không tuyến tính, vì vậy làm giảm độ chính xác khi gia công. Bài báo này trình bày thuật toán tuyến tính hóa đường chạy dao và xây dựng bộ hậu xử lý (postprocessor) cho máy CNC 5 trục nhằm giảm sai số đường chạy dao nói trên. Dựa trên phương pháp đã tính toán, các tác giả xây dựng một mô đun phần mềm trên nền windows bằng ngôn ngữ lập trình Visual Basic. Kết quả mô phỏng kiểm chứng, áp dụng cho máy phay CNC 5 trục với hai trục quay trên bàn gá phôi (Deckel Maho DMU 70 eVoluion), chỉ ra rằng bộ hậu xử lý có tính toán tuyến tính hóa đường chạy dao làm giảm sai số đường chạy dao, bộ hậu xử lý đáng tin cậy và có thể tùy biến để áp dụng cho các kiểu cấu hình máy CNC khác nhau.
Từ khóa: Tuyến tính hóa đường chạy dao; Bộ hậu xử lý; Máy CNC 5 trục; Động học ngược; Lập trình CNC; CAD/CAM/CNC.

Abstract:
When cutter location (CL) data are generated by the CAM system, it is assumed that the tool path between two CL points is a straight line relative to the workpiece. However, this is only the case for the simultaneous motion in three linear axes, for the case of five-axis machine, due to the rotary axes, the tool path between two blocks in the NC program will be non-linear relative to the workpiece, reducing the accuracy of the tool path. This paper presents a tool path linearization algorithm and a postprocessor for five-axis CNC machine that allows to reduce the tool path error. A windows-based software module written in Visual Basic was developed according to the presented method. The five-axis CNC milling machine tool with two rotary axes on the table (Deckel Maho DMU 70 eVoluion) was used to demonstrate and validate the linearized CL data generated by the proposed method and postprocessor. The simulation result shows that the proposed method can reduce the error of the tool path; the postprocessor is reliable and can be customized to apply for any types of five-axis CNC machine configuration.
Keywords: Tool path linearization, Postprocessor, Five-axis CNC machine, Inverse Kinematics, NC Programming, CAD/CAM/CNC.

Abbreviations

CAM	Computer-Aided Manufacturing
CNC	Computer Numerical Control
CC	Cutter Contact
CL	Cutter Location
MCS	Machine Coordinate System
WCS	Workpiece Coordinate System

1. Introduction

A five-axis machine is similar to two cooperating robots, one robot carrying the workpiece and one robot carrying the tool. The five-axis machine is used for machining of complex sculptured surfaces because they have advantages of reducing setup time, faster material removal rate and improving surface finish. Programming of the five-axis machining needs an interface called postprocessor that converts the CL data to the machine control data [1].

The CL data generated by the CAM software contain the cutter locations as a sequence of tool vector positions. These points are calculated in such a way that the motion between two successive points is within a certain tolerance. This tolerance is normally specified just before the start of the tool path generation with the CAM system. The selection of this tolerance depends on the required accuracy. For roughing we can select a large value, for finishing we select a value corresponding to the required tolerance on the machined part. All current CAM systems calculate the CL points with the assumption that the motion between two consecutive CL points is linear. This is however only the case of simultaneous motion in three linear axis, if the motion is in five-axis (three linear and two rotational) simultaneous then the tool path in between two CL points is a curve [2].

Post-processing for the five-axis machine is complex due to the rotary axes. The problems are related to the linearization of the tool path, the solution for forward and inverse kinematics. There have been a lot of research work are reported on a postprocessor method for the five-axis machine. A five-axis postprocessor for the spindle-tilting type five-axis machine tool with a nutating head has developed by She and Chang [1]. Lee and She [5] presented an analytical methodology to develop a postprocessor for three typical 5-axis machine tools. Jung et al. [6] developed algorithms for NC-postprocessor for 5-axis milling machine of table-rotating/tilting type. However, none of above postprocessors has further investigated the tool path linearization problem. Takeuchi and Watanabe [7] presented principles for linearization. She and Huang [8] developed a postprocessor for two five-axis machines; the linearization problem was presented in the research. An algorithm for calculating the inverse kinematics of the five-axis machine close to singular configurations was presented by Knut [9], the linearization of the tool path was also discussed in the paper.

This paper focuses on the developing a five-axis postprocessor with the tool path linearization. The algorithm has been implemented in Visual Basic, demonstrated for the five-axis machine DMU 70 eVoluion. The result shows that the proposed method can reduce the error of the tool path.

The rest of the paper is organized as follows. Section 2 presents the tool path linearization algorithm. Section 3 presents the calculation of machine coordinates. The software implementation and verification is presented in Section 4, and the last Section is the conclusion.

