Journal of Manufacturing Processes 167 (2026) 507–531

Contents lists available at ScienceDirect

Journal of Manufacturing Processes

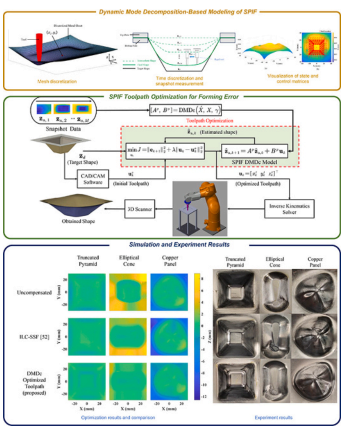
journal homepage: www.elsevier.com/locate/manpro

Dynamic mode decomposition-based modeling and toolpath optimization of a robotic single point incremental forming process

Arafat Asghar [a], Shuaiyin He [a], Pengcheng Hu [b], Molong Duan [a,c,*]

[a] Department of Mechanical and Aerospace Engineering, The Hong Kong University of Science and Technology, Hong Kong, China
[b] Thrust of Smart Manufacturing, The Hong Kong University of Science and Technology (Guangzhou), China
[c] HKUST Shenzhen-Hong Kong Collaborative Innovation Research Institute, Futian, Shenzhen, China

GRAPHICAL ABSTRACT



ARTICLE INFO

Keywords:
Incremental robotic forming
Data-driven modeling
Dynamic mode decomposition
Toolpath optimization
Springback compensation
Quadratic programming
Uncertainty
Experimental validation

ABSTRACT

Incremental sheet forming is a low-cost manufacturing technique that enables die-less and customized production of sheet metal components. Among its variants, Single Point Incremental Forming (SPIF) has the least complexity and cost with the highest formability but suffers from significant geometric inaccuracy due to springback, preventing its large-scale industrial adoption. Existing springback compensation approaches based on toolpath modification are computationally expensive or highly iterative, and are predominantly limited to convex geometries, constraining their utility across broader geometry classes and process windows. Additionally, the current SPIF models do not capture process dynamics as a direct function of the specimen shape and toolpath, thereby complicating systematic toolpath optimization. We propose a data-driven feedforward compensation framework that represents SPIF as a lifted linear state-space model identified via Dynamic Mode Decomposition with control (DMDc). The state encodes the measured specimen geometry, and the control input is the tool Cartesian position. This representation enables efficient toolpath updates via quadratic programming while maintaining an explicit link between predicted shape evolution and control actions. Experiments on 0.3 mm Al-1050-O sheets across three geometries, including a non-convex case, demonstrate improved geometric accuracy

* Corresponding author at: Department of Mechanical and Aerospace Engineering, The Hong Kong University of Science and Technology, Hong Kong, China.
E-mail address: duan@ust.hk (M. Duan).

https://doi.org/10.1016/j.jmapro.2026.03.017
Received 7 November 2025; Received in revised form 5 February 2026; Accepted 10 March 2026
Available online 11 April 2026
1526-6125/

compared with the Iterative Learning Control–Single Sheet Forming (ILC-SSF) benchmark. Across the three geometries, final-shape RMSE is reduced by 43.8% on average relative to the uncompensated toolpath and by 44.5% on average relative to ILC-SSF. Although model identification and preprocessing are performed offline, the resulting compensation is non-iterative during execution, providing a practical route to improve SPIF accuracy within the validated operating envelope.

1. Introduction

The use of sheet metal is expanding beyond traditional mass manufacturing, driven by demands for rapid prototyping, small-batch production, and complex free-form geometries. This has enabled its application in diverse fields, including aerospace, automotive, and biomedical engineering. [1]. Conventional sheet metal forming techniques are inflexible and cost-prohibitive for complex, low-volume parts. Incremental Sheet Forming (ISF) circumvents these issues through a die-less approach. A simple, rigid tool incrementally deforms the sheet along a computer-defined toolpath, enabling the cost-effective production of complex geometries and making ISF ideal for rapid prototyping and small-batch manufacturing. Consequently, ISF has emerged as a cost-effective and efficient technique for complex shapes and geometries, small batch production and rapid prototyping [2]. Nonetheless, several issues prevent the wide-scale industrial adoption of ISF, most notably geometric and dimensional inaccuracies due to springback, pillowing, and unwanted sheet bending [3,4]. The severity of these inaccuracies is influenced by a multitude of factors, including material properties, toolpath strategies, process parameters, and system dynamics.

To address the challenge of geometric inaccuracy, researchers have investigated various techniques which can be broadly grouped into three categories: (1) hardware modifications, (2) software-based process planning, and (3) integrated real-time control systems [5]. The hardware approach entails various modifications to the original ISF process and post-processing operations, such as introduction of a heating mechanism (known as heated-assisted ISF), partial or full dies (two-point ISF), additional tooling (double-sided ISF), stress-relief annealing and even hybridization of the above-mentioned techniques [6–8]. However, effective hardware modification is accompanied by an increase in cost and complexity of the ISF process. Software and integrated hardware-software approaches involve a multitude of techniques broadly categorized under process planning and control [9]. These strategies seek to improve accuracy through intelligent process sequencing and control logic, rather than permanent hardware alterations. A key example is multi-stage forming, which divides the forming process into intermediate steps or stages, instead of a single stage [10]. However, enhanced accuracy from multi-stage forming is offset by a significant increase in the overall forming time. Another prominent software-based approach uses the Finite-Element Method (FEM) to characterize the ISF process and determine optimal parameters [11,12]. To circumvent the high computational cost of FEM, there has been a growing focus on AI-based surrogate modeling to rapidly predict outcomes such as forming force, material formability, geometric accuracy, and to optimize toolpaths [13]. Despite promising results, researchers have yet to develop a generalizable and comprehensive AI-based model to encompass the entire spectrum of the ISF process.

Complementary to toolpath-centric compensation, recent SPIF/ISF research has emphasized material-behavior-aware modeling and numerical optimization to improve predictive reliability and process robustness, particularly for advanced and composite sheets. For example, coupled elastoplastic–damage formulations paired with desirability-based response surface methodology have been used to optimize SPIF parameter sets against multiple competing responses such as forming force, stress/strain measures, and achievable depth [14]. In situ characterization combined with fully coupled damage modeling has further clarified damage mechanisms and enabled more realistic simulations of composite incremental forming capability [15]. In parallel, mixed (isotropic–kinematic) hardening identification under monotonic and reverse shear has been shown to better capture path-change effects and to improve SPIF simulation outcomes compared to isotropic hardening alone [16]. Finally, RVE-based homogenization approaches have been used to identify effective elastoplastic properties across reinforcement fractions, supporting forming-process simulation and design for Fe–TiB_2 composites [17]. Collectively, these works broaden the state of the art beyond purely kinematic compensation and motivate future hybrid frameworks that combine constitutive fidelity with efficient toolpath update mechanisms. Lastly, the integrated real-time control systems represent a combined approach whereby, the toolpath is adjusted in real-time using shape and force feedback signals [18,19]. However, this integration increases the overall process cost and complexity, due to the requisite sensors, actuators, and software. Overall, these three existing compensation strategies often involve a significant trade-off between accuracy, cost, and complexity.

Within software-based approaches, toolpath planning and optimization stand out as a critical subset, directly influencing key process outcomes and offering a flexible means to enhance geometric accuracy without hardware overhaul. The toolpath is a critical factor in ISF, directly governing specimen thickness distribution, tool forces, bending moments, and effective plastic strain [20]. Toolpaths are typically generated using CAM software (e.g., CATIA, NX, Fusion360) or analytical equations. A key distinction lies in their implementation: while CNC machines execute G-code for high positional precision (at the cost of flexibility), robotic manipulators operate on Cartesian coordinates, offering greater adaptability for complex geometries despite challenges with stiffness and path planning. Consequently, the choice between platforms represents a fundamental tradeoff between cost, accuracy, and adaptability. Since standard CAM toolpaths are designed for milling, they require further optimization to maximize beneficial plastic deformation in ISF [21–23]. Research into toolpath planning aims to develop new algorithms to address these process limitations and improve geometric accuracy [24]. The literature reveals a wide spectrum of optimization frameworks, broadly categorized into iterative and non-iterative approaches, feature-based strategies, and multi-stage strategies. For instance, many strategies rely on iterative correction, such as the use of successive deviation vectors [25], experimental trial-and-error [26], mirroring of actual profiles with FEA [27], adaptive hypersurface optimization [28], and Response Surface Methodology (RSM) with Sequential Quadratic Programming [29]. These iterative approaches typically require multiple correction cycles, with FEA-based methods consuming several hours per iteration [30,31] and experimental methods necessitating multiple physical specimens resulting in extended development times and material waste [32]. Other approaches focus on geometric features, such as the Intelligent CAM (ICAM) strategy [33] or methods based on critical edges [34], while techniques like multi-directional forming [35] and two-stage compensation with counter toolpaths [36] have been developed to optimize sheet thickness distribution.

Despite the extensive efforts in toolpath compensation, significant limitations persist, underscoring the need for a new approach. Firstly, the algorithms described typically require several iterative adjustments to the toolpath, experimental conditions, and simulations before achieving a satisfactory result, making them computationally expensive. Secondly, validation has largely been confined to simple, regularly shaped geometries like truncated cones and pyramids, raising questions

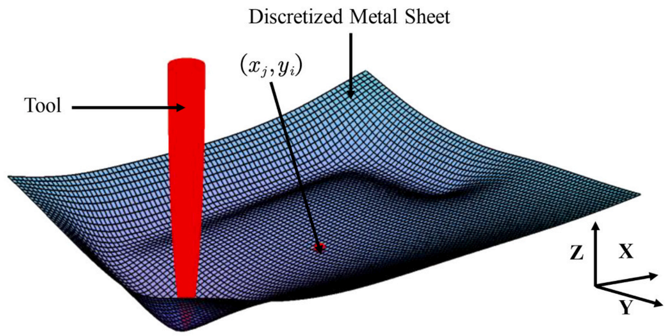


Fig. 1. Discretized mesh representation of metal sheet as a function of tool movement

about generalizability. Thirdly, most research has been conducted on expensive CNC machines [37–40], neglecting the more flexible and space-efficient robotic manipulators, which are better suited for hardware integration but introduce more complex dynamics [41]. Fourthly, despite substantial progress in constitutive modeling and numerical optimization for SPIF, there remains a lack of an explicit input–output dynamical model that directly relates measured shape evolution to the tool's position and motion in a form suitable for efficient shape prediction and toolpath optimization, particularly in robotic SPIF. This problem is further compounded by the fact that strong nonlinearity of the deformation mechanics makes derivation of a first-principles model difficult [42]. Finally, the FEM data has not been leveraged to create a practical input-output dynamic model for real-time prediction and compensation. This absence forces a reliance on feedback control, which despite its efficacy, adds cost and complexity. Physics-based FEM offers high fidelity but is computationally expensive for rapid compensation loops and systematic toolpath optimization in SPIF, especially when multiple iterations are required for each new part geometry or parameter set [11,12,30,31]. Machine-learning surrogates can improve efficiency but often lack an explicit input–output dynamical structure for multi-step prediction and optimization and may require extensive training data across the parameter space [13]. Taken together, these studies suggest that an effective compensation framework should provide (i) an explicit input–output relationship between tool motion and evolving geometry, (ii) computational efficiency for repeated design updates, and (iii) clearly stated validity conditions.

Recent advances in computational science, particularly Dynamic Mode Decomposition (DMD), offer a complementary route by identifying structured linear representations directly from high-dimensional spatiotemporal measurements [43–45]. Specifically, DMD with Control (DMDc) identifies a lifted linear state-space model that separates intrinsic dynamics from actuation effects [46]. In this context, DMDc provides a principled way to identify an explicit, data-driven input–output model linking measured shape evolution to tool motion. Such a structure can support efficient prediction and toolpath updates using linear-systems tools, provided performance is validated. In this sense, our DMDc model is complementary to material-aware SPIF modeling: constitutive/damage frameworks improve physics fidelity for a given material system, whereas DMDc provides a measurement-driven dynamic structure enabling efficient prediction and toolpath update within a validated operating envelope. From a Koopman/operator-theoretic viewpoint, DMD/DMDc can be interpreted as a practical finite-dimensional approximation to the linear Koopman operator acting on selected observables of an otherwise nonlinear system [43,47], which motivates the use of snapshot-based geometry measurements as the model state. At the same time, because the approximation is finite-dimensional and observable-dependent, accuracy is not guaranteed under large shifts in material, thickness, or forming regime, and re-identification may be required. We treat these as scope conditions and evaluate them through operating-envelope validation, out-of-sample rollout prediction error, bootstrap uncertainty estimates, initial condition sensitivity analysis and benchmarking against ILC-SSF. Accordingly, we investigate DMDc as a practical modeling option for SPIF toolpath compensation, with explicit validation of its operating envelope and robustness. The main contributions of this work are:

1. Development of a novel high-dimensional linear state-space dynamical system for the robotic SPIF process, enabling accurate prediction of future specimen geometry based on current shape and tool position.
2. Exploitation of the DMDc-based model to create a computationally efficient, non-iterative trajectory generation scheme via quadratic programming, overcoming critical limitations in toolpath optimization
3. Experimental validation across convex and non-convex geometries demonstrates (a) improved geometric accuracy compared to both uncompensated toolpaths and ILC-SSF benchmarks, with final-shape RMSE reduced by 43.8% on average relative to uncompensated and 44.5% relative to ILC-SSF and (b) sensitivity analysis under measurement noise to assess practical reliability.
4. Empirical characterization and critical discussion of the method's operating envelope under the tested SPIF conditions, including regimes where the linear Koopman/DMDc approximation is accurate and robust, and cases where retraining and/or feedback augmentation is indicated.

The remainder of this paper is structured as follows: Section 2 details the mathematical model and optimization framework. Section 3 describes the data acquisition and processing of 3D point cloud data. The results are analyzed and discussed in Section 4, and the article is concluded in Section 5.

2. Dynamic mode decomposition-based modeling and toolpath optimization

2.1. Dynamics of incremental forming process with spatial and temporal discretization

Let the desirable or target shape be described as a generalized function $f_d(x, y)$ mapping pairs of coordinates x, y in the $x-y$ plane to a corresponding height z_d as

$$z_d = f_d(x, y). \tag{1}$$

Now consider an ISF process in which the actual shape of the metal sheet, $z_a(t)$ is continuously evolving with respect to time due to contact with a rigid tool. The Cartesian position of the tooltip is represented as $\mathbf{u}(t) = \{x^t(t), y^t(t), z^t(t)\}^\top$. Consequently, $f_a : \mathbb{R}^3 \rightarrow \mathbb{R}^n$ and is described as follows:

$$\mathbf{z}_a(t) = f_a(x, y, \mathbf{u}(t)). \tag{2}$$

Let the sheet be discretized or sampled along the x and y axis as X_s and Y_s respectively in Eqs. (3) and (4). Let a square grid $\mathscr{G} \in \mathbb{R}^{N \times N}$ be defined as a Region of Interest (ROI) containing the cross product of X_s and Y_s in Eq. (5). Hence, $f_a : \mathbb{R}^3 \rightarrow \mathbb{R}^{n^2}$

$$X_s = \{x_1, x_2, \dots, x_n\}, \tag{3}$$

$$Y_s = \{y_1, y_2, \dots, y_n\}, \tag{4}$$

$$\mathscr{G} = \left\{ (x_j, y_i) \middle| x_j, y_i \subseteq X_s \times Y_s, i, j = 1, 2, \dots, n \right\}, \tag{5}$$

$$\mathbf{z}_{a,ij}(t) = f_a\left(x_i, y_j, \mathbf{u}(t)\right). \tag{6}$$

Visualization of the metal sheet as a discretized mesh evolving as a function of tool movement is depicted in Fig. 1. According to the

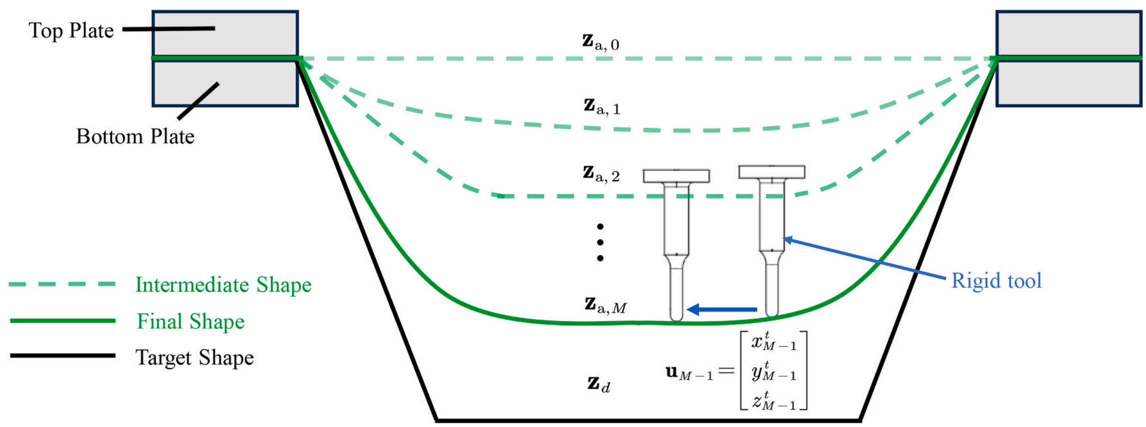


Fig. 2. DMDc modeling of SPIF as a linear state-space system

framework established in [18,48,49], it is possible to describe an SPIF process as a generalized nonlinear dynamical function. The first-order derivative of the current shape, $\dot{\mathbf{z}}_a$ is dependent upon the current shape and tooltip position, i.e., $f: \left(\mathbb{R}^{n^2}, \mathbb{R}^3\right) \to \mathbb{R}^{n^2}$:

$$\dot{\mathbf{z}}_a(t) = f(\mathbf{z}_a(t), \mathbf{u}(t)). \tag{7}$$

Since the process is implemented in a discrete time, sampled uniformly at k time instances, Eq. (7) is represented as

$$\mathbf{z}_{a,k+1} = f\left(\mathbf{z}_{a,k}, \mathbf{u}_k\right) = f(\mathbf{z}_a(kT_s), \mathbf{u}_k(kT_s)), \tag{8}$$

where $\mathscr{K} = \{1, 2, \ldots, M\}$ and $\mathscr{T} = \{t_k = kT_s | k \in \mathscr{K}\}$ describe the evolution of the state-trajectory at k discrete time intervals for a fixed sampling period of T_s seconds and M number of samples [49,50]. At each time instant k, the error between the target shape vector $\mathbf{z}_d$ and actual shape $\mathbf{z}_{a,k}$ due to the cumulative effect of springback, unwanted sheet bending, and the pillowing effect is defined as:

$$\mathbf{e}_{k+1} = \mathbf{z}_{a,k+1} - \mathbf{z}_d. \tag{9}$$

This formulation encapsulates the geometric errors in the SPIF process as a discrete-time nonlinear function. As detailed in the following section, it establishes the foundational framework for approximating these nonlinear dynamics within a high-dimensional linear state-space system.

