ACTA TECHNICA JAURINENSIS

Vol. 17, No. 2, pp. 75-83, 2024

10.14513/actatechjaur.00736

Research Article

Tool path planning of ball-end milling of free-form surfaces as a search algorithm

Abdulwahab Mgherony[1], Balázs Mikó[1,*]

[1]Institute of Mechanical engineering and Technology, Óbuda University
Népszínház u. 8, 1081 Budapest, Hungary
*e-mail: miko.balazs@bgk.uni-obuda.hu

Submitted: 31/01/2024 Accepted: 12/03/2024 Published online: 14/03/2024

Abstract: This paper introduces an innovative approach for generating three-axis CNC tool paths for machining free-form surfaces. The method is designed to minimize variations in the effective tool diameter, addressing a common challenge encountered when using ball-end tools for machining free-form surfaces. These surfaces exhibit varying inclinations, leading to fluctuations in the tool's working diameter from one point to another, resulting in inconsistent cutting speeds and milling parameters despite a constant spindle speed. Consequently, the machined surface tends to lack uniformity. In contrast to conventional tool path planning techniques, the proposed method calculates the working diameter at each adjacent point and guides the tool's movement towards the point where the smallest change in working diameter is anticipated. This approach reduces fluctuations in cutting speed and promotes the generation of a more homogeneous surface.

Keywords: Working diameter; Ball-end tool; Free-form surface; Surface roughness; Three-axis milling

I. INTRODUCTION

The productivity of the milling process of the free form surfaces is determined by the machining circumstances, like the cutting tool properties, the cutting parameters, and the applied tool path strategy. The parameters of the milling process affect the micro- and macro accuracy of the machined surface. In order to improve the machining performances, there are several ways to optimize the technology and reduce the cost of the production.

The term "tool path" denotes the specific trajectory along which a machine's cutting tool moves to shape the desired surfaces [1]. Typically, Computer-Aided Manufacturing (CAM) systems are employed to generate this tool path for guiding CNC machines. This process gains even greater significance in the context of machining free-form surfaces, which have become increasingly prevalent across various industries like die and mould manufacturing, aerospace, and automotive production. Consequently, the automated generation of tool paths has assumed paramount importance since surface quality and machining efficiency hinge on the chosen tool path [2]. Various tool path planning strategies exist, including the iso-parametric line method, iso-scallop method, and iso-metric section plane method, among others [3]. However, these methods often lack clear guidance for selecting the optimal strategy. Thus, users are compelled to rely on a trial-and-error approach, which not only consumes time but is also suboptimal and prone to errors [4].

Hence, the paramount importance lies in devising an efficient and adaptable intelligent method for the individualized generation of tool paths customized to each specific surface. Nevertheless, the development of such a system presents a formidable challenge owing to the multifaceted variables influencing the machining of free-form surfaces, encompassing factors such as surface inclination, cutting speed, and various cutting parameters. This underscores the imperative for an advanced tool path planning approach capable of effectively addressing these intricate parameters to enhance surface quality and machining efficiency.

Many studies in this field primarily focus on optimizing cutting conditions through suitable milling strategies. While these approaches work well for machining numerous parts, they face challenges in achieving satisfactory surface quality on complex surfaces. For instance, J. Varga et al. [5] conducted a comparative study on four milling strategies using a ball-end tool and found that the constant Z strategy minimized shape deviations, resulting in an un-oriented surface. D.-D. Vu et al. [6] optimized tool paths for sculpted surfaces during three-axis end milling, reducing tool path length by 20% compared to conventional methods. G. Huo et al. [7] introduced an innovative approach for generating tool paths for free-form surfaces using a three-axis machine, ensuring alignment with desired feed directions across the entire surface.

Additionally, A. Mali et al. [8] explored the influence of cutting parameters and tool-path strategies on tool wear, cutting forces, and surface quality when milling curved surfaces, emphasizing the importance of factors like cutter diameter, and cutting speed.

In a different approach, A. Kukreja and S. S. Pande [9] developed a machine learning system that selects the best toolpath planning strategy for CNC machining complex surfaces using performance parameters and a Convolutional Neural Network (CNN), achieving 96.8% accuracy. Simultaneously, U. Župerl et al. [10] presented a cloud-based system for real-time tool condition monitoring during end milling, employing IoT, an optical system, and AI to detect cutting chip size and analyse cutting force trends, achieving 85.3% accuracy in identifying tool breakage.