2. Tool path linearization algorithm

In five-axis machining the real tool path between two CL points is not linear but a curve. The solid curved lines in Fig. 1 show the real CL and CC point path. This curved real tool path clearly introduces an error due to the five-axis linear interpolation. This error should be checked and compensated for, and this is done in the postprocessor [10].

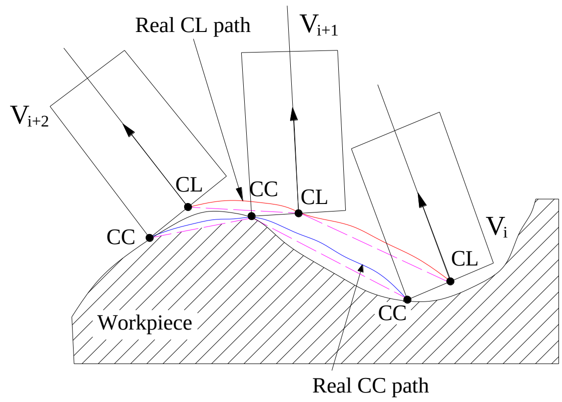

Fig. 1 *Real tool paths between two CL points [10]*

Linearization of the tool path is performed in the postprocessor by interpolating new CL data point along the ideal tool path and thereby adding new blocks to the NC program. One command in the CL file may result in several lines in the NC file, so the size of the NC program increase. The final tool path consists of positions that originate directly from the CL data file, and new positions that are generated by the CL data interpolation in the postprocessor. Linearization will not provide perfect motion of the tool, but the deviation of the ideal tool path can be reduced to an acceptable level [9].

If the real tool path deviates more than tolerance T from the piecewise linear CL path, the tool path will be linearized. The linearization problem is solved in the postprocessor by inserting intermediate points between two successive CL points, when the deviation is out of tolerance.

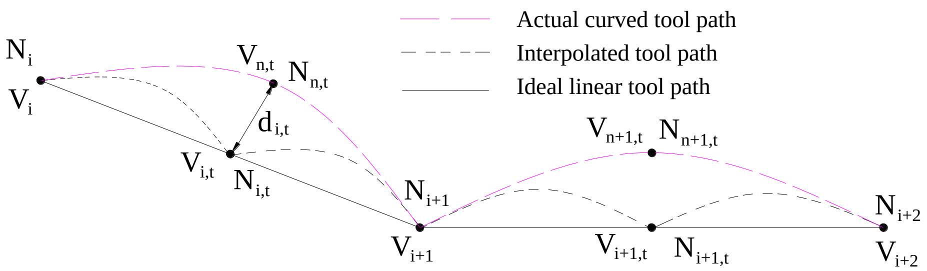

Fig. 2 *Tool path linearization*

We assume that V_i, V_{i+1} and V_{i+2} are three consecutive points in the CL data (Fig. 2). The corresponding NC program of V_i is $N_i = [X_i, Y_i, Z_i, B_i, C_i]$. Each axis is assumed to move linearly between the specified points. Therefore, each point in the actual curved path can be expressed as:

$$N_{n,t} = N_i + t(N_{i+1} - N_i) \qquad (1)$$

Where t is a dummy time coordinate ($0 \le t \le 1$).

The corresponding CL data $V_{i,t}$ for $N_{n,t}$ can be determined by the forward kinematics equations [3, 4]. Moreover, each point in the ideal linear tool path can be determined as follows:

$$V_{i,t} = V_i + t(V_{i+1} - V_i) \qquad (2)$$

It can be seen from the Fig. 2, the distance between $V_{n,t}$ and $V_{i,t}$ forms a deviation, denoted as $d_{i,t}$. If the maximum deviation $(d_{i,t})_{max}$ exceeds the prescribed tolerance, then the additional interpolated CL data $V_{i,t}$ should be inserted into the original CL data. Normally, the middle point, t = 0.5, is selected as the candidate point. After the intermediate point $V_{i,t}$ has been inserted, the corresponding NC code can be generated.

The tool path linearization algorithm is as follows:

1. Select t=0.5
2. Read V_i, V_{i+1} from the CL file
3. Calculate $V_{i,t} = V_i + t(V_{i+1} – V_i)$
4. Calculate N_n, N_{n+1} from V_i, and V_{i+1} based on inverse kinematics [3, 4]
5. Calculate $N_{n,t} = N_i + t(N_{i+1} – N_i)$
6. Calculate $V_{n,t}$ from $N_{n,t}$ based on forward kinematics
7. Calculate the distance $d_{i,t}$
8. Compare $d_{i,t}$ with tolerance T
   - If $d_{i,t} \le T$, then read next CL point, i = i+1
   - If $d_{i,t} > T$, insert $V_{i,t}$ in between V_i and V_{i+1} and repeat from step 2 to 8

The flowchart of the tool path linearization algorithm is shown in Fig. 3.

