Smooth and collision-free trajectory planning for redundant 3D laser cutting machines

Smooth and collision-free trajectory planning for redundant 3D laser cutting machines

Zhipeng Ding*†, Marina Indri*, Alessandro Rizzo*, and Pietro Soccio†
*Dipartimento di Elettronica e Telecomunicazioni
Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy
{zhipeng.ding, marina.indri, alessandro.rizzo}@polito.it
†Efort Europe s.r.l, Corso Duca degli Abruzzi 2, 10129 Torino, Italy
pietro.soccio@efort-europe.com

***Abstract*—Smooth and collision-free trajectory planning is crucial to high speed and high precision machining, such as 3D laser cutting. However, it is difficult to further enhance the kinematic performance of the primary translational axes during the process. This paper presents a novel two-phase planning strategy, which optimizes the tool orientation and leverages a redundant standoff axis to significantly enhance the smoothness of the translational movements in redundant 3D laser cutting machines. In the first phase, collision-free configuration spaces (C-spaces) are constructed along the tool path, utilizing a graph-based search approach with Dijkstra's algorithm for tool orientation optimization. Subsequently, a secondary orientation curve, namely the M path, is planned in the second phase with a variable distance from the primary tool path curve, and the motion of the redundant standoff axis is handled via a deep reinforcement learning approach. The proposed methodology provides an advancement in conventional five-axis machines lacking of flexibility. Experimental validation confirms the potential of the approach to substantially improve machining accuracy and efficiency.**

***Index Terms*—3D laser cutting, Redundancy, Collision avoidance, Tool orientation optimization, Reinforcement learning**

I. INTRODUCTION

The smoothness of trajectories is crucial in high speed and high precision machining applications, such as 3D laser cutting. However, planning a smooth and collision-free path presents significant challenges when conventional five-axis Numerical Control (NC) machines are used. Many efforts have been made to optimize tool orientation, addressing challenges related to collision avoidance and enhancing the smoothness of rotary axis movements [1]. However, excessive acceleration and inertia on translational axes during high speed machining also stand as primary factors leading to machine vibrations and lower machining accuracy [2].

This research introduces a novel approach that, while maintaining optimized tool orientations at each Cutter Location (CL) point, leverages a redundant standoff axis to enhance the motion performance of non-rotational axes through strategic trajectory planning. Such a methodology is adopted on a 3D laser cutting machine equipped with a redundant axis as the research object in this paper. This redundant 3D laser cutting machine is currently a prototype developed by EFORT Intelligent Equipment Co., Ltd. The schematic of the machine and axes directions are presented in Fig. 1.

The collision avoidance problem in five-axis machining has been addressed through tool orientation adjustment; the approaches available in literature can be distinguished in two classes: (*i*) discrete tool orientations techniques, and (*ii*) iterative processing based on Representative Tool Orientation (RTO). The discrete methods focus on segmenting the path and pinpointing collision-free trajectories at each CL point within feasible domains, as characterized by inclination and tilt angles in the Local Coordinate System (LCS). Early efforts by Jun et al. [3] took the solution of smoothing tool orientation through search methods, whereas subsequent studies, like Castagnetti et al. [4], found the construction of feasible regions in the Machine Coordinate System (MCS) more efficient due to the Inverse Kinematic Transformation (IKT)'s nonlinearity. This led to the application of sequential quadratic programming methods, framing the Domain of Admissible Orientation (DAO) within approximated linear inequality constraints directly with rotary axes. Plakhotnik and Lauwers [5] introduced a graph-based algorithm aimed at minimizing rotary axes movement, incorporating admissible arcs. Mi et al. [6] further refined this approach by including acceleration considerations for rotary axes in the orientation optimization cost function by a difference graph. The second class of methods employs RTOs for iterative tool orientation adjustments to avoid collisions, e.g., utilizing Quaternion Interpolation (QI) as in [7], to ensure smooth transitions between collision-free RTOs. High-order B-spline representations of orientation curves, followed by interpolation to ascertain the corresponding orientations, were utilized in subsequent refinements, by Xu et al. [8], enhancing kinematic performance through smoothing interpolation techniques within the RTO iterative process. Wu et al. [9] proposed a jerk-optimal tool orientation piecewise planning method to further improve the performance.

The research community is increasingly concentrating on refining the movement of rotary axes to achieve smoother trajectories: many advanced techniques dedicated to the optimization of the tool orientation were proposed recently. The tool orientation optimization problem was solved by a genetic algorithm in [10]. A computational efficiency was demonstrated by converting the tool orientation planning problem into a reinforcement learning task [11]. It is noteworthy that the dual-curves technique [8], [12], [13], employed for optimizing tool orientation, has garnered increasing attention among scholars. An equidistant double NURBS based on dual quaternion trajectory generation algorithm was proposed to reaching G^2 continuity by Zhang et al [12]. The feedrate fluctuations of rotary axes were reduced in [13], enhancing the milling process efficiency by optimizing dual NURBS curves. Regarding the dual NURBS technique, to the best of our knowledge, there are still two main unsolved issues: first, the fixed height [12] between curves can result in exaggerated tool orientation changes for paths with acute angles, such as corner etc, potentially causing the orientation curve to invert and "knot". Second, the influence of the secondary orientation curve not only improves the smoothness of rotary axes motion, but it also significantly affects the motion of the machine's translational axes, even if such an aspect has often been underestimated [8], [13], [14].

Numerous research efforts [2], [15]–[17] focus on interpolation scheduling, aiming at refining and smoothing the trajectory. However, these approaches are limited by five-axis machine's inflexibility in the mechanical structure.

In the approach proposed in this paper, this issue is addressed in two phases: first, optimizing the tool orientation, and subsequently, enhancing the performance of translational axes through the exploitation of redundancy. The approach starts with the construction of C-spaces at each CL point, incorporating specific machining constraints. The tool orientation optimization challenge is tackled by employing a graph-based search across all C-spaces, utilizing Dijkstra's algorithm for efficient traversal. Furthermore, a novel method is introduced for the planning of a secondary orientation curve, termed the M path, which maintains a variable distance from the primary tool path. This is accomplished by handling the motion of the redundant standoff axis via a deep reinforcement learning approach, which allows to significantly improve the smoothness of translational axes movements. The proposed strategy demonstrates notable gains in both efficiency and precision during simulations, offering a breakthrough solution to the lack of flexibility in five-axis machine tools, by utilizing a redundant axis strategy.

