Journal of Manufacturing Processes 87 (2023) 97–105

Contents lists available at ScienceDirect

Journal of Manufacturing Processes

journal homepage: www.elsevier.com/locate/manpro

Computer-controlled finishing via dynamically constraint position-velocity-time scheduler

Tianyi Wang [a,*], Xiaolong Ke [b], Lei Huang [a], Vipender Negi [c,d], Heejoo Choi [e,f], Wesllin Pullen [e], Daewook Kim [e,f,g], Yi Zhu [a], Mourad Idir [a]

[a] *National Synchrotron Light Source II (NSLS-II), Brookhaven National Laboratory, PO Box 5000, Upton 11973, NY. USA*
[b] *School of Mechanical and Automotive Engineering, Xiamen University of Technology, Xiamen 361024, China*
[c] *Council of Scientific and Industrial Research-Central Scientific Instruments Organization (CSIR-CSIO), Chandigarh 160030, India*
[d] *Academy of Scientific and Innovative Research (AcSIR), Ghaziabad 201002, India*
[e] *James C. Wyant College of Optical Sciences, the University of Arizona, 1630 E. University Blvd., P.O. Box 210094, Tucson, AZ 85721-0094, USA*
[f] *Large Binocular Telescope Observatory, University of Arizona, Tucson, AZ 85721, USA*
[g] *Department of Astronomy and Steward Observatory, University of Arizona, 933 N. Cherry Ave., Tucson, AZ 85721, USA*

ARTICLE INFO

Keywords:
Correction polishing
Position-velocity-time
Feed drive control
Ion beam figuring
Sub-nanometer finishing
Computer-controlled optical surfacing

ABSTRACT

In a Computer Numerical Controlled (CNC) finishing process, the target material removal from an optical surface is guided by the convolution between the influence function of a machine tool and its dwell time at certain points over the surface. To reduce dynamics stressing and increase machining efficiency, the dwell time must be converted to varying velocities, which are the actual inputs to the machine tool controller. Conventionally, the conversion assumed constant acceleration and relied on linear motion interpolation, which caused discontinuities in velocities. This unsmooth motion affects the material removal distribution, and, thus, the accuracy of the finished surface shape. Many modern CNC machines support the smoother, cubic-polynomial interpolated Position-Velocity-Time (PVT) motion mode; however, the conventional scheduler may fail to provide suitable velocities for the PVT. This study answers this challenge by proposing a novel PVT-based velocity scheduler that achieves smooth motion while considering CNC dynamic limits. Firstly, the principle of the PVT is explained, and the PVT-based velocity scheduler is formulated. Secondly, a quadratic programming is used to optimize the velocities by imposing the CNC dynamic constraints and the C^1 continuities (zeroth and first derivatives are continuous) simultaneously. Thirdly, the smoothness and accuracy of the scheduled velocities are studied on different kinds of tool paths via simulation. Finally, a sub-0.3 nm level surface finishing experiment using ion beam figuring is demonstrated to verify the feasibility of the proposed method. The PVT-based scheduler and simulator code is open-sourced.

1. Introduction

In the past decade, the need for precision optical surfaces has drastically increased in both industry and laboratory applications, such as telescopes [1,2], reflective mirrors for synchrotron radiation and free-electron laser facilities [3–7], and EUV lithography optics [8,9]. The surface profile accuracy ranges from microns for visible light applications down to sub-nanometer levels for X-ray imaging at the diffraction limit. In the manufacture of such optical components, Computer Numerical Controlled (CNC) finishing is required in the final stages of fabrication to achieve ultra-precision level of accuracy. Compliant machine tools [10], such as bonnet [11], fluid jet [12,13], magnetorheological fluid [14] and ion beam [15], are often required.

at these stages to correct the final residual errors without damaging the substrate.

In a CNC finishing process, the target material removal from an optical surface is modeled and guided by the convolution between the Tool Influence Function (TIF) of a machine tool and its dwell time varying along a tool path over the surface [16]. The level of accuracy is mainly improved in three ways: one is to optimize the dwell time to minimize residual errors within dynamic constraints. In addition, the tool paths should be properly designed to reduce dynamic errors and repetitive tool marks. Finally, even with an ideal dwell time and tool path combination, its accurate and smooth implementation in a CNC machine is

[*] Corresponding author.
E-mail address: tianyi@bnl.gov (T. Wang).

https://doi.org/10.1016/j.jmapro.2023.01.005
Received 26 September 2022; Received in revised form 24 December 2022; Accepted 1 January 2023
Available online 17 January 2023
1526-6125/Published by Elsevier Ltd on behalf of The Society of Manufacturing Engineers.

crucial to guarantee the planned material removal distribution and thus the accuracy of the finished surface.

As for dwell time optimization, the mainstream methods include the iterative method [17], Fourier transform method [18] and matrix-based method [19]. Wang et al. improved the iterative method by adjusting the initial dwell time with the volumetric removal rate of a TIF and introducing a relaxation factor to accelerate convergence [20]. Wang et al. automated the inverse filtering in the Fourier transform method and proposed a method for simultaneously minimizing the residual errors and total dwell time [5]. This method was further refined by advanced surface extensions to achieve ultra-precision level of accuracy [6]. Wu et al. reformulated the matrix-based method. Path and surface error weighting factors were introduced to the Tiknohov regularization to help search for an appropriate damping factor required by the Least-Squares with QR factorization (LSQR) solver [21]. But this method requires multiple trial-and-error exercises in determining the damping factor and is computationally expensive. Dong et al. further related the ranges of damping factors to TIF sizes [22] and the computational efficiency was improved by a stitching algorithm [23]. Adding practical constraints on residual error [24,25], dwell time local gradient [24,26] and CNC dynamic limits [25–28] were also attempted. However, these methods are often hard to apply in practice due to the computationally expensive constrained solvers. Recently, Wang et al. combined the iterative and matrix-based methods that universally smoothed the dwell time solution, minimized the residual error and reduced the computational complexity. Kang et al. and Ke et al. verified the feasibility and advantages of optimizing dwell time for multiple tools at the same time. Ke et al. and Kim et al. even proposed [29] and improved [30] the multiplexing algorithm to coordinate different tools in a CNC machine. All these methods have pushed dwell time optimization to a new era of high accuracy and robustness. However, they only provided dwell time for discrete dwell points. The order and pattern of how these points are visited should be determined by tool path optimization.

As for tool path optimization, raster paths are the simplest tool paths, in which the machine tool scans the surface along a single axis. Nonetheless, repetitive patterns (i.e. the so-called periodic Middle Spatial Frequency (MSF) errors) will be left on the surface if the tool spacing is too big or the processing time is extended. While Wan et al. claimed that these defects could be reduced by post-processing the surface with sparse bi-step raster paths [31], this challenge was mainly resolved by not generating those patterns via tool path randomization. Dunn and Walker proposed pseudorandom tool paths [32] and Wang et al. introduced unicursal maze paths [33] to restrain the MSF errors. Negi et al. adapted random paths to different surface shapes by introducing another degree of angle randomness and incorporating local material removal distribution [34]. However, all these paths require sharp cornering, which adds dynamic stressing to the machine tool. To solve this problem, Beaucamp et al. and Wang et al. introduced circular-random paths [35] and tree-shaped random paths [36], respectively, which avoided the generation of any corners in the path. By applying these methods, the periodic MSF errors were suppressed, and the surface roughness was improved. With the optimized dwell time along a well-defined tool path, the next challenge is to accurately implement the dwell time in the machine tool controller.