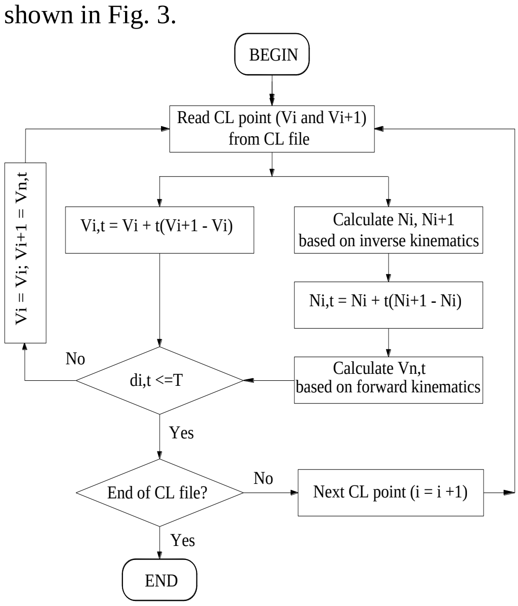

Fig. 3 *Flowchart of the tool path linearization algorithm*

3. Machine coordinates calculation

The CL data are the cutter location (x, y, z) and orientation (i, j, k) defined in the workpiece coordinate system. This, x, y, z, i, j, k, data must be transformed to the machine coordinates X, Y, Z, and (B, C) or (A, B) or (A, C) which control the motion of the machine axes. By the inverse kinematics transformation method, once the CL data are obtained, three linear joint motions (X, Y, Z) and two rotary joint motions (B, C) or (A, B) or (A, C) can be calculated. This can be done by the geometric transformation from the WCS to the MCS.

Fig. 4 shows the five-axis machine DMU 70 eVolution that used to demonstrate for the proposed method, Fig. 5 is the kinematics chain diagram of the machine. In order to calculate the machine coordinate (X, Y, Z, B, C) from the CL data (x, y, z, i, j, k) we have defined additional coordinate systems at some joints. These reference systems are defined in such a way that the transformation from the workpiece coordinates to the machine coordinates can be done in simple steps. These intermediate reference systems are shown in Fig. 6 and the transformation are calculated as follows:

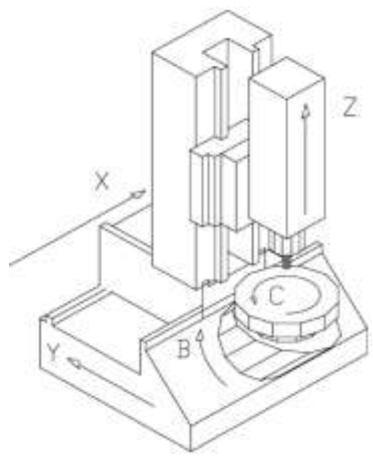

Fig. 4 *Model of the five-axis machine -DMU 70 eVolution*

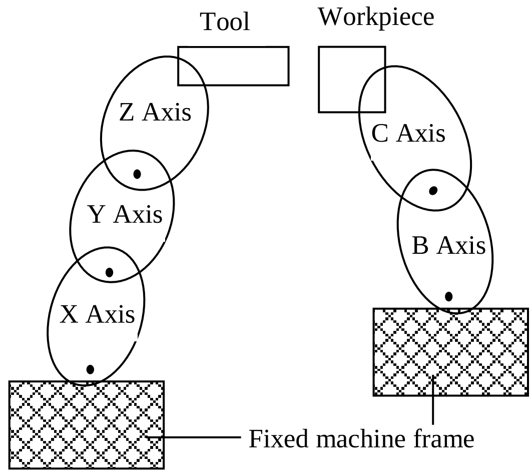

Fig. 5 *Kinematics chain diagram*

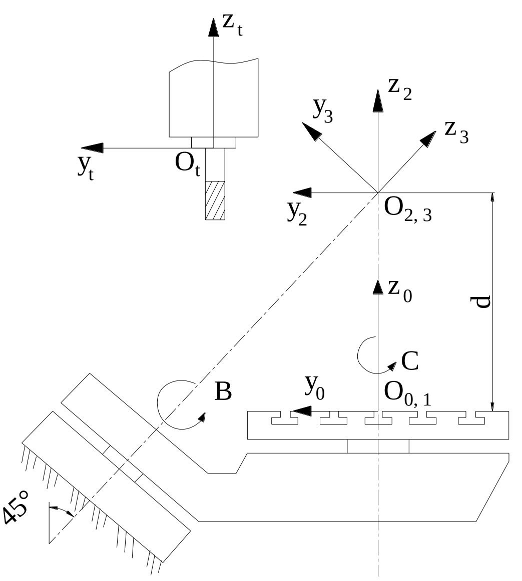

Fig. 6 *Intermediate reference coordinate systems*

O_0 ($x_0y_0z_0$): located at the center of the table surface C, when B = C = 0^0. z_0-axis coincides with the C-axis centerline.

