Optimizing the numerical algorithm in Fast Constant Engagement Offsetting Method for generating 2.5D milling tool paths

Adam Jacso¹ · Tibor Szalay¹

Received: 11 September 2019 / Accepted: 12 May 2020 / Published online: 6 June 2020
 © The Author(s) 2020

Abstract

In the case of 2.5D rough milling operations, machining efficiency can significantly be increased by providing a uniform tool load. This is underpinned by the fact that uniform load has a positive effect on both tool life and machining time. Unfortunately, conventional contour-parallel tool paths are unable to guarantee uniform tool loads. However, nowadays there are some advanced path generation methods which can offer a constant tool load by controlling the cutter engagement angle. Yet, the spread of these non-equidistant offsetting methods is hindered by their dependence on complex calculations. As a solution to this problem, the Fast Constant Engagement Offsetting Method (FACEOM), developed in the scope of our previous study, is seen to be taking a step towards reducing computational needs. In this paper, suggestions for further improvements of FACEOM are presented. Decreasing the number of path points to be calculated is made possible by implementing adaptive step size and spline interpolation. Through simulation tests, it was also analysed which of the numerical methods utilized for solving boundary value problems can be applied to obtain the shortest calculation time during tool path generation. The practical applicability of the algorithm has been proved by cutting experiments. With respect to research results, this paper also describes how a tool path created by the algorithm can be adapted to controllers of CNC machine tools. Solutions presented in this paper can promote a wider application of a modern path generation method that ensures constant tool loads.

Keywords Tool path generation · Non-equidistant offsetting · Cutter engagement · Computational cost · High-speed milling

Nomenclature

a_e	effective radial immersion [mm]	s	stepover [mm]
a_p	axial depth of cut [mm]	t	free parameter of curve equations [–]
$\bar{c}(t)$	parametric representation of the workpiece contour [{mm, mm}]	$\bar{v}$	feed vector [{mm, mm}]
f_s	sampling frequency [Hz]	v_c	cutting speed [m/min]
f_z	feed per tooth [mm/tooth]	v_f	feed rate [mm/min]
i, j	step index [–]	z	number of teeth [–]
l	length of linear segments [mm]	C	entry point of the cutting edge [{mm, mm}]
n	spindle speed [1/min]	D_{tool}	tool diameter [mm]
$\bar{p}(t)$	parametric representation of the tool path [{mm, mm}]	F_{xy}	cutting force [N]
r_{tool}	tool radius [mm]	P	tool path point [{mm, mm}]
		$R(\varphi)$	rotation matrix [–]
		Q	exit point of the cutting edge [{mm, mm}]
		α	angle parameters [°]
		Δs	step size [–]
		$\Delta \varphi$	angular deviation between adjacent line segments [°]
		θ	cutter engagement angle [°]
		$\Delta \theta$	permissible inaccuracy of cutter engagement angle [°]
		ρ	radius of curvature [mm]

✉ Adam Jacso
 jacso@manuf.bme.hu

¹ Department of Manufacturing Science and Engineering, Budapest University of Technology and Economics, Budapest, Hungary

1 Introduction

Metal cutting technologies still assume great importance in part manufacturing [1]. The associated cutting process is usually divided into several stages: roughing and smoothing steps are typically separated [2]. Approximately, half of the total machining time is spent on rough cutting [3]. For this reason, numerous studies deal with this area: in the industry quality improvement [4] and improved productivity [5] are strategic goals.

The characteristics of tool path have a significant impact on machining costs [6]. Generally, even in the case of geometries with free-form surfaces, constant Z-level (2.5D) strategies are used in rough milling [7, 8]. There are two basic tool path strategies for conventional 2.5D operations: direction-parallel and contour-parallel strategies [9]. However, a common feature of these methods is that they are calculated only on a geometric basis. Therefore, they focus only on the complete removal of the resulting machining allowance [10]. For that reason, their common shortcoming is that technological aspects, such as tool load and chip formation, are not taken into account [11]. As a result, both cutting force and cutting temperature can sharply fluctuate in the case of conventional direction-parallel or contour-parallel strategies [12]. This phenomenon has a detrimental effect on tool life and machining time [13]. In this situation, varying cutting characteristics hinder the correct choice of cutting parameters during process planning [14], hamper the stability of cutting [15], and adversely affect both the machining quality and efficiency [16]. Moreover, during machining of thin walls, where tool deflection is critical [17], and during high speed machining, where pulse-like tool load can even lead to tool breakage, the fluctuation of the cutting force cannot be allowed at all. Uniform tool load is also required when strict requirements are specified for surface quality [18, 19] or shape and size accuracy [20].

To characterize the connection between the tool and the workpiece, the most efficient parameter is cutter engagement angle (θ) [21]. Both for flat end mill [22] and ball end mill, cutter engagement is involved in cutting force formulas [23].

Cutting characteristics do not show any extreme fluctuations at trochoidal tool paths; nonetheless, Li et al. have shown that even with this strategy, controlling the cutter engagement can increase machining efficiency [24]. There are also methods for providing a uniform tool load through adjusting the feed rate based on different approaches. The offline feed rate control can be based on path curvature [25] or cutter engagement [26]. In addition, there are methods where an offline pre-optimization is followed by an adaptive control based on continuous cutting force monitoring [27, 28]. However, feed rate control alone is only a partial solution, and the following problems still need to be addressed. On the one hand, the machine tool must be capable of performing

continuous decelerations/accelerations [29]. On the other hand, occasionally emerging engagement can also lead to vibrations or a thermal shock [30]. Therefore, maximum efficiency can only be achieved through path modification [31]. Taking into account the above considerations, adjusting the cutter engagement is a factor that is capable of ensuring that tool paths can provide efficient machining [32].

In the case of milling along a straight line, stepover (s) is the same as radial immersion ($a_e = s$), i.e., stepover directly determines cutter engagement. On the contrary, when the tool moves along arcs, radial immersion and stepover are different [33]. Alterations to radial immersion also change chip thickness, which is directly related to cutting force [34]. However, when constant cutter engagement is ensured, the constant feed rate also leads to a uniform tool load [35]. In turn, for the calculation of the cutter engagement, pixel-based [36] and polygon-based numerical solutions [37] can be used.

