Enhancing Smoothness via Redundancy in 3D Laser Cutting Manufacturing: A Collision-Free, Minimized Jerk Trajectory Optimization Approach

Original
Enhancing Smoothness via Redundancy in 3D Laser Cutting Manufacturing: A Collision-Free, Minimized Jerk Trajectory Optimization Approach / Ding, Z., Indri, M., Rizzo, A.. - In: MACHINES. - ISSN 2075-1702. - ELETTRONICO. - 13:5(2025). [10.3390/machines13050339]

Publisher:
MDPI

Published
DOI:10.3390/machines13050339

machines

Article

Enhancing Smoothness via Redundancy in 3D Laser Cutting Manufacturing: A Collision-Free, Minimized Jerk Trajectory Optimization Approach †

Zhipeng Ding [1,2], **Marina Indri** [1,*] and **Alessandro Rizzo** [1,*]

[1] Department of Electronics and Telecommunications, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Turin, Italy; zhipeng.ding@polito.it
[2] EFORT Europe s.r.l., Corso Duca degli Abruzzi 2, 10128 Turin, Italy
* Correspondence: marina.indri@polito.it (M.I.); alessandro.rizzo@polito.it (A.R.)

† This paper is an extended version of our paper published in 2024 IEEE 29th International Conference on Emerging Technologies and Factory Automation (ETFA), Padova, Italy, 10–13 September 2024.

Abstract: In modern manufacturing, achieving high-speed laser cutting requires advanced robotic trajectory planning for smoothness and collision avoidance. Poorly optimized motion can cause frequent velocity changes, leading to mechanical vibrations that shorten machine service life. This study presents an innovative trajectory optimization approach for laser cutting machines equipped with a redundant standoff axis. A B-spline-based analytical model formulates rotational axes trajectories as quadratic programming problems to minimize jerk (the rate of acceleration change) under machining accuracy and kinematic constraints. Additionally, an M path, represented by the wrist center's trajectory, refines translational axes by adjusting the standoff axis through a similar optimization model, thereby reducing mechanical stress. Collision avoidance is ensured through a concurrent iterative optimization process, considering the feasible domains of representative 3D geometric tool orientations. Simulation experiments on a complex B-pillar workpiece demonstrate the framework's effectiveness, clearly indicating significant reductions in jerk and improved trajectory smoothness for both rotational and translational axes compared with conventional methods and a prior approach. This work advances high-speed machining capabilities by offering a novel, robust solution that leverages redundant structures to further improve trajectory smoothness and reliability in demanding industrial applications.

Keywords: laser cutting manufacturing; redundancy; collision avoidance; minimized jerk trajectory optimization; quadratic programming

Academic Editors: Ivan Virgala, Roman Ružarovský and Tibor Horak

Received: 26 February 2025
Revised: 15 April 2025
Accepted: 17 April 2025
Published: 22 April 2025

Citation: Ding, Z.; Indri, M.; Rizzo, A. Enhancing Smoothness via Redundancy in 3D Laser Cutting Manufacturing: A Collision-Free, Minimized Jerk Trajectory Optimization Approach. *Machines* **2025**, *13*, 339. https://doi.org/10.3390/machines13050339

1. Introduction

High-speed and high-precision manufacturing applications, especially in 3D laser cutting manufacturing (LCM) [1], demand trajectory optimization strategies that not only achieve efficiency and accuracy, but also maintain collision-free operations. Leveraging redundancy to enhance smoothness is paramount since abrupt changes in the machining path can induce vibrations, degrade part quality, and intensify mechanical stress and wear [2]. Although modern five-axis CNC machines provide advanced motion-control capabilities, the full integration of minimized-jerk methods with the simultaneous optimization of both tool orientation and translational axes remains a significant challenge, yet it is essential for maximizing performance in 3D LCM tasks.

Conventional five-axis laser cutting machines often struggle to maintain smooth movement, especially when dealing with inertia effects and rapid axis accelerations in complex machining regions. These difficulties stem not only from the mechanical limitations of traditional systems, but also from the computational burden of planning optimal, interference-free tool paths. Existing studies frequently place an excessive focus on optimizing the two rotational axes through tool orientation adjustments for smoothness and collision avoidance. However, the significance of the three translational axes' motion is often underestimated, even though optimizing it is equally crucial for improving overall machining accuracy, reducing dynamic stress, and enhancing the trajectory's smoothness.

1.1. Related Works

Numerous studies on tool orientation adjustment in five-axis machining focus on identifying discrete interference-free regions. They also aim to prevent vibrations caused by abrupt changes between adjacent cutter location (CL) points in the rotational axes. Initially, many of these works [3–5] demonstrated that smoother tool orientation transitions along the tool path within the workpiece coordinate system (WCS) lead to improved trajectories. For example, Ho et al. [3] proposed to represent the orientation of the tool using quaternions and applied a smooth interpolation between the representative directions of the tool axis in the WCS. Lauwers et al. [4] developed a method to evaluate potential collisions and to incrementally adjust tool orientation using inclination and screw angles. Jun et al. [5] developed a search algorithm to minimize machined surface errors in machining of the five-axis sculptured surface. The algorithm determines local optimal tool orientations in the configuration space (C-space) in both forward and backward directions.

However, it was later recognized that planning tool orientations solely within the WCS does not necessarily guarantee smooth rotational axis movements. This is due to the nonlinearity of the inverse kinematic transformation (IKT) used to convert from the WCS to the machine coordinate system (MCS) in a five-axis machine structure [6].

The kinematic performance of the rotational axes can be further enhanced when optimization is performed directly in the MCS, according to Castagnetti et al. [7]. In their work, the feasible domain of tool orientation, known as the domain of admissible orientation (DAO), was defined through four linear inequality constraints. This was achieved using a gradient-based optimization model.

This initial attempt to smooth the rotational axis movements directly in the MCS inspired the development of many smoothing algorithms [2,8–11]. These methods focused on avoiding gouging and collisions along the tool path while enhancing rotational axis motion in five-axis machining. Wang et al. [9] represented the nonlinear kinematic constraints and the irregular geometric feasible domain (GFD) using a linearization strategy, which allowed the optimization problem to be solved using simple linear programming. Plakhotnik and Lauwers [10] proposed a graph-based method that utilized Dijkstra's algorithm to find the shortest path for rotational axis transitions between two adjacent admissible arcs. Xu et al. [12] modeled the kinetic and frictional consumption of the rotational axes on a 3D laser cutting head. They constructed a graph with four combinations of the two rotational axes along discrete tool orientations to search for the shortest path, but the feasible regions are not analyzed. Building on the work of Plakhotnik and Lauwers [10], Mi et al. [11] improved rotational dynamics by incorporating an angular acceleration penalty component into a differential graph.

Most of the aforementioned methods determine the rotational joint coordinates along the continuous tool path using discrete points. Recent research suggests that the use of higher-order differentiable mathematical expressions, such as polynomials [7] or B-splines [13–15], offer better control and smoothness to generate tool orientation trajectories. Huang et al. [16] employed radial basis functions to construct the tool orientation field. Xu et al. [13] developed an iterative representative tool orientation (RTO)-based method to generate a continuous rotational axis trajectory, represented using a B-spline curve in the MCS. The method checks for interference along the entire tool path and iteratively inserts additional RTOs as needed. This approach helps to decouple geometric and kinematic constraints during the tool orientation process. Xiao et al. [14] minimized sudden changes in tool orientation by incorporating contour error prediction constraints. Wu et al. [15] formulated a quadratic programming (QP) problem to compute jerk-optimal B-spline coefficients, with linear constraints derived from kinematic limits and geometric constraints based on the GFD. Their experiments showed enhanced machining stability and surface quality compared to the quaternion interpolation (QI) and angular-acceleration optimal methods.

An alternative method [17] to determine the smooth tool orientation involves synchronized interpolation using dual NURBS curves. Yuen et al. [18] achieved C^3 smoothness trajectories, improving the machining accuracy by minimizing the geometrical jerk. Li et al. [19] incorporated machining error considerations into the dual spline method.

Significant advancements have been made in smoothing tool orientation through various methodologies. Another promising approach to improve trajectory smoothness in robotic systems is to use additional degrees of freedom. Additional degrees of freedom in redundant robotic systems allow trajectory planning and execution to be refined, achieving greater smoothness and precision. A promising redundancy strategy is known as macro–micro, which combines a macro-scale robot for large movements with a micro-scale robot for precision [20]. Enhancing dexterity and precision through redundancy, this configuration has been widely used in industries for manipulation [21], robotic machining [22,23], and medical surgery [24]. In laser cutting applications, which require speed and precision over large areas, the macro–micro structure is highly advantageous. Uzunoglu [25] optimized machining by distributing planar paths between primary x-y and secondary u-v mechanisms. Liu [26] demonstrated efficiency improvements using a velocity planning algorithm for macro and micro platforms. However, most laser cutting applications are limited to planar surfaces and lack consideration for complex 3D five-axis machining applications. In our previous work [27], we proposed a conic posture interpolation method to cut closed-contour holes with a "polar" robot on the laser head, achieving improved efficiency. Substantial opportunities remain to optimize trajectories by utilizing the macro–micro redundant structure in laser cutting applications.

1.2. Contributions

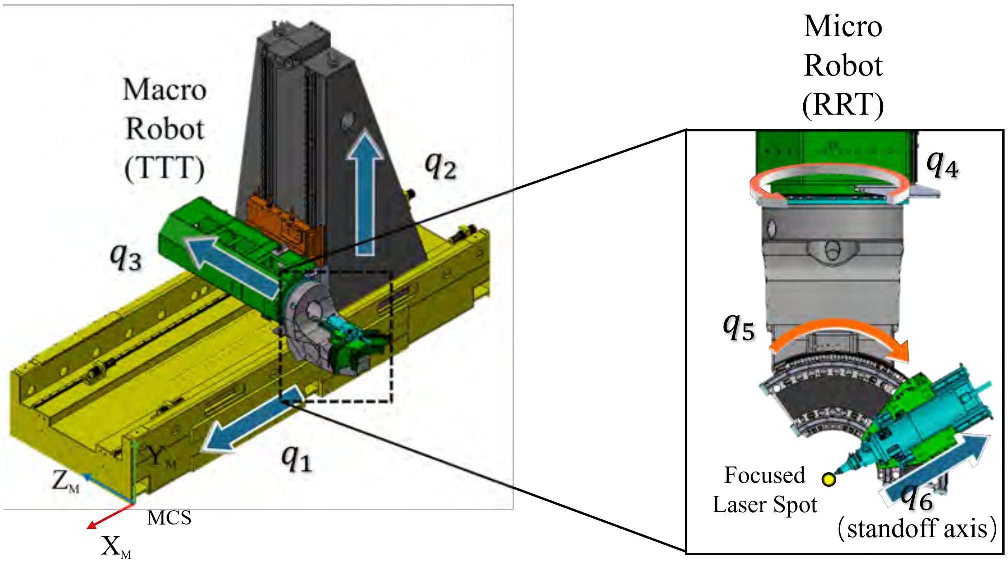
While significant advancements have been made in optimizing tool orientation and collision avoidance for five-axis machining, trajectory planning methods specifically targeting smoothness optimization along translational axes have received comparatively less attention. To address this gap, our prior work [28] introduced a two-step optimization approach, employing a graph-based algorithm for collision-free tool orientation planning, followed by reinforcement learning to optimize the redundant standoff axis. Although effective in specific scenarios, these methods were computationally expensive [10,12] due to the high dimensionality of the GFD required for graph construction and extensive collision checks. Additionally, numerical errors introduced by discrete differentiation methods in trajectory optimization increased substantially with higher-order derivatives, negatively impacting trajectory accuracy and smoothness.

To overcome these limitations, this work proposes a novel trajectory optimization strategy tailored specifically for 3D laser cutting machines equipped with a redundant standoff axis. The primary contribution of our research lies in developing an innovative optimization framework inspired by macro–micro structural approaches. This framework explicitly leverages the redundant axis to optimize translational axis trajectories jointly with tool orientations—an aspect rarely explored in the existing literature. A prototype of this machine, currently under development by EFORT Intelligent Robot Co., Ltd. (Wuhu, China), serves as the basis for our study (see schematic and axis definitions in Figure 1).

Different from discrete graph-based methods, our proposed methodology uses an analytical joint trajectory model represented by B-splines, formulated explicitly as a QP problem. This analytical formulation significantly reduces computational complexity, avoiding numerical errors typically associated with high-order discrete differentiation. Moreover, by explicitly minimizing jerk instead of acceleration, the proposed minimized jerk trajectory optimization (MJTO) algorithm directly enhances smoothness by effectively mitigating abrupt changes in acceleration.

In addition, while existing iterative algorithms using RTO-based methods [3,13,15] often successfully find collision-free trajectories, they typically overlook optimizing initial RTO selections. Recognizing the critical impact of initial RTO configurations on trajectory quality, our study introduces a dedicated optimization process for strategically selecting initial RTOs. This dedicated selection process substantially improves trajectory smoothness, as well as the efficiency and effectiveness of the overall optimization.