Furthermore, J. Zhang et al. [11] developed an optimization model to plan tool paths, aiming to minimize path length while considering tolerance and complete milling constraints. In the study by Kukreja and S. Pande, [12], an optimized tool path was introduced utilizing the iso-scallop approach. Furthermore, the algorithm in the research offers two distinct strategies: iso-scallop and hybrid iso-scallop. This methodology involves the stitching and refinement of overlapping toolpaths. Meanwhile, in the study by Liu et al. [13] a highly efficient method for generating iso-scallop tool paths was presented. This method directly computes scallop points and cutter location (CL) points for iso-scallop paths from scattered data points using iterative algorithms, eliminating the need for point offsetting surface fitting. Remarkably, this new approach has showcased superior efficiency when compared to their previous method [14]. These distinct researches endeavours collectively contribute to advancing tool path optimization and machining efficiency.

Tool path optimization has several aspects, and effects. These can be the reduction of the tool load, improvement of the surface roughness, avoidance of the burr or incensement of the dynamics of the feed.

The aim of our research is to optimize the milling tool path of the ball end milling based on the calculation of the working (or effective, D_{eff}) diameter. In the case of free form milling, the working diameter is changed because of the changing of the surface inclination. The changing working diameter has an effect on the cutting speed and the surface roughness. During the previous research stage, these effects were investigated, and an optimization algorithm was developed, which modified the spindle speed and the feed in order to compensate the negative effect of the working diameter. The algorithm requires the tool position data and the surface inclination. The APT file format is used for tool path and the STL format for surface description. The modified NC code contains the variable cutting parameters. The acceleration and the breaking of the spindle depending on the surface inclination, and the dynamic load of the spindle, can be high [15].

In the current article, a new concept is presented, when the tool path is modified based on the changing of the working diameter of the ball-end tool.

II. The concept

The algorithm has been developed with a specific focus: generating an optimal tool path for milling free-form surfaces. The primary objective is to minimize variations in the working diameter of the ball-end tool during the machining process. By achieving minimal changes in the working diameter, the resulting reduction in cutting speed fluctuations ensures a more uniform and homogeneous machined surface.

The core objective of the algorithm is to perform path planning for CNC machining. This involves determining the sequence of points (toolpath) that the CNC machine should follow while milling the freeform surfaces. The developed algorithm solves the tool path re-planning as a search algorithm. The algorithm uses the pre-generated NC code in APT form. The code contains the points of the toolpath. The search algorithm reorder the points in order to equalize the value of the working diameter, and reduce the dynamic load of the spindle. The algorithm implements a path planning that takes into account the difference in the effective diameter between the current point and its neighbour points. Once a point is processed, it is marked as visited (tabu) to prevent the algorithm from revisiting it. The overarching goal is to ensure that the algorithm systematically covers all points without duplication while minimizing fluctuations in the working diameter.

The algorithm begins by reading and processing data from an STL file, which represents 3D models using triangular facets and contains vital geometry information. Subsequently, it extracts tool position coordinates from an APT file, a format used in CNC machining. To enhance precision, the code generates additional tool positions as needed and calculates normal vectors at each position based on facet data from the STL file. It also determines neighbour points and computes the working diameter using an established mathematical model. Finally, the code's core objective is path planning for CNC machining, aiming to optimize toolpaths by considering variations in the effective diameter while efficiently covering all points on the surface.

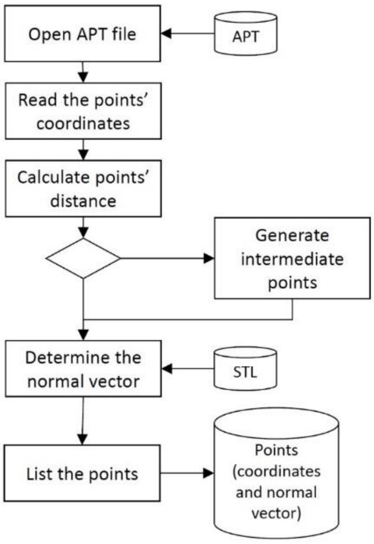


Figure 1. *Pre-processing of the tool position and surface data.*

Fig. 1 shows the pre-processing of the tool path and the surface data. The algorithm can be summarized in the following sequential steps.

The developed algorithm uses APT file to determine the tool position coordinates at each point of the surface. An APT (Automatically Programmed Tool) file is a file format used in computer-aided manufacturing (CAM) and computer numerical control (CNC) machining. APT is a high-level programming language specifically designed for defining toolpaths and machining operations for CNC machines. By reading the file, the algorithm extracts tool position coordinates.