The rest of the article is organized as follows: Section II details the graph-based methodology for optimizing tool orientation across permissible regions at each CL point. Section III delves into a novel strategy for managing the redundant standoff axis, utilizing a deep reinforcement learning approach to achieve M path smoothing. Section IV showcases the simulation outcomes and analyzes the results. Conclusively, Section V summarizes the study and outlines directions for future research.

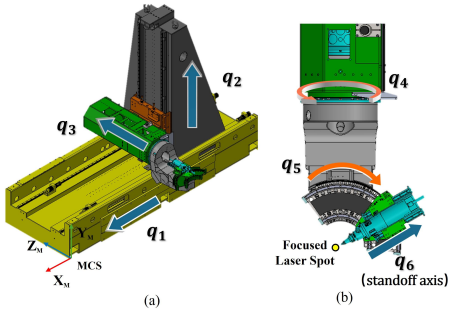


Fig. 1: The schematic of EFORT 3D laser cutting machine. (a) overview; (b) details on the laser head.

II. TOOL ORIENTATION OPTIMIZATION

A. Tool path representation

The tool path of a workpiece is described in workpiece coordinate system (WCS) and it is represented through parametric B-spline curves, denoted as

$$\mathbf{C}(u) = \sum_{i=0}^{n} B_{i,p}(u)\mathbf{P}_i, \quad u \in [0,1], \tag{1}$$

where u is the normalized geometric parameter along the path and $\mathbf{P}_i$ represents the collection of n control points dictating the shape of the B-spline. Concurrently, $B_{i,p}(u)$ denotes the i-th B-spline basis function of order p in the variable u. The values of u where the pieces of polynomial meet define the sequence $\{u_0, u_1, \ldots, u_{n+p+1}\}$, known as knot vector. Such basis functions are derived recursively as

$$\begin{cases} B_{i,0}(u) = \begin{cases} 1, & u_i \le u \le u_{i+1} \\ 0, & \text{else} \end{cases}, \\ B_{i,p}(u) = \frac{u-u_i}{u_{i+p}-u_i} B_{i,p-1}(u) + \frac{u_{i+p+1}-u}{u_{i+p+1}-u_{i+1}} B_{i+1,p-1}(u). \end{cases} \tag{2}$$

The feedrate interpolation algorithm [16] facilitates the planning of interpolated CL points along single or multiple segments of the tool path. The N discrete geometric parameters, denoted as u_t, are derived from the interpolation process, which incorporates tangential constraints, as

$$u_t = f(t), \qquad t \in 1, 2, \ldots, N, \tag{3}$$

where f represents the feedrate interpolation function, determining the CL position at index t as $\mathbf{C}(u_t) = [x(u_t), y(u_t), z(u_t)]$.

Once the CL positions along the curve are defined, the reference direction of tool orientation at each intermediate point can be obtained through interpolation of the normal vectors at the path's start and end points. In this study, the QI algorithm [7] is applied to compute the tool orientation $\mathbf{O}(u)$, where $\mathbf{O}(u_t) = [i(u_t), j(u_t), k(u_t)]$ is the interpolated unit direction of the tool, and its parameterized form can be written as:

$$\mathbf{O}(u_t) = \frac{\sin\left((1-u_t)\theta_Q\right)}{\sin\theta_Q}\mathbf{O}(0) + \frac{\sin\left(u_t\theta_Q\right)}{\sin\theta_Q}\mathbf{O}(1), \tag{4}$$

where $\theta_Q = \arccos\left(\mathbf{O}(0)\cdot\mathbf{O}(1)\right)$, $\mathbf{O}(0), \mathbf{O}(1)$ represent the tool orientations at the start and end points, respectively, and $\cdot$ is the dot product between two vectors. These directions are determined by normal vectors perpendicular to the workpiece surface.

B. C-space method

In this section, feasible C-spaces are computed to define the collision-free zone for CL points on the workpiece. Similarly to the approach in [1], the C-spaces are constructed in the plane of two rotary axes in MCS, instead of using inclination and tilt angles in LCS at CL points, as in other methods. However, to ensure that the C-space retains the constraints of machining tolerance in LCS, uniform sampling within the constrained subspace at constant intervals is performed, and the sampled candidates are mapped into MCS to construct the C-space. Interference between the laser head components and the workpiece is checked for collisions for each candidate within the constructed C-space.

Firstly, the tool orientation must be constrained in a domain for each CL point to ensure the quality of machining. A Frenet-based LCS located on a CL point is shown in Fig. 2. The classical machining constraints are simply represented in a geometric shape, as in [4]. In laser machining applications, the tool orientation is determined according to the direction of the laser beam. Then, the constrained domain of tool orientation variation is defined by the maximum allowable lead and tilt variations of the laser beam angle with respect to $\mathbf{O}(u)$. The tolerances are usually defined relating to the customized manufacturing quality and the curvature of a reference tool path.

Secondly, to facilitate the calculation of equivalent solutions in MCS, the constrained region must undergo a transformation

from the local LCS to the global WCS. This is achieved through a simple multiplication by the workpiece transformation matrix.

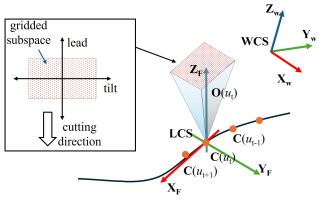


Fig. 2: Geometric representation of machining constraints based on a local Frenet frame with zoomed view of gridded coordinates on lead and tilt angles.