Lastly, the high-speed, non-contact characteristics inherent in 3D laser cutting manufacturing frequently involve rapid and extensive tool orientation changes when machining complex geometries. Such large-angle and high-rate transitions impose substantial demands on trajectory smoothness and collision avoidance, challenges inadequately addressed by traditional methods. The MJTO approach developed herein specifically tackles these demanding conditions, ensuring smooth, jerk-minimized trajectories and reliable collision-free operations, thus effectively accommodating the challenging orientation variations that characterize advanced industrial laser cutting applications.



(a) (b)

Figure 1. The schematic of the EFORT 3D laser cutting machine. (**a**) Overall view, showing the macro robot (TTT) with three translational axes q_1, q_2, q_3 (T for translation). (**b**) Close-up of the micro robot (RRT), featuring the two rotational axes q_4, q_5 (R for rotation) and the redundant standoff axis q_6.

1.3. Structure of the Paper

This paper is organized into four sections. Section 2 introduces the minimized jerk trajectory optimization framework based on B-splines, first formulating a quadratic programming model to enhance the smoothness of joint trajectories, particularly for 3D laser cutting machines equipped with a redundant standoff axis. It then integrates an optimization RTO-based iterative procedure to ensure collision-free trajectory. Section 3 presents two illustrative case studies involving automotive B-pillar components, demonstrating significant improvements in trajectory smoothness and overall effectiveness compared with conventional methods and our earlier graph-based optimization approach. Finally, Section 4 summarizes the major contributions and suggests future research directions.

2. MJTO Methodology: Minimized Jerk Trajectory Optimization

This section introduces the proposed MJTO algorithm for 3D laser cutting machines equipped with a redundant standoff axis. The approach utilizes B-spline curves [13] to represent the joint displacement and ensure trajectory continuity and smoothness while expressing kinematic constraints [15]. Known as minimized jerk trajectory optimization, the method formulates the optimization problem as two QP tasks. A QP model based on B-splines is developed for the two rotational axes to minimize jerk, thereby enhancing trajectory smoothness and machining precision. Similarly, the redundant standoff axis is modeled within a QP framework to further smooth the rotational axes center point path, as referred to the M path in our previous work [28], and improve overall system performance. In addition, a collision avoidance strategy is introduced. It is based on the GFD in 3D joint space and an iterative RTO-based process. This ensures that the trajectory remains collision-free while satisfying kinematic constraints. The following subsections outline the key components of the MJTO methodology, including the mathematical representation of joint trajectories and M path using B-splines, the formulation of QP problems for rotational axes and the redundant axis, and the integration of collision avoidance mechanisms.

Firstly, this study builds on the trajectory representation framework developed in our previous work [28]. The laser head follows the CL point trajectory $\mathbf{C}(u) = \left[C_x(u), C_y(u), C_z(u)\right]^T \in \mathbb{R}^3$, which defines the tool path in Cartesian space, and the tool orientation $\mathbf{O}(u) = \left[O_x(u), O_y(u), O_z(u)\right]^T \in \mathbb{R}^3$, a unit vector representing the tool's orientation direction. Both are defined in the workpiece coordinate system (WCS), as illustrated in Figure 2a. Here, the trajectories are parameterized using the normalized arc length parameter u, which is generated by a standard G-code interpolator with demanding machining parameters as

$$u(t) = \frac{s(t)}{\sigma} \quad \forall u \in [0, 1], \tag{1}$$

where $s(t)$ denotes the arc length with respect to time t and σ is the total length of the tool path.

The concept of the second M point $\mathbf{M}(u) = \left[M_x(u), M_y(u), M_z(u)\right]^T \in \mathbb{R}^3$ is introduced here, as shown in Figure 2b. This auxiliary point is used to characterize and enhance the smoothness of the primary translational axes during trajectory planning. Figure 2c illustrates the relationship between coordinate systems with respect to the joint coordinates $\mathbf{q} = \left[q_1, q_2, q_3, q_4, q_5, q_6\right]^T \in \mathbb{R}^6$ defined as in Figure 1. The forward kinematic equations, which transform joint coordinates $\mathbf{q}(u)$ in MCS into WCS, are given by:

$$\begin{cases} C_x(u) = q_1(u) + q_6(u)\cos q_4(u)\sin q_5(u) \\ C_y(u) = q_2(u) + q_6(u)\sin q_4(u)\sin q_5(u) \\ C_z(u) = q_3(u) + q_6(u)\cos q_5(u) \\ O_x(u) = \cos q_4(u)\sin q_5(u) \\ O_y(u) = \sin q_4(u)\sin q_5(u) \\ O_x(u) = \cos q_5(u) \end{cases}, \tag{2}$$

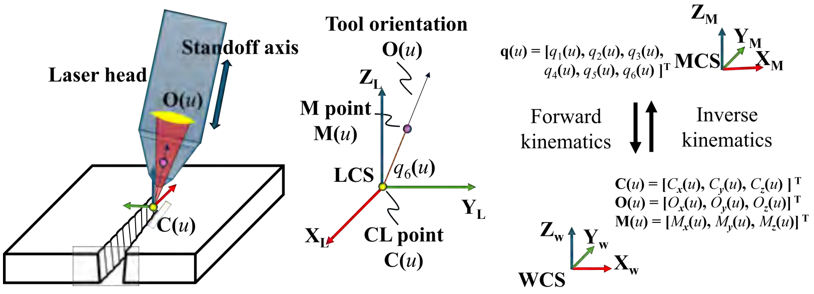
reversely, the conversion from WCS back to MCS, denoted as IKT, is formulated as

$$\begin{cases} q_1(u) = C_x(u) - q_6(u)\cos(q_4(u))\sin(q_5(u)) \\ q_2(u) = C_y(u) - q_6(u)\sin(q_4(u))\sin(q_5(u)) \\ q_3(u) = C_z(u) - q_6(u)\cos(q_5(u)) \\ q_4(u) = \arctan 2(O_x(u), O_y(u)) + k\pi, \quad \text{with } k = 0, \pm 1 \\ q_5(u) = \arccos(O_z(u)) \end{cases}, \tag{3}$$

where the $\arctan 2(\cdot)$ function determines the angle by considering the signs of $O_x(u)$ and $O_y(u)$.

We note that u is not equivalent to the time variable t. The relationship between them is governed by the feedrate $\dot{s}(t)$ along the path and can be derived using (1) as

$$\frac{du}{dt} = \frac{ds}{dt} \cdot \frac{1}{\sigma} = \frac{\dot{s}(t)}{\sigma}. \tag{4}$$



(a) (b) (c)

Figure 2. (**a**) Laser head trajectory in the Local Coordinate System (LCS); (**b**) relationship between CL point, tool orientation, and M point in the LCS; (**c**) forward and inverse kinematic transformations between the Machine Coordinate System (MCS) and the Workpiece Coordinate System (WCS).

2.1. B-Spline Representation of Joint Space and Kinematic Constraints

The proposed MJTO algorithm uses B-spline curves to represent rotational axis displacement trajectories. Their smoothness and flexibility make them suitable for modeling complex trajectories, as they ensure continuity and differentiability up to a desired order [29]. Therefore, the displacement trajectories of the rotational axes $q^*(u)$, for $* = 4$ or 5, can be mathematically represented as

$$q^*(u) = \sum_{j=0}^{n} \Theta_j^* N_{j,k}(u), \tag{5}$$

where $N_{j,k}(u)$ are the B-spline basis functions of degree k, Θ_j^* is the control coefficient, and $n+1$ is the number of the control coefficients.

The choice of degree of B-spline k determines the smoothness of the trajectory. A quintic B-spline ($k = 5$) is typically chosen to ensure C^3 continuity since third-order differentiability helps to minimize abrupt changes in acceleration. The B-spline basis function $N_{j,k}(u)$ can be derived recursively using a knot vector $\mathbf{U}$. A uniform method is employed for generating the knot vector, ensuring $(n+1)$ control points along the tool path. Consequently, the collection of control coefficients $\{\Theta_j^*\}_{j=0}^{n}$ becomes the set of decision variables in (5), to be optimized in the jerk-minimization process.

Tool path interpolation is performed in the WCS, considering Cartesian tangential dynamic parameters with u. However, joint kinematic constraints are typically ignored during this stage, which can result in abrupt or non-smooth joint motions. Therefore, using the chain rule, the kinematic parameters of the joints, including angular velocity $\omega^*(t)$, angular acceleration $\alpha^*(t)$, and angular jerk $J^*(t)$, are derived incorporating tangential dynamic parameters as

$$\begin{cases} \omega^*(t) = \frac{dq^*}{dt} = \frac{dq^*}{du}\frac{du}{dt} \\ \alpha^*(t) = \frac{d^2q^*}{dt^2} = \frac{d^2q^*}{du^2}\left(\frac{du}{dt}\right)^2 + \frac{dq^*}{du}\frac{d^2u}{dt^2} \\ J^*(t) = \frac{d^3q^*}{dt^3} = \frac{d^3q^*}{du^3}\left(\frac{du}{dt}\right)^3 + 3\frac{d^2q^*}{du^2}\frac{du}{dt}\frac{d^2u}{dt^2} + \frac{dq^*}{du}\frac{d^3u}{dt^3} \end{cases}. \tag{6}$$

The kinematic parameters can then be reformulated using (4) as

$$\begin{cases} \omega^*(t) = q_u^*(u)\frac{\dot{s}(t)}{\sigma} \\ \alpha^*(t) = q_{uu}^*(u)\left(\frac{\dot{s}(t)}{\sigma}\right)^2 + q_u^*\frac{\ddot{s}(t)}{\sigma} \\ J^*(t) = q_{uuu}^*(u)\left(\frac{\dot{s}(t)}{\sigma}\right)^3 + 3q_{uu}^*\frac{\dot{s}(t)\cdot\ddot{s}(t)}{\sigma^2} + q_u^*\frac{\dddot{s}(t)}{\sigma} \end{cases}, \tag{7}$$

where $\ddot{s}(t)$ and $\dddot{s}(t)$ represent the first and second deviates of feedrate with respect to time, respectively. Similarly, $q_u^*(u)$, $q_{uu}^*(u)$, $q_{uuu}^*(u)$ denote the first, second, and third geometricalderivatives of q^* with respect to u. Hence, q_u^* captures how the joint variable changes along the path from a purely geometric standpoint, whereas the feedrate $\dot{s}(t)$ describes how fast we traverse that path in time. The geometrical derivatives of the joint displacement are given below as

$$q_u^*(u) = \sum_{j=0}^{n} \Theta_j^* N'_{j,k}(u), \tag{8}$$

$$q_{uu}^*(u) = \sum_{j=0}^{n} \Theta_j^* N''_{j,k}(u), \tag{9}$$

$$q_{uuu}^*(u) = \sum_{j=0}^{n} \Theta_j^* N'''_{j,k}(u). \tag{10}$$

However, to impose linear constraints on the geometrical derivatives in terms of the design variable Θ_j, we can reshape the geometrical derivatives in (8)–(10) in a general form for the r-th derivative, following De Boor's algorithm for B-splines [15], as

$$q^*_{\underbrace{u\ldots u}_{r}}(u) = \sum_{j=i-k}^{i-r} \Theta_j^{*,r} N_{j,k-r}(u), \tag{11}$$

where $u \in [u_i, u_{i+1}] \subset [u_k, u_{n+1}]$, u_i is the interpolated node assigned on the CL point $\mathbf{C}(u_i)$, and the geometrical derivative of the control coefficient can be computed recursively as

$$\begin{cases} \Theta_j^{*,0} = \Theta_j^* \\ \Theta_j^{*,l} = \frac{k-l+1}{u_{j+k+1} - u_{j+1}}\left(\Theta_{j+1}^{*,l-1} - \Theta_j^{*,l-1}\right) \\ \text{for} \quad l = 1, 2, \ldots, r; j = i-k, i-k+1, \ldots, i-r \end{cases}, \tag{12}$$

where $\{\Theta_j^{*,r}\}_{j=0}^{n-r}$ are the control coefficients for the r-th geometric derivative of $q^*(u)$.

In many theoretical treatments, a constant feedrate (i.e., $\dot{s}(t) = f$) is assumed. This simplification is also adopted here to ease the parametric forms. However, in the application of practical machining, a variable value is assigned to feedrate, which is influenced by multiple factors, such as drive limitations, variations in machining trajectories, and specific manufacturing process requirements. In the current study, we assume a constant feedrate, as in [13,15,30], to highlight the core mechanisms of the minimized jerk method. Under this assumption, the kinematic parameters in (6) can be reformulated using (4) as

$$\begin{cases} \omega^*(t) = q_u^*(u)\dfrac{f}{\sigma} \\ \alpha^*(t) = q_{uu}^*(u)\dfrac{f^2}{\sigma^2} \\ J^*(t) = q_{uuu}^*(u)\dfrac{f^3}{\sigma^3} \end{cases}, \tag{13}$$

where we observe that the joint velocities, accelerations, and jerks are scaled by powers of the constant feedrate f.

