A New Trajectory Planning Approach With Motion Duration Control for Kinematic Constrained Systems

Lucas de Andrade Both and Felix Lange

***Abstract*—Trajectory planning for industrial applications such as pick-and-place, welding or material handling yields production enhancement and economic improvements. The problem of trajectory planning given an initial and target state is a fundamental issue for robotic applications and automation in general. This paper presents a novel algorithm to calculate the actuator's motion profile between two states considering its kinematic constraints. The key feature of the presented concept is to use more conservative values for the kinematic constraints to find a motion profile, such that the target state can be reached given a fixed trajectory duration. One advantage of this approach is the synchronization possibility with further actuators, whose trajectories durations are fixed and not controllable. Finally, an application case using a real system is described, and the simulation results are presented using the proposed algorithm.**

***Index Terms*—Constrained motion planning, kinematics, motion and path planning.**

I. Introduction

MECHANICAL manipulators are nowadays widely used in industrial production lines and manufacturing systems. Given the growing necessity and trend for larger scale productions considering shorter fabrication time turns the topic motion planning into a crucial field for robotics and automation [1].

According to refs. [2], [3] the motion planning can be divided intro three hierarchical subproblems: (i) the specification of a geometric path obeying the rules of the environment (*path planning*), (ii) the time parametrization along the geometric path (*trajectory planning*), and (iii) the low-level control loop (*path tracking*). This paper focuses exclusively on the trajectory planning problem.

The crucial aspect in trajectory planning is computing the actuator's motion for an initial and target state such that the whole profile remains within the kinematic limits up to a certain order [4]. The planners minimizing the execution time are prominent in research and industry since they contribute to increasing productivity [4]. Nevertheless, finding the minimum time trajectory might not be the exclusive goal for some applications. Most of the current time-optimal trajectories are computed using continuous time functions, however in recent years there has been a rapid increase in the use of digital controllers working with discrete time domain [5]. Besides, some actuators must not operate in isolation, but combined with other actuators [6], [7]. Such scenarios do not require time-optimal planners that reduce the time execution to a minimal level, but solutions to compute the trajectories based on a fixed duration, that still respect the kinematic limits.

One approach to compute the trajectory given an initial and target state for a kinematic constrained actuator is presented by refs. [8] and [9]. In this case, the system is considered as a pure sequence of integrators (e.g. double or triple integrators) and the control input is defined as the highest system's order (e.g. acceleration or jerk). Using the Pontryagin's minimum principle (PMP), a time-optimal control law in the form of Switching Phase Graphs can be obtained, where the control input switches between upper bound, 0 and lower bound [8], [9]. Besides, it is also possible to respect velocity and acceleration constraints (internal system states). The main disadvantage of the approaches described by refs. [8], [9] is that the trajectories can only be computed using integration methods (e.g. Runge Kutta) and numerical simulations. If the motion duration between initial and target state is not a multiple of the simulation time step, then it is not possible to compute exactly the motion duration, and it is not possible to reach exactly the target state, resulting in high control input oscillations near the target state [10].

One possible method of trajectory planning given a fixed time duration employs sigmoid functions during the transitions of the jerk values, which guarantees smoother motion profiles [11]. This approach assures that the trajectory between an initial and target state respects the constraints for velocity, acceleration and jerk. However, to assure multi-actuators synchronization (i.e. motion duration control) Fang et al. [11] stretched the time lengths of the trajectories for each actuator linearly with a synchronization factor. The main disadvantage of this idea is that it is only successful when the initial and target conditions are null for velocity and acceleration.

Further trajectory planning algorithms with motion duration control presented in the literature use either third order polynomials respecting velocity, acceleration and jerk bounds

Manuscript received 24 June 2023; accepted 24 October 2023. Date of publication 15 November 2023; date of current version 28 November 2023. This letter was recommended for publication by Associate Editor J. Hu and Editor H. Kurniawati upon evaluation of the reviewers' comments. This work was supported by the Digital Photonic Production DPP Research Campus as part of the Research Campus Public-Private Partnership for Innovation research funding initiative of the German Federal Ministry of Education and Research (BMBF). As part of the German Government's high-tech strategy, the BMBF is using this initiative to promote strategic and long-term cooperation between science and industry "under one roof" under Grant 13N15423. *(Corresponding author: Lucas de Andrade Both.)*

Lucas de Andrade Both is with the Chair for Laser Technology, RWTH-Aachen, 52062 Aachen, Germany (e-mail: lucas.de.andrade.both@llt.rwth-aachen.de).

Felix Lange is with the SCANLAB GmbH, 82178 Puchheim, Germany (e-mail: felix.lange2@rwth-aachen.de).

Digital Object Identifier 10.1109/LRA.2023.3333237

for non-null initial and target states [3], [12], a jerk-bounded trigonometric S-curve trajectory [13] or waypoints under kinematic constraints [14]. These algorithms generate smoother trajectories, but can not assure that maximal allowable jerk is attained during Acceleration/Deceleration segments due to a conservative synchronization strategy [11].

Berscheid et al. [15] improved the canonical third-order trajectory profile consisting of seven steps [3] and detailed deeply further trajectory profiles with fewer steps quantity. The main idea behind the fixed duration feature is to extend the canonical third-order profile with a constant velocity step, such that the time duration for the new profile is equal to the requested by the user. However, not every trajectory profile described by Berscheid et al. [15] can be easily adapted with a constant velocity step.

This letter describes a method to calculate the trajectory for a single device given initial and target states, the kinematic constraints and a fixed time duration. Compared to the methods previously described, the main advantages of this novel approach are: (i) initial and target states can be different from zero, (ii) kinematic constraints of the profile can always be kept and (iii) the method is independent of the used profile. The innovative aspect is to build the trajectory based on a conservative value for the highest-order constraint, providing control over the duration time. The content of this research is organized as follows: Section II defines mathematically the problem to be solved and presents the notation employed in this work. Sections III and IV describe trajectory planners based on the kinematic limits of the actuator. Section V presents a novel approach to adjust the computed motion profile from both previous sections such that the new profile complies with the requirements defined for the motion duration. Finally, Section VI details a real assembly, the simulation results using the algorithm developed in Section V and comparisons with the approach described by ref. [15].

