

Reparameterization of ruled surfaces: toward generating smooth jerk-minimized toolpaths for multi-axis flank CNC milling

Ali Hashemian^{*a}, Pengbo Bo^b, Michael Bartoň^{a,c}

^aBCAM – Basque Center for Applied Mathematics, Bilbao, Basque Country, Spain

^bSchool of Computer Science and Technology, Harbin Institute of Technology, Weihai, China

^cIkerbasque – Basque Foundation for Sciences, Maria Diaz de Haro 3, 48013 Bilbao, Basque Country, Spain

Abstract

This paper presents a novel jerk minimization algorithm in the context of multi-axis flank CNC machining. The toolpath of the milling axis in a flank milling process, a ruled surface, is reparameterized by a B-spline function, whose control points and knot vector are unknowns in an optimization-based framework. The total jerk of the tool's motion is minimized, implying the tool is moving as smooth as possible, without changing the geometry of the given toolpath. Our initialization stage stems from measuring the ruling distance metric (RDM) of the ruled surface. We show on several examples that this initialization reliably finds close initial guesses of jerk-minimizers and is also computationally efficient. The applicability of the presented approach is illustrated by some practical case studies.

Key words: Jerk-minimized toolpath; B-spline reparameterization; ruled surface; ruling distance metric (RDM); multi-axis flank CNC milling.

1. Introduction

Multi-axis computer numerically controlled (CNC) machining is the leading subtractive manufacturing technology, especially for hard materials such as titanium or steel-nickel alloys. The machining process has typically three stages: (i) roughing, that quickly removes the waste material, e.g. using a large ripper cutter, and generates a rough shape, (ii) semi-finishing, that is mostly realized by end milling with a smaller cutter than for roughing, and (iii) finishing, that aims at finalizing the shape of the object, reaching fine machining tolerances (e.g. for engine/aeronautical components a few micrometers for objects of the size of tens of centimeters). Our research focuses on this final (finishing) stage of machining. In particular, we consider 5-axis *flank milling* where the milling cutter (aka tool) and the to-be-manufactured object touch along a whole curve (i.e., grazing curve) [1], in contrast to (i) and (ii) where the tool has (theoretically) a single-point contact with the material block.

There are many factors that influence quality of a machined product: from geometric accuracy of the path-planning algorithms, over analysis of cutting forces and calibration of the actual machine, up to the error associated to the tool wear. In this work, we discuss the kinematic part of the path planning process that deals with the physics (velocities, acceleration, and jerks) of the tool [2]. Having a smooth and jerk-minimized profile in multi-axis machining is fundamental as it affects tracking performance and lowers deviation of the machined surface from its (ideal) designed state. Moreover, high values of jerk (as the time derivative of acceleration) are closely related to sudden changes of drive torques which induce undesirable vibrations [3]. Last but not least, small jerk reduces the dynamic cutting forces that, in a consequence, leads to lower energy consumption of the whole milling process.

1.1. Previous Work

Most of research on jerk minimization relates to contour machining [4–7]. Zhang et al. [4] use a G^4 -continuous B-spline curve for the toolpath and employ a jerk-smooth feedrate scheduling process in order to obtain a smooth jerk profile along the toolpath. By using an optimal control problem formulation, Bosetti and Bertolazzi [5] present an algorithm for CNC kernels that aims at solving the axes interpolation problem. They assume a predefined path tracking tolerance and apply constraints on the kinematic specifications of the machine axes to calculate the trajectory that satisfies the given constraints.

Jahanpour and Alizadeh [6] present an adaptive acceleration–jerk-limited NURBS interpolation method based on an optimized S-shaped C^2 quintic feedrate planning scheme. They modify the feedrate profile for each sharp corner to minimize the total execution time. Using a piecewise constant approximation of the third derivative of the path, Zhang et al. [7] perform a time-optimal tool trajectory generation for CNC manufacturing systems. In their approach, the axis jerk constraints are also introduced into the problem where the desired smoothness of the path could be accomplished by adjusting the values of constraints.

Beudaert et al. [8] use 5-axis toolpath smoothing techniques in order to maximize the feedrate and to reduce the machining time in both end and flank milling processes. Based on the velocity, acceleration, and jerk limits of each drive, they compute the maximum reachable feedrate which is then used to localize the areas where the toolpath is smoothed. In a similar research, Sencer et al. [2] present a feed optimization process to minimize the machining time based on the drive constraints. They use a B-spline model for the feedrate profile and find its unknown control points as design variables of an optimization process.

An optimization method for obtaining a minimum time feedrate profile that takes into account the kinematic characteristics of a particular machine is proposed by Beudaert et al. [9]. They consider both tangential and axis jerk and, in an iterative algorithm, compute the minimum time feedrate profile by simultane-

*Corresponding author

Email address: ahashemian@bcamath.org (Ali Hashemian)

ously satisfying the drive constraints.

A feedrate scheduling method for the 5-axis machining with geometry, process, and drive constraints is presented by Sun et al. [10]. The initial feedrate profile is constructed using a chord error, and then iteratively adjusted by a curve evolution strategy.

Hu and Tang [11] consider dynamic behavior of the milling tool in the context of high-speed machining. A tool path is generated such that the maximal angular accelerations of the rotary axes are reduced. By defining smoothing blocks on the toolpath of CNC contour machining, Essid et al. [12] perform a local toolpath smoothing. In their kinematic model, they took into account the drive parameter axes defined by the manufacturer of the CNC machine, however, in contrast to our work, the milling path is being modified. In a closely related research, Song and Ma [13] perform an interval partition-based feedrate scheduling with axial drive constraints for the 5-axis flank machining. Ma et al. [14] use a dual NURBS interpolator to detect feedrate-sensitive and nonsensitive regions and design constant and smooth transition speeds in these regions.

Huang et al. [15] present a real-time feedrate planning method for 5-axis machining to synchronize the linear and angular trajectories, in order to reach smooth linear and angular motions considering axial kinematic constraints. Recently, Weng and Kuo [16] have proposed a jerk decision making process that synchronize the path planning stage with the dynamic response for a 5-axis machine tool. Such a synchronization results in better surface finish and also shortens the machining time.

Another relevant family of research deals with optimization of tool orientations. Farouki and Li [17] consider a tool-orientation problem in the context of 5-axis ball-end milling and look for rotation-minimizing tool motions that preserve the angle between the tool axis and the surface normal. Sun et al. [18] also propose a tool orientation adjustment method for 5-axis ball-end machining to obtain an optimized toolpath considering the kinematic constraints. For the given toolpath and feed profile of the tool-end (both expressed as B-splines), they find analytic relations of angular feed, acceleration, and jerk with respect to the geometric and tangential feed parameters of the tool-end trajectory. Afterwards, conditional inequalities of the kinematic constraints are applied to adjust the tool orientation. Given a contact point, Sharma et al. [19] consider a multi-point tool positioning and propose a method to adapt the tool to have a multiple contact with the reference geometry.

Finally, and probably in the closest research to ours, focusing on freeform curves expressed by B-spline and/or NURBS representation, Hashemian et al. [20] present a NURBS reparameterization technique to minimize the jerk value in trajectories of robot manipulators. They reparameterize a single contact path (i.e., curve) and use a quintic B-spline function with uniform knot vector for relating the path parameter to the execution time variable.

1.2. Contributions

Our main contributions are as follows:

- We propose utilizing the *reparameterization* technique upon which the relationship between the motion path parameter of the flank machining toolpath and the system time variable is expressed by means of an optimal transfer function such that the total jerk of the tool's motion is minimized. To this end, we use a B-spline transfer function whose control points and the knot vector are variables in an optimization-based framework.

- We introduce a new initialization strategy using a proper *ruling distance metric* (RDM) to adequately estimate distances between neighboring positions of the milling tool. This initialization serves as a good guess for global optimization in a discretization of the space of transfer functions (reparameterizations).
- Given a milling toolpath for multi-axis flank milling, represented as a B-spline ruled surface, our proposed research preserves the geometric accuracy and optimizes the motion of the tool. Our results can therefore be easily combined with any toolpath design strategies.
- We compare our results with state of the art path planning techniques and show on several case studies that the reparameterized paths reduce the total jerk considerably.

The remainder of this paper is organized as follows. Section 2 discusses ruled surfaces as toolpaths in flank machining and introduces basic notions of the kinematics of the tool and reparameterization. In Section 3, the toolpath jerk minimization problem is formulated and a ruling distance metric is introduced, followed by various case studies and numerical examples in Section 4. Finally, Section 5 draws the conclusions of this work.

2. Ruled surfaces as toolpaths in flank CNC milling

The tool motion in a flank CNC milling process can be represented by the position of the tool-end and the direction of the tool. Therefore, the toolpath is defined as the locus of the milling tool axis in the Euclidean 3-space, which is a ruled surface as shown in Fig. 1.

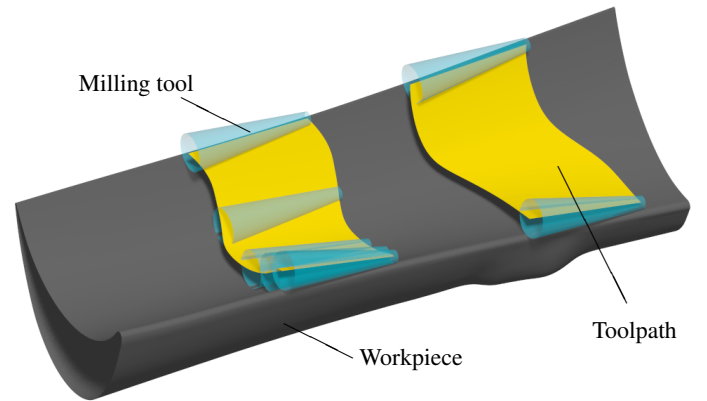


Figure 1: Toolpaths of 5-axis flank milling with conical tools, represented by ruled surfaces (yellow). The envelopes of the tool (transparent) accurately approximate a benchmark freeform workpiece (the path-planning results are courtesy of Bo et al. [21]). On the left path, observe distribution of the milling tools that is uniform in time, yet quite non-uniform in space, and would require higher acceleration and/or jerk at the upper end of the toolpath.

