INTERNATIONAL JOURNAL OF AUTOMOTIVE AND MECHANICAL ENGINEERING
ISSN: 2229-8649 e-ISSN: 2180-1606
VOLUME 22, ISSUE 3, 2025, 12665 - 12674
DOI: https://doi.org/10.15282/ijame.22.3.2025.10.0967

RESEARCH ARTICLE

Real-Time Interpolator of CNC Parametric Curves with Chord-Error and Feed-rate Constraints

W. R. W. Yusoff[1], F. R. M. Romlay[1]* and I. Ishak[2]

[1]Faculty of Mechanical and Automotive Engineering Technology, Universiti Malaysia Pahang Al-Sultan Abdullah, 26600 Pahang, Malaysia
[2]Faculty of Manufacturing and Mechatronic Engineering Technology, Universiti Malaysia Pahang Al-Sultan Abdullah, 26600 Pahang, Malaysia

ABSTRACT – Acknowledging that the core in computer numerical control (CNC) machining is the interpolator of the controller, the work presents a real-time interpolator of a class of non-uniform rational basis splines, NURBS curve, which is inherently parametric. Teardrop 2-dimensional NURBS curve was selected based on varying features, shapes and dimensions. The curve characteristics cover variations of dimensional geometry, origin position, closed or open-ended curves, varying loop count, segment type, either convex or concave, turning profile smoothness and different reflection symmetry about the x and y axes. The real-time interpolator algorithm, when applied to all the selected curves exclusively and simultaneously, satisfies both of its designed constraints, which cover its feed-rate and its chord-error tolerance. The resulting feed-rate profiles throughout the entire path of the curves are continuous and smooth. The feed-rate constraints comprise dynamic equations for allowable CNC machine parameters like the maximum and minimum axial velocities and the maximum and minimum axial accelerations. The chord-error constraints comprise geometric and kinematic properties of different parametric curves, covering bends and sharp turns. The resulting algorithm was executed both in real-time, online mode by driving the CNC machine directly, and in an offline mode by using a stored RS274/NGC G-code file. The plot conclusively shows that the chord-error (u) at every u-point in the full range of $0.00 \leq u \leq 1.00$ lies below the chord-error tolerance, which was set at 1×10^{-6} mm.

ARTICLE HISTORY
Received : 23rd July 2024
Revised : 03rd Mar. 2025
Accepted : 29th July 2025
Published : 07th Sept. 2025

KEYWORDS
Interpolator
NURBS Curve
CNC Machining

1. INTRODUCTION

Today, the use of software to control the movements of some systems is almost ubiquitous. With the authors having access to a bare-bones prototype computer numerical control (CNC) machine, the interest grew in the complex world of animating non-animated objects. This led to the current endeavor in the CNC machine and its control software, the interpolator program. Basically, to produce the physical part that is being machined, the interpolation program generates commands that move motor drives step-by-step and point-to-point to follow the desired path or trajectory. Moving inanimate objects is not limited to just cutting things, like in the CNC machine. It is also about robots moving around, avoiding obstacles, and performing actions beyond what humans can do in the air or underwater, in guided or autonomous modes. The possibilities are endless. Many CNC systems today support parametric interpolators because of their various advantages over traditional linear and circular interpolations. The linear interpolator (G01) and the circular interpolator (G02, G03) are the only two interpolators defined in the RS274 standard [1]. This standard does not yet have the non-uniform rational basic splines (NURBS) interpolator that can handle parametric curves and surfaces. Parametric interpolation is conducted as a point-to-point (PTP) movement in a CNC machine, as shown in Figure 1. At the end of each motion, it is important that the positional accuracy of the tool relative to the workpiece is achieved, that is, within a specified error tolerance, ϵ. At the end of each motion, the tool performs its required task, after which the next motion begins and the cycle repeats until all machining is completed [2].

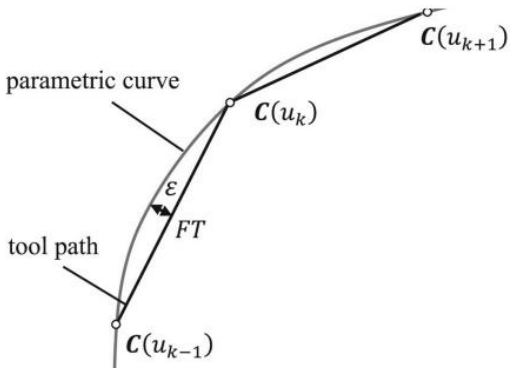

Figure 1. Concept of chord-error, feed-rate, F and interpolation time, T [3]

*CORRESPONDING AUTHOR | F.R.M. Romlay | ✉ fadhlur@umpsa.edu.my

In CNC machine operation, the function of interpolation is to generate coordinated movements to drive the separate axes of motion to achieve the desired path of the CNC cutting tool relative to the workpiece. Essentially, interpolation generates commands that move motor drives to follow the desired path or trajectory to produce the physical part that is being machined. A comprehensive approach to the simulation of multi-axis CNC machining, addressing the tool's kinematics and collision detection methods as reported by Gao et. al. [4]. Enhancements in parametric curves for CNC machining are essential for enhancing the precision, efficiency, and versatility of the machining process. Several key improvements in parametric curve interpolators have been considered, including the tool kinematics and dynamics, algorithm optimization [5], real-time processing [6], error minimization [7][8], feed-rate and adaptive optimization [9][10]. Some parallel processing strategies have been executed in order to get a real-time outcome of the data acquisition for feedback implementation [11]. Wenbin et al. initiated an effort to improve the precision of CNC interpolators and focus more on real-time interpolators for parametric curve motion [12]. To that extent, [13][14] came out with novel control approaches for the next generation of CNC systems. The invention has been applied to the CNC machining application, promising more precision in motion performance.