Furthermore, these can be expanded by integrating the B-spline form in (11) for $\forall u \in [u_i, u_{i+1}] \subset [u_k, u_{n+1}]$ into (5), yielding

$$\begin{cases} \omega^*(t) = \displaystyle\sum_{j=i-k}^{i-1} \Theta_j^{*,1} N_{j,k-1}(u) \cdot \frac{f}{\sigma} \\ \alpha^*(t) = \displaystyle\sum_{j=i-k}^{i-2} \Theta_j^{*,2} N_{j,k-2}(u) \cdot \frac{f^2}{\sigma^2} \\ J^*(t) = \displaystyle\sum_{j=i-k}^{i-3} \Theta_j^{*,3} N_{j,k-3}(u) \cdot \frac{f^3}{\sigma^3} \end{cases}. \tag{14}$$

To guarantee that the resulting trajectories remain within the feasible ranges of robot kinematics, we impose constraints on the rotational axes, including limits on angular velocity $\omega_{\max}^*$, acceleration $\alpha_{\max}^*$, jerk $J_{\max}^*$, and the axis range $[\, q_{\min}^*, q_{\max}^* \,]$. Owing to the convex hull property of B-spline curves, the resulting curve is guaranteed to lie entirely within the convex region defined by its control points. Therefore, by transferring the constant factor into the constraint formulation, the kinematic restrictions can be linearly expressed in terms of the control coefficients Θ_j^* and their geometrical derivatives $\Theta_j^{*,r}$ as

$$\begin{cases} q_{\min}^* \le \Theta_j^* \le q_{\max}^* \\ -\omega_{\max}^* \cdot \dfrac{\sigma}{f} \le \Theta_j^{*,1} \le \omega_{\max}^* \cdot \dfrac{\sigma}{f} \\ -a_{\max}^* \cdot \dfrac{\sigma^2}{f^2} \le \Theta_j^{*,2} \le \alpha_{\max}^* \cdot \dfrac{\sigma^2}{f^2} \\ -J_{\max}^* \cdot \dfrac{\sigma^3}{f^3} \le \Theta_j^{*,3} \le J_{\max}^* \cdot \dfrac{\sigma^3}{f^3} \end{cases}, \tag{15}$$

thus ensuring the desired bounds on velocities, accelerations, and jerks are not exceeded. These linearized constraints, together with the recursive relation in (12), provide a framework amenable to kinematic constraints using the B-spline representation. The subsequent section details how these constraints and objective functions are integrated into our proposed minimized jerk algorithm.

2.2. QP Minimized Jerk Modeling for Rotational Joint Trajectory Based on B-Spline

Achieving a smooth trajectory is crucial in high-precision 3D laser cutting applications, directly influencing the quality of the machining and the mechanical stress. Minimization of the jerk is an effective optimization objective, as it reduces abrupt motion changes, resulting in smoother tool movements [15,18].

Minimization of the jerk can be formulated as a QP problem subject to the kinematic constraints listed in (15). This ensures computational efficiency and guarantees the attainment of a global optimum due to the convexity of the QP, allowing efficient computation with standard solvers.

Therefore, the objective function J_{R}^* for minimizing jerk on rotational axes can first be formally defined as the integral of squared jerk with respect to time over the total trajectory duration T as

$$\min_{\Theta_j} J_{\mathrm{R}}^* = \int_0^T (J^*(t))^2 dt. \tag{16}$$

By using the relationship between time t and the normalized arc-length parameter u given by (4), the integral domain can be transformed from the time domain to the parameter domain u. Incorporating (13) and assuming a constant feedrate f, (16) can be rewritten as

$$\begin{aligned}
\min_{\Theta_j} J_{\mathrm{R}}^* &= \int_{u(0)}^{u(T)} \left(\frac{f^3}{\sigma^3} q_{uuu}^*(u) \right)^2 \frac{\sigma}{f} du \\
&= \frac{\sigma}{f} \int_0^1 \left(\frac{f^3}{\sigma^3} \sum_{j=0}^{n} \Theta_j^* N_{j,k}'''(u) \right)^2 du \\
&= \frac{f^5}{\sigma^5} \int_0^1 \sum_{i=0}^{n} \sum_{j=0}^{n} \Theta_i^* \Theta_j^* N_{i,k}'''(u) N_{j,k}'''(u) du \\
&= \frac{f^5}{\sigma^5} \sum_{i=0}^{n} \sum_{j=0}^{n} \left(\int_0^1 N_{i,k}'''(u) N_{j,k}'''(u) du \right) \boldsymbol{\Theta}_i^* \boldsymbol{\Theta}_j^*.
\end{aligned} \tag{17}$$

Defining the integral term as a scalar coefficient h_{ij}, which is nonzero only when $|j-i| \le k$, we have

$$h_{ij} = \begin{cases} \int_0^1 N_{i,k}'''(u) N_{j,k}'''(u) du, & |j-i| \le k \\ 0, & |j-i| > k \end{cases}, \quad i,j = 0,1,\ldots,n. \tag{18}$$

Observing that the constant scaling factor f^5/σ^5 does not influence the optimization search direction, we can safely omit this factor. Thus, by introducing the symmetric Hessian matrix $\mathbf{H}$ with elements h_{ij}, the objective function can be compactly expressed in standard quadratic programming form with respect to the control coefficient vector $\boldsymbol{\Theta}^*$ as

$$\min_{\boldsymbol{\Theta}^*} J_{\mathrm{R}}^* = \boldsymbol{\Theta}^{*T} \mathbf{H} \boldsymbol{\Theta}^*. \tag{19}$$

The kinematic constraints outlined in (15) play a critical role in ensuring the feasibility of the trajectory and are incorporated into the QP problem as inequality constraints. Additionally, the states of kinematic parameters at $u(0) = 0$ and $u(T) = 1$, corresponding to the initial and final trajectory conditions, must be enforced as equality constraints. These constraints ensure continuity and smooth transitions while adhering to specified boundary values, such as ω_0^*, ω_1^*, α_0^*, and α_1^*. The complete trajectory optimization problem is then expressed as a kinematic constrained QP problem, formulated as

$$\begin{cases} \min\limits_{\boldsymbol{\Theta}^*} \quad J_{\mathrm{R}}^* = \boldsymbol{\Theta}^{*T}\mathbf{H}\boldsymbol{\Theta}^* \\ \text{s.t.} \quad q_u^*(0) = \omega_0^*, \quad q_{uu}^*(0) = \alpha_0^* \\ \qquad q_u^*(1) = \omega_1^* \quad q_{uu}^*(1) = \alpha_1^* \\ \qquad q_{\min}^* \le \Theta_j^* \le q_{\max}^* \\ -\omega_{\max}^* \cdot \dfrac{\sigma}{f} \le \Theta_j^{*,1} \le \omega_{\max}^* \cdot \dfrac{\sigma}{f} \\ -\alpha_{\max}^* \cdot \dfrac{\sigma^2}{f^2} \le \Theta_j^{*,2} \le \alpha_{\max}^* \cdot \dfrac{\sigma^2}{f^2} \\ -J_{\max}^* \cdot \dfrac{\sigma^3}{f^3} \le \Theta_j^{*,3} \le J_{\max}^* \cdot \dfrac{\sigma^3}{f^3} \\ \qquad \text{for } j = 0, 1, \ldots, n. \end{cases} \tag{20}$$

As discussed in (12), linearization and recursive computation of control coefficients are crucial for optimization modeling. The linearization of kinematic constraints, detailed in Wu et al. [15], is further simplified here for practical implementation. The geometrical derivatives of the decision variables are expressed as linear combinations of Θ_j^*, as follows

$$\Theta_j^{*,1} = c_j^1\Theta_j^* - c_j^1\Theta_{j+1}^*, \tag{21}$$

$$\Theta_j^{*,2} = c_j^2\Theta_j^* - 2c_j^2\Theta_{j+1}^* + c_j^2\Theta_{j+2}^*, \tag{22}$$

$$\Theta_j^{*,3} = c_j^3\Theta_j^* - 3c_j^3\Theta_{j+1}^* + 3c_j^3\Theta_{j+2}^* - c_j^3\Theta_{j+3}^*, \tag{23}$$

where c_j^1, c_j^2, c_j^3 are computed using the recursive formulation in (12), as

$$c_j^1 = \frac{-k}{u_{j+k+1} - u_{j+1}}, \tag{24}$$

$$c_j^2 = \frac{k^2 - k}{(u_{j+k+1} - u_{j+2})(u_{j+k+1} - u_{j+1})}, \tag{25}$$

$$c_j^3 = \frac{-k^3 + 3k^2 - 2k}{(u_{j+k+1} - u_{j+3})(u_{j+k+1} - u_{j+2})(u_{j+k+1} - u_{j+1})}. \tag{26}$$

Consequently, the kinematic equality and inequality constraints, encompassing eight kinematic conditions, are reformulated into a concise vector representation as

$$\mathbf{A}_{\mathrm{kin}}^*\boldsymbol{\Theta}^* = \mathbf{b}_{\mathrm{kin}}^*, \tag{27}$$

$$\mathbf{A}_q^*\boldsymbol{\Theta}^* \le \mathbf{b}_q^*, \tag{28}$$

$$\mathbf{A}_\omega^*\boldsymbol{\Theta}^* \le \mathbf{b}_\omega^*, \tag{29}$$

$$\mathbf{A}_\alpha^*\boldsymbol{\Theta}^* \le \mathbf{b}_\alpha^*, \tag{30}$$

$$\mathbf{A}_J^*\boldsymbol{\Theta}^* \le \mathbf{b}_J^*, \tag{31}$$

$$-\mathbf{A}_\omega^*\boldsymbol{\Theta}^* \le \mathbf{b}_\omega^*, \tag{32}$$

$$-\mathbf{A}_\alpha^*\boldsymbol{\Theta}^* \le \mathbf{b}_\alpha^*, \tag{33}$$

$$-\mathbf{A}_J^*\boldsymbol{\Theta}^* \le \mathbf{b}_J^*. \tag{34}$$

The variables in constraints (27)–(34) are defined as follows:

$\mathbf{A}^*_{\mathbf{kin}}$: It is a $4 \times (n+1)$ matrix with elements at the initial and final states, representing the derivatives of the coefficient of the basis functions for equality constraints.

$\mathbf{b}^*_{\mathbf{kin}}$: It is a 4×1 column vector representing the kinematic boundary values.

$\mathbf{A}^*_q$: It is a $(2n+2) \times (n+1)$ coefficient matrix, defined as $\begin{bmatrix} \mathbf{N} \\ -\mathbf{N} \end{bmatrix}$, where $\mathbf{N}$ is a $(n+1) \times (n+1)$ B-spline basis function matrix with elements $n_{i,j}$.

$\mathbf{b}^*_q$: It is a $(2n+2) \times 1$ column vector representing position limits.

$\mathbf{A}^*_\omega$: It is a symmetric $(n-k) \times (n+1)$ matrix with elements $a^{\omega,*}_{i,j}$, representing velocity constraints.

$\mathbf{b}^*_\omega$: It is an $(n-k) \times 1$ vector representing velocity limits.

$\mathbf{A}^*_\alpha$: It is a symmetric $(n-k-1) \times (n+1)$ matrix, with elements $a^{\alpha,*}_{i,j}$, representing acceleration constraints.

$\mathbf{b}^*_\alpha$: It is an $(n-k-1) \times 1$ vector representing acceleration limits.

$\mathbf{A}^*_J$: It is a symmetric $(n-k-2) \times (n+1)$ matrix, with elements $a^{J,*}_{i,j}$, representing jerk constraints.

$\mathbf{b}^*_J$: It is an $(n-k-2) \times 1$ vector representing jerk limits.

