ASME Journal of Manufacturing Science and Engineering
Online journal at:
https://asmedigitalcollection.asme.org/manufacturingscience

RETROFIT: Real-Time Control of Time-Dependent 3D Point Cloud Profiles

Michael Biehler
H. Milton School of Industrial and Systems Engineering,
Georgia Institute of Technology,
755 Ferst Dr NW,
Atlanta, GA 30332
e-mail: michael.biehler@gatech.edu

Jianjun Shi[1]
H. Milton School of Industrial and Systems Engineering,
Georgia Institute of Technology,
755 Ferst Dr NW,
Atlanta, GA 30332
e-mail: jianjun.shi@isye.gatech.edu

In modern manufacturing processes, ensuring the precision of 3D profiles of products is crucial. Nonetheless, achieving this accuracy is challenging due to the complex interactions between process inputs and the data structure of the 3D profile data. Our solution, a 3D profile-based control framework, addresses this challenge by actively adapting and controlling the manufacturing process to enhance 3D shape accuracy. 3D profile scans represent the ultimate measure of desired part quality. Therefore, utilizing them as the system responses for control purposes yields the most direct and effective feedback. We leverage recent advancements from Koopman operator theory to create an effective model-based control strategy. Initially, we estimate the process model by exploring the relationship between 3D profiles and heterogeneous process inputs. Then, we formulate an online model predictive control law. Challenges include dealing with unstructured, high-dimensional 3D point cloud data, capturing spatial and temporal structures, and integrating heterogeneous, high-dimensional process input data into the control model. To overcome these challenges, we introduce RETROFIT, a solution designed for the real-time control of time-dependent 3D point cloud profiles. Unlike traditional models, RETROFIT is not bound by linear assumptions and can handle unstructured 3D point cloud data directly. We demonstrate its effectiveness through a wire arc additive manufacturing case study, highlighting its potential to enhance 3D profile accuracy in manufacturing processes.
[DOI: 10.1115/1.4065222]

Keywords: 3D point cloud, compensation and control, data fusion, additive manufacturing, control and automation, inspection and quality control, production systems optimization, sensing, monitoring, and diagnostics

1 Introduction

In modern manufacturing processes, the control and compensation of 3D shape deviations have become crucial for ensuring high product quality and performance. However, the accurate reproduction of complex 3D profiles is influenced by various stages and factors, making it challenging to maintain consistent 3D shape accuracy throughout the manufacturing process. In this paper, we characterize a 3D profile as the measurement of the external boundary of a three-dimensional object. This definition underscores its association with non-contact measurement techniques, such as laser scanners, enabling the capture of intricate and nonlinear shapes. While the terms "3D profile" and "3D shape" can be used somewhat interchangeably, the latter places a greater emphasis on the overall geometry and spatial attributes, including features such as curvature. In various industries, 3D scanning stands out as the preferred method for assessing the parts' dimensional quality. As 3D scanners become increasingly affordable, faster, and more accessible, they offer the most direct feedback for compensation. Conversely, incorporating other sensing data, like acoustic data, for control could introduce sensitivity to environmental noise and necessitate recalibration for each new environment, thereby introducing further uncertainty. This indirect feedback method would also require constructing a model to correlate features extracted from other sensing data with the ultimate 3D dimensions, which may prove impractical. Importantly, our paper advocates for a control strategy based on 3D point clouds, directly measuring and controlling the dimensional quality we aim to optimize. While other methods provide partial information (e.g., width and height), complete 3D scans offer the most comprehensive feedback.

As depicted in Fig. 1, the control of 3D shapes plays a pivotal role in a wide array of manufacturing fields. Whether the task is to ensure shape accuracy in 3D printing or manipulate 3D shape memory effects in 4D printing: effective control over 3D shapes is indispensable for achieving functional fulfillment of manufactured products.

To address this critical issue, we propose a 3D profile-based control framework for dynamically evolving 3D profiles. The framework leverages recent advancements in Koopman methods [1], a powerful mathematical tool for modeling and analyzing dynamical systems, to design an effective model-based control strategy. By incorporating deep learning-based techniques, our approach overcomes the limitations of traditional tensor-based models [2], allowing for the processing of unstructured and high-dimensional output data, such as 3D shapes.

[1] Corresponding author.

Manuscript received December 16, 2023; final manuscript received March 26, 2024; published online April 22, 2024. Assoc. Editor: Qing (Cindy) Chang.

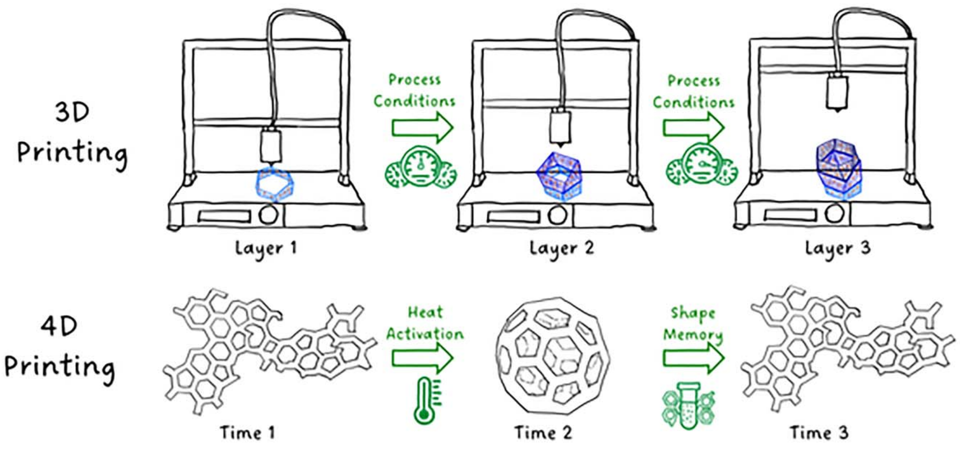

Fig. 1 Motivating applications of the RETROFIT framework

Our feedback control framework is tailored for the control of multi-stage manufacturing processes with the following the following data characteristics:

(1) **State Representations as Unstructured 3D Point Cloud Profiles:** The system outputs are represented as 3D profiles, measured as unstructured 3D point cloud data. These profiles evolve over time or stages, reflecting the system's dynamic behavior.
(2) **Multi-Stage Temporal Sequences:** The manufacturing process evolution spans multiple times or stages, each denoting a distinct phase of the control system's operation.
(3) **Multi-modal and Heterogeneous Control Inputs:** At each time (or stage), an array of control inputs of varying modalities is applied. These inputs not only influence the outputs (3D profiles) of the current time (or stage) but also its propagation to the future time (or downstream stages).
(4) **Nonlinear State Dynamics With Cascade Dependencies:** The response of the current stage is a nonlinear function of both the previous stage's state outputs and the current stage's control inputs. This cascade of dependencies characterizes the closed-loop dynamics of the control system.

While our proposed framework has wide-ranging applications in controlling and compensating for 3D shape deviations across various manufacturing processes, we highlight one specific example in this paper: wire arc additive manufacturing (WAAM). WAAM is a promising technique that enables the fabrication of complex metal components layer by layer, offering unique advantages such as rapid prototyping, material efficiency, and design flexibility. However, ensuring accurate and consistent 3D shape reproduction in WAAM poses significant challenges. In WAAM, the 3D profile of the fabricated component is influenced by numerous heterogeneous process inputs, including functional curves and tabular data. These process inputs introduce variations and inaccuracies, leading to deviations in the final 3D shape. Traditional control methods struggle to handle the complexity and high dimensionality of the data involved in 3D shape control. Our proposed 3D profile-based control approach, named RETROFIT, addresses these challenges by integrating deep-learning techniques. We show that RETROFIT can control and compensate for 3D shape deviations in WAAM. The impact of this paper extends beyond WAAM, as the control of 3D shapes is a pervasive challenge in various manufacturing applications.

This paper introduces a 3D profile-based control framework for dynamically compensating and controlling 3D profiles in manufacturing processes. By addressing the challenges associated with high-dimensional and heterogeneous data inputs, our proposed approach offers a comprehensive and adaptable solution for improving 3D shape accuracy via automatic feedback/feedforward control. The capability provided by our framework opens new possibilities for precision manufacturing, product customization, and innovative design, ultimately driving advancements in manufacturing industries and revolutionizing the way we create and produce 3D-shaped products.

The main contributions of this paper are as follows:

- We introduce a novel framework that integrates heterogeneous inputs for the control of 3D profiles in manufacturing processes, allowing for enhanced capabilities in controlling complex systems.
- We adapt and leverage the Koopman operator theory to enable efficient model predictive control (MPC) for high-dimensional (unstructured) 3D point clouds that exhibit strong short-term dependencies.
- We conduct case study experiments and show how to leverage 3D profile-based control in a unified way. We see this as a guiding example for the design and modeling of novel 3D profile-based control systems for controlling dynamic systems with 3D point cloud data as their responses.

The remainder of the article is organized as follows. Section 2 gives a brief literature review. Then the proposed RETROFIT framework for real-time control of time-dependent 3D profiles is introduced in Sec. 3. In Sec. 4, we conduct a real-world case study for controlling the 3D shape evolution in wire arc additive manufacturing. Finally, we conclude the article with a brief discussion and an outline of future work in Sec. 5.

2 Literature Review

In this section, we will review two fundamental categories of methodology that are closely related to the RETROFIT framework: the field of model-based control and Koopman operator theory.