2. RELATED WORKS

Generally, the contour error is a measure of how close the actual tool path is to the desired tool path. For a two-dimensional parametric curve, the point-to-point movement turns the contour error (positional accuracy) into the chord-error, ϵ. This is the error between two interpolated points for the parametric curve [15]. For a specific curve path, an increase in feed-rate F will decrease the total number of interpolated points but will increase the chord length (FT) of each arc segment, thereby increasing the chord-error ϵ. On the other hand, a decrease in feed-rate will increase the total interpolated points and reduce the chord-error, but will cause the overall machining operation to be slow and time-consuming. Thus, there is an interplay between the chord-error ϵ, and the machine feed-rate F. The problem is how to find an optimum balance. This work proposed a solution by developing a real-time parametric curve interpolation algorithm that constrains both the feed-rate and chord-error simultaneously. It was developed and tested for ten (10) different 2-dimensional parametric curves. 2 The curves were selected based on complex shape characteristics like closed loop curves, open-ended curves, symmetric or non-symmetric about the x-axis and y-axis. Some curves have sharp, concave, or convex turns, including cusps. In addition, the x and y dimensions, that is, their overall sizes, vary among the different curves. These are some of the geometrical constraints of the algorithm. The dynamical and kinematical constraints cover specific CNC machine characteristics like the maximum and minimum allowable machine velocities and accelerations for the different axes of motion, the user-specified command feed-rate, the jerks and jitter that, in combination, affect the smooth operation of the machine [16], [17], [18].

Recent studies have extensively explored adaptive feed-rate scheduling for NURBS interpolation under chord-error and kinematic constraints. For instance, [19] proposed a sigmoid-function-based feed-rate profile that adaptively segments curves, balancing geometric error and dynamic limits smoothly and enabling time-optimal transitions at breakpoints without abrupt acceleration jumps. Similarly, [20] developed an S-shape acceleration/jerk-continuous scheduling algorithm, which eliminates round-off errors and ensures feed-rate, acceleration, and jerk continuity at subcurve transitions, improving interpolation stability and surface finish. To reduce feed-rate fluctuation due to approximation errors in chord-length parameter mapping, recent approaches, including a two-level parameter compensation method, have shown promise. By combining Taylor-series estimations with secant-based refinement, this method delivers sub-0.01 % fluctuation in simulations, achieving high-precision interpolation with acceptable real-time performance [21]. [22] introduced an interpolation method based on arc-length compensation and iterative parameter correction, significantly reducing feed-rate deviation and enhancing smoothness during high-speed machining. Several researchers have proposed efficient NURBS interpolators that emphasize both computational speed and geometric accuracy. [23] presented an interpolator using polynomial representations and a prediction–correction framework tailored for sensitive corners: the method avoids abrupt jerk, ensures continuity, and achieves real-time performance validated in both simulation and experiments [23]. [24] applied a hash-based refinement algorithm for arc-length estimation in NURBS parameterization, thus improving computational precision and arc-length mapping efficiency, which is critical for accurate chord-error maintenance.

Furthermore, integrated feed-rate scheduling methods combining offline and online strategies have gained traction. [25] introduced an adaptive real-time NURBS interpolator for a 4-axis polishing machine, integrating S-shaped ACC/DEC control to meet chord-error and jerk constraints in real-time machining. [26] proposed a generalized NURBS interpolator that applies nonlinear feed-rate scheduling alongside interpolation-error compensation, enabling dynamic adaptation to curvature variations without sacrificing contour fidelity. Earlier foundational research continues to influence current advances: adaptation to chord-error and axis jerk limits in parametric toolpath interpolation were first addressed in articles by [27] and [28], while [29] presented look-ahead window interval adaptive feed-rate scheduling under axis drive constraints. These methods have recently formed the baseline for modern real-time interpolator development. Since emphasizing look-ahead interpolator and dynamic smoothing strategies are very crucial, the main objective of the algorithm execution is to ensure that the resulting running feed-rate (speed motion of the cutting tool) is smooth and continuous, and does not exceed the feed-rate limit throughout the full curve path. Note that the feed-rate limit varies with u, and thus, the running feed-rate also varies, for example, when negotiating curves and sharp bends. The algorithm accomplishes the tool motion strictly without violating both the chord-error and running feed-rate constraints. Since the

smoothness of running feed-rate is critical to the success of the algorithm, any acceleration jitters (rapid acceleration fluctuations) will result in a jerky machine feed-rate and need to be fine-tuned accordingly.

3. METHODOLOGY

3.1 Main Program Interpolation

Error mitigation techniques that are applicable to smoothing curves, such as spline smoothing or curvature-based adjustments, ensure the cutter path stays close to the desired path and satisfies the error threshold. Structure the methodology by starting with the formulation of the path planning problem, then describe how the Taylor expansion is applied, followed by how to handle error control for better curve smoothing. The derivations of the first and second order iterative Taylor's expansion followed. The work then continues with the calculations for the next interpolation point. The discussions and equations for the various feed-rate limits (*frate_limit*), consisting of the combined dynamical and geometrical constraints, were presented. The main program for the interpolation was described through a flowchart as shown in Figure 2. The parameters involved were curvature radius (rho), chord-error (epsilon), and current feed-rate (curr_frate). The parameter update $u = u + u\ next$ is executed for each loop.

There are three main processing modules that cover computations involving u, that is, in the rising, main and falling feed-rate sections. The computations for the main section are further separated into 3 cases. When the current feed-rate is above the feed-rate limit, it is called Case A. When the current feed-rate is below the feed-rate limit, it is called Case B. When the current feed-rate is equal to the feed-rate limit, it is called Case E. The processing in Case A is basically to push down the running current feed-rate to just below the calculated feed-rate limit. On the other hand, the processing in Case B is to push up the current feed-rate to also be just below the calculated feed-rate limit. The net result is that the current feed-rate will be very close but just below the calculated feed-rate limit. This is the feed-rate constraint part of this work. This represents modules that are invoked by the feed-rate rising, main and falling modules. This is the implementation of the Call-and-Return (CAR) software architecture. This last module is responsible for pushing down the chord-error (epsilon) below the error tolerance of 10^{-6} . This is the chord-error constraint part of this work.