According to (24)–(26), the detailed expressions for the elements of the coefficient matrices are given as

$$n_{i,j} = \begin{cases} N_{i,k}(u_i), & |j-i| \le k \\ 0, & \text{otherwise} \end{cases} \tag{35}$$

$$\text{for } i = 0,1,\ldots,n;\ j = 0,1,\ldots,n,$$

$$a^{w,*}_{i,j} = \begin{cases} c^1_i, & j = i \\ -c^1_i, & j = i+1 \\ 0, & \text{otherwise} \end{cases} \tag{36}$$

$$\text{for } i = 0,1,\ldots,n-k;\ j = 0,1,\ldots,n,$$

$$a^{\alpha,*}_{i,j} = \begin{cases} c^2_i, & j = i \\ -2c^2_i, & j = i+1 \\ c^2_i, & j = i+2 \\ 0, & \text{otherwise} \end{cases} \tag{37}$$

$$\text{for } i = 0,1,\ldots,n-k-1;\ j = 0,1,\ldots,n,$$

$$a^{J,*}_{i,j} = \begin{cases} c^3_i, & j = i \\ -3c^3_i, & j = i+1 \\ 3c^3_i, & j = i+2 \\ -c^3_i, & j = i+3 \\ 0, & \text{otherwise} \end{cases} \tag{38}$$

$$\text{for } i = 0,1,\ldots,n-k-2;\ j = 0,1,\ldots,n.$$

The right-hand-side vectors with terms of scaled boundary kinematic parameters are detailed as follows:

$$\mathbf{b}_q^* = \left[\underbrace{q^*_{\mathrm{max}} \cdots q^*_{\mathrm{max}}}_{n+1}, \underbrace{-q^*_{\mathrm{min}} \cdots -q^*_{\mathrm{min}}}_{n+1} \right]^T, \tag{39}$$

$$\mathbf{b}_\omega^* = \left[\underbrace{(\sigma/f) \cdot \omega^*_{\mathrm{max}} \quad \cdots \quad (\sigma/f) \cdot \omega^*_{\mathrm{max}}}_{n-k} \right]^T, \tag{40}$$

$$\mathbf{b}_\alpha^* = \left[\underbrace{(\sigma^2/f^2) \cdot \alpha^*_{\mathrm{max}} \quad \cdots \quad (\sigma^2/f^2) \cdot \alpha^*_{\mathrm{max}}}_{n-k-1} \right]^T, \tag{41}$$

$$\mathbf{b}_J^* = \left[\underbrace{(\sigma^3/f^3) \cdot J^*_{\mathrm{max}} \quad \cdots \quad (\sigma^3/f^3) \cdot J^*_{\mathrm{max}}}_{n-k-2} \right]^T. \tag{42}$$

Thus, the kinematic equality and inequality constraints can be summarized into a compact form for the QP problem as

$$\begin{cases} \min\limits_{\boldsymbol{\Theta}^*} \quad J_{\mathrm{R}}^* = \boldsymbol{\Theta}^{*T} \mathbf{H} \boldsymbol{\Theta}^* \\ \text{s.t.} \quad \mathbf{A}^*_{\mathrm{ineq}} \boldsymbol{\Theta}^* \leq \mathbf{b}^*_{\mathrm{ineq}} \\ \qquad \mathbf{A}^*_{\mathrm{kin}} \boldsymbol{\Theta}^* = \mathbf{b}^*_{\mathrm{kin}} \end{cases}, \tag{43}$$

where the left-side matrices and right-side columns' vector are constructed as follows

$$\mathbf{A}^*_{\mathrm{ineq}} = \begin{bmatrix} \mathbf{A}^*_q \\ \mathbf{A}^*_\omega \\ \mathbf{A}^*_\alpha \\ \mathbf{A}^*_J \\ -\mathbf{A}^*_\omega \\ -\mathbf{A}^*_\alpha \\ -\mathbf{A}^*_J \end{bmatrix}, \qquad \mathbf{b}^*_{\mathrm{ineq}} = \begin{bmatrix} \mathbf{b}^*_q \\ \mathbf{b}^*_\omega \\ \mathbf{b}^*_\alpha \\ \mathbf{b}^*_J \\ \mathbf{b}^*_\omega \\ \mathbf{b}^*_\alpha \\ \mathbf{b}^*_J \end{bmatrix}, \tag{44}$$

$$\mathbf{A}^*_{\mathrm{kin}} = \begin{bmatrix} N'_{0,k}(0) & N'_{1,k}(0) & \ldots & N'_{n,k}(0) \\ N'_{0,k}(1) & N'_{1,k}(1) & \ldots & N'_{n,k}(1) \\ N''_{0,k}(0) & N''_{1,k}(0) & \ldots & N''_{n,k}(0) \\ N''_{0,k}(1) & N''_{1,k}(1) & \ldots & N''_{n,k}(1) \end{bmatrix}, \qquad \mathbf{b}^*_{\mathrm{kin}} = \begin{bmatrix} \omega^*_0 \\ \omega^*_1 \\ \alpha^*_0 \\ \alpha^*_1 \end{bmatrix}. \tag{45}$$

The kinematic constraints of the spline curve have been formulated into the optimization problem in (43). However, the entire curve must satisfy geometric constraints $O(u) \in$ GFDs, which are nonlinear, to ensure machining accuracy and prevent collisions along the tool path. The updated via points on the planning curve are transformed using the RTO-based iterative method and incorporated as equality constraints in the QP problem, as described in Section 2.4.

2.3. QP Modeling for Redundant Joints Based on B-Spline for Optimal M Path Jerk

While the optimization problem for rotational axes has been effectively modeled using B-spline curves in Section 2.2, ensuring smooth motion along translational axes is equally critical for precision and lifespan in 3D laser cutting machining. To achieve smooth translational trajectories, we have introduced the concept of the M path here in our previous work [28]. The M path deviates from the direct tool path by introducing a redundant standoff axis q_6 along the tool direction. The spatial transformation of the M point $\mathbf{M}(u)$ is expressed as

$$\mathbf{M}(u) = \mathbf{C}(u) + q_6(u)\mathbf{O}_{\text{opt}}(u), \tag{46}$$

where $\mathbf{O}_{\text{opt}}(u)$ denotes the optimized tool orientation, computed via forward kinematics (2) based on the optimized rotational joint values $q_4(u)$ and $q_5(u)$.

To clarify the relationship between the M path and the translational axes, it is important to note that, in the kinematic model, the rotational axes q_4 and q_5 primarily define the end-effector's orientation, while the M point $\mathbf{M}(u)$ depends on the translational axes and the redundant standoff axis q_6. In other words, $\mathbf{M}(u)$ is decoupled from q_4 and q_5, since its position is derived by linear mapping from the translational joints (q_1, q_2, q_3) in combination with the offset q_6. Consequently, the B-spline trajectory of $\mathbf{M}(u)$ directly influences the motion of the translational axes. Ensuring the smoothness of the M path in Cartesian space consequently leads to smoother joint commands for the translational axes, as the transformation from $\mathbf{C}(u)$ to $\mathbf{M}(u)$ is linear with respect to the joint coordinate $q_6(u)$ combined with $\mathbf{O}(u)$. This decoupling allows for the leveraging of a minimized jerk strategy for the rotational axes, while separately optimizing and refining the M path to maintain smooth translational movement and prevent abrupt accelerations or vibrations in the three translational axes.

The redundant axis introduces an additional degree of freedom, enhancing trajectory smoothness and flexibility. This enhancement is achieved through a reinforcement learning-based exploration method. It dynamically adjusts the wrist center point position along the tool orientation. While reinforcement learning improves adaptability, the M path smoothness can be further enhanced. Specifically, quintic B-spline curves, akin to those used for rotational axes, are employed to define the M path $\mathbf{M}(u)$ as

$$\mathbf{M}(u) = \sum_{j=0}^{n} \mathbf{P}_j^{\dagger} N_{j,k}(u), \tag{47}$$

where $\{\mathbf{P}_j^{\dagger}\}_{j=0}^{n}$ represents a set of $n+1$ control points for the B-spline curve, organized as an $(n+1) \times 3$ matrix to define the geometric control points in three-dimensional space, as illustrated in Figure 3.

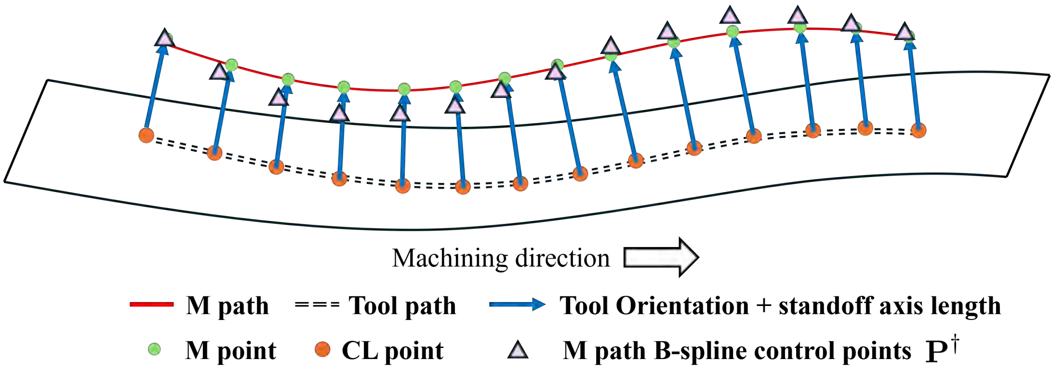


Figure 3. Schematic diagram of the introduced M path represented by a B-spline curve and its control points $\mathbf{P}_j^{\dagger}$ to be optimized.

Similar to (19), the integrated jerk along the entire M path is minimized through a quadratic objective function. It is expressed in terms of $\mathbf{P}^{\dagger}$ as

$$\min_{\mathbf{P}^{\dagger}} J_{\mathrm{M}} = {\mathbf{P}^{\dagger}}^{T} \mathbf{H} \mathbf{P}^{\dagger}. \tag{48}$$

where the Hessian matrix $\mathbf{H}$ is the same used in (18), as the B-spline knot vector $\mathbf{U}$ is unchanged.

The geometric constraint of the optimization problem mandates that the B-spline curve $\mathbf{M}(u)$ passes through the M points $\mathbf{M}_i$ at interpolated arc-length parameters u_i; the set of $\mathbf{M}_i$ can be defined using the B-spline in matrix form with the basis function coefficient matrix $\mathbf{N}$ as

$$\mathbf{M} = \mathbf{N}\mathbf{P}^{\dagger}, \tag{49}$$

where $\mathbf{M}$ is an $(n+1) \times 3$ matrix, in which each row corresponds to a single M point $\mathbf{M}_i$.

As $\mathbf{M}_i$ depends on the unknown standoff axis length w_i at u_i, the optimization problem becomes nonlinear and cannot be directly solved. Thus, by integrating (47) into (49) and reorganizing the terms, $\mathbf{P}^{\dagger}$ can be expressed as a linear function of $\mathbf{w}$ as

$$\mathbf{P}^{\dagger} = \mathbf{K}_1 + \mathbf{K}_2 \mathbf{w}, \tag{50}$$

where $\mathbf{w} = [q_0^6, q_1^6, \ldots, q_n^6]^T$ is an $(n+1) \times 1$ column vector, while $\mathbf{K}_1$ and $\mathbf{K}_2$ are expressed using the pseudo inverse of B-spline coefficient matrix $\mathbf{N}$ as

$$\mathbf{K}_1 = (\mathbf{N}^T\mathbf{N})^{-1}\mathbf{N}^T\mathbf{C}, \tag{51}$$

$$\mathbf{K}_2 = (\mathbf{N}^T\mathbf{N})^{-1}\mathbf{N}^T\mathbf{O}_{\mathrm{opt}}. \tag{52}$$

Then, according to (49)–(52), the objective function in (48) can be expanded in quadratic form in terms of $\mathbf{w}$ as

$$\min_{\mathbf{w}} \quad J_{\mathrm{M}} = \mathbf{w}^T\mathbf{D}\mathbf{w} + \mathbf{f}\mathbf{w}, \tag{53}$$

where $\mathbf{D}$ is a positive definite symmetric matrix and $\mathbf{f}$ is a column vector. The detailed mathematic derivation is reported in Appendix A.

The optimization problem in QP form has to incorporate the objective function (53) with the inequality kinematic constraints to limit the length of the standoff axis within the stroke $[q_{\min}^6, q_{\max}^6]$ as

$$\begin{cases} \min\limits_{\mathbf{w}} \quad J_{\mathrm{M}} = \mathbf{w}^T\mathbf{D}\mathbf{w} + \mathbf{f}\mathbf{w} \\ \text{s.t.} \quad q_{\min}^6 \le \mathbf{w}_j \le q_{\max}^6 \\ \qquad \text{for} \quad j = 0, 1, \ldots, n \end{cases}. \tag{54}$$

The geometric constraints will be detailed in the next section.

2.4. GFD Construction in 3D Space and Iterative Processing of RTOs

The C-space method [4,5,11] can be employed for collision-free trajectory planning, providing a robust framework for exploring all possible configurations and ensuring collision avoidance, as demonstrated in our previous work [28]. However, building graphs that define feasible and forbidden zones for each node of the overall trajectory is computationally intensive, particularly when combined with shortest-path optimization. Furthermore, our method incorporates a secondary collision-avoidance process to refine trajectory planning while optimizing the redundant standoff axis.

To create a unique GFD that represents the forbidden areas for the three wrist joints $\tilde{\mathbf{q}}_i = [q_i^4, q_i^5, q_i^6]^T$, we extend the rotational axes' C-space to a three-dimensional representation. Similar research [31] introduced cutter lift height δ to construct a 3D C-space using inclination and yaw planes. However, the extended GFD in our method incorporates an additional degree of freedom provided by q_6, alongside the rotational axes q_4 and q_5, to enhance flexibility. This design is derived from the constrained leading and tilt angles (ϕ, φ), as illustrated in Figure 4. The detailed process of constructing the GFD from the LCS to the WCS is described in our previous work [28].

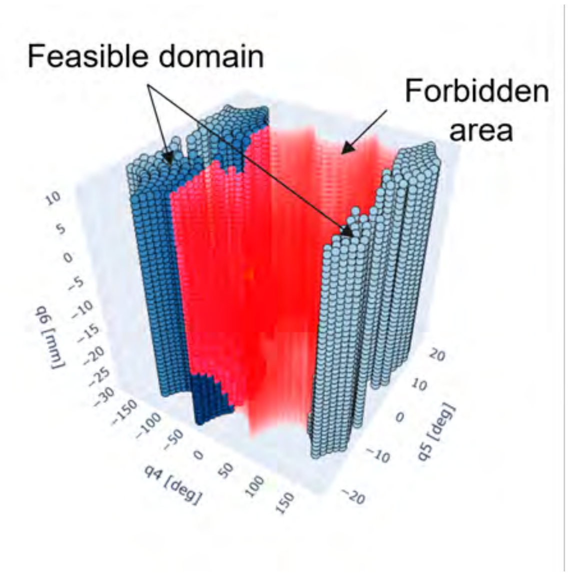


Figure 4. Proposed 3D GFD on a single CL point. The forbidden areas are indicated within the red regions, and the feasible candidates are represented by the shades of blue corresponding to the distance from the reference point.