As for dwell time implementation, the most naïve way was through position mode [37], in which the machine tool was moved to the next dwell point with infinite acceleration and dwell there for the calculated time. Position mode is simple to implement, but it incurs extra processing time and deteriorates surface accuracy because of unsmooth motions. To answer these challenges, dwell time was converted to velocities and the controller was actually driven by varying velocities at different locations along tool paths. Zhou et al. proposed a conversion method adopting the trapezoidal model [38], in which the velocities in each machining interval are composed of an accelerating/ decelerating part with constant acceleration followed by a constant velocity part. By applying this method, they saved processing time and achieved higher convergence ratio compared to the position mode. However, the CNC dynamic constraints, such as the maximum acceleration and speed, were not well considered. Unsmooth motion was found in machining intervals due to the discontinuities in velocity. Also, the motion may overshoot the planned positions (see Fig. 2) because of the fixed acceleration. Mizouea et al. identified the CNC dynamics of their machine from experiments and adjusted the original velocities accordingly using a particle swarm optimization before inputting them to the machine controller. Although the impact of controller dynamics in micro-groove polishing was reduced [39] with this method, the motion smoothness still relied on the original velocities. Han et al. introduced the Gaussian Mixture Model (GMM) to model experimental TIFs and thus obtained an analytical convolution model [40]. From this model, velocities could be directly optimized within dynamic constraints through adaptive machining intervals for raster paths. In this way, they improved the stability of their fluid jet polishing machine without compromising with accuracy. However, the GMM may fail to sufficiently model the details of complex machine tools, such as bonnet tools, and this method is hard to generalize to more advanced tool paths as reviewed above.

In fact, modern CNC controllers (e.g. Galil DMC-40×0 series, Googol Technology GTS-400 controllers, etc.) support more advanced motion modes, such as the Position-Velocity-Time (PVT) mode. This mode allows arbitrary motion profiles to be defined by position, velocity and time individually for each axis. The controller will interpolate the motion profile between each two consecutive positions using a third-order polynomial. Therefore, the velocities and accelerations in the PVT description are guaranteed to be piece-wise quadratic and linear, respectively, which leads to smoother motion and avoids overshooting. Lu et al. successfully applied the PVT to improve the precision and efficiency of sculpture surface machining [41]. However, in our finishing scenario, it is not straightforward to implement dwell time in the PVT mode, since the required velocity inputs to the PVT are unknown. Our preliminary research [15] verified the possibility of feeding the PVT with the velocities calculated using the constant-acceleration model [38], but those trapezoidal velocities did not fit the PVT description. Especially when the CNC dynamic constraints are reached, the motion may be distorted, resulting in excessive position errors, and thus unexpected material removal distribution.

The PVT is potentially a preferable motion programming mode if velocities can be reasonably obtained from dwell time under its own framework. In view of this gap, we propose a PVT-based velocity scheduler that achieves reliable and smooth motion control within machine dynamic constraints. First, two acceleration equations are introduced to the original PVT model in each machining interval. This enables to simultaneously compute the velocities and polynomial coefficients for the PVT control. Coupled with a quadratic programming formulation, CNC dynamic constraints are naturally included in the velocity optimization. More importantly, the level of continuities in velocity can be explicitly controlled in this formulation. For example, the C^1 continuities can be imposed by adding equality constraints on the derivatives of the intermediate velocities. Simulations of applying the PVT-based scheduler to the raster path, maze path [33] and Random Adaptive Path (RAP) [34] are performed to verify the validity of this scheduler. Finally, a sub-0.3 nm level surface finishing experiment using Ion Beam Figuring (IBF) is demonstrated to show that the velocities optimized from the proposed method can be perfectly followed by a Galil PVT controller. The code for the PVT-based velocity scheduler, sampler and simulator are open sourced at [42].

2. Velocity scheduling from dwell time

2.1. Planning computer-controlled finishing processes

Fig. 1 schematically describes the planning phase of any computer-controlled finishing processes. Initially, tool paths are defined based on the target removal map in a Clear Aperture (CA). A tool path, as

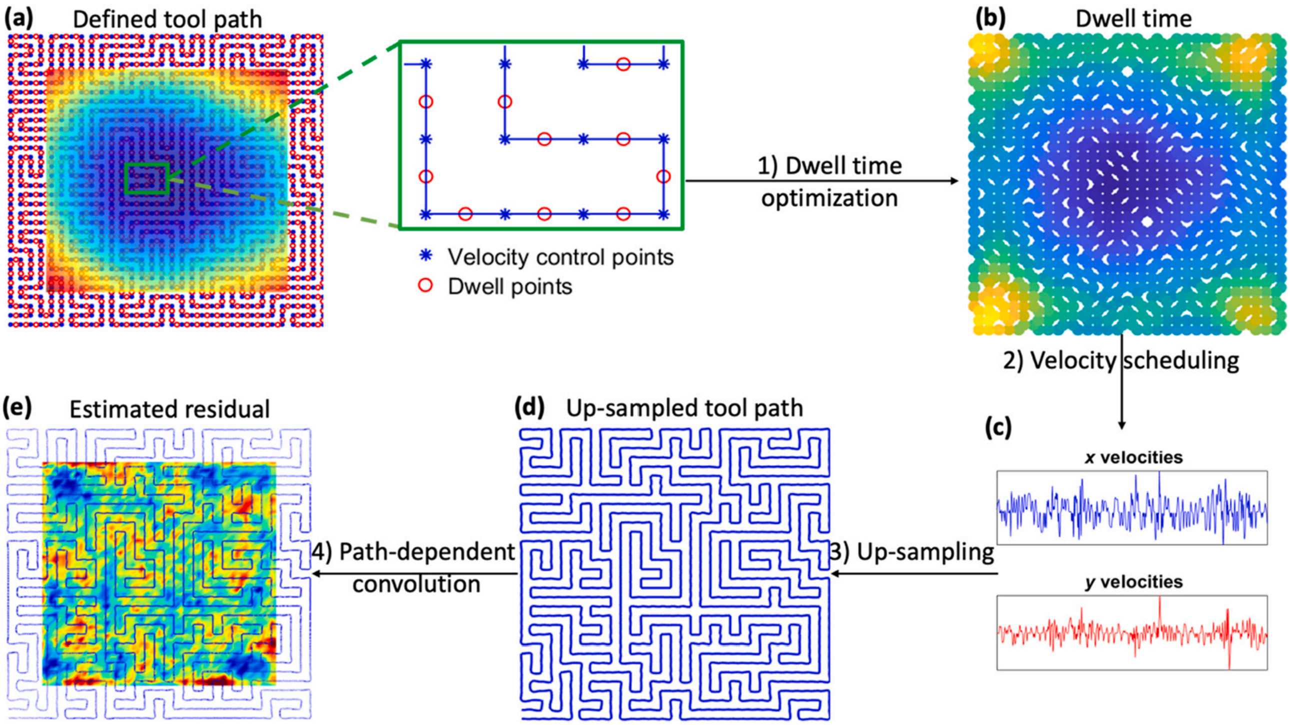

Fig. 1. Planning a finishing process: a tool path containing velocity control points and dwell points is defined for the clear aperture (a). Dwell time is calculated for the dwell points along the path (b), which is then converted to the x and y components of the velocities at the velocity control points(c). The actual tool path is calculated from up-sampling the velocities (d), from which the residual errors are estimated through the path-dependent convolution (e).

shown in Fig. 1(a), is composed of dwell points and velocity control points. Dwell points reside at the center of their two adjacent velocity control points. Ideally, it requires that the time of moving from one velocity control point to another is equal to the dwell time at the in-between dwell point.

Afterwards, dwell time is calculated for those dwell points through dwell time optimization in Fig. 1(b), which are then converted to velocities at the velocity control points in Fig. 1(c). To verify the fitness of the velocities, as shown in Fig. 1(d), they are up-sampled, from which a much denser tool path is generated [39]. This is the actual tool path that