During tool path planning, several possible solutions that consider the effects of the cutter engagement can be used. For example, the conventional contour-parallel tool paths can be improved by adding trochoidal sections in critical zones [38, 39]. However, the engagement can more extensively and all-inclusively be controlled, if non-equidistant offsetting is used. The first of these methods was presented by Stori et al. [40]. Later, their method was further developed by Ibaraki et al., who followed similar considerations, but elaborated a more generally applicable method [41]. Uddin et al. showed that this method also improves the accuracy of the machined contour [42]. Since these geometry-based solutions work only in the case of simple geometries, different pixel-based solutions have also appeared [43]. Such pixel-based solutions can offer general usability, but as a disadvantage they require more computations than geometric solutions. Given this scenario, the FACEOM algorithm has been developed to combine the advantages of the two approaches: FACEOM is characterized by fast operation and general usability, including applicability in the case of complex geometries and the handling of transition sections [44]. In this paper, some suggestions are presented for further improving the computational efficiency of the FACEOM algorithm through the reduction of the number of tool path points to be determined.

Although the method presented does not have a direct impact on production efficiency, since there are cycles in CAM systems where engagement control is already implemented, the calculation time is still a critical factor. Increasing computational speed is essential for the spread of these modern solutions. Without this, it would also be challenging to apply optimization procedures that require path re-generating several times with different settings. Taking this into account, the method can indirectly help to increase production efficiency. Furthermore, it is also worth mentioning that although this paper focuses on 2.5D milling, the methods employed can

step size (Δt) has to be used. However, if equal steps are used, an unnecessarily dense resolution can result in not so curved sections. Therefore, it is expedient to use adaptive (i.e. varying) step size. If step size is adapted to the curvature of the path, computing speed and accuracy improve.

For the application of adaptive step size, a quick-to-check criterion has been defined. This criterion concerns that the angle between two adjacent segments of the calculated polyline cannot exceed a specified value. This ensures that the points will be denser on segments with a higher curvature, and sparser on segments with a lower curvature. This is illustrated in Fig. 2. In the figure, the black curve represents the reference curve, which was obtained by approximating two different methods but using the same number of base points. Shown in the same figure, the blue polyline exhibits equidistant resolution, whereas the red polyline exhibits adaptive resolution. As it can be seen, the chord error is much lower in the case of the red polyline, i.e. when adaptive resolution is used.

To implement the approach described above, the following procedure was used to supplement the algorithm. Instead of simply specifying the value of step size (Δt), three parameters are defined as input data. $\Delta\varphi_{\text{crit}}$ represents the maximal angular deviation permissible to be formed between the tangents of adjacent path segments. Besides that, it is also necessary to establish a minimum step size (Δt_{min}) and a maximum step size (Δt_{max}), between which the current step size can fluctuate. At the initialization, the maximum value for the initial step ($\Delta t = \Delta t_{\text{max}}$) is selected. Then, after calculating each new vector ($\vec{v}_{i+1}$), the angle ($\Delta\varphi_{i+1}$) between the new ($\vec{v}_{i+1}$) and the previous vectors ($\vec{v}_i$) is examined. If this value is greater than the allowable deviation $\Delta\varphi_{\text{crit}}$, then increment Δt is proportionally reduced to the magnitude of deviation (λ), and then the status $i+1$ is calculated again. It may also happen that the criterion cannot be satisfied even with a step size close to zero. In this case, the algorithm enters an infinite loop. To avoid this, a comparison with the minimum step value is made. It is advisable to set this value low, because this parameter will be needed and used only if it is not possible to generate the path. This can occur in two cases: (1) the shape of the contour does not allow for a constant cutter engagement, (2) the initial conditions are not properly selected. In such cases, the algorithm stops. However, in normal operating conditions, after a few iteration steps the step size will take the appropriate value.

Fig. 2 Application of adaptive step size

Furthermore, the algorithm should also include the determination of the next step size. After step i is accepted, based on the angle enclosed by the adjacent sections of the path, step size can be determined with the help of the following equation:

$$\Delta t_{i+1} = \frac{\Delta\varphi_{\text{crit}}}{\Delta\varphi_i} \Delta t_i$$

Thus, the value of step size Δt will follow the magnitude of the path curvature. However, it is important that step size should not exceed the maximum value, because in that scenario the numerical calculation may become unstable if the distance between the calculated point and the contour is greater than the tool radius.

By implementing these additions, the number of path segments can significantly be reduced and the accuracy of the result can also be controlled. Furthermore, the suitability and the applicability of the solution were tested experimentally, and the referent results are presented in Sect. 4 below.

3.2 Tool path smoothing using a spline curve

The FACEOM algorithm generates a sequence of points $\{P_j\}_{j=0, \dots, i}$ by way of applying a step-by-step calculation. These points refer to the base points of the tool path. If these points are linked with the help of linear interpolation (i.e. by straight lines), the tool path will be a simple polyline. In this case, the path will only have G^0 continuity. However, it is also possible to describe the tool path by fitting a higher order polynomial spline to the point sequence. When a cubic spline is used, the path will have C^2 continuity. Since the exact solution is a continuous function, the value of the resulting engagement will fluctuate to a lesser extent along the tool path on condition the tool path is also described by applying a continuous function. As a result, fewer points will have to be calculated, and this will be enough to achieve the same level of accuracy as the one obtained by using a simple polyline. Although the advanced CNC controllers are capable of automatically performing spline fitting, it is necessary to know at what density the control points should be determined. Therefore, this topic also needs to be addressed during tool path generation.

This is illustrated in Fig. 3. For the purpose of the investigations, on the reference curve marked black, four base points were selected. Then, a first-degree spline (marked in blue) and a third-degree spline (marked in red) were fitted to these base points. While the polyline is a very rough approximation, the cubic spline almost perfectly matches the original curve. Although spline fitting requires more extensive computation,

Fig. 3 Applying a cubic spline for linking the calculated points

this extra time is still less than the one that can be gained by calculating fewer base points.

In order to justify the benefits of spline fitting, experiments were also performed. The results are presented in Sect. 4 below.

3.3 Implementing different methods to determine the direction of stepping forward

The determination of the tool path tangent vector from a given point according to FACEOM is detailed in Sect. 2. However, in the scope of step 3 of the FACEOM algorithm, not only can the next step be taken in the direction of the tangent vector, but also, alternatively, it is to be considered how the curvature changes in the environment of a given point of the contour curve. This means that the next point can be defined with the help of several strategies. Such strategies, i.e. numerical solutions, are known and used in the field of boundary value problems. In the scope of this study, seven different methods were tested, including explicit and implicit, first-order (explicit Euler, implicit Euler, semi-implicit Euler), second-order (midpoint, trapezoidal), and higher-order methods (Adam-Bashforth, Runge-Kutta). In the following, symbol $\bar{v}$ will refer to a unit-length vector approximating the tangent vector.