2.2. Dynamic mode decomposition with control of the SPIF process

The objective of DMDc modeling is to identify the SPIF process as a generalized linear state-space system of the form $\acute{X} = AX + B\gamma$ from snapshot data of the shape or states, $\mathbf{z}_{a,1}, \mathbf{z}_{a,2}, \cdots, \mathbf{z}_{a,M}$ and tool position or inputs $\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_{M-1}$, as depicted in Fig. 2. In other words, for given input and output data, DMDc identifies the function $\mathbf{z}_{a,k+1} = f\left(\mathbf{z}_{a,k}, \mathbf{u}_k\right)$ in terms of a state-matrix, $A \in \mathbb{R}^{N\times N}$ and control or input matrix $B \in \mathbb{R}^{N\times M}$. Unlike previous works [18,49,50] which model state evolution from small toolpath perturbations, the proposed DMDc model explicitly incorporates the toolpath, making it better suited for optimization, as shall be explained later. For $\mathbf{z}_{a,k} \in \mathbb{R}^N$and $\mathbf{u}_k \in \mathbb{R}^Q$, the state, one-step forward state and input snapshot matrices, $X, \acute{X} \in \mathbb{R}^{N\times M}$ and $\gamma \in \mathbb{R}^{Q\times M}$are defined respectively as

$$X = \left[\mathbf{z}_{a,1}, \mathbf{z}_{a,2}, \cdots, \mathbf{z}_{a,M-1}\right], \tag{10}$$

$$\acute{X} = \left[\mathbf{z}_{a,2}, \mathbf{z}_{a,3}, \cdots, \mathbf{z}_{a,M}\right], \tag{11}$$

$$\gamma = [\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_{M-1}]. \tag{12}$$

For a general linear system of the form $\acute{X} = AX + B\gamma$, an operator G can be defined from the state matrix A and control matrix B as

$$\acute{X} = G\Omega = [A \quad B]\begin{bmatrix} X \\ \gamma \end{bmatrix}. \tag{13}$$

To compute the best-fit approximation of G to the snapshot data, the Moore-Penrose pseudoinverse of $X$is computed as $X^\dagger$ to minimize the Frobenius norm of $\|\acute{X} - GX\|_F$:

$$G = \acute{X}\Omega^\dagger. \tag{14}$$

The first step toward this solution is to perform Singular Value Decomposition (SVD) on the input space $\Omega \in \mathbb{R}^{(N+Q)\times M}$ as follows to obtain the left singular vectors matrix, $U \in \mathbb{R}^{(N+Q)\times(N+Q)}$, diagonal matrix of singular values, $\Sigma \in \mathbb{R}^{(N+Q)\times M}$ and right singular vectors matrix, $V^* \in \mathbb{R}^{M\times M}$:

$$\Omega = U\Sigma V^*. \tag{15}$$

The ranks of U, Σ, and V^* are truncated to obtain $\widetilde{U} \in \mathbb{R}^{(N+Q)\times r}$, $\widetilde{\Sigma} \in \mathbb{R}^{r\times r}$, $\widetilde{V}^* \in \mathbb{R}^{r\times M}$ as:

$$\widetilde{U} = U\begin{bmatrix} I_{r\times r} \\ 0_{(N+Q-r)\times r} \end{bmatrix},$$

$$\widetilde{\Sigma} = \Sigma\begin{bmatrix} I_{r\times r} & 0_{r\times(M-r)} \\ 0_{(N+Q-r)\times r} & 0_{(N+Q-r)\times(M-r)} \end{bmatrix} = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_r \end{bmatrix}, \tag{16}$$

$$\widetilde{V} = V\begin{bmatrix} I_{r\times r} \\ 0_{(M-r)\times r} \end{bmatrix}.$$

The truncation is determined by the truncation value r based on the cumulative sum or energy of k singular values σ_k greater than a defined threshold τ_{abs}, as follows:

$$r = \sum_{k=1}^{\ell} (\sigma_k > \tau_{\text{abs}}), \ell = \min(N+Q, M-1), \tag{17}$$

where N is the order or total number of states in the system and Q is the total number of system inputs. Consequently, $\widetilde{U}$ is represented as

$$\widetilde{U} = \begin{bmatrix} \widetilde{U}_1 \\ \widetilde{U}_2 \end{bmatrix},$$

$$\widetilde{U}_1 = [I_{N\times N} \quad 0_{N\times Q}]\widetilde{U}, \tag{18}$$

$$\widetilde{U}_2 = [\,0_{Q\times N} \quad I_{Q\times Q}\,]\widetilde{U},$$

where $\widetilde{U}_1 \in \mathbb{R}^{N\times r}$ and $\widetilde{U}_2 \in \mathbb{R}^{Q\times r}$. The approximations of A and B are then computed as

$$A^p = \acute{X}V\widehat{\Sigma}^{-1}\widetilde{U}_1^*, \tag{19}$$

$$B^p = \acute{X}V\widehat{\Sigma}^{-1}\widetilde{U}_2^*. \tag{20}$$

The proposed model is a reduced-order surrogate designed for computational efficiency rather than exhaustive simulation. To ensure scalability and consistency across varying data fidelities, the truncation rank r is determined by a relative energy threshold $\eta_{rt.}$ rather than a fixed absolute cutoff, as given in Eq. (21):

$$\eta_{rt.} = \frac{\sum_{i=1}^{r}\sigma_i^2}{\sum_{i=1}^{\ell}\sigma_i^2}. \tag{21}$$

We set $\eta_{rt.} \geq 90\%$ allowing the framework to adaptively retain only the dominant modes necessary to represent the dynamics. This approach ensures that the model remains compact for experimental data while automatically scaling to capture the higher-dimensional nuances of high-fidelity simulations. For reproducibility the retained energy $\eta_{rt.}$ and the resulting r, and for each case are reported. The main difference between the standard DMDc formulation and Eqs. (19) and (20) lies in the fact that SVD has not been performed on the output space to obtain $A^p = \widehat{U}^*\acute{X}V\widehat{\sum}^{-1}\widetilde{U}_1^*$ and $B^p = \widehat{U}^*\acute{X}V\widehat{\sum}^{-1}\widetilde{U}_2^*$. The operator $\widehat{U}$is a low-rank approximation of the left-singular vector matrix U such that $\acute{X} = U\Sigma V$. This is to ensure that the dimension of the state vector $\mathbf{z}_{a,k}$ is the same as the target or reference vector $\mathbf{z}_d$, which will allow the toolpath $\mathbf{u}_k$ to be optimized, as shall be explained later. Finally, using Eqs. (19) and (20), the model presented in Eq. (8) is reformulated as

$$\mathbf{z}_{a,k+1} = A^p\mathbf{z}_{a,k} + B^p\mathbf{u}_k. \tag{22}$$

Although SPIF mechanics are nonlinear, DMDc can be interpreted as identifying a finite-dimensional linear approximation to the (controlled) Koopman operator acting on selected observables of the system state. Here, the observable is the discretized surface height field (mesh z-coordinates) used as the state vector. Under this view, the learned matrices (A^p, B^p) approximate the best-fit linear evolution of the chosen observables over one sampling period in a lifted space, rather than linearizing the underlying constitutive mechanics. We emphasize that this approximation is observable- and regime-dependent: no global error bound is claimed for large shifts in material, thickness, friction, step size, or geometry family. Accordingly, we treat model validity as an operating-envelope question and quantify it via out-of-sample prediction error, uncertainty estimates (bootstrap), and experimental benchmarking. To quantify the prediction accuracy of the identified linear DMDc model over the full $n\times n$ height field, we evaluate multi-step rollout predictions driven by the measured tool input sequence. Let the measured (actual) state at time index k be $\mathbf{z}_{a,k}$ and the model prediction be $\widehat{\mathbf{z}}_{a,k}$. Starting from the measured initial condition $\widehat{\mathbf{z}}_{a,1} = \mathbf{z}_{a,1}$, the rollout prediction is computed recursively as:

$$\widehat{\mathbf{z}}_{a,k} = A^p\widehat{\mathbf{z}}_{a,k-1} + B^p\mathbf{u}_{k-1}. \tag{23}$$

The node-wise prediction error is defined as $\mathbf{r}_k = \mathbf{z}_{a,k} - \widehat{\mathbf{z}}_{a,k}$, which is distinct from geometric error $\mathbf{e}_k$ defined in Eq. (9). In this work we primarily report absolute errors $|\mathbf{r}_k|$ to characterize the magnitude of prediction deviation independent of sign (over/under prediction). At each time step k, the spatial distribution of error over the full mesh is summarized using the Root Mean Squared Error (RMSE), e_R and Mean Absolute Error (MAE), e_M:

$$e_R = \sqrt{\frac{1}{n^2}\|\mathbf{r}_k\|_2^2}, \tag{24}$$

$$e_M = \frac{1}{n^2}\|\mathbf{r}_k\|_1. \tag{25}$$

Additionally, a time-aggregated spatial error map is formed by reshaping $|\mathbf{r}_k|$ back to an $n\times n$ grid for visualization. For selected cases, $|\mathbf{r}_k|$ is visualized at representative time instants (10%, 50%, and 100% of snapshots) to illustrate how error patterns evolve over the forming sequence, revealing structures tied to unmodeled nonlinearities like feature interactions. All error computations are time-aligned such that $\widehat{\mathbf{z}}_{a,k}$ is compared against $\mathbf{z}_{a,k}$, while the first snapshot ($k=1$) is excluded from error metrics since it is used as initialization. These metrics empirically assess the reasonableness of the finite Koopman approximation for nonlinear SPIF dynamics within the tested regimes. Because rollout accuracy can be limited by both identification/model mismatch and measurement/registration noise in the initial reconstructed surface, we next perform a fixed-model sensitivity study to isolate the effect of initialization noise. Specifically, we keep the identified DMDc operators (A^p, B^p) fixed and perturb only the initial condition as:

$$\widetilde{\mathbf{z}}_{a,1} = \mathbf{z}_{a,1} + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 I_N). \tag{26}$$

Final-shape errors are reported for $\sigma \in \{0, 0.25, 0.5, 1.0\}$ mm. For each geometry we reuse a single Gaussian noise realization $\boldsymbol{\varepsilon}$ and scale it by σ to ensure trends reflect noise amplitude rather than different random draws.

The DMD modes are extracted from A^p via eigenvalue decomposition as follows:

$$A^pW = W\Lambda, \tag{27}$$

where the columns of $W \in \mathbb{R}^{N\times N}$ are eigenvectors $\mathbf{w}_i \in \mathbb{R}^N$ such that $W = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_N]$, while Λ is a diagonal matrix containing corresponding eigenvalues λ_i with $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_N)$. The modes are sorted by descending $|\lambda_i|$ to prioritize dominant dynamics and reshaped to the spatial grid for interpretation. The spectral decomposition of the learned state matrix is used to characterize temporal and spatial dynamics. Eigenvalues plotted in the complex plane provide a stability and persistence diagnostic: eigenvalues inside the unit circle indicate decaying modes, eigenvalues near the unit circle indicate persistent or marginally stable behavior, and eigenvalues outside the unit circle reflect growing or unstable modes. Spatial DMD modes are obtained by reshaping eigenvectors to the spatial discretization and are ordered by an explicit importance metric to prioritize dominant dynamics for analysis and control. Complex-valued modes are visualized by their magnitude with top nine magnitudes analyzed to emphasize spatial amplitude. Dominant modes typically capture coarse, low-frequency deformation patterns, whereas higher-order modes show finer, higher-frequency structure that decays more rapidly. Visualization procedures include plotting eigenvalues with the unit circle as a reference for stability assessment, displaying selected DMD modes as grid-formatted heatmaps or surface plots with consistent color limits across panels to reveal localization and propagation, and visualizing columns of the input matrix as surface and heatmap views with toolpath overlays to link input excitation to spatial response.

2.3. Final shape error evaluation metrics and uncertainty quantification using sensitivity analysis

Let $S_c \subset \mathbb{R}^3$ denote the nominal CAD surface and let $\mathbf{p}_{m,i} \in \mathbb{R}^3$ be the i^{th} measured point (or mesh vertex) after rigid alignment. Define the closest point on the CAD surface as

$$\mathbf{q}_{c,i} = \underset{\mathbf{q}_t \in S_c}{\mathrm{argmin}} \,\|\,\mathbf{p}_i - \mathbf{q}_t\|_2. \tag{28}$$

The node-wise geometric deviation field $\delta_i \in \mathbb{R}(\text{mm})$ at n_p sampled points is then defined as the signed distance to the nominal surface along the outward unit normal $\mathbf{n}\left(\mathbf{q}_{c,i}\right)$ at $\mathbf{q}_{c,i}$ as:

$$\delta_i = \left(\mathbf{p}_i - \mathbf{q}_{c,i}\right)^\top \mathbf{n}\left(\mathbf{q}_{c,i}\right), \tag{29}$$

where the sign convention follows the CAD surface normal orientation used by AutoCAD PowerShape 2025. We use $|\delta_i|$for magnitude-based metrics to avoid sign cancellation when aggregating spatially. For each experiment and compensation method (Uncompensated U, ILC-SSF, and DMDc), performance is summarized using distributional metrics of $|\delta|$ mean absolute error δ_E and root-mean-square error δ_R, as:

$$\delta_E = \frac{1}{n_p}\sum_{i=1}^{n_p}|\delta_i|, \tag{30}$$

$$\delta_R = \sqrt{\frac{1}{n_p}\sum_{i=1}^{n_p}\delta_i^2}. \tag{31}$$

To visualize the full error distribution, we utilize the empirical cumulative distribution function (ECDF), defined as:

$$\widehat{F}_{|\delta|}(x) = \frac{1}{n_p}\sum_{i=1}^{n_p}1_{\{|\delta_i|\le x\}}, \tag{32}$$

where $1_{\{\cdot\}}$ is the indicator function. From this distribution, we also report the 95th empirical percentile P_{95}, where $P_q = \inf\{x : \widehat{F}_{|\delta|}(x) \ge q/100\}$. Between-method differences are reported as $\Delta M = M_{\text{DMDc}} - M_{\text{ref}}$ (negative indicates improvement for $|\delta|$-based metrics), where M_{ref} is the corresponding metric for ILC–SSF. Uncertainty is quantified using a spatial block bootstrap to account for spatial correlation of surface deviations. The exported point coordinates are first rigidly registered in the $x-y$ plane to a common reference frame using a planar $SE(2)$ transform. The $x-y$ domain is then partitioned into square tiles of side length ℓ (mm), and each point is assigned with a tile ID based on its $x-y$ location. For each method independently, we generate $B = 1000$ bootstrap replicates by resampling tiles with replacement and recomputing the metrics using all points contained in the resampled tiles; 95% confidence intervals are reported using the percentile bootstrap (2.5th–97.5th percentiles). To ensure that spatial aggregation and uncertainty quantification compare the same physical regions across toolpaths, we preprocess each measured point set using a rigid planar registration in the global $x-y$ plane. For each method m, we estimate a transformation $T_m \in SE(2)$ (rotation about the z axis and translation in $x-y$ plane) that aligns the measured points to a chosen reference dataset by minimizing a standard Iterative Closest Point objective on the planar coordinates:

$$T_m^* = \underset{T_m}{\operatorname{argmin}}\sum_i\left\|\Pi_{xy}\left(T_m\left(\mathbf{p}_i^{(m)}\right) - \Pi_{xy}\left(\mathbf{r}_{\pi(i)}\right)\right)\right\|_2^2, \tag{33}$$

where $\Pi_{xy}(\bullet)$ extracts the $x-y$ components and $\mathbf{r}_{\pi(i)}$ denotes the matched reference point. For pair-wise delta comparisons only, the registered datasets are additionally restricted to a common region of support by retaining only points whose planar coordinates fall inside the intersection of the methods' bounding boxes $x-y$ with a small margin. This prevents apparent differences from being driven by unequal spatial coverage rather than true geometric accuracy. When pair wise differences are reported, they are computed on matched spatial support by restricting both methods to a common $x-y$ region of interest (intersection of the methods' $x-y$ bounding boxes, with a small margin) and applying a paired tile bootstrap on the intersection of tile IDs shared by both methods. The MATLAB registration/cropping steps are used only to standardize spatial indexing and define matched comparison regions; the deviation values $\{\delta_i\}$ remain those exported by PowerShape relative to the CAD model. For each bootstrap replicate $b = 1, \cdots, B$, tiles are resampled with replacement (including all deviation samples within selected tiles), metrics are recomputed, and $\Delta M^{*(b)} = M_{\text{DMDc}}^{*(b)} - M_{\text{ref}}^{*(b)}$is formed. We report 95% confidence intervals using the percentile bootstrap, i.e., the 2.5th–97.5th empirical percentiles of $\{\Delta M^{*(b)}\}_{b=1}^{B}$. The block size ℓcontrols the bias–variance trade-off under spatial dependence; we therefore repeated the analysis for $\ell \in \{3, 5, 8\}$ mm, and the qualitative ranking of methods and conclusions remained unchanged.

2.4. DMDc model uncertainty quantification using bootstrapping

To assess the statistical robustness and representativeness of the identified DMDc modes, a nonparametric bootstrapping framework is employed [51] using confidence estimates for coherent structures under snapshot perturbations. For b resampling iterations, snapshot indices $\mathscr{I}_b = \{i_1, i_2, \ldots, i_m\}$ are drawn with replacement whereby state and control matrices $X^{(b)} = X\left[I_{m\times m} \quad 0_{(N-m)\times m}\right]^\top$ and $U^{(b)} = U\left[I_{m\times m} \quad 0_{(N-m)\times m}\right]^\top$ are constructed respectively. Consequently, $X_1^{(b)} = \left[\mathbf{z}_{a,1}, \mathbf{z}_{a,2}, \cdots, \mathbf{z}_{a,m-1}\right]$,$X_2^{(b)} = \left[\mathbf{z}_{a,2}, \mathbf{z}_{a,3}, \cdots, \mathbf{z}_{a,m}\right]$,$\gamma^{(b)} = \left[\mathbf{u}_{a,1}, \mathbf{u}_{a,2}, \cdots, \mathbf{u}_{a,m-1}\right]$ and $\Omega^{(b)} = \left[X_1^{(b)}; \gamma^{(b)}\right]^\top$. As a result, Eq. (15) will change as follows:

$$\Omega^{(b)} = U^{(b)}\Sigma^{(b)}V^{(b)}. \tag{34}$$

Similarly, Eqs. (16), (17) and (19) change as follows:

$$r^{(b)} = \sum_{k=1}^{\ell}\left(\sigma_k^{(b)} > \tau_{abs}\right), \ell = \min(N+Q, m-1),$$

$$U_{\text{til}}^{(b)} = U^{(b)}\begin{bmatrix} I_{r^{(b)}\times r^{(b)}} \\ 0_{\left(N+Q-r^{(b)}\right)\times r^{(b)}} \end{bmatrix} = \begin{bmatrix} U_{\text{til},1}^{(b)} & U_{\text{til},2}^{(b)} \end{bmatrix},$$

$$\Sigma_{\text{til}}^{(b)} = \begin{bmatrix} I_{r^{(b)}\times r^{(b)}} & 0_{r^{(b)}\times\left(m-r^{(b)}\right)} \\ 0_{\left(N+Q-r^{(b)}\right)\times r^{(b)}} & 0_{\left(N+Q-r^{(b)}\right)\times\left(m-r^{(b)}\right)} \end{bmatrix} = \begin{bmatrix} \sigma_1^{(b)} & 0 & \cdots & 0 \\ 0 & \sigma_2^{(b)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_{r^{(b)}}^{(b)} \end{bmatrix}, \tag{35}$$

$$V_{\text{til}}^{(b)} = V^{(b)}\begin{bmatrix} I_{r^{(b)}\times r^{(b)}} \\ 0_{\left(m-r^{(b)}\right)\times r^{(b)}} \end{bmatrix},$$

$$A^{(b)} = {X_2}^{(b)}{V_{\text{til}}}^{(b)}\left({\Sigma_{\text{til}}}^{(b)}\right)^{-1}U_{\text{til},1}^{(b)\top}.$$

For each realization, eigenvalues and eigenvectors of $A^{(b)}$ are computed as:

$$A^{(b)}W^{(b)} = W^{(b)}\Lambda^{(b)}, \tag{36}$$

where the columns of $W^{(b)} \in \mathbb{R}^{N\times N}$ are eigenvectors of $\mathbf{w}_i^{(b)} \in \mathbb{R}^N$ such that $W^{(b)} = \left[\mathbf{w}_1^{(b)}, \mathbf{w}_2^{(b)}, \ldots, \mathbf{w}_N^{(b)}\right]$, while $\Lambda^{(b)}$ is a diagonal matrix containing corresponding eigenvalues $\lambda_i^{(b)}$ with $\Lambda^{(b)} = \text{diag}\left(\lambda_1^{(b)}, \lambda_2^{(b)}, \ldots, \lambda_N^{(b)}\right)$. The eigenvectors are sorted by the magnitude of their corresponding eigenvalues $|\lambda_j^{(b)}|$ in descending order. Only the top k eigenvectors or dominant dynamic modes for bootstrap sample b are kept for analysis in the sorted matrix $W_s^{(b)} \in \mathbb{R}^{N\times k}$ as:

$$W_s^{(b)} = \left[\mathbf{w}_1^{(b)}, \mathbf{w}_2^{(b)}, \ldots, \mathbf{w}_k^{(b)}\right]. \tag{37}$$

For the $w_{i,j}^{(b)}$ element representing the value of the j^{th} mode at the i^{th} element of $\mathbf{w}_j^{(b)}$, the mean $\overline{\phi}_{i,j}$ and variance $\sigma_{i,j}^2$ are computed as

$$\overline{\phi}_{i,j} = \frac{1}{B}\sum_{b=1}^{B}|w_{i,j}^{(b)}|, \tag{38}$$

$$\sigma_{i,j}^2 = \frac{1}{B-1}\sum_{b=1}^{B}\left(|w_{i,j}^{(b)}| - \overline{\phi}_{i,j}\right)^2. \tag{39}$$

The robustness and representativeness of the bootstrapped DMDc modes are subsequently illustrated using both experimental measurements and simulation datasets.