2.1 Model-Based Control.

Model-based control is a widely used approach in control engineering, offering the capability to optimize control actions based on mathematical models of the system being controlled. In the past auto-regressive models such as ARIMAX [3], state space models [4], and linear models [5] have been extensively used. In recent years, the methods have evolved to nonlinear models [6]. In terms of data modality, these methods evolved from univariate (scalar) [7], to multivariate (vector) [8], to image (matrix)-based control [2] methods. These models, though effective in many applications, face limitations when the response data are in the form of 3D point clouds. Therefore, the existing literature on the control of 3D profiles heavily relies on feature extraction or sampling-based approaches [9–13]. However, conventional models struggle to capture the intricate spatial and temporal information embedded in 3D point clouds. The irregular data structure, high dimensionality, and noisy nature of 3D point clouds pose significant challenges for traditional model-based control approaches [1,14]. The integration of techniques like point cloud processing and deep learning-based models is opening new avenues for control in complex, spatially rich environments.

2.2 Koopman Operator Theory. The utilization of Koopman operators, originally formalized by Koopman (1931), has significantly influenced the field of modeling and control. The Koopman operator theory is a mathematical framework that enables the analysis of nonlinear systems by representing them as linear evolutions of nonlinear observables. It extends the analysis from finite-dimensional to infinite-dimensional systems. Essentially, it offers a means to study intricate nonlinear dynamics through techniques from linear algebra. This approach is valuable as it allows for the analysis of high-dimensional and spatiotemporal data in a computationally efficient manner. Several practical algorithms have been developed to extract the spectral properties of the Koopman operator from state-transition data. These include methods like dynamic mode decomposition (DMD) [15], EDMD [16], and Deep-DMD [17], which employ a combination of linear and nonlinear functions to encode observables and estimate the linear operator through optimization techniques [18]. Additionally, researchers have explored using neural networks to automate the design of observable functions, enabling a richer class of observable functions and overcoming the curse of dimensionality [19–21]. Despite these advancements, certain challenges persist such as the extension to multi-modal inputs and unstructured data formats such as 3D point clouds.

In this paper, we propose a novel framework specifically designed for 3D profile control in the presence of heterogeneous control inputs. Our approach leverages the Koopman operator theory to model the system dynamics and enable effective control. By utilizing the Koopman operator, we can handle the complex interactions and dependencies inherent in 3D profile control scenarios involving diverse input sources. This framework provides a powerful tool for addressing the challenges associated with controlling 3D systems and offers new opportunities for achieving precise and robust control in such contexts.

3 Retrofit Methodology

The proposed 3D profile-based control framework consists of two steps: (i) the offline modeling and parameter estimation step and (ii) the online control step. In the offline modeling and parameter estimation step, we establish the relationship between the 3D profile response sequence and control input variables, considering the multi-stage nature of manufacturing processes and the presence of multiple heterogeneous data sources. This step enables us to gain insights into the complex interplay between the 3D profile response and control variables. In the online control step, we leverage Koopman methods to design an effective control law based on the established relationships. This control law allows for dynamic adjustments of manufacturing process parameters in real-time, compensating for variations and inaccuracies introduced by the control inputs. This idea is demonstrated in Fig. 2, which depicts how the implementation of dimensional compensation control strategies can be achieved by minimizing the difference between the predicted profile, which is influenced by process variation, and the desired profile.

In our framework, real-time refers to control within process cycle times (e.g., dwell time in the WAAM case study) and operational constraints as commonly defined in the literature [22]. The subsequent sections provide detailed explanations of each step, including methodologies, algorithms, and experimental validation of the proposed framework's superiority.

3.1 Koopman Operator. Consider the nonlinear discrete-time dynamical system described by a nonlinear 3D profile propagation model as follows:

$$\mathcal{X}_S^{t+1} = f(\mathcal{X}_S^t, u^t),\ \mathcal{X}_S^t \in \mathcal{M} \subset \mathbb{R}^{N_p \times 3+h},\ u^t \subset \mathbb{R}^m \quad (1)$$

where $\mathcal{X}_S^t$ denotes the state (i.e., 3D profiles and process conditions) of the system at time $t \in T$. The time index t can also serve as a representation of the stages in a multi-stage manufacturing process. Note that t is utilized to represent both the number of 3D profiles within a sequence and the discrete number of stations in an Multistage Manufacturing Process (MMP) or layers in 3D printing. Currently, at the preliminary stage of sensor integration, we perform 3D scanning on each layer once it is printed, thereby referring to discrete, predefined layer numbers. However, we anticipate future advancements where 3D scanning will take place during printing, facilitating error compensation within layers. Consequently, any observed inaccuracies within a given layer can be compensated for while or before the remainder of the layer is printed. N_p denotes the number of cartesian coordinate points representing the 3D profile and h is the dimension of the process conditions; $u^t \in \mathcal{U}$ are the control inputs at time t, and m is the dimension of the control input data sources. Equation (1) illustrates the cascading nature inherent in many MMP systems (data characteristic 4, Sec. 1), where the subsequent printing layer $\mathcal{X}_S^{t+1}$ depends on both the preceding printing layer $\mathcal{X}_S^t$ and the deployed control action u_t from time t to $t+1$. In this paper, we exemplify this phenomenon within 3D printing applications. Our methodology is rooted in nonlinear state dynamics characterized by cascade dependencies. We explicitly model the interdependence between successive layers and control actions to address this inherent structure. As sensor integration and computational capabilities advance, our approach aims to model and control those cascade dependencies within diverse MMPs. Then, the map $f{:}\mathcal{M}' \to \mathcal{M}'$ is a (potentially nonlinear) update on a finite-dimensional manifold $\mathcal{S}$. The system can be

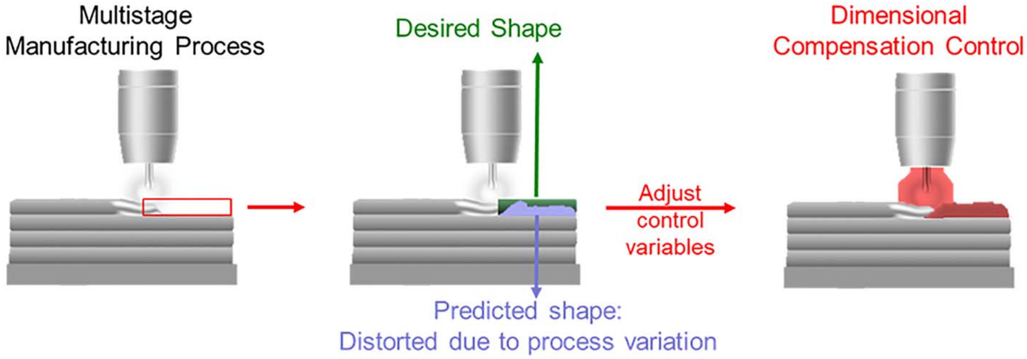

Fig. 2 Illustration of 3D profile compensation and control concept

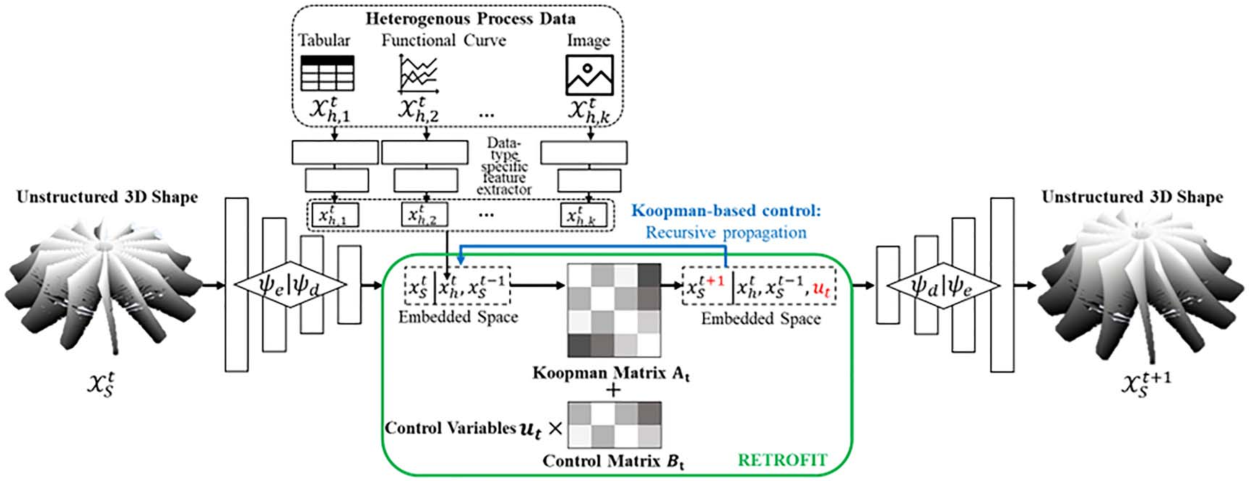

Fig. 3 Overview of RETROFIT Framework

lifted to an infinite-dimensional function space $\mathcal{M}'$ composed of all square-integrable real-valued functions within the compact domain $\mathcal{X}_S^t \times u^t \subset \mathbb{R}^{N_p \times 3+h+m}$. Elements of $\mathcal{M}'$ are called observables. On this space, the flow of the systems is characterized by the Koopman operator $\mathcal{K}$, which is an infinite-dimensional linear operator that satisfies

$$\mathcal{K}_\varphi g(\mathcal{X}_S, u) = g \circ \varphi(\mathcal{X}_S, u) \tag{2}$$

where φ denotes the observable function. However, the infinite-dimensional function space $\mathcal{M}'$ and the Koopman operator $\mathcal{K}$ are impractical. Therefore, a finite-dimensional function space $\overline{\mathcal{M}'} \subset \mathcal{M}'$, spanned by a linearly independent basis function $\varphi: \mathbb{R}^{N_p \times 3} \to \mathbb{R}^n$ is used instead, where n is the dimensionality of the observable function specified by the modeler. The linear property of the Koopman operator enables the use of linear control techniques for efficient control. The Koopman equation can be written as

$$A_t \varphi(\mathcal{X}_S) + B_t u = \varphi \circ f(\mathcal{X}_S, u) \tag{3}$$

where the Koopman operator is split into the Koopman matrix $A_t \in \mathbb{R}^{n \times n}$ and the control matrix $B \in \mathbb{R}^{n \times m}$. Note that throughout the methodology section, we will furthermore use time-varying Koopman and control matrices A_t and B_t, respectively. This allows greater modeling flexibility for systems where the dynamics change over time or stages.