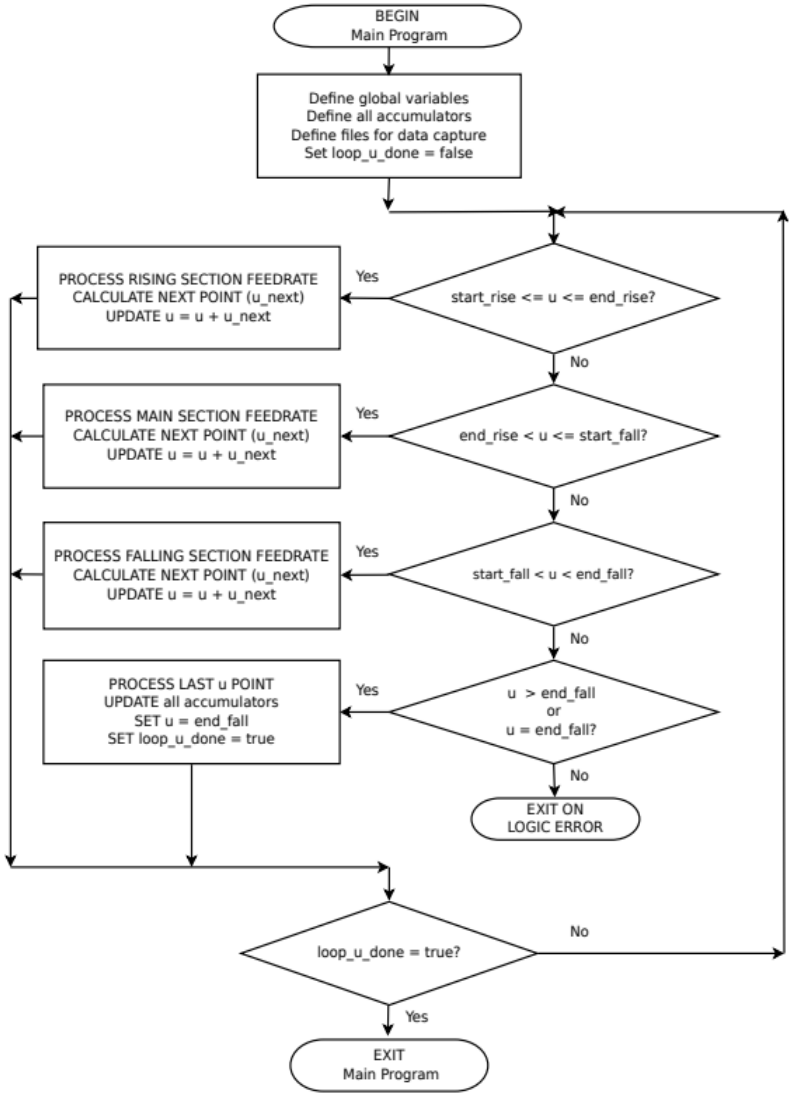

Figure 2. Flow chart of the main algorithm

3.2 Teardrop Curve Execution

The job of the interpolation algorithm is to generate successive points along the curve trajectory so that the CNC cutting tool path illustrated by the profile in Figure 3 follows the path accurately. The curve begins at parameter u = 0.00 (starting point), increasing in steps until u = 1.00 (ending point), as the entire teardrop curve is being followed, that is, from start to finish [30]. The algorithm uses the second-order Taylor's approximation to calculate the steps *(u-next)* in parameter *u*, and at the same time constrains both the chord-error (deviation from the true curve path) [31]. It is defined to be a lower set error tolerance (1E-6 mm), and the running feed-rate is to be very close but below the feed-rate limit throughout the full curve path. The feed-rate limit at every parameter u point is calculated by the algorithm based on geometrical, dynamical, and kinematical constraints. The constraints comprise four different components: user-set feed-rate command FC, minimum and maximum CNC machine axial velocities, minimum and maximum CNC machine axial accelerations, and the geometric factors of the curve path, like bends and sharp turns.

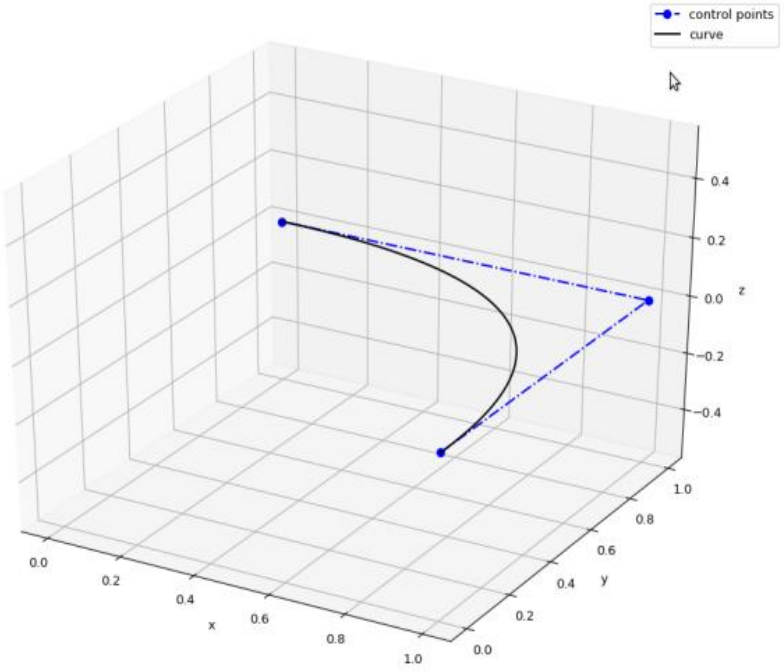

Figure 3. The curve in the 3D view

The radius of curvature, rho, for the teardrop curve is shown in Figure 4. The correctness of the radius of curvature, *rho(u),* is such that it must follow the corresponding curve it represents. For the teardrop, the value of *rho(u)* starts high at the beginning because it is near a straight line, and then slowly curves inward, thus *rho(u)* becomes lower, and at the end *rho(u)* rises high again, approaching a near straight line as it reaches its endpoint. Note that the term "near straight line" has been used because a "true straight line" represents the radius of curvature as being infinite, and that there is "no curving" effect at all.