3.3.1 Explicit Euler method

The explicit Euler—also known as the forward Euler—method is the simplest approach. This solution is the same as the basic version of FACEOM (detailed in Sect. 2). Calculating the next tool path point can be defined as follows:

1. Determine the intersection point of the half-line from point P_i along vector $\bar{v}_i$ and of the circle with a centre point C_{i+1} and a radius r_{tool} , and thereby obtain P_{i+1}

In the scope of this method, each step is obtained directly by taking a point along the tangent vector on the basis of the last tool path point. This means that it is necessary to calculate intersection points only once in the case of each step.

3.3.2 Implicit Euler method

The implicit Euler method is also a first-order method. However, in that case the steps are not taken along the tangent of the current point but along the tangent of the next point. Therefore, a fixed-point iteration has to be performed in each step:

1. Mark the tangent vector for point P_i with $\bar{v}_i^{(0)}$
2. Determine the intersection point of the half-line from point P_i along vector $\bar{v}_i^{(0)}$ and of the circle with a centre point C_{i+1} and a radius r_{tool} , and thereby obtain $P_{i+1}^{(0)}$
3. Determine tangent vector $\bar{v}_i^{(1)}$ for point $P_{i+1}^{(0)}$

4. Determine the intersection point of the half-line from point P_i along vector $\bar{v}_i^{(1)}$ and of the circle with a centre point C_{i+1} and a radius r_{tool} , and thereby obtain $P_{i+1}^{(1)}$
5. Determine the limit of the above iteration: $\lim_{n \rightarrow \infty} \bar{v}_i^{(n)} = \bar{v}_i^*$
6. Determine the intersection point of the half-line from point P_i along vector $\bar{v}_i^*$ and of the circle with a centre point C_{i+1} and a radius r_{tool} , and thereby obtain P_{i+1}

In the scope of the study, fixed-point iteration was performed only for two steps: ($P_{i+1} = P_{i+1}^{(2)}$). This did not cause a significant error because the algorithm showed fast convergence. The early termination of fixed-point iteration is also justified by the fact that calculating the tangent vector with an extreme precision significantly increases computational time. With a view to this, intersection points are calculated three times in each step.

3.3.3 Semi-implicit Euler method

The semi-implicit Euler method is a simplified version of the previous method. In the scope of this method, no fixed-point iteration is performed, but the vector obtained in the first approximation is adopted ($P_{i+1} = P_{i+1}^{(1)}$). Thus, intersection points must be calculated only twice in each step.

3.3.4 Midpoint method

The midpoint method is an advanced version of the explicit Euler method, but this algorithm already belongs to the group of second-order methods. Compared with the explicit Euler method, the midpoint method needs twice as many operations, but in return it provides second-order convergence. Taking a step can be defined as follows:

1. Substitute parameter $t_i + \Delta t/2$ in the function $\bar{c}(t)$, and thereby obtain point $C_{i+1/2}$
2. Determine the intersection point of the half-line from point P_i along vector $\bar{v}_i$ and of the circle with a centre point $C_{i+1/2}$ and a radius r_{tool} , and thereby obtain $P_{i+1/2}$
3. Determine tangent vector $\bar{v}_{i+1/2}$ for point $P_{i+1/2}$
4. Determine the intersection point of the half-line from point P_i along vector $\bar{v}_{i+1/2}$ and of the circle with a centre point C_{i+1} and a radius r_{tool} , and thereby obtain P_{i+1}

Given this, it is necessary to calculate intersection points twice in each step, but the resulting step is closer to the exact solution than in the case of the explicit Euler method.

3.3.5 Trapezoidal rule method

This algorithm also belongs to second-order methods. It is a combination of the explicit Euler and the semi-implicit Euler method. Taking a step can be defined as follows:

1. Mark the tangent vector for point P_i with $\bar{v}_i^{(0)}$
2. Determine the intersection point of the half-line from point P_i along vector $\bar{v}_i^{(0)}$ and of the circle with a centre point C_{i+1} and a radius r_{tool} , thereby obtain $P_{i+1}^{(0)}$
3. Determine tangent vector $\bar{v}_i^{(1)}$ for point $P_{i+1}^{(0)}$
4. Determine the intersection point of the half-line from point P_i along vector $\bar{v}_i^* = 1/2 (\bar{v}_i^{(0)} + \bar{v}_i^{(1)})$ and of the circle with a centre point C_{i+1} and a radius r_{tool} , thereby obtain P_{i+1}

In this scenario, it is necessary to calculate intersection points twice in each step.

3.3.6 Runge-Kutta method

The Runge-Kutta method is a general procedure. During the experiments, the classical second-order Runge-Kutta method was used. The process is as follows:

1. Mark the tangent vector for point P_i with $\bar{v}_i^{(0)}$
2. Substitute parameter $t_i + \Delta t/2$ in the function $\bar{c}(t)$, and thereby obtain point $C_{i+1/2}$
3. Determine the intersection point of the half-line from point P_i along vector $\bar{v}_i^{(0)}$ and of the circle with a centre point $C_{i+1/2}$ and a radius r_{tool} , and thereby obtain $P_{i+1/2}^{(0)}$
4. Determine tangent vector $\bar{v}_i^{(1)}$ for point $P_{i+1/2}^{(0)}$
5. Determine the intersection point of the half-line from point P_i along vector $\bar{v}_i^{(1)}$ and of the circle with a centre point $C_{i+1/2}$ and a radius r_{tool} , and thereby obtain $P_{i+1/2}^{(1)}$
6. Determine tangent vector $\bar{v}_i^{(2)}$ for point $P_{i+1/2}^{(1)}$
7. Determine the intersection point of the half-line from point P_i along vector $\bar{v}_i^{(2)}$ and of the circle with a centre point $C_{i+1/2}$ and a radius r_{tool} , and thereby obtain $P_{i+1}^{(0)}$
8. Determine tangent vector $\bar{v}_i^{(3)}$ for point $P_{i+1}^{(0)}$
9. Determine the intersection point of the half-line from point P_i along vector $\bar{v}_i^* = 1/6 (\bar{v}_i^{(0)} + 2\bar{v}_i^{(1)} + 2\bar{v}_i^{(2)} + \bar{v}_i^{(3)})$ and of the circle with a centre point C_{i+1} and a radius r_{tool} , thereby obtain P_{i+1}

Given this, it is necessary to calculate intersection points four times in each step. However, this method allows for making the most precise tracking as far as the path tangent's changes in the environment of point P_i are concerned.