2.5. Toolpath optimization for forming error

The first step is to express the target shape as a mesh $\mathscr{M}$ having a finite set of oriented triangular facets $\mathscr{T}_i^s$ as follows [52]:

$$\mathscr{M} = \bigcup_{i=1}^{N_f}\mathscr{T}_i^s, \tag{40}$$

where each facet is defined by its vertices $[v_{i,1}, v_{i,2}, v_{i,3}]^\top$ and an outward-facing unit normal vector $\mathbf{n}_i$ for the number of facets, N_f. It is assumed that the mesh forms a closed surface boundary at its minimum point $z_{\min}$ and no enclosing facets exist at the top of the mesh. Given a maximum value $z_{\max}$ of the mesh surface, each layer of the mesh z_k is represented as:

$$z_k = z_{\max} - k\Delta z, \tag{41}$$

where Δz is the thickness of each layer or step size. The intersection of the mesh $\mathscr{M}$ with the plane at each layer yields a closed polygonal surface contour $\mathscr{C}_k = \{\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_{N_s}\}$, with $\mathbf{q}_i \in \mathbb{R}^2$ and N_s representing the vertices and total number of intersections respectively [53]. The spiral toolpath $\mathbf{S}(t)$ is then parameterized as:

$$\mathbf{S}(t) = \mathbf{c} + \rho(t)\mathbf{u}(\theta(t))t \in [0, T], \tag{42}$$

$$\rho(t) = R_{\max} - \frac{s}{2\pi}t + \kappa(t), \tag{43}$$

$$\kappa(t) = \alpha\frac{d^2}{dt^2}D(\mathbf{S}(t)), \tag{44}$$

$$\alpha = \frac{r}{2}\left(1 - \frac{s}{4r}\right), \tag{45}$$

$$\mathbf{u}(\theta) = (\cos\theta, \sin\theta), \tag{46}$$

where $\mathbf{c}$ is the centroid of $\mathscr{C}_k$, $\mathbf{u}(\theta)$ is an angular unit vector, and $\theta(t)$ is a monotonically increasing angle with respect to time. Similarly $\rho(t)$ is the time-varying radius defined by $R_{\max}$, the maximum distance from $\mathbf{c}$ to any vertex of $\mathscr{C}_k$ and curvature compensation term $\kappa(t)$ [54]. The curvature term adjusts the radial distance according to an adaptive gain α and distance transform value $D(\mathbf{S}(t))$. The scallop height is governed by local radius of curvature R_c [55]:

$$h \approx \frac{s^2}{8}\left(\frac{1}{r} + \frac{\mathrm{sgn}(R_c)}{|R_c|}\right), \tag{47}$$

$$\mathrm{sgn}(R_c) = \begin{cases} +1, |R_c| \le r \\ -1 |R_c| > r \end{cases}. \tag{48}$$

Given a maximum scallop height h_{max}, curvature threshold R_∞, and boundary vertex $\mathbf{v}$, s is dynamically computed as:

$$s(\mathbf{v}) = \begin{cases} 2\sqrt{2rh_{\max}}, & R_c(\mathbf{v})| \ge R_\infty, \\ 2\sqrt{\dfrac{8rh_{\max}}{1 + \dfrac{r}{|R_c(\mathbf{v})|}}}, & R_c(\mathbf{v}) > 0 \ \& |R_c(\mathbf{v})| < R_\infty, \\ 2\sqrt{\dfrac{8rh_{\max}}{1 - \dfrac{r}{|R_c(\mathbf{v})|}}}, & R_c(\mathbf{v}) < 0 \ \& |R_c(\mathbf{v})| > r. \end{cases}, \tag{49}$$

$$R_c(\mathbf{v}) = \frac{\overline{s}}{\psi}, \tag{50}$$

$$\overline{s} = \frac{\|\mathbf{e}_1\| + \|\mathbf{e}_2\|}{2}, \tag{51}$$

$$\psi = \mathrm{sgn}(\mathbf{e}_1 \times \mathbf{e}_2)(\pi - \phi), \tag{52}$$

where $\mathbf{e}_1$ and $\mathbf{e}_2$ are adjacent edge vectors separated by an angle ϕ and ψ being the signed turning angle between the vectors [56]. Finally, the following energy functional $\mathscr{E}(\mathbf{S}(t))$ is used to optimize $\mathbf{S}(t)$ for path length, smoothness and engagement control using weighting factors λ_s and μ respectively [57]:

$$\begin{aligned} \min_{\mathbf{S}(t)} \mathscr{E}(\mathbf{S}(t)) = & \int_0^T \left\|\frac{d\mathbf{S}}{dt}\right\|^2 dt + \lambda_s \int_0^T \left(\frac{d^2\mathbf{S}}{dt^2}\right)^2 dt \\ & + \mu\int_0^T (D(\mathbf{S}(t)) - (r + s/2))^2 dt. \end{aligned} \tag{53}$$

As highlighted by Asghar et al. in [58], the overall forming error or deflection is the accumulation of localized errors or deflections occurring along the toolpath during the forming process. Therefore, static toolpath compensation based on point-wise predicted deflections can significantly reduce errors. Consequently, we propose to compute the optimized toolpath $\mathbf{u}_k$ to minimize $\mathbf{e}_{k+1}$ at each time step. Given the learned predictor, $\mathbf{e}_{k+1}$ is affine in $\mathbf{u}_k$, hence the resulting objective yields a convex Quadratic Programming (QP) as given in Eqs. (54)–(56). The purpose of adding a penalty or regularization term is to ensure that the optimal value of $\mathbf{u}_k$ does not vary significantly from the nominal or CAM-generated toolpath $\mathbf{u}_k^o$.

$$\min_{\mathbf{u}_k} J = \|\mathbf{e}_{k+1}\|_2^2 + \lambda\|\mathbf{u}_k - \mathbf{u}_k^o\|_2^2 \tag{54}$$

$$= \|\overline{A}\|_2^2 + 2\overline{A}^\top B^p\mathbf{u}_k + \|B^p\mathbf{u}_k\|_2^2 + \lambda\|\mathbf{u}_k\|_2^2 - 2\lambda\mathbf{u}_k^{o\top}\mathbf{u}_k + \lambda\|\mathbf{u}_k^o\|_2^2$$

$$= \frac{1}{2}\mathbf{u}_k^\top H\mathbf{u}_k + f^\top\mathbf{u}_k$$

$$\text{s.t. } \mathbf{u}_{k,\min} \leqslant \mathbf{u}_k \leqslant \mathbf{u}_{k,\max},$$

where $\boldsymbol{\lambda} \in \mathbb{R}^{3\times3}$, $\overline{A} \in \mathbb{R}^{n\times n}$, $H \in \mathbb{R}^{n\times n}$ and $f^\top \in \mathbb{R}^{1\times n}$ are defined as:

$$\boldsymbol{\lambda} = \begin{bmatrix} \lambda_x & 0 & 0 \\ 0 & \lambda_y & 0 \\ 0 & 0 & \lambda_z \end{bmatrix}, \tag{55}$$

$$\overline{A} = A^p\mathbf{z}_{a,k} - \mathbf{z}_d, \tag{56}$$

$$H = 2B^{p\top}B^p + \lambda I, \tag{57}$$

$$f^\top = 2(\overline{A}^\top B^p - \lambda\mathbf{u}_k^{o\top}). \tag{58}$$

The optimization variable $\mathbf{u}_k = [x_k, y_k, z_k]^\top$ is the commanded tool position at step k. Therefore, the QP computes physically implementable toolpath compensation that regulates the predicted shape error while respecting machine and forming limits. Process parameters such as sheet thickness t, tool radius R, tool speed v (and acceleration limits), and

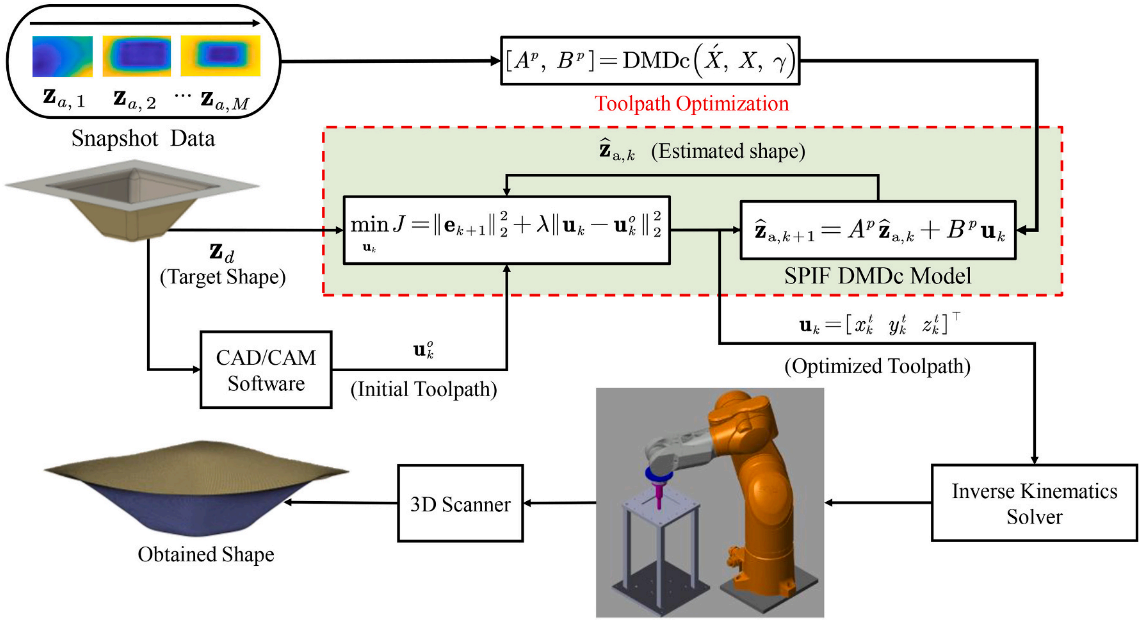


Fig. 3. DMDc-based system modeling of SPIF process, toolpath optimization and execution.

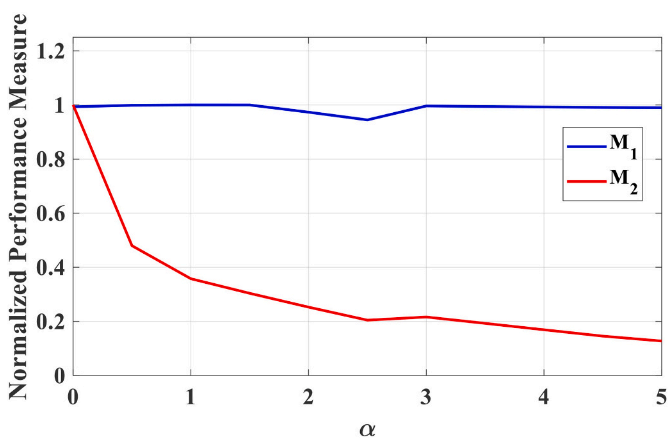


Fig. 4. Effect of regularization on final shape error and toolpath deviation.

Young's Modulus, E etc. influence the optimization in two ways: (i) implicitly, through the identified predictor (A^p, B^p), whose input–output sensitivity reflects the effective compliance/plastic response of the sheet–tool system under the specific (t, R, ν, E) operating envelope used for training; and (ii) explicitly, through the constraint bounds and penalty weights that restrict step-to-step tool motion to feasible values. Consequently, the QP should be interpreted as a bounded, envelope-aware toolpath controller rather than an abstract numerical program. The tracking term $\|\mathbf{e}_{k+1}\|_2^2$ directly penalizes the predicted geometric deviation of the formed height field from the target shape and therefore corresponds to shape-accuracy regulation. The regularization term $\left\|\mathbf{u}_k - \mathbf{u}_k^o\right\|_2^2$ has a physical role beyond numerical conditioning: it discourages overly aggressive compensation relative to the nominal CAM toolpath, which is important because SPIF exhibits unmodeled effects (contact nonlinearity, local compliance, hysteresis/plasticity) and practical machine limitations (finite acceleration/resolution). Hence, $\boldsymbol{\lambda}$ controls an accuracy–conservatism trade-off, where larger values yield smaller, smoother corrections that are more likely to remain within the validated operating envelope. The box constraints $\mathbf{u}_{k,\min} \leqslant \mathbf{u}_k \leqslant \mathbf{u}_{k,\max}$ encode hard limits on implementable tool motion at each step and serve as explicit process-limit constraints. In our implementation these limits are defined relative to the nominal CAM tool command $\mathbf{u}_k^o = \left[x_k^o, y_k^o, z_k^o\right]^\top$ such that the lateral motion is bounded to $\pm(\Delta x, \Delta y)$ around (x_k^o, y_k^o), i.e., $x_k \in \left[x_k^o - \Delta x, x_k^o + \Delta x\right]$ and $y_k \in \left[y_k^o - \Delta y, y_k^o + \Delta y\right]$. The vertical command is constrained using a one-sided scaling envelope with parameter $\beta \geq 0$, $z_k \in \left[\min\left(z_k^o, (1+\beta)z_k^o\right), \max\left(z_k^o, (1+\beta)z_k^o\right)\right]$ which remains robust to sign conventions while limiting the allowable depth update relative to the nominal toolpath. Defining the incremental compensation $\Delta\mathbf{u}_k = \mathbf{u}_k - \mathbf{u}_k^o$, these constraints can be written equivalently as $\mathbf{u}_{k,\min} \leqslant \Delta\mathbf{u}_{k,\min} \leqslant \Delta\mathbf{u}_k \leqslant \Delta\mathbf{u}_{k,\max}$, making explicit that the QP computes a bounded compensation around the baseline CAM toolpath rather than an unconstrained input. It is important to note that the matrices A^p, B^p are derived from Al-1050 (0.3 mm) rollout data. These matrices therefore embed the effective response of the specific material, thickness, tooling, and friction configuration used during testing. Changes in material, thickness, or tooling typically alter input-output sensitivity. Consequently, re-identifying A^p, B^p and retuning $\boldsymbol{\lambda}$ would be required, though this remains outside the present scope. With the above objective and process-limit constraints, the QP computes a bounded compensation around the nominal CAM toolpath at each step. The proposed approach is described as "non-iterative" because it relies on single-shot compensation at deployment. Once a DMDc model is identified from an initial dataset, the optimized toolpath is computed offline and then executed without repeated form–measure–update cycles. This does not eliminate the need for offline effort, since data preprocessing, model identification, and QP optimization are still required, but it avoids the multiple physical iteration sheets that iterative compensation schemes typically need to converge. In our experiments, the DMDc model is identified using data from a single nominal specimen, consisting of one forming and scan per geometry, or from a single simulation dataset in the simulation studies. After this step, additional compensated parts can be produced without further iterative learning cycles.

The entire modeling and optimization process is depicted in Fig. 3. Parametric sensitivity analysis evaluates the aggregate effect of λ in the constrained optimization framework for SPIF. By scaling λuniformly with a scalar parameter α, the multi-parameter cost function can be analyzed in terms of two key performance measures: final shape error, M_1 and total toolpath deviation, M_2, accumulated as the squared L_2-norm of deviations from the nominal tool position. Both measures are normalized to [0,1] for comparison, with α varied from 0 to 5.

$$M_1 = \| \mathbf{z}_{a,M} - \mathbf{z}_d \|_2, \tag{59}$$

$$M_2 = \sum_{k=1}^{M} \| \mathbf{u}_k - \mathbf{u}_k^o \|_2^2. \tag{60}$$

Fig. 4 depicts the normalized performance measures against α. M_1 starts from a normalized value of 1 for $\alpha = 0$, indicating maximum error under unconstrained optimization. As α increases to approximately 1.5, M_1 decreases to about 0.85–0.9, suggesting an initial improvement in shape accuracy. Beyond this, the curve stabilizes around 0.9, with minor fluctuations but no significant rise, implying that moderate regularization reduces shape error by stabilizing the toolpath adjustments. In contrast, M_2 exhibits a steady monotonic decrease from near 1 at $\alpha = 0$ to approximately 0.1 at $\alpha = 5$. Higher regularization penalizes deviations more heavily, constraining the optimal toolpath u_k closer to the nominal u_k^o, thereby reducing cumulative adjustments. The sharp initial drop from 1 to approximately 0.5 with $\alpha = 1$ highlights that even mild regularization effectively curbs excessive movements, with diminishing returns at higher values. Unlike a classic trade-off where curves intersect at a Pareto-optimal point, the measures do not intersect with M_1 improving and stabilizing while M_2 continues to decline. This suggests a "win-win" regime at moderate $\alpha \in [1, 2]$, where both error and deviation are minimized relative to extremes. At lower values ($\alpha \approx 0$) high deviations introduce modeling inaccuracies or process instabilities, inflating shape error, while for higher values ($\alpha > 3$), excessive conservatism limits corrective actions. However, the stabilized M_1 indicates that the nominal toolpath is already reasonably effective, with regularization providing robustness without substantial accuracy loss.

3. Data acquisition and processing with 3D scanner

3.1. Geometric modeling and signal processing

The time-varying actual point clouds are formally represented as a set of spatiotemporal tuples. Let $\mathscr{N} = \{1, \dots, M\}$ be the set of node indices. For each node $n \in \mathscr{N}$ and time index $k \in \mathscr{K}$, the spatial position is given by

$$\begin{aligned} x_n(k) &= X_{\text{node},n}(k) \\ y_n(k) &= Y_{\text{node},n}(k) \\ z_n(k) &= Z_{\text{node},n}(k) \end{aligned} \tag{61}$$

The complete spatiotemporal system is then defined as

$$\mathscr{P}_k = \{t_k, n, x_n(k), y_n(k), z_n(k) \,|\, k \in \mathscr{K}, n \in \mathscr{N}\} \subseteq \mathbb{R}^5, \tag{62}$$

where $\mathscr{P}_k \in \mathbb{R}^5$ is the subspace space of physical time instants, node identifiers, and 3D coordinates. The mapping $k \rightarrow t_k = k\Delta t$ preserves temporal ordering with uniform sampling. The outlier removal algorithm processes each time instant $t_k \in \mathscr{T}$ independently. For the spatial point cloud, $\mathscr{P}_k = \left\{ \mathbf{p}_{k,n} = (x_n(k), y_n(k), z_n(k)) \; n \in \mathscr{N} \right\}$. For each point $p_{k,n}$, define its spatial neighborhood $\mathscr{N}_{k,n}$ via k- nearest neighbors within a radius of $r_{k,n}$:

$$\mathscr{N}_{k,n} = \left\{ \mathbf{p}_{k,m} \in \mathscr{P}_k \middle| \left\| \mathbf{p}_{k,n} - \mathbf{p}_{k,m} \right\| \leqslant r_{k,n} \right\}. \tag{63}$$

where $r_{k,n} \in \mathbb{R}^3$ is the minimal radius containing exactly k fneighbors. For each neighborhood $\mathscr{N}_{k,n}$, a centroid $\mu_{k,n}$ is computed as

$$\mu_{k,n} = \sum_{\mathbf{p} \in \mathscr{N}_{k,n}} \mathbf{p}. \tag{64}$$

Similarly, a covariance matrix $\mathbf{C}_{k,n}$ is evaluated and is further deconstructed using SVD into left singular values, a singular values matrix and right singular values, $\mathbf{V}_{k,n}, \mathbf{\Sigma}_{k,n}$ and $\mathbf{V}_{k,n}{}^{\top}$ respectively:

$$\mathbf{C}_{k,n} = \sum_{\mathbf{p} \in \mathscr{N}_{k,n}} (\mathbf{p} - \mu_{k,n})(\mathbf{p} - \mu_{k,n})^{\top}, \tag{65}$$

$$\mathbf{C}_{k,n} = \mathbf{V}_{k,n}\mathbf{\Sigma}_{k,n}\mathbf{V}_{k,n}^{\top}, \mathbf{\Sigma}_{k,n} = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix}, \tag{66}$$

With $\sigma_3 \geq \sigma_2 \geq \sigma_1 \geq 0$ and the plane normal being the eigenvector for σ_3, i.e., $\mathbf{v}_{3,k,n}$. An orthogonal distance, $d_{k,n}$ is then defined by calculating the distance between $\mathbf{p}_{k,n}$ to its local plane:

$$d_{k,n} = (\mathbf{p}_{k,n} - \boldsymbol{\mu}_{k,n}) \cdot \mathbf{v}_{3,k,n}. \tag{67}$$

Next, the median $\widetilde{d}_k$ of the orthogonal distance is computed, followed by calculation of the Mean Absolute Deviation (MAD), MAD_k, of the median orthogonal distance:

$$\widetilde{d}_k = \underset{n \in \mathscr{N}}{\text{med}} |d_{k,n}|, \tag{68}$$

$$\text{MAD}_k = \underset{n \in \mathscr{N}}{\text{med}} \left| |d_{k,n}| - \widetilde{d}_k \right|. \tag{69}$$

Based on the MAD, a threshold value τ_k is set, with a multiplier, η for adjustment:

$$\tau_k = \widetilde{d}_k + \eta \text{MAD}_k. \tag{70}$$

Using τ_k an inlier point set $\mathscr{I}_k$ corresponding to a smooth continuous surface can be identified as

$$\mathscr{I}_k = \left\{ n \in \mathscr{N} \middle| |d_{k,n}| \leq \tau_k \right\}. \tag{71}$$

For denoising, consider a set $\mathscr{N}_{k,n}^*$ in the k-nearest neighborhood of $\mathscr{I}_k$. A smooth position set $\mathbf{p}_{k,n}^s$ is then evaluated as

$$\mathbf{p}_{k,n}^s = \frac{1}{k} \sum_{\mathbf{q} \in \mathscr{N}_{k,n}^*} \mathbf{q}. \tag{72}$$

The last step of denoising involves applying a planarity constant σ_3/σ_1 to sift out the final vector $\mathbf{p}_{k,n}^f$ from $\mathbf{p}_{k,n}^s$, shown as

$$\mathbf{p}_{k,n}^f = \begin{cases} \mathbf{p}_{k,n}^s & \text{if } \dfrac{\sigma_3}{\sigma_1} < \varepsilon \\ \mathbf{p}_{k,n} & \text{otherwise} \end{cases}. \tag{73}$$

After following the steps outlined in Eq. (63), a filtered point cloud set $\mathscr{P}_k^f$ is obtained as:

$$\mathscr{P}_k^f = \bigcup \left\{ \left(t_k, n, \mathbf{p}_{k,n}^f \right) | n \in \mathscr{I}_k \right\}. \tag{74}$$

Thin-Plate Spline (TPS) regression requires the solution of a large system of linear equations based on the interpolation matrix, the size of which grows cubically with the number of data points. For high-density 3D point clouds, this causes a significant computational burden, leading to significant processing time, increased memory consumption, and numerical instability. To mitigate these issues, it becomes necessary to reduce the number of data points through down-sampling by employing a grid-based approach. Grid based averaging partitions the point cloud data into a grid of cubic cells whereby all points within each cell are represented by their average coordinates. Therefore, the essential geometric features of the underlying surface are preserved while significantly reducing the number of points and thereby lowering the computational cost of executing the TPS model and reducing the risk of overfitting due to noisy or redundant data. The filtered spatiotemporal point cloud $\mathscr{P}^f$ is down-sampled independently at each time instant t_k using a grid averaging approach. For each t_k, define a 3D voxel grid