2.1. Ruled surfaces as toolpaths of rigid axes

For the sake of completeness, we begin with the definition of a B-spline curve that can be thought of as the locus of the tool-end positions in space. A B-spline space curve

$\mathbf{c}(u) : u \in [0, 1] \rightarrow \mathbb{R}^3$ of degree n with $N + 1$ control points $\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_N$ is defined as:

$$\mathbf{c}(u) = \sum_{i=0}^N B_i^n(u) \mathbf{p}_i, \quad (1)$$

where the univariate parameter space $u \in [0, 1]$ is characterized by the knot vector \mathbf{u} ,

$$\mathbf{u} = [\underbrace{0, 0, \dots, 0}_{n+1}, u_{n+1}, u_{n+2}, \dots, u_N, \underbrace{1, 1, \dots, 1}_{n+1}], \quad (2)$$

and the B-spline basis functions $B_i^n(u)$ are expressed by the Cox-deBoor recursion formula as follows [22]:

$$B_i^0(u) = \begin{cases} 1 & u_i \leq u < u_{i+1} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

$$B_i^n(u) = \frac{u - u_i}{u_{i+n} - u_i} B_i^{n-1}(u) + \frac{u_{i+n+1} - u}{u_{i+n+1} - u_{i+1}} B_{i+1}^{n-1}(u).$$

Consider two B-spline curves $\mathbf{c}_1(u)$ and $\mathbf{c}_2(u)$ (referred to as *boundary curves*) defined by Eq. (1), that is, $\mathbf{c}_k(u) = \sum_{i=0}^N B_i^n(u) \mathbf{p}_i^k$ ($k = 1, 2$). We assume that $\mathbf{c}_1(u)$ and $\mathbf{c}_2(u)$ are of the same degree n and spanned over the same knot vector \mathbf{u} . A ruled surface $\mathbf{s}(u, v)$ is then defined as

$$\mathbf{s}(u, v) = (1 - v) \mathbf{c}_1(u) + v \mathbf{c}_2(u), \quad [u, v] \in [0, 1] \times [0, 1], \quad (4)$$

where u is the parameter controlling the tool motion (i.e., pseudo-time parameter) and v controls the position of the point on the axis. The boundary curves are linearly interpolated and, therefore, we have a straight line segment (referred to as a *ruling*) in the v direction connecting the respective points on the curves $\mathbf{c}_1(u)$ and $\mathbf{c}_2(u)$. If the boundary curves have different knot vectors, the standard knot refinement algorithm is used to bring them to a common \mathbf{u} (see, e.g., [22]).

Remark 1. Alternatively, one can define a ruled surface by one boundary curve and the motion of the ruling vector. This is a more typical description of a ruled surface in scenarios such as ball-end milling where the contact paths are given and the orientations are used, e.g., for collision avoidance. However, these two representations of ruled surfaces are equivalent and one can easily convert one to another.

In our context, the ruled surface is considered as a toolpath of the milling axis in flank machining, that is, the rulings representing the tool axis have to be of equal length (see Fig. 2). This constraint is formulated as

$$\|\mathbf{c}_1(u) - \mathbf{c}_2(u)\| = \text{const.}, \quad \text{for all } u \in [0, 1], \quad (5)$$

and its differentiation with respect to u gives

$$\langle \mathbf{c}'_1(u), \mathbf{c}_1(u) - \mathbf{c}_2(u) \rangle = \langle \mathbf{c}'_2(u), \mathbf{c}_1(u) - \mathbf{c}_2(u) \rangle, \quad (6)$$

where $\langle \cdot, \cdot \rangle$ is the Euclidean scalar product. Eq. (6) is the first-order length-preserving constraint, also known as the *projection rule* (see, e.g., [23, 24]).

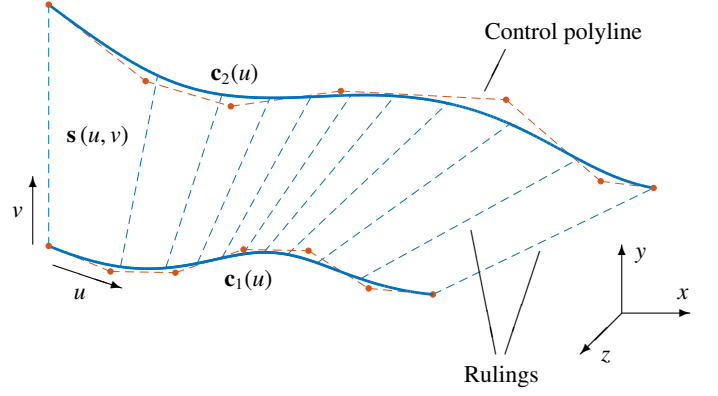


Figure 2: A ruled surface $\mathbf{s}(u, v)$ with boundary curves $\mathbf{c}_1(u)$ and $\mathbf{c}_2(u)$ and straight lines (i.e., rulings) of equal length (herein and in the following, u and v are referred to as motion and ruling parameters, respectively).

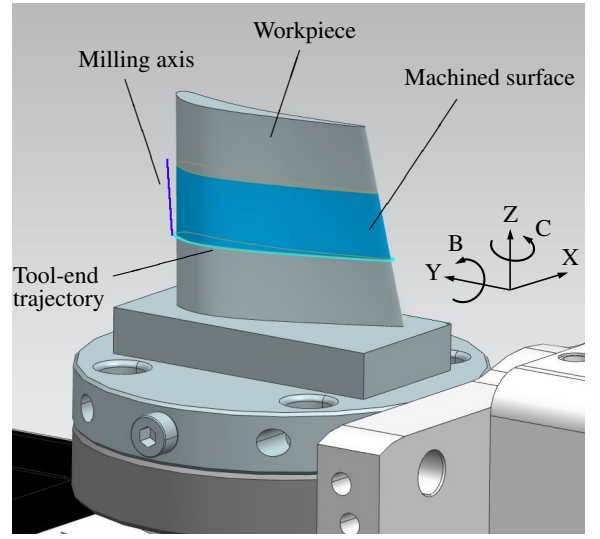


Figure 3: Simulation of flank milling in a table-tilting 5-axis CNC machine. The motions spanned by the X and Y directions are provided by translational movements of the base (i.e., the table), while the Z motion is provided by vertical translation of the spindle. The rotational movements B and C are controlled by rotations of the table around Y and Z directions, respectively.

2.2. Kinematics of the tool motion

The tool motion in the multi-axis CNC machining centers is represented in terms of G-codes and governed by the combination of movements provided by the drives of multiple axes (e.g., X, Y, Z, B, and C in the table-tilting 5-axis milling, see Fig. 3). Considering that the toolpath is given by a B-spline ruled surface, regardless of the machine type and how the motion law is governed by the G-codes, the kinematic specification of the tool motion can be represented by means of velocity (\mathbf{V}), acceleration (\mathbf{A}), and jerk (\mathbf{J}) profiles of boundary curves. These entities are computed as time derivatives of $\mathbf{c}_k(u)$ ($k = 1, 2$) using a chain rule as follows

$$\mathbf{V}_k(t) = \frac{\partial \mathbf{c}_k}{\partial t} = \frac{\partial \mathbf{c}_k}{\partial u} \frac{\partial u}{\partial t}, \quad (7)$$

$$\mathbf{A}_k(t) = \frac{\partial^2 \mathbf{c}_k}{\partial t^2} = \frac{\partial^2 \mathbf{c}_k}{\partial u^2} \left(\frac{\partial u}{\partial t} \right)^2 + \frac{\partial \mathbf{c}_k}{\partial u} \frac{\partial^2 u}{\partial t^2}, \quad (8)$$

$$\mathbf{J}_k(t) = \frac{\partial^3 \mathbf{c}_k}{\partial t^3} = \frac{\partial^3 \mathbf{c}_k}{\partial u^3} \left(\frac{\partial u}{\partial t} \right)^3 + 3 \frac{\partial^2 \mathbf{c}_k}{\partial u^2} \frac{\partial^2 u}{\partial t^2} \frac{\partial u}{\partial t} + \frac{\partial \mathbf{c}_k}{\partial u} \frac{\partial^3 u}{\partial t^3}, \quad (9)$$

where $t \in [0, \tau]$ is the system time variable.

In the above equations, the r -th derivative of B-splines curves are obtained as

$$\frac{\partial^r \mathbf{c}_k}{\partial u^r} = \sum_{i=0}^N \frac{\partial^r}{\partial u^r} B_i^n(u) \mathbf{P}_i^k, \quad (10)$$

where

$$\begin{aligned} \frac{\partial^r}{\partial u^r} B_i^n(u) &= \frac{n}{u_{i+n} - u_i} \frac{\partial^{r-1}}{\partial u^{r-1}} B_i^{n-1}(u) \\ &\quad - \frac{n}{u_{i+n+1} - u_{i+1}} \frac{\partial^{r-1}}{\partial u^{r-1}} B_{i+1}^{n-1}(u). \end{aligned} \quad (11)$$

The kinematic specifications of boundary curves provided by Eqs. (7–9) will then be used to measure and optimize the smoothness of the tool's motion.

2.3. B-spline reparameterization of the toolpath

When the flank milling toolpath is parametrically given by a B-spline ruled surface, the G-code command lines are typically generated from a dense set of points sampled along the boundary curves (or alternatively one boundary curve and the ruling direction at each point). The procedure of developing command lines in CNC programmers (e.g., Siemens NX) generally considers equal time steps for tool motion between sampled points. An important point to mention is that these sampling points are also obtained by uniform parameter increments in most occasions. As a result, the execution system time domain $t \in [0, \tau]$ is linearly transferred to the motion parameter space $u \in [0, 1]$ by means of a constant scaling coefficient (i.e., $u = t/\tau$). In this way, all time derivatives of the trajectory are scaled by the powers of $1/\tau$. However, a constant scaling cannot always guarantee a kinematically smooth motion especially at the start and end of the path (see, e.g., [25]). Therefore, finding an optimal transformation between t and u , which results in a smooth motion, is essential.

In this research, we propose utilizing the reparameterization technique upon which the motion path parameter u on both boundary curves $\mathbf{c}_k(u)$ is related to the time variable t by means of a strictly increasing *transfer function* $u = f(t)$ where $f(0) = 0$ and $f(\tau) = 1$. Fig. 4 shows an original parameterization and its reparameterized variant of a 3D curve. In the context of flank CNC machining, one needs to consider two boundary curves reparameterized by single transfer function (TF). It is interesting to note that, in contrast to linear transformation between u and t upon which equal time steps lead to equal steps on the parameter space, applying this reparameterization means that the parameter step-lengths (and their respective mappings on boundary curves) are governed by the transfer function $f(t)$. We also recall a well-known fact that reparameterization does not change the shape of boundary curves (and consequently the ruled surface), but affects solely their derivatives. More precisely, for each boundary curve, the original form $\mathbf{c}_k(u)$ and reparameterized one $\tilde{\mathbf{c}}_k(t) := \mathbf{c}_k(f(t))$ are geometrically identical but parametrically different (see, e.g.,

[20, 22, 26]). A similar scenario can be expressed for the ruled surface, i.e., $\mathbf{s}(u, v) \equiv \tilde{\mathbf{s}}(t, v)$. Consequently, by finding an appropriate expression for $f(t)$, we are able to reach an optimal smooth toolpath (noting that, herein and in the following, the optimality criterion is to reach a jerk-minimized toolpath).