The transformation between WCS and MCS is detailed in the previous work [18], and recalled here as

$$\begin{bmatrix} i(u) \\ j(u) \\ k(u) \end{bmatrix} = \begin{bmatrix} \cos q_4(u) \sin q_5(u) \\ \sin q_4(u) \sin q_5(u) \\ \cos q_5(u) \end{bmatrix}, \tag{5}$$

where q_4 and q_5 are the joint variables of the laser head, as in Fig. 1, i.e., they represent the coordinates of the rotary axes, which can be obtained using classical IKT approach as

$$\begin{cases} q_4(u) = \arctan 2(j(u), i(u)) + a\pi, a = 0, \pm 1, \\ q_5(u) = \arccos(k(u)). \end{cases} \tag{6}$$

Finally, all candidates computed in the subspace of MCS must be checked for collision avoidance. While in milling applications only checking for collisions between the tool and the workpiece is typically sufficient, in the case of a 3D laser cutting head the collision avoidance issue is more complex. The volume body of the 3D laser head is larger than typical milling tools, and its configuration is asymmetrical when viewed around the q_4 axis along the tool direction, implying that also movements of the rotary axes can potentially lead to collisions. In this paper, collision checking encompasses all components on the head that may be close to the workpiece. The C-space will then consist of feasible regions and forbidden areas, as determined by collision checking.

Fig. 3 sketches the four main steps for the construction of the entire C-space. Fig. 3(a) shows the initial step of establishing machining constraints in the LCS, focusing on lead and tilt angles. Subsequently, Fig. 3(b) depicts how the constrained subspace is represented as a global orientation ij-plane in the WCS. The third step, as shown in Fig. 3(c), involves deriving the rotary axes subspace in the MCS following IKT. Finally, Fig. 3(d) presents the completed C-space, in which feasible (blue) and forbidden (red) regions are distinguished after collision detection.

Once the C-space is established at each CL point along the tool path, attention must turn to manage the transitions between adjacent tool orientations. This step is crucial for maintaining the smoothness outlined in Section I, setting the stage for the transition into the optimization model discussion.

C. The optimization problem

The choice of candidates within the feasible region of C-spaces for all CL points along the tool path plays a critical role in ensuring the smooth motion of the rotary axes. It is intuitive that finding the most suitable solution within each C-space along a tool path, as illustrated in Fig. 4, is quite

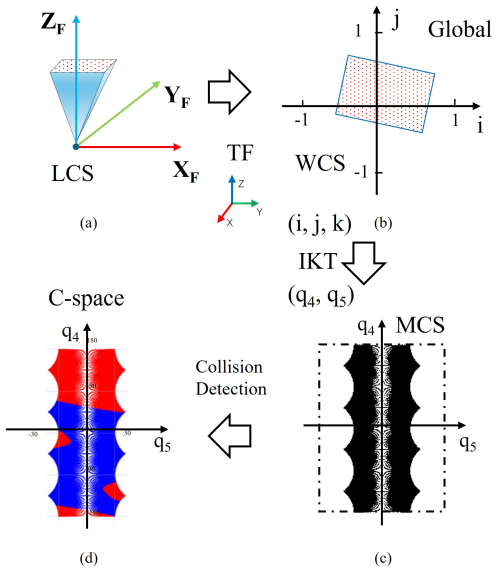


Fig. 3: A detailed visualization of the four step process in constructing the C-space.

similar to selecting vertices across each layer in a graph to determine the shortest path. Many graph-based searching methods for tool orientation optimization have been developed in past contributions [5], [6], [19]. This section is devoted to the construction of the graph representing the large-scale discrete C-spaces, and to the definition of the weight between each couple of vertices for the optimization model.

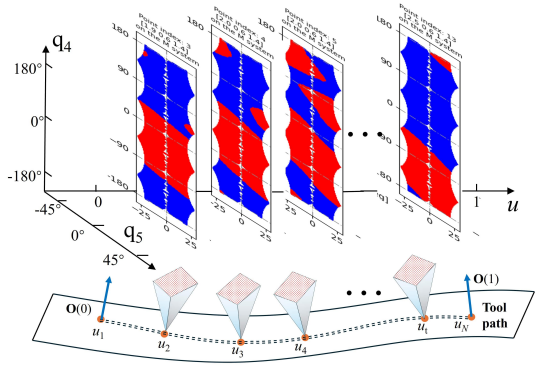


Fig. 4: A schematic diagram of discrete C-spaces and feasible tool orientation candidates at CL points on the tool path

To demonstrate the concept with a simplified model, a graph $G = \{V, E\}$ is constructed, consisting of a sequence of stages. Each stage contains vertices representing possible states within the C-space. The graph begins at a start vertex v_0 and concludes at an end vertex v_5. As depicted in Fig. 5(a), each stage, except for the first and the last one, contains multiple vertices. The graph consists of a vertex set V and an edge set E. However, in this graph, the shortest path calculation considers only directly connected states that are adjacent to each another. The difference graph $G^* = \{V^*, E^*\}$ was presented in [6] for three states evaluation by considering the difference of original edges. Each element of V^* represents

one edge, and elements in E^* are vertices connecting incident edges in G, as in Fig. 5(b).

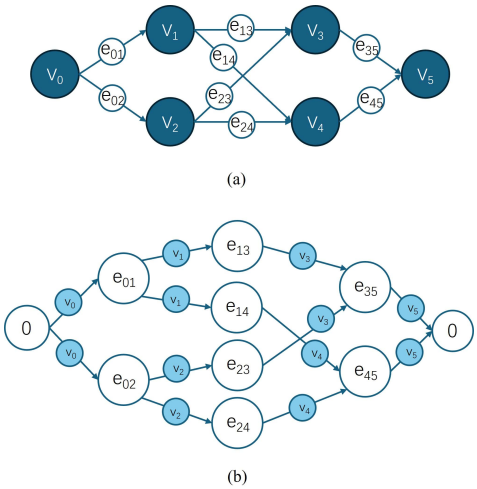


Fig. 5: A simple example of graph for orientation optimization. (a) original graph G; (b) difference graph G^* [6]

In this study, to achieve a smooth change in tool orientations, the aim is to minimize not only the deviations of the rotary axes but also the effects of accelerations.