II. Problem Definition and Notations

Using a similar notation to refs. [15], [16], let $\mathcal{C} = \mathbb{R}^{n+1}$ denote the $(n+1)$-dimensional configuration space, and let $\mathcal{F}$ denote the subset of configurations that respect the kinematic limits of one single actuator. $\mathbf{x}(t)$ describes the state of this single actuator at time t, and it is defined by the position $p(t)$ and its partial derivatives up to nth-order

$$\mathbf{x}(t) = \Big[p(t) \quad p'(t) = \tfrac{\partial p(t)}{\partial t} \quad p''(t) = \tfrac{\partial^2 p(t)}{\partial t^2} \quad \dots \quad \dots \quad p^{(n)}(t) = \tfrac{\partial^n p(t)}{\partial t^n} \Big]^\top .$$

For mechanical systems, the first, second and third derivatives represent velocity, acceleration and jerk respectively.

Given an initial state $\mathbf{x}_0$ and a target (final) state $\mathbf{x}_f$ for a single actuator, we seek the trajectory $\mathbf{x}^*(t)$ defined by

$$\mathbf{x}^*(t) = f\left(\Delta_T^*, \mathbf{x}_0, \mathbf{x}_f, p'(t), p''(t), \dots, p^{(n)}(t)\right) \tag{1}$$

TABLE I
Steps of Constant Acceleration of an Extremal Profile

Step k	Acceleration	Limit
t_1	$\overline{\text{ACC}}$	-
t_2	0	$\overline{\text{VEL}}$
t_3	$\underline{\text{ACC}}$	-

satisfying the constraints

$$\underline{\text{VEL}} = \underline{p'} \le p'(t) \le \overline{p'} = \overline{\text{VEL}}$$
$$\underline{\text{ACC}} = \underline{p''} \le p''(t) \le \overline{p''} = \overline{\text{ACC}}$$
$$\underline{\text{JERK}} = \underline{p'''} \le p'''(t) \le \overline{p'''} = \overline{\text{JERK}}$$
$$\vdots$$
$$|p^{(n)}(t)| \le \overline{p^n}$$

for all times $t \in [t_0, t_0 + \Delta_T^*]$, where Δ_T^* is called motion duration. For notation simplicity we spare the term (t) for further analysis, such that $p' = p'(t)$ or $\mathbf{x}^* = \mathbf{x}^*(t)$.

Finally, the valid **Extremal Profile** (EP) is defined as a trajectory describing the actuator's motion between t_0 and $t_0 + \Delta_T^*$. It is contained in $\mathcal{F}$ and simultaneously uses almost full kinematic resources of the respective actuator. To assure the constraint limits, the profiles present a "bang-off-bang" behavior [17]. More precisely, $\mathbf{x}^*(t)$ is formulated based on the extreme values of the nth-order derivative, $p^{(n)}$, and the motion duration, Δ_T^*, is divided into a number of fixed steps where for each step $p^{(n)} \in \{-\overline{p^n}, 0, \overline{p^n}\}$.

III. Acceleration Constrained Trajectory Planner

The first method to compute the motion between $\mathbf{x}_0$ and $\mathbf{x}_f$ is based on the results described by ref. [16]. It is relevant to point out that this approach does not consider the duration control, since its main goal is to describe a technique to minimize the motion time between start and target state, however it was the base for the novel algorithm described in Section V. According to the descriptions from [16], a time-optimal trajectory is computed for a single actuator considering up to the second partial derivative and at the same time entire kinematic resources. So given the initial and target values for velocity and position it is possible to compute the motion values between them. An additional requirement must be fulfilled: A velocity limit might only be reached once in an EP.

Introducing steps of constant acceleration before and after the step, where the velocity limit is reached leads to a maximal number of 3 steps, between initial and target state with corresponding duration t_k. Table I presents the acceleration values and reached kinematic limits at each step duration t_k. It is important to emphasize that the problem is invariant to a sign change, resulting in an equivalent extremal profile however with opposite signs for the acceleration and kinematic limit values. Furthermore, as a consequence for the definitions described in Section II, $\underline{\text{ACC}} = -\overline{\text{ACC}}$.

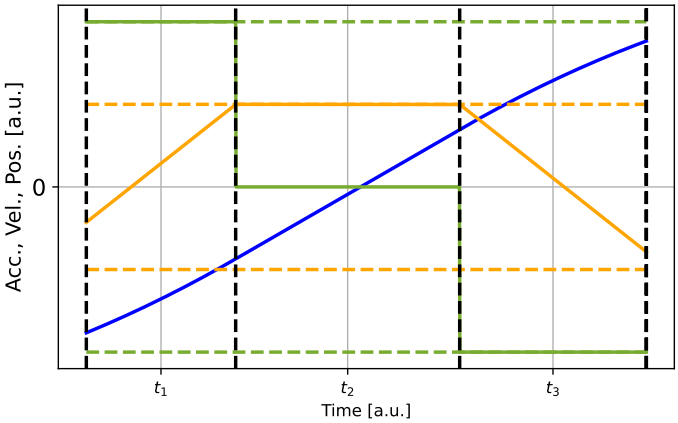

Fig. 1. Illustrative example for a motion with time-optimal profile, with position (——), velocity (——), acceleration (——), velocity bounds (- - -) and acceleration bounds (- - -).

TABLE II
EXTREMAL PROFILES FOR ACCELERATION CONSTRAINED TRAJECTORY PLANNER

# Extremal Profile	Acc. Profile	Limits	Condition
1	$(\overline{\text{ACC}}, 0, \underline{\text{ACC}})$	$\overline{\text{VEL}}$	$v_2 = \overline{\text{VEL}}$
2	$(\overline{\text{ACC}}, \underline{\text{ACC}})$	NONE	$t_2 = 0$

Fig. 1 illustrates the fundamental EP described in Table I given an initial and target state with non-null conditions. Mathematically, an extremal profile maps the given kinematic constraints, initial and target states

$$\mathcal{S}1 : \left(\mathbf{x}_0, \mathbf{x}_f, \overline{\text{VEL}}, \overline{\text{ACC}}, \underline{\text{ACC}}\right) \mapsto \mathbf{t}_k$$

to the corresponding step time array. For this case $\mathbf{t}_k = [t_1, t_2, t_3]^\top$.

Considering the enumerated kinematic (2), it is possible to find the duration of each step time, where $t_0 = 0$ and $\mathbf{x}_0 = [p_0, v_0]^\top$ are the initial conditions.

$$v_{k+1} = v_k + a_k t_k \tag{2a}$$

$$p_{k+1} = p_k + v_k t_k + \frac{a_k t_k^2}{2} \tag{2b}$$

Nevertheless, ref. [16] also considers the possibility of $t_2 = 0$, which leads to a motion without reaching $\overline{\text{VEL}}$. Table II enumerates all possible EPs, its features and the reached limits. Finally, the motion duration is then given by

$$\Delta_T = \sum_{k=1}^{N} t_k, \tag{3}$$

where N is the maximal number of steps. Based on Fig. 1, $N = 3$ for this section.