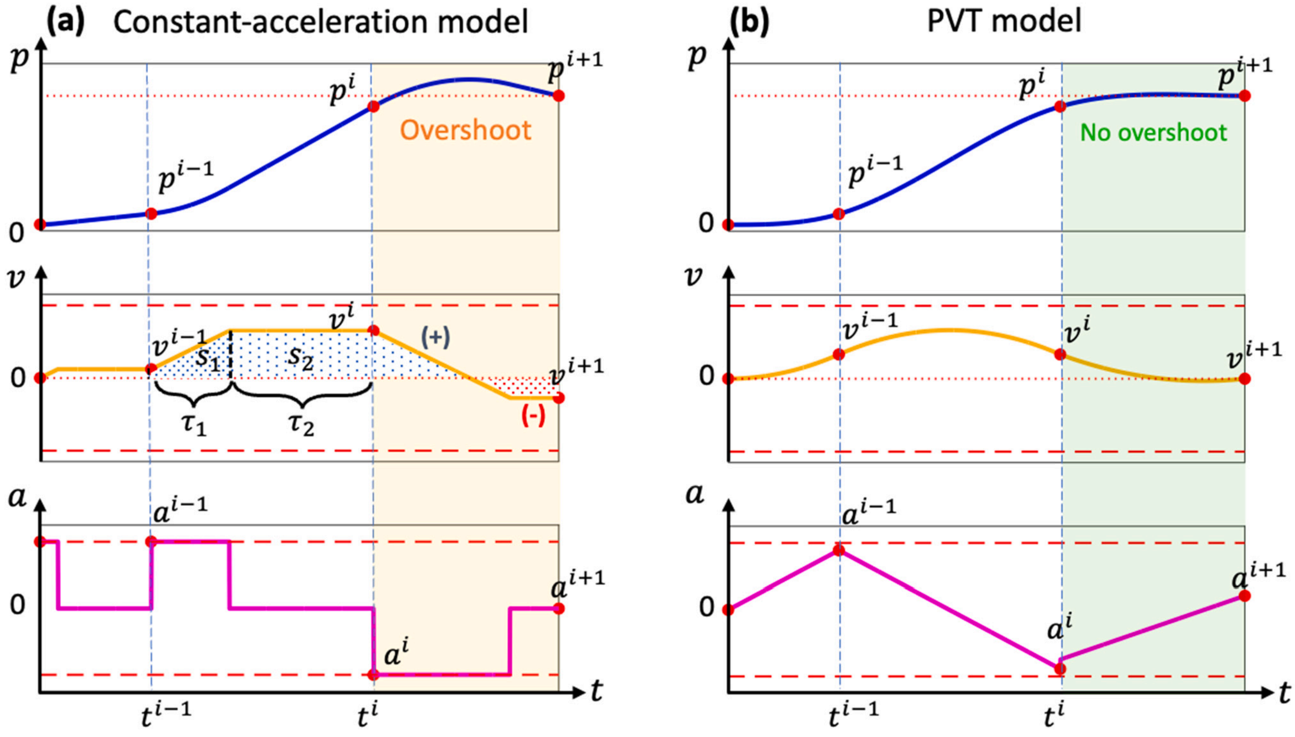

Fig. 2. The constant-acceleration-based velocity scheduler (a) may overshoot the specified positions because the acceleration is fixed. The PVT-based velocity scheduler (b) can avoid this problem by linearly adjusting the acceleration.

is used to estimate the residual errors via the path-dependent convolution in Fig. 1(e). It can be observed from Fig. 1, that even with ideal dwell time, reasonable velocity scheduling is crucial for the ultra-precision accuracy and fast convergence to target shapes.

2.2. Notations and constraints in velocity scheduling

Positions, velocities and accelerations constitute a feed drive system. Fig. 2 describes the motions among four velocity control points, where p_i, v_i and a_i ($i = 1, 2, 3, \ldots, N$, where N is the number of dwell points) represent the position, velocity and acceleration for the arbitrary χ-axis (where $\chi \in \{x, y, z\}$) at the ith velocity control point, respectively, while $t_i \geq 0$ denotes the moment when the tool arrived at that point.

With $v_0 = 0$, $a_0 = 0$ and $t_0 = 0$, the dwell time, $d_i > 0$, for the dwell point between p_{i-1} and p_i shown in Fig. 1, is calculated as

$$d_i = t_i - t_{i-1}. \tag{1}$$

In other words, the time a tool travelling from p_{i-1} to p_i is d_i. For a specific stage, its motion is constrained by its maximum velocity and acceleration as

$$\begin{matrix} |v_i| \leq v_{\max} \\ |a_i| \leq a_{\max} \end{matrix}, \tag{2}$$

where $v_{\max}$ and $a_{\max}$ represent the maximum velocity and acceleration in the χ-axis. The key in velocity scheduling is thus to covert d_i to v_i at p_i within the $v_{\max}$ and $a_{\max}$ constraints. These notations are used in the rest of this study.

2.3. Velocity scheduling with constant acceleration

Most simply, v_i can be calculated as $v_i = (p_i - p_{i-1})/d_i$, i.e. the average velocity for the $[p_{i-1}, p_i]$ segment. Nevertheless, the actual dwell time is not equal to d_i, since any feed drive system must have a course of acceleration or deceleration. Zhou et al. proposed a constant-acceleration model [38] to solve this problem. As shown in Fig. 2(a), the segment between p_{i-1} and p_i in this model is composed of an acceleration part with a fixed acceleration, a, and a constant-velocity part. The motion in this segment is summarized in the following five equations as,

$$\begin{cases} s_1 + s_2 = s_i \\ \tau_1 + \tau_2 = d_i \\ \left(v_i^2 - v_{i-1}^2\right)/s_1 = 2a \\ (v_i - v_{i-1})/\tau_1 = a \\ s_2 = v_i \tau_2 \end{cases}, \tag{3}$$

where $s_i = p_i - p_{i-1}$, τ_1 and s_1 are the time and displacement for the acceleration part, respectively; and $\tau_2$and s_2 are the time and displacement for the constant-velocity part, respectively. From Eq. (3),

$$v_i^2 - 2(v_{i-1} + ad_i)v_i + \left(v_{i-1}^2 + 2as_i\right), \tag{4}$$

where v_i is the only unknown and is solved as

$$v_{i+} = (v_{i-1} + ad_i) - \sqrt{(v_{i-1} + ad_i)^2 - \left(v_{i-1}^2 + 2as_i\right)}, \tag{5}$$

$$v_{i-} = (v_{i-1} + ad_i) + \sqrt{(v_{i-1} + ad_i)^2 - \left(v_{i-1}^2 + 2as_i\right)}, \tag{6}$$

if $(v_{i-1} + ad_i)^2 \geq (v_{i-1}^2 + 2as_i)$. When the tool accelerates, i.e. $s_i/d_i > v_{i-1}$ and $a > 0$, the current velocity $v_i = v_{i+}$; when the tool decelerates, i.e. $s_i/d_i < v_{i-1}$ and $a < 0$, the current velocity $v_i = v_{i-}$. If $(v_{i-1} + ad_i)^2 < (v_{i-1}^2 + 2as_i)$, which means that a is so small that s_i cannot be achieved within d_i, v_i is calculated as the maximum achievable velocity on s_i as

$$v_i = v_{i\times} = \sqrt{v_{i-1}^2 + 2as_i} \tag{7}$$

and the actual dwell time is $d_{ia} = (v_{i\times} - v_{i-1})/a$ so that $d_{ia} \neq d_i$.

In the real implementation of the constant-acceleration model, $a = a_{\max}$ is always used to maximally avoid $v_{i\times}$. Also, $v_i = v_{\max}$ is enforced when the maximum velocity of the stage is exceeded. Although these extreme situations seldom happen in finishing of surfaces, another two problems that greatly affect the accuracy of the constant-acceleration model are reflected in Fig. 2(a). First, the velocities are discontinuous. This unsmooth motion adds dynamic stressing to CNC controllers. More seriously, due to the fixed a, overshoot may happen when a is too large for a small s_i. This back-and-forth motion will remove extra material and thus seriously deteriorate the local surface profile. Even with a mathematically valid a, the actual CNC controller has its own feed drive system that does not implement the mechanism described in Eq. (3).

3. PVT-based velocity scheduler

3.1. Principle of PVT motion control

Based on the above analysis, an appropriate velocity scheduler should not only ensure the motion smoothness, but also consider the actual control mode of a specific CNC controller. PVT is a motion mode that has been supported by many modern CNC controllers. As shown in Fig. 2(b), it describes the positions between each two consecutive velocity control points as a third order polynomial. The velocities and accelerations are thus quadratic and linear on that segment, respectively, which is smoother and more flexible than the constant-acceleration model. The motion between p_{i-1} and p_i in PVT is controlled as

$$\begin{cases} c_{i,1}t_{i-1}^3 + c_{i,2}t_{i-1}^2 + c_{i,3}t_{i-1} + c_{i,4} = p_{i-1} \\ c_{i,1}t_i^3 + c_{i,2}t_i^2 + c_{i,3}t_i + c_{i,4} = p_i \\ 3c_{i,1}t_{i-1}^2 + 2c_{i,2}t_{i-1} + c_{i,3} = v_{i-1} \\ 3c_{i,1}t_i^2 + 2c_{i,2}t_i + c_{i,3} = v_i \end{cases} \tag{8}$$

where $\mathbf{c}_i = (c_{i,1}, c_{i,2}, c_{i,3}, c_{i,4})^T$ are the polynomial coefficients for the $[p_{i-1}, p_i]$ segment. Known the positions, velocities and times at the $(i-1)$th and ith points, $\mathbf{c}_i$ can be uniquely solved from Eq. (8). Fig. 2(b) shows the positions, velocities and accelerations generated with the PVT mode using the same position, velocity and time parameters at the velocity control points shown in Fig. 2(a). It is obvious that the overshoot problem in the constant-acceleration model is well mitigated in the PVT model thanks to the linearly variable accelerations.