Here, the model that defines the kinematic performance of the angular deviation of rotary axes is presented. Let q_{R} denote the coordinate of one of the rotary axes (i.e., q_4 or q_5), as given in (6). The angular velocities and accelerations can be computed by using forward and central difference method as

$$\dot{q}_{\mathrm{R}}(u_t) = \frac{q_{\mathrm{R}}(u_{t+1}) - q_{\mathrm{R}}(u_t)}{u_{t+1} - u_t}, \tag{7}$$

$$\ddot{q}_{\mathrm{R}}(u_t) = \frac{2\left(\dot{q}_{\mathrm{R}}(u_t) - \dot{q}_{\mathrm{R}}(u_{t-1})\right)}{u_{t+1} - u_{t-1}}, \tag{8}$$

where u_{t-1} and u_{t+1} are the previous and the next interpolated parameter, respectively. Notably, the continuity of the multi-turn capability of the rotary axis q_4 is achieved as

$$\Delta q_4 = \begin{cases} \Delta q_4' - 2\pi, & \text{if} \quad \Delta q_4' > \pi \\ \Delta q_4', & \text{if} \quad -\pi < \Delta q_4' < \pi \\ \Delta q_4' + 2\pi, & \text{if} \quad \Delta q_4' < -\pi \end{cases}, \tag{9}$$

where $\Delta q_4' = q_4(u_{t+1}) - q_4(u_t)$.

Velocities and accelerations in (7), (8) are bounded by the drive limits of the rotary axes as

$$|\dot{q}_{\mathrm{R}}(u_t)| \le \dot{q}_{\mathrm{R,max}}, \quad |\ddot{q}_{\mathrm{R}}(u_t)| \le \ddot{q}_{\mathrm{R,max}}, \tag{10}$$

where $\dot{q}_{\mathrm{R,max}}$ and $\ddot{q}_{\mathrm{R,max}}$, indicate the limits of the corresponding axis. It is worth to note that those vertices for which the kinematic constraints in (10) are not satisfied, are automatically removed while constructing the graph G to enhance the computational efficiency.

In order to take into account also the effects of the accelerations, and achieve a smooth tool orientation trajectory, the following optimization objective function is introduced as

$$\min \sum_{t=1}^{N} \sqrt{\dot{q}_{\mathrm{R}}(u_t)^2 + \lambda \ddot{q}_{\mathrm{R}}(u_t)^2}, \tag{11}$$

subject to the constraints in (10). λ is a scalar used as penalty parameter on the accelerations to tune the smoothness of the tool orientation trajectory.

Overall, the optimized tool orientations $\mathbf{O}^*(u_t)$ can be found by searching the shortest path in G^*. Dijkstra's algorithm is adopted to efficiently solve the graph-based tool orientation optimization problem.

III. PLANNING FOR REDUNDANT STANDOFF AXIS

The challenge of planning collision-free and smooth movements for rotary axes was addressed in the previous section. However, in many high speed machining applications, particularly in 3D laser cutting, significant reductions in the feedrate typically occur due to sharp angular changes within short segments of the tool path, despite efforts to ensure smooth angular transitions. This issue is attributed to the inherent mechanical offsets present in the rotational mechanisms of traditional five-axis machines, which result in unnecessary displacements at the tool tip during rotary axes movements. Consequently, the machine translational axes are responsible for compensating these rotational offsets within short segments to maintain the tool's tip along the intended tool path.

The standoff axis, an external translational axis commonly found in laser cutting machines, as illustrated in Fig. 1, usually serves the single purpose of maintaining a constant distance between the nozzle tip and the workpiece surface. In this paper, for the first time, the standoff axis is proposed to be employed as a further redundant sixth DOF q_6 on five-axis machines, denoted simply as the w axis for simplicity. The stroke of the redundant standoff axis w axis is extended with respect to the traditional 3D laser machines, as detailed in Table I.

TABLE I: Comparison of standoff axis stroke range between the EFORT machine and traditional 3D laser cutting 5-axis machine

Stroke	$w_{\min}$ (mm)	$w_{\max}$ (mm)
EFORT redundant 3D laser machine	-100	20
Traditional 3D laser machine	-12	12

A. Concept of M path

Many studies [12], [13], [15], [17] have demonstrated that second orientation curve methods in five-axis machines can effectively smooth angular movements. On the other hand, the capability of dual curves methods to further smooth the movement of translational axes is often underestimated.

Leveraging the characteristic of the standoff axis to move along the tool axis direction, the optimization of the second curve is proposed through the planning of the w axis value. A novel curve is introduced from the tool curve, named M path in this paper, having variable height. The concept of M path is illustrated in Fig. 6.

Thanks to the zero-offset mechanical structure design of the 3D laser cutting head, as shown in Fig. 7(a), the discrete M points on the M path can be represented by a crossing point of rotary axis q_4 and q_5. The M points are denoted as $\mathbf{M}(u_t)$, the position of each point is calculated by moving from the corresponding CL point $\mathbf{C}(u_t)$ by the amount $w(u_t)$ along the optimized tool orientation $\mathbf{O}^*(u_t)$, as:

$$\mathbf{M}(u_t) = \mathbf{C}(u_t) + w(u_t)\mathbf{O}^*(u_t). \tag{12}$$

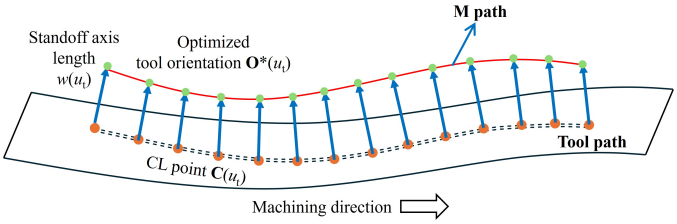


Fig. 6: The M path concept

To maintain the generality of the M path concept, the equation (12) for typical 3D laser cutting head structures is then adapted by incorporating offset values as

$$\mathbf{M}(u_t) = \mathbf{C}(u_t) + (w(u_t) + l_2)\mathbf{O}^*(u_t) + l_1\vec{v}(u_t), \quad (13)$$

where l_1 and l_2 represent the mechanical offsets, as shown in Fig. 7(b). The figure also compares the mechanical structure of the EFORT 3D laser cutting head with the one of a conventional head, to better highlight the differences with the conventional approach used in 3D laser cutting machines. The vector $\vec{v}(u_t)$ denotes the unit direction orthogonal to the optimized tool orientation $\mathbf{O}^*(u_t)$, calculated as

$$\vec{v}(u_t) = \frac{\mathbf{C}(u_{t+1}) - \mathbf{C}(u_t)}{\|(\mathbf{C}(u_{t+1}) - \mathbf{C}(u_t))\|_2} \times \mathbf{O}^*(u_t). \quad (14)$$