O_1 ($x_1y_1z_1$): Obtained by rotating ($x_0y_0z_0$) around z_0 at an angle C.

O_2 ($x_2y_2z_2$): Obtained by translating ($x_1y_1z_1$) at a distance d along z_0

O_3 ($x_3y_3z_3$): Obtained by rotating ($x_2y_2z_2$) around x_2 at an angle $+45^0$

O_4 ($x_4y_4z_4$): Obtained by rotating ($x_3y_3z_3$) around z_3 at an angle B

O_5 ($x_5y_5z_5$): Obtained by rotating ($x_4y_4z_4$) around x_4 at an angle -45^0

O_w ($x_wy_wz_w$): Obtained by translating ($x_5y_5z_5$) at a distance -d along z_4

O_t ($x_ty_tz_t$): The machine coordinate system fixed to the tool spindle tip.

Step 1: Rotation around z_0 at an angle C

$$\begin{aligned} x_1 &= x_0\cos C - y_0\sin C \\ y_1 &= x_0\sin C + y_0\cos C \\ z_1 &= z_0 \end{aligned} \tag{3}$$

Step 2: Translation $O_1 \rightarrow O_2$ along z_0 axis at a distance d

$$\begin{aligned} x_2 &= x_1 + x_{o1o2} \\ y_2 &= y_1 + y_{o1o2} \quad \text{Where } z_{o1o2} = d \\ z_2 &= z_1 + z_{o1o2} \end{aligned} \tag{4}$$

Step 3: Rotation around x_2 at an angle $+45^0$

$$\begin{aligned} x_3 &= x_2 \\ y_3 &= y_2 cos45^0 - z_2 sin45^0 \\ z_3 &= y_2 sin45^0 + z_2 cos45^0 \end{aligned} \tag{5}$$

Step 4: Rotation at an angle B around z_3

$$\begin{aligned} x_4 &= x_3 cosB - y_3 sinB \\ y_4 &= x_3 sinB + y_3 cosB \\ z_4 &= z_3 \end{aligned} \tag{6}$$

Step 5: Rotation around x_4 at an angle -45^0

$$\begin{aligned} x_5 &= x_4 \\ y_5 &= y_4 cos45^0 + z_4 sin45^0 \\ z_5 &= y_4 sin45^0 + z_4 cos45^0 \end{aligned} \tag{7}$$

Step 6: Translation at a distance -d along z_5

$$\begin{aligned} x_w &= x_5 + x_{o5ow} \\ y_w &= y_5 + y_{o5ow} \quad \text{Where } z_{o5ow} = -d \\ z_w &= z_5 + z_{o5ow} \end{aligned} \tag{8}$$

x_w, y_w, z_w can be solved by the above equations (8) to (3). The solutions for X, Y, Z of the NC data are found by solving:

$$\begin{cases} X = x_t = x_w \\ Y = y_t = y_w \\ Z = z_t = z_{O_wO_t} - z_T + z_w \end{cases} \tag{9}$$

Where Z_T is the tool length

With $\cos B = 2k_0 - 1$, X, Y, Z can be expressed as:

$$X = \left[-y_0\sqrt{2\ k_0 - k_0^{\ 2}} - x_0 + 2x_0k_0 \right]\cos C + \left[x_0\sqrt{2\ k_0 - k_0^{\ 2}} + 2y_0k_0 - y_0 \right]\sin C + \ d - z_0\ \sqrt{2\ k_0 - k_0^{\ 2}} \tag{10}$$

$$Y = \left[x_0\sqrt{2\ k_0 - k_0^{\ 2}} + y_0k_0 \right]\cos C + \left[y_0\sqrt{2\ k_0 - k_0^{\ 2}} - x_0k_0 \right]\sin C - z_0 + d - dk_0 + z_0k_0 \tag{11}$$

$$Z = \left[x_0\sqrt{2\ k_0 - k_0^{\ 2}} + y_0k_0 - y_0 \right]\cos C + \left[y_0\sqrt{2\ k_0 - k_0^{\ 2}} - x_0k_0 + x_0 \right]\sin C + d - dk_0 + zk_0 \tag{12}$$

Where $x_0, y_0, z_0, i_0, j_0, k_0$ are the tool tip position and the tool orientation given in the CL data.