3.3.7 Adam-Bashforth method

The most common representative of linear multistep methods is the Adams-Bashforth method. In the experiments, a two-step version of the method was tested. Taking a step can be defined as follows:

1. Mark the tangent vector for point P_i with $\bar{v}_i$
2. Determine the intersection point of the half-line from point P_i along vector $\bar{v}_i^* = 3/2 \bar{v}_i - 1/2 \bar{v}_{i-1}$ and of the circle with a centre point C_{i+1} and a radius r_{tool} , and thereby obtain P_{i+1}

In that case, the tangent vector of the previous point is used in each step. The first step can be made using the explicit Euler method.

4 Simulation analysis for the comparison of computational efficiency

All seven methods detailed in Sect. 3.3 were coded in Wolfram Mathematica. Each algorithm was tested with constant and adaptive step sizes, as well as with linear and cubic spline fittings. This meant a total of 28 combinations. During the simulation tests, different geometries as well as diverse cutter engagement values and accuracy limits were set.

For the comparison of the different methods, the time required to achieve a given accuracy constituted the basis. For the analysis, the numerical equation solving algorithms of Wolfram Mathematica version 11 was used; this scenario also made it possible to measure the CPU time required for the referent calculations (CPU type: Intel Core i3-3220 3.30 GHz). To check the cutter engagement, a self-developed discrete model-based simulator [11] was used. The parameters for adaptive step size were set by way of using an iterative search to obtain appropriate accuracy.

4.1 Geometries used for analysis

Three sample geometries were used during the analysis. The first two geometries were right-angled convex or concave corners, where the corners were rounded with a radius r_{tool} . In the case of the convex corner, the algorithm works well even with a zero-corner radius. However, in the case of the concave corner, a geometry at which the tool can reach the allowance had to be formed; otherwise, the criterion of constant engagement could not have been met. The third geometry was a two-period length sine wave-shaped contour. This geometry aptly represents the geometrical circumstances that can occur at path generation. This is because this curve contains convex sections, concave sections, and almost fully straight sections where the curvature is zero. This means that the most comprehensive picture of the efficiency of algorithms can be obtained when this geometry is used. These sample geometries were chosen because they made it possible to analyse the computational efficiency separately at different contour types. If a complex contour is to be machined, the computational efficiency depends on the proportion of contour types. In other

words, the results of the experiments can be generalized to arbitrary geometry.

Figure 4 shows the tool paths which provide a cutter engagement of 60° . It is noticeable that along convex sections the tool removes less material, and along concave sections the tool removes more material if a constant engagement tool path (marked in blue) is used instead of a contour-parallel tool path (marked in green). This manner of controlling the radial immersion of the tool is essential for providing a constant tool load, which is key to achieving higher machining efficiency.

4.2 The development of cutter engagement at different numerical methods

At first, a brief overview of the methods will be presented from the point of view of accuracy. Figure 5 presents how the cutter engagement has changed along the tool path for the sine wave-shaped contour shown in Fig. 4. The desired engagement was $60^\circ \pm 1^\circ$ for all the 28 variations. At this accuracy limit, the tool path shapes are almost the same, but there are differences in the evolution of engagement angle. In the diagrams, vertical grey lines indicate different parts of the contour. The first and third parts are the convex half-periods, the second and fourth parts are the concave half-periods.

It can be noticed that some methods required much more time for tool path generation. The experiments have shown that multiple calculations in the case of the implicit methods and in the case of the Adam-Bashforth method did not help to reduce the number of points to be calculated. It can also be stated that both cubic spline fitting and adaptive step size significantly reduced the number of base points and the time required for path generation. The greatest improvement was

detected in the case of the Runge-Kutta method, where both indicators showed a tenfold improvement over the basic version introduced in Sect. 2. With reference to the other methods, the results became at least twice or three times more favourable using the newly developed algorithm.

In Fig. 5, it can be observed that in the case of first-order methods, error development is asymmetric. In case of the implicit and semi-implicit Euler methods, deviation is positive along the convex parts, and negative along the concave parts, while the explicit Euler method shows opposite development. It can be considered more advantageous if deviation fluctuates symmetrically around the nominal value, since in this case it is easier to keep the symmetric accuracy limit. As a result, higher-order methods, such as the midpoint, trapezoidal rule, and Runge-Kutta methods, can meet the accuracy criteria by exhibiting significantly fewer base points. The scenario and the obtained results were similar in the case of all other geometries and parameter settings; therefore, in the following only higher-order methods and—as a reference—the explicit Euler method will be detailed. In addition, the results presented below all refer to tool paths generated by using cubic spline fitting and adaptive step size.

4.3 Computational times at different geometries

Figure 6 shows a comparative analysis, where the above-described path generation algorithms were applied for three different sample geometries. The cutter engagement was uniformly $60^\circ \pm 1^\circ$. In the case of simpler geometries, where the contour contained long straight sections, the explicit Euler method proved to be quite efficient. This is explained by the fact that in the case of straight

Fig. 4 The geometries used for analysis

Fig. 5 Cutter engagement along the sine wave-shaped contour

lines the intelligent consideration of tangent vector change yields no benefits, but—concurrently—taking a step requires more extensive calculation. However, the midpoint method proved to be as effective as the explicit Euler method. Furthermore, in the case of more complicated contour geometries, higher-order methods clearly yielded better results. The least amount of computation time was required by the Runge-Kutta method, which was followed by the midpoint method with almost the same value.

For further experiments, the sine wave-shaped geometry was used, since the real circumstances of application are best represented by this geometry.

4.4 Computational times at different cutter engagements

Comparative analyses were also performed in the case of different contact angles ($30^\circ \pm 1^\circ$, $60^\circ \pm 1^\circ$, $90^\circ \pm 1^\circ$). In all three cases, the midpoint and Runge-Kutta methods offered the

Fig. 6 Computational times at different geometries ($\theta = 60^\circ \pm 1^\circ$)

shortest calculation times (see Fig. 7). It can be observed that if the nominal value of cutter engagement increases, the computational efficiency of advanced algorithms becomes even more advantageous.

4.5 Computational times at different accuracy limits

The effect of accuracy limit is even more interesting compared with the nominal value of cutter engagement. Experiments were performed with the following parameter settings: $60^\circ \pm 0.2^\circ$, $60^\circ \pm 1^\circ$, and $60^\circ \pm 5^\circ$ (see Fig. 8).