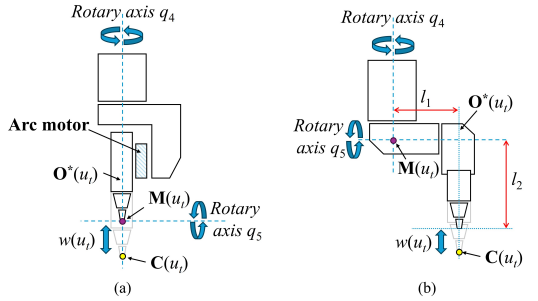


Fig. 7: Mechanical structure of 3D laser cutting heads. (a) zero-offset head on EFORT machine; (b) conventional head on traditional 3D laser cutting machines.

Given the discrete geometric parameter u_t, the tangential velocity and acceleration along the M path can be computed as

$$v_{\mathrm{M}}(u_t) = \frac{\|\mathbf{M}(u_{t+1}) - \mathbf{M}(u_t)\|_2}{u_{t+1} - u_t}, \quad (15)$$

$$a_{\mathrm{M}}(u_t) = \frac{v_{\mathrm{M}}(u_{t+1}) - v_{\mathrm{M}}(u_t)}{u_{t+1} - u_t}, \quad (16)$$

where $\|\cdot\|_2$ represents the Euclidean norm.

B. Reinforcement learning framework

In automotive body-in-white industrial manufacturing, parts within the same category, such as A-pillars, B-pillars, C-pillars, door rings, etc., often exhibit slight variations due to the customized designs specific to different car models and brands. The diversity in shapes and paths complicates the process, as different parts require tailored optimization models. Reinforcement Learning (RL) offers significant advantages in this context. It enables systems to adapt quickly to varying conditions without the need for reprogramming or extensive retraining. The proposed solution uses an RL-based approach to achieve a smoother M path. The fundamental principle of RL is to identify and execute the most appropriate action for a given state, and then adjust subsequent actions based on environmental feedback.

This section outlines the construction of the action space, state space, and the formulation of the reward function, all of which are crucial components of the RL framework.

Action Space Configuration: The action a_t, defined as a one-dimensional axis value w, necessitates configuration within a continuous space to accommodate the flexible plan required for precise operations. Additionally, the action space must conform to the stroke constraints outlined in Table I:

$$a_t \in [w_{\min}, w_{\max}]. \quad (17)$$

State Space Design: It is essential to define a comprehensive set of state space observations for effective planning and optimization. Firstly, our primary focus is on monitoring potential collisions that may occur following adjustments to the w axis; the episode terminates if a collision, denoted by $c(u_t) = 1$, is detected. Further observations include the tangential velocity $v_{\mathrm{M}}(u_t)$ and acceleration $a_{\mathrm{M}}(u_t)$ at each parameter u_t, providing crucial insights into the dynamic behavior necessary to achieve the optimization objectives. Additionally, the state of the redundant axis value w is included as a reference to inform and refine the action strategies. These observations collectively constitute a rich state space that facilitates the learning process. Consequently, the state s_t is defined as

$$s_t = [c(u_t), v_{\mathrm{M}}(u_t), a_{\mathrm{M}}(u_t), w(u_t)]. \quad (18)$$

Reward function: Another critical aspect of RL involves the immediate reward function $r(s_t, a_{t-1})$, which serves as feedback to the agent based on its present state s_t and the preceding action a_{t-1} that led to that state. To continuously identify a smoother M-path, it is imperative to define a quantitative indicator that can assess the improvement of prospective solutions. To achieve this goal, a variable Γ_{M} is then defined, depending on the acceleration, to be used as a key indicator for the reward function, to evaluate the progression towards an optimized trajectory. The indicator is defined as

$$\Gamma_{\mathrm{M}}(t) = \sum_{t=1}^{N} \|a_{\mathrm{M}}(u_t))\|_2. \quad (19)$$

If no collision occurs in the current state, a smooth transition is desirable between the two adjacent points on M path to ensure optimal kinematic performance. This requirement guides the configuration of the correlated rewards based on the extent of performance improvements. The final reward function is thus defined as:

$$r(s_t, a_{t-1}) = \begin{cases} -1000 & \text{if } c(u_t) = 1, \\ -\Delta\Gamma_{\mathrm{M}}(t) & \text{if } c(u_t) = 0, \end{cases} \quad (20)$$

where $\Delta\Gamma_{\mathrm{M}}(t) = \Gamma_{\mathrm{M}}(t) - \Gamma^{\dagger}_{\mathrm{M}}(t)$, being $\Gamma^{\dagger}_{\mathrm{M}}(t)$ the reference indicator, computed as in (19) but using the acceleration values obtained before the trajectory optimization. $\Delta\Gamma_{\mathrm{M}}(t)$ is crucial for understanding how changes in the M path's smoothness impact the reward mechanism.

C. Deep deterministic policy gradient

The task requires a continuous action space, as the redundant standoff axis w must vary smoothly within the working range to optimize M path effectively. In reinforcement learning, Policy Gradient (PG) methods are widely used for managing continuous action spaces, by determining the optimal policy through action probabilities. Building on PG methods, the Deterministic Policy Gradient (DPG) method utilizes a deterministic policy function, μ, which directly specifies actions, thereby mitigating the computational intensity of sampling across the action space. Deep Deterministic Policy Gradient [20] (DDPG), an enhancement of DPG, incorporates characteristics from Deep Q-Networks (DQN) to improve performance and efficiency through deterministic policies.

In this study, DDPG is adopted to manage the redundant standoff axis w for complex tool paths, driven by its efficacy in continuous spaces and its robust adaptation capabilities essential for precise M path adjustments. The architecture of DDPG is illustrated in Fig. 8, and a brief overview of DDPG algorithm is provided.