The original CNC code contains the points of the tool path. The distance of the consecutive points depends on the surface inclination. In the case of plane surface segments, which can be horizontal or inclined, the distances can be large. But where the curvature of the surface is greater, the distance between the points is smaller. In order to reconfigure the tool path, quasi-equal density of the points is required. If the distance of points is larger than the defined limit (d_{lim}), new points are added. The suggested limit is the value of the width of cut parameter (a_e), which is the distance between two parallel paths. The procedure divides the line between two points into pieces, where the distance is smaller than the defined limit. It has no effect on the accuracy of the machining because the original tool path follows the same linear segment.

The distance between two points is:

$$d_i = \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2 + (z_i - z_{i+1})^2} \quad (1)$$

$$I = \text{Roundup}\left(\frac{d_i}{d_{lim}}\right) \quad (2)$$

If $I > 1$, additional points must be added, the number of new points is (I-1). The coordinates of the new points can be calculated by the next equation:

$$\underline{P}_{new_j} = \underline{P}_i + \frac{j}{I} \cdot (\underline{P}_{i+1} - \underline{P}_i) = \begin{bmatrix} x_i + \frac{j}{I} \cdot (x_{i+1} - x_i) \\ y_i + \frac{j}{I} \cdot (y_{i+1} - y_i) \\ z_i + \frac{j}{I} \cdot (z_{i+1} - z_i) \end{bmatrix};$$

$$j = 1 \ldots (I - 1) \quad (3)$$

The algorithm proceeds by processing data from the STL file, which is commonly used for representing 3D models with triangular facets. STL file contains information that characterizes the surface's geometry. This file format presents the surface as a collection of triangles along with their associated normal vectors within a three-dimensional coordinate system. It's important to note that the STL file does not contain any additional surface details, such as colour or texture information. As the algorithm reads the STL file, it accumulates a list of the normal vectors, and calculates the in-center point of each triangle (**Fig. 2**). The resolution of the STL file can be adjusted. The STL file describes the free form surface with some deviation, but this has no effect on the accuracy of the method, because only the normal vector of the machined area is required, and in case of technical free form surfaces, the changing is limited.

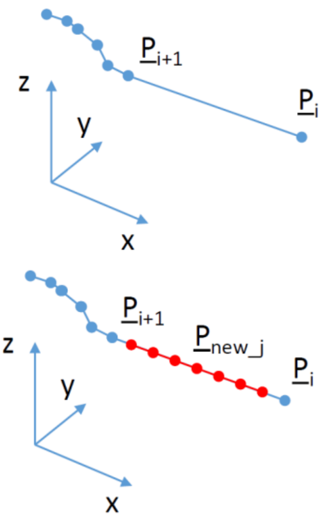


Figure 2. *Intermediate points.*

The algorithm calculates the normal vectors at each tool position using the facet normal obtained from the STL file. These normal vectors are associated with the corresponding tool positions by identifying the nearest in-center point. Essentially, this process ensures that at each position, the tool aligns with a specific triangle and adopts the same normal vector as that triangle.

The search algorithm of the reordering the points of the tool path is started by the selection of the starting point (**Fig. 3**). It has to be on the border of the machined surface. The point can be defined by rules, like the lowest or the highest point of the contour or it can be a random point.

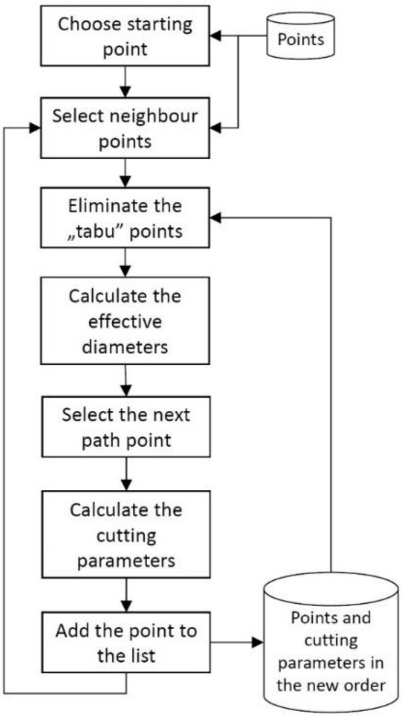


Figure 3. *Search algorithm.*

The next step is finding the neighbour points. Because of the non-uniform and non-regular distribution of the tool points, an adaptive method has to be used. This code then performs some neighbour point calculations based on a distance threshold R. It iterates over tool position elements and checks for nearby points. If the count of nearby points is less than a specific number, it incrementally increases the distance threshold R until it reaches or exceeds specific number of points. The resulting points are stored in a list. If a point was chosen previously, it is deleted from the neighbour set, as a tabu point.