In the case of the strictest accuracy limit, the efficiency of the explicit Euler method is far behind the other three methods. In the case of this setting, the midpoint method and the Runge-Kutta method proved to be the most advantageous. However, in the case of the lowest accuracy limit, the Runge-Kutta method required the most extensive computational time. The reason for this is that, for most of the steps, the maximum step size also provided an accuracy of $\pm 5^\circ$, so the advantage of higher-order methods requiring fewer points for calculation was this way lost. For example, when using the Runge-Kutta method, the same time was required to meet accuracy limits of 1° and 5° , because it was no longer possible to increase the maximum step size and concurrently maintain

numerical stability. As opposed to this, the midpoint method gave good results in the case of all three accuracy limits.

4.6 Results of the simulation analysis

From the comparative analysis presented above, it can be concluded that the midpoint method is the most advantageous for the calculation of tool path points. This method works efficiently whether with simple or complex geometry, with strict or loose accuracy limits, and likewise the extent of cutter engagement is of no interest, either. The Runge-Kutta method produced a shorter computational time under certain circumstances; nevertheless, in cases when it proved to be a slower method following its comparison with other methods, its disadvantages were rather significant. This is due to the fact that the midpoint method requires only two intersection point calculations per step, while the Runge-Kutta method requires four. The trapezoidal method was slower in the case of every comparison, and the explicit Euler method proved to be competitive only in the case of very simple geometries or very loose precision requirements. Because of these drawbacks, in the following it will only be focused on the midpoint method.

The experiments provided valuable insights into choosing the proper input parameters. For a maximum step size, 25–

Fig. 7 Computational times at different cutter engagements

Fig. 8 Computational times at different accuracy limits

50% of the tool diameter is recommended. In the vast majority of cases, the algorithm can operate at a greater increment, too. However, in that case multiple recalculations are required before making sharp changes in direction which undermines computational efficiency, and the numerical stability may also be compromised.

FACEOM's primary field of application is the planning of rough milling tool paths. This determines the accuracy limit to be set. It is not reasonable to use unreasonably strict constraints on the value of cutter engagement, since this no longer has any technological advantage above a certain level, but the strict constraints do increase calculation time. Similarly, it is not worth setting to low accuracy limits either, because the advantages of machining with a constant cutter engagement will be lost. The experiments also showed that an accuracy of $\pm 1^\circ$ is sufficient for ensuring a uniform tool load during machining. In the scope of the experiments, uniform tool load was achieved by adjusting the settings through iterative trials and by calculating the cutter engagement through simulation. However, adaptive step size established on the basis of the maximum angles enclosed by the adjacent tangent vectors allows for the control of accuracy without any simulation analysis. For any given geometry, a linear relationship was found to exist between the maximal angular deviation and the error of cutter engagement. Based on the experiments, it has been concluded that, as a rule, it is expedient for the allowable angular deviation $\Delta\varphi_{\text{crit}}$ to be smaller or equal to the permissible error. For example, following the iterative setting of the algorithm for the purpose of providing a cutter engagement of $60^\circ \pm 1^\circ$, the value of $\Delta\varphi_{\text{crit}}$ was $\sim 1.5^\circ$ in the case of the concave corner, and $\sim 2.4^\circ$ in the case of the convex corner. Furthermore, with respect to the sine wave-shaped geometry, the C^2 continuity of the contour allowed for even greater angular deviation, because $\sim 6.8^\circ$ was already sufficient to meet the referent requirement.

After the incorporation of the improvements presented in Sect. 3, the path generation algorithm operates according to the flow chart shown in Fig. 9.

At this point, it must also be mentioned how the algorithm behaves at exceptional workpiece profiles. During equidistant offsetting, removing of gaps and invalid loops is considered as a critical point. When using FACEOM, gaps cannot develop since the path is defined as a set of matching sections. However, the occurrence of loops has also to be considered. Similar to conventional offsetting, invalid loops can be classified into two groups: global and local loops. Global loops can develop when the contour contains such bottlenecks where the tool cannot traverse without touching a later section of the contour. This must also be taken into account when applying FACEOM, and the critical parts must be handled separately. For equidistant offset curves, a local loop occurs if the radius of curvature along the contour (ρ) is less than the offset distance which is equal to $s - r_{\text{tool}}$ in case of convex arcs, and $r_{\text{tool}} - s$ in case of concave arcs. When using FACEOM, these conditions are similar, but cannot be established so explicitly, whereas the earlier part of the contour also affects how the path develops around a given point. The presence of local loops does not stop the path generation process, but the evolution of cutter engagement should be checked and, if necessary, a local path modification may be required.

5 Practical application

In order to verify simulation results, cutting experiments were carried out. During these experiments cutting force was measured with a piezoelectric dynamometer.

5.1 Experimental conditions

With respect to cutting, the calculation of the tool path is irrelevant; given this, only the midpoint method was tested, which proved to be the most effective in terms of computational time. Experiments were performed for each of the geometries shown in Fig. 4, but since the experiments yielded the same results, the referent results are only reported for the

Fig. 9 Flow chart of the path generation algorithm

sine wave-shaped geometry. The following vector function describes the shape of this contour:

$$\vec{c}(t) = \{t, 10 \sin(2\pi t/50)\}, t \in [0, 100]$$

The same nominal cutter engagement ($\theta = 60^\circ$) was set during path generation, but different accuracy limits were ($\Delta\theta = \{\pm 0.2^\circ, \pm 0.5^\circ, \pm 1^\circ, \pm 2^\circ, \pm 5^\circ, \pm 10^\circ\}$) used for the investigation of the effects of the accuracy limits on the cutting process. The experimental conditions are shown in Table 1.

5.2 Linearization of tool path

During tool path generation, adaptive step size and cubic spline curve fitting were used. If the controller of the CNC machine tool to be used is capable of processing a spline-defined tool path, it is worthwhile to utilize this option, because this may shorten the path tacking time [48]. In many cases, however, controllers are not able to directly process tool paths created in a spline form. Below, with reference to this scenario, the issue of how to adapt the machining program to the controller is addressed.

In CAM systems, it is a general solution to approximate the paths using short straight or circular arc segments. In addition to this, there are also different methods [49] to optimize the approximation of spline curves. However, since a roughing process was investigated in the scope of the experiments, using only straight lines for the substitution of the spline has been deemed adequate. Thus, the machining program contained only G1 interpolation.

When a continuous tool path is discretized by short straight elements, choosing the right length of segments is extremely important [50]. The longer the segment is, the greater the chord error between the original and the approximating curve gets. However, if too short segments are used, the computational speed of the controller may be not adequate: if the execution of one sentence in the NC program takes more time than the completion of the motion with the programmed feed, the program run may become discontinuous. During the experiments, a segment length of $l = 0.02$ mm and a programmed feed rate of $v_f = 159$ mm/min were used. In that setup, a duration of 7.5 ms was allotted for the execution of an NC sentence, which was approximately twice the critical value of the controller.