As described in Eq. (8), the velocities are the inputs to the PVT model. However, they are the unknowns that should be calculated in our velocity scheduling problem. Although the velocities calculated with constant acceleration can be simply fed into the PVT model, they fail to fit the PVT description and the expected motion may thus be distorted.

Assuming that v_{i-1} is known when calculating v_i, Eq. (8) still contains five unknowns, i.e. $c_{i,1}$, $c_{i,2}$, $c_{i,3}$, $c_{i,4}$ and v_i, but four equations. At least one more equation is required to uniquely solve for the velocities. Moreover, Eq. (8) lacks the control of accelerations, making it impossible to fully consider the CNC dynamic limits. In the rest of this section, the new PVT formulation that is able to solve for the velocities is introduced, followed by the explanation of the necessary strategies to consider CNC dynamic limits and ensure the smoothness of the calculated velocities.

3.2. Velocity scheduling with PVT

To uniquely solve for the velocities and obtain the control of accelerations, we introduce two additional equations to Eq. (8) which describe the accelerations at p_{i-1} and p_i as

$$\left\{\begin{array}{c} c_{i,1}t_{i-1}^3 + c_{i,2}t_{i-1}^2 + c_{i,3}t_{i-1} + c_{i,4} = p_{i-1} \\ c_{i,1}t_i^3 + c_{i,2}t_i^2 + c_{i,3}t_i + c_{i,4} = p_i \\ 3c_{i,1}t_{i-1}^2 + 2c_{i,2}t_{i-1} + c_{i,3} = v_{i-1} \\ 3c_{i,1}t_i^2 + 2c_{i,2}t_i + c_{i,3} = v_i \\ 6c_{i,1}t_{i-1} + 2c_{i,2} = a_{i-1} \\ 6c_{i,1}t_i + 2c_{i,2} = a_i \end{array}\right. . \tag{9}$$

With v_{i-1} and a_{i-1} known, Eq. (9) contains six equations with six unknowns, which are $c_{i,1}$, $c_{i,2}$, $c_{i,3}$, $c_{i,4}$, v_i and a_i, therefore, it can be uniquely solved.

The CNC dynamic limits given in Eq. (2) can be naturally added as the constraints to Eq. (9). To further ensure the smoothness of the motion, the C^1 continuity constraints on the velocities are imposed by enforcing the equality of the derivatives of v_i as

$$\frac{dv_i^-}{dt} = \frac{dv_i^+}{dt}, \tag{10}$$

where

$$v_i^- = 3c_{i,1}t_i^2 + 2c_{i,2}t_i + c_{i,3} \tag{11}$$

and

$$v_i^+ = 3c_{i+1,1}t_i^2 + 2c_{i+1,2}t_i + c_{i+1,3} \tag{12}$$

are theare the velocities at the ith velocity control point calculated from the [p_{i-1},p_i] and [p_i,p_{i+1}] segments, respectively. Substituting Eqs. (11) and (12) to Eq. (10), the final smoothness constraint is obtained as

$$6t_i(c_{i,1} - c_{i+1,1}) - 2(c_{i+1,2} - c_{i,2}) = 0. \tag{13}$$

By combining Eqs. (2), (9) and (13) and rewriting Eq. (9) in matrix form, the PVT scheduler for the ith velocity control point is formulated as

$$\textbf{solve} \quad \underbrace{\begin{bmatrix} t_{i-1}^3 & t_{i-1}^2 & t_{i-1} & 1 & 0 & 0 \\ t_i^2 & t_i^2 & t_i & 1 & 0 & 0 \\ 3t_{i-1}^2 & 2t_{i-1} & 1 & 0 & 0 & 0 \\ 3t_i^2 & 2t_i & 1 & 0 & -1 & 0 \\ 6t_{i-1} & 2 & 0 & 0 & 0 & 0 \\ 6t_i & 2 & 0 & 0 & 0 & -1 \end{bmatrix}}_{\mathbf{C}_i} \underbrace{\begin{bmatrix} c_{i,1} \\ c_{i,2} \\ c_{i,3} \\ c_{i,4} \\ v_i \\ a_i \end{bmatrix}}_{\mathbf{x}_i} = \begin{bmatrix} p_{i-1} \\ p_i \\ v_{i-1} \\ 0 \\ a_{i-1} \\ 0 \end{bmatrix}, \tag{14}$$

$$\textbf{subject to} \left\{\begin{array}{c} |v_i| \le v_{max} \\ |a_i| \le a_{max} \\ 6t_i(c_{i,1} - c_{i+1,1}) - 2(c_{i+1,2} - c_{i,2}) = 0. \end{array}\right.$$

For all the PVT segments, there will be $6N$ equations and $6N$ unknowns in total. Eq. (14) can be combined to solve for all the N velocities simultaneously. In addition, by imposing the initial and the ending velocities and accelerations as

$$v_0 = 0, a_0 = 0, v_N = 0, \text{and } a_N = 0, \tag{15}$$

respectively, the stage is guaranteed to be properly started and stopped. Combining Eqs. (14) and (15) for all the N segments, the PVT scheduler can be thought of containing $6N + 4$ equations with $6N$ unknowns, which becomes a constrained least-squares problem that can be described as

$$\textbf{solve} \quad \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & \dots & 0 \\ \mathbf{C}_1 & & & & & & & \\ \mathbf{E} & \mathbf{C}_2 & & & & & & \\ & \ddots & \ddots & & & & & \\ & & \ddots & \ddots & & & & \\ & & & \mathbf{E} & \mathbf{C}_i & & & \\ & & & & \ddots & \ddots & & \\ & & & & & \ddots & \ddots & \\ & & & & & & \mathbf{E} & \mathbf{C}_N \\ 0 & \dots & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & \dots & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{\mathbf{C}} \underbrace{\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \vdots \\ \mathbf{x}_i \\ \vdots \\ \vdots \\ \mathbf{x}_N \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} v_0 \\ a_0 \\ \mathbf{d}_1 \\ \mathbf{d}_2 \\ \vdots \\ \vdots \\ \mathbf{d}_i \\ \vdots \\ \vdots \\ \mathbf{d}_N \\ v_N \\ a_N \end{bmatrix}}_{\mathbf{d}}, \tag{16}$$

$$\textbf{subject to} \left\{\begin{array}{l} |\mathrm{x}_{i,5}| \le v_{max} \\ |x_{i,6}| \le a_{max} \\ \mathbf{A}_{\text{eq}}\mathbf{x} = \mathbf{b}_{\text{eq}} \end{array}\right.$$

in which

$$\mathbf{E} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \mathbf{d}_i = \begin{bmatrix} p_{i-1} \\ p_i \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

$x_{i,\ j}$ denotes the jth element of the x_i vector, and $\mathbf{A}_{\text{eq}}\mathbf{x} = \mathbf{b}_{\text{eq}}$ are the equality constraints that enforce the C^1 continuities in velocity for the intermediate velocity counterpoints from p_1 to p_{N-1}. Therefore, based on Eq. (13), $\mathbf{A}_{\text{eq}}$ is a $(N-1) \times 6N$ matrix with the (m,n) element defined as

$$\mathbf{A}_{\text{eq}}(m,n) = \left\{\begin{array}{ll} 6t_m, & n = 6m-5 \\ -6t_m, & n = 6m+1 \\ 2, & n = 6m-4 \\ -2, & n = 6m+2 \\ 0, & \text{otherwise} \end{array}\right. ,$$

for $m = 1, 2, \ldots, N$, $n = 1, 2, \ldots, 6N$, while $\mathbf{b}_{\text{eq}} = \mathbf{0}^T$ is a zero vector. Eq. (16) can then be simply solved as a quadratic programming problem,

$$\begin{array}{l} \textit{minimize } \frac{1}{2}\mathbf{x}^T\mathbf{P}\mathbf{x} + \mathbf{q}^T\mathbf{x} \\ \textbf{subject to} \left\{\begin{array}{l} |x_{i,5}| \le v_{max} \\ |x_{i,6}| \le a_{max} \\ \mathbf{A}_{\text{eq}}\mathbf{x} = \mathbf{b}_{\text{eq}} \end{array}\right. , \end{array} \tag{17}$$

where $\mathbf{P} = \boldsymbol{C}^T\boldsymbol{C}$ and $\mathbf{q} = -\boldsymbol{C}^T\boldsymbol{d}$.