The main feature of this approach is that all time steps can be analytically solved for each extremal profile described in Table II using (2), resulting in exact solutions and short computing time. Furthermore, it is interesting to point that by means of the PMP, the Switching Phase Graph described by [8] results in the acceleration profiles from Table II [11].

TABLE III
STEPS OF CONSTANT JERK OF AN EXTREMAL PROFILE

Step k	Jerk	Limit
t_1	$\overline{\text{JERK}}$	-
t_2	0	$\overline{\text{ACC}}$
t_3	$\underline{\text{JERK}}$	-
t_4	0	$\overline{\text{VEL}}$
t_5	$\underline{\text{JERK}}$	-
t_6	0	$\underline{\text{ACC}}$
t_7	$\overline{\text{JERK}}$	-

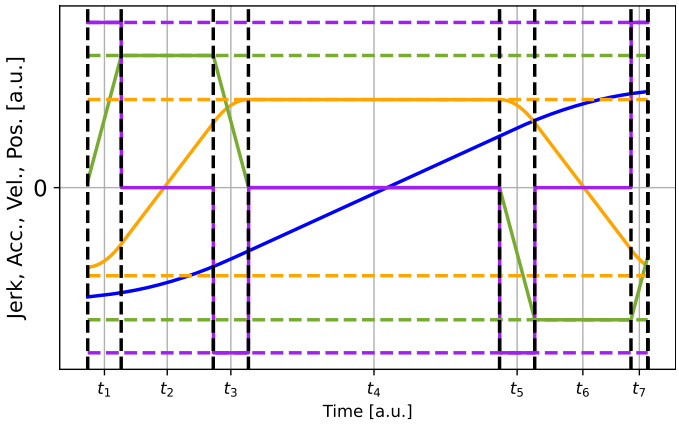

Fig. 2. Illustrative example for a motion with time-optimal profile, with position (——), velocity (——), acceleration (——), jerk (——), velocity bounds (- - -), acceleration bounds (- - -) and jerk bounds (- - -).

IV. JERK CONSTRAINED TRAJECTORY PLANNER

The second approach to compute the motion between $\mathbf{x}_0$ and $\mathbf{x}_f$ is based on the method described by ref. [15]. Differently of the technique detailed in Section III, it considers up to the third partial derivative to the time-optimal trajectory, resulting in a "bang-off-bang" jerk profile. As mentioned by ref. [15], third-order constraints are desirable to reduce mechanical stress, wear and overall cost of robots over their lifetime.

For this approach, the motion is computed for one single actuator considering its constraints and the following requirements:

- A velocity limit might only be reached once in an extremal profile.
- There are only up to two acceleration limits in an extremal profile.
- $a_f \leq \sqrt{2 \cdot \overline{\text{JERK}} \cdot \max(|\overline{\text{VEL}} - v_f|, |\underline{\text{VEL}} - v_f|)}$, where a_f and v_f are target acceleration and target velocity respectively.

Thus, only up to three limits can be reached: First, the upper acceleration bound, $\overline{\text{ACC}}$, second the upper velocity bound, $\overline{\text{VEL}}$, and third the lower acceleration bound, $\underline{\text{ACC}}$. Introducing steps of constant jerk before, after and between the limits lead to a maximal number of $N = 7$ steps between initial and target state with corresponding duration t_k. Using the results described in ref. [15], Table III displays the jerk values and the reached kinematic limits at each step. Similar to the technique described in Section III the problem is invariant to a sign change, resulting in an equivalent extremal profile however with opposite signs for the jerk and kinematic limit values. Fig. 2 illustrates the

TABLE IV
EXTREMAL PROFILES FOR JERK CONSTRAINED TRAJECTORY PLANNER

# Extremal Profile	Jerk Profile	Reached Kin. Limits	Condition I	Condition II	Condition III	Condition IV
1	$(\overline{\text{JERK}}, 0, \underline{\text{JERK}}, 0, \underline{\text{JERK}}, 0, \overline{\text{JERK}})$	$\overline{\text{ACC}}$, $\overline{\text{VEL}}$, $\underline{\text{ACC}}$	$a_1 = \overline{\text{ACC}}$	$v_3 = \overline{\text{VEL}}$	$a_5 = \underline{\text{ACC}}$	$a_3 = 0$
2	$(\overline{\text{JERK}}, 0, \underline{\text{JERK}}, 0, \underline{\text{JERK}}, \overline{\text{JERK}})$	$\overline{\text{ACC}}$, $\overline{\text{VEL}}$	$a_1 = \overline{\text{ACC}}$	$v_3 = \overline{\text{VEL}}$	$t_6 = 0$	$a_3 = 0$
3	$(\overline{\text{JERK}}, \underline{\text{JERK}}, 0, \underline{\text{JERK}}, 0, \overline{\text{JERK}})$	$\overline{\text{VEL}}$, $\underline{\text{ACC}}$	$t_2 = 0$	$v_3 = \overline{\text{VEL}}$	$a_5 = \underline{\text{ACC}}$	$a_3 = 0$
4	$(\overline{\text{JERK}}, \underline{\text{JERK}}, 0, \underline{\text{JERK}}, \overline{\text{JERK}})$	$\overline{\text{VEL}}$	$t_2 = 0$	$v_3 = \overline{\text{VEL}}$	$t_6 = 0$	$a_3 = 0$
5	$(\overline{\text{JERK}}, 0, \underline{\text{JERK}}, 0, \overline{\text{JERK}})$	$\overline{\text{ACC}}$, $\underline{\text{ACC}}$	$a_1 = \overline{\text{ACC}}$	$t_4 = 0$	$a_5 = \underline{\text{ACC}}$	$t_5 = 0$
6	$(\overline{\text{JERK}}, 0, \underline{\text{JERK}}, \overline{\text{JERK}})$	$\overline{\text{ACC}}$	$a_1 = \overline{\text{ACC}}$	$t_4 = 0$	$t_6 = 0$	$t_5 = 0$
7	$(\overline{\text{JERK}}, \underline{\text{JERK}}, 0, \overline{\text{JERK}})$	$\underline{\text{ACC}}$	$t_2 = 0$	$t_4 = 0$	$a_5 = \underline{\text{ACC}}$	$t_5 = 0$
8	$(\overline{\text{JERK}}, \underline{\text{JERK}}, \overline{\text{JERK}})$	NONE	$t_2 = 0$	$t_4 = 0$	$t_6 = 0$	$t_5 = 0$

fundamental extremal profile described in Table III given $\mathbf{x}_0$ and $\mathbf{x}_f$ with values for acceleration and velocity different from null. Finally, it is relevant to mention that for this algorithm $\underline{\text{JERK}} = -\overline{\text{JERK}}$, as previously described in the Section II.