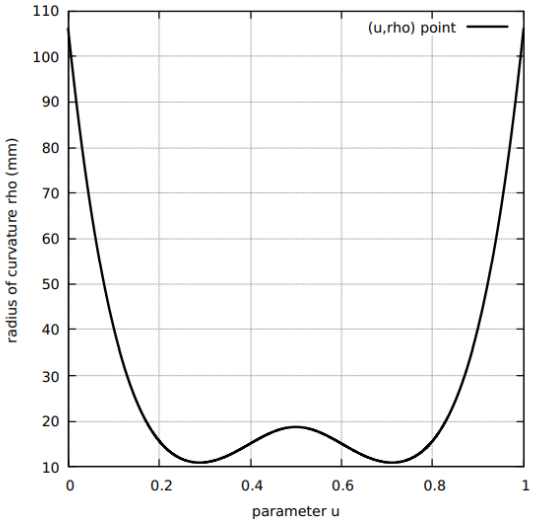

Figure 4. The teardrop radius of curvature

On the other hand, a zero value for the radius of curvature, *rho(u)*, represents a single point, and the curving effect is meaningless. Note that the *rho(u)* over the entire parameter range is symmetrical because the teardrop curve for the parameter *(u)* movement is symmetrical. If the parameter *(u)* movement is not symmetrical, then the *rho(u)* over the entire parameter range is still correct, but just not symmetrical. This will be the case for some of the rest of the curves in this work. The *rho(u),* in general, is a good initial visualization tool to validate and indicate the correctness of shapes and features of any curve.

3.3 Total Algorithm Executions

In this work, the teardrop curve algorithm was executed at four (4) different feed-rate commands (FC 10, 20, 30, 40) at four (4) different Lambda safety factors (Lambda 0.10, 0.18, 0.20, 0.50). This combination makes the total number of algorithm executions 160. For each execution, three (3) different input parameters are required; the curve type selected, the feed-rate command, FC, and the lambda safety factor. It is expected that a variety of issues arise in these executions due to the diverse characteristics and sizes of the four (4) curves, the different running feed-rates against the machine limits, and the most suitable value of the lambda safety factor. The teardrop Taylor's order approximation, feed-rate components profile, feed-rate absolute constraint, and chord-error absolute constraint must follow the corresponding curve as closely as possible, which will be discussed in the forthcoming sections.

4. RESULTS AND DISCUSSION

4.1 Teardrop Taylor's Order Approximation

The first-order and second-order terms in Taylor's approximation are utilized in characterizing the teardrop profile and are provided in Figure 5. The plot for the difference in value between the first-order and second-order terms in Taylor's approximation is provided in Figure 6 on the next page. Take note that in Taylor's series expansion, the largest term is the first-order term, the second-order term is smaller, and the third-order term is even smaller, so it continues to an infinite number of smaller terms, theoretically. In addition, as in Figure 6, the sign of the successive terms in Taylor's series expansion alternates between positive (+) and negative (-), with the starting first-order term being positive (+). That is the reason the plot of the difference is calculated as (first order - second order) terms, thus giving a positive value. It is the same reason for separating (shifting) the two curves in the previous figure, making the second-order term lower than the first-order term. The difference in value between the two terms exists but is too small to be visible on the same scale as the previous figure. Therefore, the current figure displays only the difference on a single and expanded y-axis scale.

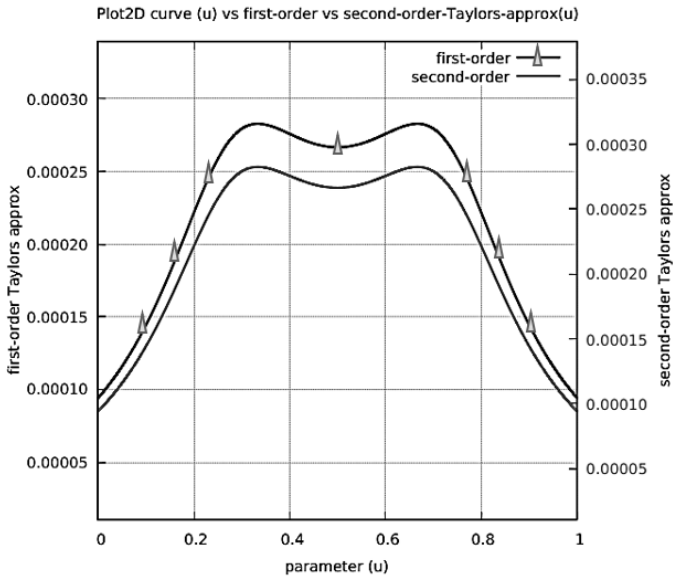

Figure 5. The teardrop 1st and 2nd Order Taylor's approximation

The performance result for each curve is the ratio of total accumulated error against the total path length traversed. This is a meaningful performance measure for the interpolation algorithm since it measures the amount of chord-error generated by the algorithm per unit length of curve traveled. The total interpolated points is not a meaningful measure because each curve has a different total length. The results also provide trending data for algorithm executions against four (4) different feed-rate commands (FC10, FC20, FC30 and FC40) values for each of the curves. The interpretation of the total number of interpolated points for each FC value for comparison becomes meaningful. However, results showed that there were jitters (severe fluctuations) in feed-rate at several segments along the path. The feed-rate profile was no longer smooth. Knowing that the chord-error is already below tolerance, the advice taken is to ignore the push up (keep it if it is already below tolerance). And it is sensible because this would keep the total chord-error small and the feed-rate profile smooth. Therefore, the chord-error push-up idea was abandoned. Only a chord-error pushdown was executed in the algorithm.