The algorithm calculates the milling direction for each tool position based on the slopes between the tool position and its neighbour points. Then it calculates the working diameter of each neighbour point using a mathematical model presented by Mikó and Zentay [16].

At the last step, the point is selected, which has the smallest difference value in the working diameter. If there is no difference in the working diameter, because the surface is horizontal, the next point is selected in the feed direction. The modified cutting parameters (cutting speed and feed) are calculated, and add to the list of the points of the new tool path. The algorithm starts again by the selection of the neighbour points. The reordering of the points of the tool path is ended, when all points are selected.

III. Results

The algorithm was tested on a free form surface (**Fig. 4**), the size of the part was 50 x 50 mm, the height of the profile was 10 mm. The machining parameter in the simulation is shown in **Table 1**.

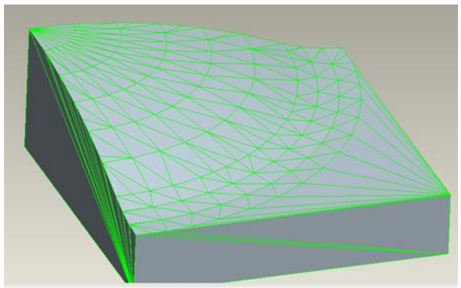
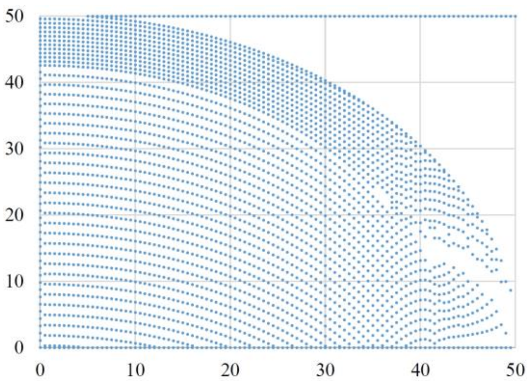


Figure 4. *Test part geometry.*

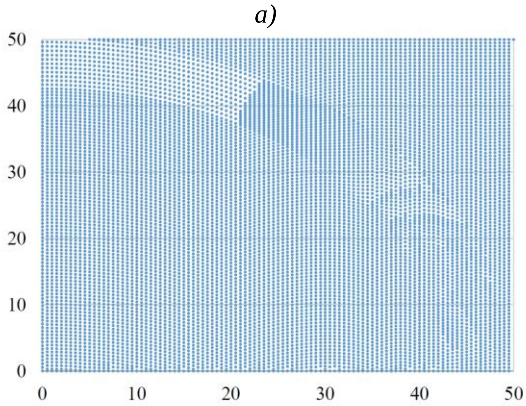
Table 1. *Parameters of the simulation.*

Parameter	Value
Tool diameter (D)	10 mm
Cutting speed (v_c)	63 m/min
Feed (v_f)	500 mm/min
Depth of cut (a_p)	0.3 mm
Width of cut (a_e)	0.5 mm
Feed direction (A_f)	90°

Fig. 5 illustrates the outcome of point processing. The first image displays the original distribution of path points, while the second reveals the extended point cloud. This extension of the point cloud provides valuable data for optimizing tool paths. Algorithms can leverage this data to determine the most efficient and safe routes for the cutting tool. To maintain relatively uniform point spacing, the width of cut (a_e) was used as a standard distance between points. This approach ensures reasonably consistent inter-point distances. Moreover, extending the original point cloud allows for a faithful representation of the complex free-form surface geometry, thereby enhancing the precision of tool path planning and reducing the risk of errors or defects in the final product.



a)



b)