In the proposed method, along with the B-spline representation of the ruled surface, the reparameterization function $u = f(t)$ is also introduced as a B-spline curve in terms of time variable. B-spline formulation offers sufficient flexibility (e.g., arbitrary order of continuity in the function and its derivatives) for our jerk minimization purpose. Hence, we recall Eq. (1) to define the TF as an m -th degree B-spline function with $M + 1$ control points (i.e., coefficients) q_0, q_1, \dots, q_M spanned over the knot vector \mathbf{t} as follows:

$$u = f(t) := \sum_{i=0}^M B_i^m(t) q_i, \quad (12)$$

$$\mathbf{t} = [0, \underbrace{0, \dots, 0}_{m+1}, t_{m+1}, t_{m+2}, \dots, t_M, \underbrace{\tau, \tau, \dots, \tau}_{m+1}]. \quad (13)$$

It should be noted that since the curve parameter u is supposed to be in $[0, 1]$, the first and last control points of $f(t)$ are $q_0 = 0$ and $q_M = 1$, respectively. All other q_i s must lie within the interval $[0, 1]$ and be monotone (i.e., $q_i > q_{i-1}$). The control points and knot sequence of TF play crucial roles in smoothing the kinematic specifications of the toolpath. In the next section, we will present an optimization-based framework that looks for the control points q_i s and the knots t_j s of the TF to minimize the jerk of the toolpath.

3. Toolpath jerk minimization

3.1. Optimization problem

As stated in the Introduction, the jerk minimization is taken into consideration in this research as a significant optimality criterion in smoothing toolpaths of multi-axis flank CNC machining. Hence, in order to find the jerk-minimized toolpath, we need to determine the optimal reparameterized form of the respective ruled surface, i.e., $\tilde{\mathbf{s}}(t, v)$. Since the optimized boundary curves $\tilde{\mathbf{c}}_k(t)$ need to share the same parameterization to preserve the projection rule of Eq. (6), we seek a transfer function $u = f(t)$ to be applied to both curves. For this purpose, we perform an optimization process in the context of functional analysis, i.e., we minimize the jerk magnitudes on both boundary curves considering the reparameterization function $f(t)$, represented by control points q_i s and knot values t_j s in Eqs. (12) and (13), as the unknowns. The objective function is defined as a weighted sum of L^2 -norms of jerks of the respective boundary curves as

$$F := w_1 \int_0^\tau \|\mathbf{J}_1(t)\|^2 dt + w_2 \int_0^\tau \|\mathbf{J}_2(t)\|^2 dt, \quad (14)$$

where $\|\mathbf{J}_k(t)\|^2 = \langle \mathbf{J}_k(t), \mathbf{J}_k(t) \rangle$ is the square of Euclidean norm of the jerk. The weights w_1 and w_2 in the above integral might be set based on different considerations, e.g., due to conditions imposed on the boundary curves or due to special type of the tool or milling machine. The optimization problem can be further

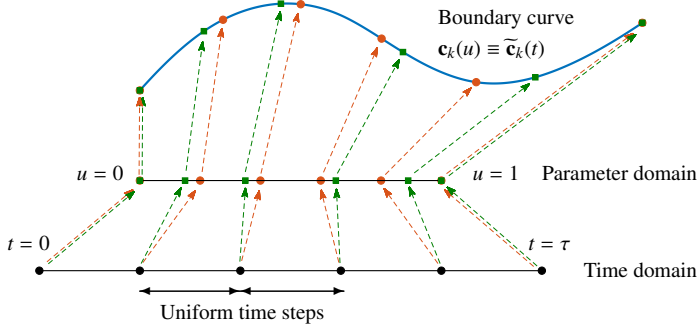


Figure 4: Reparameterization. Left: A boundary curve in the typical and (arbitrarily) reparameterized forms. Red circles show the typical linear transformation between u and t upon which equal time steps leads to equal steps on the parameter space, while green squares illustrate the parameters created by reparameterization upon which the parameter step-lengths are governed by the transfer function $u = f(t)$. Right: The respective linear (red) and general (green) transfer functions.

written as:

$$\text{Find } \left\{ \begin{array}{l} q_i \quad i = 1, 2, \dots, M-1 \\ t_j \quad j = m+1, m+2, \dots, M \end{array} \right\} \text{ which minimize } F$$

$$\text{subject to } \left\{ \begin{array}{l} q_i - q_{i-1} \geq \frac{1}{\alpha(M-m+1)} \\ t_j - t_{j-1} \geq \frac{1}{\beta(M-m+1)} \end{array} \right. \quad (15)$$

where $\alpha, \beta \geq 1/(M-m+1)$ are parameters to control the proximity of adjacent control points and knots, respectively. Large values of α and β push the consecutive knots and control points to be closer, and to eventually lower the continuity in the limit ($\beta \rightarrow \infty$). On the other hand, smaller values result in wider gaps between the knots and/or the control points (for instance, for knots, this constraint pushes the knot vector toward the uniform case). If not stated differently, we set the default values to $\alpha = \beta = 10$ in our implementation.

The optimization constraints express our wish of increasing sequence of knots and control points such that $f(t)$ is well-defined and increasing. Finally, we also require the boundary constraints as $q_0 = 0$, $q_M = 1$, $t_m = 0$, and $t_{M+1} = \tau$ assuming the total execution time is τ . However, these boundary knots and values are set directly and therefore are not included as variables in Eq. (15).

Remark 2. Since obtaining the jerk profiles entails computing the third derivative of $\mathbf{c}_k(u)$, employing quartic boundary curves for the toolpath would guarantee only a C^0 -continuous jerk. In order to have a smooth jerk profile (at least C^1 -continuous), $\mathbf{c}_k(u)$ is required to be of degree $n = 5$ or higher. In addition, referring to Eq. (9), and for the same reason, the transfer function needs to be at least of degree $m = 5$ to reach a smooth jerk–time representation.

3.2. Evaluation of the objective function

In order to perform the jerk minimization, we need to calculate the objective function F defined by Eq. (14). Depending on the degrees of $\mathbf{c}_k(u)$ and $u = f(t)$, and based on the chain derivative rule of Eq. (9), the degree of the objective function can grow drastically. For instance, if the TF and boundary curves are all expressed in terms of quintic B-spline bases (i.e., $n = m = 5$), the degree of F would be 44. Since the analytical integration

of functions of such high degrees is computationally expensive, and needs to be computed repeatedly in an iterative optimization procedure, we prefer Gaussian quadrature rules as numerical integration schemes, because they are known to be computationally cheap (see, e.g., [27]). To test the numerical exactness of the integration, we experimented with various Gaussian rules as follows.

Consider an example boundary curve that is a quintic B-spline with $N + 1 = 11$ control points spanned over a uniformly-spaced knot vector (see Fig. 5). It is assumed that this path, which is a reproduction of a 3D example in [20], is to be traversed in $\tau = 1$ min.

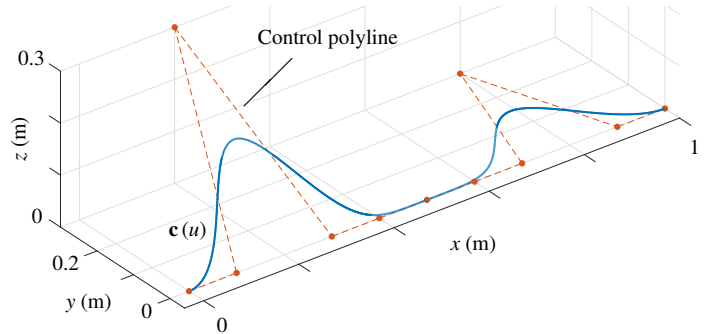


Figure 5: A 3D curve example reproduced from [20] for accuracy assessment of the function evaluation procedure of the proposed toolpath jerk minimization method.

For the first part of the test, Table 1 shows the absolute and relative errors of *global* Gaussian integration of the objective function F for the 3D curve shown in Fig. 5 (herein, the term *global* denotes the integration over the entire time domain $[0, \tau]$). The error values are obtained by comparing the numerical integral with the exact analytical one. We consider a random TF (i.e., non-optimized) as its shape is not relevant to this exactness test. The degree and number of control points of the TF are $m = 5$ and $M + 1 = 15$, respectively. For performing the integration, various numbers of Gauss points (GPs) are employed to evaluate the objective function.

In the second part of the test, Table 2 shows the absolute and relative errors of *local* (or so-called *element-wise*) Gaussian integration of the objective function F for the same curve where the integration is performed locally over the knot spans

Table 1: Accuracy assessment of global numerical integration of the objective function for the 3D curve shown in Fig. 5 with an arbitrary (non-optimal) quintic TF with 15 control points (10 knot spans). Absolute and relative errors are obtained by various numbers of Gauss points (# GPs) and the path is assumed to be traversed in $\tau = 1$ min.

Analytical integration F (m ² /min ⁶)	Global Gaussian quadrature		Error	
	# GPs	F (m ² /min ⁶)	Absolute	Relative
35470	5	18239	17230	48.58%
	10	39298	3828.4	10.79%
	20	36528	643.0	2.99%
	30	34987	482.5	1.36%
	40	35616	146.11	0.41%
	50	35495	25.0	0.07%

$[t_j, t_{j+1}]$ of the transfer function. Again, we assume the degree of the TF is $m = 5$, but the number of control points varies as $M + 1 = 7, 11$, and 15 , so that we can observe the accuracy of the element-wise numerical integration for different numbers of knot spans. For lower number of knot spans (i.e., lower M), we need more GPs to reach an accurate integration since, in this case, the knot spans are wider and the integrand may oscillate more. On the other hand, by increasing the number of knot spans of the TF, lower number of GPs is required for the element-wise integration, guaranteeing fast and accurate evaluation of the objective function for the iterative optimization process. The results of these tables indicate that local integration with 7–8 GPs can provide sufficient accuracy if a suitable number of knot spans is considered for the TF. Another option to reduce the computational cost of integration with such integrand of the polynomial spline type is to apply the spline Gaussian rules (see, e.g., [28, 29]) that require even fewer quadrature points, yet guaranteeing the exactness of the integration.