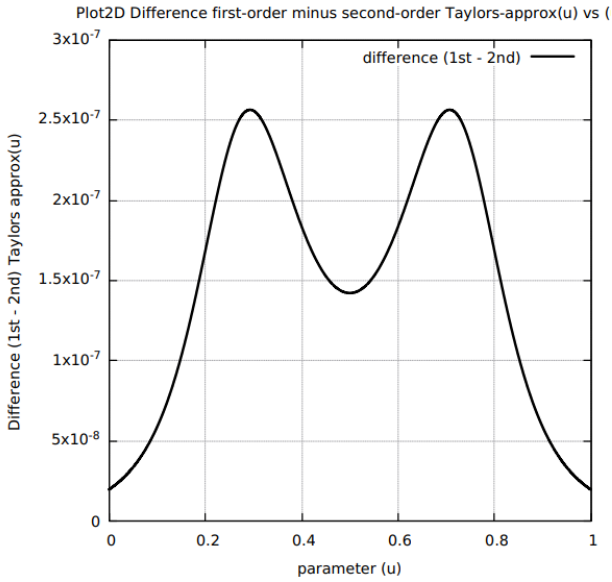

Figure 6. The teardrop (1st minus 2nd) Order Taylor's approximation

4.2 Teardrop Feed-Rate Limit Components Profile

The results for the four (4) teardrop feed-rate limit components are provided in Figure 7. The value of the feed-rate limit at any point *u*, feed-rate *limit(u)*, is the minimum value among the four components, evaluated at the same point u. The figure shows that the curve of the limit-4 acceleration confinement is the most dominant (lowest value) among the four limit components throughout the middle range, outside of the rising S-curve and falling S-curve sections.

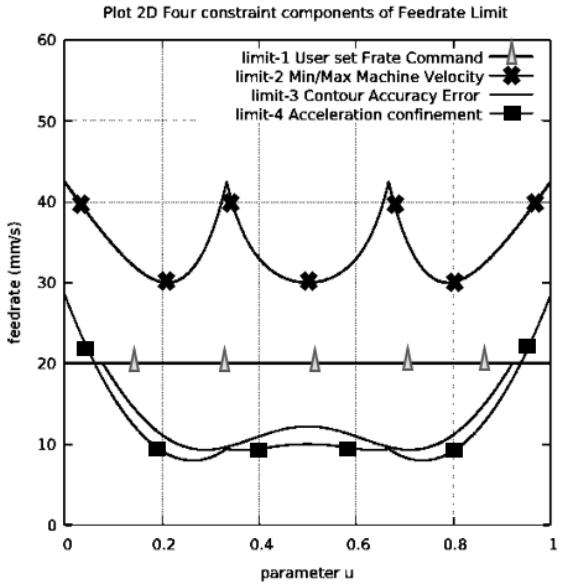

Figure 7. Teardrop four (4) feed-rate limit components profile

This plot is based on the Teardrop curve running at FC20 and lambda = 0.18 (restrictive acceleration confinement). If the value of Lambda is increased, for example, to Lambda = 0.25 (relaxed acceleration confinement), then the limit-4 acceleration confinement curve will increase, thus crossing with the Limit-3 contour accuracy error or Chord-error accuracy. This resulted in the minimum value changing between these two curves. This is how the value of the Feed-rate *Limit(u)* is calculated by the algorithm. The finding showed that the Lambda = 0.25 for the Teardrop curve running at FC20 will cause acceleration jitters, thus making the running feed-rate jerky in the affected regions. This result violates the requirement for absolute smoothness in feed-rate.

4.3 Teardrop Feed-Rate Curve Execution

The teardrop curve execution results for the feed-rate limit and the current feed-rate that is shown in Figure 8. The terms current feed-rate and running feed-rate are used interchangeably in this document. The feed-rate command *(frate-command(u))* is a user-specified constant value, and in this case, FC = 20 mm/s. The feed-rate limit is calculated by the interpolation algorithm at every parameter point u based on the minimum of four (4) feed-rate component limits. The current or running feed-rate is recursively calculated by the algorithm as close to but below the feed-rate limit *(frate-*

limit(u)). In the figure, only two out of the three curves are visible. This is due to the current or running feed-rate being seen as overlapping with the feed-rate limit. Their values are close but different.

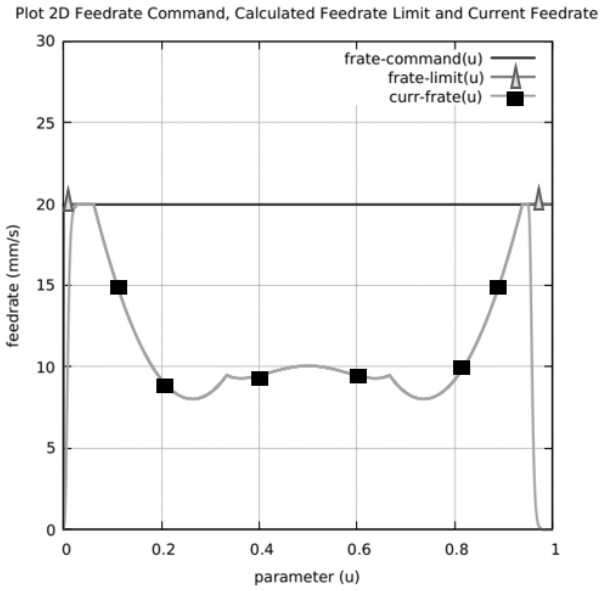

Figure 8. Teardrop frate command, frate limit and current frate

4.4 Teardrop Current Feed-Rate Absolute Constraint

The feed-rate limit is always higher than the running feed-rate, even though the difference is very small. Figure 9 displays the difference in value between the feed-rate limit and the current or running feed-rate for the teardrop curve. The display shows a positive or zero value difference for the entire range of u values of the teardrop curve, without exception. There are no negative differences. Therefore, the running feed-rate is constrained to be below the feed-rate limit. This absolute constraint on the running feed-rate is one of the main concerns in the study.