Figure 5. *(a) Original and (b) extended distribution of the points.*

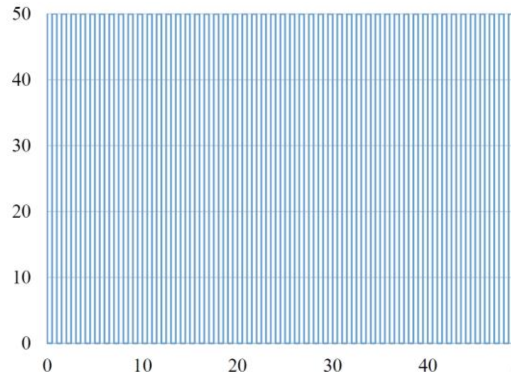
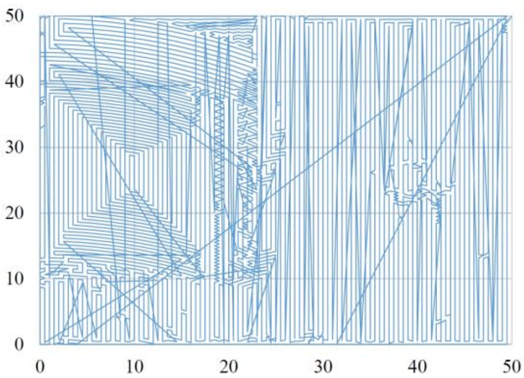


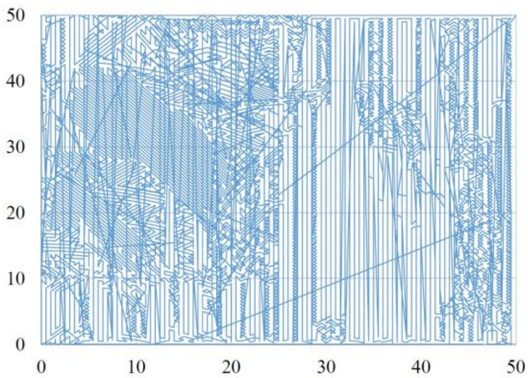
Figure 6. *Original tool path.*

In **Fig. 6 and 7**, a tool path comparison between the original and optimized versions is presented. The optimized tool path originates from two positions: the point with index 0, coordinates 50, 50, 10 (located in the top right corner of the workpiece), and the point with index 4269, coordinates 28.500, 43.919, 10.00 (situated in the middle of the workpiece). The optimization considers both 4 and 8 neighbour points.

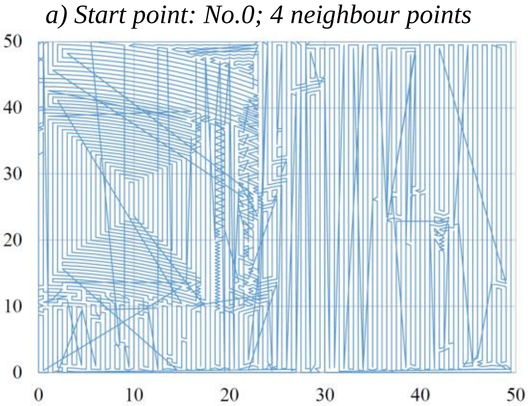
It is important to note that the generated path contains jump points. These occur when all adjacent points to the current position are marked as visited. In such instances, the algorithm seeks the closest unvisited point in the tool-position list. However, this optimization strategy involves a trade-off; the



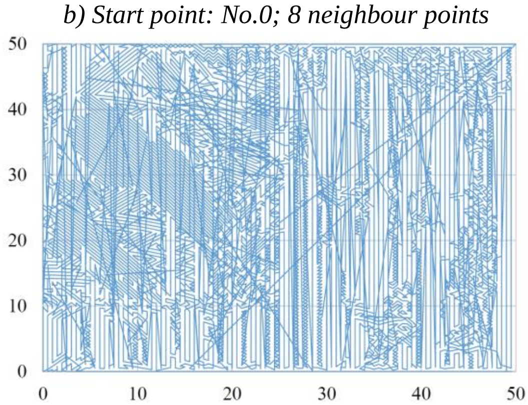
a) Start point: No.0; 4 neighbour points



b) Start point: No.0; 8 neighbour points



c) Start point: No.4269; 4 neighbour points



d) Start point: No.4269; 8 neighbour points

Figure 7. *Optimized tool path.*

tool may traverse a greater distance in the optimized path compared to the original path.

The result of the optimized toolpath depends on the position of the starting point, and the number of the investigated neighbour points. Therefore, four cases were studied (**Table 2**).