Mathematically, the new extremal profile maps the given kinematic constraints, initial and target states

$$\mathcal{S}2 : \left(\mathbf{x}_0, \mathbf{x}_f, \overline{\text{VEL}}, \overline{\text{ACC}}, \underline{\text{ACC}}, \overline{\text{JERK}}, \underline{\text{JERK}}\right) \mapsto \mathbf{t}_k$$

to corresponding step time array, $\mathbf{t}_k$. For this case $\mathbf{t}_k = [t_1, t_2, \ldots, t_7]^\top$. Given the system with these 7 variables, it is necessary to use the 3 kinematic equations ((4)) to find the motion considering the limits detailed in Table III, where the initial conditions of motion are $t_0 = 0$ and $\mathbf{x}_0 = [p_0, v_0, a_0]^\top$.

$$a_{k+1} = a_k + j_k t_k \tag{4a}$$

$$v_{k+1} = v_k + a_k t_k + \frac{j_k t_k^2}{2} \tag{4b}$$

$$p_{k+1} = p_k + v_k t_k + \frac{a_k t_k^2}{2} + \frac{j_k t_k^3}{6} \tag{4c}$$

However, ref. [15] improves it and, similarly to the method detailed in Section III, develops further extremal profiles that are still valid but do not reach the kinematic limits. In such cases it is still necessary to use the 3 equations for position, velocity and acceleration. The remaining additional conditions are basically set by a zero step duration, where the value of jerk was originally zero. Table IV shows all possible extremal profiles considering the jerk values sequence developed by [15]. Thus, given an initial state, $\mathbf{x}_0$, and a target state, $\mathbf{x}_f$, it is necessary to find the step time array considering all 16 possible extremal profiles and the (4). However, not all solutions are physically possible or depending on initial and target states some extremal profiles lead to no solution. Hence, using the results from ref. [15] an EP is valid if

- $t_k \geq 0$;
- $\underline{\text{ACC}} \leq a_k \leq \overline{\text{ACC}}$;
- $\underline{\text{VEL}} \leq v_k \leq \overline{\text{VEL}}$;

for all $1 \leq k \leq 7$. Finally, the motion duration is given by (3) with $N = 7$.

It is relevant to point out that by means of the PMP, the Switching Phase Graph described in [9] results in the jerk profiles from Table IV [11]. Besides, in order to find the step duration for some profiles described in Table IV it is necessary to resolve non-analytical equations. For this purpose the Newton-Raphson method was adopted with kinematic constraints as initial estimation.

Algorithm 1: Motion Duration Control.

Input: Δ_T^*, $\mathbf{x}_0$ and $\mathbf{x}_f$ for an actuator and its respective constraints.
Output: New trajectory respecting actuator's constraints.

1: $\mathbf{T}$ ▷ Matrix with step time array for each EP.
2: N ▷ Quantity of EPs.
3: **for** i $\leftarrow$ 1 to N **do**
4: **if** $i^{th}-$EP is valid **AND** $\Delta_{T_i} \leq \Delta_T^*$
5: $\mathbf{T}_i \leftarrow \mathbf{t}_{k_i}$ ▷ Add $\mathbf{t}_{k_i}$ to step array matrix.
6: $\mathbf{t}_k \leftarrow \arg\min(\sum \mathbf{T}_i - \Delta_T^*)$
7: $\Delta_T = \sum_k^i t_k$. ▷ Equivalent to (3).
8: **if** $\Delta_T = \Delta_T^*$
9: $\mathbf{x}^*(t) \leftarrow f(\Delta_T^*, \mathbf{x}_0, \mathbf{x}_f, p', p'', \ldots, p^{(n)})$
10: **return** $\mathbf{x}^*(t)$
11: **else**
12: Compute new $\widetilde{p^n} < \overline{p^n}$ such that $|\Delta_T(\widetilde{p^n}) - \Delta_T^*| < \epsilon$
13: $\widetilde{\mathbf{x}^*}(t) \leftarrow f(\Delta_T(\widetilde{p^n}), \mathbf{x}_0, \mathbf{x}_f, p', p'', \ldots, \widetilde{p^n})$
14: **return** $\widetilde{\mathbf{x}^*}(t)$

V. MOTION DURATION CONTROL WITH KINEMATIC CONSTRAINTS

The main advantage of the trajectory planning methods such as described in Section III and IV is the possibility of determining exactly the trajectory duration prior to its execution. Hence, changing the constraints values for the highest orders is also possible to control the motion duration. This is a completely new approach of controlling the motion duration. It can be used to generate trajectory durations that are equal to a desired value pre-defined by the user, e.g. to synchronize additional devices to the existing actuators.

Algorithm 1 describes the pseudo-code that enables the trajectory duration control for synchronization using the planners described in the previous sections.

The algorithm to compute the motion profile controlling the time duration is basically divided into two segments. It begins with the initialization of the $\mathbf{T}$ matrix, that stores each time step array for the N profiles. The first segment (Lines 1–7 in Algorithm 1) aims on finding valid EPs, given the actuator's constraints, initial and target conditions. As additional condition the motion duration for the EP must be shorter than Δ_T^* (Line 4), which is vital for the second part of the algorithm. For the methods described in Sections III and IV there are respectively

2 and 8 EPs (see Tables II and IV). Depending on constraints and boundary conditions, multiple EPs can provide a valid $\mathbf{t}_k$ (Line 5). Since the objective is to find a profile with motion duration Δ_T^*, one needs to choose $\mathbf{t}_{k_i}$[1] with the closest duration to Δ_T^* (Line 6 in Algorithm 1). It is important to point out that Algorithm 1 used a more general notation, since it can be used for the planner from Section III as well from Section IV.