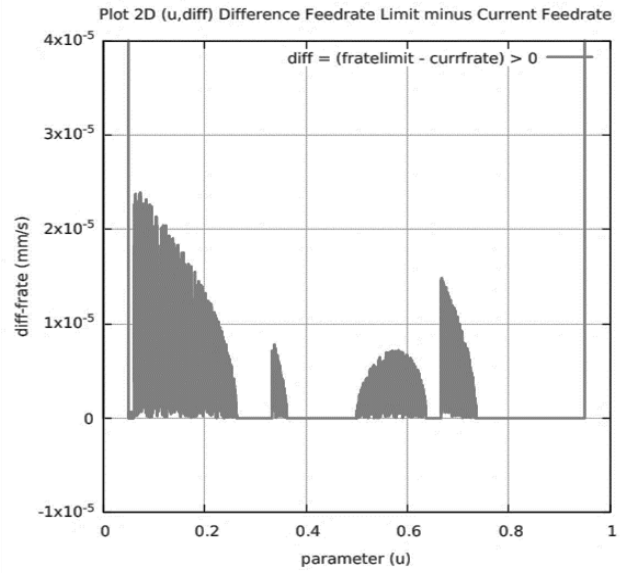

Figure 9. Teardrop current feed-rate absolute constraint

4.5 Teardrop Chord-Error Absolute Constraint

The results for the teardrop chord-error absolute constraint are provided in Figure 10. The two curves are chord-error, but are plotted on two different scales, using the scale on the left y-axis and the right y-axis. Note that it is the same value dataset, but plotted on two different scales. The situation is different from the case of the first-order and second-order terms in Taylor's expansion, which comprises two different calculated data sets. This means that if only a single curve is displayed for the chord error, a straight horizontal line will be seen in the middle section. The idea for a second expanded scale for the close view (same chord-error data set) is to visualize the fluctuations or jitters calculated by the algorithm. This graphical plot conclusively shows that the chord-error(u) at every u-point in the full range of ($0.00 \leq u \leq 1.00$) lies below the chord-error tolerance, which was set at 1E-6 mm.