The comparison of the four tool path planning cases unveils distinct trends influenced by the selected starting points and the number of neighbouring points. **Fig. 8a** illustrates the traveling distance in the four cases. Cases 1 and 3, both originating from the top right corner but differing in neighbouring points, demonstrating that an increase in neighbouring points leads to a longer tool path (6183) for Case 1 with four neighbours and 8805 for Case 3 with eight neighbours. Similarly, Cases 2 and 4, both starting from the middle but varying in neighbouring points, follow a similar pattern, with Case 4 (10645) having a longer tool path compared to Case 2 (7097). Conversely, **Fig. 8b** highlights the impact of starting points and neighbouring points on

Table 2. *Parameters of the simulation.*

Case	Starting point	Neighbours
1	0	4
2	4948	4
3	0	8
4	4948	8

jumping points. Initially comparing Cases 1 and 2, both starting from distinct locations but with four neighbouring points, reveals a minimal difference in jumping points (181 in Case 1 and 180 in Case 2). This suggests that the starting point's influence on jumping points is relatively limited when the number of neighbouring points remains constant. A parallel trend is observed in Cases 3 and 4, where starting from the middle but with different neighbouring points results in comparable jumping points (426 in Case 3 and 432 in Case 4). This underscores that the number of neighbouring points plays a more significant role than the starting point in determining the number of jumping points.

On the other hand, the main aim of the optimization was to decrease the changing of the effective diameter and moderate the dynamic load of the spindle. This aim was measured by the sum of the absolute value of the changing of effective diameter. Two cases were compared. The first case was the result of the spindle speed optimization, without any modification of the toolpath, and the second was the tool path optimization.

Fig. 9 and 10 provide a visual representation of the change in the working diameter when employing two different milling strategies: the traditional zig-zag milling tool path and an optimized tool path. The working diameter, a critical parameter in machining operations, is analysed to understand its variation and impact on the machining process.

In the case of the traditional zig-zag milling tool path, **Fig. 9** illustrates a significant range of change in the working diameter, spanning from 0 to 0.9155. This wide variation suggests that the machining process under traditional zig-zag milling conditions results in a less consistent effective diameter throughout the operation.

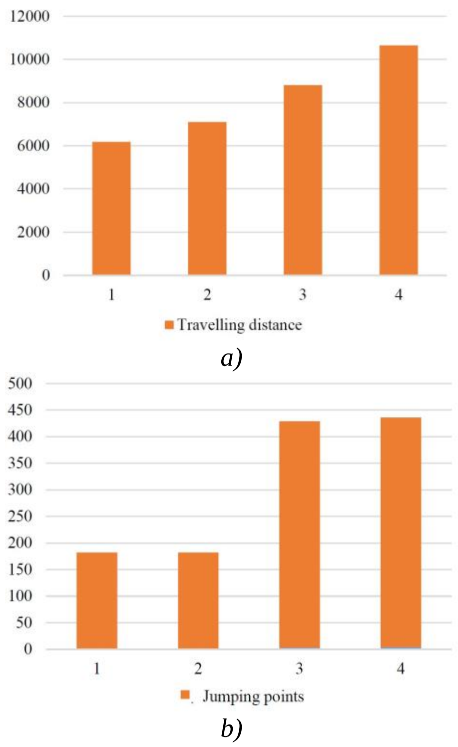


Figure 8 *a) Travelling distance and b) Number of jumping points at each case.*

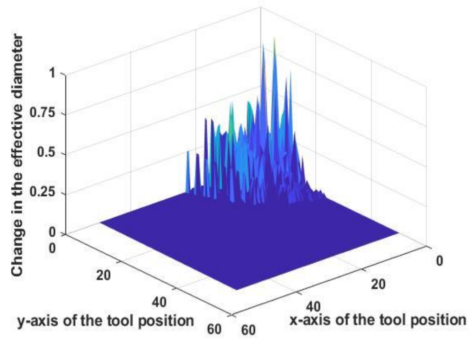


Figure 9. *The change in the working diameter in the case of original tool path.*

On the other hand, **Fig. 10** showcases the working diameter variation when utilizing an optimized tool path. In this scenario, the effective diameter fluctuates within a narrower range, specifically between 0 and 0.6. This narrower range indicates that the optimized tool path leads to a more controlled and predictable working diameter during the milling process.