It is worth mentioning that special care should be taken in configuring the bounding constraints for the velocities, i.e. $|x_{i,\ 5}| \le v_{\max}$ in Eq. (17). We found that the calculated velocities might have different signs than the movement direction. Defining the positive direction of any χ-axis as the direction in which p_i increases, $|x_{i,\ 5}| \le v_{\max}$ should be rewritten as

$$\left\{\begin{array}{ll} 0 < x_{i,5} \le v_{\max}, & p_{i+1} \ge p_i \\ -v_{\max} \le x_{i,5} < 0, & p_{i+1} < p_i \end{array}\right. . \tag{18}$$

With this modification, the inappropriate velocities are avoided. Also, the calculated velocities v_i can be substituted back to Eq. (8) to refine the polynomial coefficients c_i.

4. Simulation

The smoothness and accuracy of the PVT-scheduled velocities are studied via simulation. As shown in Fig. 3(a) and (b), a flat Silicon mirror with the size of 92 mm × 16 mm is measured with our stitching interferometry system [43–45]. The initial height error is 6.32 nm Root Mean Square (RMS). The raster path, maze path and RAP shown in Fig. 3(c), (d) and (e), respectively, are used to test the PVT-based velocity scheduler. The tool spacing in each path is set as 0.5 mm. The TIF used in the simulation given in Fig. 4 is extracted our IBF system [15]. The dwell time is optimized with our universal Dwell Time Optimization (UDO) algorithm [7] and the residual error after processing with each tool path is estimated by following the procedure shown in Fig. 1. The dynamic limits of the x and y stages in the IBF system are $v_{\max}^{x} = 250$ mm/s, $a_{\max}^{x} = 2000$ mm/s^2, $v_{\max}^{y} = 150$ mm/s, and $a_{\max}^{y} = 1000$ mm/s^2, respectively.

In the rest of this section, the smoothness of the PVT-scheduled velocities is first studied by the comparison between the PVT-based and constant-acceleration-based schedulers. The accuracy of the PVT-based scheduler is examined by applying it to different tool paths and checking the estimated residual errors.

4.1. PVT-based scheduler vs. constant-acceleration-based scheduler

To examine the smoothness of the PVT-scheduled velocities, the constant-acceleration-based scheduler and PVT-based scheduler are applied to compute the velocities for the maze path shown in Fig. 3(d). In the constant-acceleration-based scheduler, the maximum accelerations, i.e. $a_{\max}^{x}$ and $a_{\max}^{y}$, are used.

Based on the same dwell time (which is 8.17 min in total), Fig. 5(a) and (b) demonstrated the tool paths up-sampled with the constant-acceleration-based and PVT-based schedulers, respectively. It can be found that all the velocity control points in both cases are properly visited. However, as shown in Fig. 5(c) and (d), when we further examine the velocities of the highlighted tool path segments, it is obvious that the PVT-scheduled velocities are much smoother than those scheduled with the constant accelerations. Therefore, with a PVT-enabled machine controller, the PVT-based scheduler will exert less dynamic stressing on the feed drive system.

4.2. PVT-based scheduler applied to different tool paths

Fig. 6(a), (b) and (c) demonstrate the corresponding dwell time maps with the up-sampled tool paths obtained through the PVT-based scheduler. The accuracy of the PVT-scheduled velocities is studied by applying the PVT-based scheduler to the raster path, maze path and RAP, comparing the respective velocities along considered paths in x and y directions as shown in Fig. 3(d) and respective residual error topography in Fig. 3(e) to (f).

It can be observed that, the total dwell time optimized for the three kinds of tool paths (which is 8.15 min, 8.17 min and 7.64 min, respectively) are similar to each other and every velocity control point is properly visited, which verifies the stability of the UDO algorithm and the accuracy of the proposed PVT-based scheduler. However, as we closely study the scheduled velocities in the x and y directions for the three tool paths in Fig. 6(d), we found that the RAP has the highest variations in velocities, the maze path comes to the second, and the raster path is the most uniform, which indicates that the sharp cornering in the RAP and maze path requires frequent accelerations and decelerations, and the smaller the cornering angles are, the more dynamic stressing will be added to the feed drive system. This is further manifested by examining the residual errors estimated in Fig. 6(e), (f) and (g) for the three tool paths. With the raster path, as shown in Fig. 6(e), only the periodic tool marks along the y direction can be seen. For the maze.

path and RAP shown in Fig. 6(f) and (g), respectively, even though the repeated tool marks are absent, it is clear that the errors concentrate on the corners in the tool paths instead. Again, the smaller the cornering angles are, the larger the final residual errors will result. Therefore, the trade-off between the complexity of a tool path and the expected level of residual errors should be carefully evaluated in practice.

5. Experiment

To confirm the feasibility of the proposed PVT-based scheduler in real applications, it was applied to finish a flat multi-layer silicon substrate (see Fig. 3(a)) using our IBF system. This mirror was requested by one of the beamlines at the National Synchrotron Light Source II with the specification on the residual error of ≤0.3 nm RMS. To achieve such a high level of accuracy, as described in Fig. 7(a), we chose to use the PVT-based scheduler with the raster path, for it provides the best

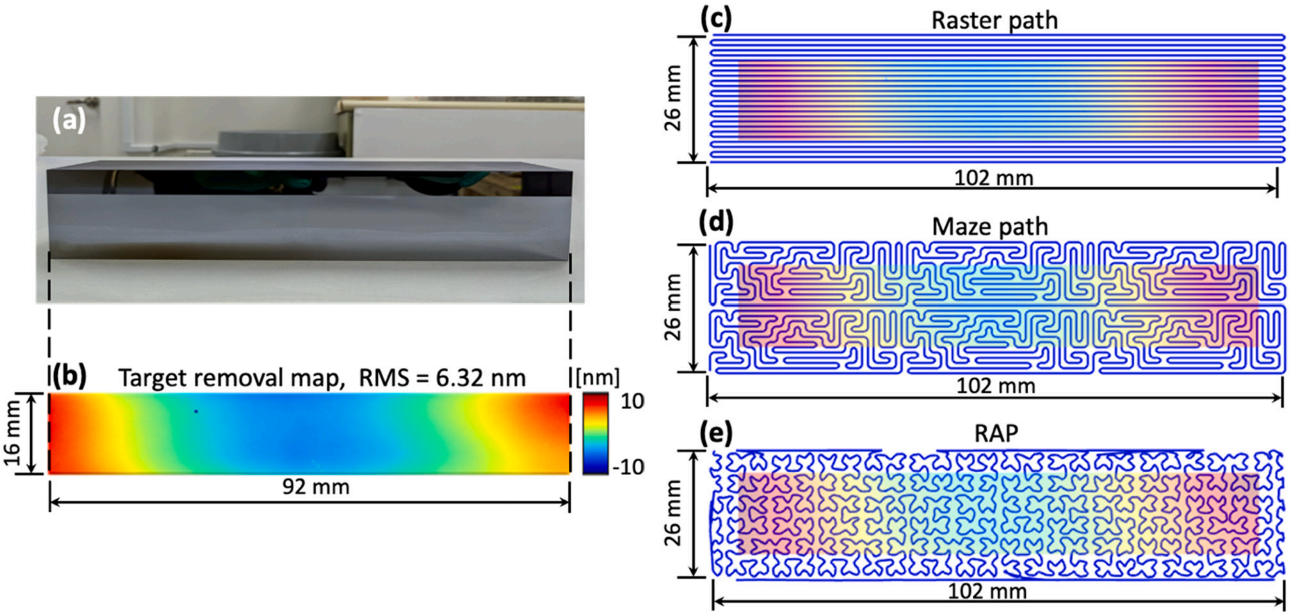

Fig. 3. The simulation is performed using (a) a rectangular flat mirror with the size of 92 mm × 16 mm. (b) The target removal map is measured to be 6.32 nm RMS. The (c) raster path, (d) maze path, and (e) RAP are used to test the proposed PVT-based velocity scheduler.