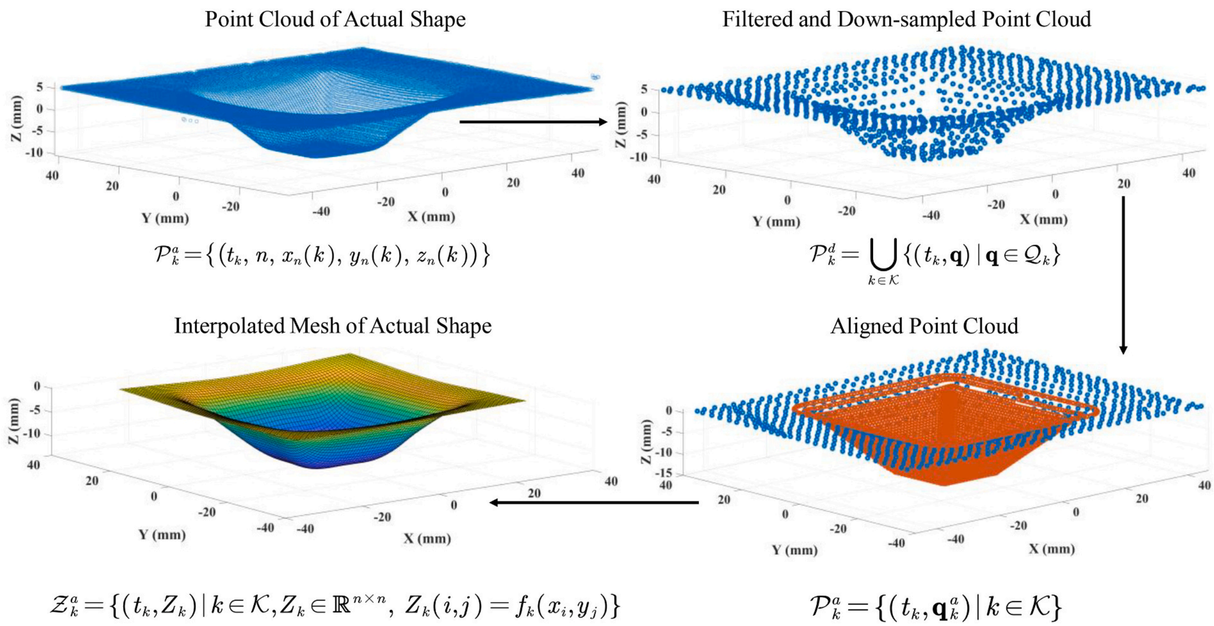


Fig. 5. Point cloud to mesh conversion process for actual shape

partitioning the spatial domain as $\mathscr{G}_{k,i,j,l} \in \mathbb{R}^3$ with each voxel being a cube with side length δ and $x_{\min}, y_{\min}, z_{\min}$ the minimum values of the voxel edges along the x, y and z axes:

$$\mathscr{G}_{k,i,j,l} = \{(x,y,z) \,|\, x \in (x_{\min} + i\delta, x_{\min} + (i+1)\delta), y \in (y_{\min} + j\delta, y_{\min} + (j+1)\delta), z \in (z_{\min} + i\delta, z_{\min} + (l+1)\delta)\},\tag{75}$$

where $i = \{0, 1, 2, 3, \ldots, N_x - 1\}$, and N_x is the number of x coordinate in $\mathscr{P}_k^f$, j and l are corresponding to similar sets for y and z axes respectively. For each voxel $\mathscr{G}_{k,i,j,l}$, all points inside the voxel are collected as $\mathscr{V}_{k,i,j,l}$, while for non-empty voxels $\mathscr{V}_{k,i,j,l} > 0$, the centroid $\mathbf{q}_{k,i,j,l}$ is computed as

$$\mathscr{V}_{k,i,j,l} = \left\{\mathbf{p}_{k,n}^f \in \mathscr{P}_k^f \middle| \mathbf{p}_{k,n}^f \in \mathscr{G}_{k,i,j,l}\right\},\tag{76}$$

$$\mathbf{q}_{k,i,j,l} = \frac{1}{|\mathscr{V}_{k,i,j,l}|} \sum_{\mathbf{p} \in \mathscr{V}_{k,i,j,l}} \mathbf{p}\tag{77}$$

The down-sampled point cloud set at time t_k is defined as $\mathscr{Q}_k$, with the entire set of down-sampled point clouds represented as $\mathscr{P}_k^d$:

$$\mathscr{Q}_k = \left\{\mathbf{q}_{k,i,j,l} \middle| |\mathscr{V}_{k,i,j,l}| > 0\right\},\tag{78}$$

$$\mathscr{P}_k^d = \bigcup_{k \in \mathscr{K}} \{(t_k, \mathbf{q}) | \mathbf{q} \in \mathscr{Q}_k\}.\tag{79}$$

Let the toolpath having a P number of Cartesian points be expressed as $\mathscr{T}^t$ and defined as a reference point cloud in $\mathbb{R}^3$ as:

$$\mathscr{T}^t = \{\mathbf{t}_m m = 1, \ldots, P\}, \mathbf{t}_m = (x_m^t, y_m^t, z_m^t),\tag{80}$$

where $\mathbf{t}_m$ is a set of ordered tuples containing the Cartesian coordinates of the toolpath x_m^t, y_m^t, z_m^t. Since all point clouds in $\mathscr{P}_k^d$ correspond to the tool position, let the final point cloud be defined as $\mathscr{Q}_f = \{\mathbf{q}_k | (t_k, \mathbf{q}_k) \in \mathscr{P}_k^d\}$ such that it is completely aligned with $\mathscr{T}^t$. The transformation for aligning $\mathscr{Q}_f$ with $\mathscr{T}^t$ shall then be applied to all the remaining point clouds in $\mathscr{P}_k^d$ using Iterative Closest Point (ICP) registration. ICP finds a rigid transformation pair $(R^*, \mathbf{b}^*)$ where $R^* \in SO(3), \mathbf{b}^* \in \mathbb{R}^3$ to minimize the following objective function:

$$\min_{R^*, \mathbf{b}^*} \sum_{i=1}^{k_{\max}} \min_{\mathbf{t}_m \in \mathscr{T}} \| R^* \mathbf{q}_i + \mathbf{b}^* - \mathbf{t}_m \|^2.\tag{81}$$

For a maximum n_p number of iterations, the cost function $c^{(n)}(\mathbf{q}_i)$ is refined iteratively by means of correspondence and transformation update as:

$$c^{(n)}(\mathbf{q}_i) = \underset{\mathbf{t}_m \in \mathscr{T}}{\operatorname{argmin}} \| R^{(n)} \mathbf{q}_i + \mathbf{b}^{(n)} - \mathbf{t}_m \|,\tag{82}$$

$$\left(R^{(n+1)}, \mathbf{b}^{(n+1)}\right) = \underset{R, \mathbf{b}}{\operatorname{argmin}} \sum_{i=1}^{n_p} \| R\mathbf{q}_i + \mathbf{b} - c^{(n)}(\mathbf{q}_i) \|^2.\tag{83}$$

After computing the optimal rigid transformation pair $(R^*, \mathbf{b}^*)$ using ICP, all points cloud vectors are updated to form an aligned vector $\mathbf{q}_k^a$. Finally, all $\mathbf{q}_k^a$ are gathered to form a set of aligned point clouds $\mathscr{P}_k^a$ as follows:

$$\mathbf{q}_k^a = R^* \mathbf{q}_k + \mathbf{b}^* \forall (t_k, \mathbf{q}_k) \in \mathscr{P}_k^d,\tag{84}$$

$$\mathscr{P}_k^a = \left\{ (t_k, \mathbf{q}_k^a) k \in \mathscr{K} \right\}.\tag{85}$$

3.2. Point-cloud to mesh reconstruction

The final step is to convert point-clouds inside $\mathscr{P}_k^a$ into mesh surfaces using a TPS interpolation technique. Fig. 5 shows the point-cloud to mesh conversion process for the actual point cloud. TPS regression has proven to be a powerful method for modeling complex, non-linear surfaces from scattered point cloud data, as often encountered in single point incremental forming processes. This technique effectively minimizes bending energy while ensuring smoothness in the interpolated surface, making it particularly suitable for applications where maintaining the continuity of the surface is crucial. TPS regression can handle irregularly spaced data points, allowing for accurate surface reconstruction even in cases with significant variability in point density. Additionally, its flexibility in accommodating various boundary conditions makes it an ideal choice for modeling intricate shapes and surfaces that arise in advanced manufacturing and material forming techniques [59]. The objective is to find a function $f : \mathbb{R}^2 \to \mathbb{R}$ according to the

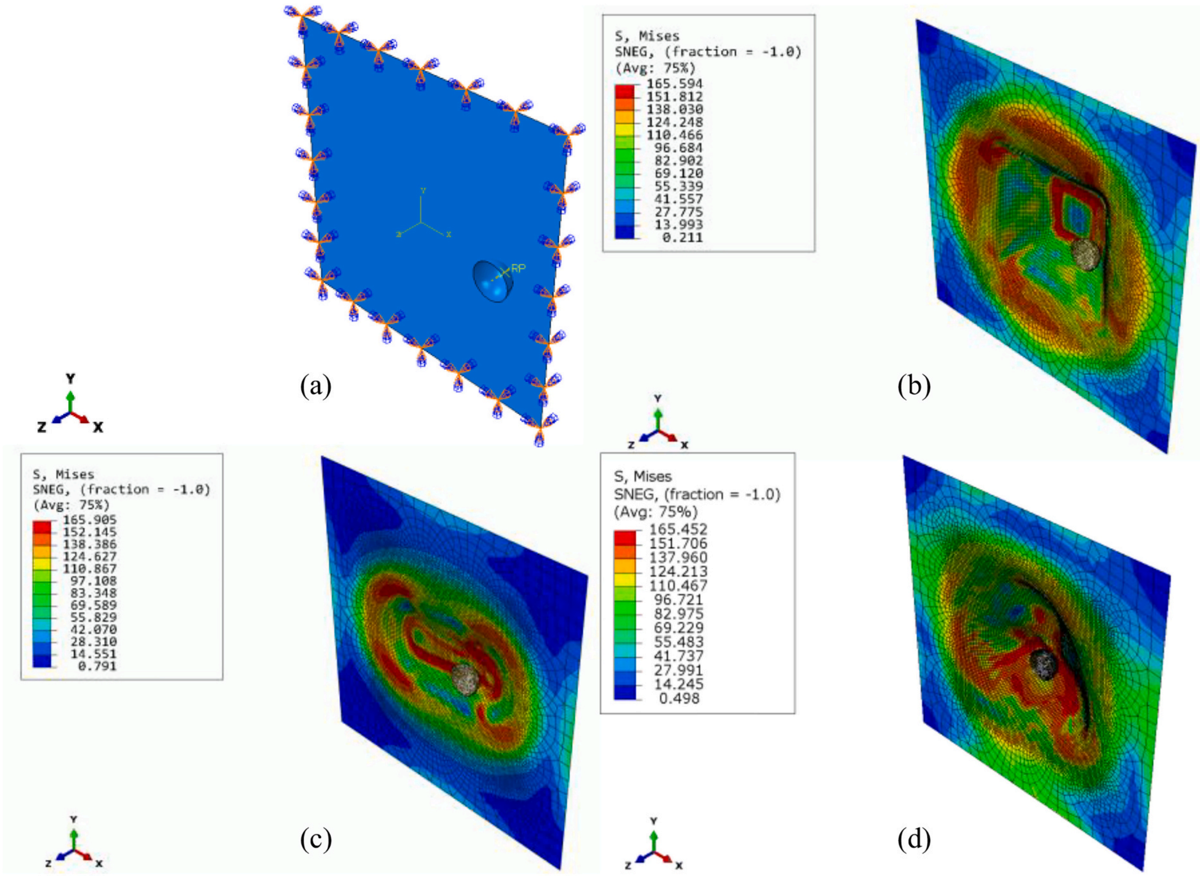


Fig. 6. (a) Boundary condition of SPIF process, (b) final shape of deformed metal sheet – Frustrum pyramid, (c) elliptical cone, (d) copper panels.

following formulation:

$$f_k(x,y) = a_{0,k} + a_{1,k}x + a_{2,k}y + \sum_{i,j \in \mathcal{I}_k} v_{ij,k} U\Big(\| (x,y) - \big(x_i, y_j\big) \| \Big), \tag{86}$$

where x_i, y_j are Cartesian coordinates in the $x-y$ plane for each $\mathscr{P}_k^a$, $U(r) = r^2 \log(r)$, $r = \| (x,y) - \big(x_i, y_j\big) \|$ is the Euclidean distance in the 2D plane, $v_{ij,k}$ are weights for nonlinear deformation and the constants $a_{0,k}$, $a_{1,k}$, and $a_{2,k}$ represent the affine part. For a unique z-axis value, z_i corresponding to each X_i, Y_i and a total number of N_p points in each point cloud, the interpolant satisfies the following conditions:

$$f\big(x_i, y_j\big) = z_{i,k} \text{ for } i,j = 1, \ldots, N_p. \tag{87}$$

To ensure uniqueness and control affinity,

$$\sum_{i,j \in \mathcal{I}_k} v_{ij,k} = 0, \sum_{i,j \in \mathcal{I}_k} v_{ij,k} x_i = 0, \sum_{i,j \in \mathcal{I}_k} v_{ij,k} y_j = 0. \tag{88}$$

The interpolant minimizes the bending energy, J_e

$$J_e = \iint_{\mathbb{R}^2} \left(\left(\frac{\partial^2 f}{\partial x^2} \right)^2 + 2 \left(\frac{\partial^2 f}{\partial x \partial y} \right)^2 + \left(\frac{\partial^2 f}{\partial y^2} \right)^2 \right) dxdy. \tag{89}$$

The family of TPS interpolants is represented by the set $\mathscr{F}_k^a$:

$$\mathscr{F}_k^a = \Big\{ (t_k, f_k) | k \in \mathscr{K}, f_k : \mathbb{R}^2 \rightarrow \mathbb{R}, f_k\big(x_i, y_j\big) = z_i, i,j \in \mathcal{I}_k \Big\}. \tag{90}$$

For a square grid $\mathscr{G}$ containing coordinate vectors $X_s = [x_1, x_2, \ldots, x_n]$ and $Y_s = [y_1, y_2, \ldots, y_n]$ are defined as:

$$x_i = x_{\min} + \frac{(x_{\max} - x_{\min})(i-1)}{n-1}, \forall i = 1, 2, \ldots, n, \tag{91}$$

$$y_j = y_{\min} + \frac{(y_{\max} - y_{\min})(j-1)}{n-1}, \forall j = 1, 2, \ldots, n, \tag{92}$$

$$\mathscr{G} = \Big\{ \big(x_i, y_j\big) x_j \in \mathbf{x}, y_i \in \mathbf{y}, i,j = 1, 2, \ldots, n \Big\}. \tag{93}$$

Consequently, at each time step k, Z_k is evaluated as:

$$Z_k \in \mathbb{R}^{n \times n}, Z_k(i,j) = f_k\big(x_j, y_i\big), i,j = 1, 2, \ldots, n. \tag{94}$$

The entire set of Z_k is finally represented by $\mathscr{Z}_k^a$ as:

$$\mathscr{Z}_k^a = \{ (t_k, Z_k) | k \in \mathscr{K}, Z_k \in \mathbb{R}^{n \times n}, Z_k(i,j) = f_k\big(x_i, y_j\big), i,j = 1, 2, \ldots n \}, \tag{95}$$

3.3. Toolpath synchronization and interpolation

Assuming that the point clouds and tool motion are recorded or sampled independently of each other, $\mathscr{Z}_k^a$ needs to be synchronized with $\mathscr{T}^t$. Moreover, it is also assumed that the surface in contact with the tool has the minimum value. Consequently, the minimum value $Z_{k,\min}$ is computed as:

Table 2
Abaqus simulation parameters.

Parameter	Value
Sheet metal	100 × 100 × 0.3 mm
Mesh size (Sheet – Forming Area)	0.7 mm
Mesh size (Sheet – Non-Forming Area)	5 mm
Mesh size (Tool)	0.35 mm
Element type	Shell (S4R)
Mass scaling factor	10,000
Contact method	Penalty
Interaction	Tangential behavior
Coefficient of friction	0.1

$$Z_{k,\min} = \min_{i,j} Z_k(ij),$$
$$(i_k j_k) = \operatorname*{argmin}_{i,j} Z_k(ij),$$
$$\left(x_{k,\min} y_{k,\min}\right) = \mathscr{G}(i_k j_k),$$

where (i_k, j_k) is the pair of indices at which Z_k attains its minimum value, while $\left(x_{k,\min}, y_{k,\min}\right)$ is the grid point corresponding to $Z_{k,\min}$ value. To this end, the reference or minimum point $\mathbf{p}_k$ becomes:

$$\mathbf{p}_k = \left\{x_{k,\min}, y_{k,\min}, Z_k(i_k, j_k)\right\}. \tag{97}$$

For each toolpath point $\mathbf{t}_m$, the Euclidean distance to $\mathbf{p}_k$ is calculated as:

$$d_m = \|\mathbf{t}_m - \mathbf{p}_k\|_2 = \sqrt{\left(x_m^t - x_{k,\min}\right)^2 + \left(y_m^t - y_{k,\min}\right)^2 + \left(z_m^t - Z(i_k, j_k)\right)^2}. \tag{98}$$

The index m^* is then computed to minimize d_m, with the corresponding point in the toolpath and time stamp denoted as $\mathbf{t}_k^{\min}$ and $t_k^{\min}$:

$$m^* = \operatorname*{argmin}_m d_m, \tag{99}$$

$$\mathbf{t}_k^{\min} = \left(x_{m^*}^t, y_{m^*}^t, z_{m^*}^t\right)^\top, \tag{100}$$

$$t_k^{\min} = t_{m^*}. \tag{101}$$

The synchronized toolpath and time stamps are represented as a set, $\mathscr{S}^s$:

$$\mathscr{S}^s = \left\{\left(t_k, \mathbf{t}_k^{\min}, t_k^{\min}\right) | k \in \mathscr{K}\right\}. \tag{102}$$

It is assumed that the total number of point clouds in $\mathscr{Z}_k^a$ is less than the total number of points in $\mathscr{T}^t$, i.e., $k < P$. The set $\mathscr{S}^s$ is interpolated using piecewise cubic Hermite interpolation (PCHIP). Let $x_{\text{itp}}(t)$, $y_{\text{itp}}(t)$, and $z_{\text{itp}}(t)$ be the PCHIP interpolation of $\left\{\left(t_m, x_m^{\text{tool}}\right) | m = 1, 2, \dots, P\right\}$, $\left\{\left(t_m, y_m^{\text{tool}}\right) | m = 1, 2, \dots, P\right\}$ and $\left\{\left(t_m, z_m^{\text{tool}}\right) | m = 1, 2, \dots, P\right\}$ respectively. For each query time $t_k^{\min}$ from $\mathscr{S}^s$, the interpolated tool path vector $\mathbf{t}_{\text{itp}}\left(t_k^{\min}\right)$ is computed as:

$$\mathbf{t}_{\text{itp}}\left(t_k^{\min}\right) = \left(x_{\text{itp}}\left(t_k^{\min}\right), y_{\text{itp}}\left(t_k^{\min}\right), z_{\text{itp}}\left(t_k^{\min}\right)\right)^\top. \tag{103}$$

Finally, $\mathscr{S}^{s,\text{itp.}}$ is the set containing t_k, $\mathbf{t}_{\text{itp.}}\left(t_k^{\min}\right)$ and $t_k^{\min}$:

$$\mathscr{S}^{s,\text{itp}} = \left\{\left(t_k, \mathbf{t}_{\text{itp}}\left(t_k^{\min}\right), t_k^{\min}\right) | k \in \mathscr{K}\right\}. \tag{104}$$

4. Simulation and experiment results

4.1. Training data generation via simulation and experiment results

High-fidelity finite element simulation is a critical step to understanding the overall mechanics and dynamics of SPIF. Abaqus is the preferred choice for FEM simulations due to its ability to handle large nonlinearities and provide different types of analysis, especially for ISF [60]. The material used is aluminum alloy 1050-O having a Young's modulus of 86,055 MPa and Poisson's ratio of 0.33. The material is assumed to be isotropic with a von Mises yielding criteria and inclusion of nonlinear geometry/effects. A variable meshing scheme has been adopted whereby a coarser mesh has been used in the non-forming region (in the vicinity of the sheet edge) while a finer mesh has been used in the forming region. As seen in Fig. 6, Encastre boundary conditions have been employed at the sheet edges. This condition fully constrains the nodes at the edges, preventing any displacement or rotation and thereby effectively immobilizing the specified nodes in all degrees of freedom. The simulation utilizes shell elements, specifically the S4R element type, which consists of four nodes and employs reduced integration. Shell elements are particularly effective for modeling thin structures, as they capture the bending behavior while reducing the computational cost compared to solid elements. The reduced integration technique minimizes the number of integration points, thereby enhancing computational efficiency while maintaining sufficient accuracy. Table 2 provides a summary of the simulation details. The penalty method is employed for contact interactions in the simulation. This approach allows for the enforcement of contact constraints by introducing a penalty stiffness that governs the contact forces between interacting bodies. The penalty method is advantageous in explicit simulations as it provides a straightforward way to model contact without the need for iterative solvers, which can be computationally expensive. Tangential behavior of contact interactions is specified, indicating that frictional effects will be considered during the simulation. This is important for accurately capturing the behavior of contacting surfaces, particularly in applications where slip and wear may influence performance. A lower value for the coefficient of friction of 0.1 is set for the contact interactions to simulate lubrication on the metal sheet.

As shown in Fig. 7 (a), a horizontal fixture, modelled and designed using SolidWorks 2022, was used in the experimental study. An equivalent simulation model of the setup was developed using Simscape, as seen in Fig. 7 (b). The sheet is gripped by a pair of clamping plates, one at the top and the other at the bottom. The plates and sheet are further clamped together by eight M8 screws. While the actual forming operation takes place within the 100 × 100 mm^2 square area, the sheet metal has a larger dimension relative to the forming area. This ensures sufficient friction between the sheet and clamping plates, preventing the sheet from slipping during the forming operation. The fixture consists of a 155 × 155 mm sheet metal, 0.3 mm thick, mounted on a 250 × 250 mm base plate and supported by 350 mm columns. The assembly is closed by 200 × 200 mm bottom and top plates. A SEAL 3D optical scanner, with its camera origin aligned with the center of the metal sheet, is placed directly beneath the sheet to dynamically capture and record the deformation. The specifications of the scanner are listed in Table 3.

Forming is accomplished by a rigid tool with a hemispherical end and diameter of 10 mm, as depicted in Fig. 7 (c). The tool is made of ASSAB XW42 high alloy tool steel with a Hardness Rockwell C (HRC) value of 56–62 and Young's modulus of 210 GPa. It is connected to a 6-axis Hypersen FT-060E force-torque sensor/load cell, which is then mounted to the end-effector of a 6-DOF Staubli industrial robot arm via a connecting flange. The Staubli robot arm has a work envelope of 190 mm to 610 mm with a maximum speed of 8 m/s and minimum repeatability of ±0.02 mm. Joint specifications of the Staubli robot manipulator are given in Table 4.