The second part (Lines 8–14) computes the new motion profile. If the motion duration for the chosen EP is equal to the desired Δ_T^*, then the new motion profile can be computed using (2) or (4) and the respective EP (Line 8 to 10). Otherwise, it is necessary to adapt the chosen extremal profile. The main idea is defined in line 12 of Algorithm 1, where $\widetilde{\overline{p^n}}$ is the calculated highest-order constraint (Acceleration or Jerk) that is smaller than $\overline{p^n}$ and at the same time provides a trajectory $\widetilde{\mathbf{x}^*}(t)$ with motion duration $\Delta_T(\widetilde{\overline{p^n}})$ such that $|\Delta_T(\widetilde{\overline{p^n}}) - \Delta_T^*| < \epsilon$, where ϵ is a user defined error limit for the difference between the computed and expected duration time along the iterations.

Since there is no analytical solution for computing $\widetilde{\overline{p^n}}$ in order to achieve $\Delta_T(\widetilde{\overline{p^n}}) = \Delta_T^*$, iterative methods must be applied and a "stop condition" between the iterations must be defined to maintain the deviation between $\Delta_T(\widetilde{\overline{p^n}})$ and Δ_T^* as small as possible.

One efficient way to search for $\widetilde{\overline{p^n}}$ is to use numerical methods to find the minimum of a function considering one variable. Eq (5) summarizes this idea, where in this case $\overline{p_x^n}$ is defined as the n^{th} derivative variable.

$$\widetilde{\overline{p^n}} = \underset{\overline{p_x^n}}{\arg\min} \left|\Delta_T\left(\overline{p_x^n}\right) - \Delta_T^*\right| \tag{5}$$

From the first part of the algorithm 1 one has the EP and the respective duration time. Since $\Delta_T(\overline{p^n}) < \Delta_T^*$ (see line 4 in Algorithm 1) the motion must be "slowed down" automatically resulting in $\widetilde{\overline{p^n}} < \overline{p^n}$. In other words $\widetilde{\overline{p^n}}$ is more conservative than the original value respecting the kinematic limits of the actuator. Methods like Nelder–Mead or Gradient Descent can be used to solve (5). Restructuring (5) it is also possible to apply Newton root-finding algorithm, however based on the tests reported in Section VI high constraints values cause ill-conditioned systems, for which might difficult to find a solution. On the other hand, minimizer methods proved to be more stable and returned valid $\widetilde{\overline{p^n}}$ for the tests detailed in the next section. Hence a valid "stop condition" for (5) is

$$\left|\Delta_T\left(\widetilde{\overline{p^n}}\right) - \Delta_T^*\right| < \epsilon. \tag{6}$$

The main advantage of Algorithm 1 compared to ref. [15] is that it can be applied to every EP, since all profiles are based on the maximal value of $\overline{p^n}$. Defining a constant velocity step, as proposed by ref. [15], can not be easily applied to all EPs from Table IV for the motion duration control. This new feature is illustrated in section VI, where simulation results for an industrial application scenario are provided.

[1]For better comprehension, it is relevant to mention $\Delta_T = \sum_k^{t_{k_i}}$ as defined in (3).

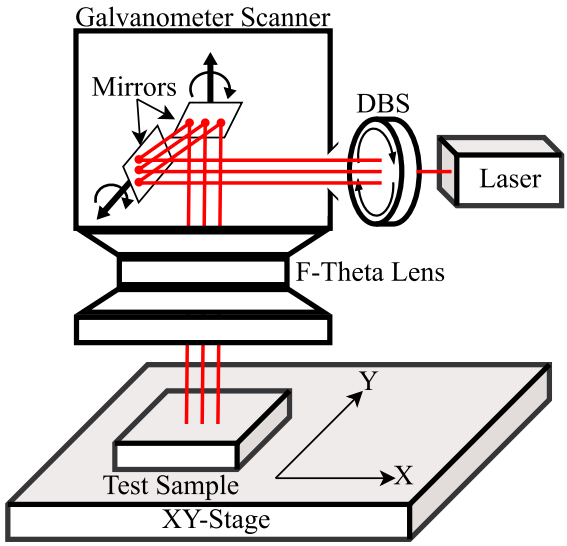

Fig. 3. Galvanometer Scanner, XY-Stage and rotating DBS.

VI. Application Case

To validate the method detailed in Section V using the motion planners described in Sections III and IV an application case in micromachining processes with laser was adopted. The assembly illustrated in Fig. 3 is composed by a galvanometer scanner (excelliSCAN 14 from the company Scanlab GmbH), a granite XY servo stage from the company BUSCH Microsystems GmbH and the Diffractive Beam Splitter (DBS) model MS-581-I-N-X from the company HOLO/OR. The Laser source is an ultrashort pulse (USP) laser module from the company EdgeWave GmbH.

This is a typical system setup for laser material processing applications such as microdrilling in printed circuit boards for electronics [18], [19] or metal plates used as spray holes designed to create specific fluid flow patterns [20], [21].

The Galvanometer Scanner presents high kinematic limits, which provides short time responses, however it covers a small imaging field (for this application case about 40 mm × 40 mm). On the other hand, the XY-Stage provides greater displacements (150 mm × 150 mm), but the kinematic limits are lower than those of the galvanometer scanner. Hence, the combination of both devices results in a system that makes the galvanometer scanners' dynamics available over an increased total processing range. Besides that, a F-Theta lens was attached to the scanner to focus the laser beam onto a planar image plane, which improves the process. The DBS provides the system with a multibeam matrix that is used to parallelize processing yielding a higher machine productivity [22], [23].

To control galvo-scanner and stage during the laser processes, the software solution syncAXIS (Scanlab GmbH) was used. Based on a target geometry syncAXIS computes the trajectories for a Galvanometer Scanner and XY-Stage such that the combination of both trajectories produces the target geometry and at the same time keep the kinematic limits of both devices. These trajectories are discretized using a sampling frequency

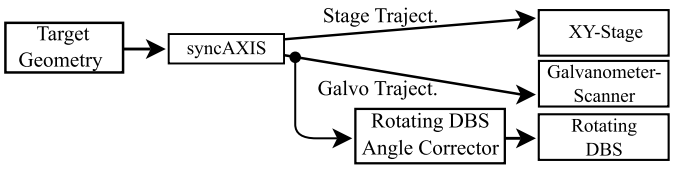

Fig. 4. Data Generation Scheme for the Laser Process using Galvo-Scanner, XY-Stage and rotating DBS.

