








Universal dwell time optimization for deterministic optics fabrication

TIANYI WANG,^{1,10}  LEI HUANG,^{1,11}  MATTHEW VESCOVI,¹
DENNIS KUHNE,¹ YI ZHU,¹ VIPENDER S. NEGI,^{2,3}  ZILI ZHANG,⁴
CHUNJIN WANG,⁴  XIAOLONG KE,⁵ HEEJOO CHOI,^{6,7} WESLIN C.
PULLEN,⁶ DAEWOOK KIM,^{6,7,8} QIAN KEMAO,⁹  KASHMIRA
NAKHODA,⁴ NATHALIE BOUET,¹ AND MOURAD IDRIR¹

¹National Synchrotron Light Source II (NSLS-II), Brookhaven National Laboratory, PO Box 5000, Upton, NY 11973, USA

²Council of Scientific and Industrial Research-Central Scientific Instruments Organisation (CSIR-CSIO), Chandigarh- 160030, India

³Academy of Scientific and Innovative Research (AcSIR), Ghaziabad- 201002, India

⁴State Key Laboratory of Ultra-precision Machining Technology, Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, China

⁵School of Mechanical and Automotive Engineering, Xiamen University of Technology, Xiamen, 361024, China

⁶James C. Wyant College of Optical Sciences, the University of Arizona, 1630 E. University Blvd., P.O. Box 210094, Tucson, AZ 85721-0094, USA

⁷Large Binocular Telescope Observatory, University of Arizona, Tucson, AZ 85721, USA

⁸Department of Astronomy and Steward Observatory, University of Arizona, 933 N. Cherry Ave., Tucson, AZ 85721, USA

⁹School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798

¹⁰tianyi@bnl.gov

¹¹huanglei0114@gmail.com

Abstract: Computer-Controlled Optical Surfacing (CCOS) has been greatly developed and widely used for precision optical fabrication in the past three decades. It relies on robust dwell time solutions to determine how long the polishing tools must dwell at certain points over the surfaces to achieve the expected forms. However, as dwell time calculations are modeled as ill-posed deconvolution, it is always non-trivial to reach a reliable solution that 1) is non-negative, since CCOS systems are not capable of adding materials, 2) minimizes the residual in the clear aperture 3) minimizes the total dwell time to guarantee the stability and efficiency of CCOS processes, 4) can be flexibly adapted to different tool paths, 5) the parameter tuning of the algorithm is simple, and 6) the computational cost is reasonable. In this study, we propose a novel Universal Dwell time Optimization (UDO) model that universally satisfies these criteria. First, the matrix-based discretization of the convolutional polishing model is employed so that dwell time can be flexibly calculated for arbitrary dwell points. Second, UDO simplifies the inverse deconvolution as a forward scalar optimization for the first time, which drastically increases the solution stability and the computational efficiency. Finally, the dwell time solution is improved by a robust iterative refinement and a total dwell time reduction scheme. The superiority and general applicability of the proposed algorithm are verified on the simulations of different CCOS processes. A real application of UDO in improving a synchrotron X-ray mirror using Ion Beam Figuring (IBF) is then demonstrated. The simulation indicates that the estimated residual in the $92.3 \text{ mm} \times 15.7 \text{ mm}$ CA can be reduced from 6.32 nm Root Mean Square (RMS) to 0.20 nm RMS in 3.37 min. After one IBF process, the measured residual in the CA converges to 0.19 nm RMS, which coincides with the simulation.

© 2021 Optical Society of America under the terms of the [OSA Open Access Publishing Agreement](#)

1. Introduction

With the rapid development of precision technologies, the demand of high-precision optics has drastically increased in various cutting-edge applications, such as telescopes for space exploration [1,2], X-ray mirrors for synchrotron radiation and free-electron laser facilities [3–8], and optics in EUV lithography [9,10]. Precision optical surfaces are manufactured with Computer Controlled Optical Surfacing (CCOS) [11,12] processes, including small-tool polishing [13], bonnet polishing [14,15], magnetorheological finishing [16,17], Ion Beam Figuring (IBF) [7,8,18–23], *etc.* Different CCOS processes can be adopted based on the requirement of the precision and shape of the desired optical surfaces.

In a typical CCOS process, an effective area within an optical surface, named Clear Aperture (CA), needs to be corrected to the specified error residual requirement. This process is deterministically guided by the dwell time calculated along different tool paths from the convolutional polishing model [11]. In this model, the removed material is equal to the convolution between the Tool Influence Function (TIF) and the dwell time as

$$z(x, y) = b(x, y) * t(x, y), \quad (1)$$

where "*" represents the convolution operator, $z(x, y)$ is the removed material, $b(x, y)$ is the TIF that describes the material wear of the polishing tool that can be measured in practice, and $t(x, y)$ is the dwell time of the TIF at (x, y) . Substituting $z(x, y)$ in Eq. (1) with the desired removal $z_d(x, y)$, $t(x, y)$ can be solved via deconvolution. Note that, as shown in Fig. 1, in order to obtain the valid result in a CA, the size of $z_d(x, y)$ should be at least larger than the outline perimeter of the CA by the radius of $b(x, y)$ [7].

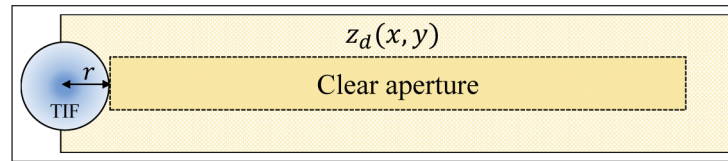


Fig. 1. The size of $z_d(x, y)$ should be at least greater than the outline perimeter of the CA by the radius of a TIF.

Since deconvolution is an ill-posed inverse problem, it does not converge to a unique solution. Therefore, the objective of dwell time calculation becomes obtaining a reasonable solution that satisfies the following criteria.

1. $t(x, y) \geq 0$, since CCOS systems are not capable of adding materials.
2. $[z_d(x, y) - z(x, y)]_{CA}$ is minimized, where $[\cdot]_{CA}$ represents only keeping the CA region of " \cdot ".
3. The total dwell time, *i.e.* $\sum t(x, y)$, should be as short as possible to guarantee the stability and efficiency of CCOS processes.
4. The samplings of $z_d(x, y)$ and $t(x, y)$ can vary so that the calculated dwell time can be flexibly adapted to different tool paths in practice.
5. The parameter tuning is simple that no manual selection of the hyper-parameters is needed in the calculation.
6. The computational and memory cost of obtaining $t(x, y)$ should be reasonable.

Various dwell time calculation (*i.e.* deconvolution) algorithms that partially satisfy these criteria have been proposed in the literature, which are the Fourier transform algorithm [24] (including our proposed Robust Iterative Fourier Transform-based dwell Time Algorithm (RIFTA) [8] and Robust Iterative Surface Extension (RISE) [7]), the iterative algorithm [25,26], the Bayesian algorithm [27], and the matrix-based algorithms (including the minimum-norm algorithm [28–31] and the least-squares algorithm [20,32–34]). The performances of the existing algorithms based on the six criteria mentioned above are summarized in Table 1. The details of each algorithm will be reviewed in Section 2.

Table 1. Evaluation of the existing dwell time calculation algorithms versus the proposed UDO algorithm based on the six criteria of a desirable dwell time solution.

Algorithm	$t \geq 0$	min. $(z_d - z)_{CA}$	min. $\sum t$	Flexible dwell	Simple parameter tuning	Speed
Fourier transform						
Conventional [24]	✓	✓	×	×	×	✓✓
RIFTA [8]	✓	✓	✓	×	✓	✓✓
RISE [7]	✓	✓	✓	×	✓	✓✓
Iterative [25,26]	✓	✓	×	×	×	✓
Bayesian [27]	✓	✓	×	×	×	✓
Matrix-based						
Minimum-norm [28–31]	×	✓	×	✓	×	×
Least-squares [20,32–34]	✓	✓	×	✓	×	×
UDO (this work)	✓	✓	✓	✓	✓	✓

The complexities of obtaining a desirable dwell time solution, as described above, are mainly a result from solving it as an inverse deconvolution problem. Indeed, these complexities can be simplified if the solution process is more appropriately modeled. Therefore, in this study, we propose a novel Universal Dwell time Optimization (UDO) algorithm that universally considers all the criteria of a desirable dwell time solution (see Table 1). In UDO, the matrix-based discretization of Eq. (1) is first performed so that the dwell time can be directly calculated for arbitrary dwell points. Second, instead of the inverse deconvolution, a forward scalar optimization model is proposed to solve for the dwell time, which greatly increases the solution stability and reduces the heavy computational burden of parameter tuning and equation solving in the conventional matrix-based algorithm [28–30]. Finally, the RISE concept [7] is employed to iteratively refine the dwell time solution by further minimizing the estimated residual in a certain CA. A total dwell time reduction scheme is introduced in each iteration to minimize the increase of the total dwell time.

To verify the superiority and general applicability of UDO, it has been applied to different CCOS simulations with both the raster and spiral tool paths. We also adopt UDO in a real application of improving an existing synchrotron X-ray mirror using Ion Beam Figuring (IBF) [21] for the Coherent Hard X-ray scattering (CHX) beamline at the National Synchrotron Light Source II (NSLS-II). The UDO simulation indicates that the figure error in a $92.3 \text{ mm} \times 15.7 \text{ mm}$ CA could be reduced from 6.32 nm Root Mean Square (RMS) to 0.20 nm RMS in 3.37 min. After one real IBF run, the measured figure error residual in the CA converged to 0.19 nm RMS, which verifies the effectiveness of the proposed UDO algorithm.

The rest of the paper is organized as below. Section 2 reviews the existing dwell time calculation algorithms. The proposed UDO concept is explained in Section 3. The simulations of applying UDO to different CCOS processes are given in Section 4, followed by the details of the real application of UDO to IBF in Section 5. Section 6 concludes the paper.

2. Existing dwell time calculation algorithms

The existing dwell time calculation algorithms, including the Fourier transform algorithm (including RIFTA and RISE), the iterative algorithm, the Bayesian algorithm, and the matrix-based algorithms are reviewed in this section.