It is worth mentioning that the IKT in (3) allows two configurations of the rotational axes for a single tool orientation. However, typical 3D laser head configurations, as shown in Figure 5, result in interference patterns distinct from those of CNC machine tool spindles. Consequently, the proposed GFD is constructed based on wrist joint coordinates, similar to the DAO representation in [7], rather than the tilt and yaw planes used in other studies [5,13].

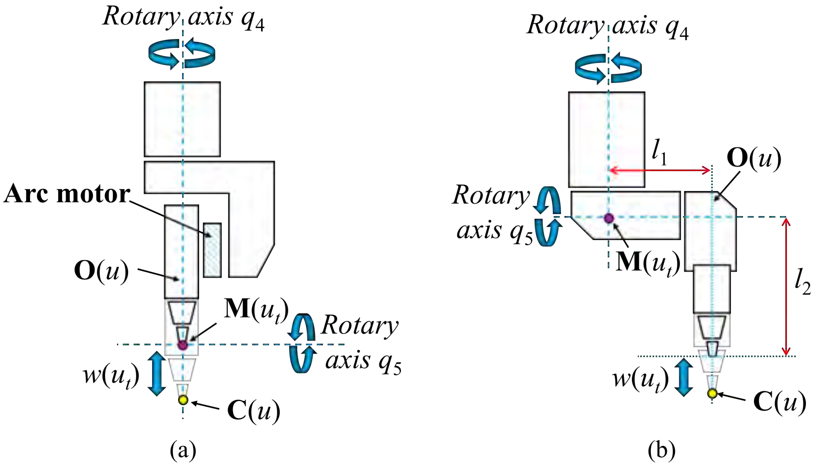


Figure 5. Mechanical structure of 3D laser cutting heads. (**a**) Zero-offset head on EFORT machine; (**b**) conventional head on traditional 3D laser cutting machines with offsets l_1 and l_2.

The concept of C-distance , introduced in [5], quantifies the distance between the collided tool orientation and the reference orientation using ϕ and φ angles. The shortest C-distance identifies the minimal change in tool orientation between consecutive local optimal candidates. In our method, the distance equation adopts this principle for consistency. However, since joint trajectories are directly influenced by kinematic constraints, the Euclidean distance is weighted based on the dynamic performance of each joint, as

$$\Delta d_{i,p} = \sqrt{k_4\left(q^4_{i,p} - \hat{q}^4_i\right)^2 + k_5\left(q^5_{i,p} - \hat{q}^5_i\right)^2 + k_6\left(q^6_{i,p} - \hat{q}^6_i\right)^2} \quad \text{for } i = 0, 1, \ldots, m. \tag{55}$$

Here, p represents the index of the non-collided candidate $q^*_{i,p}$ within the feasible region of the ith GFD. The value $\hat{q}^*_i$ serves as the reference joint configuration derived via IKT based on the geometric features of RTOs at the ith CL point among $(m+1)$ initialized or updated RTO collections. Additionally, k_4, k_5, and k_6 serve as the dynamic weights in this process.

The RTO-based processing strategy, initially introduced in [3], minimizes collisions and mitigates abrupt tool orientation changes by leveraging RTOs. This strategy involves three main steps to generate a collision-free trajectory:

1. Initialize RTOs: Define initial RTOs to act as via points, serving as the foundation for each joint trajectory.
2. Trajectory Optimization: Utilize an optimization algorithm to compute smooth, feasible trajectories, leveraging the current set of RTOs as constraints.
3. Update RTOs: Add new RTOs when necessary and iteratively refine the optimization process until the entire trajectory is verified to be collision-free.

This iterative method systematically refines the trajectory, eliminating collisions and preserving smooth tool orientations along the entire path. The initial selection of RTOs is crucial for successful trajectory optimization. It significantly impacts both the convergence speed of the optimization algorithm and the effectiveness of collision avoidance. A well-chosen set of RTOs enables the efficient exploration of feasible domains, facilitating the identification of smooth, collision-free trajectories. Conversely, an inadequate or poorly initialized selection of RTOs can lead to suboptimal trajectory solutions, increased computational time or even failure to find a collision-free path.

The selection of initial RTOs should prioritize candidates with minimal Euclidean distance $\Delta d_{i,p}$ to identify the closest safe ones between the candidate $q^*_{i,p}$ and the ith GFD. Additionally, the relative distance between two adjacent RTOs concerning the segment length should be considered to avoid abrupt changes in tool orientation. A greedy algorithm (i.e., Dijkastra algorithm) addresses this problem by optimizing a multi-objective function. This function, parameterized by the decision variable $\tilde{\mathbf{q}}$, represents all the defined RTOs $[\tilde{\mathbf{q}}_0, \tilde{\mathbf{q}}_1, \ldots, \tilde{\mathbf{q}}_m]$ and is expressed as

$$\min_{\tilde{\mathbf{q}}} \quad k_1 \sum_{i=0}^{m} \Delta d_{i,p} + k_2 \sum_{i=1}^{m} \frac{||\tilde{\mathbf{q}}_i - \tilde{\mathbf{q}}_{i-1}||_2}{\sigma_i}, \tag{56}$$

where $||\cdot||_2$ indicates the Euclidean distance between two adjacent RTOs, and σ_i represents the tool path length between the CL points $\mathbf{C}_i$ and $\mathbf{C}_{i-1}$. Here, k_1 and k_2 are the weights assigned to the two components of the objective function to be tuned.

Let $\tilde{u}_i$ be the arc length parameter corresponding to the resolved RTO $\tilde{\mathbf{q}}_i$ for $i \in [0, m]$. The spline form of a set of joint coordinates can be represented as follows

$$q^*(\tilde{u}_i) = \sum_{j=0}^{n} \Theta^*_j N_{j,k}(\tilde{u}_i) \quad \text{for} \quad i = 0, 1, \ldots, m. \tag{57}$$

According to (27), (36) and (43), the above set of via joint coordinates can be expressed as an equality constraint equation in the following form:

$$\mathbf{A}^*_{\mathrm{RTO}}\mathbf{\Theta}^* = \mathbf{b}^*_{\mathrm{RTO}}, \tag{58}$$

where $\mathbf{A}^*_{\mathrm{RTO}}$ is the $(m+1) \times (n+1)$ coefficient matrix and $\mathbf{b}^*_{\mathrm{RTO}}$ is the $(m+1) \times 1$ vector including $q^*(\tilde{u}_i)$, as follows

$$\mathbf{A}^*_{\mathrm{RTO}} = \begin{bmatrix} N_{0,k}(\tilde{u}_0) & N_{1,k}(\tilde{u}_0) & \cdots & N_{n,k}(\tilde{u}_0) \\ N_{0,k}(\tilde{u}_1) & N_{1,k}(\tilde{u}_1) & \cdots & N_{n,k}(\tilde{u}_1) \\ \vdots & \vdots & \ddots & \vdots \\ N_{0,k}(\tilde{u}_m) & N_{1,k}(\tilde{u}_m) & \cdots & N_{n,k}(\tilde{u}_m) \end{bmatrix}, \tag{59}$$

$$\mathbf{b}^*_{\mathrm{RTO}} = \left[q^*(\tilde{u}_0), q^*(\tilde{u}_1), \ldots q^*(\tilde{u}_m)\right]^T. \tag{60}$$

At least two RTOs are required, typically specified at the start and the end of the tool path. Once a collision occurs with the currently generated trajectory, the dynamic $m+1$ dimension of the equality constraint matrix needs to be updated by inserting new RTOs into the optimization equations during iterative processing.

The final formulation of the minimized jerk trajectory optimization for the rotational axes, incorporating the additional geometric equality constraint (58), can be expressed as an extension of (43):

$$\begin{cases} \min\limits_{\Theta^*} \quad J^*_{\mathrm{R}} = \mathbf{\Theta}^{*T}\mathbf{H}\mathbf{\Theta}^* \\ \text{s.t.} \quad \mathbf{A}^*_{\mathrm{ineq}}\mathbf{\Theta}^* \le \mathbf{b}^*_{\mathrm{ineq}} \\ \qquad \mathbf{A}^*_{\mathrm{kin}}\mathbf{\Theta}^* = \mathbf{b}^*_{\mathrm{kin}} \\ \qquad \mathbf{A}^*_{\mathrm{RTO}}\mathbf{\Theta}^* = \mathbf{b}^*_{\mathrm{RTO}} \end{cases}. \tag{61}$$

Similarly, the geometric equality constraint for the extension value of the redundant standoff axis q_6 must be incorporated into (54). The design variable $\mathbf{w}$ must be constrained within the feasible domain to ensure collision-free conditions during the adjustment of q_6. Consequently, the minimized jerk optimization problem for the M path can be refined as follows:

$$\begin{cases} \min\limits_{\mathbf{w}} \quad J_{\mathrm{M}} = \mathbf{w}^T\mathbf{D}\mathbf{w} + \mathbf{f}\mathbf{w} \\ \text{s.t.} \quad \mathbf{A}^{\mathbf{w}}_{\mathrm{RTO}}\mathbf{w} = \mathbf{b}^{\mathbf{w}}_{\mathrm{RTO}}, \\ \qquad q^6_{\min} \le \mathbf{w}_j \le q^6_{\max}, \\ \qquad \text{for} \quad j = 0, 1, \ldots, n, \end{cases}, \tag{62}$$

where $\mathbf{A}^{\mathbf{w}}_{\mathrm{RTO}}$ is an $(m+1) \times (n+1)$ coefficient matrix with a single value of '1' in the column corresponding to $\tilde{u}_i$ as a $1 \times (n+1)$ row vector $\mathbf{e}_i$, and zeros elsewhere. The vector $\mathbf{b}^{\mathbf{w}}_{\mathrm{RTO}}$ is $(m+1) \times 1$, containing the values of $q_6(\tilde{u}_i)$. These matrices are defined as follows:

$$\mathbf{A}^{\mathbf{w}}_{\mathrm{RTO}} = \begin{bmatrix} \mathbf{e}_i(\tilde{u}_0) \\ \mathbf{e}_i(\tilde{u}_1) \\ \vdots \\ \mathbf{e}_i(\tilde{u}_m) \end{bmatrix}, \quad \mathbf{b}^{\mathbf{w}}_{\mathrm{RTO}} = \begin{bmatrix} q_6(\tilde{u}_0) \\ q_6(\tilde{u}_1) \\ \vdots \\ q_6(\tilde{u}_m) \end{bmatrix}, \tag{63}$$

$$\mathbf{e}_{i,k} = \begin{cases} 1, & \text{if } k = \text{index corresponding to } \tilde{u}_i, \\ 0, & \text{otherwise.} \end{cases} \tag{64}$$

for $k = 0, 1, \ldots, n$.

Finally, the two optimization problems, incorporating both kinematic and geometric constraints, are summarized in (61) and (62). These formulations collectively aim to achieve a minimized jerk and collision-free trajectory, as outlined in the preceding sections.

2.5. Summary of the Methodology

The proposed MJTO algorithm for 3D laser cutting machines integrates smoothness optimization, collision avoidance, and redundancy exploitation. Figure 6 illustrates the structured, iterative process of the methodology:

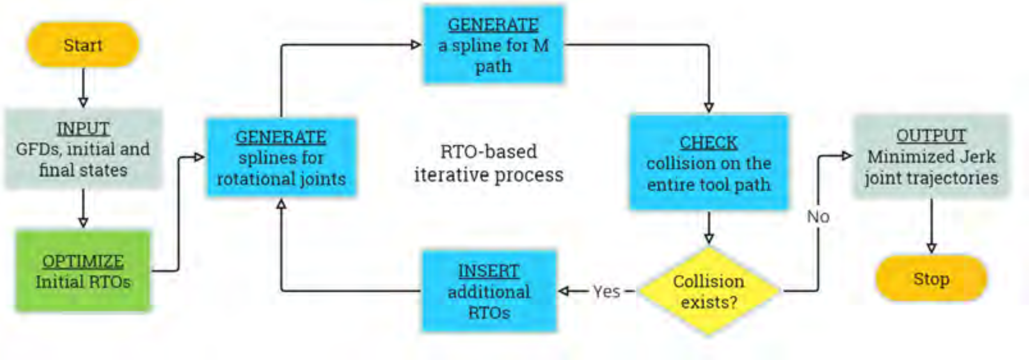


Figure 6. Flow chart of MJTO algorithm

Initial RTO Optimization: Initial RTOs are refined to minimize orientation changes and align with reference points according to (56), forming the basis for trajectory planning.

Rotational Joint Trajectory Generation: B-spline modeling (5) is used to generate minimized jerk for the displacement of rotational axis trajectories according to the QP optimization (61), ensuring smooth motion satisfying the kinematic constraints (15) and the geometric constraints (58) in the GFDs.

M Path Optimization: The redundant standoff axis trajectory, modeled with B-splines as (47), is optimized via a separate QP problem in (62) to achieve collision-free via points and smooth translational motion.