The actor-critic architecture of DDPG plays a central role, where the actor network is $\mu(s_t|\theta^\mu)$ responsible for proposing actions that the critic network $Q(s_t, a_t|\theta^Q)$ then evaluates, where θ^μ and θ^Q are the actor and critic network parameters, respectively. This interplay ensures that the policy developed by the actor is not only feasible, but also optimized according to the long-term expected rewards assessed by the critic.

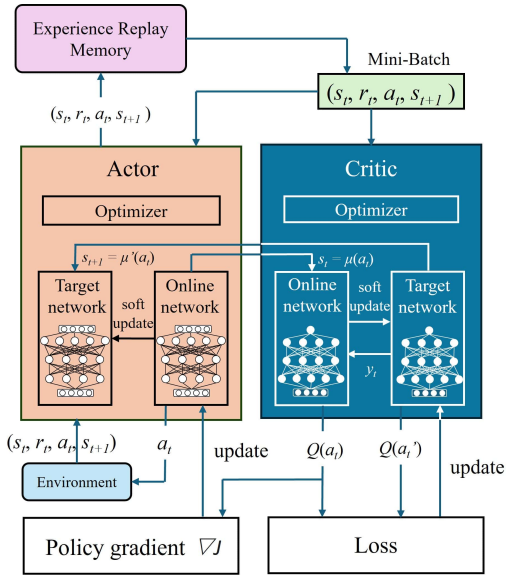


Fig. 8: DDPG architecture

The critic's role is to compute the Q-function $Q^\mu(s_t, a_t)$, which quantifies the expected rewards for a given state-action pair as defined in (21). It is instrumental in evaluating the performance of the actions suggested by the actor. Similarly to DQN algorithm, function approximators are used to estimate the policy and value function through network evaluations. Throughout the learning process, the DDPG agent continuously updates the properties of both the actor and critic at each timestep. Additionally, the use of target networks helps to provide stable learning targets during temporal difference updates, which helps avoiding potential negative interferences in the learning process. The Q-function is defined as:

$$Q^\mu(s_t, a_t) = \mathbb{E}\left[r(s_t, a_t) + \gamma Q^\mu(s_{t+1}, \mu(s_{t+1}))\right], \quad (21)$$

where the $\mathbb{E}$ symbol represents the expected value. γ is the discount factor that weighs the importance of future rewards.

In each iteration, the experience (s_t, a_t, r_t, s_{t+1}) is stored in the experience buffer. Then, random mini-batches of m experiences are sampled from the experience buffer to update actor and critic.

The learning efficiency is assessed through the computation of the critic's target and online losses L. The equation of loss is defined as:

$$L = \frac{1}{2m}\sum_{t=1}^{m}\left(y_t - Q\left(s_t, a_t|\theta^Q\right)\right)^2. \quad (22)$$

By minimizing these losses, the critic network increasingly refines its predictions of the Q-function, thereby directing the actor network to select more beneficial actions, ultimately leading to smooth control and optimization.

IV. EXPERIMENTAL VALIDATION

The primary goal of the experiments is to validate the effectiveness of the proposed optimization algorithm for the intricate and industry-relevant task of cutting the outer contours of a B-pillar in an automotive body-in-white. This component was selected due to its inclusion of bent sections or steps, which pose a significant challenge in five-axis machining, because of the extensive tool orientation changes required over short path lengths. The object workpiece and tool path are depicted in Fig. 9.

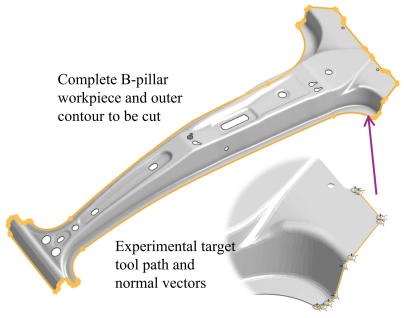


Fig. 9: Overview of B-pillar workpiece and the path focused on the experiment

The methodologies outlined in Sections II and III were executed on an i5-10210 CPU @ 1.6GHz (8 cores) equipped with 32 GB of memory platform utilizing Python along with several open-source packages including pythonocc, pybullet, networkx, and gym.

The experimental simulation is conducted in several key phases: 1) Target tool paths and normal vectors at the CL points are generated from the CAD file of the B-pillar workpiece; 2) Tool orientations are optimized for collision avoidance and smoothness using a graph-based approach; 3) The proposed RL model is trained to derive a smooth M path, with the trained parameters saved in a zipped dataset. 4) The tool path is executed using the loaded dataset to validate the enhanced smoothness of the translational axes.

The results of the graph-based tool orientation optimization are reported in Fig. 10. In the objective function (11), the

penalty parameter λ is set to 0.05 to account for angular rotational accelerations, resulting in smoother rotary axes movements compared to the case where $\lambda = 0$, which only considers angular differences. These results demonstrate the benefits of using difference graph [6] and the successful achievement of collision-free tool orientations path.

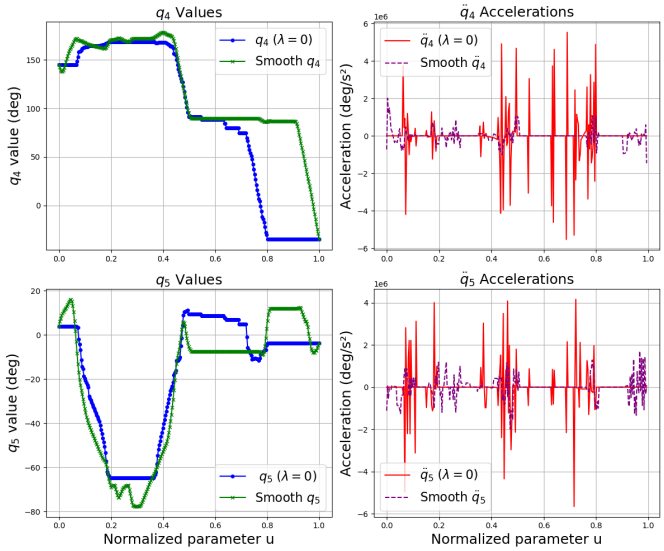


Fig. 10: Collision-free rotation axes q_4, q_5 plots in axes position and acceleration.