2.1. Fourier transform algorithm

The conventional Fourier transform algorithm. The Fourier transform algorithm transfers the convolution in Eq. (1) to a frequency-domain pointwise multiplication as

$$Z_d(u, v) = B(u, v) \times T(u, v), \quad (2)$$

where $Z_d(u, v)$, $B(u, v)$, and $T(u, v)$ are the Fourier transforms of $z_d(x, y)$, $b(x, y)$, and $t(x, y)$, respectively. The dwell time $t(x, y)$ is then solved as

$$t(x, y) = \mathcal{F}^{-1} \left[\frac{Z_d(u, v)}{B(u, v)} \right], \quad (3)$$

where $\mathcal{F}^{-1}(\cdot)$ represents the inverse Fourier transform. However, as the amplitudes at certain frequencies of $B(u, v)$ are very close to zero, the high-frequency errors and noise in $Z_d(u, v)$ at those frequencies are enormously amplified, resulting in inaccurate $t(x, y)$. Therefore, a thresholded inverse filter $\bar{B}(u, v; \alpha)$ has been employed to stabilize Eq. (3) as

$$t(x, y; \alpha) = \mathcal{F}^{-1} \left[\frac{Z_d(u, v)}{\bar{B}(u, v; \alpha)} \right], \quad (4)$$

where

$$\bar{B}(u, v; \alpha) = \begin{cases} B(u, v), & \text{if } |B(u, v)| > \alpha \\ \alpha, & \text{otherwise} \end{cases}, \quad (5)$$

and α is the inverse filtering threshold.

The Fourier transform algorithm guarantees $t(x, y) \geq 0$ as long as $z_d(x, y)$ is piston-adjusted to be non-negative. Also, it is computationally efficient thanks to the Fast Fourier Transform (FFT) algorithm. However, because α depends on the spectrum of $Z_d(u, v)$, it is hard to be automatically tuned and thus cannot be guaranteed to minimize $[z_d(x, y) - z(x, y)]_{CA}$. Also, no constraint is put on $\sum t(x, y)$, and the samplings of $z_d(x, y)$ and $t(x, y)$ should be the same because of the Fourier transform.

RIFTA and RISE. RIFTA automates the tuning of α with a direct search algorithm [8]. The optimal α , *i.e.* α_{opt} , is obtained by minimizing the RMS of the estimated residual in a CA as

$$\begin{aligned} \alpha_{opt} &= \underset{\alpha}{\operatorname{argmin}} \operatorname{RMS} [z_d(x, y) - z(x, y; \alpha)]_{CA} \\ &= \underset{\alpha}{\operatorname{argmin}} \operatorname{RMS} [z_d(x, y) - b(x, y) * t(x, y; \alpha)]_{CA}. \end{aligned} \quad (6)$$

In addition, RIFTA uses a two-level iterative scheme to minimize the total dwell time without affecting the estimated residual in a CA. The inner-level iterations progressively adjust the piston of $z_d(x, y)$ based on the residual in the CA, which guarantees that $z_d(x, y)$ is always adjusted by the smallest (*i.e.* optimal) piston in each iteration so that the penalty on $\sum t(x, y)$ is minimized. The outer-level iterations are added to minimize the size of the dwell grid, which further reduces $\sum t(x, y)$.

Based on RIFTA, RISE [7] further refines a dwell time solution by iteratively extending the estimated residual map in a CA until it is reduced to the specified Peak-to-Valley (PV) or RMS

requirement. A polynomial fitting-based extension strategy is proposed to avoid the influence of noise and data discontinuity, and ensures that only the correctable figure errors are extended. In each iteration, an incremental dwell time is calculated using the extended surface map. The final dwell time solution is then obtained as the sum of these incremental dwell time maps.

Note that $t(x, y)$ calculated from RIFTA or RISE still has the same sampling as $z_d(x, y)$. To adapt it to different machining intervals in practice, we proposed to use the bicubic resampling algorithm. This works fine for raster tool paths with uniform sampling intervals, however, it can hardly be adapted to the dwell points on a spiral or even random tool path.

2.2. Iterative algorithm

The iterative deconvolution algorithm was proposed by Van Cittert in 1931 [35] and introduced in the optimization of CCOS [25] in 1977. In the iterative algorithm, $\beta z_d(x, y)$ is taken as $t^0(x, y)$, the initial guess to $t(x, y)$, where $\beta \in (0, 1]$ is a constant. The Van Cittert iterations converge to the inverse filtering in the Fourier transform algorithm [36] when noise is absent (which is impossible in real applications). Various convergence criteria have been studied [16,36], however, they are too strict for CCOS processes, especially when a TIF is not axially-symmetric and does not have a central peak (*i.e.* the strongest removal rate at the center) [16,25].

To improve the convergence performance, the Volumetric Removal Rate (VRR) of a TIF has been introduced to determine a better $t^0(x, y)$ [26], in which $\beta = \frac{1}{\text{VRR}}$ is selected. Also, a relaxation factor κ [16,26] has been introduced to damp the iterations as

$$t^{k+1}(x, y) = t^k(x, y) + \kappa [z_d(x, y) - t^k(x, y) * b(x, y)]. \quad (7)$$

The iterative algorithm is fast, since the convolutions in Eq. (7) can be accelerated by FFT. The stopping condition for Eq. (7) can be set by examining $[z_d(x, y) - z(x, y)]_{CA}$. However, a reasonable κ depends on both $z_d(x, y)$ and $b(x, y)$, which has to be manually selected by trial-and-error. To ensure $t(x, y) \geq 0$, either the negative entries of $t^k(x, y)$ is set as zero or the entire $t^k(x, y)$ is offset by its smallest entry. Therefore, $\sum t(x, y)$ is not minimized. Also, the samplings of $z_d(x, y)$ and $t(x, y)$ must be identical.

2.3. Bayesian algorithm

The Bayesian algorithm (*i.e.* Richardson–Lucy deconvolution [37,38]) has been introduced to CCOS by assuming that $t(x, y)$ satisfies the uniform distribution [27]. The Bayesian statistical model describes the relationship among the posterior $P(t|z_d)$, the prior $P(t)$, and the likelihood $P(z_d|t)$ as

$$P(t|z_d) = P(z_d|t) \frac{P(t)}{P(z_d)}. \quad (8)$$

Assuming that $P(z_d|t)$ satisfies the Poisson distribution with the parameter $b * t$, maximizing Eq. (8) can be transferred into the following minimization problem [27],

$$t_{opt} = \underset{t}{\operatorname{argmin}} J_1(t), \quad (9)$$

where

$$J_1(t) = \int \int [b * t - z_d \times \log(b * t)] dx dy. \quad (10)$$

Setting the gradient $\nabla J_1(t) = 0$, with the calculus of variation, Eq. (10) is solved by a multiplicative algorithm [37] as

$$t^{k+1} = t^k \times \left[\frac{z_d(-x, -y)}{\int \int z_d(x, y) dx dy} * \frac{z}{b * t^k} \right]. \quad (11)$$

The division in Eq. (11) also brings the same noise amplification problem as that in the Fourier transform algorithm. Therefore, to smooth and denoise the dwell time, a total variation

regularization term

$$J_2(t) = \delta \iint \|\nabla t\| dx dy \quad (12)$$

is added to the MAP objective as

$$t_{opt} = \underset{t}{\operatorname{argmin}} (J_1 + J_2), \quad (13)$$

which is solved as

$$t^{k+1} = \frac{t^k}{1 - \delta \frac{\operatorname{div}(\nabla t^k)}{|\nabla t^k|}} \times \left[\frac{z_d(-x, -y)}{\iint z_d(x, y) dx dy} * \frac{z_d}{b * t^k} \right], \quad (14)$$

where $\operatorname{div}(\cdot)$ represents the divergence of " \cdot " and δ is a constant.

Equation 14 has a good property that $t(x, y) \geq 0$ is guaranteed if $t^0(x, y) \geq 0$. It can also be accelerated by FFT to calculate the two convolutions [27]. The stopping condition for Eq. (14) can be set by examining $[z_d(x, y) - z(x, y)]_{CA}$. However, δ in the denominator is hard to tune in practice. Both $t(x, y)$ and $[z_d(x, y) - z(x, y)]_{CA}$ depend on δ . A small δ may achieve the expected residual but cannot filter the noise of the dwell time, while a large δ smooths the dwell time so much that the estimated residual would deviate from the expectation. Also, the samplings of $z_d(x, y)$ and $t(x, y)$ cannot vary.

2.4. Matrix-based algorithm

The matrix-based algorithm discretizes Eq. (1) as

$$z_d(x_j, y_j) = \sum_{i=0}^{N_t-1} b(x_j - \xi_i, y_j - \eta_i) t(\xi_i, \eta_i), \quad (15)$$

for $j = 0, 1, \dots, N_z - 1$, where N_z is the number of elements in $z_d(x_j, y_j)$, N_t is the number of dwell points, (ξ_i, η_i) is the i th dwell point, and $b(x_j - \xi_i, y_j - \eta_i)$ represents the material removed per unit time at point (x_j, y_j) when the TIF dwells at (ξ_i, η_i) . Therefore, this discretization allows $z_d(x_j, y_j)$ to have different samplings from $t(\xi_i, \eta_i)$. Equation 15 can be expressed in a matrix form as

$$\underbrace{\begin{pmatrix} z_{d0} \\ z_{d1} \\ \vdots \\ z_{dN_z-1} \end{pmatrix}}_{\mathbf{z}_d} = \underbrace{\begin{pmatrix} b_{0,0} & b_{0,1} & \dots & b_{0,N_t-1} \\ b_{1,0} & b_{1,1} & \dots & b_{1,N_t-1} \\ \vdots & \vdots & \ddots & \vdots \\ b_{N_z-1,0} & b_{N_z-1,1} & \dots & b_{N_z-1,N_t-1} \end{pmatrix}}_{\mathbf{B}} \underbrace{\begin{pmatrix} t_0 \\ t_1 \\ \vdots \\ t_{N_t-1} \end{pmatrix}}_{\mathbf{t}}, \quad (16)$$

where $\mathbf{z}_d \in \mathbb{R}^{N_z}$ and $\mathbf{t} \in \mathbb{R}^{N_t}$ are vectors representing the flattened $z_d(x_j, y_j)$ and $t(\xi_i, \eta_i)$, respectively, and $z_{dj} = z_d(x_j, y_j)$, $b_{j,i} = b(x_j - \xi_i, y_j - \eta_i)$, and $t_i = t(\xi_i, \eta_i)$.