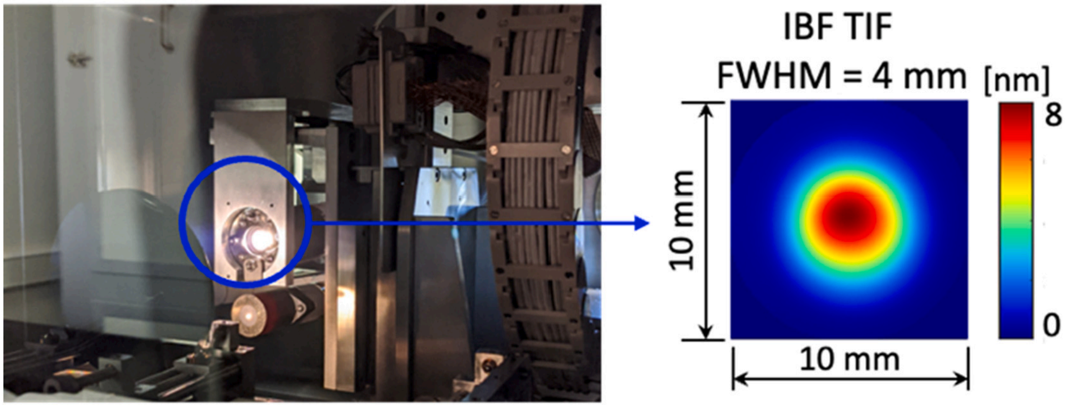

Fig. 4. The IBF machine tool is extracted as a Gaussian-shape TIF.

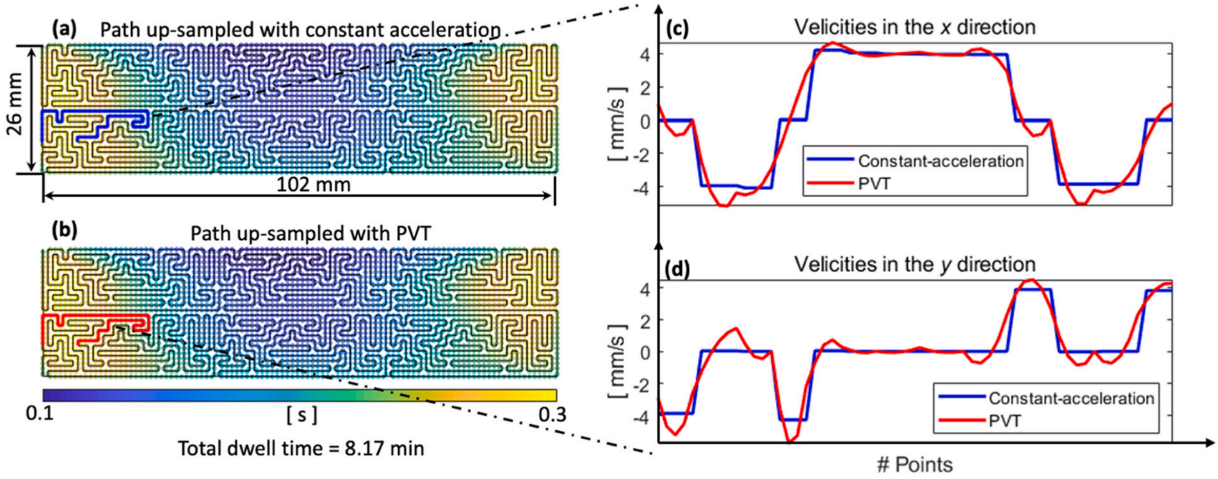

Fig. 5. Up-sampled tool paths calculated from the (a) constant-acceleration-based schedule and (b) PVT-based scheduler. The PVT-scheduled velocities in both the (c) x and (d) y directions are smoother than those obtained with the constant accelerations.

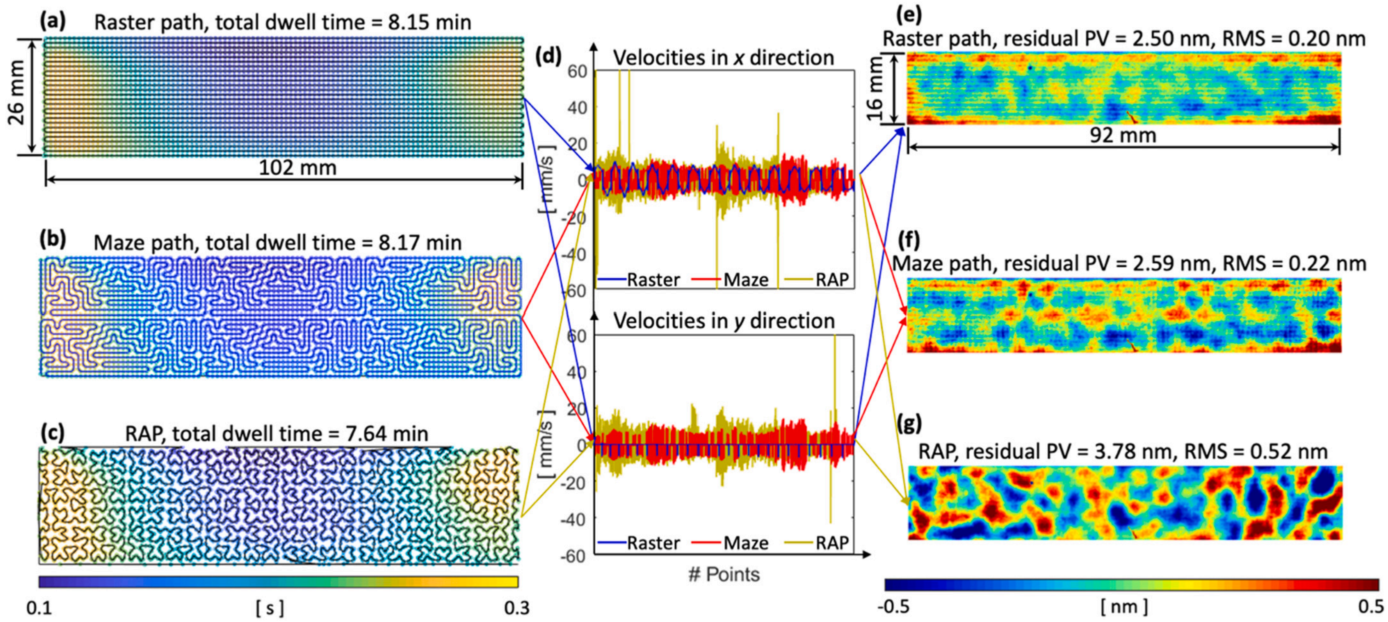

Fig. 6. Up-sampled tool paths with the PVT-based scheduler for the (a) raster path (Visualization-1), (b) maze path (Visualization-2) and (c) RAP (Visualization-3). From (d) the velocity profiles in the x and y directions, it is found that sharper cornering results in larger variations in velocities, and thus affects the local material removal distributions as shown in the residual maps in (e), (f), and (g).