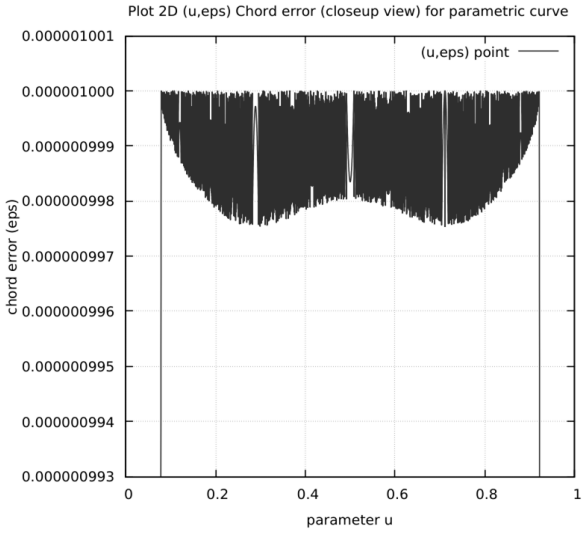

Figure 10. Teardrop chord-error absolute constraint

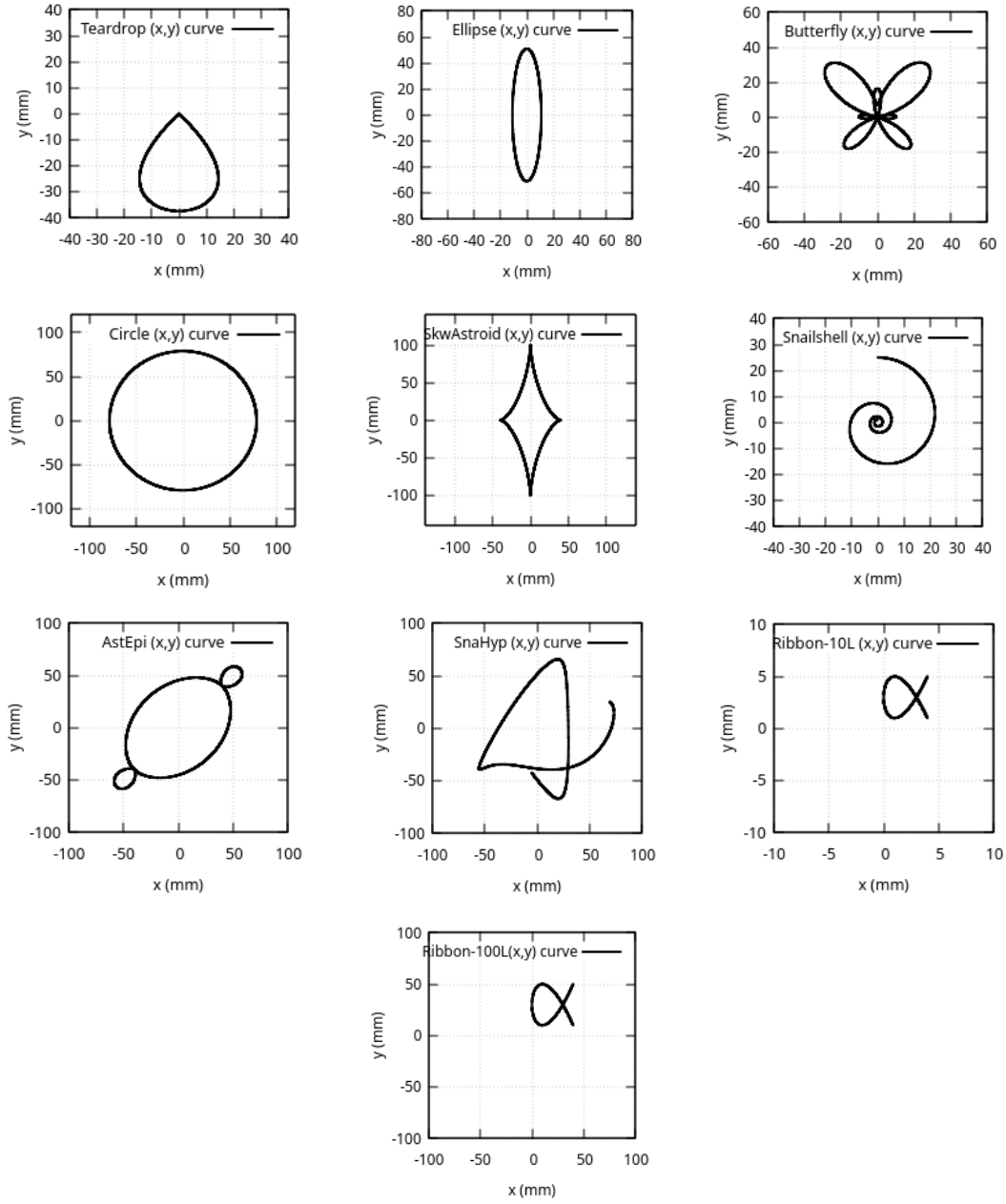

Figure 11. Ten curves profile that generated and tested

The plots have been generated for ten parametric 2D curves, which are displayed in Figure 11. The curves represent various types of curves with different levels of difficulties that cover ellipse, butterfly, circle, skwastroid, snailshell, astepi, snasyp, ribbon 10L and 100L. The figure RIBBON-100L is 10 times larger (scale-up) for the same parametric equation as RIBBON-10L to see the possibility of inaccuracy due to the scale-up case. The algorithm execution is to ensure the speed of the cutting tool is smooth and does not exceed the feed-rate limit throughout the full curve path. The feed-rate limit varies with u, and thus, the running feed-rate also varies when approaching curves and sharp bends. The algorithm accomplishes the tool motion strictly without violating both the running feed-rate constraints and chord-error. The running feed-rate is maintained as close to but below the feed-rate limit. Further verification is made through the chord-error data to look at the fluctuations or jitter patterns. Those 10 plots of curves with different levels of complexity maintain the chord-error(u) data set at every u-point in a range of ($0.0 \leq u \leq 1.0$) that lies below the chord-error tolerance, which was set at 10^{-6} mm.

The RIBBON-10L and RIBBON-100L result shows that there is a possibility to achieve a higher current feed-rate for the larger curve that moves upward toward the user-provided feed-rate command (FC). Ideally, the current feed-rate is as close to and equal to the command feed-rate (horizontal line) throughout the run (from u = 0.0 to the end u = 1.0). Obviously, the current feed-rate cannot jump instantly, so we force a rising S-curve (feed-rate rising from zero) and a falling S-curve (feed-rate falling to zero) to make the feed-rate transition smooth and continuous. Thus, the algorithm successfully performs as the *chord-error(u)* between *u_next* and *u* below the error tolerance. Since the smoothness of running feed-rate is critical to the success of the algorithm, any acceleration jitters (rapid acceleration fluctuations) will result in a jerky machine feed-rate. This situation is successfully avoided.

5. CONCLUSIONS

The developed real-time interpolation algorithm was executed against the selected curves, exclusively and simultaneously satisfying both the design constraints on the feed-rate and its chord-error tolerance. The summary of work achievements is that the resulting feed-rate profiles throughout the entire path are continuous and smooth. The maximum feed-rate at every interpolated point remains below and close to the calculated feed-rate limit. The current running feed-rate does not exceed the user-specified command feed-rate at every point during the traversal along the entire parametric curve. The chord-error at every interpolated arc segment remains below the chord-error tolerance throughout the curve traversal. Without fail, a single and robust interpolation algorithm was developed that can handle the complexities of a wide range of 2D parametric curves comprising different sizes and shapes. The algorithm can be implemented both in real-time, online mode by driving the CNC machine directly, or in an offline mode by using a stored RS274/NGC G-code 238 standard file. This work satisfies the real-time CNC interpolation requirement. A parametric curve interpolation method that successfully constrains chord-error and feed-rate simultaneously with confinement of tangential acceleration. The importance of having a fully functional and modularized implementation code for interpolation that follows good software engineering practice.

ACKNOWLEDGEMENTS

This work was supported by the Research Scheme Grant (PGRS1903165) of Universiti Malaysia Pahang Al-Sultan Abdullah.

CONFLICT OF INTEREST

It is disclosed that the authors do not have any conflicts of interest regarding the manuscript contents.

AUTHORS CONTRIBUTION

W.R.W. Yusoff (Methodology; Validation; Formal analysis; Data curation; Formal analysis; Investigation; Resources; Software; Visualization; Writing - original draft)

F.R.M. Rahman (Conceptualization; Formal analysis; Writing – Formal Writing; Supervision)

I. Ishak (Conceptualization, Methodology; Resources)

REFERENCES

[1] S. H. Suh, S. K. Kang, D. H. Chung, and I. Stroud, *Theory and design of CNC systems*. London: Springer London, 2008.

[2] W. Zhong, X. Luo, W. Chang, F. Ding, and Y. Cai, "A real-time interpolator for parametric curves," *International Journal of Machine Tools and Manufacture*, vol. 125, pp. 133-145, 2018.

[3] N. H. Quang and B. T. Long, "A method of real-time nurbs interpolation with confined chord error for CNC Systems," *Vietnam Journal of Science and Technology*, vol. 55, no. 5, pp. 650-657, 2017.

[4] L. Gao, L. Chen, and H. Xie, "Collision detection for multi-axis CNC machining based on geometric modeling and kinematic simulation," *Journal of Materials Processing Technology*, vol. 209, no. 6, pp. 2870–2878, 2009.

[5] S. Jia, S. Wang, N. Zhang, W. Cai, Y. Liu, J. Hao, et al., "Multi-objective parameter optimization of CNC plane milling for sustainable manufacturing," *Environmental Science and Pollution Research*, pp. 1-22, 2022.