Minimum-norm solution. Since \mathbf{B} is always rank deficient [28], the least-squares solution to Eq. (16) is not unique, which may result in arbitrary \mathbf{t} that hardly achieves the desired $[z_d(x, y) - z(x, y)]_{CA}$ [20,33]. Therefore, the Singular Value Decomposition (SVD) can be used to obtain a unique, minimum-norm solution (*i.e.* minimizing $\|\mathbf{t}\|$) to Eq. (16) [28] as

$$\mathbf{t} = \sum_{j=1}^{N_z} \frac{\mathbf{u}_j^T \mathbf{z}_d}{\sigma_j} \mathbf{v}_j, \quad (17)$$

where σ_i , $i = 1, 2, 3, \dots, N_t$ are the singular values appearing in non-increasing order; \mathbf{u}_i and \mathbf{v}_i are the left and right singular vectors of \mathbf{B} , respectively. However, the problem of Eq. (17) is

that many of the singular values are very close to zero because \mathbf{B} is not full-rank. These small singular values can be truncated [30] and only the first M singular values are kept as

$$\mathbf{t} = \sum_{j=1}^M \frac{\mathbf{u}_j^T \mathbf{z}_d}{\sigma_j} \mathbf{v}_j, \quad M \leq N_z. \quad (18)$$

Alternatively, a damping factor ζ has been introduced to Eq. (17) as

$$\mathbf{t} = \sum_{j=1}^{N_z} \frac{\sigma_j \mathbf{u}_j^T \mathbf{z}_d}{\sigma_j^2 + \zeta^2} \mathbf{v}_j, \quad (19)$$

and the more efficient Least-Squares with QR factorization (LSQR) solver for sparse linear systems is used [28,29,31].

In the minimum-norm solution, $[z_d(x, y) - z(x, y)]_{CA}$ is minimized by including enough singular values. However, $\sum t(x, y)$ is not guaranteed to be minimal, even though $\|\mathbf{t}\|$ is minimized. Also, $\mathbf{t} \geq 0$ cannot be automatically guaranteed. An extra piston adjustment of \mathbf{z}_d or offsetting the negative entries in \mathbf{t} is required. In addition, either the number of singular values M or the damping factor ζ is a hyper-parameter that has to be manually selected by trial-and-error, which adds extra computational burden to the already computationally expensive SVD and LSQR solvers when the resolution of \mathbf{z}_d is high or the dwell points are dense.

Least-squares solution. Equation 16 can still be solved in a least-squares manner by adding certain constraints to regularize the search space. The Constrained Linear Least-Squares (CLLS) algorithm [20,32,34] has been proposed to introduce different constraints to solve Eq. (16) as

$$\begin{aligned} \mathbf{t}_{opt} &= \underset{\mathbf{t}}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{z}_d - \mathbf{B}\mathbf{t}\|_2^2 \\ \text{s.t. } \mathbf{A}\mathbf{t} &\leq \mathbf{b} \\ \tau_{min} &\leq \mathbf{t} \leq \tau_{max} \end{aligned} \quad (20)$$

where \mathbf{A} is the adjacent element constraint matrix of \mathbf{t} , \mathbf{b} is the upper limit of the adjacent elements of \mathbf{t} [20,32–34], and $\tau_{min} \leq \mathbf{t} \leq \tau_{max}$ is the bounding constraint that guarantees $\mathbf{t} \geq 0$ and can be tuned based on the kinematic property of translation stages [32,34]. If the inequality constraints are too strict, however, \mathbf{t} would be so smooth that it may fail to achieve the desired estimated residual in the CA. To solve this problem, a coarse-to-fine scheme [20] was employed. On the coarse level, the constraints are applied to obtain a coarse result \mathbf{t}_{coarse} , which is then polynomial fitted as \mathbf{t}_{fit} . On the finer level, more strict constraints are applied to calculate \mathbf{t}_{fine} based on the RMS of the estimated residual in the CA. The final dwell time map is the summation of the coarse and the fine level results as $\mathbf{t} = \mathbf{t}_{coarse} + \mathbf{t}_{fine}$.

CLLS adds constraints to regularize the inverse deconvolution and the quadratic programming is used to solve it. However, these constraints are not easy to properly define in practice and will bring even heavier computational burden to the already time-consuming least-squares solver.

3. Principle of the UDO algorithm

With the motivation to universally satisfy all the criteria of a desired dwell time solution, the UDO algorithm is proposed. As shown in Table 1, one distinctive advantage of the matrix-based algorithm over the others is that it allows the dwell points to have different sampling than the measured surface data. UDO also keeps this matrix-based formulation. While searching for the optimal dwell time solution, a physical intuition on the expected distribution of the dwell time is incorporated, turning the time-consuming vector optimization to a simple and efficient scalar optimization. An iterative refinement is then applied to minimize the residual error in the CA.

3.1. Analysis of the matrix-based formulation

We adopt the matrix-based formulation as elaborated in Eq. (16), from which the dwell time could have been immediately determined by a pseudo inverse as $\mathbf{t} = (\mathbf{B}^T \mathbf{B})^{-1} \mathbf{z}_d$. However, as the TIF removal effect overlaps multiple dwell points, the convolution matrix \mathbf{B} is always rank-deficient so that $\mathbf{B}^T \mathbf{B}$ is near singular and the solution of \mathbf{t} is not unique [28]. To obtain a unique \mathbf{t} , certain constraints have been included to regularize the optimization of \mathbf{t} . The SVD [28,30] and LSQR [27,28] algorithms, as shown in Eqs. (18) and 19, constrain \mathbf{t} by minimizing $\|\mathbf{t}\|$. To avoid the singularity problem of $\mathbf{B}^T \mathbf{B}$, either the small singular values are truncated [30] or the damping factor ζ is incorporated. Although this minimum-norm solution is mathematically correct, it does not consider the specific requirement of the smoothness of \mathbf{t} . On the other hand, the CLLS algorithm [20,33] in Eq. (20) adds more rigorous constraints to the adjacent elements in \mathbf{t} , which regularizes the local distributions of \mathbf{t} and thus has an effect on its smoothness. However, these constraints are hard to determine properly, and the solution process is so time-consuming that it restricts the application of CLLS to large-scale problems.

It is necessary to highlight the unique feature of the dwell time optimization in CCOS against the conventional deconvolution. In the conventional deconvolution, the task is usually to minimize the residual between the convolved signal ($\mathbf{B}^T \mathbf{t}$) and the observed signal (\mathbf{z}_d), where \mathbf{t} is deconvolved and often expected to contain higher-frequency details (*i.e.* sharper) than \mathbf{z}_d because the convolution is usually a low-pass operation. Nonetheless, in a CCOS process, our intuition is that \mathbf{t} should smoothly duplicate the shape of \mathbf{z}_d rather than contain any higher-frequency components than those in \mathbf{z}_d , since an unsmooth \mathbf{t} will incur frequent acceleration or deceleration to the translation stages in a CCOS system and thus reduces the fabrication stability and affect the convergence to the expected residual in the CA. In the existing algorithms, this physical intuition can by no means be achieved automatically. We are thus pursuing a solution of \mathbf{t} that directly and smoothly resembles the distribution of \mathbf{z}_d .

3.2. Deconvolution as scalar optimization

With the above analysis, we now return to Eq. (16), and multiply the transpose of \mathbf{B} on both sides of it as

$$\mathbf{B}^T \mathbf{z}_d = \mathbf{B}^T \mathbf{B} \mathbf{t}. \quad (21)$$

Equation (21) has two intriguing properties. First, on the left-hand side, \mathbf{B}^T transforms the vector space of \mathbf{z}_d from \mathbb{R}^{N_z} to \mathbb{R}^{N_t} , *i.e.* the same vector space as \mathbf{t} . Since \mathbf{B}^T is low-pass, this transformation not only changes the sampling of \mathbf{z}_d but also smooths it. Second, on the right-hand side of Eq. (21), $\mathbf{B}^T \mathbf{B}$ is symmetric and semi-positive definite. As the columns of \mathbf{B} are constructed by centering $b(x, y)$ at different dwell points, the peak in a certain column i appears to be at the $b_{i,i}$ entry of \mathbf{B} . Many of the other entries in this column always rapidly attenuates to zero. Therefore, as shown in Fig. 2, most of the energy of $\mathbf{B}^T \mathbf{B}$ concentrates on its diagonal.

Based on these two properties of Eq. (21), we found that the distribution described by $\mathbf{B}^T \mathbf{z}_d$ is already an excellent smoothness constraint on \mathbf{t} . Moreover, based on the energy-concentration effect of $\mathbf{B}^T \mathbf{B}$, we propose to approximate Eq. (21) by assuming that $\mathbf{B}^T \mathbf{B}$ is an identity matrix and \mathbf{t} is in the vector space defined by $\mathbf{B}^T \mathbf{z}_d$ as

$$\mathbf{t} = \mathbf{B}^T \mathbf{z}_d. \quad (22)$$

With this assumption, our solution \mathbf{t} smoothly resembles $\mathbf{B}^T \mathbf{z}_d$, as we desire. However, such a brute-force solution is not ideal since it is an intuitive approximation and does not minimize the fabrication residue in the CA. Therefore, while keeping \mathbf{t} smooth, it is necessary to simultaneously minimize the residual in the CA. To achieve this, we propose to damp the

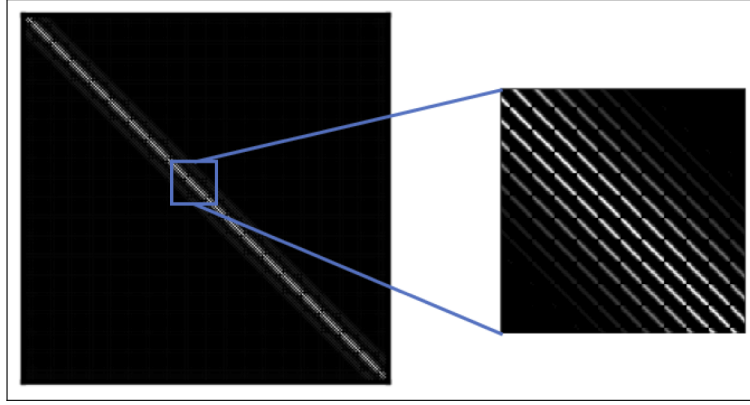


Fig. 2. Schematic of the distribution of $\mathbf{B}^T \mathbf{B}$.

obtained \mathbf{t} by introducing flexibility into the brute-force approximation, and then iteratively refine the solution by minimizing the residual in the CA.

