Interactive system to control the kinematic error of a five-axis milling machine

Supatpong Narmnark\textsuperscript{1,a}, Stanislav S. Makhanov\textsuperscript{2,b}
\textsuperscript{1,2}School of Information, Computer and Communication Technology, Sirindhorn International Institute of Technology, Thammasat University, Pathumthani 12121, Thailand
\textsuperscript{a}s.narmmark1993@gmail.com, \textsuperscript{b}Makhanov@siit.tu.ac.th

Abstract. The kinematic error (KE) of a five-axis milling machine is an important measure of the accuracy. The proposed software allows to interactively position the cutter contact (CC) points to reduce the KE. The system helps to optimize the tool path by a trial and error approach and provides a learning experience for the users of five-axis machines. The code works with the STL and OFF files as well as with parametric surfaces. The Interactive system (IS) was offered to unexperienced users as a game with a goal to reduce visible kinematic errors while keeping the number of available CC points. The users were able to achieve a decrease of of the maximum KE by 91.73\% and of the average KE by 43.70\%. The program has been verified with a post processor from HAAS VF-2TR machining center (Trunnion VMCs).

1. Introduction

Five-axis milling machines (Fig 1.) become popular and common in manufacturing, bio-medical, automobile and mold industries [1], [2]. Compared to three-axis milling machines, the two additional rotary axes provide higher accuracy and improve machinability of complex shaped parts [3].

The five-axis geometric errors can be attributed to three main types. The first type is related to the kinematic of the machine and includes the systematic errors and errors attributed to the initial set-up. The machines with three linear axes have a total of 21 linear independent geometric error components whereas the five-axis milling machine has 42 components (twice of the three-axis machine!) [4]. The second source is the imperfection of the movements the machine components and its fixtures. Finally, an important source of the geometric errors is an inaccurate tool positioning which leads to a curvature interference and gouging (see, for instance, survey [5]). In many cases, this type of error is unavoidable but can be considerably reduced.

Some other, less prominent but still important, sources of errors are: machine operating conditions such as the material removal rate (feed rate), the depth of the cut, wet or dry cutting, clamping conditions, the tool wear and other tool imperfections. In this paper, we offer a software to analyze and reduce the KE of the five-axis machine with the understanding that the above-mentioned sources of errors exist. Under certain conditions, some of the errors could actually exceed the KEs. However, our numerical experiments and the actual machining show that these errors are important and could lead to significant inaccuracies (even collusions) in the five-axis mode [6]. At the moment there exists a variety of tools to generate a tool path and simulate the five-axis machines such as MasterCam [7], NX11[8] and SolidWorks[9]. However, up to now an IS that allows to interactively change the tool...
path has not been developed. In the future the system can be integrated with a knowledge database which combines the experience of the machinist, operator and G-code programmers.

Figure 1. Five-axis milling machine (HAAS VF-2TR)

2. Methodology
Our five-axis machining operation includes the following steps.
1. Input data (Fig. 2a): part surface (STL, OFF, etc.), tool type, tool size, required machining accuracy $\delta_{\text{max}}$.
2. Flatten the surface using a MATLAB library [10] and detect the corresponding parametric domain (Fig. 2b).
3. Establish a minimum bounding rectangle (MBR) and map it onto $\Delta = [0,1] \times [0,1]$ . Note that at this point the surface has been parameterized i.e. $S = S(u,v) = (x(u,v), y(u,v), z(u,v))$, where $(u, v) \in \Delta$ are the parametric coordinates.
4. Create a tool path on $\Delta$. In this paper we show the system applied to correct a zigzag tool path [11]. However, it is applicable to a spiral and a radial tool path as well.
5. Use the barycentric interpolation to find the tool path (CC points) on the STL part surface.
6. Find the normals (Fig. 2c) and the curvature at the CC points.
7. Detect the curvature interference. Find an appropriate inclination of the tool for the flat end cutter. In case of a ball nose tool replace it by a tool with a smaller radius.
8. Create an interference free tool path and check the scallop height $h$ [12] between the tracks. If the system shows a violation of the scallop constraint $h < \delta_{\text{max}}$, insert interactively new paths between the corresponding tracks.
9. Use a specific postprocessor (in our case HAAS-2TR) evaluate the KE. The system shows the violations of the KE in the 2D parametric domain and the actual tool trajectories on the 3D part surface.
10. The user interactively changes the positions and the number of the CC points along the tool path in order to reduce the KE.