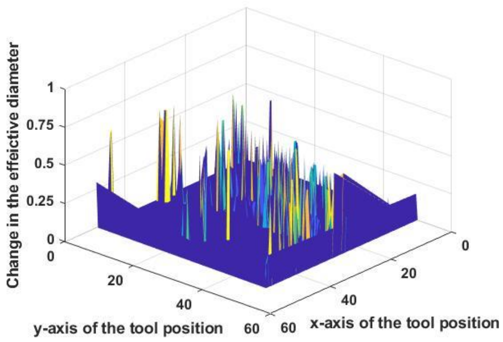


Figure 10. *The change in the working diameter in the case of optimized tool path.*

The observed difference in the effective diameter range between the two milling strategies has noteworthy implications, particularly in terms of surface homogeneity. A smaller variation in the working diameter, as achieved with the optimized tool path, contributes to enhanced surface homogeneity.

Fig. 11 illustrates the spindle speed change in the case of the traditional zig-zag milling approach, showcasing a wide range from 0 to 622. This broad variation implies that under traditional zig-zag milling conditions, the spindle speed undergoes significant changes throughout the machining process. Such fluctuations may introduce challenges such as tool wear, vibration, and inconsistent cutting conditions, potentially impacting the overall quality of the machined surface.

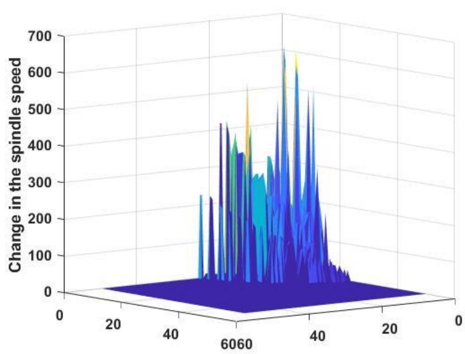


Figure 11. *The change in the spindle speed in the case of original tool path*

Conversely, **Fig. 12** represents the spindle speed variation for the optimized tool path. In this case, the spindle speed changes within a smaller range, specifically from 0 to 128. The narrower variation indicates that the optimized tool path leads to a more controlled and stable adjustment of spindle speed during machining.

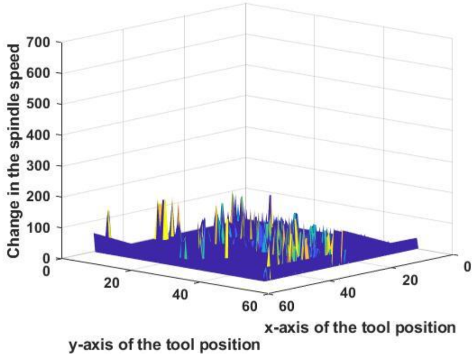


Figure 12. *The change in the spindle speed in the case of optimized tool path*

IV. CONCLUSION

The article presents a new concept for milling tool path optimization, when the tool path is modified based on the changing of the working diameter of the ball-end tool.

The presented algorithm was designed for optimizing toolpaths in CNC machining, with a primary focus on minimizing variations in the working diameter of the ball-end tool during milling of free-form surfaces. The algorithm reorders the tool points of the CNC program, and it works as a searching algorithm. It systematically calculates the working diameter of neighbour points and directs tool movement to points with minimal diameter differences, ensuring minimal changing of the working diameter and the changing of the spindle speed in order to maintain the constant cutting speed.

Key steps involve reading and processing data from an STL file to extract surface geometry and tool position data from an APT file. Normal vectors are calculated and associated with tool positions, aligning the tool with specific triangles of the STL file.

Neighbour point calculations consider a distance threshold, and working diameters are determined for each point. The core path planning algorithm minimizes working diameter fluctuations, preventing point revisits for efficiency.

The comparative analysis between the original and optimized tool paths highlighted the nuanced impact of starting points and neighbouring points on the resultant tool path. Successfully achieving the primary optimization goal - reducing the change in effective diameter and moderating spindle dynamic loads - was evidenced in the controlled working diameter variation observed in the optimized tool path, contributing to enhanced surface homogeneity compared to traditional zig-zag milling, from the viewpoint of the variation of the surface roughness. From the aesthetic aspect, the surface will not homogenous, because of the non-regular tool path.

Author contributions

Mgherony A.: Conceptualization, Implementation, Testing , Writing, Data visualization.

Mikó B.: Writing, Review and editing. Supervision.