By the same method and noted that $\begin{cases} i_4 = 0 \\ j_4 = 0 \\ k_4 = 1 \end{cases}$, the solutions for C-axis and B-axis can be found:

$$C = \arctan\left[\ 1-k_0\ \ i_0 + \sqrt{2\ \ k_0 - k_0^{\ 2}}\ j_0,\ \ k_0 - 1\ \ j_0 + \sqrt{2\ \ k_0 - k_0^{\ 2}}\ i_0\ \right] \tag{13}$$

$$\cos B = 2k_0 - 1 \ \Rightarrow\ B = \arccos(2k_0 - 1) \tag{14}$$

Based on the above inverse kinematics transformation method the equations to generate NC data for other types of five-axis machine tools can also be determined.

4. Software implementation and verification

A window-based software module for tool path linearization has been developed in Visual Basic language. The user interface of the program is shown in Fig. 7.

To clearly show the efficiency of the tool path linearization algorithm we first test for the case of four CL points and plotted the tool path before and after linearization on the graph. Table 1 shows four CL data points for testing, and table 2 shows the CL data after linearization by the proposed algorithm. In the example we selected tolerance T = 0.1, the CL data after linearization has 11 points instead of four points; the bold lines in the table 2 are the inserted points. Fig. 8 is the user interface to input the testing CL data. Fig. 9 shows the graph of the tool path before and after linearization, the straight lines are the linear tool path, the curves are the actual NC tool path, the distance from the peak of the curve to the straight line is the deviation $d_{i,t}$.

Table 1. CL data before linearization

	x	**y**	**z**	**i**	**j**	**k**
Point 1	20	0	-155	0.6327447124	0.465569252	0.6187742456
Point 2	100	0	-150	0.628284235	0.461387437	0.62640867
Point 3	200	0	-146	0.623829809	0.457180279	0.634039219
Point 4	300	0	-139	0.618938519	0.452948814	0.6416794216

Table 2. CL data after linearization

	x	**y**	**z**	**i**	**j**	**k**
Point 1	20	0	-155	0.6327447124	0.465569252	0.6187742456
Point 11	**60**	**0**	**-152.5**	**0.63051447415**	**0.4634771812**	**0.06225885563**
Point 2	100	0	-150	0.628284235	0.461387437	0.62640867
Point 21	**125**	**0**	**-149**	**0.62713392215**	**0.460335647825**	**0.628311955**
Point 22	**150**	**0**	**-148**	**0.6259836048**	**0.45928385845**	**0.630221043**
Point 23	**175**	**0**	**-147**	**0.62483329465**	**0.458232069075**	**0.632130131**
Point 3	200	0	-146	0.623829809	0.457180279	0.634039219
Point 31	**255**	**0**	**-144**	**0.6224968656**	**0.45612241345**	**0.63594926965**
Point 32	**250**	**0**	**-142.5**	**0.6213107503**	**0.4550645472**	**0.6378593203**
Point 33	**275**	**0**	**140.75**	**0.620126435**	**0.45400668095**	**0.63976937095**
Point 4	300	0	-139	0.618938519	0.452948814	0.6416794216