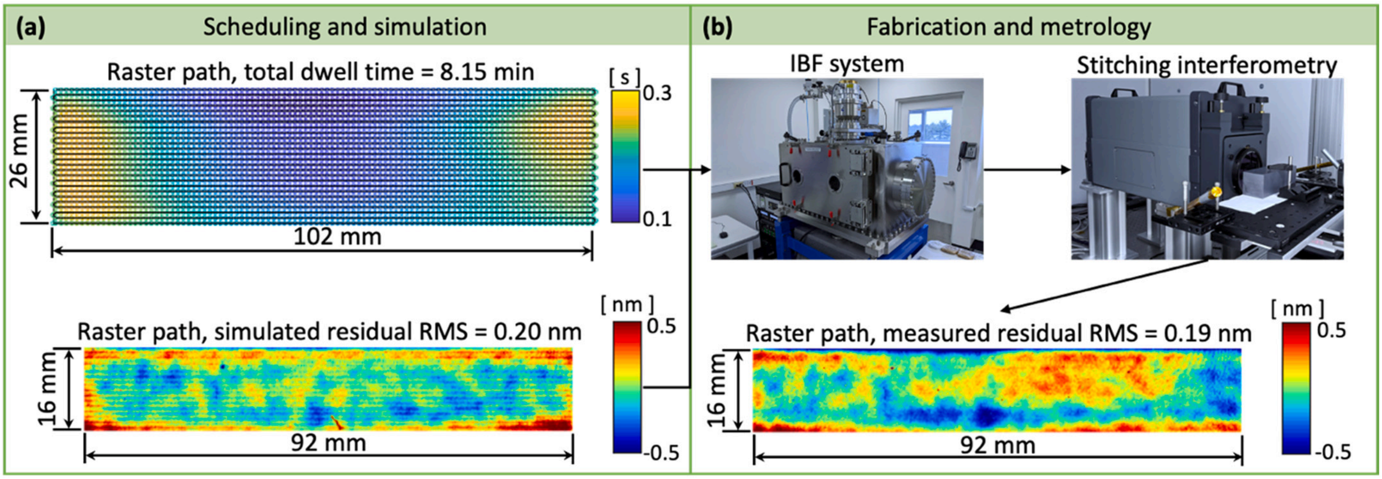
(a) Scheduling and simulation
Raster path, total dwell time = 8.15 min
[s]
0.3
0.1
26 mm
102 mm
Raster path, simulated residual RMS = 0.20 nm
[nm]
0.5
-0.5
16 mm
92 mm
(b) Fabrication and metrology
IBF system
Stitching interferometry
Raster path, measured residual RMS = 0.19 nm
[nm]
0.5
-0.5
16 mm
92 mm

Fig. 7. (a) Raster tool path based on PVT-scheduled velocities (Visualization-1) and initial residual error to be corrected is sent to (b) the fabrication and meteorology machine tools for IBF. The measured residual height error is 0.19 nm RMS, which coincides with the simulation.

residual error estimation and adds the smallest dynamic stressing to the IBF machine.

The IBF system [15] used in the experiment, as shown in Fig. 7(b), is equipped with a KDC10 ion source from Kaufman & Robinson. The operational parameters are beam voltage, $U_b = 600$ V; beam current, $I_b = 10$ mA; accelerator voltage, $U_a = -90$ V; and accelerator current, $I_a = 2$ mA. The CNC unit in the IBF system is composed of three translation stages (one NLS8–500 and two NLE-50) from Newmark and an NSC-G3-E-41× feed drive controller equipped with a CMC-41×3 motion card from Galil, which supports PVT motion mode.

The experiment procedure is described in Fig. 7, where the scheduled velocities for the raster path were sent to the IBF machine controller. The stitching interferometry platform [44] was used to measure the surface form errors before and after IBF. As shown in Figs. 3(b) and 7(b), the height error on the mirror surface was greatly improved from 6.32 nm RMS to 0.19 nm RMS in only one IBF run. Moreover, the measured residual error coincides with the estimation shown in Fig. 7(a), which confirms the feasibility of applying the proposed PVT-based velocity scheduler to real finishing processes.

6. Open-sourced PVT-based velocity scheduler and simulator

The PVT modulated velocity scheduler proposed in this study is open sourced with both MATLAB and C++ implementations [42]. Steps 2–4 shown in Fig. 1 are all included in the implementations. For small-scale problems, the MATLAB version is sufficient to provide the results in a reasonable time. Also, we found that the velocity-based estimation is preferred to the dwell-time-based estimation, since it reflects the effects of tool paths on the material removal distribution.

However, due to the heavy computational burden in Steps 3 and 4, this velocity-based estimation cannot be realized when the number of the velocity control points on a tool path are large. Therefore, we accelerated these two steps in C++ by exploiting their parallel computing patterns. As tested on a workstation equipped with an Intel (R) Xeon(R) Gold 5118 CPU (2.30GHz and 12 cores) and 64 GB RAM, the parallelized implementation is 30 to 60 times faster than the MATLAB version.

As an example, the raster-path simulation shown in Fig. 6(a) contains 8434 points on the up-sampled tool path. The surface error map was measured with 168 × 985 pixels. The C++ based simulation only cost 1.7 s to obtain the estimation shown in Fig. 6(e).

7. Conclusion

In this study, we proposed a Position-Velocity-Time (PVT) modulated velocity scheduler to obtain smooth and reliable velocities considering CNC dynamic limits for PVT-enabled machine controllers. The PVT model used to convert dwell times to velocities were formulated, and the quadratic programming was introduced to optimize the velocities by considering the dynamic and continuity constraints simultaneously. Compared with the conventional constant-acceleration-based velocity scheduler, the proposed method avoided the overshooting problem and achieved much smoother motion. From the simulation of applying the proposed method to different kinds of tool paths, it was found that, even with sharp corning, the scheduled velocities could accurately be used to traverse the planned tool paths. Finally, the experimental result verified that the PVT-based scheduler could be reliably applied to achieve a high level of accuracy at sub-0.3 nm root mean square.

Supplementary data to this article can be found online at https://doi.org/10.1016/j.jmapro.2023.01.005.

CRediT authorship contribution statement

Tianyi Wang: Conceptualization, methodology, data curation, experiment, writing – original draft preparation. Xiaolong Ke: Methodology, simulation, data curation, validation. Lei Huang: Methodology, validation, writing – review and editing. Vipender Negi: Simulation, data curation, writing - review and editing. Heejoo Choi: Validation, writing – review and editing. Weslin Pullen: Validation, Writing - review and editing. Daewook Kim: Validation, Writing - review and editing. Yi Zhu: Instrumentation, writing – review and editing. Mourad Idir: Supervision, project administration, funding acquisition, writing – review and editing.

Declaration of competing 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 paper.

Acknowledgements

This work was supported by the Accelerator and Detector Research Program, part of the Scientific User Facility Division of the Basic Energy Science Office of the U.S. Department of Energy (DOE), under the Field

Work Proposal No. FWP-PS032. This research was performed at the Optical Metrology Laboratory at the National Synchrotron Light Source II, a U.S. DOE Office of Science User Facility operated by Brookhaven National Laboratory (BNL) under the Contract No. DE-SC0012704. This work was performed under the BNL LDRD 17-016 “Diffraction limited and wavefront preserving reflective optics development”. This work was also supported by Natural Science Foundation of Fujian Province, China, under the grant number of 2022J011245.

References

[1] Ghigo M, Vecchi G, Basso S, Citterio O, Civitani M, Mattaini E, Pareschi G, Sironi G. Ion figuring of large prototype mirror segments for the E-ELT. In: Navarro R, Cunningham CR, Barto AA, editors. Advances in optical and mechanical technologies for telescopes and instrumentation. vol. 9151. International Society for Optics and Photonics, SPIE; 2014. p. 225–36.

[2] Fanson J, Bernstein R, Angeli G, Ashby D, Bigelow B, Brossus G, Bouchez A, Burgett W, Contos A, Demers R, Figueroa F, Fischer B, Groark F, Laskin R, Millan-Gabet R, Pi M, Wheeler N. Overview and status of the giant magellan telescope project. In: Ground-based and airborne telescopes VIII. volume 11445. International Society for Optics and Photonics; 2020.

[3] Beaucamp A, Namba Y. Super-smooth finishing of diamond turned hard x-ray molding dies by combined fluid jet and bonnet polishing. CIRP Ann 2013;62:315–8.

[4] Peverini L, Kozhevnikov I, Rommeveaux A, Vaerenbergh P, Claustre L, Guillet S, Massonnat J-Y, Ziegler E, Susini J. Ion beam profiling of aspherical x-ray mirrors. Nucl Instrum Methods Phys Res, Sect A 2010;616:115–8. X-Ray Mirror.

[5] Wang T, Huang L, Kang H, Choi H, Kim DW, Tayabaly K, Idir M. RIFTA: a robust iterative fourier transform-based dwell time algorithm for ultra-precision ion beam figuring of synchrotron mirrors. Sci Rep 2020;10:1–12.