In detail, first, a damping factor γ is introduced to Eq. (22), and \mathbf{t} is calculated as

$$\mathbf{t} = \gamma_{opt} \mathbf{B}^T \mathbf{z}_d, \quad (23)$$

where γ_{opt} is the optimal damping that can be determined from a scalar optimization as

$$\begin{aligned} \gamma_{opt} &= \underset{\gamma}{\operatorname{argmin}} \operatorname{RMS}(\mathbf{e}_{CA}) \\ &= \underset{\gamma}{\operatorname{argmin}} \operatorname{RMS}(\mathbf{z}_d - \mathbf{B}\mathbf{t}) \\ &= \underset{\gamma}{\operatorname{argmin}} \operatorname{RMS}(\mathbf{z}_d - \gamma \mathbf{B} \mathbf{B}^T \mathbf{z}_d)_{CA}, \end{aligned} \quad (24)$$

in which \mathbf{e}_{CA} represents the estimated residual in the CA. Second, \mathbf{t} obtained from Eqs. (23) and 24 may still be sub-optimal due to the incomplete assumption made in Eq. (22) and the uncorrectable high-frequency components and noise in \mathbf{z}_d . Therefore, the RISE concept [7] is employed to further refine the dwell time solution iteratively. In the k th ($k \geq 1$) iteration, the RISE-based surface extension is applied to \mathbf{e}_{CA} to obtain the desired removal \mathbf{z}_d^k , from which $\Delta \mathbf{t}^k$ is calculated using Eqs. (23) and 24 as

$$\Delta \mathbf{t}^k = \gamma_{opt}^k \mathbf{B}^T \mathbf{z}_d^k, \quad (25)$$

which is then used to update the k th dwell time solution \mathbf{t}^k as

$$\mathbf{t}^k = \mathbf{t}^{k-1} + \Delta \mathbf{t}^{k-1}, \quad k \geq 1, \quad (26)$$

until the desired \mathbf{e}_{CA} is achieved.

3.3. Reduction of the total dwell time

By applying Eqs. (25) and 26, the final dwell time solution \mathbf{t} that produces the desired fabrication residual is obtained. If a CCOS machine can remove and add materials simultaneously, \mathbf{t} can contain either positive or negative entries. However, since CCOS processes are only capable of removing material, \mathbf{t} is expected to be non-negative. Nonetheless, this non-negativity is not necessary for the intermediate dwell time results, *i.e.* $\Delta \mathbf{t}^k$ and \mathbf{t}^k . Based on this point, the total

dwelling time reduction scheme is further proposed to eliminate the unnecessary increase of the total dwelling time during the iterations.

In detail, letting $\hat{\mathbf{z}}_d^k = \mathbf{z}_d^k - \mu_{\mathbf{z}_d^k}$, where $\mu_{\mathbf{z}_d^k}$ is the mean of \mathbf{z}_d^k . Equation 25 is rewritten as

$$\Delta \mathbf{t}^k = \gamma_{opt}^k \mathbf{B}^T (\hat{\mathbf{z}}_d^k + \mu_{\mathbf{z}_d^k}) = \gamma_{opt}^k (\mathbf{B}^T \hat{\mathbf{z}}_d^k + \mathbf{B}^T \mu_{\mathbf{z}_d^k}), \quad (27)$$

where $\mu_{\mathbf{z}_d^k} = [\mu_{\mathbf{z}_d^k}, \mu_{\mathbf{z}_d^k}, \dots, \mu_{\mathbf{z}_d^k}]^T \in \mathbb{R}^{N_z}$. It is found in Eq. (27) that the distribution of $\Delta \mathbf{t}$ described by $\mathbf{B}^T \hat{\mathbf{z}}_d^k$ is the same with that described by $\mathbf{B}^T \mathbf{z}_d^k$, while $\mathbf{B}^T \mu_{\mathbf{z}_d^k}$ only introduces constant dwelling time to each entry in $\Delta \mathbf{t}^k$ except the ones along the boundary, which is unnecessary and can be removed. Therefore, \mathbf{z}_d^k in Eq. (25) is substituted by its zero-mean version, *i.e.* $\hat{\mathbf{z}}_d^k$, in calculating $\Delta \mathbf{t}^k$ as

$$\Delta \mathbf{t}^k = \gamma_{opt}^k \mathbf{B}^T \hat{\mathbf{z}}_d^k, \quad (28)$$

where γ_{opt}^k is obtained by modifying Eq. (24) as

$$\begin{aligned} \gamma_{opt}^k &= \underset{\gamma^k}{\operatorname{argmin}} \operatorname{RMS}(\mathbf{e}_{CA}^k) \\ &= \underset{\gamma^k}{\operatorname{argmin}} \operatorname{RMS}(\mathbf{z}_d^k - \gamma^k \mathbf{B} \mathbf{B}^T \hat{\mathbf{z}}_d^k)_{CA}. \end{aligned} \quad (29)$$

By doing so, \mathbf{t}^k has an equal possibility of positive and negative adjustment, and thus reduces the increase of the total dwelling time during the iterations. To keep the non-negativity of the final \mathbf{t} , a piston adjustment is first performed before calculating $\Delta \mathbf{t}^0$ as

$$\Delta \mathbf{t}^0 = \gamma_{opt}^0 [\mathbf{B}^T \hat{\mathbf{z}}_d^0 - \min(\mathbf{B}^T \hat{\mathbf{z}}_d^0)], \quad (30)$$

where $\min(\cdot)$ takes the minimum value in ". If \mathbf{t} still contains negative entries after the iterations, their magnitudes are always small and can thus be safely offset to be non-negative.

To solve Eq. (29), a simple direct search algorithm [39] can be employed, since the scalar γ^k is the only unknown and it is easy to optimize. A good initial guess for γ^k can be chosen as the least-squares solution of $\frac{1}{2} \|\mathbf{z}_d^k - \gamma \mathbf{B} \mathbf{B}^T \hat{\mathbf{z}}_d^k\|_2^2$ as

$$\gamma_{ini}^k = \left[(\mathbf{B} \mathbf{B}^T \hat{\mathbf{z}}_d^k)^T \mathbf{z}_d^k \right] / \left[(\mathbf{B} \mathbf{B}^T \hat{\mathbf{z}}_d^k)^T (\mathbf{B} \mathbf{B}^T \hat{\mathbf{z}}_d^k) \right]. \quad (31)$$

3.4. Flow of UDO

The flow of the proposed UDO algorithm is schematically illustrated in Fig. 3, where Eqs. 26, 28, and 29 are included as the main steps of the algorithm. We divide the entire flow of UDO in two phases, including the precomputation phase and the iterative scalar optimization phase. First, since the convolution matrix \mathbf{B} remain invariant, it is precomputed in Steps 1 ~ 4. The coordinates of the dwelling points, *i.e.* (ξ_i, η_i) , can be arbitrary. They can be either on a spiral path or on a rectangular grid. The coordinates of \mathbf{z}_d , *i.e.* (x_j, y_j) , should be at least larger than the outline perimeter of the CA by the radius of $b(x, y)$. In the case that certain entries of \mathbf{B} , *i.e.* $b(x_j - \xi_i, y_j - \eta_i)$, may reside at a position where $b(x, y)$ has no corresponding values, an interpolation Look-Up Table (LUT) is constructed to interpolate these entries.

In UDO, the desired removal in the CA, *i.e.* $z_{CA}(x, y)$, is first extended with RISE to generate \mathbf{z}_d^0 . Note that this step can be omitted if \mathbf{z}_d^0 is directly measured. The iterative scalar optimization process described in Eqs. (28), 29 and 26 is then performed. In each iteration, the estimated residual in the CA, \mathbf{e}_{CA}^k , is calculated. If \mathbf{e}_{CA}^k satisfies the convergence criterion,

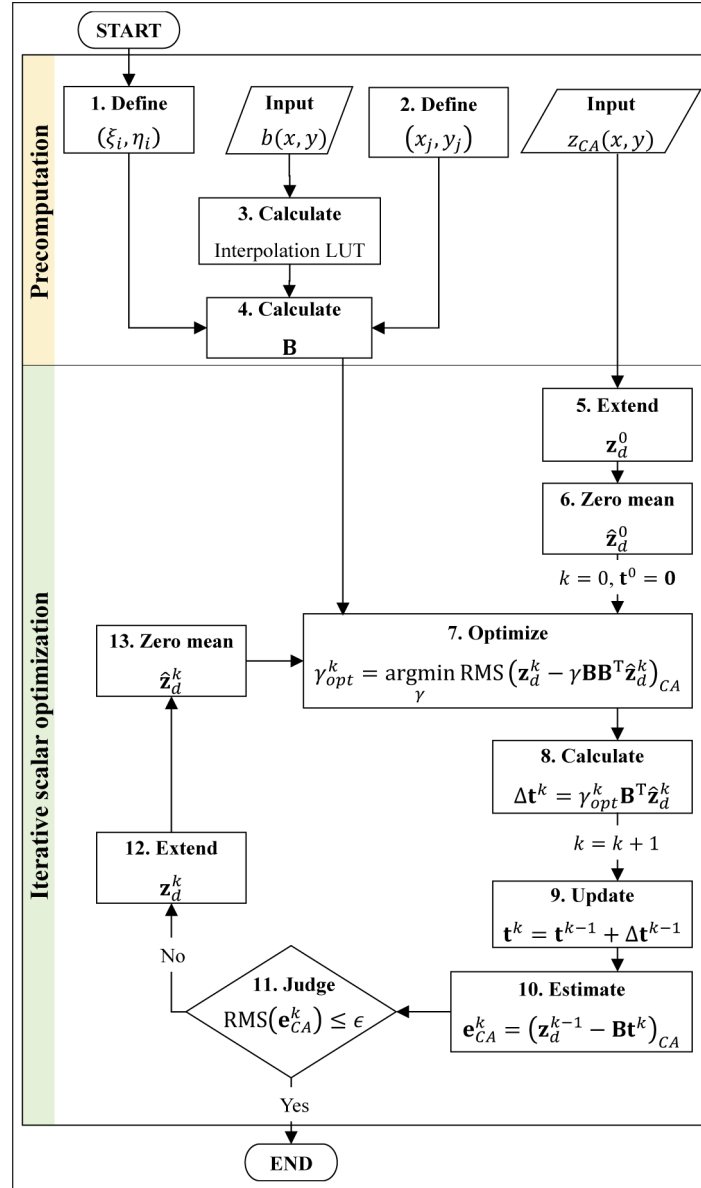


Fig. 3. Flowchart of the proposed UDO algorithm.

i.e. $\text{RMS}(\mathbf{e}_{CA}^k) \leq \epsilon$, the iterative refinement ends. Otherwise, \mathbf{e}_{CA}^k is extended to \mathbf{z}_d^k with RISE and the same operations in Steps 7 ~ 11 are repeated. The convergence threshold ϵ can be set according to the repeatability of the metrology instrument [7] or the high-frequency components of the surface.