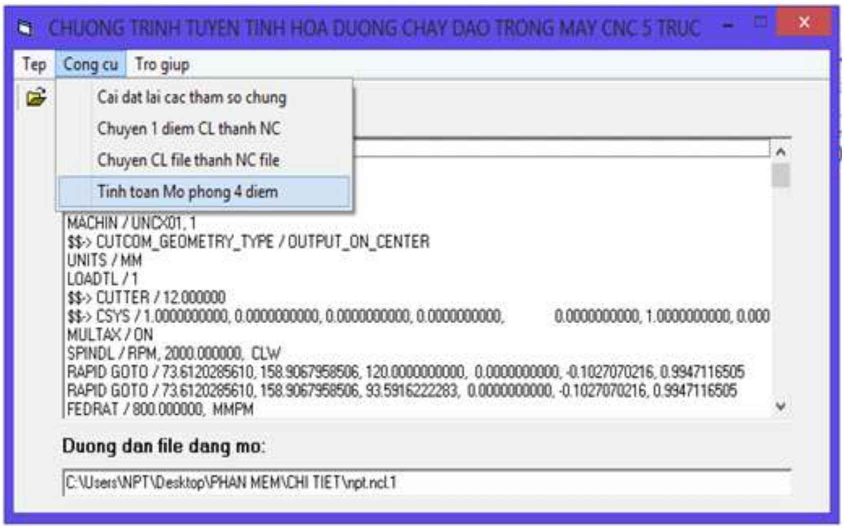

Fig. 7 *User interface of the tool path linearization program*

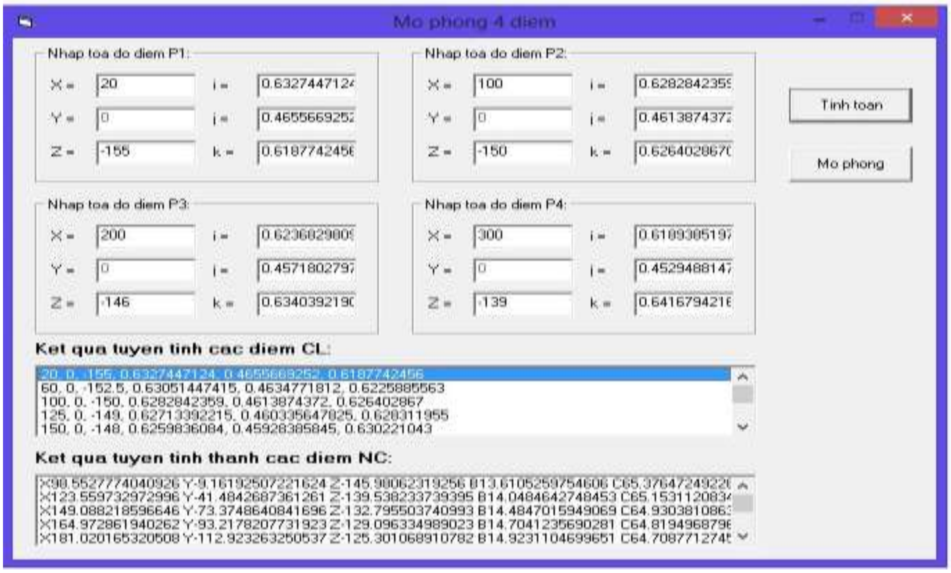

Fig. 8 *User interface for inputting 4 CL data points*

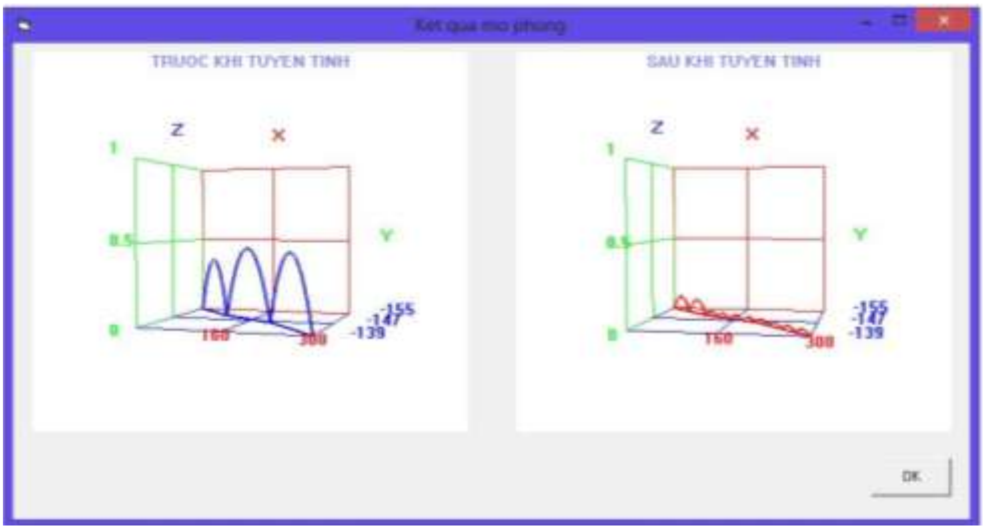

Fig. 9 *The graph of the CL tool path and NC tool path before and after tool path linearization*

The linearized CL data is inputted into the postprocessor. A window-based postprocessor for five-axis with two rotary axes on the table has been also developed in Visual Basic programming language. The user interface of the developed postprocessor is shown in Fig. 10. The linearized CL file is opened by selecting the "File\Open" menu or clicking the Open icon on the toolbar and the CL data is displayed in the left below window corner. The NC data are generated by selecting the "Run\Start" menu or clicking the Run icon on the toolbar and the NC data is displayed in the right below window corner. The generated NC file is saved by selecting "File\Save" menu or clicking the Save icon (Fig. 10).

To demonstrate and confirm the correctness of the NC data generated by the proposed tool path linearization and postprocessor, an air-compressor turbine blade (Fig. 11) was used. The CL data after linearization are inputted into the proposed postprocessor to generate NC data. The generated NC data are then verified by the VERICUT© software with the model of five-axis machine DMU 70 eVolution (Fig. 12).

The results of the cutting simulation of the CL data and generated NC data are the same (Fig. 13 and Fig. 14). This proved that the tool path of the NC data generated by postprocessor is correct.