3.2 Offline Modeling and Parameter Estimation of 3D Profile Propagation Function.

The offline estimation framework consists of two building blocks: an autoencoder (AE)-based architecture that encodes the observables, and a Koopman operator that propagates into the future. In the following section, we will describe the construction, implementation, and optimization of the offline estimation framework, which is based on the recently developed DETONATE model [1]. We acknowledge the need for careful selection of the neural network architecture for each dataset. To address this issue, we recommend the utilization of established techniques, including Bayesian hyperparameter tuning and warm-starting (e.g., pretraining an autoencoder model). For further details and practical guidance, interested readers are directed to the supplementary code repository. The big picture of the RETROFIT framework is provided in Fig. 3. The horizontal direction describes the Koopman-based temporal modeling, while the vertical direction shows the integration of heterogeneous inputs used to enhance the modeling of the nonlinear system dynamics.

The model utilizes an autoencoder-based framework to embed the high-dimensional 3D profiles $\mathcal{X}_S$ into a low-dimensional subspace, where the forward Koopman matrix A_t allows advancing of the temporal dynamics forward in time. The vertical direction in Fig. 3 shows the integration of various heterogeneous input data sources $\mathcal{X}_h$ to enhance the temporal propagation model in the horizontal direction. More specifically, we minimize the spatiotemporal 3D profile propagation loss, which is based on the DETONATE framework

$$\begin{aligned}\mathcal{L}_{\text{RETROFIT,offline}} &= \mathcal{L}_{\text{rec}} + \lambda_1 \cdot \mathcal{L}_{\text{fwd}} + \lambda_2 \cdot \mathcal{L}_h \\ &= \sum_{i=1}^{N} \sum_{t=2}^{T-1} d_{eCH}(\mathcal{X}_S, \hat{\mathcal{X}}_S) + \lambda_1 \cdot \sum_{i=1}^{N} \sum_{t=2}^{T-1} d_{eCH}(\mathcal{X}_S^{t+1}, \hat{\mathcal{X}}_S^{t+1}) \\ &+ \lambda_2 \cdot \sum_{i=1}^{N} \sum_{t=1}^{T-2} \sum_{j=1}^{k} f_{h,k}(x_{h,j}^t, \theta_h) - \mathcal{X}_{h,jF}^{t\ \ 2}\end{aligned} \tag{4}$$

where $\lambda_1, \lambda_2 \in \mathbb{R}^+$ are tuning parameters. The first loss term ($\mathcal{L}_{\text{rec}}$) embeds the high-dimensional, unstructured 3D profiles $\mathcal{X}_S$ into a low-dimensional embedding by utilizing a permutation-invariant autoencoder. The second loss term ($\mathcal{L}_{\text{fwd}}$) optimizes the temporal forward predictions. The third loss term ($\mathcal{L}_h$) integrates the heterogeneous input data sources and learns the features that are most correlated with the temporal propagation model. Note in our notation, we concatenated the heterogeneous input data of dimension h into the state representation (i.e., $\mathcal{X}_S^t \in \mathcal{M} \subset \mathbb{R}^{N_p \times 3+h}$).

The DETONATE approach represents a new milestone in the fusion of unstructured, heterogeneous, and high-dimensional data. It enables end-to-end learning of dynamic 3D point clouds and accurate predictions for both past and future temporal observations. Based on this strategy, we can develop a model for the offline modeling of the RETROFIT framework. With given observation $\mathcal{X}_S^t$ and heterogeneous inputs $\mathcal{X}_{h,j}^t$, the yields the forward prediction as

$$\hat{\mathcal{X}}_S^{t+1} = \psi_d \circ A_t \circ \psi_e(\mathcal{X}_S^t) \tag{5}$$

where ψ_e denotes the encoder that embeds the 3D profile into a low-dimensional space, ψ_d is the decoder that maps back to the original space, and A_t is the temporal, forward Koopman matrix. Subsequently, we will utilize this model in the control setting to obtain temporal forecasts based on heterogeneous control inputs. The main difference in the proposed RETROFIT framework compared to DETONATE lies in the integration of control inputs. In RETROFIT, we separate temporal dynamics into a Koopman matrix A_t and a control matrix B_t. This deliberate separation allows for a rigorous theoretical analysis, where we utilize advanced concepts from Lie algebra to understand control properties. It is important to note that our approach differs from conventional linear control and its associated proof techniques. Since our method targets nonlinear systems, achieved through mapping to a low-dimensional manifold, our theoretical analysis focuses on the behavior within this manifold. Consequently, we introduce additional structure and constraints on A_t and B_t to derive theoretical results on stability and controllability. By introducing this structure, we minimize the risk of inaccuracies when naively applying the DETONATE model to systems with actuation. In such cases, there is a potential for the internal dynamics captured by heterogeneous process data to be confused as the effects of control.

3.3 Online Retrofit Model Predictive Control.

After introducing the offline model, we will now propose the concepts necessary for online control. The overarching goal of this paper is to control a real system to the target profile $\mathcal{X}_S^{t,\text{Target}}$ by applying control actions u^t according to a RETROFIT model predictive control framework to ensure the resulting process output $\hat{\mathcal{X}}_S^t$ is as close to the target profile as possible. This control procedure is illustrated in Fig. 4. In this context, t serves as the time index for tracking the evolution of 3D profiles over time. It can also be interpreted as the number of 3D profiles within a sequence in spatial terms or as the number of layers in 3D printing. Throughout the subsequent discussions in this paper, we will consistently denote this time index as t.

While the Koopman operator theory is well-established for uncontrolled dynamical systems, its application has been extended when dealing with controlled dynamical systems. Naively applying Koopman operator theory to data from a system with actuation would often result in incorrect dynamics, as the effects of internal dynamics are confused with the effects of actuation. Several prior studies have addressed this particular challenge [23]. Therefore, we exploit the MPC framework to achieve robust and stabilizing control guarantees. By imposing constraints on the set of observables the resulting approximation of the operator follows the form of a linear controlled dynamical system. This procedure is completely data-driven and does not require any a priori knowledge about the underlying nonlinear system dynamics. It consists of two components: a nonlinear autoencoder-based transformation and a least squares problem in the embedded space, which can be readily solved for large datasets.

The focus of this paper is on controlling the trajectory of the 3D profile propagation modeled as Eq. (1) given an initial condition $\mathcal{X}_S^0$ and control inputs $\{u^0, u^1, \ldots\}$. For predictors with linear structure, we can then utilize readily available concepts from linear control. Since the RETROFIT model encodes the 3D profiles $\mathcal{X}_S$ into a low-dimensional space, the nominal dynamic of the observables can be characterized by

$$\begin{aligned} x_S^{t+1} &= A_t x_S^t + B_t u^t \\ \hat{\mathcal{X}}_S^{t+1} &= \psi_d(x_S^{t+1}) \end{aligned} \tag{6}$$

where $x_S^t \in \mathbb{R}^n$ denotes the low-dimensional embedding of the system state at time t, $A_t \in \mathbb{R}^{n\times n}$ the Koopman matrix, $u^t \in \mathbb{R}^m$ the control input at time t, $B_t \in \mathbb{R}^{n\times m}$ the control matrix, $\hat{\mathcal{X}}_S^{t+1}$ is the prediction of the future system state $\mathcal{X}_S^{t+1}$ (i.e., a 3D profile), and ψ_d is the nonlinear decoder mapping from the low-dimensional embedding x_S^{t+1} to the high-dimensional 3D system state obtained from the RETROFIT offline propagation model. The initial condition of the predictor (Eq. (6)) is given by

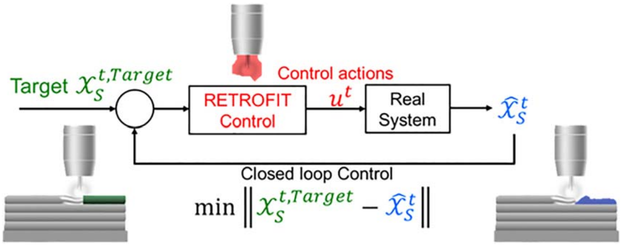
Target $\mathcal{X}_S^{t,Target}$
Control actions
RETROFIT Control
u^t
Real System
$\hat{\mathcal{X}}_S^t$
Closed loop Control
$\min \left\| \mathcal{X}_S^{t,Target} - \hat{\mathcal{X}}_S^t \right\|$