3.3. Ruling distance metric

In this research, we perform a gradient-based optimization by employing the sequential quadratic programming (SQP) as an iterative method for constrained nonlinear optimizations, widely used on spline-based problems with a twice continuously differentiable objective function (see, e.g., [30, 31]). The SQP method solves a sequence of subproblems, each of which optimizes a quadratic model of the objective function, guaranteeing a super-linear convergence by using a quasi-Newton updating procedure (for more details regarding the mathematical aspects of this method, the readers are referred to, e.g., [32, 33]). As the objective function in Eq. (14) is highly nonlinear, one needs to have a good initial guess of the unknown design variables (i.e., q_i s and t_j s) to avoid the optimization process to be terminated at an undesirable local minimum. Even by performing thousands of optimization runs with different random initial guesses for unknown variables, there may be still the risk that the best optimal TF cannot be captured. Furthermore, considering the computational cost of the iterative optimization process, this cluster of runs is completely far from the desired fast toolpath smoothing strategy that is the main goal of this research.

In order to overcome the above-mentioned obstacles, an innovative initialization technique for the jerk minimization process is presented by means of an adequate *ruling distance metric* (RDM). This metric is defined on a space of finite lines (i.e.,

rulings) in 3D-space and reliably captures distance between the positions of the milling tool. Using this metric, we show that almost-optimal jerk minimized path can be found with a *single* round of the optimization process.

Defining line $\ell_i := (\mathbf{a}_i, \mathbf{b}_i)$ as a ruling that connects points $\mathbf{a}_i := \mathbf{c}_1(\hat{u}_i)$ and $\mathbf{b}_i := \mathbf{c}_2(\hat{u}_i)$ on the boundary curves (where \hat{u} is an arbitrary parameter in $[0, 1]$), for any set of two consecutive rulings ℓ_i and ℓ_{i+1} obtained by uniformly-spaced parameters \hat{u}_i and \hat{u}_{i+1} , and their uniformly-spaced time domain counterparts \hat{t}_i and \hat{t}_{i+1} , we construct a bilinear patch on the ruled surface $\mathbf{s}(u, v)$ as shown in Fig. 6. We follow the idea of the metric of finite rulings [34] and define the RDM metric $d(\ell_i, \ell_{i+1})$ as a measure for the distance traversed by the milling tool between rulings ℓ_i and ℓ_{i+1} in the time step $[\hat{t}_i, \hat{t}_{i+1}]$. This metric is introduced as the L^2 -norm of the Euclidean distance function of rulings as follows (multiplied by 3 for convenience):

$$\begin{aligned}
 d(\ell_i, \ell_{i+1}) &:= 3 \int_0^1 \|(1-v)(\mathbf{a}_i - \mathbf{a}_{i+1}) + v(\mathbf{b}_i - \mathbf{b}_{i+1})\|^2 dv \\
 &= \|\mathbf{a}_i - \mathbf{a}_{i+1}\|^2 + \|\mathbf{b}_i - \mathbf{b}_{i+1}\|^2 \\
 &\quad + \langle (\mathbf{a}_i - \mathbf{a}_{i+1}), (\mathbf{b}_i - \mathbf{b}_{i+1}) \rangle. \quad (16)
 \end{aligned}$$

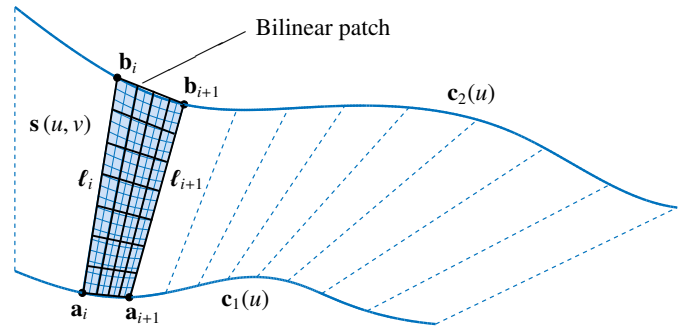


Figure 6: A bilinear patch (black) that approximates the ruled surface patch (blue) of $\mathbf{s}(u, v)$ between rulings ℓ_i and ℓ_{i+1} .

From the kinematic point of view, the RDM metric d is a measure that indicates how fast the tool is traveling between rulings in uniform time steps. Equally importantly, and unlike the Plücker's metric of infinite lines (see, e.g., [35]), this metric captures sudden changes of one endpoint of the ruling, while the other end stays (almost) fixed, see Fig. 7.

It is obvious that for larger RDM values, under uniform time steps, the tool should move faster to reach the next ruling (i.e., the next parameter \hat{u}) that will result in higher acceleration and jerk values. Taking into account that most of CNC programmers consider linear transformation between u and t , the generated toolpath may reveal undesirable kinematic specifications if the respective ruled surface is inappropriately parameterized. The main advantage of defining this metric is that we can *reorient* the time domain based on the RDM magnitude to have larger time steps for larger ruling deviations, leading to a good initialization for jerk minimization. Since the ruling distances are governed by the original parameterization of boundary curves, this reparameterization implies that the tool goes on a shorter time step when the rulings are close and on a larger one when the rulings are further one from another. We herein refer to time reorientation as an alternative definition of reparameterization in which we keep the parameters equally-spaced while the time steps are different (see Fig. 8). This definition will help to create the initial TF for

Table 2: Accuracy assessment of local numerical integration of the objective function for the 3D curve example of Fig. 5 with arbitrary (non-optimal) quintic TFs with 7, 11, and 15 control points (i.e., 2, 6, and 10 knot spans). Absolute and relative errors obtained by different numbers of Gauss points (# GPs) and the path is assumed to be traversed in $\tau = 1$ min.

#GPs	2 knot spans				6 knot spans				10 knot spans			
	F (m ² /min ⁶)		Error		F (m ² /min ⁶)		Error		F (m ² /min ⁶)		Error	
	Analytical	Numerical	Abs.	Rel.	Analytical	Numerical	Abs.	Rel.	Analytical	Numerical	Abs.	Rel.
5		141830	84358	147%		37839	1552	4.28%		35797	327.72	0.92%
6	57472	119950	62482	109%	36287	35257	1030	2.84%	35470	35321	148.91	0.42%
7		87841	30370	52.8%		36618	331.7	0.91%		35507	37.35	0.11%
8		51604	5868	10.2%		36587	300.1	0.83%		35477	6.99	0.02%

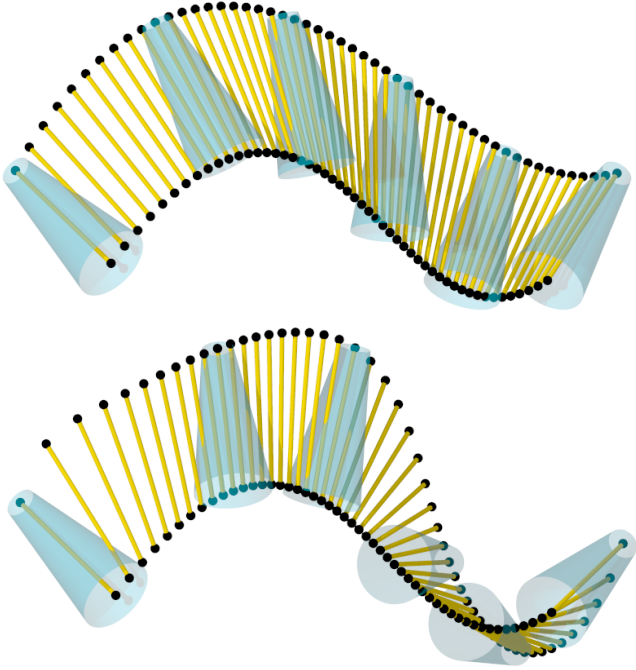


Figure 7: RDM metric. Two milling paths with a conical tool (transparent) are shown. While the Plücker’s metric gives almost the same value for the neighboring positions of the milling axis (yellow), RDM assigns larger values to the motion with the more intensive deviation of the axis (below). Consequently, low values of RDM capture better motions preferable for milling.

the optimization process based on the RDM metric in an effective manner. Consider now $N_d + 1$ uniformly-spaced parameters $\hat{u}_0, \hat{u}_1, \dots, \hat{u}_{N_d} \in [0, 1]$ in Fig. 8, then in this discrete setup, the re-oriented time values \tilde{t}_i corresponding to each \hat{u}_i are defined as follows:

$$\tilde{t}_i := \frac{\sum_{j=0}^i d_j}{\sum_{j=0}^{N_d} d_j} \tau, \quad i = 1, 2, \dots, N_d, \quad (17)$$

where $\tilde{t}_0 := 0$, $d_0 := 0$, and $d_j := d(\ell_{j-1}, \ell_j)$. To get a smooth initial TF : $u = f(t) = \sum_{i=0}^M B_i^m(t) q_i$ for the jerk-optimization process, we apply B-spline fitting on the (\tilde{t}_i, \hat{u}_i) values. In particular, we follow the scheme proposed in [22] and initialize the (non-uniform) knot vector \mathbf{t} by applying the De Boor’s algorithm and initialize the control points q_i by minimizing the fitting error in the least-squares sense.

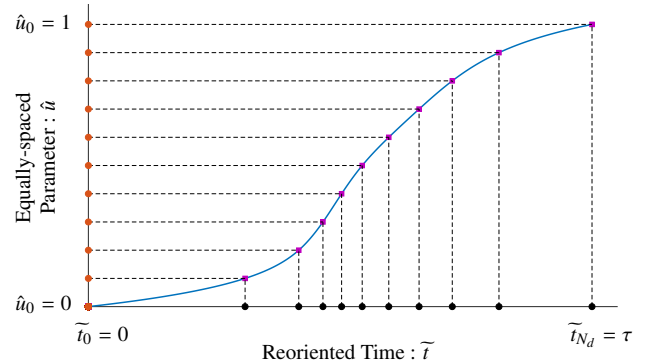


Figure 8: An example of the initial transfer function $u = f(t) = \sum_{i=0}^M B_i^m(t) q_i$ for jerk-optimization process obtained by ruling distance metric. The function is fitted to equally-spaced parameters \hat{u}_i corresponding to reoriented time values \tilde{t}_i .

4. Case studies

In this section, the applicability of the proposed method of this paper is demonstrated on several benchmark examples. We again emphasize that the objective of this work is to kinematically smooth a toolpath obtained *a priori* by some toolpath design strategies. We also assume that the given toolpath already meets the axes constraints of a particular milling machine. It is important for the smoothing procedure not to violate these constraints. It will be shown in numerical examples that the kinematic specification (i.e., velocity, acceleration, and jerk) generally become lower than the original ones after reparameterization.