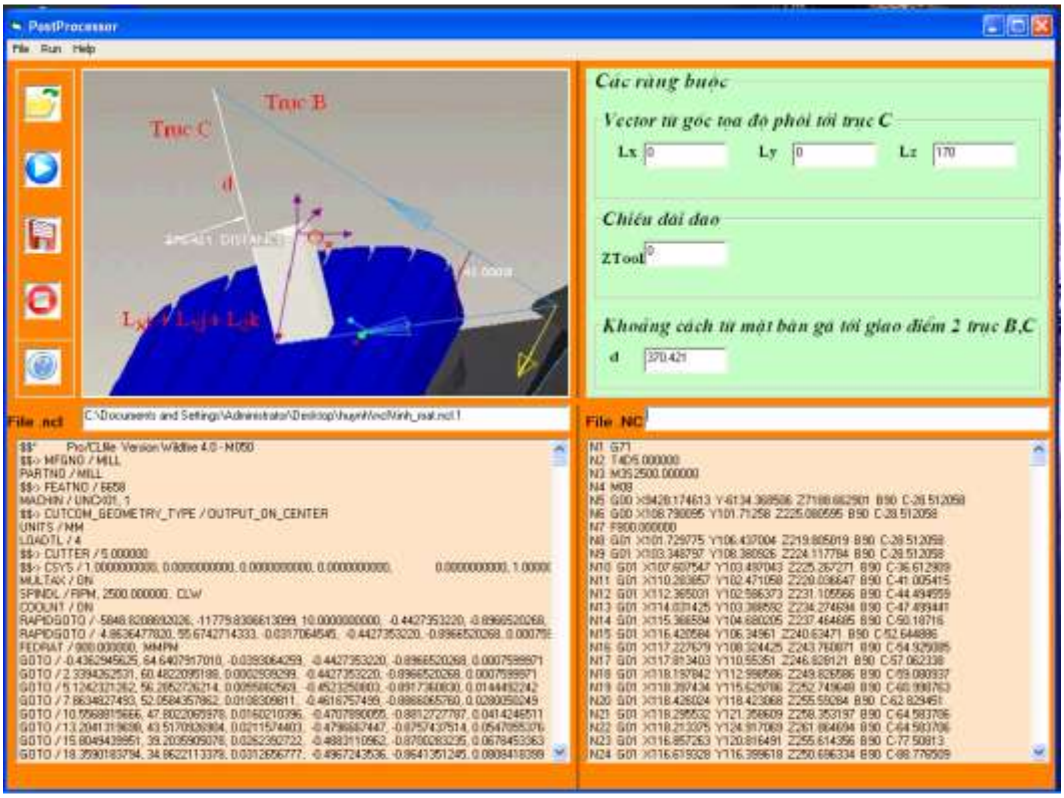

Fig. 10 *Interactive user interface of the proposed postprocessor*

Fig. 11 *Real air-compressor turbine blade and its 3D model*

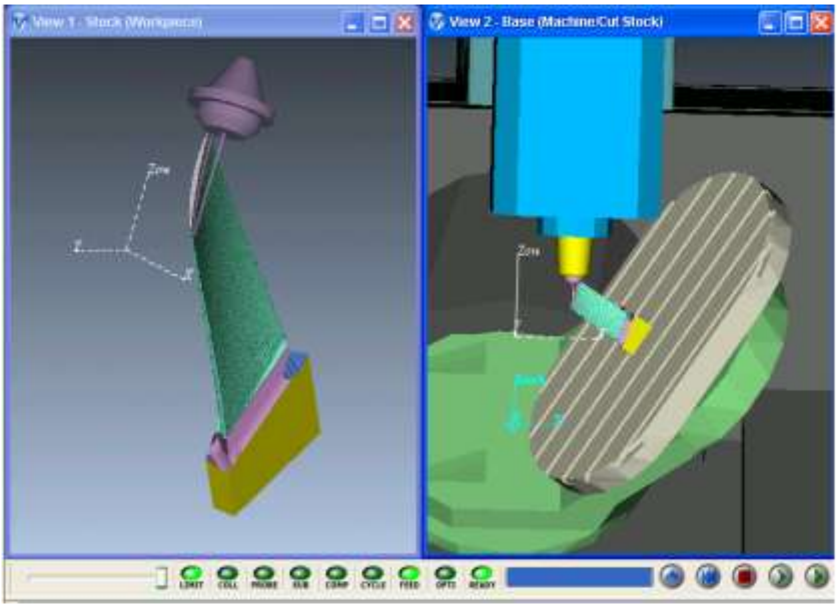

Fig. 12 *In progress solid cutting simulation of the air-compressor turbine blade in VERICUT® software*