The learning process for the redundant w axis in the RL model is depicted in Fig. 11. The first image (a) shows the non-smooth M path when the w axis remains static, suggesting that the three discontinuities observed may due to rapid angular changes in the rotary axes. The second image (b) presents the exploration process prior to the activation of the learning algorithm. Subsequent images (c) and (d) display the exploitation of the M path below or above the workpiece surface, respectively. The final image (f) demonstrates a smoother M path compared to image (e), achieved on the basis of the application of learning from experience replay and rewards obtained after 3,000 learning steps.

The learning model, including DDPG actor and critic network weights, is stored in a database file. To evaluate if the smoothed M path effectively optimizes acceleration on the translational axes, the impacts of a fixed-length w axis (pre-optimization) were compared with those of a flexibly moving w axis (post-optimization). The results, reported in Fig. 12, clearly show that the acceleration of the translational axes at sharp peaks has been significantly reduced thanks to the optimization applied, resulting in a noticeable smoothing of the trajectory.

V. Conclusions and Future Works

Smooth and collision-free trajectory planning significantly enhances the efficiency and precision of five-axis machines. Abrupt changes in both translational and rotary axes can induce vibrations in the machine, adversely affecting machining accuracy and necessitating reduced speeds. However, enhancing the kinematic performance of primary translational axes beyond that of rotary axes poses a challenge due to the inherent inflexibility of five-axis machines. Consequently, leveraging a redundant axis becomes essential to augment the machine's capabilities, thereby boosting its performance in high-speed and high-precision machining tasks.

In this research, collision-free C-spaces are firstly constructed at each CL point along the tool path. Candidates of feasible regions are sampled gridded subspaces, adhering to the machining constraints from LCS to MCS. The optimal rotary axes are determined by evaluating the deviation and acceleration between two C-spaces in the shortest path from the difference graph, through the application of the Dijkstra's algorithm. After the introduction of the M path concept, a reinforcement learning task is then implemented to optimize the M path by planning the redundant standoff axis using DDPG. The simulation results validate the proposed approach, showing smoother performance in the primary translational axes, while maintaining optimal tool orientation. The proposed methodology represents a significant advancement in high-speed, high-precision machining, especially for 3D laser cutting and other non-contact machining applications. By leveraging the standoff as a redundant axis, the approach introduces a new level of flexibility that improves the capabilities of conventional five-axis machines. This innovation not only enhances the adaptability and efficiency of machining processes, but also represents a substantial step forward in the evolution of machining technology.

The proposed two-phase solution is still in its initial stage, but it shows potential improvement through the definition of a single, multi-objective optimization problem. Moving forward, the experiment will be extended to more paths with pretrained RL dataset. Next, a more advanced strategy will be adopted, integrating tool orientation optimization and redundant axis planning within a unified reinforcement learning framework, achieved by expanding the action space dimension. Furthermore, the applicability of this approach will be explored beyond 3D laser cutting to include a wider range of practical machining applications.

References

[1] Y. Wang, J. Xu, and Y. Sun, "Tool orientation adjustment for improving the kinematics performance of 5-axis ball-end machining via cpm method." *Robotics and Computer-Integrated Manufacturing*, vol. 68, p. 102070, 2021.

[2] S. Sun, P. Zhao, T. Zhang, B. Li, and D. Yu, "Smoothing interpolation of five-axis tool path with less feedrate fluctuation and higher computation efficiency," *Journal of Manufacturing Processes*, vol. 109, pp. 669–693, 2024.

[3] C.-S. Jun, K. Cha, and Y.-S. Lee, "Optimizing tool orientations for 5-axis machining by configuration-space search method," *Computer-Aided Design*, vol. 35, no. 6, pp. 549–566, 2003.

[4] C. Castagnetti, E. Duc, and P. Ray, "The domain of admissible orientation concept: a new method for five-axis tool path optimisation," *Computer-Aided Design*, vol. 40, no. 9, pp. 938–950, 2008.

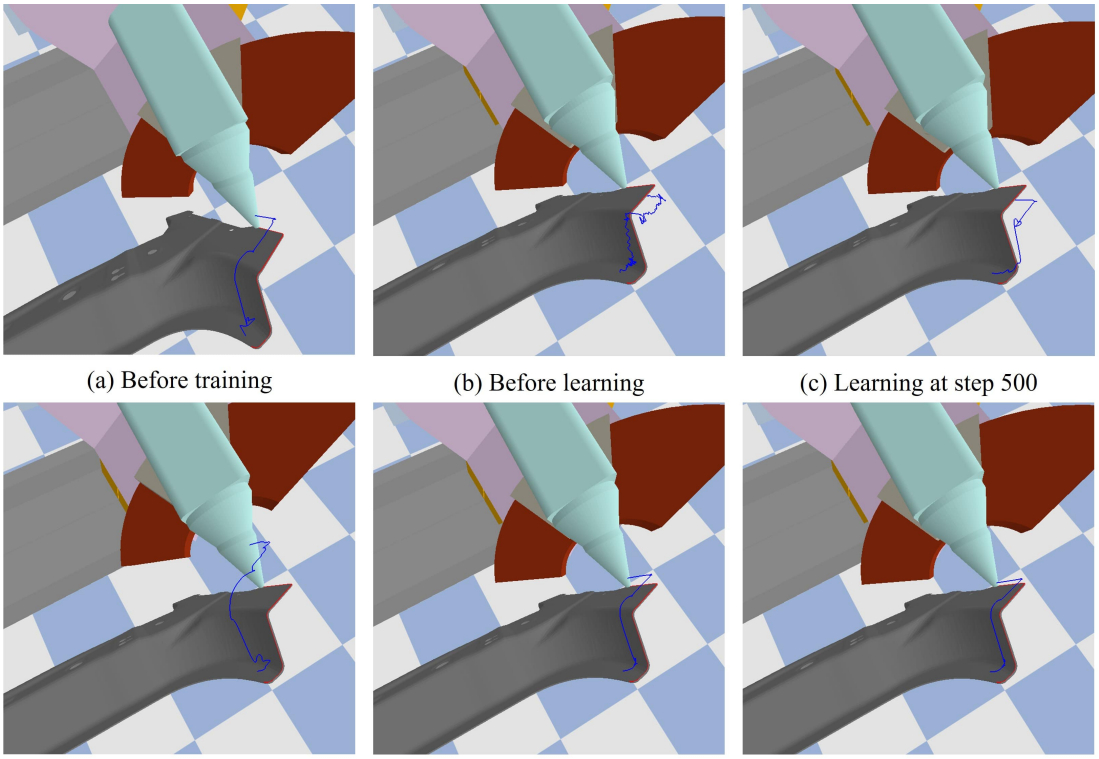
[5] D. Plakhotnik and B. Lauwers, "Graph-based optimization of five-axis machine tool movements by varying tool orientation," *The International Journal of Advanced Manufacturing Technology*, vol. 74, pp. 307–318, 2014.