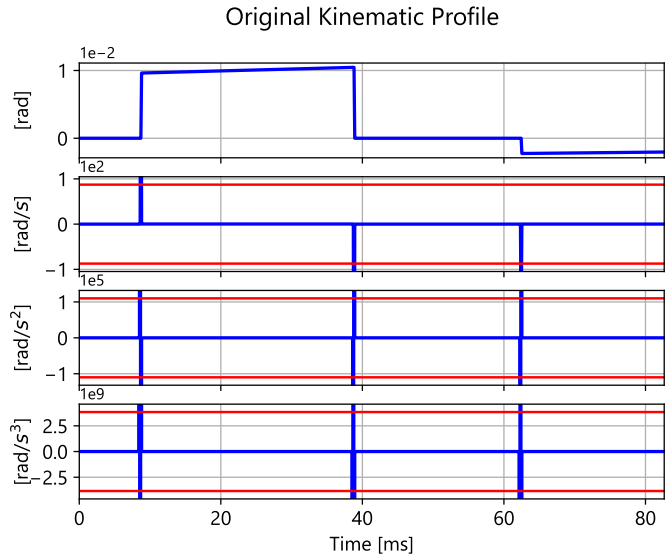

Fig. 5. Original kinematic profile without corrections. Red lines (—) represent the respective kinematic limits.

of 100 kHz. The main feature of syncAXIS is that the motion of Scanner and Stage is synchronous and complementary during the job, which reduces the total process time and avoids stitching effects [24].

However, based on results from ref. [25], combining a DBS with a F-Theta lens implies a distortion of the expected multi-beam matrix. Besides that, the distortion of the beam matrix depends on the Galvo-Scanner deflection angle, where the greater the deflection by the Galvo-Scanner, the greater the distortion of the beam matrix. This effect can be partially corrected by rotating the DBS, since the distortion is proportional to the mirrors deflections. Using the results from ref. [25] it was possible to build a lookup table relating Galvo-Scanner deflection with the necessary DBS rotation to partially correct the distortion. Thus, based on the points computed by the syncAXIS for Galvo-Scanner given a target geometry, one can calculate the respective DBS rotation using an interpolation of the lookup table to correct the distortion from the multi-beam laser matrix.

Fig. 4 presents a short overview for the data generation and control considering the assembly illustrated in Fig. 3.

Since syncAXIS does not consider the rotational DBS, when computing the trajectories for Galvo-Scanner and XY-Stage, there is no guarantee that the kinematic limits of its actuator will be respected while computing the control values for steering the DBS from all consequently required correction angles. Fig. 5

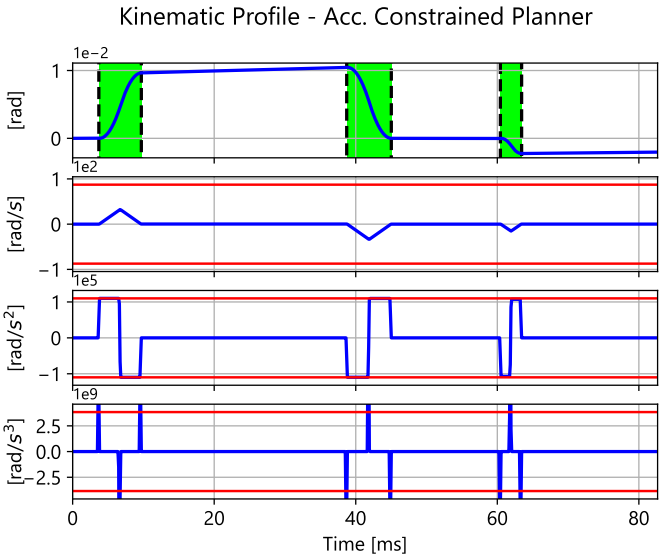

Fig. 6. Kinematic profile with Acceleration Constrained Trajectory Planner (Section III). The red lines (—) represent the respective kinematic limits. The green areas (■) represent the new computed motion profile using Algorithm 1.

exposes the motion profile for the DBS calculated directly from the Galvo-Scanner trajectories for an example geometry. In this profile the maximal velocity, acceleration and jerk are exceeded when the DBS must turn abruptly. Finally, for the experiments illustrated in Fig. 5 the manufacturer's default constraints values were considered, i.e., $-\underline{\text{VEL}} = \overline{\text{VEL}} = 9\text{E}1\ \text{rad/s}$, $-\underline{\text{ACC}} = \overline{\text{ACC}} = 1.2\text{E}5\ \text{rad/s}^2$, $-\underline{\text{JERK}} = \overline{\text{JERK}} = 4\text{E}9\ \text{rad/s}^3$.

The idea is to use the algorithm 1 to smooth the motion, when the original kinematic limits are not maintained, such that the target state can be reached given a time duration. On the other hand, since the controller used to steer the DBS also presents a cycle frequency of 100 kHz, the imposed trajectory duration needs to be multiple of 10 μs. Figs. 6 and 7 display the corrected motion profiles applying algorithm 1 using the planners described in Sections III and IV respectively. The curves inside the green shaded intervals display the new computed profiles and the motion profile outside the green intervals have the same values as presented in Fig. 5. Furthermore, the left vertical and right vertical edges of each shaded interval (black dashed lines) indicate the initial and target states respectively used in Algorithm 1.

It is also important to point out that for both figures the Nelder-Mead simplex algorithm was applied to find a minimum as described in (5). In order to improve the results such that the deviation between $\Delta_T(\widetilde{p^n})$ and Δ_T^* is as small as possible, the error limit from (6) was set $\epsilon = 1\text{E}{-}10$ to finish the iterations. Despite the small error limit, ϵ, the planned trajectories respected the kinematic limits imposed by the actuator[2] as seen in Figs. 6 and 7. Additionally, the corrected trajectory has the same length as the original one, which assures the synchronization of the

[2]It is important to point out that the jerk was not respected in Fig. 6. This result was expected since the method described in Section III doesn't consider jerk to compute the trajectories.

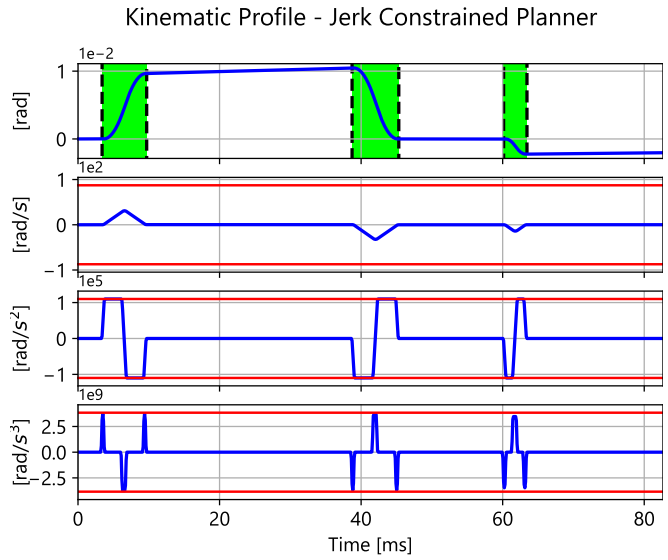

Fig. 7. Kinematic profile with Jerk Constrained Trajectory Planner (Section IV). The red lines (—) represent the respective kinematic limits. The green areas (■) represent the new computed motion profile using Algorithm 1.