Table 1 Experimental conditions used during cutting tests

Machine tool	KONDIA 640B 3-axis milling centre, NCT100 controller
Tool	$D_{\text{tool}} = \varnothing 6$ mm flat-end mill (Garant, item no.: 2010406), number of teeth: $z = 1$, uncoated solid carbide, W-type geometry, 26° helix
Workpiece	Al6061 aluminium alloy
Cooling	Flood cooling
Spindle speed	$n = 6366$ rpm ($v_c = 120$ m/min)
Feed rate	$v_f = 159$ mm/min ($f_z = 0.025$ mm/tooth)
Nominal cutter engagement	$\theta = 60^\circ$
Accuracy of cutter engagement	$\Delta\theta = \{\pm 0.2^\circ, \pm 0.5^\circ, \pm 1^\circ, \pm 2^\circ, \pm 5^\circ, \pm 10^\circ\}$
Nominal stepover	$s = 0.25$ $D_{\text{tool}} = 2.5$ mm
Axial depth of cut	$a_p = 6$ mm
Measuring device	Kistler 9257B-type piezoelectric dynamometer
Measuring software	LabVIEW, National Instruments
Sampling frequency	$f_s = 8000$ Hz (120 measured points per tool revolution)

Before performing the cutting experiments, the effect of discretization on the theoretical value of cutter engagement was also examined. In the scope of the approximation, from a C^2 continuous curve a polyline exhibiting only G^0 continuity is generated. The resulting deviation of the engagement is shown in Fig. 10. It can be seen that the original and the approximated paths produced almost the same results.

When the NC code of continuous cutting mode is on (G64), the tool moves along a path where the corners of the polyline are rounded. In addition, the path tracking error can also increase the fluctuation of cutter engagement. However, the effects of the above are negligible compared with the other factors interfering with the cutting process. Thus, it can be stated that the tool path linearization does not significantly change cutter engagement.

5.3 Results of the experiment

During the cutting experiments, the force acting on the workpiece was measured with a piezoelectric dynamometer. In the

Fig. 10 Effect of path linearization on cutter engagement

scope of the evaluation of the results of the experiment, the projection of the cutting force on the machining plane was calculated first, and then the development of maximum forces at each tool revolution was examined. For processing the data, Gaussian filter was used. The results are shown in Fig. 11.

Before machining the wave-shaped contour, a reference measurement was performed, where a straight line-shaped contour was machined with a 25% stepover, which provides a cutter engagement of 60° . In the scope of this measurement, all the other cutting parameters were the same. The result is shown in Fig. 11a. As for the results, cutting force changed from 96 to 113 N, and the resulting fluctuation of $\pm 8\%$ is a combined effect of the dynamic characteristic of the milling process and external distractions. Given this, a smaller fluctuation than the above cannot be expected from the elaborated algorithm in the case of the wave-shaped contour, either.

As another reference measurement, a measurement where the wave-shaped contour was machined with a 25% constant stepover was also performed. Figure 11b shows that the cutting force was significantly reduced at convex curves, while it was significantly increased at concave curves. Due to intense changes in cutter engagement angle ($50^\circ \leq \theta \leq 74^\circ$), the force fluctuation rate for this tool path was $\pm 34\%$.

The remaining diagrams in Fig. 11 show the results obtained with tool paths created with the FACEOM algorithm. It can be observed that the results were nearly the same for $\Delta\theta = \pm 1^\circ$ and for even stricter accuracy criteria than that. In these cases, force fluctuation was around $\pm 11\%$. In addition, a larger inaccuracy limit was applied also in the case of force measurement. In the case of $\Delta\theta = \pm 2^\circ$, force fluctuation was $\pm 13\%$; in case of $\Delta\theta = \pm 5^\circ$, it was $\pm 15\%$; and in case of $\Delta\theta = \pm 10^\circ$, it

Fig. 11 Cutting forces measured during the experiments

was $\pm 30\%$, the last scenario was barely more favourable than the constant stepover strategy.

In summary, it can be concluded that the cutting experiments have proved the proper operation of the algorithm. In addition, they have also proved the assumption that too strict tolerances should not be set. Moreover, as suggested in Sect. 4.6, the value of $\Delta\varphi_{\text{crit}}$ should be set so that the value of $\Delta\theta$ will be about $1\text{--}2^\circ$.

6 Conclusions

The FACEOM algorithm can easily and speedily generate tool paths which can provide uniform tool loads for 2.5D contour milling operations. The method behind FACEOM is based on the following principle: this newly elaborated geometric method avoids complicated intersection point calculations during path generation. The algorithm used for FACEOM determines the path from point to point in line with the desired accuracy requirements in question. Further developments of FACEOM presented in this paper are suitable for reducing the number of points to be calculated.

Seven different numerical methods were tested for tool path generation including both explicit and implicit, first-order (the explicit Euler, the implicit Euler, the semi-implicit Euler), second-order (the midpoint and the trapezoidal), and higher-order (the Adam-Bashforth and the Runge-Kutta) methods. With the help of generating tool paths for three sample geometries with different cutter engagement values and accuracy limits, the algorithms were investigated as to their effectiveness in terms of computational time. The suitability of the algorithm was validated by cutting experiments. In addition, the paper also described how the tool path created by the algorithm can be adapted to CNC controllers.

The results can be summarized as follows:

- The following led to an increase in computational efficiency: the application of adaptive step size (rather than using a uniform step size), and, during the process of linking the calculated points, the application of a cubic spline (instead of a polyline);
- The comparative analysis showed that the midpoint method is preferential to be used, because it works efficiently for simple and complex geometries, for different cutter engagement values, and for strict and loose accuracy limits alike.
- The allowable angular deviation between the tangents of adjacent path segments ($\Delta\varphi_{\text{crit}}$) should be smaller or equal than the permissible inaccuracy of cutter engagement angle ($\Delta\theta$): experiments have shown that this is sufficient to meet the requirements.

- The permissible inaccuracy of cutter engagement angle ($\Delta\theta$) should be set about $1\text{--}2^\circ$: experiments have shown that applying a stricter limit would no longer have any practical advantage so that the calculation time would increase aimlessly.
- The tool path linearization does not significantly change cutter engagement.

It can be stated that by optimizing the numerical calculations within the FACEOM algorithm, a further reduction of computation time can be attained while retaining the general applicability of the method.