The proposed UDO algorithm satisfies the six criteria of a desirable dwell time solution mentioned in Table 1 in the following manner.

1. $\mathbf{t} \geq 0$ is achieved by intelligently offsetting its negative entries as mentioned in Section 3.3.
2. \mathbf{e}_{CA} is minimized, since it is the objective of the scalar optimization problem.
3. $\sum \mathbf{t}$ is reduced by the total dwell time reduction scheme described in Section 3.3.
4. The flexible dwell points are reserved thanks to the matrix-based discretization of the convolutional polishing model.
5. No manual parameter tuning is required. The only unknown γ^k is obtained via the scalar optimization.
6. The most computationally expensive and memory consuming operation in UDO, as will be illustrated in Section 4, is the construction of the convolution matrix \mathbf{B} . This bottleneck is eliminated by precomputing \mathbf{B} .

4. Verification of UDO by simulations

The superiority and general applicability of the proposed UDO algorithm are verified with three simulations. In Simulation 1, the superiority of the UDO model over the conventional matrix-based algorithm is verified by comparing it with the LSQR algorithm. In the next two simulations, the general applicability of UDO is studied by applying it to calculate the dwell time for two different CCOS processes with different TIFs and tool paths. Simulation 2 is conducted on a ZEEKO IRP200 rubber ball-end polishing machine with a raster tool path (Fig. 4(a)), while Simulation 3 is for the OptoTech MCP250 bonnet polishing machine [14] with an equal-arc-length spiral tool path (Fig. 4(b)). All the simulations are run until \mathbf{e}_{CA} cannot be improved further.

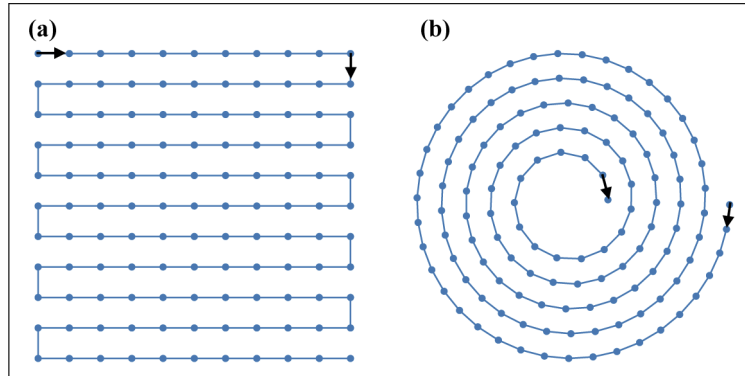


Fig. 4. Schematics of a raster tool path (a) and an equal-arc-length spiral tool path (b).

4.1. Simulation 1: verification of the UDO model with IBF

To study the effectiveness of the basic assumption on \mathbf{t} described in Section 3, a simulation based on the same experimental data shown in Section 5 is included, in which the figure error of a

rectangular synchrotron X-ray mirror needs to be improved by IBF. The TIF and the desired removal in the CA ($92.3 \text{ mm} \times 15.7 \text{ mm}$), z_{CA} used in the simulation are shown in Figs. 5(a) and 5(b), respectively. The resolution of z_{CA} is 0.09 mm/pixel , and the number of elements in $z_d(x, y)$ is $N_z = 298934$. A raster tool path is used, and the sampling interval between each two consecutive dwell points is 1 mm . The total number of the dwell points is 2899. More details of the experiment are postponed until Section 5.

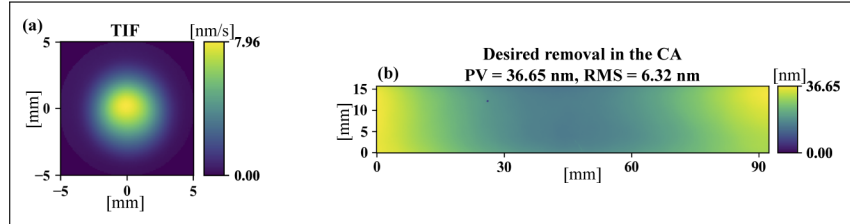


Fig. 5. The TIF (a) and the desired removal in the CA (b) for Simulation 1.

RISE, as the state-of-the-art, non-matrix-based algorithm, is included in the simulation as a reference. To make the comparison fair, the RISE solution is calculated on the 0.09 mm/pixel grid then resampled to the 1 mm interval. For the matrix-based algorithms, only LSQR optimized by the strategies mentioned in [31] is studied, since the TSVD and the CLLS algorithms are too computationally expensive and memory consuming to be included in this simulation. We would like to see if UDO can achieve the same performance as RISE on the simplest raster tool path with regular sampling intervals.

Figure 6(a) shows \mathbf{t} calculated from LSQR [27,28] with a manually tuned damping factor ζ proposed in [31] to guarantee the non-negativity of \mathbf{t} . Without damping, the corresponding \mathbf{e}_{CA} is shown in Fig. 6(b). As the estimation suggests, $\text{RMS}(\mathbf{e}_{CA})$ is reduced from 6.32 nm to 0.21 nm in 6.41 min . However, \mathbf{t} calculated from LSQR in Fig. 6(a) contains higher-frequency components than the desired removal map in the CA (especially along the boundary) shown in Fig. 5(b). As illustrated in Section 3.1, even though these high-frequency details in \mathbf{t} are mathematically correct, they are undesired in a practical CCOS process. The high-frequencies can be more clearly seen in Fig. 6(d), which plots the center profile of Fig. 6(a) in the tangential direction. In fact, these high-frequency components are a result of the objective of minimizing $\|\mathbf{t}\|$ in LSQR, which does not include any constraints on the smoothness of \mathbf{t} .

The RISE estimation, as shown in Figs. 6(c) and 6(d), achieves the similar estimated residual in the CA in shorter total dwell time than that obtained from LSQR thanks to the total dwell time minimization scheme employed in RISE [7]. Also, the dwell time map smoothly duplicate the shape of the desired removal map in Fig. 5(b), since RISE adds the smoothness constraints by only considering the correctable figure errors in its iterative refinement of the dwell time solution.

As proposed in Section 3.2, UDO adds the simple, yet effective, smoothness constraints to \mathbf{t} by forcing it to resemble the shape of $\mathbf{B}^T \mathbf{z}_d$. As shown in Fig. 6(e), \mathbf{t} is first optimized from Eqs. (24), 25, and 26 without using the total dwell time reduction scheme, and the corresponding \mathbf{e}_{CA} is shown in Fig. 6(f). It is found that UDO has achieved the same level of \mathbf{e}_{CA} as that calculated from LSQR, while the dwell time map, as shown in Figs. 6(e) and 6(i), is smoother compared with that obtained from LSQR. This result verifies that the smoothness assumption in UDO is appropriate and the iterative scalar optimization is effective. However, the total dwell time (6.55 min) is slightly higher than that calculated from LSQR due to unnecessary the dwell time as introduced in Section 3.3.

With the total dwell time reduction included, \mathbf{t} optimized by UDO and the corresponding \mathbf{e}_{CA} are demonstrated in Figs. 6(g) and 6(h), respectively. The total dwell time is reduced to 3.37 min while the same level of \mathbf{e}_{CA} is remained. Moreover, as shown in Fig. 6(i), the smoothness of \mathbf{t} is