[6] Wang T, Huang L, Choi H, Vescovi M, Kuhne D, Zhu Y, Pullen WC, Ke X, Kim DW, Kemao Q, et al. RISE: robust iterative surface extension for sub-nanometer x-ray mirror fabrication. Opt Express 2021;29:15114–32.

[7] Wang T, Huang L, Vescovi M, Kuhne D, Zhu Y, Negi VS, Zhang Z, Wang C, Ke X, Choi H, et al. Universal dwell time optimization for deterministic optics fabrication. Opt Express 2021;29:38737–57.

[8] Beaucamp A, Simon P, Charlton P, King C, Matsubara A, Wegener K. Brittle-ductile transition in shape adaptive grinding (sag) of sic aspheric optics. Int J Mach Tools Manuf 2017;115:29–37.

[9] Wischmeier L, Graeupner P, Kuerz P, Kaiser W, van Schoot J, Mallmann J, de Pee J, Stoeldraijer J. High-na euv lithography optics becomes reality. In: Extreme Ultraviolet (EUV) lithography XI. volume 11323. International Society for Optics and Photonics; 2020. p. 1132308.

[10] Zhu W-L, Beaucamp A. Compliant grinding and polishing: a review. Int J Mach Tools Manuf 2020;158:103634.

[11] Walker DD, Brooks D, King A, Freeman R, Morton R, McCavana G, Kim S-W. The ‘precessions’ tooling for polishing and figuring flat, spherical and aspheric surfaces. Opt Express 2003;11:958–64.

[12] Fähnle OW, Brug HVan, Frankena HJ. Fluid jet polishing of optical surfaces. Appl Opt 1998;37:6771–3.

[13] Wang C, Cheung CF, Ho L, Liu M, Lee WB. A novel multi-jet polishing process and tool for high-efficiency polishing. Int J Mach Tools Manuf 2017;115:60–73.

[14] Kansal H, Singh AK, Grover V. Magnetorheological nano-finishing of diamagnetic material using permanent magnets tool. Precis Eng 2018;51:30–9.

[15] Wang T, Huang L, Zhu Y, Vescovi M, Khune D, Kang H, Choi H, Kim DW, Tayabaly K, Bouet N, et al. Development of a position–velocity–time-modulated two-dimensional ion beam figuring system for synchrotron x-ray mirror fabrication. Appl Opt 2020;59:3306–14.

[16] Cheng H. Independent variables for optical surfacing systems. Springer; 2016.

[17] Jones RA. Optimization of computer controlled polishing. Appl Opt 1977;16:218–24.

[18] Wilson S, McNeil J. Neutral ion beam figuring of large optical surfaces. In: Current developments in optical engineering II. volume 818. SPIE; 1987. p. 320–4.

[19] Carnal CL, Egert CM, Hylton KW. Advanced matrix-based algorithm for ion-beam milling of optical components. In: Current developments in optical design and optical engineering II. volume 1752. International Society for Optics and Photonics; 1992. p. 54–62.

[20] Wang C, Yang W, Wang Z, Yang X, Hu C, Zhong B, Guo Y, Xu Q. Dwell-time algorithm for polishing large optics. Appl Opt 2014;53:4752–60.

[21] Wu JF, Lu ZW, Zhang HX, Wang TS. Dwell time algorithm in ion beam figuring. Appl Opt 2009;48:3930–7.

[22] Dong Z, Cheng H. Toward the complete practicability for the linear-equation dwell time model in subaperture polishing. Appl Opt 2015;54:8884–90.

[23] Dong Z, Cheng H, Tam H-Y. Robust linear equation dwell time model compatible with large scale discrete surface error matrix. Appl Opt 2015;54:2747–56.

[24] Song C, Dai Y, Peng X. Model and algorithm based on accurate realization of dwell time in magnetorheological finishing. Appl Opt 2010;49:3676–83.

[25] Zhang Y, Fang F, Huang W, Fan W. Dwell time algorithm based on bounded constrained least squares under dynamic performance constraints of machine tool in deterministic optical finishing. Int J Precis Eng Manuf Green Technol 2021;8:1415–27.

[26] Wang T, Huang L, Vescovi M, Kuhne D, Tayabaly K, Bouet N, Idir M. Study on an effective one-dimensional ion-beam figuring method. Opt Express 2019;27:15368–81.

[27] Li L, Xue D, Deng W, Wang X, Bai Y, Zhang F, Zhang X. Positive dwell time algorithm with minimum equal extra material removal in deterministic optical surfacing technology. Appl Opt 2017;56:9098–104.

[28] Zhu W, Beaucamp A. Zernike mapping of optimum dwell time in deterministic fabrication of freeform optics. Opt Express 2019;27:28692–706.

[29] Ke X, Wang T, Choi H, Pullen W, Huang L, Idir M, Kim DW. Dual-tool multiplexing model of parallel computer controlled optical surfacing. Opt Lett 2020;45:6426–9.

[30] Kim D, Ke X, Pullen W, Wang T, Choi H, Negi VS, Huang L, Idir M. Generalized large optics fabrication multiplexing. J Eur Opt Soc 2022;18:2.

[31] Wan K, Wan S, Jiang C, Wei C, Shao J. Sparse bi-step raster path for suppressing the mid-spatial-frequency error by fluid jet polishing. Opt Express 2022;30:6603–16.

[32] Dunn CR, Walker DD. Pseudo-random tool paths for CNC subaperture polishing and other applications. Opt Express 2008;16:18942–9.

[33] Wang C, Wang Z, Xu Q. Unicursal random maze tool path for computer-controlled optical surfacing. Appl Opt 2015;54:10128–36.

[34] Negi VS, Wang T, Garg H, Pullen WC, Ke X, RR SK, Choi H, Tiwari UK, Karar V, Kim D. Random adaptive tool path for zonal optics fabrication. Opt Express 2022;30:29295–309.

[35] Beaucamp A, Takizawa K, Han Y, Zhu W. Reduction of mid-spatial frequency errors on aspheric and freeform optics by circular-random path polishing. Opt Express 2021;29:29802–12.

[36] Wang C, Han Y, Zhang H, Liu C, Jiang L, Qian L. Suppression of mid-spatial-frequency waviness by a universal random tree-shaped path in robotic bonnet polishing. Opt Express 2022;30:29216–33.

[37] Dai Yifan ZL, Xuhui X, et al. Deterministic figuring in optical machining by ion beam. Acta Opt Sin 2008;28:1131–5.

[38] Zhou L, Xie X, Dai Y, Jiao C, Li S. Realization of velocity mode in flat optics machining using ion beam. J Mech Eng 2009;45:152–6.

[39] Mizoue Y, Sencer B, Beaucamp A. Identification and optimization of cnc dynamics in time-dependent machining processes and its validation to fluid jet polishing. Int J Mach Tools Manuf 2020;159:103648.

[40] Han Y, Zhu W-L, Zhang L, Beaucamp A. Region adaptive scheduling for time-dependent processes with optimal use of machine dynamics. Int J Mach Tools Manuf 2020;156:103589.

[41] Lu L, Han J, Fan C, Xia L. A predictive feedrate schedule method for sculpture surface machining and corresponding b-spline-based irredundant pvt commands generating method. Int J Adv Manuf Technol 2018;98:1763–82.

[42] Wang T, Ke X, Huang L, Idir M. PVT-based velocity scheduler for computer contrplled optical surfacing. URL: https://github.com/TWANG006/pvt-simulator.git; 2022.

[43] Huang L, Wang T, Nicolas J, Vivo A, Polack F, Thomasset M, Zuo C, Tayabaly K, Kim DW, Idir M. Two-dimensional stitching interferometry for self-calibration of high-order additive systematic errors. Opt Express 2019;27:26940–56.

[44] Huang L, Wang T, Tayabaly K, Kuhne D, Xu W, Xu W, Vescovi M, Idir M. Stitching interferometry for synchrotron mirror metrology at national synchrotron light source ii (nsls-ii). Opt Lasers Eng 2020;124:105795.

[45] Huang L, Wang T, Polack F, Nicolas J, Nakhoda K, Idir M. Measurement uncertainty of highly asymmetrically curved elliptical mirrors using multi-pitch slope stitching technique. Front Phys 2022:414.