Collision Checking: The entire tool path is checked for collisions. Detected collisions are resolved by introducing additional RTOs as constraints with the shortest distance in (55), repeating the trajectory generation.

Iterative Refinement: The process cycles through rotational axes generation, M path optimization, and collision checking until achieving a smooth, collision-free trajectory.

As shown in Figure 6, this methodology ensures collision-free operation while balancing smoothness and precision. By integrating B-spline-based MJTO, iterative refinement, and redundancy exploitation, the framework excels in high-speed, high-precision machining.

3. Illustrative Simulations on B-Pillar Machining

The proposed Minimized Jerk Trajectory Optimization method was implemented in Python 3.11.8 using CVXPY for QP modeling and interfaced with the MOSEK 10.2.5 solver [32]. While the QP-based structure facilitates efficient optimization, the computational costs are affected by the complexity of 3D geometric representations for collision detection and the iterative nature of the RTO-based framework. At this stage, the proposed algorithm runs in an offline computation on a PC platform with a i5-10210U CPU (1.6 GHz, 8-core) and 32 GB of RAM. Two typical cases are presented to demonstrate the effectiveness and advancements of the method, illustrated with an automotive component, as shown in Figure 7. The first case study illustrates the methodology in a straightforward and long machining scenario with minimal tool orientation variation, whereas the second case compares the proposed algorithm with our previous work [28] in a challenging corner scenario involving significant tool orientation adjustments.

Figure 7. B-pillar workpiece and corresponding tool paths. The purple line shows the trajectory followed in Case 1, while the red line shows that of Case 2

The data of machining parameters and joint kinematics parameters used in the following two cases are provided in Tables 1 and 2.

Table 1. Machining parameters.

Machining Parameters		
Parameter	**Case Study 1**	**Case Study 2**
Feedrate f	20 m/min	15 m/min
Acceleration $\dot{f}_{\max}$	8 m/s^2	8 m/s^2
Jerk $\ddot{f}_{\max}$	80 m/s^3	80 m/s^3
Lead angle $\phi_{\max}$	5°	15°
Tilt angle $\varphi_{\max}$	5°	10°
Sampling Interval Δt	0.01 s	0.004 s

Table 2. Joint kinematic parameter of the objective laser machine.

Joint Kinematic Parameters					
Joint	$\boldsymbol{q^*_{\min}}$	$\boldsymbol{q^*_{\max}}$	$\boldsymbol{\omega^*_{\max}}$	$\boldsymbol{\alpha^*_{\max}}$	$\boldsymbol{J^*_{\max}}$
q_1	0 m	1.6 m	2 m/s	12 m/s ^{2}	120 m/s^3
q_2	0 m	1.4 m	2 m/s	12 m/s ^{2}	120 m/s^3
q_3	0 m	3.0 m	2 m/s	12 m/s ^{2}	120 m/s^3
q_4	-2π	2π	10 rad/s	80 rad/s^2	200 rad/s^3
q_5	$\pi/2$	$\pi/2$	10 rad/s	80 rad/s^2	200 rad/s^3
q_6	-0.1 m	0.02 m	2 m/s	40 m/s^2	400 m/s^3

3.1. Case Study 1: Long Segment Machining

This study case aims to illustrate the application of the proposed minimized jerk trajectory optimization method in a straightforward machining scenario. The trajectory details and corresponding 3D GFDs across six RTOs for a long segment with minor tool orientation variations are depicted in Figure 8.

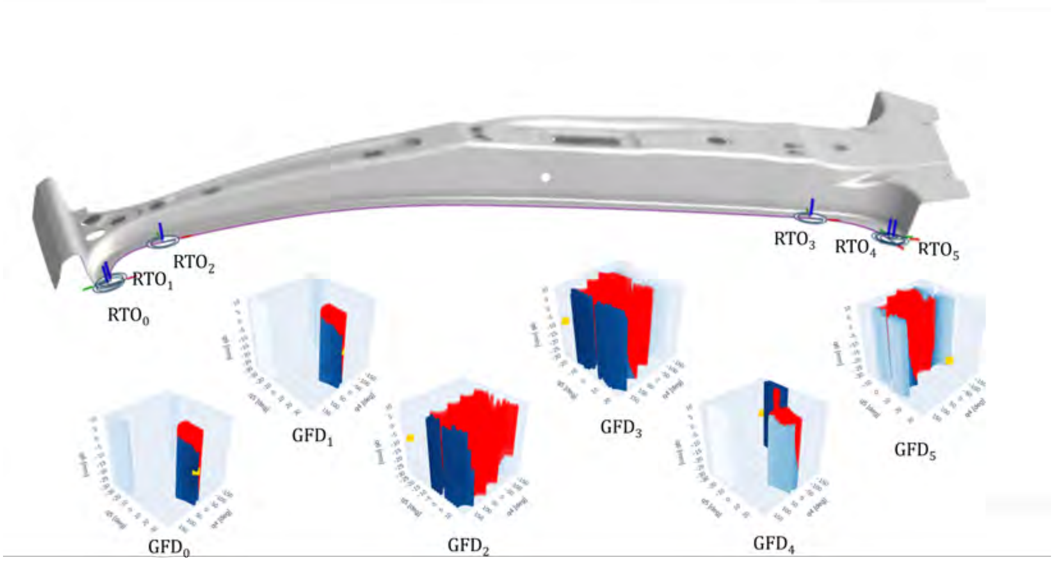


Figure 8. The long tool path in purple line and its corresponding RTOs, represented as six GFDs for Case 1 prior to optimization.

This scenario involves simulating the machining of a long, straight segment on the side of the B-pillar workpiece using the Python-based physical simulation environment pybullet, as shown in Figure 9. The initial path of the tool, generated without optimization, is shown in Figure 9a. However, Figure 9b illustrates an accidental collision between the cutting head and the workpiece caused by significant adjustments in joint angle q_4 following the conventional QI interpolation result, despite the minor angle variations between the initial RTOs.

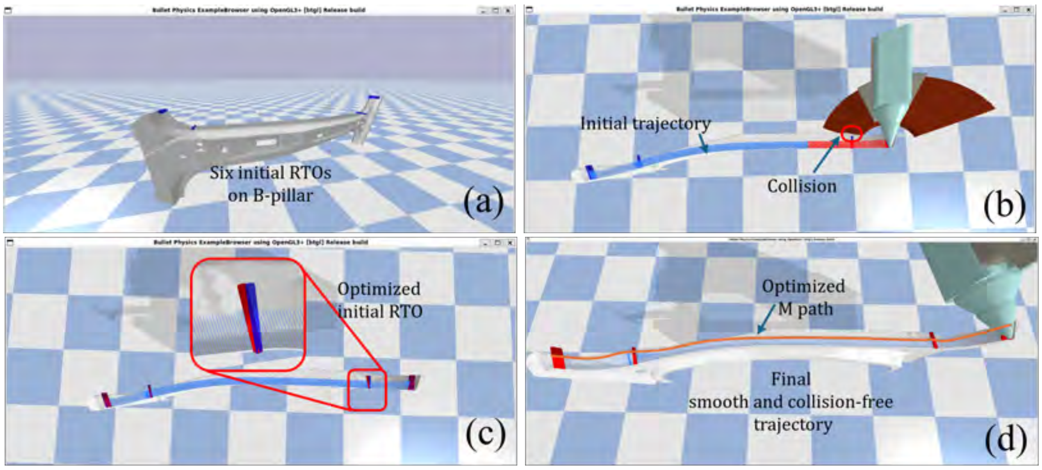


Figure 9. Pybullet simulation of the long trajectory using the MJTO algorithm for case study 1. (**a**) Initial position setup and six RTOs shown in blue; (**b**) collision observed with the original trajectory using QI interpolation, highlighted in red; (**c**) zoomed-in view of the initial RTO adjustments, optimized via Equation (56) transitioning from the blue RTO to the updated red RTO; (**d**) final trajectory with the optimized M path represented by the orange line.

The optimal trajectory is obtained within the first iteration of the optimization process, as shown in Figure 9d. Consequently, no additional RTOs are needed, as no further collisions are detected due to the effective initial RTO optimization depicted in Figure 9c. After optimizing the initial RTOs to ensure a collision-free trajectory, the angular changes between consecutive RTOs were carefully controlled. By keeping these angle adjustments within the tolerance for the allowable processing error, excessive movements in the joint q_4 were avoided, even with minimal changes in tool orientation.

The joint trajectory plots in Figure 10 highlight the significantly smoother kinematic performance of the proposed MJTO algorithm compared to the conventional QI method. This smoothness is evident across velocity, acceleration, and jerk levels, primarily due to the global optimization applied across all RTOs. Unlike the traditional QI method, which often causes discontinuities in acceleration and jerk between trajectory segments, the MJTO algorithm ensures holistic trajectory optimization, minimizing abrupt motion changes. This improvement not only enhances machining precision, but also reduces mechanical stress and wear on cutting components, thereby improving the the longevity and reliability of the manufacturing system.

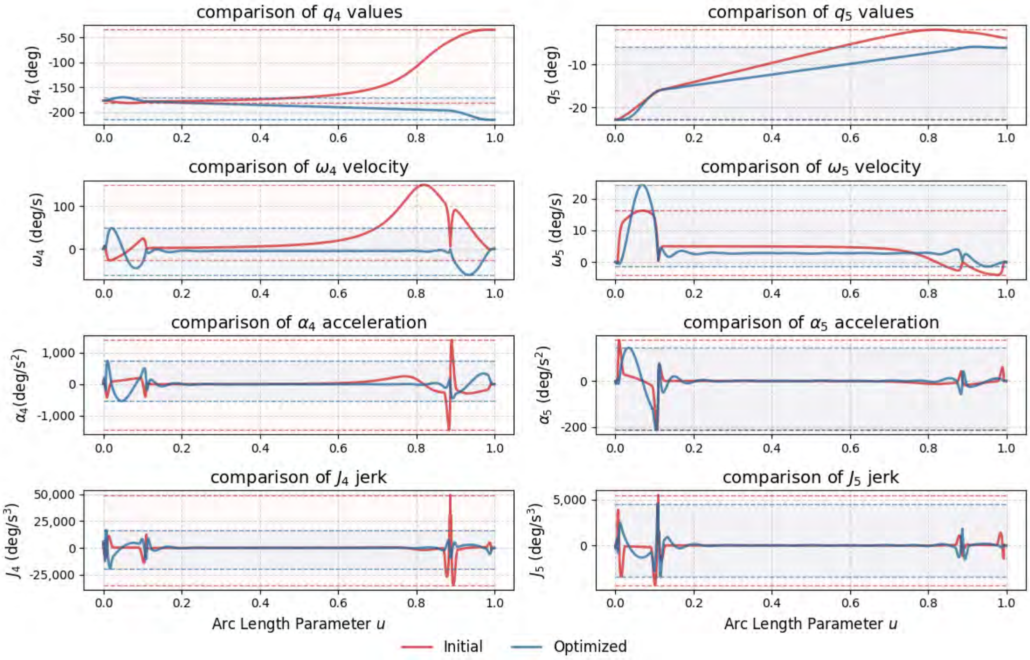


Figure 10. Comparison of initial (red) and optimized (blue) rotational axes trajectories for position q^*, velocity ω^*, acceleration α^*, and jerk J^* levels, along with their boundaries, represented by the dotted lines.

Furthermore, with the assistance of the high dynamic performance of the redundant standoff axis, the curvature of the M trajectory was optimized, resulting in improved performance on the translational axes. Although the original tool path in this case is relatively simple and long, posing minimal challenges to most translational axes, the integration of the redundant axis allows for finer control and smoother motion even in straightforward scenarios. This ensures translational movements are not only smoother, but also more efficient, contributing to reduced mechanical stress and improved overall system performance. The benefits of the redundant standoff axis become more evident in complex tasks, such as sharp corner navigation, as it will be demonstrated in the following case study, which validates the effectiveness of the MJTO algorithm in enhancing trajectory smoothness and reducing mechanical stress.

3.2. Case Study 2: Shape Corner Segment Machining

Sheet metal components with large angles of bend are exceedingly common in the automotive manufacturing industry. These pronounced bends pose significant challenges for 3D LCM processes. To meet high-speed machining requirements, the tool orientation must execute rapid and substantial rotations within short segments to accurately cut the lateral surfaces of bent sheet metal. In this case, the objective trajectory follows an irregular and complex corner surface with ten RTOs at key waypoints, as illustrated in Figure 11. Notably, there are two significant orientation changes of approximately 77° between adjacent tool orientations over short distances, posing a major challenge for smooth trajectory planning. This scenario, also considered in our previous work [28], demonstrates the effectiveness of the proposed optimization method in handling large-angle rotations smoothly and maintaining collision-free machining under demanding conditions. Furthermore, the optimized M trajectory, based on the proposed redundant standoff axis, ensures smoother translational motions during high-curvature machining, further enhancing overall performance and reliability.