Funding information Open access funding provided by Budapest University of Technology and Economics (BME). The research reported in this paper has been supported by the National Research, Development and Innovation Fund of Hungary (TUDFO/51757/2019-ITM, Thematic Excellence Program). Research work for this paper was partly supported by the European Commission through project H2020 EPIC under grant no. 739592. The results introduced in this paper are applied in project no. ED_18-22018-0006, whose project has been implemented with support provided by the National Research, Development and Innovation Fund of Hungary, and was financed in the scope of (publicly funded) funding schemes according to Section 13(2) of the Hungarian Act on Scientific Research, Development and Innovation.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Comak A, Altintas Y (2017) Mechanics of turn-milling operations. *Int J Mach Tools Manuf* 121:2–9. <https://doi.org/10.1016/j.ijmachtools.2017.03.007>
2. Childs T, Maekawa K, Obikawa T, Yamane Y (2000) Metal machining: theory and applications. Elsevier
3. Abdullah H, Ramli R, Wahab DA (2017) Tool path length optimisation of contour parallel milling based on modified ant colony optimisation. *Int J Adv Manuf Technol* 92(1–4):1263–1276. <https://doi.org/10.1007/s00170-017-0193-5>
4. Korosec M, Kopac J (2007) Neural network based selection of optimal tool - path in free form surface machining. *J Autom Mob Robot Intell Syst* 1(4):41–50
5. Car Z, Mikac T, Veza I (2006) Utilization of GA for optimization of tool path on a 2D surface, vol. 6th International Workshop on Emergent Synthesis, pp. 231–236
6. Karuppanan BRC, Saravanan M (2019) Optimized sequencing of CNC milling toolpath segments using metaheuristic algorithms. *J*

- Mech Sci Technol 33(2):791–800. <https://doi.org/10.1007/s12206-019-0134-3>
7. Miko B (2012) Study of z-level finishing milling strategy. *Dev Mach Technol Crac* 83–90
 8. Chen L, Li Y, Tang K (2018) Variable-depth multi-pass tool path generation on mesh surfaces. *Int J Adv Manuf Technol* 95(5–8): 2169–2183. <https://doi.org/10.1007/s00170-017-1367-x>
 9. Held M, de Lorenzo S (2018) On the generation of spiral-like paths within planar shapes. *J Comput Des Eng* 5(3):348–357. <https://doi.org/10.1016/j.jcde.2017.11.011>
 10. Patel DD, Lalwani DI (2017) Quantitative comparison of pocket geometry and pocket decomposition to obtain improved spiral tool path: a novel approach. *J Manuf Sci Eng* 139(3):031020–031020–10. <https://doi.org/10.1115/1.4034896>
 11. Jacso A, Szalay T, Jauregui JC, Resendiz JR (2018) A discrete simulation-based algorithm for the technological investigation of 2.5D milling operations. *Proc Inst Mech Eng Part C J Mech Eng Sci*, pp 78–90, 0. <https://doi.org/10.1177/0954406218757267>
 12. Adesta EYT, Hamidon R, Riza M, Alrashidi RFFA, Alazemi AFFS (2018) Investigation of tool engagement and cutting performance in machining a pocket. *IOP Conf Ser Mater Sci Eng* 290:012066. <https://doi.org/10.1088/1757-899X/290/1/012066>
 13. Shixiong W, Zhiyang L, Chengyong W, Suyang L, Wei M (2018) Tool wear of corner continuous milling in deep machining of hardened steel pocket. *Int J Adv Manuf Technol* 97:1–19. <https://doi.org/10.1007/s00170-018-1994-x>
 14. Kao Y-C, Lin D-M, Wu J-Z, Vi T-K (2018) An integrated smarter cutting parameter selection system with a case study for pocket milling. *Int J Autom Smart Technol* 8(2):89–97–97. <https://doi.org/10.5875/ausmt.v8i2.1681>
 15. Pérez-Canales D, Álvarez-Ramírez J, Jáuregui-Correa JC, Vela-Martínez L, Herrera-Ruiz G (2011) Identification of dynamic instabilities in machining process using the approximate entropy method. *Int J Mach Tools Manuf* 51(6):556–564. <https://doi.org/10.1016/j.ijmachtools.2011.02.004>
 16. Cheng K (ed) (2009) *Machining dynamics: fundamentals, applications and practices*. Springer-Verlag, London
 17. Feng J, Wan M, Gao T-Q, Zhang W-H (2018) Mechanism of process damping in milling of thin-walled workpiece. *Int J Mach Tools Manuf* 134:1–19. <https://doi.org/10.1016/j.ijmachtools.2018.06.001>
 18. Wojciechowski S, Wiackiewicz M, Krolczyk GM (2018) Study on metrological relations between instant tool displacements and surface roughness during precise ball end milling. *Measurement* 129: 686–694. <https://doi.org/10.1016/j.measurement.2018.07.058>
 19. Twardowski P, Wojciechowski S, Wiciorowski M, Mathia T (2011) Surface roughness analysis of hardened steel after high-speed milling. *Scanning* 33(5):386–395. <https://doi.org/10.1002/sca.20274>
 20. Pimenov DY, Guzeev VI, Krolczyk G, Mia M, Wojciechowski S (2018) Modeling flatness deviation in face milling considering angular movement of the machine tool system components and tool flank wear. *Precis Eng* 54:327–337. <https://doi.org/10.1016/j.precisioneng.2018.07.001>
 21. Zhang X, Zhang J, Zhao W (2016) A new method for cutting force prediction in peripheral milling of complex curved surface. *Int J Adv Manuf Technol* 86(1–4):117–128. <https://doi.org/10.1007/s00170-015-8123-x>
 22. Shi K, Liu N, Wang S, Ren J (2019) Effect of tool path on cutting force in end milling. *Int J Adv Manuf Technol*. <https://doi.org/10.1007/s00170-019-04120-3>
 23. Zhu K, Zhang Y (2017) Modeling of the instantaneous milling force per tooth with tool run-out effect in high speed ball-end milling. *Int J Mach Tools Manuf* 118–119:37–48. <https://doi.org/10.1016/j.ijmachtools.2017.04.001>
 24. Li Z, Xu K, Tang K (2019) A new trochoidal pattern for slotting operation. *Int J Adv Manuf Technol* 102(5):1153–1163. <https://doi.org/10.1007/s00170-018-2947-0>
 25. Farouki RT, Manjunathaiah J, Nicholas D, Yuan G-F, Jee S (1998) Variable-feedrate CNC interpolators for constant material removal rates along Pythagorean-hodograph curves. *Comput Aided Des* 30(8):631–640. [https://doi.org/10.1016/S0010-4485\(98\)00020-7](https://doi.org/10.1016/S0010-4485(98)00020-7)
 26. Wei Z, Wang M, Han X (2010) Cutting forces prediction in generalized pocket machining. *Int J Adv Manuf Technol* 50(5):449–458. <https://doi.org/10.1007/s00170-010-2528-3>
 27. Zuperl U, Cus F, Reibenschuh M (2012) Modeling and adaptive force control of milling by using artificial techniques. *J Intell Manuf* 23(5):1805–1815. <https://doi.org/10.1007/s10845-010-0487-z>
 28. Zhang Z, Luo M, Zhang D, Wu B (2018) A force-measuring-based approach for feed rate optimization considering the stochasticity of machining allowance. *Int J Adv Manuf Technol* 97:1–12. <https://doi.org/10.1007/s00170-018-2127-2>
 29. Pateloup V, Duc E, Ray P (2004) Corner optimization for pocket machining. *Int J Mach Tools Manuf* 44(12–13):1343–1353. <https://doi.org/10.1016/j.ijmachtools.2004.04.011>
 30. Xu J, Sun Y, Zhang X (2012) A mapping-based spiral cutting strategy for pocket machining. *Int J Adv Manuf Technol* 67(9–12):2489–2500. <https://doi.org/10.1007/s00170-012-4666-2>
 31. Desai KA, Rao PVM (2016) Machining of curved geometries with constant engagement tool paths. *Proc Inst Mech Eng Part B J Eng Manuf* 230(1):53–65. <https://doi.org/10.1177/0954405415616787>
 32. Guerrero-Villar F, Dorado-Vicente R, Romero-Carrillo P, López-García R, Mercado-Colmenero J (2015) Computation of instantaneous cutter engagement in 2.5D pocket machining. *Procedia Eng* 132:464–471. <https://doi.org/10.1016/j.proeng.2015.12.520>
 33. Kramer TR (1992) Pocket milling with tool engagement detection. *J Manuf Syst* 11(2):114–123. [https://doi.org/10.1016/0278-6125\(92\)90042-E](https://doi.org/10.1016/0278-6125(92)90042-E)
 34. Biró I, Szalay T, Geier N (2018) Effect of cutting parameters on section borders of the empirical specific cutting force model for cutting with micro-sized uncut chip thickness. *Procedia CIRP* 77: 279–282. <https://doi.org/10.1016/j.procir.2018.09.015>
 35. Jacso A, Szalay T (2018) Analysing and optimizing 2.5D circular pocket machining strategies. *Lect Notes Mech Eng* (201519):355–364. https://doi.org/10.1007/978-3-319-68619-6_34
 36. Wang H, Jang P, Stori JA (2005) A metric-based approach to two-dimensional (2D) tool-path optimization for high-speed machining. *J Manuf Sci Eng* 127(1):33–48. <https://doi.org/10.1115/1.1830492>
 37. Gong X, Feng H-Y (2016) Cutter-workpiece engagement determination for general milling using triangle mesh modeling. *J Comput Des Eng* 3(2):151–160. <https://doi.org/10.1016/j.jcde.2015.12.001>
 38. Ibaraki S, Yamaji I, Matsubara A (2010) On the removal of critical cutting regions by trochoidal grooving. *Precis Eng* 34(3):467–473. <https://doi.org/10.1016/j.precisioneng.2010.01.007>
 39. Deng Q, Mo R, Chen ZC, Chang Z (2018) A new approach to generating trochoidal tool paths for effective corner machining. *Int J Adv Manuf Technol* 95(5–8):3001–3012. <https://doi.org/10.1007/s00170-017-1353-3>
 40. Stori JA, Wright PK (2000) Constant engagement tool path generation for convex geometries. *J Manuf Syst* 19(3):172–184. [https://doi.org/10.1016/S0278-6125\(00\)80010-2](https://doi.org/10.1016/S0278-6125(00)80010-2)
 41. Ibaraki S, Ikeda D, Yamaji I, Matsubara A, Kakino Y, Nishida S (2004) Constant engagement tool path generation for two-dimensional end milling, vol. 2004 Japan-USA Symposium on Flexible Automation
 42. Uddin MS, Ibaraki S, Matsubara A, Nishida S, Kakino Y (2006) Constant engagement tool path generation to enhance machining accuracy in end milling. *JSME Int J Ser C Mech Syst Mach Elem Manuf* 49(1):43–49. <https://doi.org/10.1299/jsmec.49.43>
 43. Dumitrache A, Borangiu T (2012) IMS10-image-based milling toolpaths with tool engagement control for complex geometry.