4.1. The applicability of the ruling distance metric

The main goal of the first case study is to validate the correctness of our proposed ruling distance metric (RDM) described in Section 3.3 for the jerk minimization problem. For this purpose, we explore the space of admissible transfer functions that should be monotone and meet the conditions of $u(0) = 0$ and $u(\tau) = 1$. We then sample a set of 5000 random initializations of $u = f(t)$ for the optimization process and compare the results with that obtained by the RDM metric. Fig. 9 demonstrates a toolpath of a flank milling process on the turbine *blisk* benchmark workpiece. The ruled surface has quasi-parallel rulings (i.e., with slight ruling rotations) and is constructed by two quintic boundary curves with $N + 1 = 15$ control points spanned over a uniformly-spaced knot vector. The toolpath of this example, which is assumed to be traversed in $\tau = 5$ sec, is taken from [23].

Starting with the case of 5000 random initializations, Fig. 10 depicts the optimized TFs obtained by the optimization process,

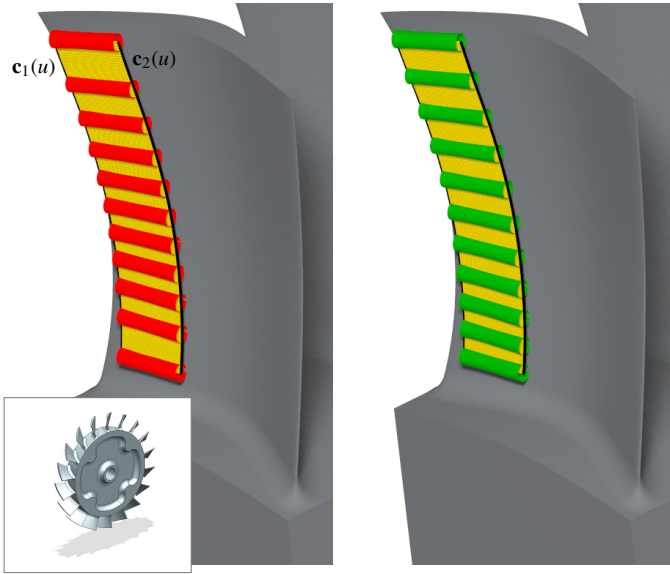


Figure 9: Left: An original path of a cylindrical milling tool (red) uniformly sampled in time (we used 10 equal time steps for visualization of the tool). No reparameterization is applied, i.e., $u = t$, on the boundary curves $c_1(u)$ and $c_2(u)$. Right: The tool-path after the jerk-minimized reparameterization is applied, c.f. Figs. 10–13. The time-space t is sampled uniformly, while the parameter space is obtained via $u = f(t)$. Observe a more uniform distribution of the green cylinders after reparameterization (with the same 10 equal time steps). The toolpaths correspond to 5-axis flank milling of a blade of the blisk model (framed) [23].

where equal weights are considered for the objective function F (i.e., the total L^2 -norms of jerks on both endpoints of the tool’s axis). In this figure, the color coding reflects the quality with respect to the quality measure, i.e., the value of the objective function Eq. (14). The red color indicates large values of F while dark blue signals vanishing F . Observe that not all initial guesses converge to the true minimizers and get stuck at some local minimum, including the linear transformation depicted as a dashed black line. Most of the initial guesses converge to a roughly similar TFs highlighted as a blue cluster at the middle of the graph. In this example, we considered a quintic TF : $u = f(t)$ (see Remark 2) with $M + 1 = 15$ control coefficients q_i where the integrations are performed by element-wise Gaussian quadrature with 7 GPs.

As a comparison with the random sampling of the initial transition function, we employ the RDM metric to construct a single initialization based on the geometry of the ruled surface. Considering $N_d = 200$ in Eq. (17), the initial and optimal TFs with different numbers of control points are obtained by the proposed metric and visualized in Fig. 11. These TFs are also compared with the best case of the 5000 random initializations. The figure indicates that optimal TFs computed by both RDM metric and random sampling are (almost) identical. More interestingly, the RDM-based TF with 15 control points is almost the same as the one obtained by random sampling with the same number of control points (see the cyan curve in Fig. 11). The respective objective function values and maximum jerk magnitudes on boundary curves are reported in Table 3. Again, it can be seen that both random and RDM-based initializations lead to almost similar optimal values, considering that using the RDM metric, the results are computed by a single optimization process and

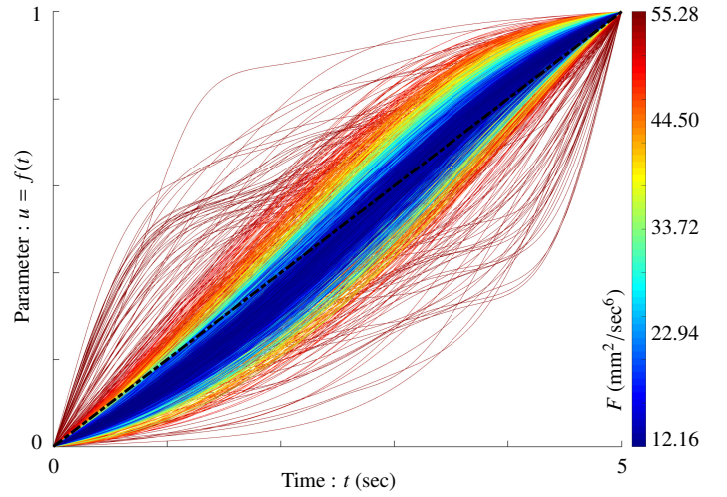


Figure 10: Optimal transfer functions $u = f(t)$ for the first case study obtained by 5000 samples. The colormap shows how good the optimal TFs are with respect to the objective function value F (i.e., the total L^2 -norms of jerks). The black dashed diagonal line represents the linear parameter–time transformation (i.e., no reparameterization).

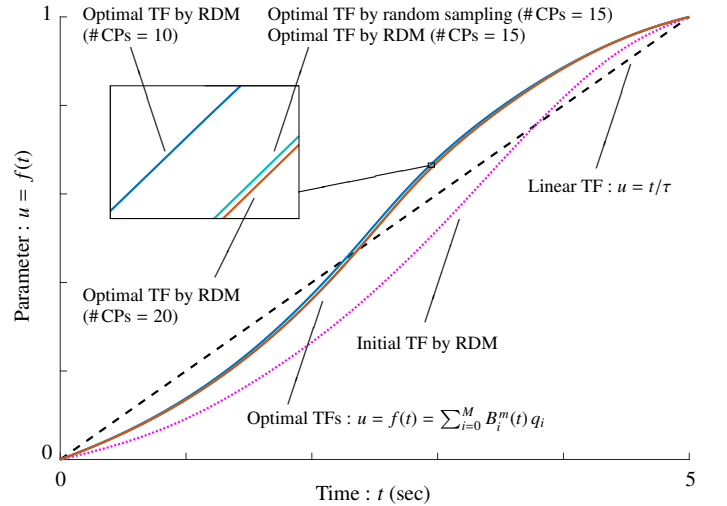


Figure 11: Optimal transfer functions obtained by the proposed RDM metric with different numbers of control points (#CPs) compared with that obtained by the random initializations. The respective values of the objective function (F) and maximum jerk magnitudes are reported in Table 3.

therefore with a significantly lower computational time. Particularly in this comparison, the computational time is governed by the number of iterations, which depends on how far the initial guesses are from the optimized values. Using the proposed RDM metric, we find an initialization for the TF that is closer to the final optimal one and, consequently, the computational time is decreased. Regarding the number of control coefficients, it should be also noted that by increasing the number of q_i s, the TF is more flexible and a (slightly) better jerk profile may be found. On the other hand, it leads to more variables and therefore a more expensive optimization. As a result, a compromise between computational cost and required accuracy should be considered.

Table 3: Maximum jerk magnitudes on boundary curves and the objective function value (F) for the first case study. Optimal TFs are obtained by the random initializations and the proposed RDM metric, respectively. The effect of employing different numbers of control points (# CPs) for the transfer function is shown and the advantage of implementing the RDM metric (in terms of computational cost) is highlighted.

	Linear TF (non-optimal)	Optimal TF by 5K random initializations (# CPs = 15)	Optimal TF by proposed RDM metric		
			# CPs = 10	# CPs = 15	# CPs = 20
$\max(\ \mathbf{J}_1(t)\)$ (mm/sec ³)	31.2402	2.4308	2.5456	2.4303	2.4207
$\max(\ \mathbf{J}_2(t)\)$ (mm/sec ³)	78.0993	2.6108	2.3940	2.6133	2.6037
Objective function value F (mm ² /sec ⁶)	853.7908	12.1626	12.6436	12.1628	12.1195
Computational time (sec)	–	61.7* (average time per case)	14.8**	45.5	111.4
# Iterations	–	73 (average # iterations per case)	31	39	44

* The reported time for the case of 5000 random initializations, is the average of parallel runs on 50 CPUs (100 cases per run). The total time is 6171 sec.

** The reported time for the proposed method is obtained by a single run on one CPU.

The velocity, acceleration, and jerk plots of the original toolpath with linear parameter–time relationship (i.e., without reparameterization) and the optimized toolpath obtained by employing an optimal TF with 15 control points are shown in Fig. 12. It can be seen that the jerk profile is substantially optimized and smoothed compared to the original input toolpath (see Table 3). We again emphasize that the geometric shape of the toolpath remains unchanged through the jerk minimization procedure so that any consideration in the path planning stage is kept.

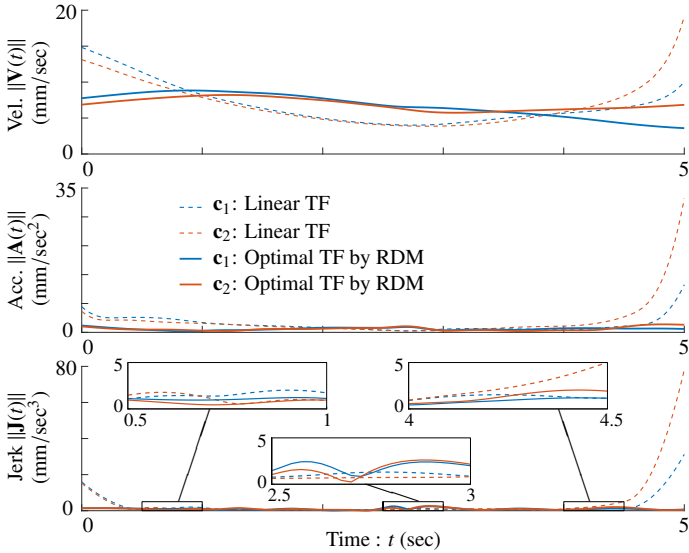


Figure 12: Velocity (top), acceleration (middle) and jerk (bottom) profiles for boundary curves \mathbf{c}_1 and \mathbf{c}_2 of the example shown in Fig. 9. The values are computed by RDM-based optimal TF : $u = f(t)$ and compared to the linear TF : $u = t/\tau$. The zoomed-in boxes show in detail the comparison of the four TFs. When applying the reparameterization, observe a minor worsening in the middle of the domain in contrast to a major improvement close to the boundaries.