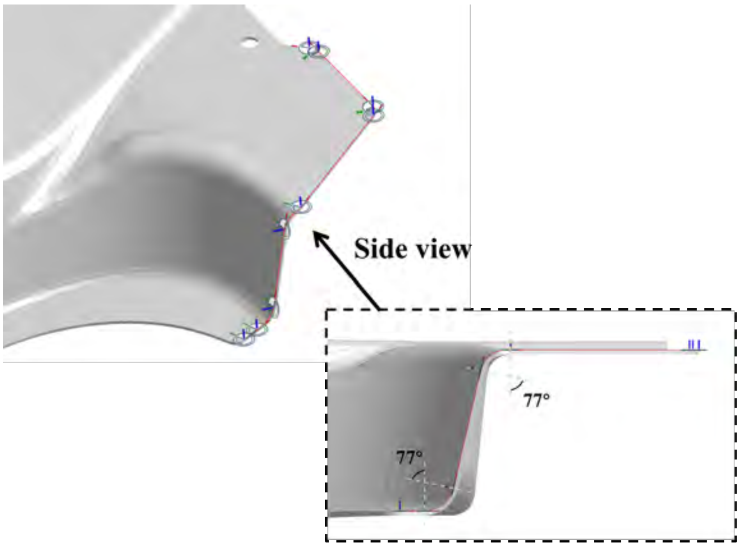


Figure 11. The objective trajectory on the target surface with a side view; note the 77° angular deviation between two adjacent tool orientations.

The experimental test data can be summarized as follows. The initial tool-path trajectory is interpolated by a standard G-code interpolator using QI for the orientation, leading to a total machining time of 2.51 s with the dynamic machining parameters. Based on the 4 ms sampling interval (Δt = 4 ms) of the motion controller, 628 CL points are generated. The proposed MJTO framework demonstrates significant computational advantages compared to our previous graph-based method: While conventional collision checking required exhaustive evaluation at all 628 CL points, our current implementation significantly reduces the number of collision detection operations from 628 to 12 critical RTO positions. Calculating the large-scale Hessian matrix in (18) takes 9.5 s, while the QP solver in CVXPY requires an additional 0.8s of computation time, yielding a total optimization time faster than the time of our prior approach. This acceleration stems from both reduced collision detection iterations and the unified 3D GFD method introduced in Section 2.4, which eliminates three axes collision checks together.

Figure 12 illustrates the evolution of RTOs throughout the optimization stages. In Figure 12a, the initial RTOs are first reselected to minimize abrupt angle changes in the short segments. This redistributes angular adjustments across multiple RTOs (Figure 12b). In subsequent iterations (Figure 12c), additional RTOs are inserted when intermediate collisions or interference issues are detected. These extra RTOs serve as additional constraints in the optimization. Ultimately, the minimized-jerk planning produces a smoother, collision-free trajectory, as shown in Figure 12d.

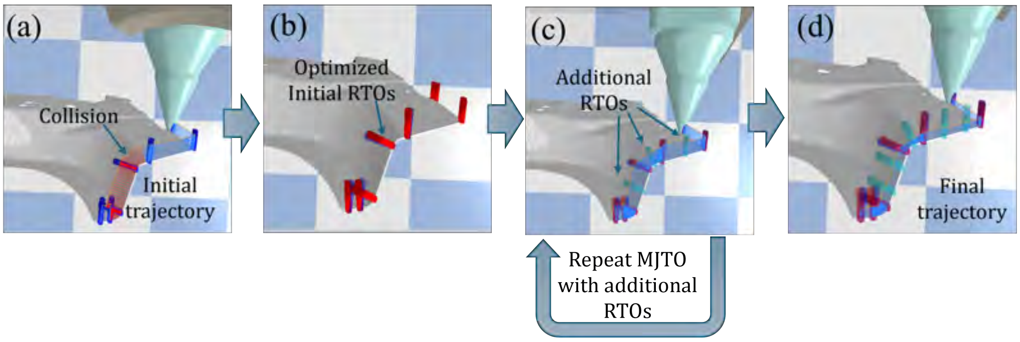


Figure 12. Evolution of RTOs through the optimization process: (**a**) initial RTOs; (**b**) RTOs after the first optimization; (**c**) additional RTOs added; (**d**) final collision-free and smoother trajectory.

To demonstrate the improvement in smoothness, Figure 13 compares the rotational axes q_4 and q_5 among three methods: the initial trajectory generated by quaternion interpolation (blue), a graph-based approach (green), and the proposed MJTO algorithm (orange). In the position plots, MJTO noticeably alters the original q_4 profile, redistributing the large angular changes more evenly. This is enabled by adjusting the RTOs in the early phases. In terms of acceleration, MJTO achieves much smaller and fewer spikes than the other two methods, indicating reduced dynamic stress and smoother motion when facing large orientation changes. However, one should note that the B-spline curves used here may produce overshooting or oscillations between non-keyway points in the trajectory, as they minimize the integral of the jerk over the entire path. This phenomenon can introduce angular errors or posture drifts, especially in regions of high curvature.

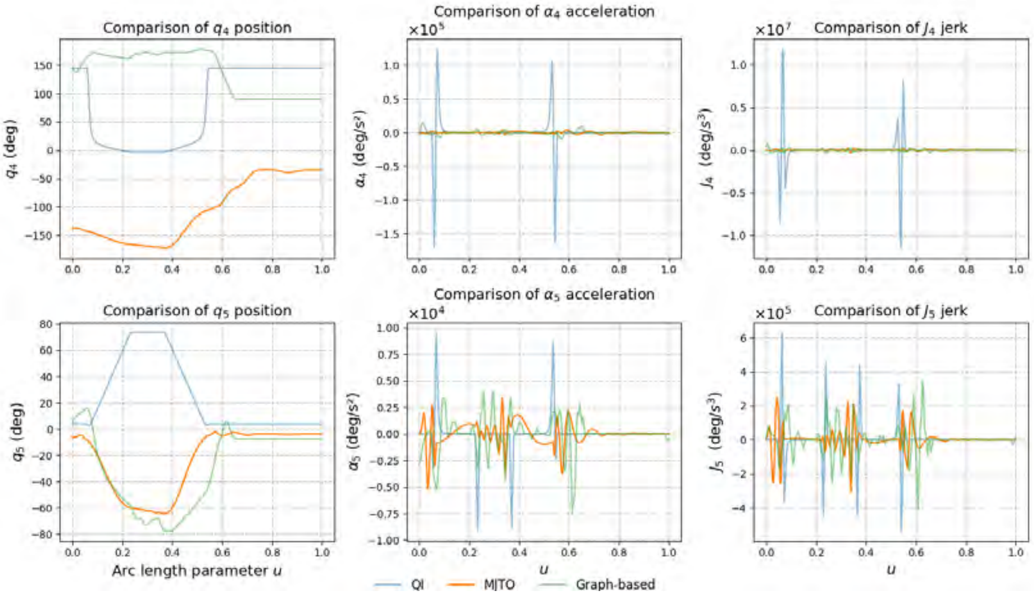


Figure 13. Comparison of rotational axes q_4 and q_5 in position (**left**), acceleration (**center**), and jerk (**right**) among QI (blue), graph-based (green), and MJTO (orange) methods.

Beyond the rotational axes, we also optimized the redundant translational axis M. Figure 14 compares the M path trajectory generated by QI (yellow) and MJTO (red) from the CL tool path (blue). By eliminating unnatural "knot" bends, the MJTO-based approach achieves a much smoother, more continuous path, which is particularly advantageous for high-speed machining tasks, where excessive velocity or acceleration discontinuities can lead to vibrations or inaccuracy.

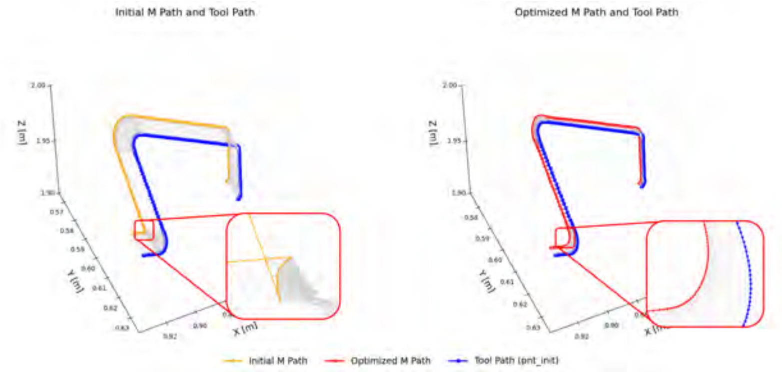


Figure 14. Comparison of M paths before (QI in blue) and after (MJTO in orange) optimization. A zoomed view highlights how MJTO smooths out any abrupt "knot" in the original path.

A more detailed view of smoothness enhancement can be observed in Figure 15, where q_1, q_2, q_3 are compared in terms of position, acceleration, and jerk. The MJTO trajectories (orange) show significantly reduced acceleration peaks and smoother jerk profiles compared to QI (blue) and graph-based methods (green). To better visualize the distribution of jerk values for each joint, Figure 16 shows a violin plot, where the horizontal axis represents different planning methods QI, graph-based, and MJTO, whereas the vertical axis shows the corresponding range and probability distribution of jerk for each joint. The "width" of the violin at each vertical level indicates how densely the jerk values are concentrated around that magnitude. Therefore, narrower violins (especially at higher jerk magnitudes) indicate fewer extreme jerk values; from these plots, one can see that MJTO's distributions tend to be narrower and more centered around lower jerk levels, implying a significant improvement in smoothness.

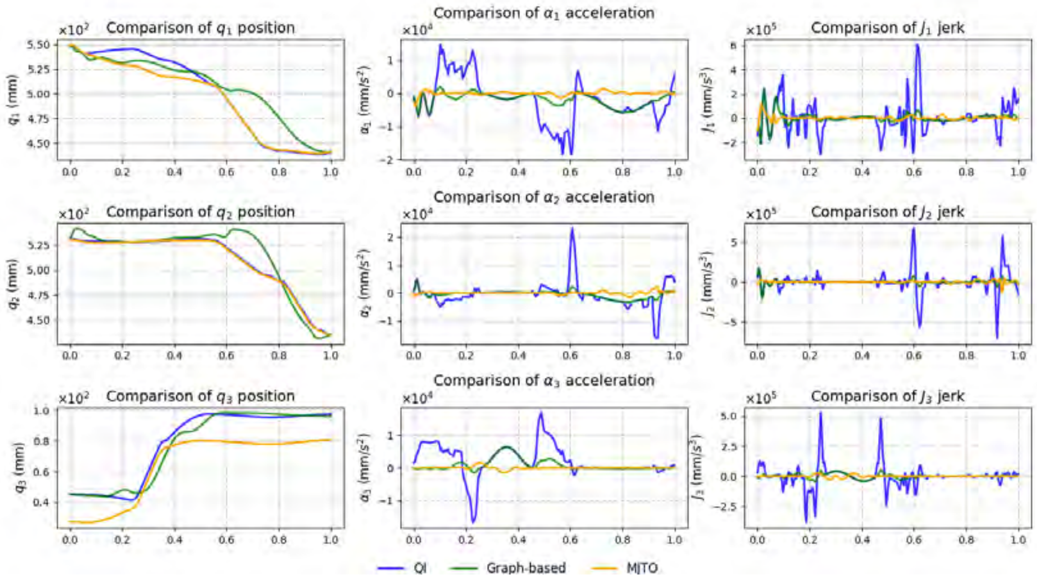


Figure 15. Translational joints q_1, q_2, q_3 under QI, graph-based, and MJTO: position (**left**), acceleration (**center**), and jerk (**right**).

In summary, by introducing refined RTO placement and employing minimized-jerk B-spline planning, the proposed MJTO method effectively handles large-angle rotations (up to 77°) while also enhancing the smoothness of both the rotational and translational axes. Compared to the QI and graph-based methods, MJTO demonstrates smaller acceleration peaks, reduced jerk, and fewer sudden orientation shifts. Although local overshoots may arise from the spline formulation, these can be controlled through additional keyway constraints until machining precision and posture accuracy are satisfied. Overall, the MJTO approach provides a stable, high-speed solution for sheet-metal corner-segment machining, ensuring accuracy and collision-free performance even under significant orientation variations.

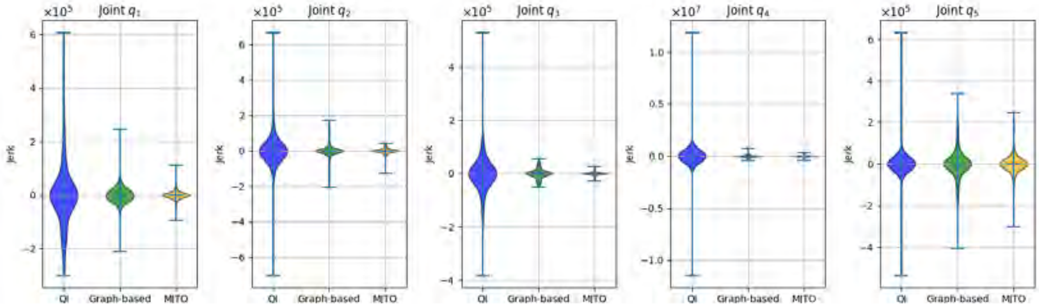


Figure 16. Violin plots of jerk distributions across five main axes (q_1 to q_5) for QI, graph-based, and MJTO methods. The horizontal axis indicates different methods; the vertical axis is the jerk value. Narrower shapes at high jerk ranges signify fewer extreme jerk points, reflecting smoother overall motions.