11. Generate a G-code and feed it into a solid modelling simulation software such as Vericut or NX - 11 in order to eliminate global collisions.
12. The solid model is verified visually and compared with the design surface.
13. The G code is verified on a real machine using a soft material.
14. The resulting part is measured against the design surface.
15. The real industrial part is produced and verified.

The proposed IS to control the KE is used by steps 8-10, whereas steps 1-7 can be interpreted as preparation of the data and steps 11-14 as post processing steps.

The software works with STL or OFF surfaces. Alternatively, the user inputs parametric equations or polynomial approximations such as the Bezier patch or the Gregory patch. The STL/OFF surface is
flattened and assigns an initial equal spaced CC points on a zigzag toolpath (Fig. 2d). The user manipulates the positions of the CC points interactively with in order to achieve a better accuracy.

3. Interactive System to control the KE
The software is characterized by a minimalistic design and simplicity. The users who are not familiar with the intricate theory of five axis machining should be able to intuitively generate a reasonable tool path, verify it visually and perform a solid modeling. Our simple menu and the user screen are depicted in Fig. 3.
The IS provides a heuristic adaptation and automatic point insertion proposed in [13],[14]. The heuristic adaptation includes four functions as follows:

1. Move: the user moves the CC points in the parametric space \((u,v)\) along the \(u\)–axis (see Fig. 3) in order to re-distribute the points and reduce the KE without increasing their number.
2. Add: the user adds CC points to the toolpath. The function reduces the KE but also increases the operation time in the case of high speed machining.
3. Delete: the user deletes CC points in order to reduce the machining time.
4. Select: the user selects the CC points in order to perform operations 1–3 on a set of points.

The system evaluates the KE as follows: let 
\[ W^D(s_p,s_{p+1},t) \in S(u,v) \]
be a space curve between two tool positions \(W_p\) and \(W_{p+1}\) extracted from the machined surface \(S(u,v)\), where \(t \in [s_p,s_{p+1}]\) is a parametric coordinate along the curve. The total KE is the Hausdorff distance \(\text{dist}_H\) between the desired trajectories \(W^D_{p,p+1}(t)=W^D(s_p,s_{p+1},t)\) and the actual trajectories \(W^r_{p,p+1}(t)=W(s_p,s_{p+1},t)\) defined by
\[
\varepsilon = \sum_t \text{dist}_H(W^D_{p,p+1}(t),W^r_{p,p+1}(t)).
\]

The average KE is then defined by
\[
\varepsilon_H = \frac{\varepsilon}{N_{CC}-1},
\]
where \(N_{CC}\) is the number of the CC points. In many cases, the machining quality is defined by the maximum error given by
\[
\varepsilon^H = \max_t (\text{dist}_H(W^D_{p,p+1}(t),W^r_{p,p+1}(t))).
\]

The trajectories \(W^r_{p,p+1}(t)\) are obtained by the corresponding inverse kinematics transformations[1].

An interactive classification the prescribed the levels of the KE indicated by color (see TABLE I). The experienced user can modify the levels of the KE for a particular part surface.

### TABLE I. ERROR CLASSIFICATION TABLE

| Error level | Range      | Description                                      |
|-------------|------------|--------------------------------------------------|
| Blue        | 0 to <0.1  | The error is acceptable                          |
| Green       | 0.1 to <0.05 | The error is very small, delete a considerable # of the CC points or move them out of that area. |
| Violet      | 0.05 to <0.3 | The error is too large, insert a few CC points or move CC points. |
| Red         | ≥0.3       | The error is huge, insert considerable # of the CC points |

#### 4. Numerical example

As an example consider a manual adaptation of the KE for an STL surface displayed in Fig. 2. Five operators (Bachelor students of SIIT) were assigned to reduce the KE using the proposed IS. The students did not have any knowledge about five-axis machining. The system was treated as a game with a goal to reduce the visible KE using the available operations. The error classification TABLE I was set up by the administrator. The students were instructed to use the proposed menu to reduce the kinematic errors \(\varepsilon^H\), \(\varepsilon_{av}^H\) while keeping or even reducing the number of CC points. If the student was able to reduce the error by 30% without increasing the number of CC points more than by 20%, the win was registered and the gaming time was recorded in Table II.
TABLE II shows results of the experiment, the KE after the manual adaptation has been reduced up to 91.73% of ε_{H} and 43.70% of ε_{av}.