In order to have a better interpretation of the reparameterization technique and see how it may change the kinematic specifications along the toolpath, Fig. 13 depicts the velocity, acceleration, and jerk vectors obtained by linear and optimal transformations at some sampled points on the toolpath. Under either of parameterizations we set (linear or optimal), the milling tool reaches the shown locations at the same parameter u , but at different times t . The reason is that, for the same param-

eter values, reparameterization reorients the time domain and the velocity magnitude is changed accordingly. However, referring to Eqs. (7–9), the velocity direction remains unchanged and tangent to the path (for the acceleration and jerk, both direction and magnitude are changed). Observe that the acceleration and jerk values, which are excessive at both ends of the original path (see Fig. 12) are well decreased by the presented reparameterization algorithm.

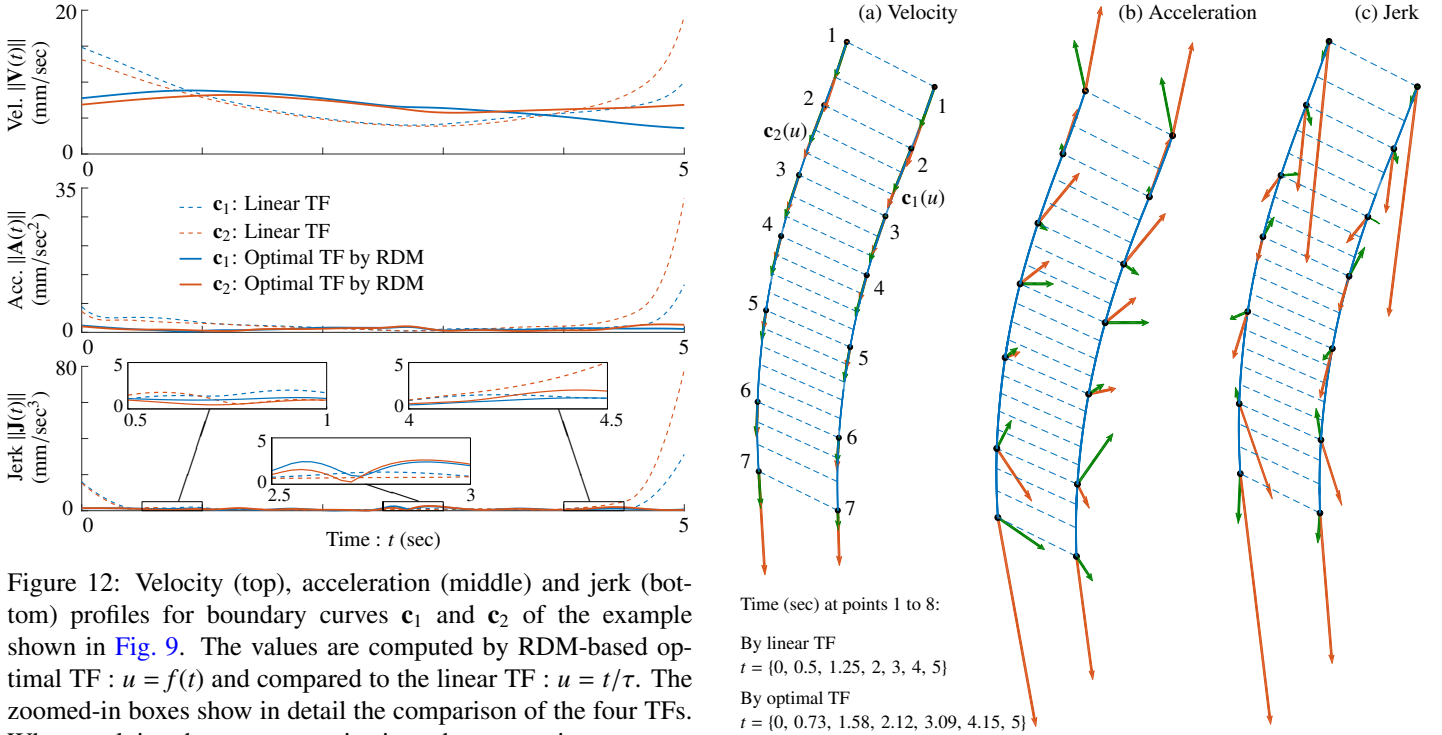


Figure 13: (a) Velocity, (b) acceleration, and (c) jerk vectors at some sampled points on the ruled surface of the example shown in Fig. 9 (the view is changed to better visualize the vectors). Red vectors correspond to (original) linear transformation between u and t , while the green ones correspond to the TF obtained by the RDM metric. The time values of reaching each point in different configurations are also reported. The acceleration and jerk vectors of the original toolpath are scaled at both ends to be fitted in the figure.

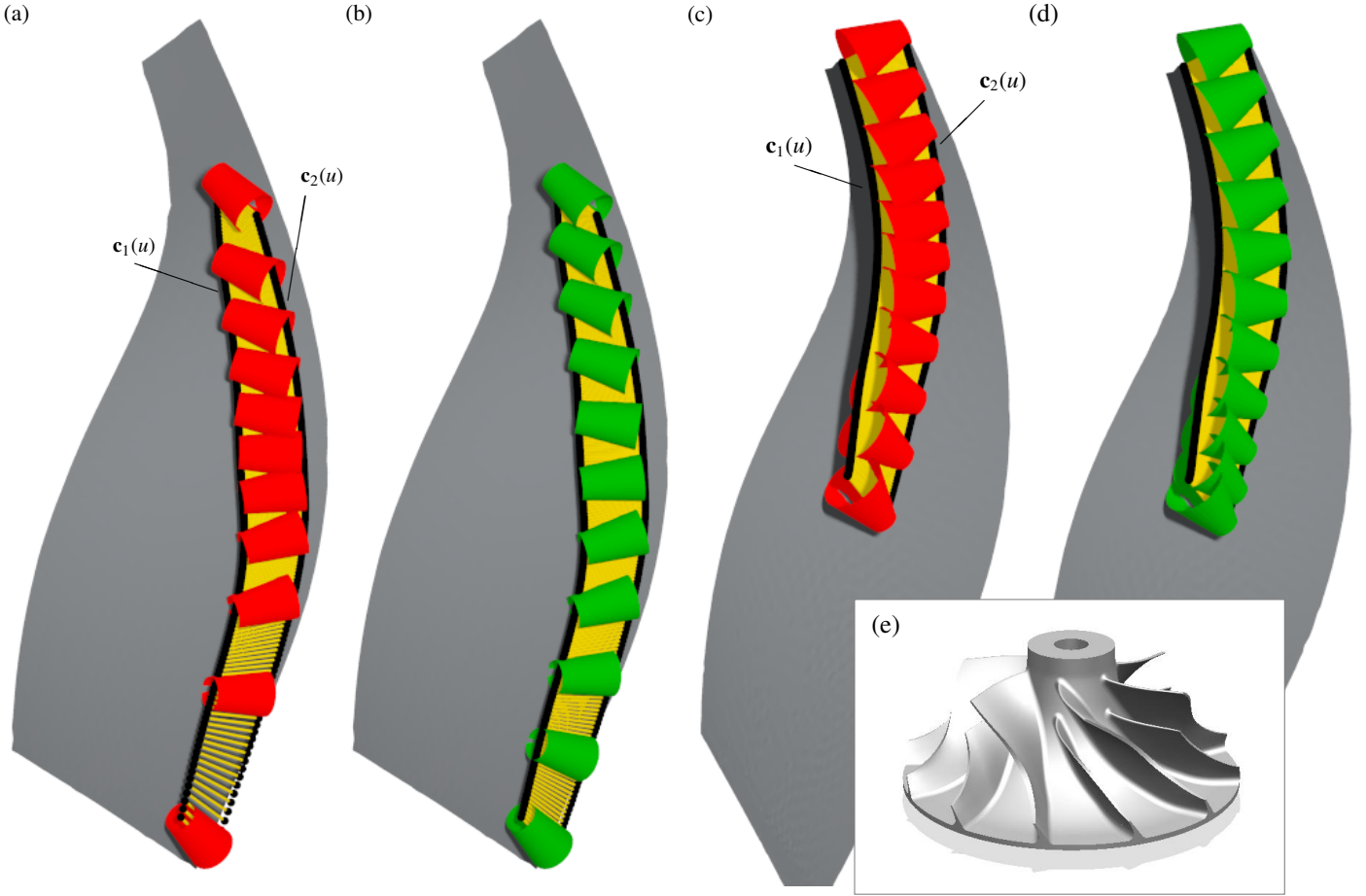


Figure 14: Two milling paths, (a,b) and (c,d), respectively, on a blade (gray) of an impeller (e). It is a simulation of 5-axis flank milling, data courtesy of [1], using two distinct conical milling tools. The red positions of the tool are related the original (linear) parameterization, while the green positions show the motion after jerk-minimized reparameterization. All positions are uniformly sampled in the time space. Similarly to Fig. 9, the reparameterized motion better captures distances between particular positions of the tools, resulting in smoother motion (in terms of jerk).

4.2. The impeller benchmark workpiece

In order to show the applicability of the presented methodology, in this section, we study two toolpaths from the *impeller* benchmark workpiece where a higher variation of ruling directions occurs. Fig. 14 shows two toolpaths on the impeller workpiece obtained by a toolpath generation algorithm (see [1]). Both corresponding ruled surfaces have quintic boundary curves, where the first toolpath (Figs. 14a and 14b) has uniform knots and 15 control points, while the second one (Figs. 14c and 14d) is spanned over a non-uniform knot sequence with 17 control points.

The optimal (quintic) transfer function for each toolpath is computed by the presented reparameterization technique where the initialization for the optimization process is obtained by the proposed RDM metric. The respective optimal TFs and the plots of velocity, acceleration and jerk are depicted in Figs. 15 and 16, respectively. We assume that both paths are to be traversed in $\tau = 10$ sec and the kinematic specifications do not violate the machine constraints. Both results consider equally important jerk minimization on either ends, i.e., $w_1 = w_2$ in Eq. (14).

4.3. Discussions and limitations

Various optimization weights. In all our case studies we considered equal weights in Eq. (14) because our research is motivated by 5-axis flank milling and the jerk of the tool needs to be

minimized throughout the whole grazing curve. Since the jerk of the intermediate axis points is just a linear combination of the jerks on the axis' ends, we require the jerk to be minimized on both ends equally. One can, however, consider jerk minimization in the context of ball-end milling and there it makes sense to minimize the jerk of the whole tool by prioritizing the weight on the side closer to the contact point. This can be easily controlled by changing the weights in Eq. (14).