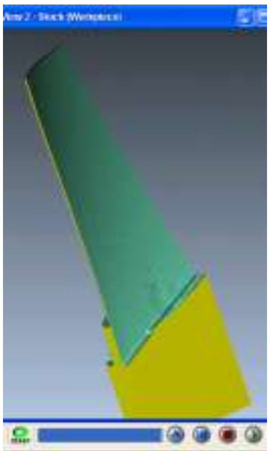

Fig. 13 *Cutting simulation by CL data*

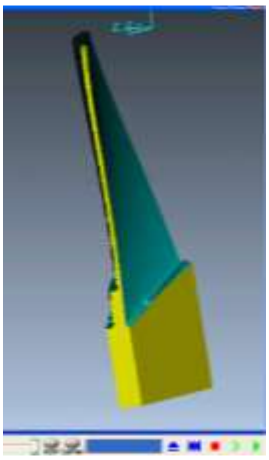

Fig. 14 *Cutting simulation by generated NC data*

5. Conclusion

In this paper, we have developed a five-axis postprocessor with the tool path linearization. The simple example of four CL points and the complex air-compressor turbine blade surface to be machined validated the efficiency of the proposed method. The proposed methodology allows to reduce the error of the tool path in five-axis CNC machine, the proposed postprocessor is reliable and can be customized to apply for any types of five-axis CNC machine.

References

[1] She, C.-H.; Chang, C.-C., *Development of a Five-Axis Post-processor System with A Nutating Head*, Journal of Materials Processing Technology 187-188, pp.60-64, 2007.

[2] Erik L. J. Bohez, *Computer control of manufacturing I*, Asian Institute of Technology, Bangkok, Thailand, 1995.

[3] Tran Duc Tang, *A five-axis postprocessor based on inverse kinematics transformation*, Advanced Material Research 662-663, pp.525-530, 2013.

[4] Tran Duc Tang, *Calculation of machine coordinates and postprocessor for 5-axis CNC milling machine*, Tuyển tập công trình Hội nghị Cơ điện tử toàn quốc lần thứ 6, tr350-356, 2012.

[5] Lee, R.-S. and She, S.-H., *Developing a Postprocessor for Three Types of Five-Axis Machine Tool*, International Journal of Advanced Manufacturing Technology 13, pp.658-665, 1997.

[6] Jung, Y.H.; Lee, D.W.; Kim, J.S.; Mok, H.S., *NC Postprocessor for 5-axis milling machine of table-rotating/tilting type*, Journal of Materials Processing Technology 130-131, pp.641-646, 2002.

[7] Yoshimi Takeuchi, Takahiro Watanbe, *Generation of 5-axis control collision-free tool path and postprocessor for NC data*, Annual of the CIRP 41 (1), pp.539-542, 1992.
[8] She, C.-H.; Huang, Z.-T., *Postprocessor Development of A Five-Axis Machine Tool with Nutating Head and Table Configuration*, International Journal of Advanced Manufacturing Technology 38, pp.728-740, 2008.
[9] Knut Sorby, *Inverse Kinematics of Five-Axis Machines Near Singular Configurations*, International Journal of Machine Tools & Manufacture 47, pp.299-306, 2007.
[10] Erik L.J. Bohez, *Compensating system errors in 5-axis NC machining*, Computer-Aided Design 34, pp.391-403, 2002.

Tran Duc Tang is an Associate Professor and Deputy Dean of Aerospace Engineering Faculty, Le Quy Don Technical University, Hanoi, Vietnam. He received his PhD and M.Eng in Design and Manufacturing Engineering from Asian Institute of Technology (AIT) - Thailand in 2007 and 2002, respectively. He is teaching the courses of Advanced Manufacturing Processes, CAD/CAM/CAE, Flexible Manufacturing Systems, Multi-axis machine tools. His research interests are five-axis NC machining, CAD/CAM/ CAE/CNC, Modeling of FMS by Petri Net, Reverse engineering, Rapid prototyping, NC simulation & programming, and Computer Graphics. His email address is: tangtd@mta.edu.vn or tangtd@gmail.com

Nguyen Phu Thuy is a student at Le Quy Don Technical University, Hanoi, Vietnam. His research interests are CAD/CAM/CNC, five-axis NC machining, NC simulation & programming. His email address is phuthuynguyen1702@gmail.com

Vuong Si Kong is currently a PhD student at Le Quy Don Technical University, Hanoi, Vietnam. He received his M.Eng and B.Eng in Mechanical Engineering from Hanoi University of Science and Technology, Vietnam in 2007 and 2003, respectively. Mr. Kong is also a lecturer in the Engineering Faculty, Hung Yen University of Technical and Education. His research interests are five-axis NC machining, CAD/CAM/CAE/CNC. His email address is: vuongsikong@gmail.com