4. Conclusions and Future Work

In this study, we presented a novel trajectory optimization strategy specifically for 3D laser cutting machines equipped with a redundant standoff axis. Building on our previous work, the proposed approach employs a Minimized Jerk Trajectory Optimization framework with B-spline modeling to formulate two QP tasks: one focus on smoothing the rotational axes trajectories and the other dedicated to refining the M path governed by the redundant standoff axis. By integrating an iterative RTO-based optimization process within the feasible domains defined by the 3D joint-coordinate GFD, our method not only reduces abrupt angular transitions among multiple RTOs, but also significantly improves the continuity of the translational axis movements. Consequently, the optimized trajectories demonstrated enhanced smoothness, reduced mechanical stress, and consistent collision-free operation.

The effectiveness of our proposed method was demonstrated through two simulation cases involving the machining of an automotive B-pillar workpiece. In the first case, characterized by minimal orientation changes, a single iteration of our algorithm successfully produced a smooth, collision-free trajectory. Kinematic performance evaluations revealed reductions in acceleration and jerk peaks compared to the conventional QI method, underscoring the improved trajectory stability. In the second more demanding case featuring large-angle rotations and high-curvature corners, our iterative strategy effectively managed abrupt orientation transitions by strategically inserting additional RTOs and leveraging optimized adjustment to the M path. These results highlight the method's capability of accommodating complex geometries while maintaining trajectory smoothness and ensuring precise, collision-free movements.

Future work will explore the expension of the optimization framework to more intricate machining scenarios and broader industrial applications. While a constant feedrate assumption was adopted in this study, future research could incorporate adaptive feedrate control strategies to further enhance the trajectory smoothness. Additionally, integrating real-time collision detection and avoidance mechanisms will be investigated to accommodate unexpected disturbances. Meanwhile, two specific open issues remain to be addressed. First, since the free-form nature of B-splines can occasionally introduce local overshoot or oscillations, employing monotonic spline interpolation strategies [33] may help to mitigate these effects by enforcing strictly monotonic interpolation patterns. Second, further algorithmic refinements of MJTO, along with more computationally efficient implementations for the RTO insertion process, could reduce the computational overhead and improve scalability. Lastly, future efforts will include experimental validations conducted on physical laser cutting machines to verify simulation results and assess practical machining performance.

Author Contributions: Writing—original draft preparation, Z.D.; writing—review and editing, M.I. and A.R.; supervision, M.I. and A.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article.

Acknowledgments: The authors gratefully acknowledge EFORT Europe s.r.l. for sponsoring the Industrial Ph.D. project. Author Z.D. would like to extend his sincere appreciation to his colleagues Pietro Soccio and Daniele Costamagna for their invaluable inspiration and insightful contributions during the technical discussions.

Conflicts of Interest: Author Zhipeng Ding is employed by the company EFORT Europe s.r.l. The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

LCM	laser cutting manufacturing
CNC	computer numerical control
CL	cutter location
WCS	workpiece coordinate system
MCS	machine coordinate system
LCS	local coordinate system
IKT	inverse kinematic transformation
C-space	Configuration space
DAO	domain of admissible orientation
GFD	geometric feasible domain
RTO	representative tool orientation
QP	quadratic programming
QI	quaternion interpolation
MJTO	minimized jerk trajectory optimization

Appendix A. Derivation of the Quadratic Objective Function

In this appendix, we provide a concise derivation of the quadratic objective function used to minimize the integrated jerk along the M path.

Starting from the relationship between the control points $\mathbf{P}^{\dagger}$ and the redundant standoff axis $\mathbf{w}$ defined in (50), the objective function defined in (48) can be rewritten as

$$J_{\mathrm{M}} = (\mathbf{K}_1 + \mathbf{K}_2\mathbf{w})^T\mathbf{H}(\mathbf{K}_1 + \mathbf{K}_2\mathbf{w}). \tag{A1}$$

Expanding the quadratic form as

$$J_{\mathrm{M}} = \mathbf{w}^T\mathbf{K}_2^T\mathbf{H}\mathbf{K}_2\mathbf{w} + 2\mathbf{K}_1^T\mathbf{H}\mathbf{K}_2\mathbf{w} + \mathbf{K}_1^T\mathbf{H}\mathbf{K}_1, \tag{A2}$$

the objective function can be finally expressed in the standard quadratic form

$$J_{\mathrm{M}} = \frac{1}{2}\mathbf{w}^T\mathbf{D}\mathbf{w} + \mathbf{f}\mathbf{w} + c, \tag{A3}$$

where the components are defined as

$$\mathbf{D} = 2\mathbf{K}_2^T\mathbf{H}\mathbf{K}_2, \tag{A4}$$

$$\mathbf{f} = 2\mathbf{K}_2^T\mathbf{H}\mathbf{K}_1, \tag{A5}$$

$$c = \mathbf{K}_1^T\mathbf{H}\mathbf{K}_1. \tag{A6}$$

As the constant part c of the quadratic objective function can be omitted, this formulation transforms the optimization problem into a standard QP form in (53), facilitating an efficient solution using QP solvers.

References

1. Alsaadawy, M.; Dewidar, M.; Said, A.; Maher, I.; Shehabeldeen, T.A. A comprehensive review of studying the influence of laser cutting parameters on surface and kerf quality of metals. *Int. J. Adv. Manuf. Technol.* **2024**, *130*, 1039–1074. [CrossRef]
2. Beudaert, X.; Pechard, P.Y.; Tournier, C. 5-Axis tool path smoothing based on drive constraints. *Int. J. Mach. Tools Manuf.* **2011**, *51*, 958–965. [CrossRef]
3. Ho, M.C.; Hwang, Y.R.; Hu, C.H. Five-axis tool orientation smoothing using quaternion interpolation algorithm. *Int. J. Mach. Tools Manuf.* **2003**, *43*, 1259–1267. [CrossRef]
4. Lauwers, B.; Dejonghe, P.; Kruth, J. Optimal and collision free tool posture in five-axis machining through the tight integration of tool path generation and machine simulation. *Comput.-Aided Des.* **2003**, *35*, 421–432. [CrossRef]
5. Jun, C.S.; Cha, K.; Lee, Y.S. Optimizing tool orientations for 5-axis machining by configuration-space search method. *Comput.-Aided Des.* **2003**, *35*, 549–566. [CrossRef]
6. Tutunea-Fatan, O.R.; Feng, H.Y. Configuration analysis of five-axis machine tools using a generic kinematic model. *Int. J. Mach. Tools Manuf.* **2004**, *44*, 1235–1243. [CrossRef]
7. Castagnetti, C.; Duc, E.; Ray, P. The domain of admissible orientation concept: A new method for five-axis tool path optimisation. *Comput.-Aided Des.* **2008**, *40*, 938–950. [CrossRef]
8. Wang, N.; Tang, K. Five-axis tool path generation for a flat-end tool based on iso-conic partitioning. *Comput.-Aided Des.* **2008**, *40*, 1067–1079. [CrossRef]
9. Wang, Y.; Xu, J.; Sun, Y. Tool orientation adjustment for improving the kinematics performance of 5-axis ball-end machining via CPM method. *Robot. Comput.-Integr. Manuf.* **2021**, *68*, 102070. [CrossRef]
10. Plakhotnik, D.; Lauwers, B. Graph-based optimization of five-axis machine tool movements by varying tool orientation. *Int. J. Adv. Manuf. Technol.* **2014**, *74*, 307–318. [CrossRef]
11. Mi, Z.; Yuan, C.M.; Ma, X.; Shen, L.Y. Tool orientation optimization for 5-axis machining with C-space method. *Int. J. Adv. Manuf. Technol.* **2017**, *88*, 1243–1255. [CrossRef]
12. Xu, H.; Hu, J.; Wu, W. Optimization of 3D laser cutting head orientation based on the minimum energy consumption. *Int. J. Adv. Manuf. Technol.* **2014**, *74*, 1283–1291. [CrossRef]
13. Xu, J.; Zhang, D.; Sun, Y. Kinematics performance oriented smoothing method to plan tool orientations for 5-axis ball-end CNC machining. *Int. J. Mech. Sci.* **2019**, *157*, 293–303. [CrossRef]
14. Xiao, Q.B.; Wan, M.; Zhang, W.H.; Yang, Y. Tool orientation optimization for the five-axis CNC machining to constrain the contour errors without interference. *J. Manuf. Process.* **2022**, *76*, 46–56. [CrossRef]
15. Wu, L.; Xu, J.; Yin, X.; Sun, Y. Jerk-optimal piecewise planning of tool orientation for 5-axis ball-end machining with linearized kinematic constraints. *J. Manuf. Sci. Eng.* **2023**, *145*, 071003. [CrossRef]
16. Huang, K.T.; Zhang, Z.; Gong, H.; Li, Z.; Fang, F.; Wang, D. Constructing smooth tool orientation field based on radial basis function for 5-axis machining. *Int. J. Adv. Manuf. Technol.* **2017**, *91*, 1369–1379. [CrossRef]
17. Langeron, J.M.; Duc, E.; Lartigue, C.; Bourdet, P. A new format for 5-axis tool path computation, using Bspline curves. *Comput.-Aided Des.* **2004**, *36*, 1219–1229. [CrossRef]
18. Yuen, A.; Zhang, K.; Altintas, Y. Smooth trajectory generation for five-axis machine tools. *Int. J. Mach. Tools Manuf.* **2013**, *71*, 11–19. [CrossRef]
19. Li, D.; Zhang, W.; Zhou, W.; Shang, T.; Fleischer, J. Dual NURBS path smoothing for 5-axis linear path of flank milling. *Int. J. Precis. Eng. Manuf.* **2018**, *19*, 1811–1820. [CrossRef]
20. Sharon, A.; Hogan, N.; Hardt, D.E. The macro/micro manipulator: An improved architecture for robot control. *Robot. Comput.-Integr. Manuf.* **1993**, *10*, 209–222. [CrossRef]
21. Quan, B.T.; Huang, J.; Harada, M.; Yabuta, T. Control of a macro-micro robot system using manipulability of the micro robot. *JSME Int. J. Ser. C Mech. Syst. Mach. Elem. Manuf.* **2006**, *49*, 897–904. [CrossRef]

22. Zhou, Y.; Chen, C.Y.; Yang, G.; Li, Y. A sampling-based motion assignment strategy with multi-performance optimization for macro-micro robotic system. *IEEE Robot. Autom. Lett.* **2022**, *7*, 11649–11656. [CrossRef]
23. Kakinuma, Y.; Ogawa, S.; Koto, K. Robot polishing control with an active end effector based on macro-micro mechanism and the extended Preston's law. *Cirp Ann.* **2022**, *71*, 341–344. [CrossRef]
24. Cursi, F.; Bai, W.; Yeatman, E.M.; Kormushev, P. Optimization of surgical robotic instrument mounting in a macro–micro manipulator setup for improving task execution. *IEEE Trans. Robot.* **2022**, *38*, 2858–2874. [CrossRef]
25. Uzunoglu, E.; Dede, M.I.C.; Kiper, G. Trajectory planning for a planar macro-micro manipulator of a laser-cutting machine. *Ind. Robot. Int. J.* **2016**, *43*, 513–523. [CrossRef]
26. Liu, J.; Xie, X.; Zhou, Z.; Gao, W.; Liang, Z. Research on high-speed and high-precision laser processing algorithm based on macro-micro platform. *Proc. Inst. Mech. Eng. Part B J. Eng. Manuf.* **2023**, *237*, 899–910. [CrossRef]
27. Ding, Z.; Soccio, P.; Indri, M.; Rizzo, A. Through hole-cutting conic posture optimization for a redundant 3D laser cutting machine. *Int. J. Adv. Manuf. Technol.* **2024**, *132*, 443–461. [CrossRef]
28. Ding, Z.; Indri, M.; Rizzo, A.; Soccio, P. Smooth and Collision-Free Trajectory Planning for Redundant 3D Laser Cutting Machines. In Proceedings of the 2024 IEEE 29th International Conference on Emerging Technologies and Factory Automation (ETFA), Padova, Italy, 10–13 September 2024.
29. Gordon, W.J.; Riesenfeld, R.F. B-spline curves and surfaces. In *Computer Aided Geometric Design*; Elsevier: Amsterdam, The Netherlands , 1974; pp. 95–126.
30. Hu, P.; Tang, K. Improving the dynamics of five-axis machining through optimization of workpiece setup and tool orientations. *Comput.-Aided Des.* **2011**, *43*, 1693–1706. [CrossRef]
31. Lu, J.; Cheatham, R.; Jensen, C.; Chen, Y.; Bowman, B. A three-dimensional configuration-space method for 5-axis tessellated surface machining. *Int. J. Comput. Integr. Manuf.* **2008**, *21*, 550–568. [CrossRef]
32. Mosek ApS. *The MOSEK Optimization Toolbox for MATLAB Manual*, version 10.1; Mosek ApS: Copenhagen, Denmark , 2024.
33. Fritsch, F.N.; Butland, J. A method for constructing local monotone piecewise cubic interpolants. *SIAM J. Sci. Stat. Comput.* **1984**, *5*, 300–304. [CrossRef]