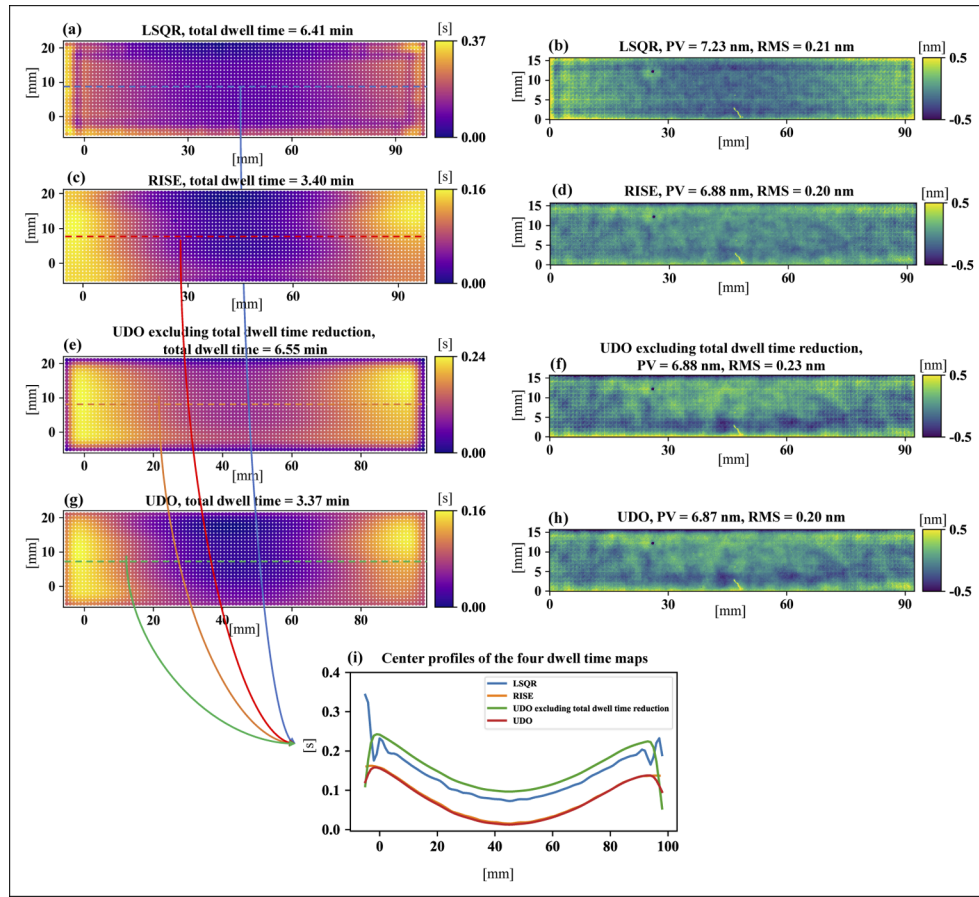


Fig. 6. Simulation 1: Dwell time solutions are calculated with LSQR (a), RISE (c), UDO excluding the total dwell time reduction scheme (e), and UDO (g). The corresponding estimated residual in the CA are given in (b), (d), (f), and (h), respectively. The profiles of the center lines of the dwell time solutions are compared in (i).

not affected. These results prove that UDO is superior to the LSQR algorithm, and it achieves the performance comparable to the state-of-the-art RISE algorithm in terms of the minimized total dwell time and the estimated residual in the CA. It is worth noting that, however, RISE is only applicable to tool paths defined on regular grids, since the bicubic resampling can hardly be adapted to non-uniform dwell points. For dwell points on spiral tool paths or even random tool paths, as will be shown in Section 4.3, only the matrix-based methods can be directly applied to obtain the dwell time.

4.2. Simulation 2: UDO applied to a ZEEKO IRP200 polishing machine with an asymmetric TIF

To further study the general applicability of UDO to different CCOS processes, it is first applied to a ZEEKO IRP200 rubber ball-end polishing machine in Simulation 2. The measured TIF for the machine and the desired removal map in the CA are shown in Fig. 7(a) and 7(b), respectively. The radius of the TIF is 3 mm. However, it is not axially-symmetric since the polishing head is set to be non perpendicular to the surface to avoid the zero central removal effect. Thus, the Peak Removal Rate (PRR = 149.9 nm/s) is not obtained at the center of the TIF. The CA is a 35

mm \times 35 mm square. The resolution of the desired removal map in the CA is 0.21 mm/pixel, and the total number of elements in $z_d(x, y)$ is $N_z = 38025$. The raster tool path is used, and the sampling interval between each two consecutive dwell points on the path is set to 0.5 mm. The total number of dwell points is $N_t = 6889$. The final dwell time \mathbf{t} and the corresponding \mathbf{e}_{CA} optimized based on Figs. 7(a) and 7(b) using UDO are demonstrated in Figs. 7(c) and 7(d). As the estimation suggests, RMS(\mathbf{e}_{CA}) is reduced from 1174.1 nm to 24.4 nm in 7 iterations, achieving a convergence ratio of 97.9% in 58.4 min.

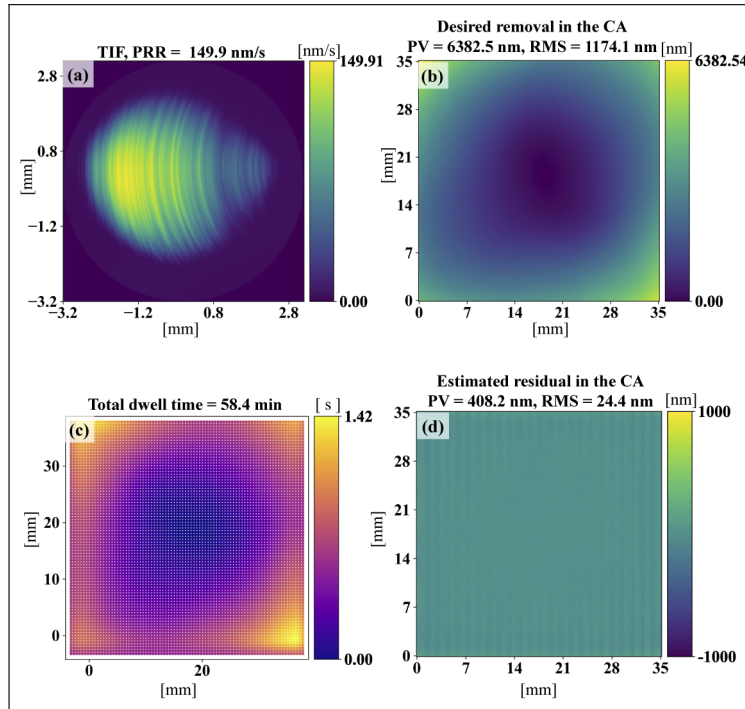


Fig. 7. Simulation 2: the TIF (a) and desired removal map in the CA (b), from which the dwell time (c) and the estimated residual in the CA (d) is optimized by UDO.

4.3. Simulation 3: UDO applied to an OptoTech MCP250 polishing machine with a w-shaped TIF

The TIF and the desired removal map in the CA for Simulation 3 are given in Figs. 8(a) and 8(b), respectively. The TIF of the MCP250 is axially-symmetric with an effective radius of 2.5 mm and the PRR is 47.3 nm/s. The CA is a ring with the outer radius of 80 mm and the inner radius of 20 mm. The resolution of the desired removal map in the CA is 1 mm/pixel, and the number of valid elements in $z_d(x, y)$ is $N_z = 24313$. The equal-arc-length spiral path is employed in this simulation, in which case the non-matrix-based methods, e.g. RISE, cannot be applied. Both the distance between each ring on the spiral path and the arc length between each two consecutive dwell points are set to 1 mm. The total number of dwell points on the spiral path is $N_t = 24055$. The final \mathbf{t} and the corresponding \mathbf{e}_{CA} optimized based on Figs. 8(a) and 8(b) are demonstrated in Figs. 8(c) and 8(d), respectively. As the estimation suggests, RMS(\mathbf{e}_{CA}) is reduced from 67.7 nm to 8.2 nm in 9 iterations, achieving a convergence ratio of 87.9% in 213.9 min. The high-frequency components on the \mathbf{e}_{CA} seen in Fig. 8(d) are tool marks from previous fabrication processes. A smaller TIF or a smoothing process is required to remove these tool marks.

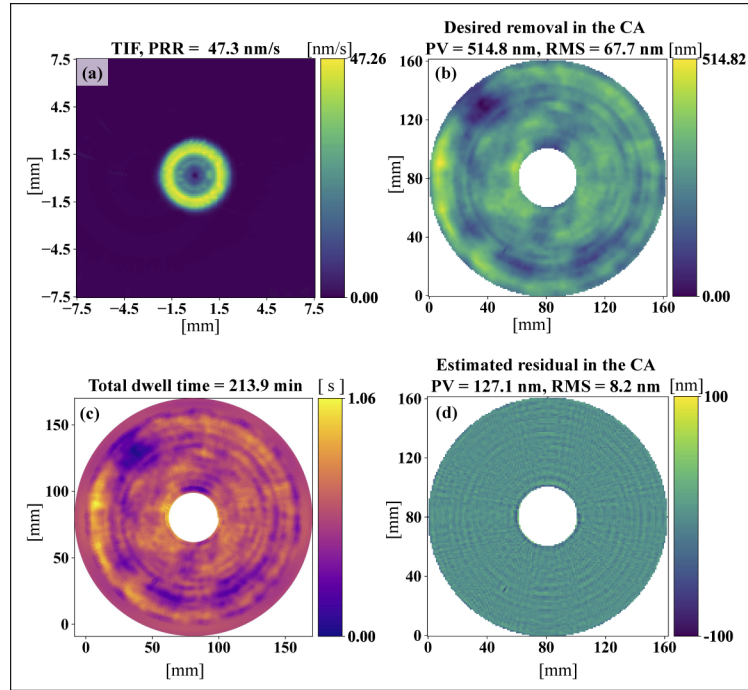


Fig. 8. Simulation 3: the TIF (a) and desired removal map in the CA (b), from which the dwell time (c) and the estimated residual in the CA (d) is optimized by UDO.

4.4. Computational efficiency of UDO

The simulation results obtained in Sections 4.1, 4.2 and 4.3 have proved that UDO is an effective dwell time optimization algorithm for various CCOS processes. The computational efficiency of applying UDO to the three simulations is further studied in this subsection. The proposed UDO algorithm is programmed in Python 3.8, and tested on a workstation equipped with an Intel Xeon Gold 5118 CPU (2 processors, 12 cores/processor, 2.30 GHz main frequency) and 64 GB RAM. Table 2 summarizes the computational statistics of the three simulations.

Table 2. Computational statistics for the three simulations, where the Convergence Ratio refers to the ratio between the RMSs of the actual removal and the desired removal.

#	N_z	N_t	Total [s]	Precomputation [s]	Average [s/Iter.]	Iteration #	Convergence Ratio
1	298,934	2,899	73.9	60.5	6.7	2	96.8%
2	38,025	6,889	47.2	14.9	4.6	7	97.9%
3	24,313	24,055	134.4	40.0	10.5	9	87.9%

The computation time for each simulation is obtained by running the program five times and taking the average. It is found that larger N_z and N_t result in longer computation time. The most time-consuming operation in UDO is the precomputation of the convolution matrix **B**. It dominates the total computation time (30% ~ 81%) while the average computation time per iteration is much shorter. Also, since only several iterations are required to minimize \mathbf{e}_{CA} , the iterative scalar optimization in UDO is thus a computationally efficient process.