Disclosure statement

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

ORCID

Mgherony A. http://orcid.org/0009-0001-4590-3057

Mikó B. http://orcid.org/0000-0003-3609-0290

References

[1] Q. Zou, J. Zhang, B. Deng, and J. Zhao, Iso-level tool path planning for free-form surfaces. Computer-Aided Design 53 (2014) pp.117–125. https://doi.org//10.1016/j.cad.2014.04.006

[2] L. C. Magalhães and J. C. E. Ferreira, Assessment of tool path strategies for milling complex surfaces in hardened H13 steel. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture 233 (3) (2019) pp. 834–849. https://doi.org// 10.1177/0954405418755824

[3] X. Niu, T. Wang, B. Shen, T. Zhao, and Q. Huang, Study on Tool Path Optimization in Multi-axis NC Machining. MATEC Web of Conferences 34 (2015) 02012. https://doi.org//10.1051/matecconf/20153402012.

[4] Y. Sun, J. Jia, J. Xu, M. Chen, and J. Niu, Path, feedrate and trajectory planning for free-form surface machining: A state-of-the-art review. Chinese Journal of Aeronautics 35 (8) (2022) pp. 12–29. https://doi.org//10.1016/j.cja.2021.06.011

[5] J. Varga, T. Tóth, Ľ. Kaščák, and E. Spišák, The Effect of the Machining Strategy on the Surface Accuracy When Milling with a Ball End Cutting Tool of the Aluminum Alloy AlCu4Mg. Applied Sciences 12 (20) (2022) 10638. https://doi.org//10.3390/app122010638

[6] D.-D. Vu, F. Monies, and W. Rubio, A new optimization tool path planning for 3-axis end milling of free-form surfaces based on efficient machining intervals. AIP Conference Proceedings (2018) 070011. https://doi.org//10.1063/1.5034907.

[7] G. Huo et al., CNC Tool Path Generation for Freeform Surface Machining Based on Preferred Feed Direction Field. International Journal of Precision Engineering and Manufacturing 20 (5) (2019) pp. 777–790. https://doi.org//10.1007/s12541-019-00084-2.

[8] R. A. Mali, R. Aiswaresh, and T. V. K. Gupta, The influence of tool-path strategies and cutting parameters on cutting forces, tool wear and surface quality in finish milling of Aluminium 7075 curved surface. The International Journal of Advanced Manufacturing Technology 108 (1-2) (2020) pp. 589–601. https://doi.org//10.1007/s00170-020-05414-7.

[9] A. Kukreja and S. S. Pande, Optimal toolpath planning strategy prediction using machine learning technique. Engineering Applications of Artificial Intelligence 123 (2023) 106464. https://doi.org//10.1016/j.engappai.2023.106464

[10] U. Župerl, K. Stepien, G. Munđar, and M. Kovačič, A Cloud-Based System for the Optical Monitoring of Tool Conditions during Milling through the Detection of Chip Surface Size and Identification of Cutting Force Trends. Processes 10 (4) (2022) p. 671. https://doi.org//10.3390/pr10040671

[11] J. Zhang, R. Mo, N. Wan, and C. Xia, Tool path planning for five-axis flank milling of free-form surfaces. The International Journal of Advanced Manufacturing Technology 108 (1-2) (2020) pp. 73–90. https://doi.org//10.1007/s00170-020-05283-0

[12] A. Kukreja and S. S. Pande, "An Efficient Iso-Scallop Toolpath Planning Strategy Using Voxel-Based Computer Aided Design Model," Journal of Computing and Information Science in Engineering 23 (3) (2023) 031009. https://doi.org//10.1115/1.4055372

[13] W. Liu, S.-M. Zhu, T. Huang, and C. Zhou, An efficient iso-scallop tool path generation method for three-axis scattered point cloud machining. The International Journal of Advanced Manufacturing Technology 107 (7-8) (2020) pp. 3471–3483. https://doi.org//10.1007/s00170-020-05188-y

[14] W. Liu, L. Zhou, and L. An, Constant scallop-height tool path generation for three-axis discrete data points machining. The International Journal of Advanced Manufacturing Technology 63 (1-4) (2012) pp. 137–146.

https://doi.org//10.1007/s00170-011-3892-3

[15] A. Mgherony and B. Mikó, "Controlling the Spindle Speed when Milling Free-Form Surfaces using Ball-End Milling Cutter," Acta Polytechnica Hungarica 20 (6) (2023) pp. 135–149. https://doi.org//10.12700/APH.20.6.2023.6.8

[16] B. Mikó and P. Zentay, A geometric approach of working tool diameter in 3-axis ball-end milling. The International Journal of Advanced Manufacturing Technology 104 (1-4) (2019) pp. 1497–1507. https://doi.org//10.1007/s00170-019-03968-9