CAD assembly models of the Staubli robot, tool holder assembly and fixture are imported using the Simulink Simscape Multibody toolbox of MATLAB to create a dynamical model of the SPIF system. Simscape Multibody is a powerful tool for modeling and simulation of multi-body mechanical systems. It allows the creation of detailed models of complex mechanical assemblies, incorporating rigid and flexible bodies, joints, and actuators. This efficiently simulates the dynamic behavior of systems under various operating conditions and is particularly valuable in

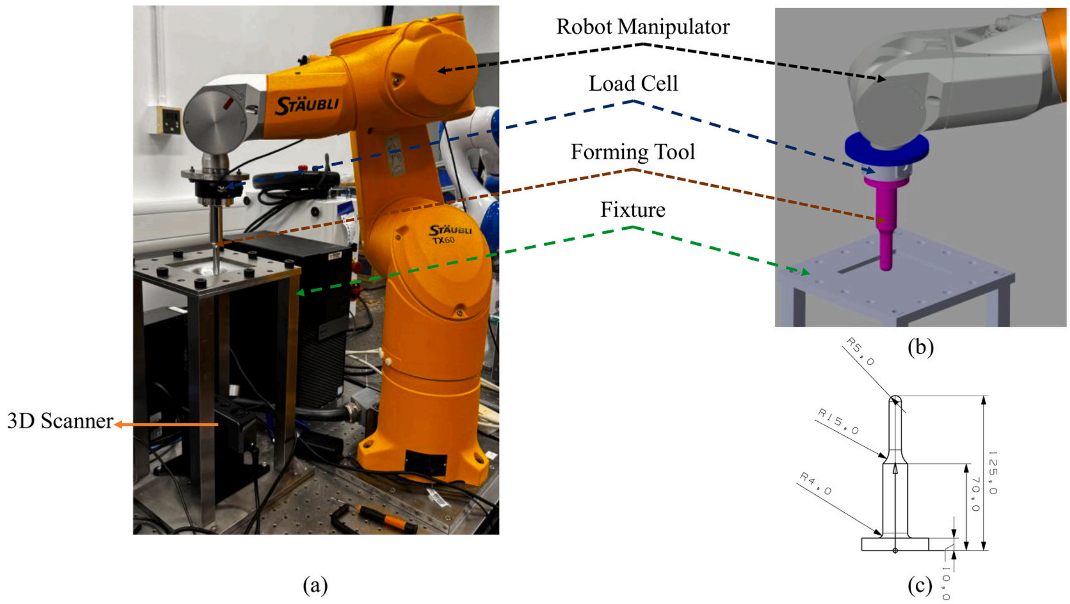


Fig. 7. SPIF experiment and simulation configuration: (a) experiment setup, (b) Simscape model, (c) tool dimensions.

Table 3
SEAL 3D optical scanner specifications.

Parameter	Value
Accuracy	0.01 mm
Resolution	0.05 mm
Frame rate	10 Frames Per Second
Working distance	180–280 mm
Field of view	100 × 75 mm

applications such as robotics, automotive engineering, and aerospace to conveniently analyze interactions between multiple moving parts for design optimization and performance. For conversion of task space coordinates of the trajectory, the inverse kinematics solver in the Robotics System Toolbox of MATLAB was used as it integrates seamlessly with Simulink. By defining the target pose of the tool, the joint configurations space of the Staubli TX-60 robotic arm was computed and implemented successfully. The joint space variables are interpolated using a cubic spline. In the following sections, complete details of the experimental procedure shall be presented.

4.2. Benchmark comparison using ILC-SSF

The performance of the DMDc-based toolpath optimization will be analyzed for three different shapes, as shown in Fig. 8:

Table 4
Joint specifications of Staubli robot manipulator.

Parameter	Joint 1 (θ_1)	Joint 2 (θ_2)	Joint 3 (θ_3)	Joint 4 (θ_4)	Joint 5 (θ_5)	Joint 6 (θ_6)
Range (°)	$-180 \leq \theta_1 \leq 180$	$-127.5 \leq \theta_2 \leq 127.5$	$-142.5 \leq \theta_3 \leq 142.5$	$-270 \leq \theta_4 \leq 270$	$-122.5 \leq \theta_5 \leq 132.5$	$-270 \leq \theta_6 \leq 270$
Nominal speed (°/s)	287	287	431	410	320	700
Maximum speed (°/s)	435	410	540	995	1065	1445
Angular resolution (°)	0.057×10^{-3}	0.057×10^{-3}	0.057×10^{-3}	0.114×10^{-3}	0.122×10^{-3}	0.172×10^{-3}

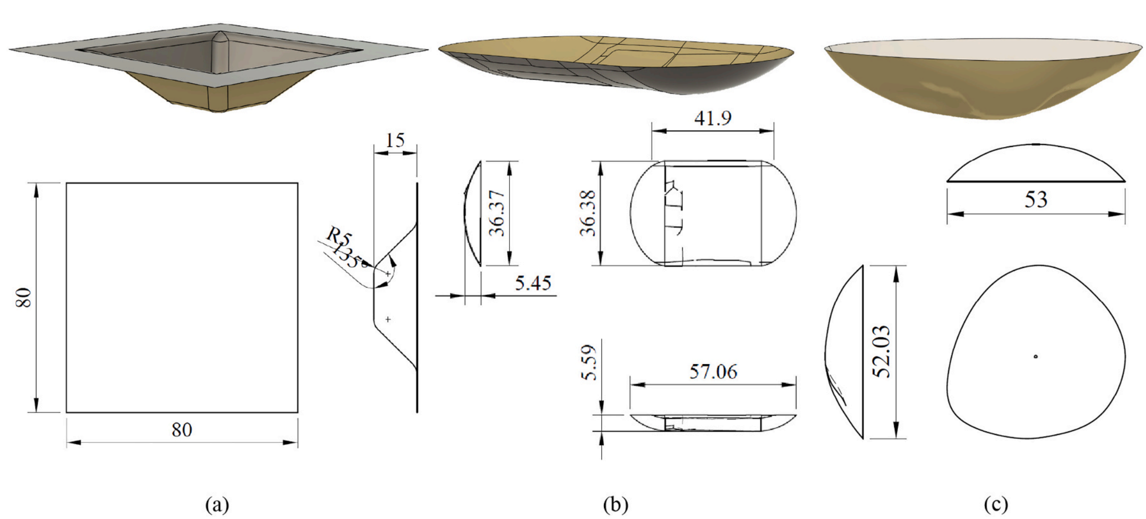


Fig. 8. Target shapes for SPIF performance benchmarking: (a) frustrum pyramid, (b) elliptical cone, (c) copper panel.

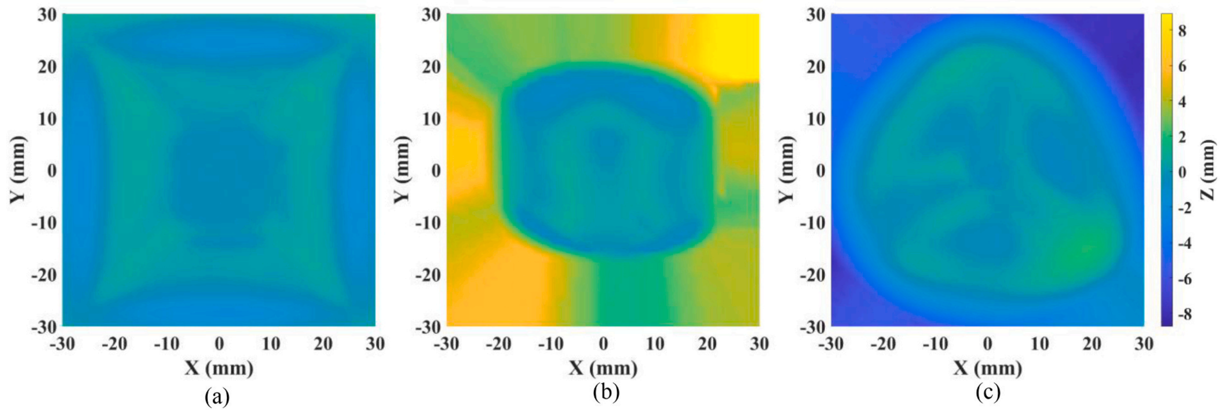


Fig. 9. Geometric deviation profiles for uncompensated toolpaths: (a) frustrum pyramid, (b) elliptical cone, (c) copper panel.

Table 5
Geometric error for uncompensated case.

Shape	Mean deviation (uncompensated) (mm)
Frustrum pyramid	0.5295
Elliptical cone	0.7884
Copper panel	1.2276

(a) Frustrum pyramid
(b) Elliptical cone
(c) Copper panel

The frustrum pyramid is a completely convex shape, while the elliptical cone is convex on its sides with non-convex features at its bottom. The copper panel is fully non-convex shape on its sides and bottom. The purpose of introducing shapes with varying features and geometries is to test and compare the generality and robustness of both the ILC-SSF and proposed DMDc-based toolpath optimization method. Before proceeding with the comparison, the geometric deviation of the geometries will be analyzed without applying any compensation. Autodesk PowerShape Ultimate 2025 was used for comparison and visualization of the error between the formed shape and target geometry. Fig. 9 shows the geometric error or deviation profiles for the uncompensated case, with the values provided in Table 5.

For performance benchmarking, the proposed DMDc-QP optimization is compared against the ILC-SSF baseline of Fischer et al. [61], as shown in Fig. 10. This benchmark was selected because it provides a direct comparison under the same control parameterization used here: in both cases, the in-plane geometry is treated as fixed while the out-of-plane height/depth is iteratively corrected. Specifically, ILC-SSF keeps x_j, y_j coordinates of the toolpath unchanged and updates only the tool depth z_j at each control point p via a proportional error law given as follows:

$$x_{j+1}(p) = x_j(p), \tag{105}$$

$$y_{j+1}(p) = y_j(p),$$

$$z_{j+1}(p) = z_j(p) + ke_j(p),$$

where $e_j(p) = z_{\text{ref}}(p) - z_m(j,p)$ is the error between the reference shape z_{ref} and measured shape z_m. The proportional factor $k \in (0,1]$ is a fractional value, with $k = 0.5$ usually chosen to mitigate between over-forming and under-forming. In a comparable way, DMDc identifies the measured shape using a height-field state $\mathbf{z}_{a,k}$ defined over a fixed (x,y) grid/mesh, and enables one-step prediction of the next shape and error as seen in Eq. (22). A constrained quadratic update $\mathbf{u}_k$ is computed via Eq. (54) to minimize $\| \mathbf{e}_{k+1} \|_2^2$ while regularizing deviations from the

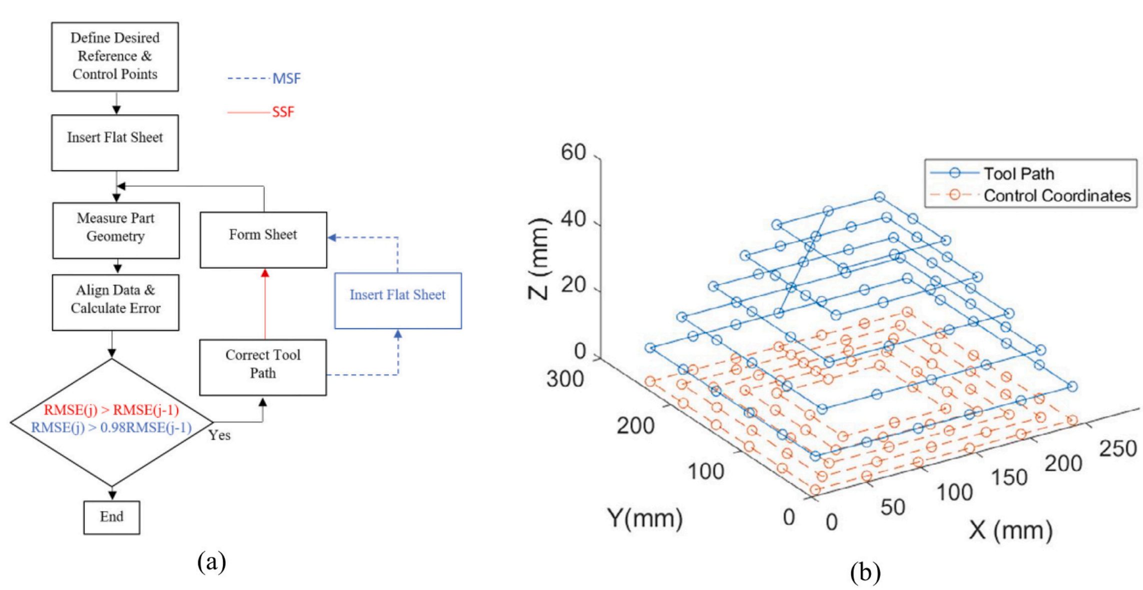


Fig. 10. ILC – SSF method for performance benchmark [61].

Table 6
Performance of ILC-SSF.

Shape	Mean deviation (uncompensated) (mm)	Mean deviation (compensated) (mm)	Change in deviation
Frustrum pyramid	0.5297	0.4342	−18.0%
Elliptical cone	0.7884	0.8832	+12.0%
Copper panel	1.2276	1.2867	+4.89%

nominal toolpath $\mathbf{u}_k^o$ using $\boldsymbol{\lambda}$. In practice, choosing λ_x, λ_y to be sufficiently large yields predominantly depth-direction corrections, consistent with the ILC-SSF actuation structure. Conceptually, the ILC-SSF update can be interpreted as a local steepest-descent step on the pointwise squared tracking error $\| e_j(p)\|^2$ under a diagonal approximation of the toolpath-to-geometry sensitivity. Conversely DMDc-QP framework leverages the learned influence in B^p together with convex optimization to produce a globally consistent, constrained correction. It is evident from Table 6 and Fig. 11 that, while ILC-SSF achieves an 18% error reduction for the frustrum pyramid from 0.5295 mm to 0.4342 mm, it fails to enhance the profiles of the elliptical cone and copper panel. Instead, the mean deviation for the elliptical cone shape increases by 23.5% from 0.7884 mm to 0.9740 mm, and that for the copper panel by 4.89% from 1.2276 mm to 1.2867 mm. Therefore, it can be concluded that the ILC-SSF method is not generalizable across different shapes and geometries.

4.3. Dynamic mode decomposition with control-based toolpath optimization

This section validates the learned DMDc representation of the SPIF process and summarizes its predictive fidelity prior to toolpath optimization. We first examine the structure of the identified operators to confirm that the extracted dynamics are physically plausible and stable under bootstrap resampling, then assess rollout prediction accuracy across benchmark geometries, and finally justify the low-rank truncation used in the reduced-order model.

To illustrate the dynamics captured by the identified DMDc model, Figs. 12 and 13 visualize the learned operators $A^p$and B^p, respectively,

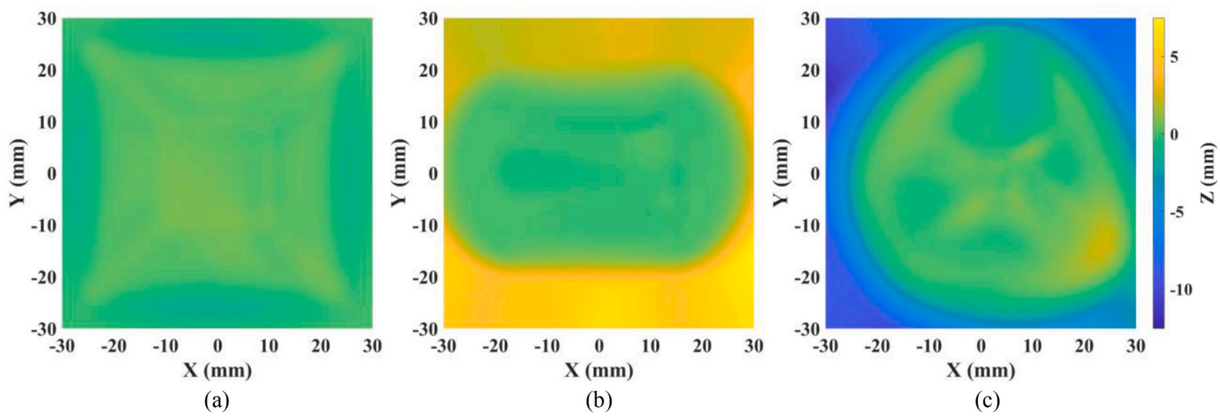


Fig. 11. ILC – SSF experimental results for error between target and actual shape: (a) frustrum pyramid, (b) elliptical cone, (c) copper panel.

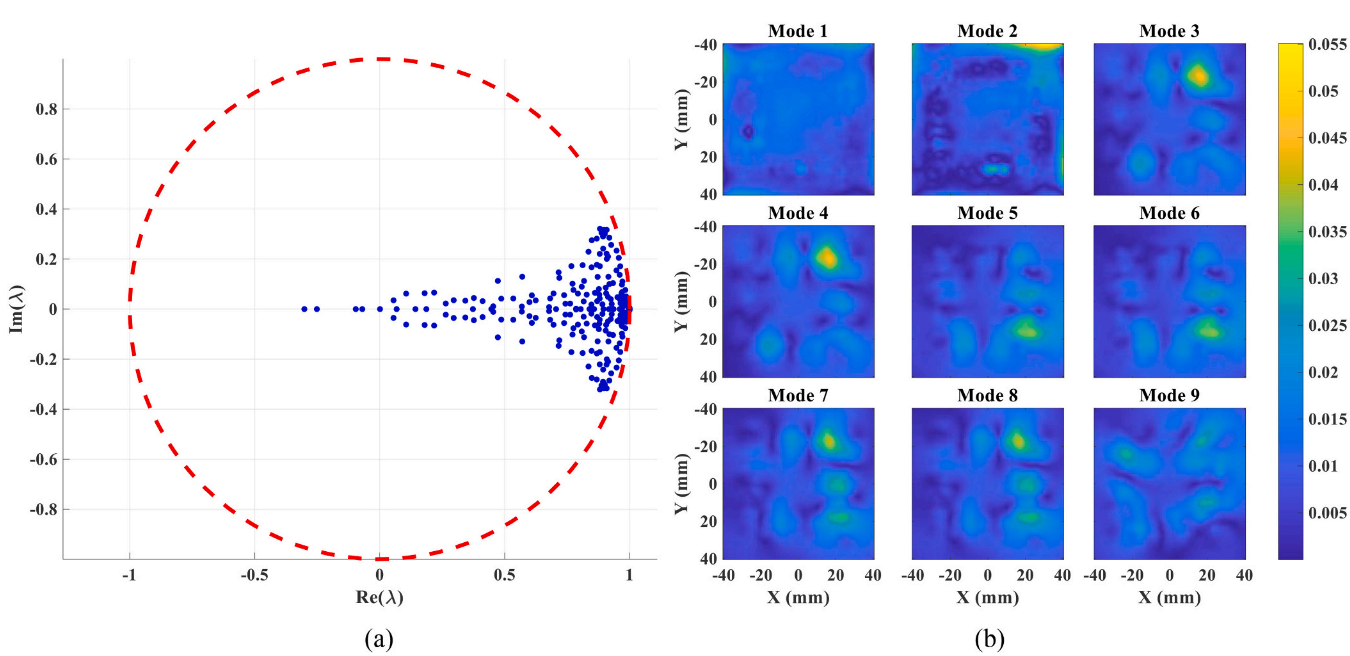


Fig. 12. Visualization of the control matrix A^p: (a) polar map of eigenvalues, (b) spatial DMD of modes 1–9.

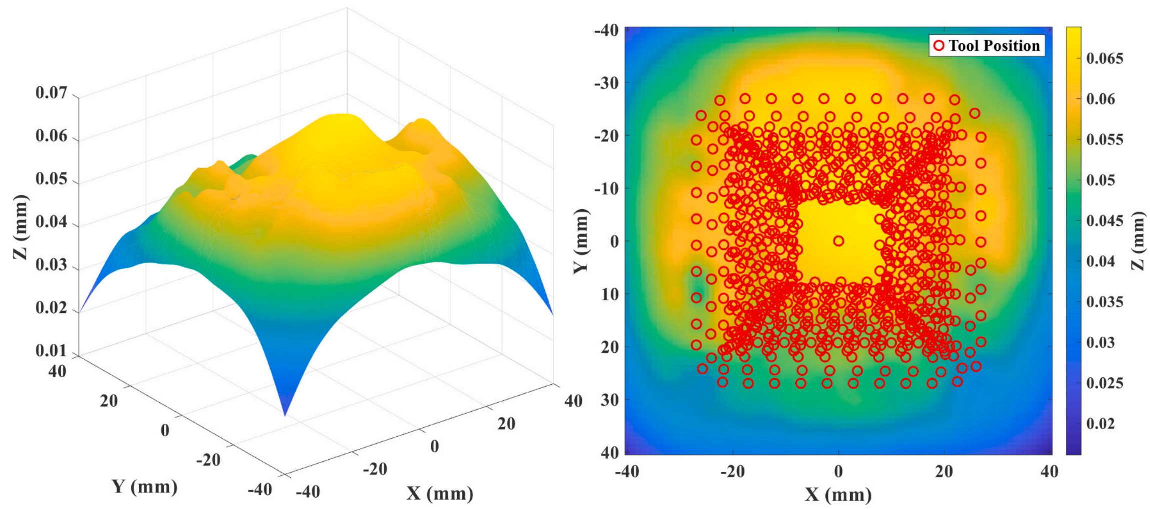


Fig. 13. Visualization of the control matrix B^p: (a) 3D surface plot, (b) 2D surface plot.