- Eng Appl Artif Intell 25(6):1161–1172. <https://doi.org/10.1016/j.engappai.2011.09.026>
44. Jacso A, Matyasi G, Szalay T (2019) The fast constant engagement offsetting method for generating milling tool paths. *Int J Adv Manuf Technol*. <https://doi.org/10.1007/s00170-019-03834-8>
 45. Guo ML, Wei ZC, Wang MJ, Li SQ, Liu SX (2018) Force prediction model for five-axis flat end milling of free-form surface based on analytical CWE. *Int J Adv Manuf Technol* 99(1):1023–1036. <https://doi.org/10.1007/s00170-018-2480-1>
 46. Zhang X, Zhang J, Zheng X, Pang B, Zhao W (2017) Tool orientation optimization of 5-axis ball-end milling based on an accurate cutter/workpiece engagement model. *CIRP J Manuf Sci Technol*. <https://doi.org/10.1016/j.cirpj.2017.06.003>
 47. Póka G, Németh I (2019) The effect of radial rake angle on chip thickness in the case of face milling. *Proc Inst Mech Eng Part B J Eng Manuf*. <https://doi.org/10.1177/0954405419849245>
 48. Msaddek EB, Bouaziz Z, Baili M, Dessein G (2014) Influence of interpolation type in high-speed machining (HSM). *Int J Adv Manuf Technol* 72(1–4):289–302. <https://doi.org/10.1007/s00170-014-5652-7>
 49. Maier G (2014) Optimal arc spline approximation. *Comput Aided Geom Des* 31(5):211–226. <https://doi.org/10.1016/j.cagd.2014.02.011>
 50. Medina-Sánchez G, Torres-Jimenez E, Lopez-Garcia R, Dorado-Vicente R (2017) Cutting time in pocket machining for different tool-path approximation segments. *Procedia Manuf* 13:59–66. <https://doi.org/10.1016/j.promfg.2017.09.009>

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