Fig. 4 Illustration of the online 3D profile compensation and control concept

$$x_S^0 = \psi_e(\mathcal{X}_S^0) \tag{7}$$

where $\mathcal{X}_S^0$ is the initial condition of Eq. (1), ψ_e is the nonlinear encoder function from the DETONATE model that maps the high-dimensional, unstructured 3D point clouds to a low-dimensional embedding utilizing a permutation-invariant 3D autoencoder architecture [22]. Note that the control input u in Eq. (4) is not embedded into a low-dimensional space and hence linear constraints on the control inputs can be imposed in a linear fashion. Predictors of this form lend themselves immediately to linear feedback control design methodologies. Importantly, however, the resulting feedback controller will be nonlinear in the original state $\mathcal{X}_S$ allowing the modeling of complex nonlinear systems. For any feedback controller $\kappa_{\text{embed}}:\mathbb{R}^n \to \mathbb{R}^m$, we obtain a feedback controller $\kappa:\mathbb{R}^{N_p\times 3} \to \mathbb{R}^m$ for the original system (Eq. (1)) defined by

$$\kappa(\mathcal{X}_S) = \kappa_{\text{embed}}(\psi_e(\mathcal{X}_S)) \tag{8}$$

The idea is that if the true trajectory of $\mathcal{X}_S$ generated by Eq. (1) and the predicted trajectory of $\hat{\mathcal{X}}_S$ generated by Eq. (4) are close for each admissible input sequence, then the optimal controller for Eq. (4) should be close to the optimal controller for Eq. (1).

However, it is important to note that, in general, one cannot expect the trajectory of a linear system (Eq. (4)) to accurately predict the trajectory of a nonlinear system for all future times. Nonetheless, if the predictions remain accurate over a sufficiently long time interval, these predictors can aid in the utilization of linear control system design methodologies. Particularly well-suited for this purpose is MPC [24], which relies solely on finite-time predictions to generate the control input. There are several ways of generalizing the Koopman operator to controlled systems [25,26]. Here, we adapt the ideas from Ref. [24].

We define the Koopman operator associated with the controlled dynamical system (Eq. (1)) as the Koopman operator linked to the uncontrolled dynamical system evolving in the extended state space. The extended state space is defined as the Cartesian product of the original state space and the space encompassing all control sequences. In our case, this boils down to $\mathbb{R}^{N_p\times 3} \times \ell(\mathcal{U})$, where $\ell(\mathcal{U})$ is the space of all sequences $\boldsymbol{u}:(u^i)_{i=0}^{\infty}$. Consequently, the dynamics of the extended state χ are described by

$$\chi^{t+1} = F(\chi^t) = F\left(\begin{bmatrix} \mathcal{X}_S^t \\ \boldsymbol{u} \end{bmatrix}\right) : \begin{bmatrix} f(\mathcal{X}_S^t, \boldsymbol{u}(0)) \\ S\boldsymbol{u} \end{bmatrix} \tag{9}$$

where S is the shift operator $(S\boldsymbol{u})(i) = \boldsymbol{u}(i+1)$, and $\boldsymbol{u}(i)$ denotes the ith element of the sequence $\boldsymbol{u}$. The Koopman operator $\mathcal{K}:\mathcal{H} \to \mathcal{H}$ associated with Eq. (9) is defined by

$$(\mathcal{K}\phi)(\chi) = \phi(F(\chi)) \tag{10}$$

where each ϕ: $\mathbb{R}^{N_p\times 3} \times \ell(\mathcal{U}) \to \mathbb{R}$ belonging to some space of observables $\mathcal{H}$. The Koopman operator in Eq. (10) is a linear operator fully describing the nonlinear dynamical system (Eq. (1)) given that $\mathcal{H}$ contains the components of the non-extended state $\mathcal{X}_S^i,\ i = 1, \ldots, N$. To obtain a predictor of the format given in Eq. (6), we construct a finite-dimensional approximation of the operator $\mathcal{K}$. To this end, we assume a dataset $(\chi_j^t, \chi_j^{t+1}), j = 1, .., K,$ satisfying $\chi_j^{t+1} = F(\chi_j^t)$ is available and learn a matrix $\mathcal{A}$, which denotes the transpose of the finite-dimensional approximation of $\mathcal{K}$ by minimizing

$$\sum_{j=1}^{K} \phi(\chi_j^{t+1}) - \mathcal{A}_t\phi(\chi_j^t)_2^2 \tag{11}$$

where $\phi(\chi) = [\phi_1(\chi), \ldots, \phi_{N_\phi}(\chi)]^T$ is a vector of embedding functions ϕ_i: $\mathbb{R}^{N_p\times 3} \times \ell(\mathcal{U}) \to \mathbb{R},\ i \in \{1, \ldots, N_\phi\}$. However, in general $\chi = (\mathcal{X}_S,\ \boldsymbol{u})$ is an infinite-dimensional object and hence the loss of Eq. (11) cannot be evaluated in finite time unless some additional constraints are imposed. Therefore, we impose the following form on the functions ϕ_i:

$$\phi_i(\mathcal{X}_S,\ \boldsymbol{u}) = \psi_e(\mathcal{X}_S) + \mathcal{L}_i(\boldsymbol{u}), \quad i = 1, \ldots, N_\phi \tag{12}$$

where $\psi_e\colon \mathbb{R}^{N_p\times 3} \to \mathbb{R}$ is the nonlinear encoder function from the offline propagation model, but $\mathcal{L}_i\colon \ell(\mathcal{U}) \to \mathbb{R}$ is linear. By linearity and causality, we can assume without loss of generality that $N_\phi = N + m$ for some $N > 0$ and that the vector of embedding functions has the form

$$\phi(\mathcal{X}_S, \boldsymbol{u}) = \begin{bmatrix} \psi(\mathcal{X}_S) \\ \boldsymbol{u}(0) \end{bmatrix} \tag{13}$$

where $\boldsymbol{u}(0) \in \mathbb{R}^m$ denotes the first component of the sequence $\boldsymbol{u}$. Given that our objective is solely to acquire the present control input and subsequently to observe the system state to control the system we can disregard the last m components of each term $\phi(\chi_j^{t+1}) - \mathcal{A}_t\phi(\chi_j^t)$ in Eq. (11). Letting $\bar{\mathcal{A}}$ denote the first N rows of $\mathcal{A}_t$ and decomposing the matrix such that $\bar{\mathcal{A}}_t = [A_t, B_t]$ with $A_t \in \mathbb{R}^{N\times N}$, $B_t \in \mathbb{R}^{N\times m}$ leads to the minimization problem

$$\min_{A_t,B_t} \sum_{j=1}^{K} \psi_e(\mathcal{X}_{S,j}^{t+1,\text{Target}}) - A_t\psi_e(\mathcal{X}_{S,j}^t) - B_t u_j(0)_2^2 \tag{14}$$

where $\mathcal{X}_S^{t,\text{Target}}$ denotes the target profile at time t. Minimizing Eq. (14) over A and B leads to a predictor of the form in Eq. (6) starting from the initial condition $z_0 = \psi_e(\mathcal{X}_S^0)$. Given the dataset, the matrices A and B are obtained as the best linear one-step predictor in the low-dimensional embedded space in the least squares sense by optimizing

$$\min_{A_t,B_t} x_S^{t+1,\text{Target}} - A_t x_S^t - B_t U_F \tag{15}$$

where $x_S^t = [\psi_e(\mathcal{X}_{S,1}^t), \ldots, \psi_e(\mathcal{X}_{S,K}^t)]$, $U = [u_1, \ldots, u_K]^T$, and $\|\cdot\|_F$ denotes the Frobenius normal of a matrix. Therefore, the big picture of the RETROFIT control strategy is illustrated in Fig. 5 as follows: first, the high-dimensional 3D inputs are embedded in a low-dimensional subspace via the offline propagation encoder ψ_e. In the embedded space, the Koopman matrix A_t and the control matrix B_t define the temporal forward evolution. The original state space is a complex nonlinear system, so we employ an autoencoder to map it to a lower-dimensional manifold. On this low-dimensional manifold, we impose constraints to disentangle internal dynamics captured by heterogeneous inputs in A_t and the control dynamics in B_t, facilitating further theoretical analysis. By utilizing recursive propagation and minimizing the difference between the predicted 3D shape output and the target output, an effective control scheme can be derived.