The PV and RMS evolution versus the computation time for Simulations 2 and 3 are further illustrated in Figs. 9(a) and 9(b), respectively. The most significant improvements on \mathbf{e}_{CA} happen in the first iteration, which prove the appropriateness of our basic smoothness assumption on

t in UDO proposed in Section 3.2. Moreover, the PVs and RMSs of e_{CA} in both simulations keep decreasing as the numbers of iterations increase, which verifies the effectiveness of the iterative refinement process. Last, the computation time for each iteration is nearly constant, which indicates that the direct search algorithm used to solve the scalar optimization per iteration is stable.

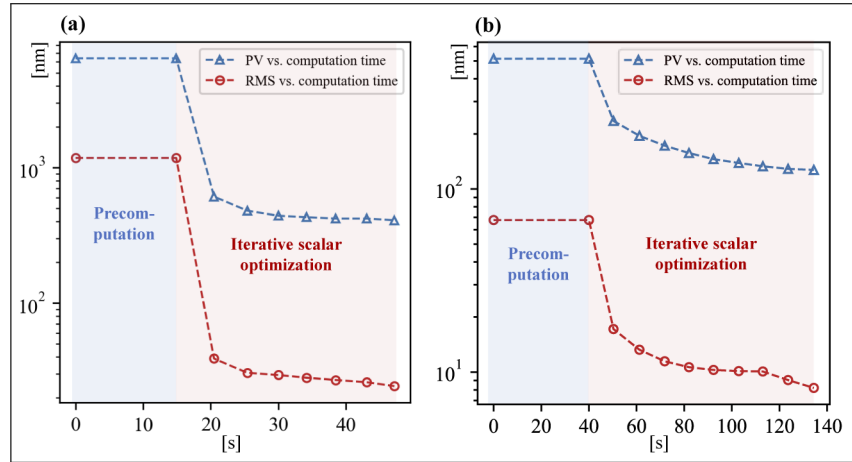


Fig. 9. The PV and RMS evolution vs. computation time of Simulation 2 (a) and Simulation 3 (b).

It is worth mentioning that \mathbf{B} may be so large that a dense representation of it may not fit the limited RAM. Since \mathbf{B} is sparse, a sparse representation of it can be thus further exploited to save memory if the surface scale is even larger or the dwell points are denser.

5. Application of UDO

As shown in Fig. 10, UDO has been applied to a collaborated project where the task was to improve the existing silicon flat synchrotron mirrors for the CHX beamline using our in-house developed IBF system [21] demonstrated in Fig. 10(f). The rectangular shaped mirror (92.3 mm \times 15.7 mm), as shown in Fig. 10(c), already exhibits with excellent roughness. Therefore, our main goal was to improve the flatness of the mirror without affecting the roughness.

The CA in the experiment is defined to be the entire mirror surface. The desired removal in the CA to a perfect flat surface, as shown in Fig. 10(b), was measured with the in-house developed Stitching Interferometry (SI) platform [40,41] at NSLS-II with the lateral resolution of 0.09 mm/pixel. The ion source parameters used in the experiment were beam voltage = 600 V, beam current = 10 mA, accelerator voltage = -90 V, and accelerator current = 2 mA.

The TIF of IBF given in Fig. 10(a) was obtained by bombarding several ion beam footprints on a silicon substrate for 30 s and taking the average of them. The radius of the TIF is 5 mm, and the PRR is 7.96 nm/s. A rectangular raster tool path was used, and the distance between each two consecutive dwell points was set to 1 mm. Since IBF does not have the overhang issue at the edges of the mirror, the outline perimeter of the tool path is defined to be larger than the mirror by the radius of the TIF, *i.e.* 5 mm.

The dwell time optimization process with UDO is given in Figs. 10(d) and 10(e). As the estimation suggests, the figure error in the CA could be improved from 6.32 nm RMS to 0.20 nm RMS (about a factor of 32) in 3.37 min after two iterations of UDO. This dwell time solution was then sent to the IBF system. As shown in Fig. 10(g), after one IBF process, the figure error in the

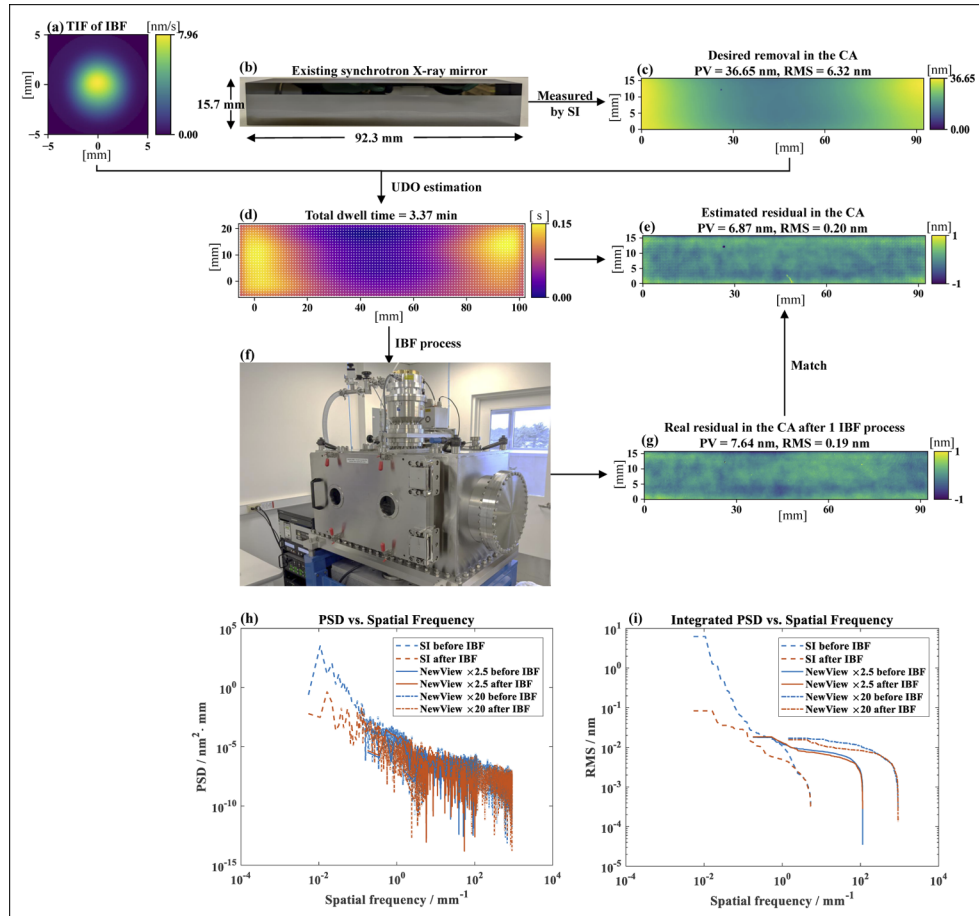


Fig. 10. Improving the figure error of an existing synchrotron mirror (b) with the proposed UDO algorithm and the IBF system (f): based on the TIF of IBF (a) and the desired removal in the CA (c) measured using the SI platform, the dwell time solution (d) and the corresponding estimated residual in the CA (e) are optimized by UDO. The measured residual in the CA after one IBF run (g) coincides with the estimation. The PSD (h) and integrated PSD (i) curves along the tangential direction prove that the surface roughness is not affected by the IBF process.

CA was successfully reduced from 6.32 nm RMS to 0.19 nm RMS, which coincides with the estimation given in Fig. 10(e).

The Power Spectral Density (PSD) curves and their corresponding integrated PSD RMS curves along the tangential direction of the CA before and after the IBF process are shown in Figs. 10(h) and 10(i), respectively. The SI measurements demonstrate the contribution of the proposed UDO algorithm. In addition, the middle to high frequency surface roughness before and after the IBF process was measured by a Zygo NewView white-light interferometer with 2.5× and 20× magnifications, respectively. The spatial resolution of the 2.5× measurement was 4.38 μm and the Field Of View (FOV) was 2.1 mm \times 2.8 mm. For the 20× measurement, the spatial resolution was 0.55 μm and the FOV was 0.26 mm \times 0.35 mm. It can be seen that the surface roughness, at the explored resolution levels, is not affected by the IBF process and still remain at an excellent level for synchrotron X-ray mirrors.

The experimental results demonstrated above prove that the proposed UDO is applicable even to the sub-nanometer level fabrication processes and our IBF system is highly deterministic.

6. Concluding remarks

In this study, a simple yet effective Universal Dwell time Optimization (UDO) algorithm was proposed to obtain desirable dwell time solutions for various computer controlled optical surfacing applications with different tool influence functions and tool paths. The optimized dwell time satisfies the criteria of a desirable dwell time in the following manner.

First, it is non-negative. Second, the estimated residual in the CA is minimized by the iterative scalar optimization. Third, in each iteration, the increase of the total dwell time is reduced by the total dwell time reduction scheme. Fourth, the matrix-based discretization of the convolutional polishing model is preserved to ensure the flexible dwell positions. Fifth, the UDO algorithm is fully automatic. No manual parameter tuning is required. Last, the computational complexity of UDO is low thanks to the precomputation of the convolution matrix. The superiority and computational efficiency of UDO have been verified on the simulations by applying it to different CCOS processes. The real experiment of successfully improving the figure quality of a synchrotron X-ray mirror with ion beam figuring has even proved the applicability of UDO to the extreme sub-nanometer level optical fabrication tasks.

It is also worth mentioning that UDO is a very computationally efficient algorithm which has simplified the deconvolution process by eliminating many unnecessary operations. Due to these advances, UDO is an excellent candidate to improve the performance of the more computationally complicated non-sequential dwell time optimization for different tool sizes. [13]. We are currently working on this aspect, and will report our progress in the near future.

Funding. Brookhaven National Laboratory (BNL LDRD 17-016); Office of Science (DE-SC0012704).

Disclosures. The authors declare no conflicts of interest.

Data availability. Data underlying the results presented in this paper are not publicly available at this time but may be obtained from the authors upon reasonable request.