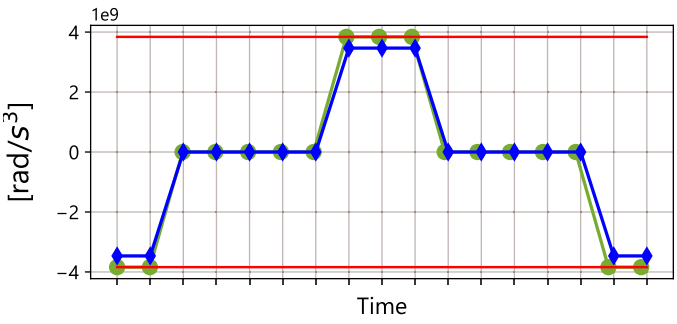

Fig. 8. Kinematic profile around time $t = 60$ ms from Fig. 7. Green curve (—●—) represents the non-corrected profile and blue curve (—◆—) represents corrected profile with Algorithm 1 for the requested duration time Δ_T^*. The red lines (—) represent the respective kinematic jerk limits.

process among all devices. Finally it is also possible to see that $\widetilde{\overline{p^n}} < \overline{p^n}$ for both cases.

Fig. 8 displays the corrected jerk-profile around $t = 60$ ms regarding Fig. 7. Applying the planner described in Section IV leads to a motion duration $\Delta_T = 307.45$ μs, that is not exact a multiple of 10 μs. In a similar procedure using algorithm 1 for an imposed $\Delta_T^* = 320.00$ μs the simulations resulted in a more conservative value for jerk, such that $\Delta_T(\widetilde{\overline{p^n}}) - \Delta_T^* = 2\text{E} - 13$ μs. The time step size of 20 μs given in Fig. 8 is chosen for better visibility of the differences in the kinematic profiles. It is possible to realize that the markers for the blue curves are better fitted with the vertical grid than the green ones, as a consequence of the imposed duration times Δ_T^*, which are a multiple of the sampling time.

The main contribution of the method presented in Section V compared with ref. [15] is that computing iteratively $\widetilde{\overline{p^n}}$ can be applied for all the profiles described in Table II and IV. The idea described by ref. [15] of using a constant velocity step to reach

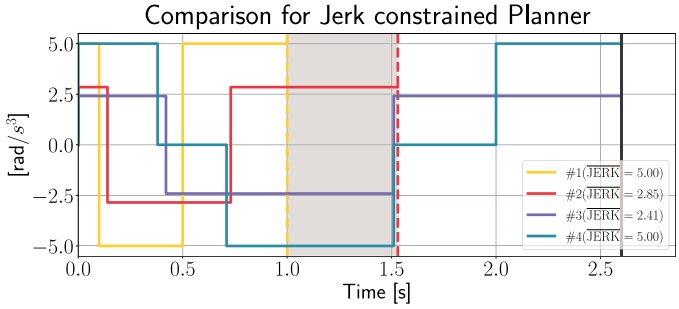

Fig. 9. Comparison between Algorithm 1 and ref. [15]. Yellow dashed line (- - -) represents the motion time for curve #1, red dashed line (- - -) represents the motion time for curve #2 and black solid line (—) represents the motion time for curves #3 and #4.

motion duration control is not directly applicable to EP #8 in Table IV. Fig. 9 displays a new test comparing the advantage of the Algorithm 1 to the results of ref. [15]. It is important to point out that these results are not related to the tests regarding the system of Fig. 3.

Given the initial state $\mathbf{x}_0 = [0.10, -1.00, 0.10]^\top$, final state $\mathbf{x}_f = [-1.02, -1.20, 1.10]^\top$ and the constraints $-\underline{\text{VEL}} = \overline{\text{VEL}} = 4$ rad/s, $-\underline{\text{ACC}} = \overline{\text{ACC}} = 2$ rad/s^2, $-\underline{\text{JERK}} = \overline{\text{JERK}} = 5$ rad/s^3, the curve #1 in Fig. 9 illustrates the original jerk profile without motion duration control. In this case the motion takes 1 s to reach $\mathbf{x}_f$ using the EP #8 described in Table IV. The algorithm disposed by [15] provides a valid solution for motion duration control only from 2.62 s, employing a different EP with a constant velocity step (curve #4). Hence, between 1 s and 2.62 s it is not possible to control the motion duration using ref. [15].

The curves #2 and #3 at Fig. 9 present the jerk profile computed by Algorithm 1 for different Δ_T^*, respectively 1.53 s and 2.62 s. The target states could still be achieved using the original EP however with $\widetilde{\overline{p^n}} < \overline{p^n}$.

VII. Conclusion

This letter describes an approach for trajectory planning considering the kinematic constraints for a single actuator and a required duration time Δ_T^*. The algorithm is basically divided into two parts. In the first part, based on initial and target state, two planners were described considering up to second and third derivative of position. Due to the "bang-off-bang"-like portrait, both methods guarantee that the target state can be reached respecting the kinematic constraints of the actuator. The second part of the algorithm describes a novel approach to adapt the profile considering the actuator's constraints and the required duration time Δ_T^*. It is relevant to remark that the proposed method does not minimize a metric, but uses time-optimal motion planners to discover the minimum trajectory time between initial and target states considering the constraints. Then, based on these results the user knows that any trajectory duration shorter than the time-optimal trajectory is unfeasible.

Since there is no analytical method capable of finding $\widetilde{\overline{p^n}}$ such that $\Delta_T(\widetilde{\overline{p^n}}) = \Delta_T^*$, iterative methods were applied using

a conservative "stop condition". Using function minimizers it is possible to compute new values for the highest nth-order derivative, resulting in $\widetilde{\overline{p^n}} < \overline{p^n}$. The application of this method allows the motion duration control for all EPs described in Tables II and IV, which is an improvement compared to existing methods, such as those proposed by Berscheid et al. [15].