In particular, MPC is a control strategy that involves determining the control input at each time-step during closed-loop operation. This is achieved by solving an optimization problem, where a user-defined cost function is minimized over a prediction horizon while considering constraints on the control inputs and state variables. At each time instance t of the closed-loop operation, the MPC solves the following optimization problem.

$$\begin{aligned}
&\min_{u^t, x_S^t} J((u^t)_{t=0}^{N_{\text{Pr}}-1}, (x_S^t)_{t=0}^{N_{\text{Pr}}-1}) \\
&\text{s.t.} \quad x_S^{i+1} = A_t\, x_S^i + B_t u^i, \ \ i = 0, \ldots, N_{\text{Pr}} - 1 \\
&E_i x_S^i + F_i u^i \le b_i, \ \ i = 0, \ldots, N_{\text{Pr}} - 1 \\
&E_{N_{\text{Pr}}} z^{N_{\text{Pr}}} \le b_{N_{\text{Pr}}}
\end{aligned} \tag{16}$$

Here $z^0 = \psi_e(\mathcal{X}_S^0)$ and N_{Pr} is the prediction horizon of the convex quadratic cost function J given by solving the following discrete-time equation:

$$\begin{aligned}
J((u^i)_{i=0}^{N_{\text{Pr}}-1}, (x_S^i)_{i=0}^{N_{\text{Pr}}}) &= x_S^{\prime N_{\text{Pr}}} Q_{N_{\text{Pr}}} x_S^{N_{\text{Pr}}} + q_{N_{\text{Pr}}} x_S^{N_{\text{Pr}}} \\
&+ \sum_{i=0}^{N_{\text{Pr}}} x_S^{\prime i} Q_i x_S^i + u^{\prime i} R_i u^i + q_i x_S^i + r_i u^i
\end{aligned} \tag{17}$$

where $Q_i \in \mathbb{R}^{N\times N}$ and $R_i \in \mathbb{R}^{m\times m}$ are positive semidefinite matrices. The matrices $E_i \in \mathbb{R}^{n_c\times N}$ and $F_i \in \mathbb{R}^{n_c\times m}$ and the vector $b_i \in \mathbb{R}^{n_c}$ define state and input polyhedral constraints. The optimization problem in Eq. (16) is parametrized by the current state of the nonlinear dynamical system $\mathcal{X}_S^t$ and defines a feedback controller of the form

$$\kappa(\mathcal{X}_S^t) = u^*(\mathcal{X}_S^t) \tag{18}$$

where $u^*(\mathcal{X}_S^t)$ denotes the optimal solution to Eq. (16) and κ is the control gain matrix. Note, that at each time-step t, the predictions are initialized from the embedded space $\psi_e(\mathcal{X}_S^t)$. Therefore, the optimization problem is a convex quadratic programming problem on the embedded space and can be solved efficiently via interior point methods. Given the system dynamics and the control law (Eq. (18)), we can compute the closed-loop transition matrix $A_{cl,t}$ as follows:

$$A_{cl,t} = A_t + B_t\kappa \tag{19}$$

The overview of the control procedure is summarized in Algorithm 1.

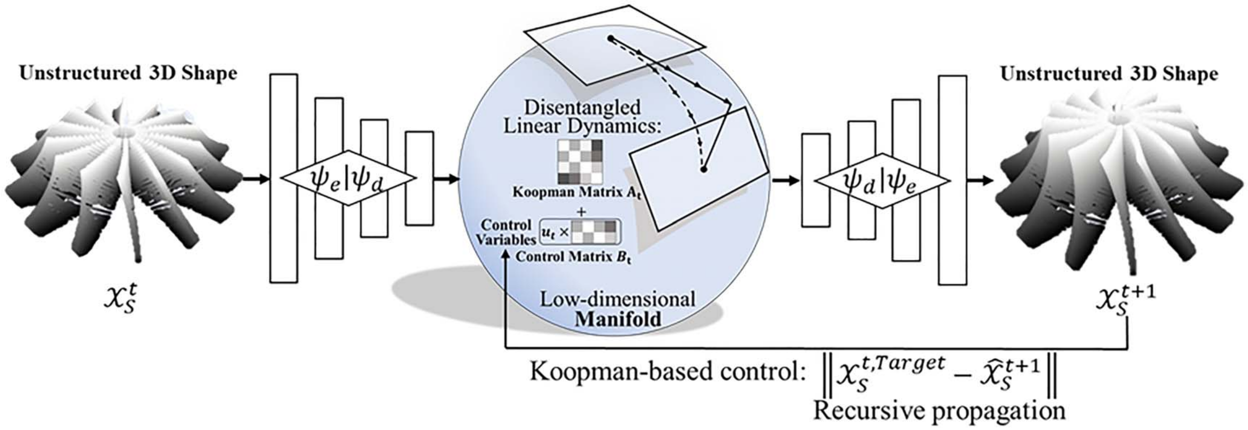

Fig. 5 Details of the RETROFIT online 3D shape compensation and control concept

Algorithm 1. RETROFIT MPC

```
Input:
    – Offline propagation model: encoder $\psi_e$, decoder $\psi_d$, Koopman
      matrix $A_t$, control matrix $B_t$
    – Weighting matrices: $Q, R$
    – Control system constraints: $E, F, b$
    – Prediction horizon: $N_{Pr}$
    – Control target: $\mathcal{X}_S^{t,\ Target}$
Initialize: $x_S^0 = \psi_e(\mathcal{X}_S^0)$
Output: Control strategy $u^*(\mathcal{X}_S^t)$
Algorithm:
for $t = 1, 2, \ldots$ do
        1. Solve Eq. (16) via interior point method
        2. Apply $u^*(\mathcal{X}_S^t)$ to the system (Eq. (1))
        3. $\hat{x}_S^{t+1} \leftarrow A_t \hat{x}_S^t + B_t u^*(\mathcal{X}_S^t)$
  end for
```

While the dynamics and dimensionality of the problem influence the required training sample size, leveraging theoretical insights alongside practical considerations can guide this determination effectively. As a standard practice, one can train the autoencoder structures on a large dataset, such as ShapeNet, which equips them with well-initialized weights. Consequently, the autoencoder already possesses a strong foundation for mapping high-dimensional 3D point clouds to a low-dimensional embedding. Fine-tuning this pretrained model to the specifics of our experiments significantly reduces the required training samples. Additionally, our theoretical bounds (Sec. 3.4) depend on the Lipschitz constant of our neural network model. A lower Lipschitz constant implies that the function changes more slowly in response to perturbations in the input space. Previous research indicates, that a lower Lipschitz constant leads to better generalization [27]. Conversely, a higher Lipschitz constant may indicate a need for more data to accurately capture the complex dynamics of the system. Furthermore, the eigenvalues of the Koopman matrix A_t provide insights into the system's dynamics. The distribution of eigenvalues can indicate whether the system is stable, oscillatory, or chaotic. If the eigenvalues exhibit significant spread or are densely distributed across the complex plane, it suggests that the system's dynamics is complex. Analyzing the eigenvalue spectrum of the Koopman matrix can help assess whether the training dataset sufficiently captures the diversity of system behavior. If certain eigenmodes are poorly represented or absent, it may indicate a need for additional samples to ensure comprehensive coverage of the system's dynamics [28]. While there is no one-size-fits-all answer to the number of required training samples, one can utilize both practical considerations and theoretical insights to guide the determination of an appropriate dataset size.

3.4 Theoretical Analysis.

In this section, we will focus on the stability analysis of the exact and approximate Koopman operators, as well as the controllability analysis of the system. Stability is a fundamental property of control systems. A stable system ensures that it does not deviate uncontrollably or exhibit erratic behavior when subjected to disturbances or changes in the input. In many practical applications, ensuring stability is the primary concern because an unstable system can lead to catastrophic consequences. Controllability, on the other hand, pertains to the system's ability to reach a desired state or trajectory from an initial state using available control inputs. A highly controllable system allows for precise control over its behavior and can be directed effectively to achieve specific objectives. Controllability is essential when you need to design and implement control strategies to meet specific performance requirements.

3.4.1 Stability of Exact Koopman Operators. First, we investigate the case where the exact Koopman operators are available. To establish the stability guarantee, we make the following assumptions:

Assumption 1. *The neural network encoder ψ_e is Lipschitz continuous, $\|\psi_e(x+y) - \psi_e(x)\| \le L\|y\|$.*

Neural networks frequently satisfy the Lipschitz continuity assumption organically due to the inherent characteristics of their training process. Gradient-based optimization techniques, such as stochastic gradient descent, implicitly introduce a form of regularization that encourages smooth mappings. This is aided by smooth activation functions and techniques like batch normalization. This assumption can be verified by methods such as AutoLip [29], which estimates an upper bound of the Lipschitz constant for a given differentiable neural network.

Assumption 2. *If considering random noise in the system (i.e., $\mathcal{X}_S^{t+1} = f(\mathcal{X}_S^t, u^t) + \varepsilon^t$), the random noise ε^t in the system has a bounded energy, i.e., a finite constant $b > 0$ exists such that $\mathbb{E}_\varepsilon\|\varepsilon\| \le b$, where $\mathbb{E}_\varepsilon$ is the expected value of the random variable ε^t.*

Assumption 2 is reasonable as it reflects the common observation that real-world systems are subject to bounded levels of random noise or disturbances. By asserting a finite upper bound on the energy of this noise, we acknowledge the practical limitations of disturbances in our system.

With the assumptions above, we can establish the stability guarantee as follows:

Proposition 1. *Consider the system in Eq. (1) controlled by the MPC controller in Eqs. (16)–(18). Then the closed-loop system (Eq. (1)) is uniformly ultimately bounded with bound $\frac{\beta\sigma L b}{1-\beta}$, where β denotes the maximum eigenvalue of the closed-loop transition matrix $A_{cl} = A + B\kappa$ and $\sigma := \|\psi_d\|$.*

Proof. The detailed proof is deferred to Appendix A. ■

3.4.2 Stability of Approximate Koopman Operator. Proposition 1 shows that the controller is stable with the exact Koopman operator. However, in practice, the exact Koopman matrices A^*, B^*, and decoder ψ_d^* are generally infeasible and only the sub-optimal solutions A, B, ψ_d can be obtained. Now, we will show that stability can be assured even though the Koopman matrices and decoder are approximated with bounded prediction error.