References

1. J. Fanson, R. Bernstein, G. Angeli, D. Ashby, B. Bigelow, G. Brossus, A. Bouchez, W. Burgett, A. Contos, R. Demers, F. Figueroa, B. Fischer, F. Groark, R. Laskin, R. Millan-Gabet, M. Pi, and N. Wheeler, "Overview and status of the giant magellan telescope project," in *Ground-based and Airborne Telescopes VIII*, vol. 11445 (International Society for Optics and Photonics, 2020), p. 114451F.
2. M. Ghigo, G. Vecchi, S. Basso, O. Citterio, M. Civitani, E. Mattaini, G. Pareschi, and G. Sironi, "Ion figuring of large prototype mirror segments for the E-ELT," in *Advances in Optical and Mechanical Technologies for Telescopes and Instrumentation*, vol. 9151 R. Navarro, C. R. Cunningham, and A. A. Barto, eds., International Society for Optics and Photonics (SPIE, 2014), pp. 225–236.
3. A. Beaucamp and Y. Namba, "Super-smooth finishing of diamond turned hard x-ray molding dies by combined fluid jet and bonnet polishing," *CIRP Ann.* **62**(1), 315–318 (2013).

4. L. Peverini, I. Kozhevnikov, A. Rommeveaux, P. Vaerenbergh, L. Claustre, S. Guillet, J.-Y. Massonnat, E. Ziegler, and J. Susini, "Ion beam profiling of aspherical x-ray mirrors," *Nucl. Instrum. Methods Phys. Res., Sect. A* **616**(2-3), 115–118 (2010). X-Ray Mirror.
5. A. Schindler, T. Haensel, A. Nickel, H.-J. Thomas, H. Lammert, and F. Siewert, "Finishing procedure for high-performance synchrotron optics," in *Optical Manufacturing and Testing V*, vol. 5180 (International Society for Optics and Photonics, 2003), pp. 64–72.
6. H. Thiess, H. Lasser, and F. Siewert, "Fabrication of x-ray mirrors for synchrotron applications," *Nucl. Instrum. Methods Phys. Res., Sect. A* **616**(2-3), 157–161 (2010).
7. T. Wang, L. Huang, H. Choi, M. Vescovi, D. Kuhne, Y. Zhu, W. C. Pullen, X. Ke, D. W. Kim, Q. Kema, K. Tayabaly, N. Bouet, and M. Idir, "RISE: robust iterative surface extension for sub-nanometer x-ray mirror fabrication," *Opt. Express* **29**(10), 15114–15132 (2021).
8. T. Wang, L. Huang, H. Kang, H. Choi, D. W. Kim, K. Tayabaly, and M. Idir, "RIFTA: A robust iterative fourier transform-based dwell time algorithm for ultra-precision ion beam figuring of synchrotron mirrors," *Sci. Rep.* **10**(1), 1–12 (2020).
9. M. Weiser, "Ion beam figuring for lithography optics," *Nucl. Instrum. Methods Phys. Res., Sect. B* **267**(8-9), 1390–1393 (2009).
10. L. Wischmeier, P. Graeupner, P. Kuerz, W. Kaiser, J. van Schoot, J. Mallmann, J. de Pee, and J. Stoeldraijer, "High-na euv lithography optics becomes reality," in *Extreme Ultraviolet (EUV) Lithography XI*, vol. 11323 (International Society for Optics and Photonics, 2020), p. 1132308.
11. H. Cheng, *Independent variables for optical surfacing systems* (Springer, 2016).
12. W. Zhu and A. Beaucamp, "Compliant grinding and polishing: A review," *Int. J. Mach. Tools Manuf.* **158**, 103634 (2020).
13. D. W. Kim, S.-W. Kim, and J. H. Burge, "Non-sequential optimization technique for a computer controlled optical surfacing process using multiple tool influence functions," *Opt. Express* **17**(24), 21850–21866 (2009).
14. V. S. Negi, H. Garg, S. K. Rr, V. Karar, U. K. Tiwari, and D. W. Kim, "Parametric removal rate survey study and numerical modeling for deterministic optics manufacturing," *Opt. Express* **28**(18), 26733–26749 (2020).
15. C. Wang, Z. Wang, Q. Wang, X. Ke, B. Zhong, Y. Guo, and Q. Xu, "Improved semirigid bonnet tool for high-efficiency polishing on large aspheric optics," *Int. J. Adv. Manuf. Technol.* **88**(5-8), 1607–1617 (2017).
16. D. Golini, S. D. Jacobs, W. I. Kordonski, and P. Dumas, "Precision optics fabrication using magnetorheological finishing," in *Advanced Materials for Optics and Precision Structures: A Critical Review*, vol. 10289 (International Society for Optics and Photonics, 1997), p. 102890H.
17. S. Wan, C. Wei, C. Hu, G. Situ, Y. Shao, and J. Shao, "Novel magic angle-step state and mechanism for restraining the path ripple of magnetorheological finishing," *Int. J. Mach. Tools Manuf.* **161**, 103673 (2021).
18. M. Demmler, M. Zeuner, F. Allenstein, T. Dunger, M. Nestler, and S. Kiontke, "Ion beam figuring (IBF) for high precision optics," *Proc. SPIE* **7591**, 75910Y (2010).
19. M. Idir, L. Huang, N. Bouet, K. Kaznatcheev, M. Vescovi, K. Lauer, R. Conley, K. Rennie, J. Kahn, R. Nethery, and L. Zhou, "A one-dimensional ion beam figuring system for x-ray mirror fabrication," *Rev. Sci. Instrum.* **86**(10), 105120 (2015).
20. T. Wang, L. Huang, M. Vescovi, D. Kuhne, K. Tayabaly, N. Bouet, and M. Idir, "Study on an effective one-dimensional ion-beam figuring method," *Opt. Express* **27**(11), 15368–15381 (2019).
21. T. Wang, L. Huang, Y. Zhu, M. Vescovi, D. Khune, H. Kang, H. Choi, D. W. Kim, K. Tayabaly, N. Bouet, and M. Idir, "Development of a position-velocity-time-modulated two-dimensional ion beam figuring system for synchrotron x-ray mirror fabrication," *Appl. Opt.* **59**(11), 3306–3314 (2020).
22. L. Zhou, L. Huang, N. Bouet, K. Kaznatcheev, M. Vescovi, Y. Dai, S. Li, and M. Idir, "New figuring model based on surface slope profile for grazing-incidence reflective optics," *J. Synchrotron Radiat.* **23**(5), 1087–1090 (2016).
23. L. Zhou, M. Idir, N. Bouet, K. Kaznatcheev, L. Huang, M. Vescovi, Y. Dai, and S. Li, "One-dimensional ion-beam figuring for grazing-incidence reflective optics," *J. Synchrotron Radiat.* **23**(1), 182–186 (2016).
24. S. Wilson and J. McNeil, "Neutral ion beam figuring of large optical surfaces," in *Current Developments in Optical Engineering II*, vol. 818 (International Society for Optics and Photonics, 1987), pp. 320–324.
25. R. A. Jones, "Optimization of computer controlled polishing," *Appl. Opt.* **16**(1), 218–224 (1977).
26. C. Wang, W. Yang, Z. Wang, X. Yang, C. Hu, B. Zhong, Y. Guo, and Q. Xu, "Dwell-time algorithm for polishing large optics," *Appl. Opt.* **53**(21), 4752–4760 (2014).
27. C. Jiao, S. Li, and X. Xie, "Algorithm for ion beam figuring of low-gradient mirrors," *Appl. Opt.* **48**(21), 4090–4096 (2009).
28. C. L. Carnal, C. M. Egert, and K. W. Hylton, "Advanced matrix-based algorithm for ion-beam milling of optical components," in *Current Developments in Optical Design and Optical Engineering II*, vol. 1752 (International Society for Optics and Photonics, 1992), pp. 54–62.
29. J. F. Wu, Z. W. Lu, H. X. Zhang, and T. S. Wang, "Dwell time algorithm in ion beam figuring," *Appl. Opt.* **48**(20), 3930–3937 (2009).
30. L. Zhou, Y.-f. Dai, X.-h. Xie, C.-j. Jiao, and S.-y. Li, "Model and method to determine dwell time in ion beam figuring," *Nanotechnol. Precis. Eng.* **5**(2), 107–112 (2007).
31. Z. Dong, H. Cheng, and H.-Y. Tam, "Robust linear equation dwell time model compatible with large scale discrete surface error matrix," *Appl. Opt.* **54**(10), 2747–2756 (2015).

32. T. Huang, D. Zhao, and Z.-C. Cao, "Trajectory planning of optical polishing based on optimized implementation of dwell time," *Precis. Eng.* **62**, 223–231 (2020).
33. Y. Li and L. Zhou, "Solution algorithm of dwell time in slope-based figuring model," in *AOPC 2017: Optoelectronics and Micro/Nano-Optics*, vol. 10460 (International Society for Optics and Photonics, 2017), p. 104601X.
34. Y. Zhang, F. Fang, W. Huang, and W. Fan, "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. Technol.* **8**(5), 1415–1427 (2021).
35. P. Van Cittert, "Zum einfluss der spaltbreite auf die intensitätsverteilung in spektrallinien. ii," *Z. Phys.* **69**(5-6), 298–308 (1931).
36. N. Hill and G. Ioup, "Convergence of the van cittert iterative method of deconvolution," *J. Opt. Soc. Am.* **66**(5), 487–489 (1976).
37. L. B. Lucy, "An iterative technique for the rectification of observed distributions," *The Astronomical J.* **79**, 745 (1974).
38. W. H. Richardson, "Bayesian-based iterative method of image restoration," *J. Opt. Soc. Am.* **62**(1), 55–59 (1972).
39. R. M. Lewis, V. Torczon, and M. W. Trosset, "Direct search methods: then and now," *J. computational Appl. Math.* **124**(1-2), 191–207 (2000).
40. L. Huang, T. Wang, J. Nicolas, A. Vivo, F. Polack, M. Thomasset, C. Zuo, K. Tayabaly, D. W. Kim, and M. Idir, "Two-dimensional stitching interferometry for self-calibration of high-order additive systematic errors," *Opt. Express* **27**(19), 26940–26956 (2019).
41. L. Huang, T. Wang, K. Tayabaly, D. Kuhne, W. Xu, W. Xu, M. Vescovi, and M. Idir, "Stitching interferometry for synchrotron mirror metrology at national synchrotron light source ii (nsls-ii)," *Opt. Lasers Eng.* **124**, 105795 (2020).