Considerations on the transfer functions. We initialize the transfer function using a quintic B-spline to meet the minimum requirement of having a C^1 -continuous jerk-time profile (see Remark 2). One can eventually experiment with higher degrees in situations where the initial data of the transfer function (obtained by the RDM metric) would indicate more degrees of freedom. We experimented with a higher number of control coefficients of the transfer function to increase the number of degrees of freedom which increased the computational cost and improved the jerk only negligibly (as reported in Sections 3.2 and 4.1). After these experiments, we set the degree to 5 and the number of the control points to 15 in our implementation.

G-code input. In the case when the toolpaths are given in terms of discrete data points or in terms of a G-code, our algorithm can be applied as well. We just need to know the directions and length of the rulings that can be easily extracted from motion

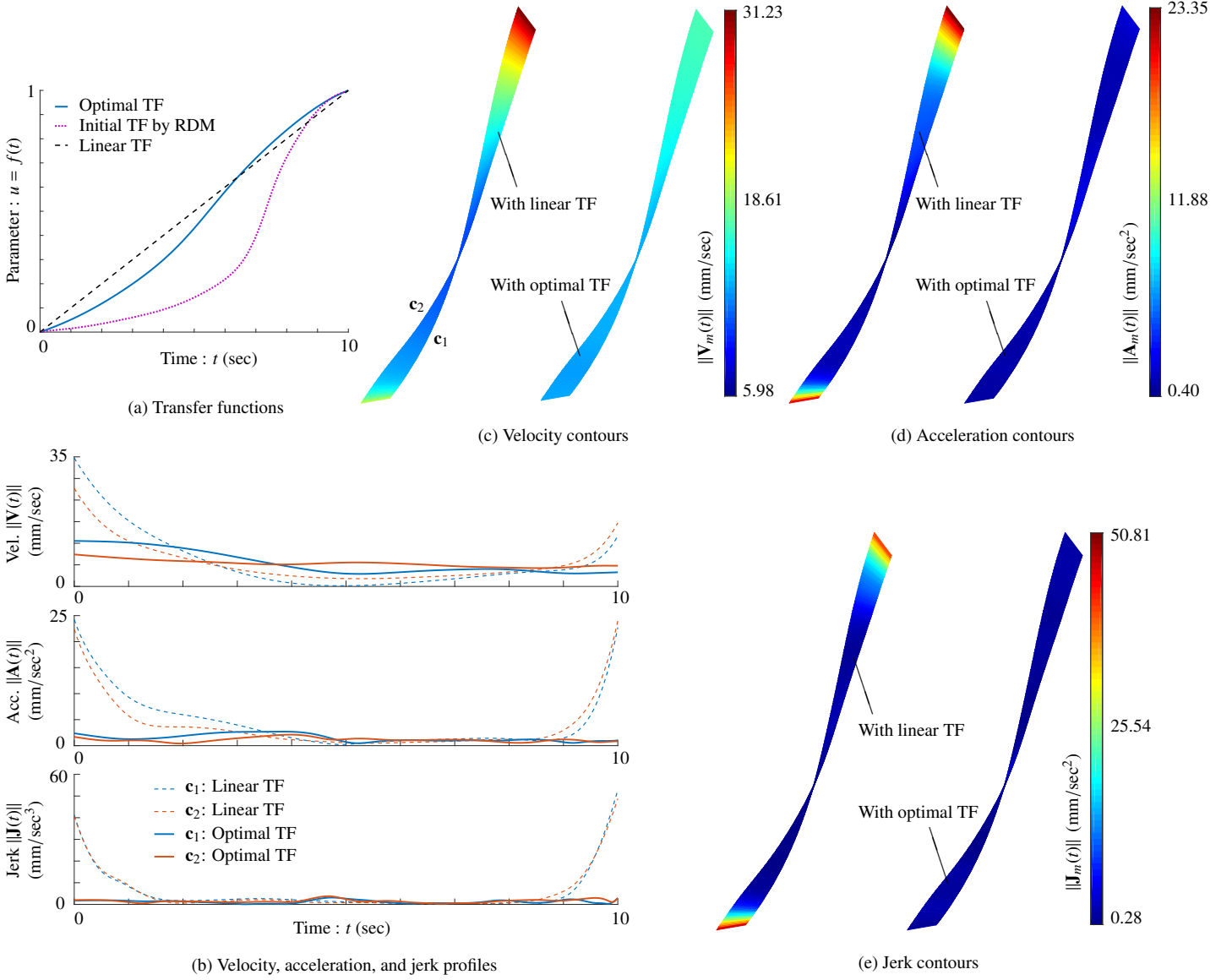


Figure 15: Jerk minimization of the 1st toolpath of the impeller workpiece (Figs. 14a and 14b). (a) Optimal vs. linear TFs compared to the RDM-based initial TF. (b) Velocity, acceleration, and jerk profiles over time obtained by linear and optimal TFs on boundary curves c_1 and c_2 . (c), (d), and (e) Contours of velocity, acceleration, and jerk on the ruled surface obtained by linear and optimal TFs where the values of $\|\mathbf{V}_m(t)\|$, $\|\mathbf{A}_m(t)\|$, and $\|\mathbf{J}_m(t)\|$ are the means of respective values on boundary curves in (b). Note we changed our view to be in accordance with global xyz coordinate system.

path *goto* files or from G-codes. Our experiments were applied to a smooth (B-spline) input, see e.g. Fig. 9, however, our follow-up project that deals with physical validations will deal directly with G-codes.

Kinematic limits and smooth motions on machine drives.

In this work, we focus on the jerk values of the tool motion that are derived from the geometry of the toolpath. Hence, the optimization objective is evaluated with reference to the *workpiece coordinate system* (WCS). From the practical point of view, it is more relevant to optimize the jerk values on machine drives, i.e., X, Y, Z, B and C in 5-axis machining, which are referred to as the *machine coordinate system* (MCS). Note that the currently smoothed toolpath in WCS may result in a non-smooth motion in MCS. Such an approach, requires to consider a particular machine (and the limits on its drives) as a part of the optimization algorithm. However, optimization in MCS would require a dif-

ferent metric since the current RDM measures distances between the rulings and does not correspond to an equal distribution of the motion in MCS, considering each drive may have different kinematic limits. Then, with such a metric at hand, fixing the workpiece setup and going through the *inverse kinematic transformation* (IKT) to convert the optimization objective from WCS to its counterpart in MCS is an interesting topic for a follow-up project.

5. Conclusions

We have introduced a new jerk-minimizing algorithm for multi-axis machining. The algorithm looks for a reparameterization of the milling motion such that the jerk is globally minimized. In our initialization–optimization based framework, we use the ruling distance metric that measures the distance between

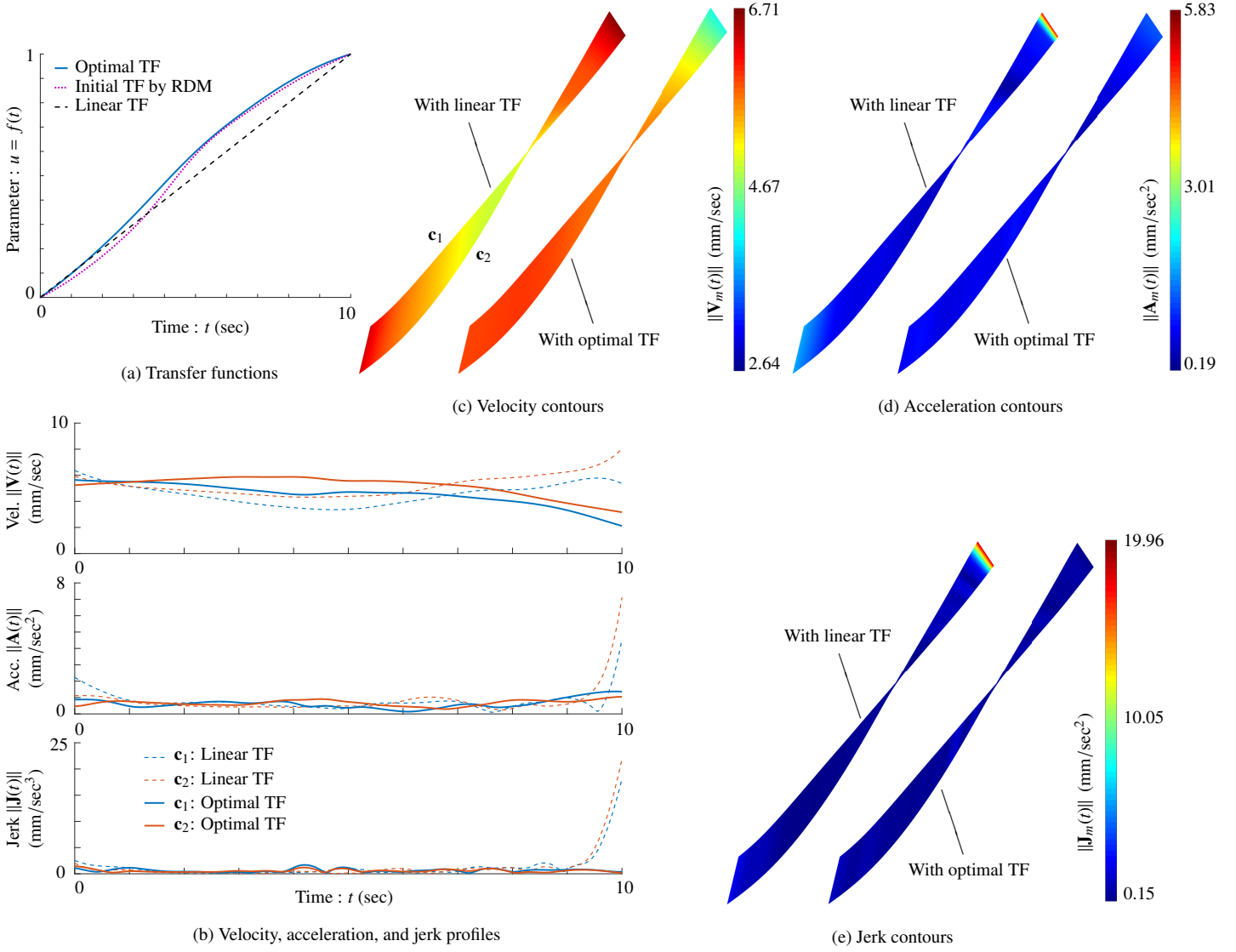


Figure 16: Jerk minimization of the 2nd toolpath of the impeller workpiece (Figs. 14c and 14d). (a) Optimal vs. linear TFs compared to the RDM-based initial TF. (b) Velocity, acceleration, and jerk profiles over time obtained by linear and optimal TFs on boundary curves c_1 and c_2 . (c), (d), and (e) Contours of velocity, acceleration, and jerk on the ruled surface obtained by linear and optimal TFs where the values of $\|\mathbf{V}_m(t)\|$, $\|\mathbf{A}_m(t)\|$, and $\|\mathbf{J}_m(t)\|$ are the means of respective values on boundary curves in (b). Note we changed our view to be in accordance with global xyz coordinate system.

two finite lines in 3D-space. We have shown that this metric can be used for the initialization of the motion reparameterization as it reliably captures changes of the milling tool's axis. We have validated our algorithm on several case studies and have shown that the proposed method quickly decreases the jerk, resulting in smooth milling motions. Our follow-up research aims at physical validations of the proposed methodology, and incorporating other physical entities such as kinematic limits on machine drives, tool vibration and/or cutting forces into the optimization loop.