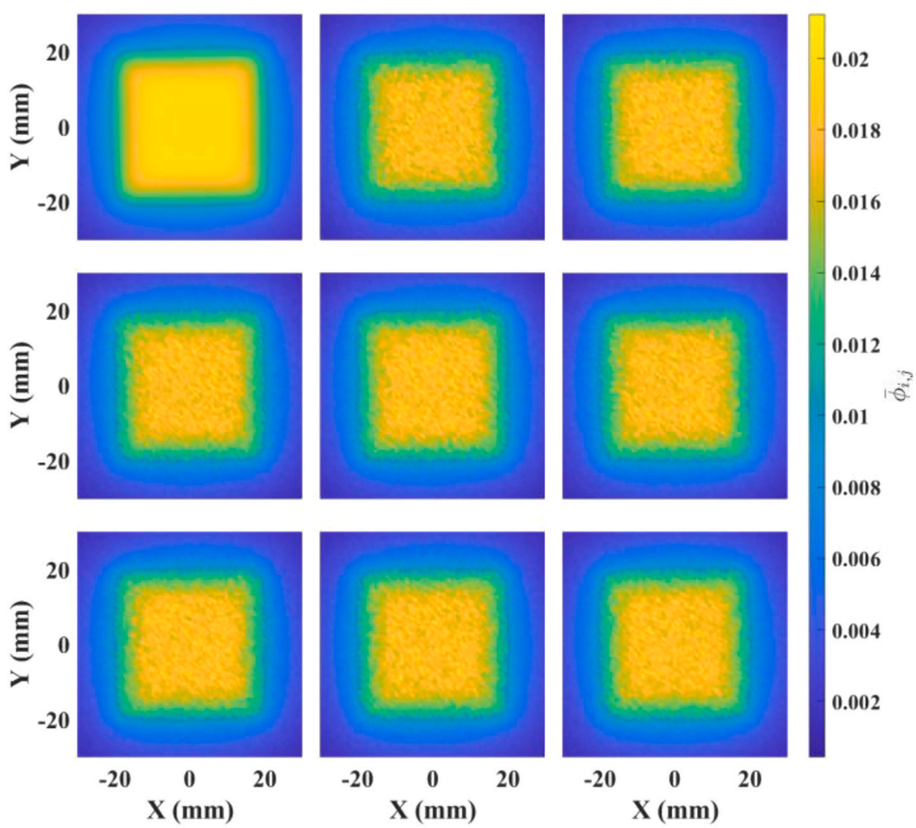


Fig. 14. Mean absolute amplitude spatial pattern of the top nine DMDc modes from bootstrap resampling, arranged in descending order of eigenvalue magnitude.

highlighting (i) the intrinsic deformation dynamics and (ii) the localized influence of the tool input on the global shape evolution. The eigenvalue plot in Fig. 12(a) shows the spectrum of A^p in the complex plane with a dense cluster near the unit circle. Eigenvalues satisfying $|\lambda| < 1$ correspond to decaying modes associated with dissipative effects (e.g., relaxation/springback), whereas eigenvalues on or near $|\lambda| = 1$ indicate persistent or marginally stable behaviors consistent with sustained bending and slow geometric propagation during forming. The spatial DMD modes in Fig. 12(b) (shown as absolute amplitudes and ordered to emphasize dominant dynamics) exhibit localized structures near the tool-affected region together with outward propagation patterns across the sheet, which is consistent with localized forcing and global deformation transport. Complementarily, the spatial influence of the tool input on the learned state dynamics is summarized by B^p in Fig. 13. The 3D visualization in Fig. 13(a) reveals a pronounced localized peak in the vicinity of the tool-contact region with rapid decay toward the periphery, while the 2D map in Fig. 13(b) shows the same structure over the mesh grid. The asymmetric ridge-like pattern is consistent with the directionality and nonuniform dwell density induced by the spiral toolpath, indicating that the identified control influence is strongest where the tool interaction is most concentrated.

The stability of the extracted coherent structures is evaluated via

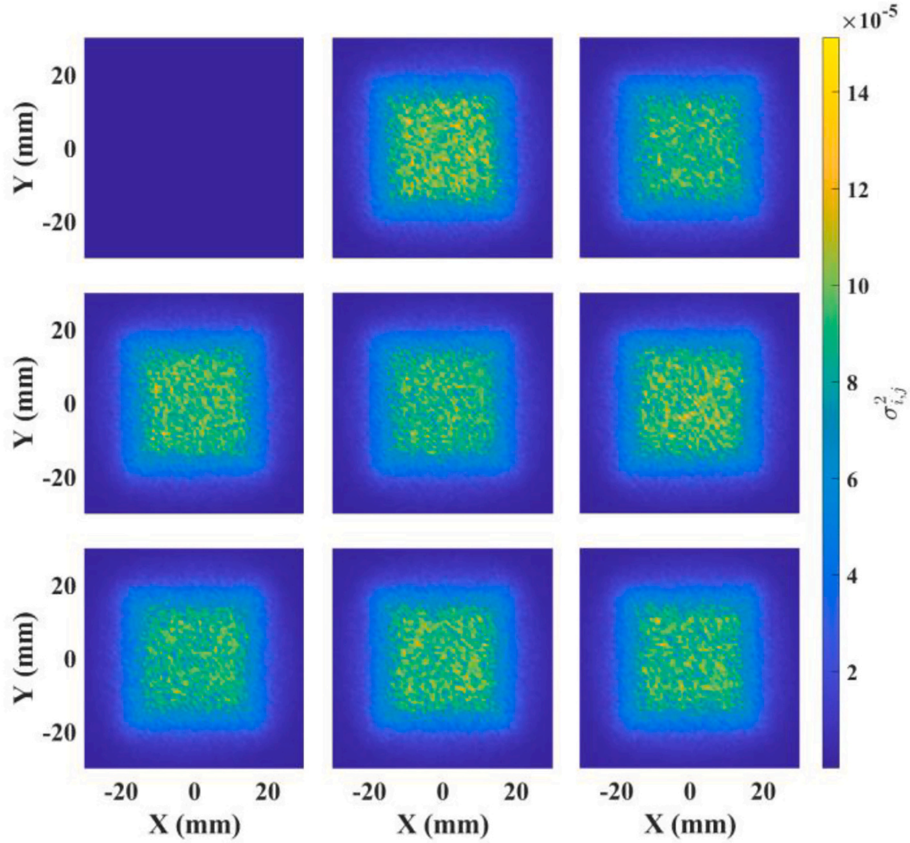


Fig. 15. Variance spatial pattern of the top nine DMDc modes from bootstrap resampling, arranged in descending order of eigenvalue magnitude.

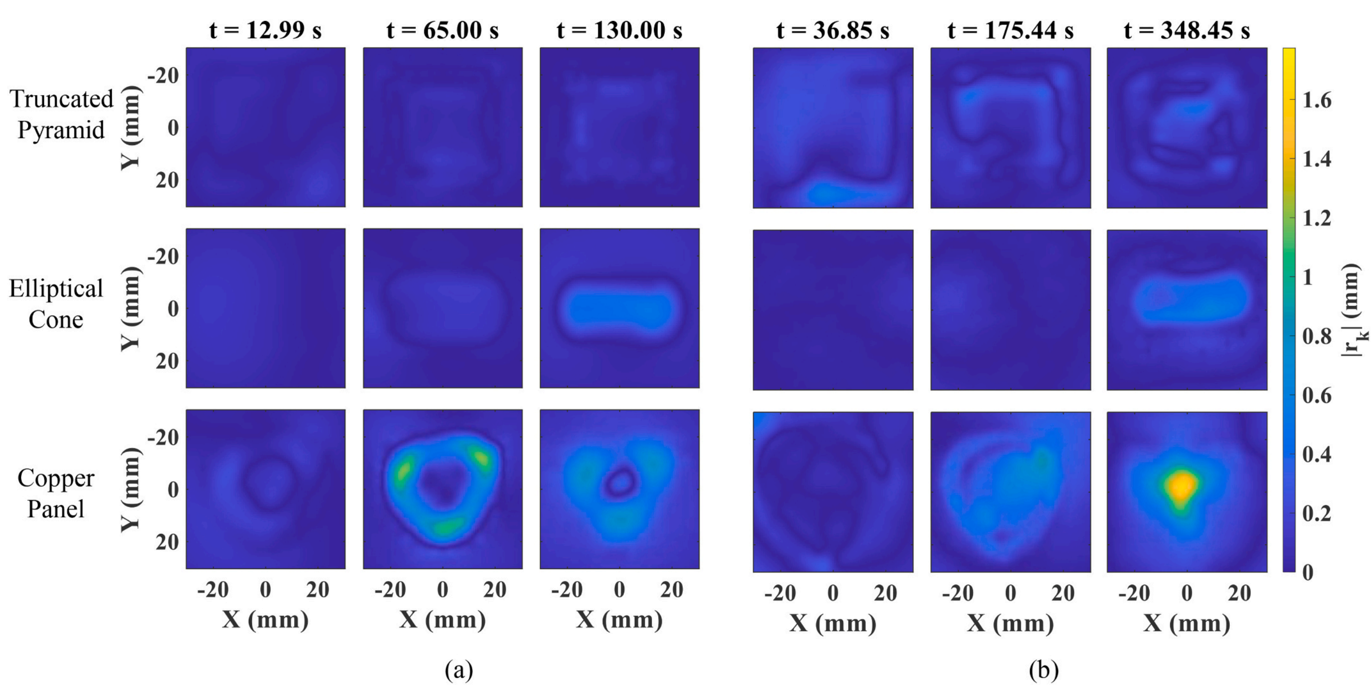


Fig. 16. Absolute DMDc rollout prediction error maps $|\mathbf{r}_f|$ for three geometries (rows) at 10%, 50%, and 100% of total forming time (columns), shown for (a) simulation and (b) experiment.

bootstrap resampling. Fig. 14 presents the spatial distributions of the mean of bootstrapped modes, whereby each mode exhibits a radially symmetric pattern centered at the origin, with peak values around 0.02 in the central region, gradually decaying toward the edges. This structure indicates coherent dynamic behaviors localized in the middle of the truncated pyramid mesh, likely representing fundamental deformation

Table 7
Final-snapshot prediction error under fixed-model initial-condition noise.

Shape	σ (mm)	RMSE (simulation) (mm)	MAE (simulation) (mm)	RMSE (experiment) (mm)	MAE (experiment) (mm)
Frustrum pyramid	$\sigma = 0$	0.1161	0.0791	0.1031	0.0710
	$\sigma = 0.25$	0.3802	0.3046	0.1026	0.0703
	$\sigma = 0.5$	0.6918	0.5720	0.1390	0.1057
	$\sigma = 1$	1.3250	1.1156	0.2523	0.2035
Elliptical cone	$\sigma = 0$	0.4207	0.3009	0.1740	0.1026
	$\sigma = 0.25$	0.3395	0.3017	0.1331	0.1159
	$\sigma = 0.5$	0.4387	0.4086	0.1932	0.1626
	$\sigma = 1$	0.8739	0.7646	0.4159	0.3875
Copper panels	$\sigma = 0$	0.7602	0.6353	0.4225	0.2997
	$\sigma = 0.25$	0.7622	0.6367	0.6465	0.4843
	$\sigma = 0.5$	0.7643	0.6380	0.8743	0.6702
	$\sigma = 1$	0.7684	0.6407	1.3334	1.0433

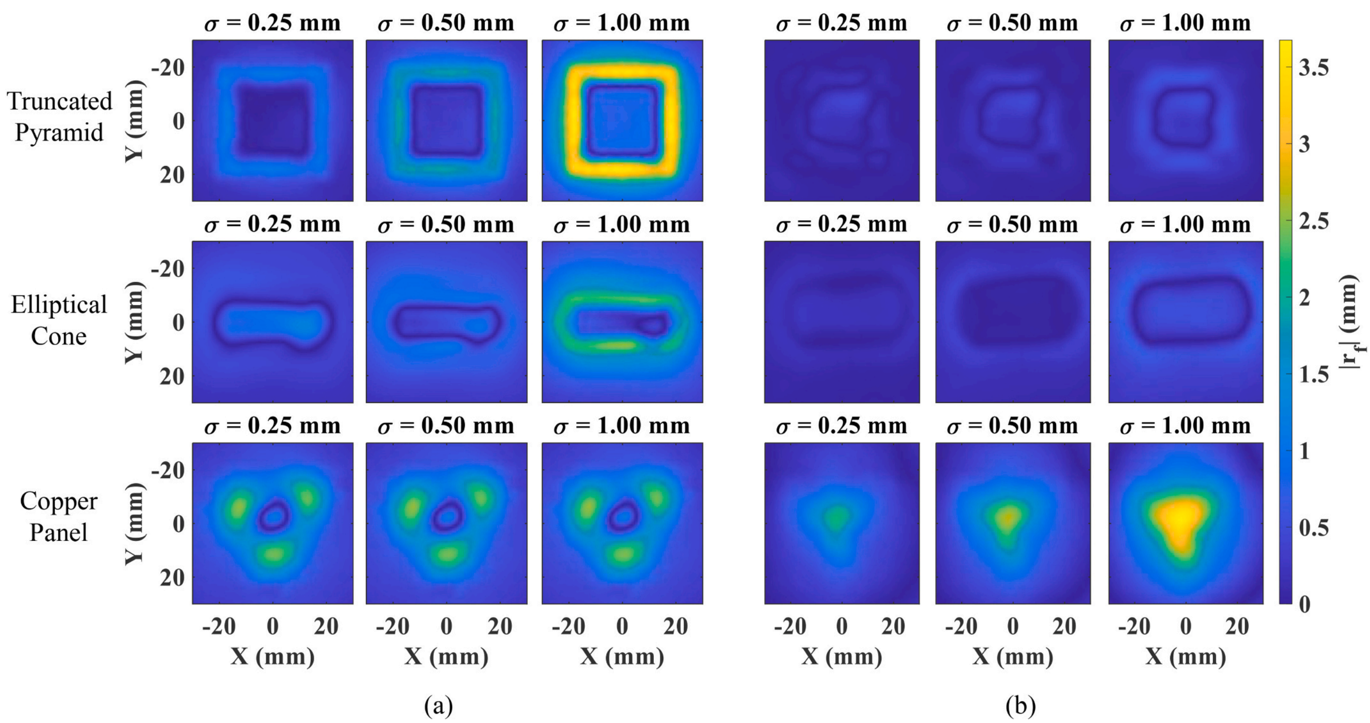


Fig. 17. Fixed-model initialization-noise sensitivity (final snapshot). Absolute final-snapshot residual maps $|\mathbf{r}_f|$ (mm), where $\mathbf{r}_f = \mathbf{z}_{a,f} - \widehat{\mathbf{z}}_{a,f}$, obtained by perturbing only the initial reconstructed height field with i.i.d. Gaussian noise of standard deviation σ. Columns show $\sigma \in \{0.25, 0.50, 1.00\}$ mm; rows correspond to the Frustrum Pyramid, Elliptical Cone, and Copper Panels. Panel (a) shows simulation; panel (b) shows experiment. RMSE/MAE in each subplot summarize the same $|\mathbf{r}_f|$ field over all mesh nodes. (The $\sigma = 0$ baseline is reported in Table 7)

modes in the Z-coordinates. Lower-order modes (e.g., Mode 1) display slightly higher central intensities and broader spreads compared to higher modes, suggesting a hierarchy where primary dynamics dominate the core while secondary modes refine edge or peripheral contributions. The consistency across modes highlights the low-rank nature of the system, with overlapping spatial features implying potential mode interactions. Meanwhile, Fig. 15 shows the variance distributions of bootstrapped modes, with maximum values on the order of 1.4×10^{-5} confined to the central hotspots. These variances are exceedingly low relative to the mean amplitudes, demonstrating minimal sensitivity to resampling noise. The patterns mirror the profiles of Fig. 14 but with diminishing intensity for higher modes, further affirming the stability of the DMDc extraction process. Such low uncertainty validates the bootstrapping framework for confidence estimation in coherent structures, particularly in noisy or perturbation-prone datasets. These spatial visualizations underscore the robustness of the identified modes, supporting their application in reduced-order modeling and control strategies for the structural system.

Predictive fidelity is next assessed using rollout error maps. Fig. 16 presents absolute DMDc rollout prediction error maps $|\mathrm{r}_k|$ for three geometries at 10%, 50%, and 100% of total forming time, annotated with corresponding RMSE and MAE values. In simulation, errors remain low and spatially uniform for convex shapes (Truncated Pyramid: ~0.11/0.08 mm RMSE/MAE). Modest growth occurs in non-convex cases, with Copper Panels peaking at 0.86/0.58 mm mid-process before decreasing to 0.76/0.64 mm at the end of forming. Experimental results mirror these spatial patterns but at approximately 1.5-2 times higher magnitudes, likely due to unmodeled disturbances. Across both datasets, errors localize near high-curvature regions, indicating where unresolved interactions exceed the representational capacity of the selected observable/basis. Overall, these results support the use of DMDc as an empirically validated lifted linear surrogate for prediction within the tested operating envelope, while transparently indicating regimes (strongly non-convex geometries) where residual nonlinearities become more pronounced. This framing aligns with prior demonstrations of DMD applied to deformation/displacement fields in solids, including reduced-order prediction of nonlinear elastic large-deformation responses and analysis of elastic–plastic transitions using deformation

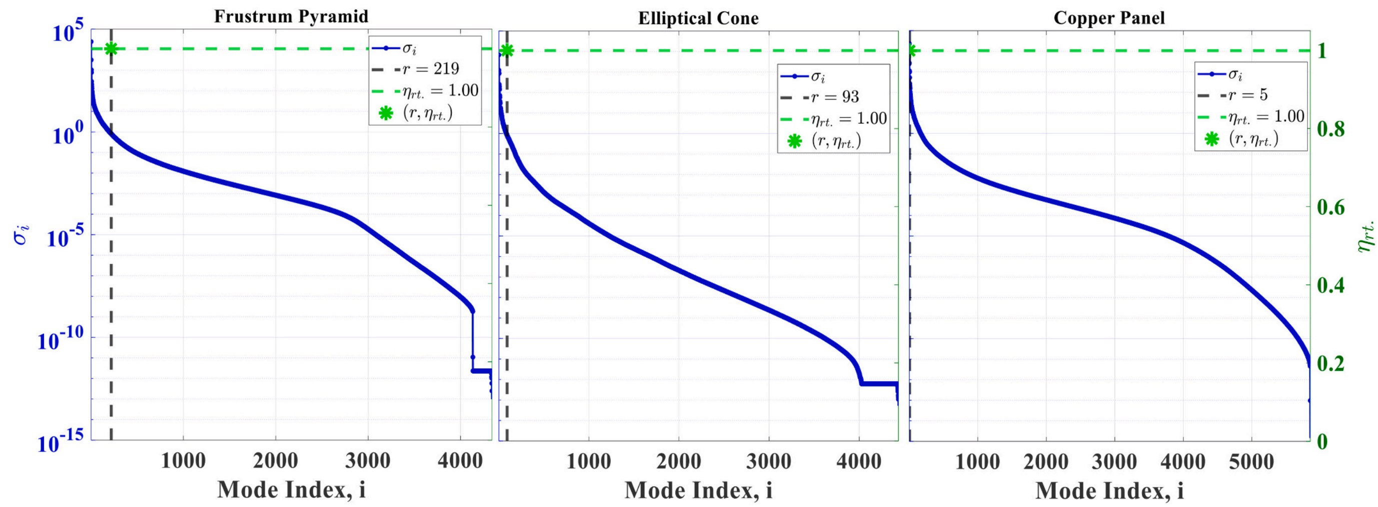


Fig. 18. Singular-value spectra and cumulative energy retention for DMDc rank truncation (Simulation data).

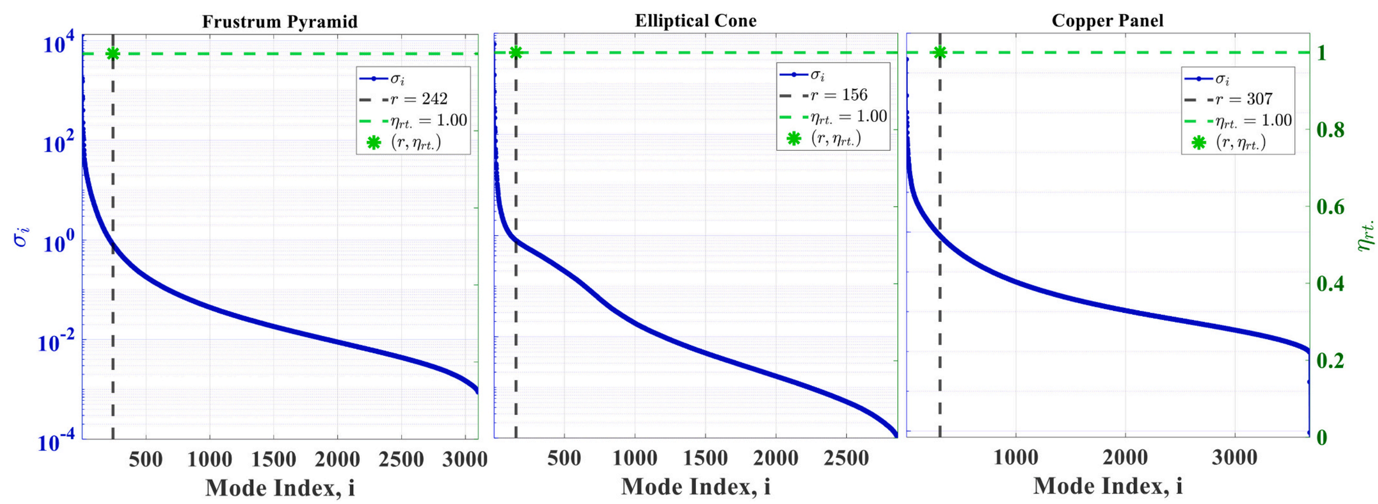


Fig. 19. Singular-value spectra and cumulative energy retention for DMDc rank truncation (Experimental data).

Table 8
DMDc truncation settings and resulting retained order for each shape, simulation and experiment datasets.