[6] Z. Mi, C.-M. Yuan, X. Ma, and L.-Y. Shen, "Tool orientation optimization for 5-axis machining with c-space method," *The International Journal of Advanced Manufacturing Technology*, vol. 88, pp. 1243–1255, 2017.

[7] M.-C. Ho, Y.-R. Hwang, and C.-H. Hu, "Five-axis tool orientation smoothing using quaternion interpolation algorithm," *International Journal of Machine Tools and Manufacture*, vol. 43, no. 12, pp. 1259–1267, 2003.

[8] J. Xu, D. Zhang, and Y. Sun, "Kinematics performance oriented smoothing method to plan tool orientations for 5-axis ball-end cnc machining," *International Journal of Mechanical Sciences*, vol. 157, pp. 293–303, 2019.

[9] L. Wu, J. Xu, X. Yin, and Y. Sun, "Jerk-optimal piecewise planning of tool orientation for 5-axis ball-end machining with linearized kinematic constraints," *Journal of Manufacturing Science and Engineering*, vol. 145, no. 7, p. 071003, 2023.



(a) Before training (b) Before learning (c) Learning at step 500

(d) Learning at step 1000 (e) Learning at step 2000 (f) Learning at step 3000

Fig. 11: Reinforcement learning training process.

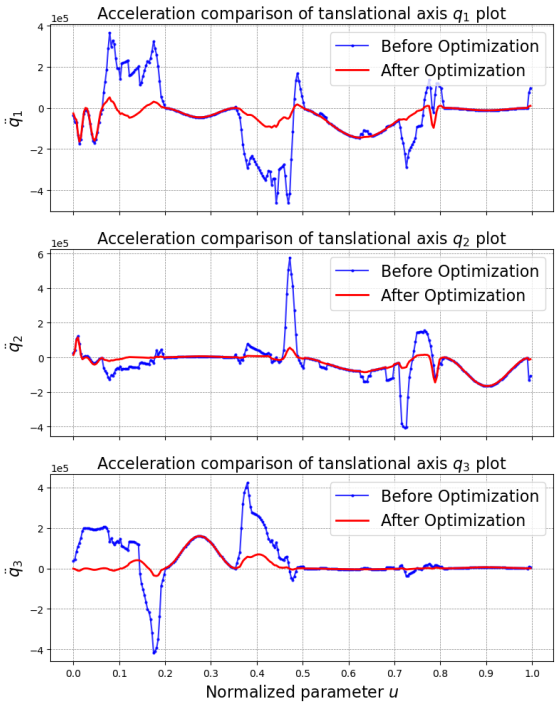


Fig. 12: The comparison of three primary translational axes q_1, q_2, q_3 accelerations before and after the redundant standoff axis w optimization

[10] Z. Gong, B. Li, H. Zhang, and P. Ye, "Tool orientation optimization method based on ruled surface using genetic algorithm," *The International Journal of Advanced Manufacturing Technology*, pp. 1–14, 2022.

[11] Y. Zhang, Y. Li, and K. Xu, "Reinforcement learning–based tool orientation optimization for five-axis machining," *The International Journal of Advanced Manufacturing Technology*, vol. 119, no. 11, pp. 7311–7326, 2022.

[12] J. Zhang, L. Zhang, K. Zhang, and J. Mao, "Double nurbs trajectory generation and synchronous interpolation for five-axis machining based on dual quaternion algorithm," *The International Journal of Advanced Manufacturing Technology*, vol. 83, pp. 2015–2025, 2016.

[13] D. Li, W. Zhang, W. Zhou, T. Shang, and J. Fleischer, "Dual nurbs path smoothing for 5-axis linear path of flank milling," *International Journal of Precision Engineering and Manufacturing*, vol. 19, pp. 1811–1820, 2018.

[14] X. Gao, S. Zhang, L. Qiu, X. Liu, Z. Wang, and Y. Wang, "Double b-spline curve-fitting and synchronization-integrated feedrate scheduling method for five-axis linear-segment toolpath," *Applied Sciences*, vol. 10, no. 9, p. 3158, 2020.

[15] J. Xiao, H. Zhang, and B. Li, "A dual nurbs curve five-axis interpolator with interval division under axial kinematic constraints," *IEEE/ASME Transactions on Mechatronics*, 2023.

[16] X. Beudaert, S. Lavernhe, and C. Tournier, "Feedrate interpolation with axis jerk constraints on 5-axis nurbs and g1 tool path," *International Journal of Machine Tools and Manufacture*, vol. 57, pp. 73–82, 2012.

[17] J.-w. Ma, Z.-y. Jia, F.-z. Qin, D.-n. Song, W.-w. Jiang, and 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*, vol. 141, no. 6, p. 061002, 2019.

[18] Z. Ding, P. Soccio, M. Indri, and A. Rizzo, "Through hole-cutting conic posture optimization for a redundant 3d laser cutting machine," *The International Journal of Advanced Manufacturing Technology*, pp. 1–19, 2024.

[19] Y. Gao, S. Mi, H. Zheng, Q. Wang, and Z. Wei, "An energy efficiency tool path optimization method using a discrete energy consumption path model," *Machines*, vol. 10, no. 5, p. 348, 2022.

[20] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.