First, the nominal system (Eq. (6)) is a conceptual system with the approximated Koopman matrixes and decoder, and thus its dynamic and resulting MPC problem remain the same as in Eqs. (16)–(18). On the other hand, the true evolution of 3D profiles is characterized by the following nonlinear relationship:

$$\begin{aligned} x_S^{t+1} &= A_t^* x_S^t + B_t^* u^t \\ \hat{\mathcal{X}}_S^{t+1} &= \psi_d^*(x_S^{t+1}) \end{aligned} \tag{20}$$

We define the dynamic residual caused by the sub-optimal approximation $d^t = (A^* - A)x_S^t + (B^* - B)u^t$ and the reconstruction residual $\psi_d^*(x_S^t) = \psi_d(x_S^t) + r^t$. When taking into account the approximation errors d^t and r^t, along with the system noise ε^t, the system described in Eq. (20) can be reformulated as follows:

$$\begin{aligned} x_S^{t+1} &= A x_S^t + B u^t + \varepsilon^t + d^t \\ \hat{\mathcal{X}}_S^{t+1} &= \psi_d(x_S^{t+1}) + r^t \end{aligned} \tag{21}$$

Assumption 3. *There exist positive constants $\gamma, \eta \in \mathbb{R}^+$, such that $\|d^t\| \le \gamma$ and $\|r^t\| \le \eta$.*

Assumption 3 is rooted in the behavior of real-world systems. It recognizes that errors are inherently constrained by physical factors such as sensor limitations, actuator capabilities, and system design. These constraints prevent errors from accumulating. Moreover, this assumption carries significant implications for safety-critical systems, where bounded errors become a vital component in ensuring that the system remains within predefined safety margins, reducing the risk of unforeseen behaviors and enhancing overall system reliability.

With Assumptions 1–3, we can establish the stability guarantee as follows:

PROPOSITION 2. *Consider the system in Eq. (1) controlled by the approximate Koopman matrices and decoder functions via the MPC controller in Eqs. (16)–(18). Then the closed-loop system (Eq. (6)) is uniformly ultimately bounded with bound* $\frac{\beta\sigma(Lb+\gamma)}{1-\beta}+\eta$ *, where* β *denotes the maximum eigenvalue of the closed-loop transition matrix* $A_{cl} = A + B\kappa$ *and* $\sigma := \|\psi_d\|$.

Proof. The proof of Proposition 2 is deferred to Appendix B. ■

Proposition 2 shows that even with approximated Koopman matrices and decoder models, the stability of the system can still be guaranteed, though the uniform bound is inevitably larger than the ideal case in Proposition 1.

We note that Assumption 3 is equivalent to assuming that $\|A - A^*\|$, $\|B - B^*\|$, and $\|\psi_d^* - \psi_d\|$ are bounded respectively, and the state space $\mathbb{X}$ and the action space $\mathbb{U}$ are bounded. To simplify the notations, we chose the form in Assumption 3.

3.4.3 Controllability. Controllability is a foundational principle in systems theory. It plays a pivotal role in understanding the behavior of dynamic systems, providing insights into the ability to manipulate these systems effectively. Controllability assesses the degree to which a system's state trajectory can be moved from any initial state to any desired final state within a finite-time frame through the appropriate selection of control inputs. In the context of nonlinear systems, controllability encompasses various nuanced aspects, including accessibility, local controllability, and global controllability. Each of these notions offers distinct perspectives on the system's manipulability. Here we provide some formal definitions for clarification.

DEFINITION 1. *Accessibility* [30]: A nonlinear system is accessible from the state $x_S^0 \in \mathbb{R}^d$ if the attainable set from x_S^0 has a non-empty interior. However, this does not imply controllability or even local controllability.

DEFINITION 2. *Local Controllability* [30]: *A nonlinear system is locally controllable from* $x_S^0 \in \mathbb{R}^d$ *if the reachable set from* x_S^0 *contains a neighborhood of* x_S^0. *In other words, local controllability implies that starting from* x_S^0 *the system can reach any point in the state space which is near* x_S^0.

PROPOSITION 3. *The control-affine system (Eq. (6)) is locally controllable at* x_S^0 *if the Lie algebra of the vector field at* x_S^0 *span the tangent space at* x_S^0, *that is the rank of the Lie algebra generated by the vector field of Eq. (6) at* x_S^0, *is* d_m, *where* d_m *is the dimension of the low-dimensional manifold.*

Proof. Interested readers are referred to Ref. [30] for a formal proof. ■

In the case of linear systems, local controllability implies that the system is globally controllable, that is if the trajectories from x_S^0 can be steered to all points of the state space that are in the local neighborhood of x_S^0, it implies that the system can be driven to any point in the state space from any other point. However, for general nonlinear systems, this is not the case, and the notion of global controllability for nonlinear systems is defined as follows.

DEFINITION 3. *Global Controllability* [30]: *If a nonlinear system is locally controllable for all* x_S^0 *of the configuration manifold, then the system is globally controllable.*

We will first consider a general linear time-invariant (LTI) system of the following form to prove the controllability of our proposed framework:

$$x_S^{t+1} = Ax_S^t + Bu^t \tag{22}$$

where $x_S^t \in \mathbb{R}^d$ are the embedded states, u^t are scalar inputs with input matrix B. For the LTI system, the drift vector field is $f_d = Ax_S$.

PROPOSITION 4. *The LTI system is globally controllable if the accessibility distribution matrix* Q *has full rank for all initial states* $x_S^0 \in \mathbb{R}^{d_m}$. *Specifically, if* $\text{rank}(Q) = n$, $\forall x_s^t \in \mathbb{R}^d$, *then the system is controllable. The controllability matrix* $\mathcal{C}$ *is defined as* $\mathcal{C} := \begin{bmatrix} B & AB & A^2B & \dots & A^{n-1}B \end{bmatrix}$.

Proof. The proof is deferred to Appendix C. ■

We now demonstrate that the findings established for the LTI system also extend to the Koopman control within the original state space. Proposition 4 provides us with the necessary and sufficient conditions for ensuring global controllability of the system within the embedded space. Leveraging this insight, we can formulate the following proposition that establishes a connection between the controllability of the system in the embedded (Koopman) space and its counterpart in the original state space.

PROPOSITION 5. *Consider a control of the form* $x_S^{t+1} = Ax_S^t + Bu^t$ *and* $\hat{\mathcal{X}}_S^{t+1} = \psi_d(\,x_S^{t+1})$. *Then if the embedded LTI system is controllable, the nonlinear Koopman system is also controllable.*

Proof. The proof is deferred to Appendix D. ■

This completes our theoretical analysis on the stability and controllability of our framework demonstrating its favorable properties for the control of complex, nonlinear systems leveraging recent advances from Koopman operator theory.

4 Retrofit Case Study—Wire Arc Additive Manufacturing

WAAM is an important process in the field of additive manufacturing due to its ability to fabricate large-scale metal parts with cost-effectiveness and reduced lead times compared to traditional manufacturing methods. It involves the deposition of material through an electric arc welding process using a wire as the feedstock.

One of the key challenges in WAAM is achieving high 3D shape accuracy, which refers to the alignment between the intended computer-aided design (CAD) and the actual geometry of the manufactured part. Several factors contribute to the difficulty in obtaining accurate 3D shape quality in WAAM. Process disturbances such as thermal stresses and deformation can occur during the deposition of material. The wire feeding mechanism, arc stability, and weld pool dynamics can influence the material deposition and result in variations in the deposited layers. Controlling the heat input, wire feed rate, and arc parameters becomes crucial to maintain dimensional accuracy. Furthermore, material-related issues such as wire composition, wire diameter variation, and the presence of impurities can impact the quality and shape accuracy of the deposited layers.

Addressing these challenges requires advanced process monitoring and control techniques. Real-time monitoring and control of process variables such as deposition rate, and arc parameters can enable adaptive control strategies to mitigate distortions and improve 3D shape accuracy.

In the existing literature, there is limited work focused on the real-time control of 3D shapes in additive manufacturing. Several offline batch compensation strategies [31–33], and strategies such as toolpath optimization have been proposed [34]. Additionally, serval studies have employed 3D point clouds to analyze part dimensions but have primarily restricted their control efforts to 2D features like width and height [12,35]. This limitation results in a loss of valuable information and control feedback when compared to the direct utilization of 3D point clouds. Limited efforts have been devoted to enhancing in-process quality improvement with 3D point cloud data, despite its recent recognition as a critical avenue for research [36]. Based on a recent review article [37], we have compiled a summary and comparative analysis of the current state of the art in the control WAAM processes (Table 1). As underscored in this review, "compared to other areas of research, online sensing and control of the WAAM process are still in the early stages of development" [37]. Present research predominantly relies on limited sensing equipment and addresses only specific aspects of the process, resulting in constrained controllability.

As Xia et al. [37] pointed out, several defects are more pronounced in WAAM compared to traditional welding. While solutions for traditional welding defects like porosity or poor bead profile are readily available, issues that are more pronounced in WAAM, such as distortion, geometric accuracy, repeatability, and surface finish, require new developments. Therefore, we conducted experiments in a WAAM setting to demonstrate the potential of RETROFIT in this significant 3D printing process by leveraging advanced data fusion and nonlinear process control techniques.