Shape	r (simulation)	$\eta_{rt.}$ (simulation)	r (experiment)	$\eta_{rt.}$ (experiment)
Frustrum pyramid	219	95%	242	92.2%
Elliptical cone	93	97.9%	156	94.6%
Copper panels	5	99.9%	307	91.7%

snapshots [44,45]. While Fig. 16 validates rollout accuracy within the tested envelope, the present study does not include a systematic benchmark against alternative predictors. Future work will therefore perform controlled comparisons against (i) lower-order linear baselines such as POD–Galerkin or subspace-identified LTI models using the same observables and (ii) representative nonlinear predictors such as kernel regressors or neural sequence models under matched training data, model selection procedures, and evaluation metrics.

Table 7 summarizes fixed-model initialization-noise sensitivity at the final snapshot using scalar RMSE/MAE metrics. In simulation, the Frustrum Pyramid exhibits strong amplification of initialization perturbations, with RMSE increasing from 0.116 mm at $\sigma = 0$ to 1.325 mm at $\sigma = 1.0$ mm, while the Elliptical Cone shows a more moderate increase (0.421 mm to 0.874 mm). In contrast, the Copper Panels

Table 9
Performance of DMDc-based feedforward optimization.

Shape	Mean deviation (uncompensated) (mm)	Mean deviation (compensated) (mm)	Change in deviation	Number of trajectory points	Execution time (s)
Frustrum pyramid	0.5295	0.3871	−26.8%	4343	73.0966
Elliptical cone	0.7884	0.2465	−68.7%	4437	101.8317
Copper panels	1.2276	0.8338	−32.1%	5856	99.0042

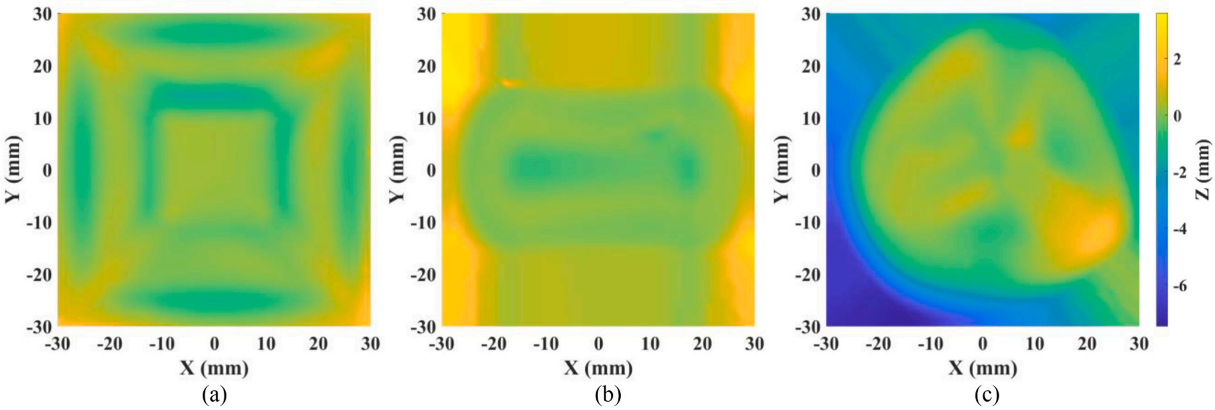


Fig. 20. DMDc-based toolpath compensation experimental results for error between target and actual shape: (a) frustrum pyramid, (b) elliptical cone, (c) copper panel.

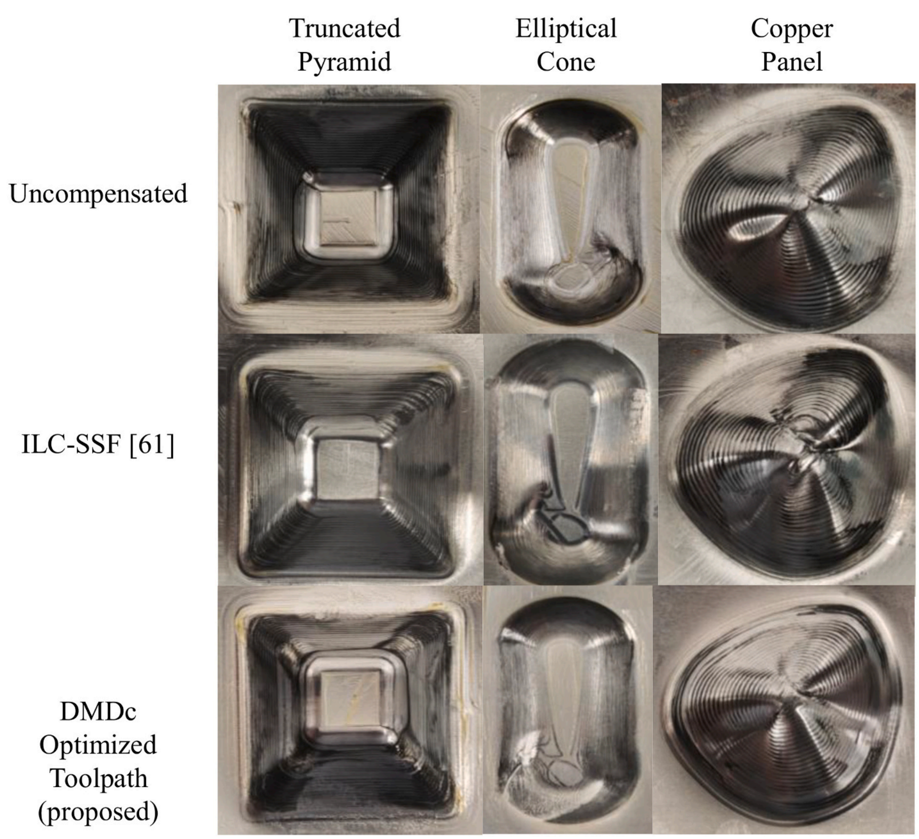


Fig. 21. Comparison of experiment specimens for uncompensated, ILC-SSF compensated and DMDc optimized toolpaths.

simulation case is essentially σ-invariant (RMSE $\approx$0.76 mm across all σ), indicating that its final error is dominated by a geometry-dependent floor rather than by uncertainty in the initial reconstructed surface. Experimental trends are generally monotone with σ, with the largest degradation again occurring for Copper Panels (RMSE 0.423 mm at $\sigma = 0$ rising to 1.333 mm at $\sigma = 1.0$ mm), whereas the Frustrum Pyramid remains comparatively robust (RMSE 0.103 mm to 0.252 mm). The small non-monotonic change observed for the Elliptical Cone at $\sigma = 0.25$ mm is consistent with incidental cancellation between the baseline residual and the perturbation direction used and should not be interpreted as an improvement in expected performance. Fig. 17 complements these scalars by localizing where the final error accumulates on the surface. In simulation, the Frustrum Pyramid and Elliptical Cone develop pronounced boundary/side-wall error bands that intensify with σ, consistent with error growth driven by initialization uncertainty, whereas the Copper Panels maps retain a persistent three-lobed structure with nearly unchanged magnitude across σ, visually confirming a dominant model-mismatch pattern. In experiment, the Copper Panels maps evolve into a strong central hotspot as σ increases, explaining the sharp rise in RMSE/MAE, while the pyramid and cone retain weaker, more spatially diffuse structures. Overall, the combined table-and-map evidence separates two regimes: bias-limited cases, where errors are governed by a repeatable spatial pattern and improvements in initialization noise yield limited benefit, and noise-limited cases, where final-shape accuracy degrades primarily through amplification of initial reconstruction/registration uncertainty, notably Copper Panels in experiment, and the pyramid/cone in simulation. This distinction clarifies when effort should be directed toward improved sensing/

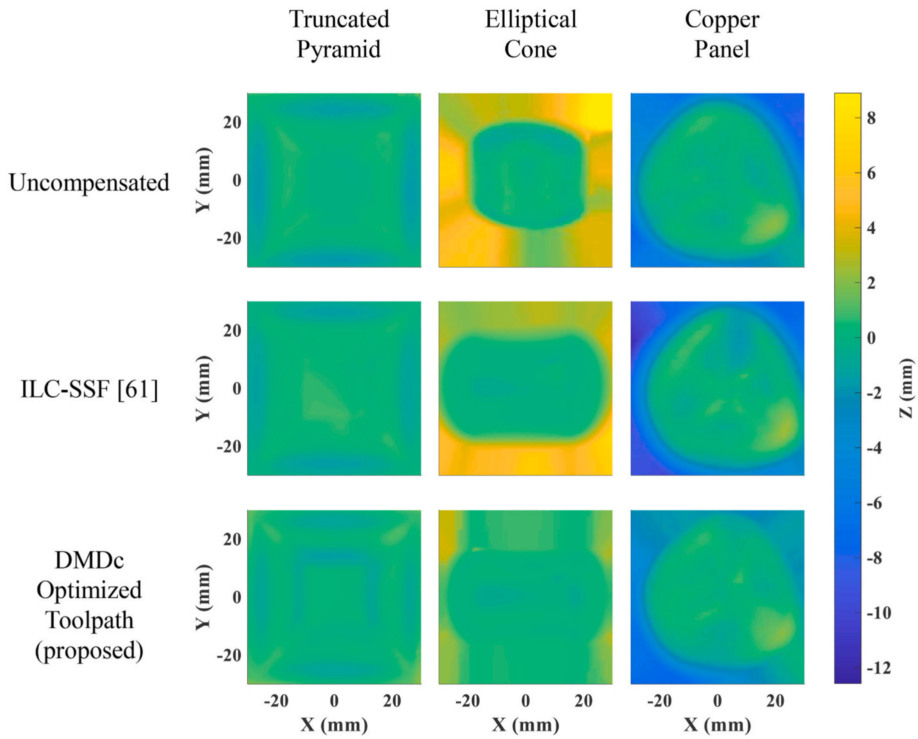


Fig. 22. Error between target shape and experimental final shape for uncompensated, ILC-SSF compensated and DMDc optimized toolpaths.

Table 10
Summary of geometric deviation metrics (mm). Values are the metric computed on the full dataset with 95% percentile block-bootstrap confidence intervals ($B = 1000$, $\ell = 5$ mm).

Shape	Method	n_p	δ_E (mm)	δ_R (mm)	P_{95}
Frustrum pyramid	Uncompensated	29,389	0.530 [0.481, 0.582]	0.705 [0.650, 0.760]	1.470 [1.381, 1.558]
	ILC-SSF	30,811	0.434 [0.397, 0.474]	0.534 [0.488, 0.582]	1.106 [1.001, 1.183]
	DMDc Optimized Toolpath	31,517	**0.387** [0.355, 0.421]	**0.489** [0.449, 0.529]	**0.934** [0.866, 1.009]
Elliptical cone	Uncompensated	27,692	0.788 [0.637, 0.969]	1.421 [1.156, 1.699]	3.607 [2.869, 4.324]
	ILC-SSF	29,135	0.883 [0.686, 1.091]	1.561 [1.264, 1.823]	3.882 [3.062, 4.523]
	DMDc Optimized Toolpath	28,407	**0.247** [0.197, 0.303]	**0.434** [0.332, 0.534]	**0.871** [0.625, 1.226]
Copper Panel	Uncompensated	32,433	1.228 [0.998, 1.507]	2.124 [1.795, 2.455]	5.421 [4.620, 6.002]
	ILC-SSF	33,075	1.821 [1.481, 2.205]	3.097 [2.623, 3.584]	7.739 [6.620, 8.687]
	DMDc Optimized Toolpath	32,672	**0.834** [0.668, 1.020]	**1.458** [1.158, 1.756]	**3.662** [2.401, 4.730]

registration versus toward improving the model representation.

The reduced-order truncation is justified through the singular-value spectra of the augmented snapshot matrix Ω. Figs. 18 and 19 show a rapid, multi-order-of-magnitude decay observed across all geometries justifies the low-rank truncation strategy. For simpler geometries, such as the Truncated Pyramid, the spectrum drops precipitously, allowing over 90% cumulative energy retention $\eta_{rt.}$ with a compact set of modes. In contrast, the Copper Panel spectra, particularly the experimental data, exhibit a relatively 'longer tail.' As detailed in Table 8, while the idealized simulation data for the Copper Panels collapses onto a highly compact basis $r = 5$, the experimental counterpart necessitates a higher-dimensional basis $r = 307$ to maintain comparable reconstruction fidelity. This discrepancy signifies that localized physical nonlinearities and measurement noise in the real-world process require a more complete modal representation than numerical models. This energy-based rank selection ensures that the methodology remains scalable: it automatically prioritizes operational simplicity when the data is low-dimensional (as seen in the simulation cases) while maintaining modeling completeness when real-world complexity demands higher dimensionality.

Comparative statements from this point onward are restricted to the experimental conditions evaluated in this study (material, thickness, tooling, boundary conditions, scan/registration pipeline, and the three benchmark geometries) and to the specific baseline implemented here (ILC-SSF). Accordingly, the reported improvements should be interpreted as performance gains under the tested SPIF setup and toolpath parameterization, rather than a universal dominance over all ILC or model-based controllers in other operating regimes. To ensure a fair comparison, we distinguish offline preparation cost from per-part iteration cost. All methods are evaluated using the same measurement/registration pipeline and the same geometric-deviation export from PowerShape. The ILC-SSF baseline uses an iterative update that, by design, typically requires repeated forming iterations to reduce error. In contrast, the proposed DMDc pipeline requires offline data processing and model identification, followed by a single constrained optimization to compute a compensated toolpath that can be executed without additional physical trial iterations. Practically, this limits compensation to one nominal data-collection step per geometry (experiment) or a single dataset (simulation), thereby reducing material waste and experimental time relative to multi-iteration calibration approaches. Table 9 therefore summarizes the achieved reductions in geometric deviation obtained using the DMDc-optimized toolpath relative to the uncompensated and ILC-SSF cases under the above controlled experimental setup.

Journal of Manufacturing Processes 167 (2026) 507–531

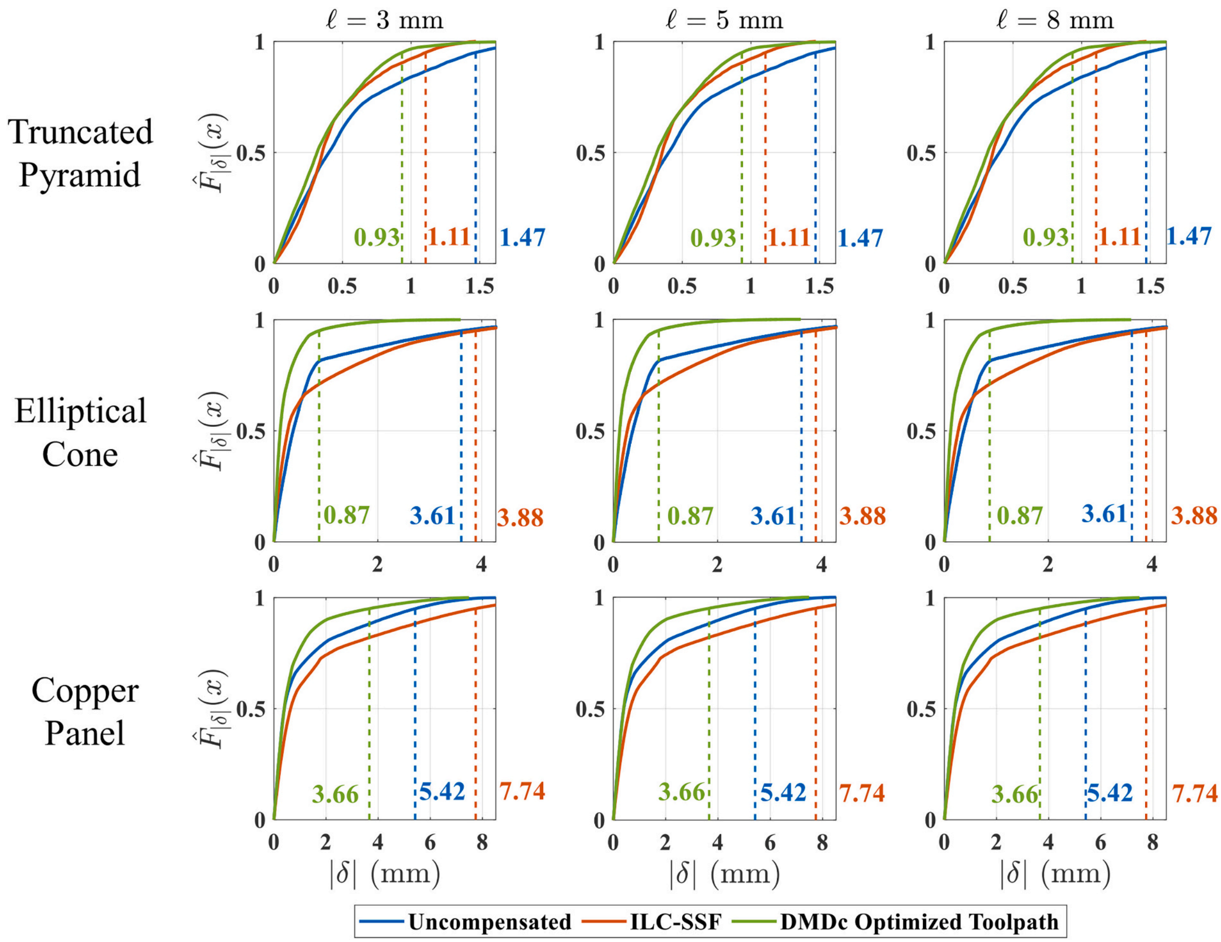


Fig. 23. Empirical CDFs $\widehat{F}_{|\delta|}(x)$ of absolute surface deviation $|\delta|$ (mm) for the three geometries (rows: Truncated Pyramid, Elliptical Cone, Copper Panel) and spatial block bootstrap tile sizes (columns: $\ell = 3, 5, 8$ mm). Curves correspond to Uncompensated, ILC-SSF, and DMDc-optimized toolpath. Dashed vertical lines mark the method-specific P_{95} of $|\delta|$ (values annotated).

As evidenced from Table 9, the mean deviation in the frustrum pyramid decreases by approximately 26.8% from 0.5295 mm to 0.3871 mm, while the standard deviation decreases by 35.7% from 0.4636 mm to 0.2982 mm. Similarly, the error for the elliptical cone shows a marked improvement, with a 60.5% reduction in mean deviation from 0.7884 mm to 0.3115 mm and 69.8% reduction in standard deviation from 1.1824 mm to 0.3573 mm. Finally, for the non-convex copper panel, a 32.1% reduction is achieved, bringing the deviation down from 1.2276 mm to 0.8338 mm. Fig. 20 shows a comparison of the error profile before and after the compensations. As evidenced from Fig. 17 (c), the geometric error from the formed profile of the copper panel is greater in comparison to the frustrum pyramid and elliptical cone. This can be attributed to the multiple convex features of copper panel geometry. However, uncertainties in the DMDc model limit the effectiveness of DMDc-based toolpath optimization. To achieve further reductions in geometric error, a feedback or combined feedforward-feedback approach is required. An overall comparison of the DMDc optimized toolpath specimens relative to the uncompensated and ILC-SSF-fabricated specimens is shown in Fig. 21 and Fig. 22 respectively. The uncompensated approach results in pronounced deviations, often exceeding 5 mm in magnitude with irregular patterns across all shapes, leading to suboptimal surface quality.

The comparisons above establish that DMDc reduces average deviation and improves the spatial uniformity of the error field. However, manufacturing acceptability is often governed by upper-tail deviations (localized outliers) rather than the mean alone, and the finite sampling and spatial correlation require explicit uncertainty quantification. Moreover, because signed mean deviation (Table 9) can partially mask large local errors through sign cancellation, the remaining analysis focuses on magnitude-based metrics of $|\delta|$. Accordingly, Table 10 reports three complementary summary metrics, i.e., δ_E, δ_R, and P_{95} together with 95% confidence intervals obtained from a spatial block bootstrap ($B = 1000$) using a nominal block size of $\ell = 5$ mm. These tabulated estimates provide a quantitative baseline for comparing methods across geometries and confirm that the DMDc-optimized toolpath yields the lowest metric values overall, including substantial reductions in the upper-tail statistic P_{95}. To visualize how these improvements arise across the entire surface-error distribution, Fig. 23 presents empirical CDFs of $|\delta|$ for each geometry (rows) and block size $\ell \in \{3, 5, 8\}$ mm (columns), with method-specific P_{95} markers highlighting tail behavior. Complementarily, Fig. 24 summarizes δ_E, δ_R, and P_{95} as grouped bars with their corresponding 95% bootstrap confidence intervals, making the relative performance and the uncertainty of each estimate directly comparable across methods and geometries. Finally, robustness to the assumed spatial correlation scale is assessed by varying the bootstrap block size: the consistency of the method ranking and uncertainty bounds across $\ell = 3, 5, 8$ mm in Figs. 23–24 is corroborated by the tabulated δ_E sensitivity results in Table 11, indicating that the conclusions are not an artifact of a particular choice of ℓ.