The error reduction of ε_{H} and ε_{av} were evaluated by \( \varepsilon_{\text{reduction}} = \frac{\varepsilon_{\text{old}} - \varepsilon_{\text{new}}}{\varepsilon_{\text{old}}} \times 100\% \)

| Operators | Operating time | The reduction of ε_{H}, % | The reduction of ε_{av}, % | CC points moved | CC points inserted | CC points deleted | Total # of CC points |
|-----------|----------------|-------------------------|-----------------------|----------------|-----------------|-----------------|---------------------|
| Operator 1 | 7 min.         | 36.70%                  | 16.86%                | 120            | 0               | 0               | 400                 |
| Operator 2 | 8 min.         | 33.20%                  | 12.54%                | 85             | 15              | 15              | 400                 |
| Operator 3 | 12 min.        | 74.75%                  | 29.34%                | 110            | 29              | 10              | 419                 |
| Operator 4 | 13 min.        | 86.19%                  | 35.73%                | 150            | 38              | 0               | 438                 |
| Operator 5 | 10 min.        | 91.73%                  | 43.70%                | 120            | 76              | 0               | 476                 |

As an example consider a manual adaptation of the CC points for a surface displayed in Fig. 4 was created by operator 1 and operator 5

![The original tool trajectories](image1.png)  
(a) The original tool trajectories  
(b) The tool trajectories by operator 1  
(c) The tool trajectories by operator 5  
Figure 4a-c. The tool trajectories

5. Conclusion

The preliminary experiments show that the proposed system allows an unexperienced user to substantially improve the quality of the tool path. For approximately ten minutes of an interactive adjustment, the users were able reduce ε_{H} by 33.20-91.73% and ε_{av} by 12.54-43.70%.

As for the future work, the real cutting experiment would provide further verification on the IS.

6. Acknowledgment

This research is supported by the Center of Excellence in Biomedical Engineering of Thammasat University.

7. References

[1] W. Anotaipaiboon and S. Makhanov, "Minimization of the kinematic error for five-axis machining", Computer-Aided Design, vol. 43, no. 12, pp. 1740-1757, 2011.
[2] J. Li, H. Zhouyang and Y. Lou, "Tool path optimization in postprocessor of five-axis machine tools", The International Journal of Advanced Manufacturing Technology, vol. 68, no. 9-12, pp. 2683-2691, 2013.
[3] Y. Lu, Q. Bi and L. Zhu, "Five-axis Flank Milling Tool Path Generation with Smooth Rotary Motions", *Procedia CIRP*, vol. 56, pp. 161-166, 2016.

[4] E. Bohez, B. Ariyajunya, C. Sinlapreecheewa, T. Shein, D. Lap and G. Belforte, "Systematic geometric rigid body error identification of 5-axis milling machines", *Computer-Aided Design*, vol. 39, no. 4, pp. 229-244, 2007.

[5] S. Makhanov, "Adaptable geometric patterns for five-axis machining: a survey", The International Journal of Advanced Manufacturing Technology, vol. 47, no. 9-12, pp. 1167-1208, 2009.

[6] S. Ivanenko, S. Makhanov and M. Munlin, "New numerical algorithms to optimize cutting operations of a five-axis milling machine", *Applied Numerical Mathematics*, vol. 49, no. 3-4, pp. 395-413, 2004.

[7] Information on https://www.mastercam.com/en-us/

[8] S. Samuel and A. Ericksen, *Basic to advanced computer aided design using NX11 modeling, drafting, assemblies, and sheet metal*. Design Visionaries, 2017.

[9] D. Planchard, *SolidWorks 2018 reference guide*. Sdc Pubns, 2017.

[10] Information on https://www.mathworks.com/products/matlab.html

[11] S. Yuwen, G. Dongming, J. zhenyuan and W. Haixia, "Iso-parametric tool path generation from triangular meshes for free-form surface machining", The International Journal of Advanced Manufacturing Technology, vol. 28, no. 7-8, pp. 721-726, 2005.

[12] C. Tournier and E. Duc, "Iso-scallop tool path generation in 5-axis milling", The International Journal of Advanced Manufacturing Technology, vol. 25, no. 9-10, pp. 867-875, 2004.

[13] M. Munlin, “Tool Path Optimization for Five-Axis Machining”, *Journal of Information Science and Technology*, vol. 1, pp. 63-73, 2010.

[14] M. Munlin and S. Makhanov, "Iterative Tool Path Optimization for Five-Axis Machines with Optimal Point Insertion", *Computer-Aided Design and Applications*, vol. 9, no. 1, pp. 33-41, 2012.