Table 1 Summary of the state-of-the-art WAAM process control and comparison with our RETROFIT method

Paper	Control object	Controlled variables	Sensing method	Control strategy
Xiong and Zhang [38]	Layer height	Wire feed speed	Image	Adaptive control
Doumanidis and Kwak [39]	Layer height	Wire feed speed	Laser scanner	Internal model control
Xiong et al. [40]	Layer width	Travel speed	Image	Neural self-learning control
Bonaccorso et al. [41]	Layer height	Wire feed rate	Image	Linear control
Ours (RETROFIT)	3D profile/shape	Travel speed Wire feed rate Arc voltage	3D laser scanner Image as outputs Functional curves as inputs	3D profile-based nonlinear control

4.1 Experimental Setup. The setup of the WAAM system in this research as depicted in Fig. 6 consists of an industrial welder system coupled with a wire feeder and a six-axis industrial robot. The torch was mounted on the robot arm moving in the designated path to fabricate metal parts. The setup is complemented by a 3D scanner and high-speed camera. The data of arc voltage and current were acquired by an industrial acquisition card, which was controlled by the industrial PLC units. The filler wire used in this research was 316L stainless steel wire with a diameter of 1.2 mm. The substrate plate was a commercial 316L stainless steel plate with a thickness of 12 mm.

Before deposition, the surface of the substrate was cleaned by an angle grinder and then washed with acetone and alkaline solution, together with a subsequent drying process. Argon with a 99.99% purity was used as the shielding gas. The manufactured workpieces consist of straight lines, with 150 mm in length (y-direction), 7 mm in height (z-direction), and 7 mm in width (x-direction). The scanning path was scanned back and forth, which means that the scanning direction in the current layer is opposite to that in the previous layer. The standstill/dwell time during the transition to the next adjacent layer is 60 s. This dwell time is critical for ensuring proper solidification and cooling of each layer, thereby mitigating structural weaknesses and defects in the final product and managing residual stresses to minimize distortion. Additionally, it allows sufficient time for processing in situ (i.e., 3D scan) data and calculating control actions for the subsequent layer. Notably, the 60-s dwell time falls well within the constraints of our data acquisition and control deployment, which average 34.8 s and 14.1 s, respectively. Shorter dwell times can be achieved through better integration of the scanning system into the process loop and optimization of interfaces and computing. In this study, we show the prototypical application of the RETROFIT framework.

4.2 Process Modeling. To understand the process variability of the system, we conducted a space-filling Latin hypercube design of experiments (DOEs) with $N = 100$ experiments in total. The ranges of the process parameters are determined by preliminary experiments, which are shown in Table 2. Drawing from those initial experiments, we employed Argon gas as the shielding medium for the welding operation, maintaining a flowrate of 20 l/min throughout all experiments. Additionally, a contact tip-to-work distance of 10 mm was applied. The DOE produced samples with varying quality, even though the process limits were chosen within reasonable engineering limits.

To monitor the process, we collected various in situ measurements. For each experiment, we obtained 3D point cloud data for each of the $L = 5$ printing layers per part, allowing us to model the layer-wise 3D profile evolution and the propagation of the printed parts. In total, we collected in situ sensing data from six different sources, encompassing functional curves, image data, and 3D point clouds. An illustrative frame from the in situ video can be seen in Fig. 7. This video has been cropped to a size of 150×150, centered on the torch, and reduced to 150 frames per layer.

Even though the video and the 3D point clouds have some overlapping features, integrating both enhances understanding of the welding process: high-speed imaging captures dynamic weld arc behavior, while 3D scans offer spatial details of the weld joint. Combining these data sources enables the identification of correlations between process dynamics and geometric attributes, addressing the limitations of individual modalities. The heterogeneous process measurements, along with their typical dimensions, are detailed in Table 3. Regarding data preprocessing, the functional curves were standardized to a fixed length of 200 using dynamic time warping. The 3D point cloud for each build was cropped to remove any data points generated from scanning the work plate and preceding layers to reduce the data dimensionality. The 3D point clouds of each printing layer are further up- or down-sampled

Table 2 Process parameter ranges during wire arc additive manufacturing DOE experiments

Process setting	Range
Arc voltage (V)	20–30 V
Wire feed rate (m/min)	1000–10,000 mm/min
Travel speed (mm/min)	100–200 mm/min

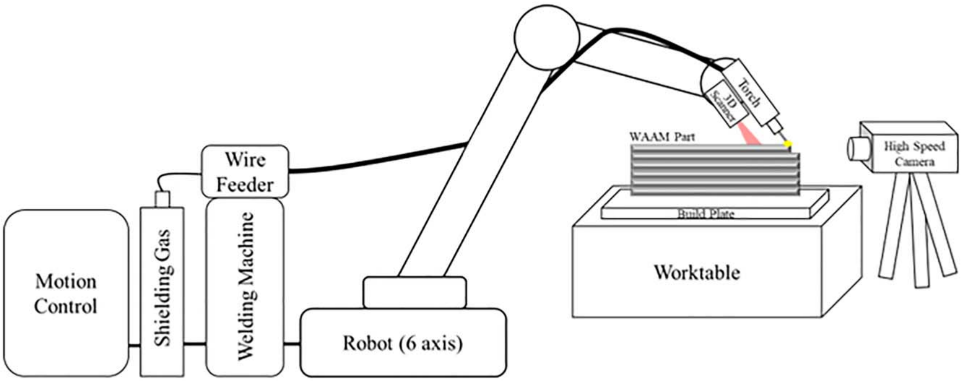

Fig. 6 Experimental Setup

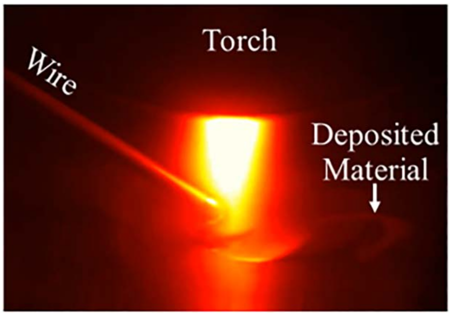

Fig. 7 Exemplary frame from in situ video

Table 3 In situ data description

Data source	Data type	Data dimension
Process settings: Monitored via welding machine:		
– Arc voltage (V) – Wire feed rate (m/min) – Travel speed (mm/min) – Arc current (A)	Functional curve	$\mathbb{R}^{200\times L}$
In situ imaging	Video	$150\times 150\times 150\times L_n$
3D scan	3D point cloud	$\mathbb{R}^{10{,}000\times 3\times L}$

to a fixed-point number of $N_p = 10{,}000$ resulting in a data dimension $\mathbb{R}^{10{,}000\times 3\times L_n}$ for each sample, where L_n denotes the number of printing layers.

To model the in situ data, we use the following data type-specific feature extractors: For the two curves (e.g., wire feed rate and travel speed), we utilize the deep convolutional neural network architecture proposed by Ref. [42] in an AE setting. As the feature extractor for the 3D scan video signals, we utilize a convolutional AE structure proposed by Ref. [43]. For specific implementation details, we refer readers to the online code repository, which will be released upon paper acceptance.

To assess the validity of our assumptions established in the theoretical analysis, we verified our assumptions on stability and controllability in the case study. The stability assessment relies on three crucial assumptions. First, we confirmed the Lipschitz continuity of the neural network architecture, a verification carried out using the AutoLip package [29], offering an automated approach for this purpose. The second assumption pertained to bounded noise, a natural assumption for physical systems. Third, we ensured the boundedness of prediction errors by comparing model predictions to ground truth data. Regarding controllability, we examined the full rank of the accessibility matrix. Although this matrix can be very large, its rank can be readily computed using a variety of software packages for SVD analysis.

Detailed information on the verification process can be found in the "Assumption Verification" section within the supplementary software documentation.

4.3 Offline Propagation Model.

To assess the effectiveness of our framework's two primary components, the offline propagation model and the online control model, we first examine the accuracy of the offline propagation model. Inaccuracies in offline predictions will propagate to the downstream control strategies, leading to inaccurate outcomes. Thus, we utilize two common point cloud metrics to assess prediction quality: first, the Chamfer (CH) distance, which calculates the distance between point clouds X_1 and the predicted point cloud X_2 as follows:

Table 4 Prediction results of the offline propagation model

Metric	Average	Standard deviation
CH	0.892	0.068
EMD	0.613	0.046

$$d_{\mathrm{CH}}(X_1, X_2) = \max\left\{\frac{1}{|X_1|}\sum_{x\in X_1}\min_{y\in X_2}\|x-y\|_2,\ \frac{1}{|X_2|}\sum_{y\in X_2}\min_{x\in X_1}\|y-x\|_2\right\} \tag{23}$$

Second, the Earth mover distance (EMD) or Wasserstein distance is utilized, which is defined as follows:

$$d_{\mathrm{EMD}}(X_1, X_2) = \min\left\{\frac{1}{|X_1|}\sum_{x\in X_1}\min_{y\in X_2}\|x-y\|_2,\ \frac{1}{|X_2|}\sum_{y\in X_2}\min_{x\in X_1}\|y-x\|_2\right\} \tag{24}$$

The average and standard deviation of a tenfold cross-validation for the entire offline training dataset are presented in Table 4, demonstrating strong agreement with ground truth predictions and limited variance.