While the results in Tables 9–11 and Figs. 21–24 demonstrate

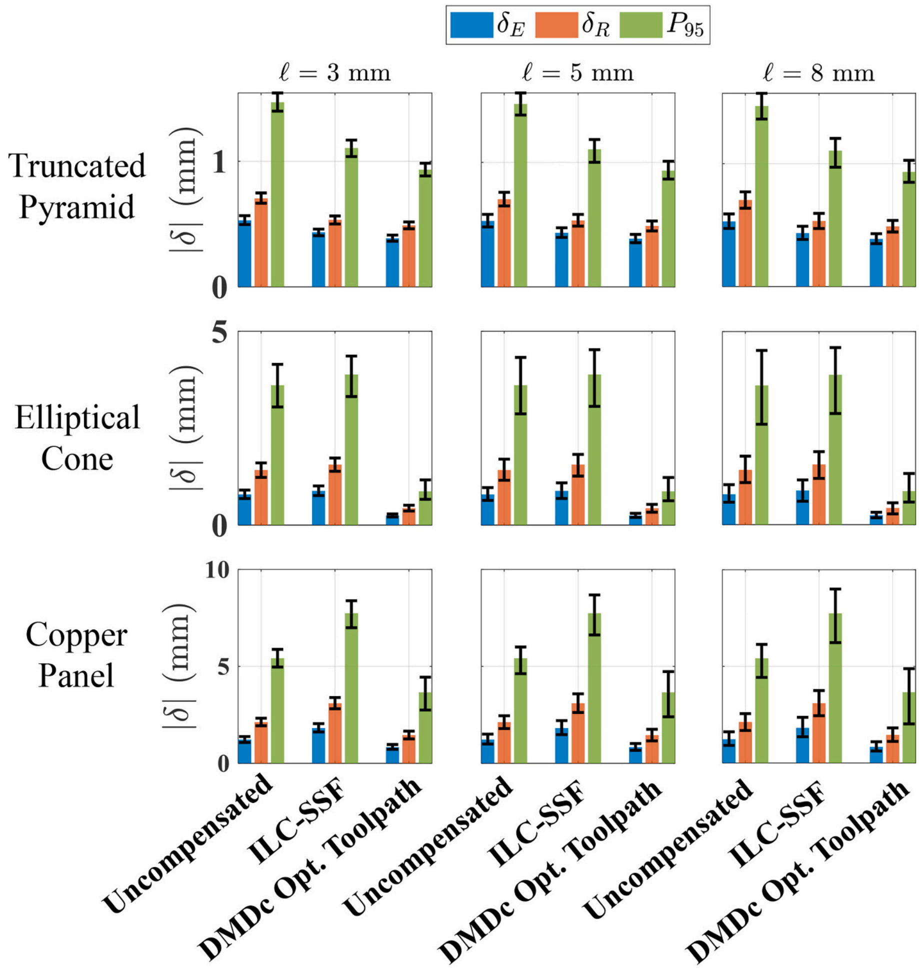


Fig. 24. Sensitivity of summary error metrics to block size ℓ in the spatial block bootstrap. For each geometry (rows) and tile size (columns: $\ell = 3, 5, 8$ mm), grouped bars report δ_E (MAE of $|\delta|$), δ_R (RMSE of $|\delta|$), and P_{95} of $|\delta|$ (mm) for Uncompensated, ILC-SSF, and DMDc-optimized toolpath. Error bars indicate 95% percentile confidence intervals from a spatial block bootstrap with tile resampling ($B = 1000$ replicates) performed separately within each method.

Table 11
Sensitivity analysis: impact of block size l on δ_E confidence intervals ($B = 1000$).

Shape	Method	$l = 3$ mm [95% CI]	$l = 5$ mm [95% CI]	$l = 8$ mm [95% CI]
Frustrum pyramid	Uncompensated	0.530 [0.496, 0.567]	0.530 [0.481, 0.582]	0.530 [0.473, 0.591]
	ILC-SSF	0.434 [0.407, 0.460]	0.434 [0.397, 0.474]	0.434 [0.383, 0.491]
	DMDc Optimized Toolpath	**0.387 [0.363, 0.413]**	**0.387 [0.355, 0.421]**	**0.387 [0.349, 0.429]**
Elliptical cone	Uncompensated	0.788 [0.685, 0.902]	0.788 [0.637, 0.969]	0.788 [0.583, 1.038]
	ILC-SSF	0.883 [0.761, 1.010]	0.883 [0.686, 1.091]	0.883 [0.607, 1.161]
	DMDc Optimized Toolpath	**0.247 [0.210, 0.286]**	**0.247 [0.197, 0.303]**	**0.247 [0.178, 0.323]**
Copper Panel	Uncompensated	1.228 [1.078, 1.383]	1.228 [0.998, 1.507]	1.228 [0.908, 1.615]
	ILC-SSF	1.821 [1.615, 2.050]	1.821 [1.481, 2.205]	1.821 [1.355, 2.365]
	DMDc Optimized Toolpath	**0.834 [0.721, 0.967]**	**0.834 [0.668, 1.020]**	**0.834 [0.617, 1.096]**

consistent improvements over ILC-SSF across the tested cases, two primary limitations persist. First, the benchmarking is restricted to a single representative baseline to ensure methodological consistency under fixed (x, y) and depth-correction toolpath parameterization. Consequently, the performance of the proposed method relative to alternative ILC variants, MPC formulations, or hybrid adaptive-feedback strategies has not been established. Second, the approach is predominantly feedforward and depends heavily on the fidelity of the learned DMDc operator. This will quantify the accuracy–complexity trade-off and clarify when DMDc provides the best balance of prediction fidelity and optimization tractability. In the case of strongly non-convex, multi-feature panels, unmodeled effects such as friction variability, local plasticity path dependence, or measurement noise may diminish optimization effectiveness. These factors necessitate the future integration of robust feedback or combined feedforward-feedback control architectures.

5. Conclusion

This research presents a novel data-driven framework for dynamic modeling and toolpath optimization in SPIF. By leveraging DMDc, we established a high-dimensional linear state-space model that predicts workpiece deformation as a function of the current shape and toolpath kinematics, capturing the nonlinear dynamics of SPIF while maintaining computational tractability. The key contributions of this work are threefold: First, we developed a physics-informed data pipeline that integrates FEM simulations, 3D scanning, point-cloud processing, and Thin-Plate Spline reconstruction to produce synchronized spatiotemporal deformation data. Second, our shape-prediction framework achieves experimentally validated accuracy across diverse geometries with less than 10% RMSE. Third, we introduced a quadratic programming–based toolpath optimization method that reduces geometric errors by 26.8–68.7% relative to uncompensated baseline and achieves up to ~78% reduction in upper-tail deviation metrics relative to ILC-SSF baseline for the tested geometries. The method's advantage over ILC-SSF stems from eliminating iterative correction via DMDc-based prediction and from computational efficiency through convex optimization. However, the current workflow is executed offline and is data intensive. Notably, performance degrades as geometric complexity increases, motivating future integration with feedback/hybrid control and online model adaptation. Overall, this work establishes a foundation for offline, model-based toolpath planning in flexible sheet metal manufacturing. Future work will extend benchmarking across additional compensation techniques and evaluate robustness across broader materials, thicknesses, and process windows. This will include controlled benchmarks of DMDc against lower-order linear surrogates and representative nonlinear predictors under matched data budgets and identical error metrics to quantify accuracy–complexity trade-offs.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work the authors used Copilot, Grok and DeepSeek for coding assistance, data analysis, and grammar checks. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

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.

Acknowledgments

The authors acknowledge support from the Research Grants Council (RGC) of the Government of Hong Kong Special Administrative Region (HKSAR) with RGC/ECS Project No. 26215824, HKUST – HKUST(GZ) Cross-campus Collaborative Research Scheme under the Great Bay Area "1 + 1 + 1" Joint Funding Program, i.e. Guangdong S&T Program (No. 2025A0505000030), and HKUST Frontier Technology Research for Joint Institutes with Industry Scheme OKT26EG04. Dr. Sattar Ullah from the School of Mechanical Engineering and Automation, Beihang University, Dr. Patrick Lam, from the Materials Design and Manufacturing Facility, Hong Kong University of Science and Technology, Mr. Ammar Kalo, Associate Professor, the American University of Sharjah, and Dr. Xiangyu Li, Assistant Professor, School of Mechanical Engineering, Northwestern Polytechnic University, provided valuable feedback and assistance. Ms. Tania Wilmshurst provided proofreading and English editing.

References

[1] Amino M, Mizoguchi M, Terauchi Y, Maki T. Current status of "Dieless" Amino's incremental forming. Procedia Eng 2014;81:54–62. https://doi.org/10.1016/j.proeng.2014.09.128.

[2] Echrif SBM, Hrairi M. Research and progress in incremental sheet forming processes. Mater Manuf Process 2011;26:1404–14. https://doi.org/10.1080/10426914.2010.544817.

[3] Katajarinne T. On the behaviour of the process and material in incremental sheet forming [dissertation]. Helsinki: Aalto University; 2014. Aalto University publication series Doctoral Dissertations 83/2014. ISBN 978–952–60-5719-4. https://aaltodoc.aalto.fi/handle/123456789/13356.ing 2014.

[4] Li Y, Pang Q, Hu Z li. Research on the accuracy of dieless single point incremental forming based on machine vision. J Manuf Process 2024;114:59–66. https://doi.org/10.1016/j.jmapro.2024.01.075.

[5] Cao J, Banu M. Opportunities and challenges in metal forming for lightweighting: review and future work. J Manuf Sci Eng Trans ASME 2020;142. https://doi.org/10.1115/1.4047732/1085053.

[6] Zhang Z, Zhang H, Shi Y, Moser N, Ren H, Ehmann KF, et al. Springback reduction by annealing for incremental sheet forming. Procedia Manuf 2016;5:696–706. https://doi.org/10.1016/J.PROMFG.2016.08.057.

[7] Kumar A, Mishra GJ, Gulati V, Srivastava AK, Kumar P, Kumar V, et al. A comprehensive review on heat-assisted incremental sheet forming. Int J Interact Des Manuf 2024;18:3583–601. https://doi.org/10.1007/S12008-023-01670-5/FIGURES/18.

[8] Kumar SP, Elangovan S, Mohanraj R, Boopathi S. A comprehensive review in incremental forming on approaches of deformation analysis and surface morphologies. Mater Today Proc 2021;43:3129–39. https://doi.org/10.1016/J.MATPR.2021.01.599.

[9] Moser N. Deformation mechanisms and process planning in double-sided incremental forming [dissertation]. Evanston (IL): Northwestern University; 2019.

[10] Vignesh G, Pandivelan C, Sathiya Narayanan C. Review on multi-stage incremental forming process to form vertical walled cup. Mater Today Proc 2019;27:2297–302. https://doi.org/10.1016/j.matpr.2019.09.116.

[11] Nguyen DT, Park JG, Lee HJ, Kim YS. Finite element method study of incremental sheet forming for complex shape and its improvement. Proc Inst Mech Eng B J Eng Manuf 2010;224:913–24. https://doi.org/10.1243/09544054JEM1825.

[12] Jaglan A, Dogra N, Kumar A, Kumar P. Finite element simulation approach in incremental sheet forming process. In: Handbook of flexible and smart sheet forming techniques: industry 4.0 approaches; 2023. p. 129–39. https://doi.org/10.1002/9781119986454.CH7/FORMAT/CHAPTER-EPUB.

[13] Harfoush A, Haapala KR, Tabei A. Application of artificial intelligence in incremental sheet metal forming: a review. Procedia Manuf 2021;53:606–17. https://doi.org/10.1016/j.promfg.2021.06.061.

[14] Bouhamed A, Ellouz H, Jrad H. Numerical optimization of SPIF for steel matrix composites using an elastoplastic damage model and desirability-based RSM. Int J Mater Form 2025;18. https://doi.org/10.1007/S12289-025-01920-9.

[15] Dammak M, Bouhamed A, Jrad H, Dammak F. In situ experimental characterization and numerical investigation of Fe-TiB2 Steel Matrix Composite behavior considering fully coupled damage model: Simulation during incremental forming process. Mater Today Commun 2024;38:107741. https://doi.org/10.1016/J.MTCOMM.2023.107741.

[16] Dammak M, Bouhamed A, Jrad H. Experimental analysis and modeling of mixed hardening of Steel Matrix Composites under monotonic and reverse simple shear loading: Application to SPIF process. Mech Adv Mater Struct 2025;2025(32): 718–29. https://doi.org/10.1080/15376494.2024.2354935.

[17] Bouhamed A, Jrad H, Wali M. A multifaceted approach using RVE homogenization methodology for identification of elastoplastic behavior of steel matrix composite: Application to deep-drawing process. Mech Adv Mater Struct 2025;32:3530–44. https://doi.org/10.1080/15376494.2024.2394853.

[18] Allwood JM, Music O, Raithathna A, Duncan SR. Closed-loop feedback control of product properties in flexible metal forming processes with mobile tools. CIRP Ann 2009;58:287–90. https://doi.org/10.1016/J.CIRP.2009.03.065.

[19] Ren H, Li F, Moser N, Leem D, Li T, Ehmann K, et al. General contact force control algorithm in double-sided incremental forming. CIRP Ann 2018;67:381–4. https://doi.org/10.1016/J.CIRP.2018.04.057.

[20] Darzi S, Kinsey BL, Ha J. Reforming toolpath effect on deformation mechanics in double-sided incremental forming. Int J Mech Sci 2024;280:109548. https://doi.org/10.1016/J.IJMECSCI.2024.109548.

[21] Ullah S, Li Y, Li X, Xu P, Li D. A review on the deformation mechanism and formability enhancement strategies in incremental sheet forming. Arch Civ Mech Eng 2023;23. https://doi.org/10.1007/s43452-022-00585-4.

[22] Trzepieciński T, Oleksik V, Pepelnjak T, Najm SM, Paniti I, Maji K. Emerging trends in single point incremental sheet forming of lightweight metals. Metals (Basel) 2021;11. https://doi.org/10.3390/met11081188.

[23] Kim TJ, Yang DY. Improvement of formability for the incremental sheet metal forming process. Int J Mech Sci 2000;42:1271–86. https://doi.org/10.1016/S0020-7403(99)00047-8.

[24] Behera AK, Lu B, Ou H. Characterization of shape and dimensional accuracy of incrementally formed titanium sheet parts with intermediate curvatures between two feature types. Int J Adv Manuf Technol 2016;83:1099–111. https://doi.org/10.1007/s00170-015-7649-2.

[25] Hirt G, Ames J, Bambach M, Kopp R, Kopp R. Forming strategies and process modelling for CNC incremental sheet forming. CIRP Ann 2004;53:203–6. https://doi.org/10.1016/S0007-8506(07)60679-9.

[26] Attanasio A, Ceretti E, Giardini C. Optimization of tool path in two points incremental forming. J Mater Process Technol 2006;177:409–12. https://doi.org/10.1016/J.JMATPROTEC.2006.04.047.

[27] Giraud-Moreau L, Belchior J, Lafon P, Lotoing L, Cherouat A, Courtielle E, et al. Springback effects during single point incremental forming: optimization of the tool path. In: AIP Conf Proc; 2018. p. 1960. https://doi.org/10.1063/1.5035035.

[28] Malhotra R, Reddy NV, Cao J. Automatic 3D spiral toolpath generation for single point incremental forming. J Manuf Sci Eng 2010;132. https://doi.org/10.1115/1.4002544.
[29] Azaouzi M, Lebaal N. Tool path optimization for single point incremental sheet forming using response surface method. Simul Model Pract Theory 2012;24:49–58. https://doi.org/10.1016/J.SIMPAT.2012.01.008.
[30] Yanagimoto J, Banabic D, Banu M, Madej L. Simulation of metal forming – visualization of invisible phenomena in the digital era. CIRP Ann 2022;71: 599–622. https://doi.org/10.1016/J.CIRP.2022.05.007.
[31] Thibaud S, Ben Hmida R, Richard F, Malécot P. A fully parametric toolbox for the simulation of single point incremental sheet forming process: Numerical feasibility and experimental validation. Simul Model Pract Theory 2012;29:32–43. https://doi.org/10.1016/J.SIMPAT.2012.07.004.
[32] Nimbalkar DH, Nandedkar VM. Prediction of forces in incremental sheet forming using finite element method. Mater Today Proc 2023. https://doi.org/10.1016/j.matpr.2023.04.339.
[33] Rauch M, Hascoet JY, Hamann JC, Plenel Y. Tool path programming optimization for incremental sheet forming applications. Comput Aided Des 2009;41:877–85. https://doi.org/10.1016/J.CAD.2009.06.006.
[34] Lu B, Chen J, Ou H, Cao J. Feature-based tool path generation approach for incremental sheet forming process. J Mater Process Technol 2013;213:1221–33. https://doi.org/10.1016/J.JMATPROTEC.2013.01.023.
[35] Buffa G, Gucciardi M, Fratini L, Micari F. Multi-directional vs. mono-directional multi-step strategies for single point incremental forming of non-axisymmetric components. J Manuf Process 2020;55:22–30. https://doi.org/10.1016/J.JMAPRO.2020.03.055.
[36] Jung KS, Yu JH, Chung WJ, Lee CW. Tool path design of the counter single point incremental forming process to decrease shape error. Materials 2020;13:1–16. https://doi.org/10.3390/ma13214719.
[37] Bârsan A, Racz SG, Breaz R, Crenganiș M. Dynamic analysis of a robot-based incremental sheet forming using Matlab-Simulink Simscape™ environment. Mater Today Proc 2022;62:2538–42. https://doi.org/10.1016/J.MATPR.2022.03.134.
[38] Paniti I. Adaptation of Incremental Sheet Forming into cloud manufacturing. CIRP J Manuf Sci Technol 2014;7:185–90. https://doi.org/10.1016/J.CIRPJ.2014.04.003.
[39] Bârsan A, Popp MO, Rusu GP, Maroșan AI. Robot-based incremental sheet forming – the tool path planning. IOP Conf. Ser. Mater. Sci. Eng., vol. 1009, IOP Publishing Ltd. 2021. https://doi.org/10.1088/1757-899X/1009/1/012004.
[40] Duflou JR, Callebaut B, Verbert J, De Baerdemaeker H. Laser Assisted Incremental Forming: Formability and Accuracy Improvement. CIRP Ann 2007;56:273–6. https://doi.org/10.1016/J.CIRP.2007.05.063.
[41] Schafer T, Schraft RD. Incremental sheet metal forming by industrial robots. Rapid Prototyp J 2005;11:278–86. https://doi.org/10.1108/13552540510623585/FULL/PDF.
[42] Dixit US. Modeling of metal forming: a review. In: Mechanics of materials in modern manufacturing methods and processing techniques; 2020. p. 1–30. https://doi.org/10.1016/B978-0-12-818232-1.00001-1.
[43] Kutz JN, Brunton SL, Brunton BW, Proctor JL. Chapter 1: Dynamic mode decomposition: an introduction. In: Other Titles Appl Math; 2016. p. 1–24. https://doi.org/10.1137/1.9781611974508.CH1.
[44] Simha CHM. Dynamic mode decomposition of deformation fields in elastic and elastic–plastic solids. Eur J Mech A Solids 2024;103:105168. https://doi.org/10.1016/J.EUROMECHSOL.2023.105168.
[45] Simha CHM, Biglarbegian M. A note on the use of Dynamic Mode Decomposition in mechanics. Mech Res Commun 2022;120:103848. https://doi.org/10.1016/J.MECHRESCOM.2022.103848.
[46] Proctor JL, Brunton SL, Kutz JN. Dynamic mode decomposition with control. SIAM J Appl Dyn Syst 2016;15:142–61. https://doi.org/10.1137/15M1013857. Https://DoiOrg/101137/15M1013857.
[47] Tu JH, Rowley CW, Luchtenburg DM, Brunton SL, Kutz JN. On dynamic mode decomposition: theory and applications. J Comput Dyn 2014;1:391–421. https://doi.org/10.3934/JCD.2014.1.391.
[48] Wang H, Duncan SR. Model predictive based incremental sheet forming toolpath optimisation. IET Semin Dig 2010;2010:1172–7. https://doi.org/10.1049/IC.2010.0446.
[49] Han F, Mo JH, Gong P, Li M. Method of closed loop springback compensation for incremental sheet forming process. J Cent South Univ Technol Engl Ed 2011;18: 1509–17. https://doi.org/10.1007/S11771-011-0867-3/METRICS.
[50] Music O, Allwood JM. The use of spatial impulse responses to characterise flexible forming processes with mobile tools. J Mater Process Technol 2012;212:1139–56. https://doi.org/10.1016/J.JMATPROTEC.2011.12.018.
[51] Horowitz JL. The bootstrap. In: Handbook of econometrics, vol. 5; 2001. p. 3159–228. https://doi.org/10.1016/S1573-4412(01)05005-X.
[52] Skala V. A brief survey of clipping and intersection algorithms with a list of references (including triangle-triangle intersections). Informatica (Slovenia) 2023; 34:169–98. https://doi.org/10.15388/23-INFOR508/ASSET/741944AF-F33E-44C6-92F5-2A27C467B042/ASSETS/IMAGES/LARGE/10.15388_23-INFOR508-FIG 11.JPG.
[53] Patrikalakis NM, Maekawa T. Intersection problems. In: Shape interrogation for computer aided design and manufacturing; 2010. p. 109–60. https://doi.org/10.1007/978-3-642-04074-0_5.
[54] Bieterman MB, Sandstrom DR. A curvilinear tool-path method for pocket machining. J Manuf Sci Eng 2003;125:709–15. https://doi.org/10.1115/1.1596579.
[55] Suresh K, Yang DCH. Constant scallop-height machining of free-form surfaces. J Eng Ind 1994;116:253–9. https://doi.org/10.1115/1.2901938.
[56] Tapp K. The curvature of a surface. 2016. p. 193–245. https://doi.org/10.1007/978-3-319-39799-3_4.
[57] Piegl L, Tiller W. Curve and surface fitting. 1997. p. 361–453. https://doi.org/10.1007/978-3-642-59223-2_9.
[58] Asghar J, Lingam R, Shibin E, Reddy NV. Tool path design for enhancement of accuracy in single-point incremental forming. Proc Inst Mech Eng B J Eng Manuf 2014;228:1027–35. https://doi.org/10.1177/0954405413512812.
[59] Wood SN. Thin plate regression splines. J R Stat Soc Ser B Stat Methodol 2003;65: 95–114. https://doi.org/10.1111/1467-9868.00374.
[60] Ullah S, Li X, Li D. Fast simulation of incremental sheet metal forming by multi-tooling. J Manuf Process 2022;84:669–80. https://doi.org/10.1016/J.JMAPRO.2022.10.025.
[61] Fischer JD, Woodside MR, Gonzalez MM, Lutes NA, Bristow DA, Landers RG. Iterative learning control of single point incremental sheet forming process using digital image correlation. Procedia Manuf 2019;34:940–9. https://doi.org/10.1016/J.PROMFG.2019.06.108.