Acknowledgment

The first author has been partially funded by the BCAM Severo Ochoa accreditation of excellence, Spain (SEV-2017-0718), and the Basque Government, Spain through the BERC 2018–2021 program. The second author has been partially supported by the National Natural Science Foundation of China

(61672187) and the Shandong Provincial Key R&D Program (2018GGX103038). The third author has been partially supported by Spanish Ministry of Science, Innovation and Universities: Ramón y Cajal with reference RYC-2017-22649 and the European Union's Horizon 2020 research and innovation programme under agreement No. 862025.

References

- [1] A. Calleja, P. Bo, H. González, M. Bartoň, L. N. López de Lacalle, Highly accurate 5-axis flank CNC machining with conical tools, *The International Journal of Advanced Manufacturing Technology* 97 (2018) 1605–1615. doi:10.1007/s00170-018-2033-7.
- [2] B. Sencer, Y. Altintas, E. Croft, Feed optimization for five-axis CNC machine tools with drive constraints, *International Journal of Machine Tools and Manufacture* 48 (2008) 733–745. doi:10.1016/j.ijmactools.2008.01.002.
- [3] D. Constantinescu, E. A. Croft, Smooth and time-optimal trajectory planning for industrial manipulators along specified paths, *Jour-*

- nal of Robotic Systems 17 (2000) 233–249. doi:10.1002/(SICI)1097-4563(200005)17:5<233::AID-ROB1>3.0.CO;2-Y.
- [4] Y. Zhang, M. Zhao, P. Ye, H. Zhang, A G^4 continuous B-spline transition algorithm for CNC machining with jerk-smooth feedrate scheduling along linear segments, *Computer-Aided Design* 115 (2019) 231–243. doi:10.1016/j.cad.2019.04.004.
- [5] P. Bosetti, E. Bertolazzi, Feed-rate and trajectory optimization for CNC machine tools, *Robotics and Computer-Integrated Manufacturing* 30 (2014) 667–677. doi:10.1016/j.rcim.2014.03.009.
- [6] J. Jahanpour, M. R. Alizadeh, A novel acc-jerk-limited NURBS interpolation enhanced with an optimized S-shaped quintic feedrate scheduling scheme, *The International Journal of Advanced Manufacturing Technology* 77 (2015) 1889–1905. doi:10.1007/s00170-014-6575-z.
- [7] Q. Zhang, S. Li, J. Guo, Smooth time-optimal tool trajectory generation for CNC manufacturing systems, *Journal of Manufacturing Systems* 31 (2012) 280–287. doi:10.1016/j.jmsy.2012.06.001.
- [8] X. Beudaert, P.-Y. Pechard, C. Tournier, 5-axis tool path smoothing based on drive constraints, *International Journal of Machine Tools and Manufacture* 51 (2011) 958–965. doi:10.1016/j.ijmactools.2011.08.014.
- [9] X. Beudaert, S. Lavernhe, C. Tournier, Feedrate interpolation with axis jerk constraints on 5-axis NURBS and G1 tool path, *International Journal of Machine Tools and Manufacture* 57 (2012) 73–82. doi:10.1016/j.ijmactools.2012.02.005.
- [10] Y. Sun, Y. Zhao, J. Xu, D. Guo, The feedrate scheduling of parametric interpolator with geometry, process and drive constraints for multi-axis CNC machine tools, *International Journal of Machine Tools and Manufacture* 85 (2014) 49–57. doi:10.1016/j.ijmactools.2014.05.001.
- [11] P. Hu, K. Tang, Improving the dynamics of five-axis machining through optimization of workpiece setup and tool orientations, *Computer-Aided Design* 43 (2011) 1693–1706. doi:10.1016/j.cad.2011.09.005.
- [12] M. Essid, B. Gassara, M. Baili, M. Hbaieb, G. Dessenin, W. Bouzid Saï, Analytical modeling of the CNC machine axis motion in high-speed milling with local smoothing, *The International Journal of Advanced Manufacturing Technology* 105 (2019) 457–470. doi:10.1007/s00170-019-04157-4.
- [13] D.-N. Song, J.-W. Ma, Interval partition-based feedrate scheduling with axial drive constraints for five-axis spline toolpaths, *The International Journal of Advanced Manufacturing Technology* 105 (2019) 4701–4714. doi:10.1007/s00170-019-04433-3.
- [14] J.-W. Ma, Z.-Y. Jia, F.-Z. Qin, D.-N. Song, W.-W. Jiang, S.-Y. Chen, A Five-Axis Dual NURBS Interpolator With Constant Speed at Feedrate-Sensitive Regions Under Axial Drive Constraints, *Journal of Manufacturing Science and Engineering* 141. doi:10.1115/1.4043256.
- [15] J. Huang, Y. Lu, L.-M. Zhu, Real-time feedrate scheduling for five-axis machining by simultaneously planning linear and angular trajectories, *International Journal of Machine Tools and Manufacture* 135 (2018) 78–96. doi:10.1016/j.ijmactools.2018.08.006.
- [16] W.-H. Weng, C.-F. J. Kuo, Jerk decision for free-form surface effects in multi-axis synchronization manufacturing, *The International Journal of Advanced Manufacturing Technology* 105 (2019) 799–812. doi:10.1007/s00170-019-04164-5.
- [17] R. T. Farouki, S. Li, Optimal tool orientation control for 5-axis CNC milling with ball-end cutters, *Computer Aided Geometric Design* 30 (2013) 226–239. doi:10.1016/j.cagd.2012.11.003.
- [18] Y. Sun, Y. Bao, K. Kang, D. Guo, A cutter orientation modification method for five-axis ball-end machining with kinematic constraints, *The International Journal of Advanced Manufacturing Technology* 67 (2013) 2863–2874. doi:10.1007/s00170-012-4699-6.
- [19] S. K. Sharma, R. K. Duvedi, S. Bedi, S. Mann, A method for generating multiple solutions for multipoint five-axis tool positioning, *The International Journal of Advanced Manufacturing Technology* 100 (2019) 2509–2520. doi:10.1007/s00170-018-2871-3.
- [20] A. Hashemian, S. F. Hosseini, S. N. Nabavi, Kinematically smoothing trajectories by NURBS reparameterization – an innovative approach, *Advanced Robotics* 31 (2017) 1296–1312. doi:10.1080/01691864.2017.1396923.
- [21] P. Bo, M. Bartoň, H. Pottmann, Automatic fitting of conical envelopes to free-form surfaces for flank CNC machining, *Computer-Aided Design* 91 (2017) 84–94. doi:10.1016/j.cad.2017.06.006.
- [22] L. Piegl, W. Tiller, *The NURBS Book*, 2nd Edition, Springer-Verlag, New York, NY, 1997.
- [23] P. Bo, M. Bartoň, D. Plakhotnik, H. Pottmann, Towards efficient 5-axis flank CNC machining of free-form surfaces via fitting envelopes of surfaces of revolution, *Computer-Aided Design* 79 (2016) 1–11. doi:10.1016/j.cad.2016.04.004.
- [24] P. Bo, M. Bartoň, On initialization of milling paths for 5-axis flank CNC machining of free-form surfaces with general milling tools, *Computer Aided Geometric Design* 71 (2019) 30–42. doi:10.1016/j.cagd.2019.04.012.
- [25] L. Biagiotti, C. Melchiorri, *Trajectory Planning for Automatic Machines and Robots*, Springer-Verlag, Berlin, Heidelberg, 2008.
- [26] S. F. Hosseini, A. Hashemian, A. Reali, On the application of curve reparameterization in isogeometric vibration analysis of free-form curved beams, *Computers & Structures* 209 (2018) 117–129. doi:10.1016/j.compstruc.2018.08.009.
- [27] W. Gautschi, On Generating Gaussian Quadrature Rules, in: G. Hämmerlin (Ed.), *Numerische Integration: Tagung im Mathematischen Forschungsinstitut Oberwolfach*, Birkhäuser Basel, Basel, 1979, pp. 147–154.
- [28] M. Bartoň, V. M. Calo, Gauss–Galerkin quadrature rules for quadratic and cubic spline spaces and their application to isogeometric analysis, *Computer-Aided Design* 82 (2017) 57–67. doi:10.1016/j.cad.2016.07.003.
- [29] M. Bartoň, V. M. Calo, Optimal quadrature rules for odd-degree spline spaces and their application to tensor-product-based isogeometric analysis, *Computer Methods in Applied Mechanics and Engineering* 305 (2016) 217–240. doi:10.1016/j.cma.2016.02.034.
- [30] D. Fußeder, B. Simeon, Algorithmic aspects of isogeometric shape optimization, in: B. Jüttler, B. Simeon (Eds.), *Isogeometric Analysis and Applications 2014*, Springer International Publishing, Cham, 2015, pp. 183–207.
- [31] S. F. Hosseini, B. Moetakef-Imani, S. Hadidi-moud, B. Hassani, Pre-bent shape design of full free-form curved beams using isogeometric method and semi-analytical sensitivity analysis, *Structural and Multidisciplinary Optimization* 58 (2018) 2621–2633. doi:10.1007/s00158-018-2041-0.
- [32] J.-F. Bonnans, J. Gilbert, C. Lemarechal, C. Sagastizábal, *Numerical Optimization, Theoretical and Practical Aspects*, 2nd Edition, Springer-Verlag, Berlin, Heidelberg, 2006.
- [33] J. Nocedal, S. J. Wright, *Numerical Optimization*, Springer-Verlag, New York, NY, 2006.
- [34] H.-Y. Chen, H. Pottmann, Approximation by ruled surfaces, *Journal of Computational and Applied Mathematics* 102 (1999) 143–156. doi:10.1016/S0377-0427(98)00212-X.
- [35] H. Pottmann, J. Wallner, *Computational Line Geometry*, Springer-Verlag, Berlin, Heidelberg, 2001.