[6] X. Zhu, Y. Chen, and J. Wang, "A real-time look-ahead control and feedrate optimization strategy for CNC machining," *Journal of Manufacturing Processes*, vol. 50, pp. 123-134, 2020.

[7] L. Chen, H. Xu, Q. Huang, and P. Wang, "An integrated method for compensating and correcting nonlinear error in five-axis machining utilizing cutter contacting point data," *Scientific Reports*, vol. 14, p. 8763, 2024.

[8] R. Jamuna and U. Natarajan, "Measurement of error in computer numerical control machines and optimization using teaching–learning-based optimization algorithm," *Measurement and Control*, vol. 52, no. 7-8, pp. 929-937, 2019.

[9] X. Zhang, and M. Chen, "Adaptive parametric interpolation for CNC machining," *International Journal of Advanced Manufacturing Technology*, vol. 82, no. 5-8, pp. 1029-1039, 2016.

[10] Z. Li, Q. Yan, and K. Tang, "Multi-pass adaptive tool path generation for flank milling of thin-walled workpieces based on the deflection constraints," *Journal of Manufacturing Process*, vol. 68, pp. 690–705, 2021.

[11] L. Zhang, and H. Li, "Parallel processing for enhanced CNC interpolation performance," *International Journal of Advanced Manufacturing Technology,* vol. 112, no. 5-6, pp. 1487-1502, 2021.

[12] Z. Wenbin, L. Xichun, C. Wenlong, D. Fei, and C. Yukui, "A real-time interpolator for parametric curves," *International Journal of Machine Tools and Manufacture*, vol. 125, pp. 133-145, 2018.

[13] W. Zhong, "Novel Control Approaches for the Next Generation Computer Numerical Control (CNC) System for Hybrid Micro-machines," PhD Thesis, Centre for Precision Manufacturing, Department of Design, Manufacture and Engineering Management, University of Strathclyde, 2018.

[14] Y. Xiao, Z. Jiang, Q. Gu, W. Yan, and R. Wang, "A novel approach to CNC machining center processing parameters optimization considering energy-saving and low-cost," *Journal of Manufacturing Systems*, vol. 59, pp. 535–548, 2021.

[15] S. Sun, D. Yu, C. Wang, and C. Xie, "A smooth tool path generation and real-time interpolation algorithm based on B-spline curves," *Advances in Mechanical Engineering*, vol. 10, no.1, p. 1687814017750281, 2018.

[16] R. Ward, "Smooth trajectory generation for 5-axis CNC machine tools," PhD Thesis, University of Sheffield, 2022.

[17] S. S. Yeh, "Feed rate determination method for tool path interpolation considering piecewise-continued machining segments with cornering errors and kinematic constraints," *International Journal of Mechanical Engineering and Robotics Research*, vol. 8, no. 3, pp. 354-360, 2019.

[18] B. F. Yu, and J. S. Chen, "Development of an analyzing and tuning methodology for the CNC parameters based on machining performance," *Applied Sciences*, vol. 10, no. 8, p. 2702, 2020.

[19] H. Li, X. Jiang, and G. Huo, "A novel feedrate scheduling method based on Sigmoid function with chord error and kinematic constraints," *International Journal of Advanced Manufacturing Technology*, vol.119, pp. 1531–1552, 2022.

[20] C. C. Chen and J. Sun, "Real-time NURBS interpolator for CNC machining with axis jerk constraints," *The International Journal of Advanced Manufacturing Technology*, vol. 89, no. 9, pp. 3335–3344, 2017.

[21] S. Wenbin and D. Li, "Real-time interpolation of CNC machining with enhanced precision and stability," *Sensors*, vol. 23, no. 8, p. 3789, 2022.

[22] T. Wang, X. Qiao, and Y. Zhang, "Iterative feedrate compensation for high-accuracy CNC interpolators using arc-length correction," *Micromachines*, vol. 16, no. 4, p. 402, 2023.

[23] H. Huang and M. Chen, "Real-time CNC interpolator with prediction-correction and geometric smoothness control," *Machines*, vol. 13, no. 11, p. 456, 2024.

[24] S. Han, R. Liu, and Y. Wang, "Efficient arc-length parameter refinement in NURBS interpolation using hash-based search," *Journal of Mechanical Science and Technology*, vol. 38, no. 4, pp. 987–996, 2024.

[25] X. Ji, L. Wang, and F. Zhao, "Adaptive real-time NURBS interpolator for a 4-axis CNC polishing machine," *The International Journal of Advanced Manufacturing Technology*, vol. 117, pp. 1145–1158, 2021.

[26] T. Zhang, R. Ma, and J. Qiu, "A generalized NURBS interpolator using nonlinear feedrate adaptation and interpolation-error correction," *The International Journal of Advanced Manufacturing Technology*, vol. 120, no. 1, pp. 253–263, 2022.

[27] J. Sun and Y. Lin, "A real-time interpolator for NURBS curve considering velocity, acceleration, and jerk constraints," *Journal of Intelligent Manufacturing*, vol. 25, no. 2, pp. 345–356, 2014.

[28] C. C. Chen and J. Sun, "Real-time NURBS interpolator for CNC machining with axis jerk constraints," *The International Journal of Advanced Manufacturing Technology*, vol. 89, no. 9, pp. 3335–3344, 2017.

[29] R. Song, L. Zhang, and M. Wu, "Look-ahead window interval adaptive feedrate scheduling under axis constraints," *Electronics*, vol. 12, no. 14, p. 3150, 2020.

[30] S. Bai, C. Han, X. Sun, H. Zhang, and J. Jiang, "Teardrop hovering formation for elliptical orbit considering J2 perturbation," *Aerospace Science and Technology*, vol. 106, p. 106098, 2020.

[31] B. Bhattacharjee, A. Azeem, S. M. Ali, and S. K. Paul, "Development of a CNC interpolation scheme for CNC controller based on Runge-Kutta method," *International Journal of Computer Aided.* 2012.