Finally, the developed algorithm was tested on a real application case, where the motion profile of an actuator needs to be synchronized with additional devices considering the sampling time of the controller. Despite $|\Delta_T(\widetilde{\overline{p^n}}) - \Delta_T^*| < \epsilon$, because of the iterative calculation, the new motion profiles displayed in Figs. 6 and 7 respect the kinematic limits. Additionally, the novel approach is compared to other algorithms and the improvements are presented.

As described in Section V, function minimizers as well as root-finding algorithms can be applied to resolve (5). Function minimizers tend to provide valid solutions for ill-conditioned systems, but require more computation time compared with root-finding algorithms. The next step is to investigate further methods to resolve (5), that still provide valid solutions in a short computation time, independently of input and constraint values.

VIII. Declaration of Conflict of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this letter.

References

[1] A. Gasparetto, P. Boscariol, A. Lanzutti, and R. Vidoni, "Path planning and trajectory planning algorithms: A general overview," in *Motion and Operation Planning of Robotic Systems: Background and Practical Approaches*, Berlin, Germany: Springer, 2015, pp. 3–27.

[2] Q.-C. Pham, "Trajectory planning," in *Handbook of Manufacturing Engineering and Technology*, Berlin, Germany: Springer, 2015, pp. 1873–1887.

[3] R. Haschke, E. Weitnauer, and H. Ritter, "On-line planning of time-optimal, jerk-limited trajectories," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2008, pp. 3248–3253, doi: 10.1109/IROS.2008.4650924.

[4] A. Gasparetto, P. Boscariol, A. Lanzutti, and R. Vidoni, "Trajectory planning in robotics," in *Math. Comput. Sci.*, vol. 6, pp. 269–279, 2012.

[5] K. Abidi and J.-X. Xu, "Advanced discrete-time control," in *Studies in Systems, Decision and Control*, vol. 23, Berlin, Germany: Springer, 2015.

[6] S. He, C. Hu, S. Lin, and Y. Zhu, "An online time-optimal trajectory planning method for constrained multi-axis trajectory with guaranteed feasibility," *IEEE Robot. Automat. Lett.*, vol. 7, no. 3, pp. 7375–7382, Jul. 2022.

[7] Y. Liang, C. Yao, W. Wu, L. Wang, and Q. Wang, "Design and implementation of multi-axis real-time synchronous look-ahead trajectory planning algorithm," in *Int. J. Adv, Manuf. Technol.*, vol. 119, pp. 4991–5009, 2022.

[8] A. Locatelli, *Optimal control of a double integrator: A primer on maximum principle*, Cham, Switzerland: Springer, 2017.

[9] S. He, C. Hu, Y. Zhu, and M. Tomizuka, "Time optimal control of triple integrator with input saturation and full state constraints," *Automatica*, vol. 122, 2020, Art. no. 109240.

[10] J. Adamy, *Nichtlineare Systeme Und Regelungen*. Berlin, Germany: Springer, 2018.

[11] Y. Fang, J. Hu, W. Liu, Q. Shao, J. Qi, and Y. Peng, "Smooth and time-optimal S-curve trajectory planning for automated robots and machines," in *Mechanism Mach. Theory*, vol. 137, pp. 127–153, 2019.

[12] T. Kroger, A. Tomiczek, and F. M. Wahl, "Towards on-line trajectory computation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2006, pp. 736–741, doi: 10.1109/IROS.2006.282622.

[13] S. Perumaal and N. Jawahar, "Synchronized trigonometric S-curve trajectory for jerk-bounded time-optimal pick and place operation," *Int. J. Robot. Automat.*, vol. 27, no. 4, 2012, Art. no. 385.

[14] X. Broquere, D. Sidobre, and I. Herrera-Aguilar, "Soft motion trajectory planner for service manipulator robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*,2008, pp. 2808–2813, doi: 10.1109/IROS.2008.4650608.

[15] L. Berscheid and T. Kröger, "Jerk-limited real-time trajectory generation with arbitrary target states," in *Proc. 17th Robot.: Sci. Syst.*, 2021.

[16] K. Hauser and V. Ng-Thow-Hing, "Fast smoothing of manipulator trajectories using optimal bounded-acceleration shortcuts," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2010, pp. 2493–2498.

[17] Q.-C. Pham, "A general, fast, and robust implementation of the time-optimal path parameterization algorithm," *IEEE Trans. Robot.*, vol. 30, no. 6, pp. 1533–1540, Dec. 2014.

[18] J. Kim, W. S. Chang, K. K. Yoon, S. Jeong, B. S. Shin, and K. H. Whang, "Microdrilling of PCB substrate using DPSS third-harmonic laser," in *Proc. 3rd Int. Symp. Laser Precis. Microfabrication*, 2003, pp. 105–109.

[19] J. Nishimae, Y. Satoh, T. Kojima, and T. Fukushima, "Laser processing system for microdrilling of printed circuit boards," in *Proc. 1st Int. Symp. Laser Precis. Microfabrication*, 2000, pp. 209–211.

[20] M. Kraus, D. Walter, A. Michalowski, and J. König, "Processing techniques and system technology for precise and productive microdrilling in metals," in *Ultrashort Pulse Laser Technology: Laser Sources and Applications*, Berlin, Germany: Springer, 2016, pp. 201–230.

[21] T. Barthels, M. Reininghaus, and H. Westergeling, "High-precision ultrashort pulsed laser drilling of micro and nano holes using multibeam processing," in *Proc. SPIE*, vol. 11107, pp. 120–130, 2019.

[22] T. Hirai, K. Fuse, K. Kurisu, and K. Ebata, "Characteristics of diffractive optical element for multispot beam homogenizing," in *Proc. SPIE*, vol. 5525, pp. 148–155, 2004.

[23] F. Zibner, C. Fornaroli, J. Holtkamp, L. Shachaf, N. Kaplan, and A. Gillner, "High-precision laser microcutting and laser microdrilling using diffractive beam-splitting and high-precision flexible beam alignment," in *Proc. SPIE*, vol. 10377, pp. 146–155, 2017.

[24] M. Cui, L. Lu, Z. Zhang, and Y. Guan, "A laser scanner–stage synchronized system supporting the large-area precision polishing of additive-manufactured metallic surfaces," in *Engineering*, vol. 7, no. 12, pp. 1732–1740, 2021.

[25] L. Büsing, S. Eifel, and P. Loosen, "Design, alignment and applications of optical systems for parallel processing with ultra-short laser pulses," in *Proc. SPIE*, vol. 9131, pp. 89–100, 2014.