Additionally, we provide a qualitative comparison for three exemplary samples in Fig. 8, revealing a significant overlap between the ground truth 3D profile (green) and the predicted 3D profile (blue), one step ahead by the offline propagation model. Notably, our method exhibits robustness to outliers, focusing on dense regions essential for quality assessment, occasionally missing measurement points captured from preceding layers, which are challenging to segment automatically for the sensing method.

4.4 Process Compensation and Control.

To provide closed-loop-feedback control and implement our proposed RETROFIT control strategy, we programmed several interfaces between the sensor equipment, the welding machine, and the six-axis robot. The compensation calculation is performed using a desktop computer that connects various system components. To prototype and demonstrate our control strategy while ensuring timely calculations, compensation actions are exclusively computed on a layer-by-layer basis. To validate our control strategy, we printed five samples under the optimal process conditions found during our design of experiments. The optimal conditions were found by measuring the mean average percentage error (MAPE) between the optimal CAD design and the 3D point clouds at respective layers. For our control method, we printed five samples, by starting the first layer with those same settings and then allowing our control strategy to take compensation actions for the subsequent layers. We use the following equation for the MAPE to measure the quality of the parts:

$$\mathrm{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\| p_i - q_i\|}{\| p_i\|}\cdot 100\%\right) \tag{25}$$

where n is the total number of points in the point cloud, p_i is the position vector of the ith point in the 3D point cloud, q_i is the position vector of the nearest point on the CAD model mesh to the p_i, and $\|.\|$ denotes the Euclidian norm. By averaging the respective five samples with and without control, we can see the superiority of our approach as summarized in Table 5. Compared to a scenario without control, our RETROFIT framework achieves approximately a 46.5% improvement.

Figure 9 displays an exemplary 3D point cloud plot for the third layer, with the build plate and other layer geometries cropped out to enhance visualization. With our RETROFIT control, it is evident

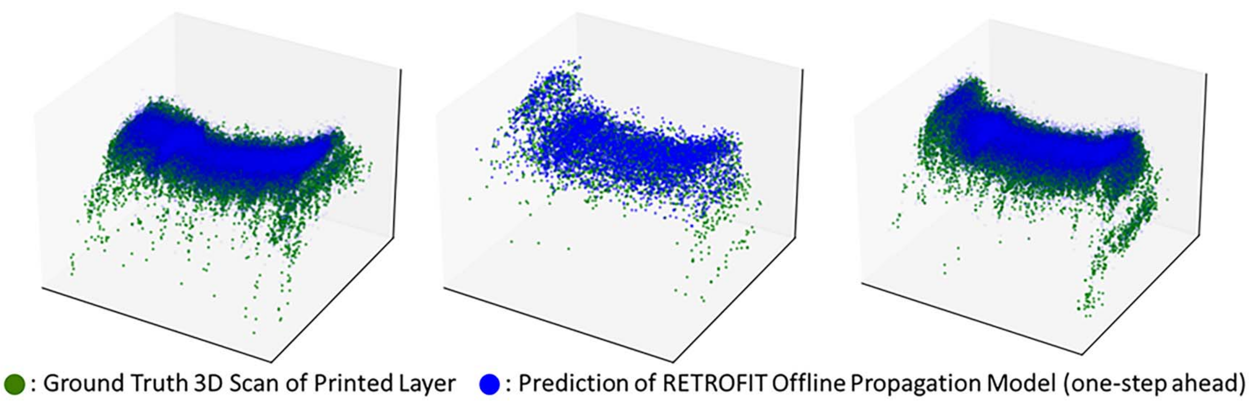

Fig. 8 Exemplary prediction results of the offline propagation model (Color version online.)

Table 5 MAPE of the five samples with RETROFIT control and five samples without control

Method	Average MAPE	Standard deviation of MAPE
Without control	34.81%	8.79%
With RETROFIT control (ours)	18.61%	3.55%

that the 3D shape geometry becomes significantly more uniform and approaches their near net shape.

Based on our experimental findings, it is evident that our control approach yields significantly more consistent WAAM deposition results, as depicted in Fig. 10.

Our experiments in WAAM have showcased the effectiveness of the RETROFIT framework in achieving precise 3D profile control within the additive manufacturing processes. While there are opportunities for further enhancements, such as sensor integration and

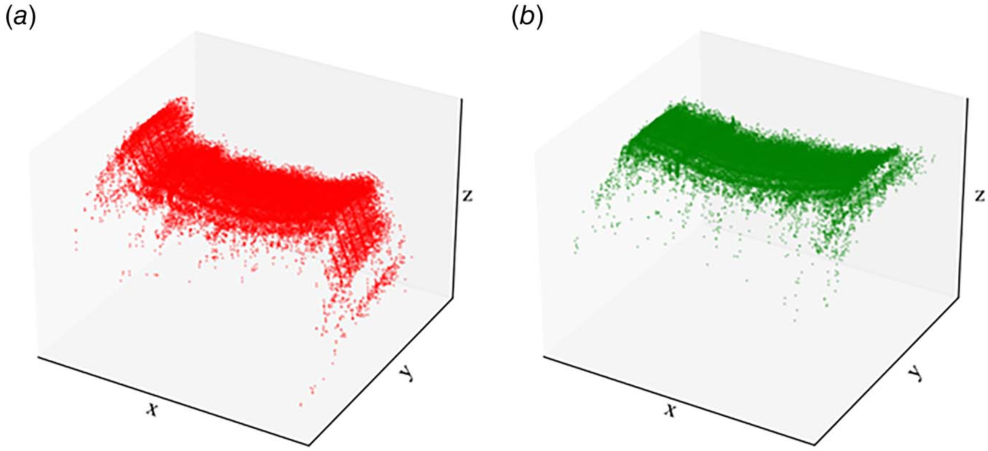

Fig. 9 Exemplary comparison of layer 3D geometry for (*a*) without control and (*b*) with our RETROFIT control

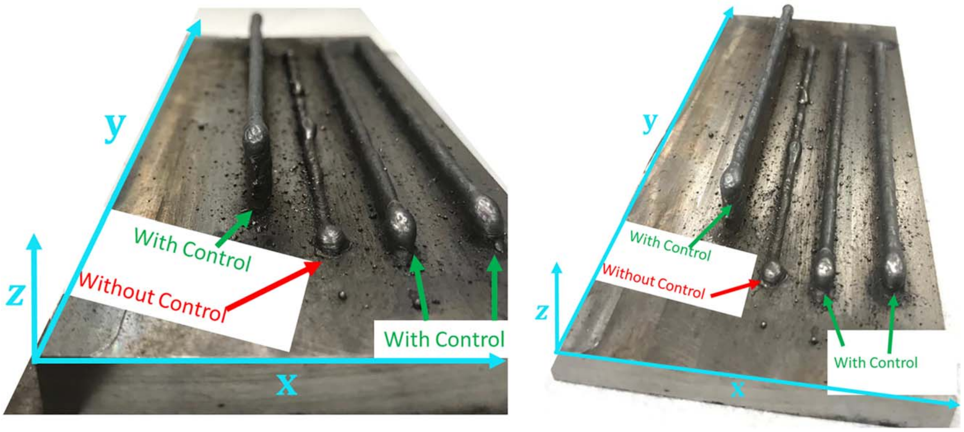

Fig. 10 Exemplary images of weld parts with and without control

control frequency, this case study underscores the potential of our methodological framework. Our framework supports multi-step ahead predictions (where N_{Pr} in Eq. (16) represents the prediction horizon). However, we argue that employing a one-step-ahead prediction approach suffices for several reasons. In layer-by-layer 3D printing, we adopt a compensation strategy that adjusts one layer at a time, considering inputs required for subsequent layers only if one-step-ahead prediction compensation proves insufficient. Errors in 3D printing processes are typically smaller than the printing thickness or equipment bandwidth, rendering one-step prediction optimal for each step without the need for dividing compensation into multiple steps.

5 Conclusion

This article introduces a methodological approach for feedback control in scenarios where the response takes the form of a 3D point cloud, the state evolution is nonlinear, and the process variables are high-dimensional data. Our proposed method consists of two key steps: (i) an offline modeling and estimation step that focuses on establishing a model connecting the output, and process variables, and (ii) control input variables using a training dataset, followed by an online control step that generates an optimal control law to minimize the deviation of the response from its target.

In the offline estimation step, we employ a novel approach for modeling dynamic 3D point cloud time-series data. After obtaining the model coefficients, we utilize a Koopman-based nonlinear control model to minimize the control objective function. To validate the effectiveness of our proposed method, we conducted a case study in WAAM.

The results of our study demonstrate that our method can significantly improve 3D shape accuracy, underscoring its efficacy in designing and implementing control systems utilizing 3D point cloud data. Further research is necessary to investigate the robustness of this approach across a wider range of process conditions and its applicability to other domains and processes.

Acknowledgment

This research is funded by National Science Foundation Award ID 2019378.

Conflict of Interest

There are no conflicts of interest.

Data Availability Statement

The datasets generated and supporting the findings of this article are obtainable from the corresponding author upon reasonable request.

Nomenclature

d	=	random noise bound
m	=	dimension of control input
t	=	time index
L	=	Lipschitz constant
J	=	convex quadratic cost function
S	=	shift operator
A_{cl}	=	closed-loop transition matrix
A_t	=	Koopman matrix
B_t	=	control matrix
N_p	=	number of Cartesian coordinate points
N_{pr}	=	prediction horizon
u^t	=	control inputs
AM	=	target profile
AE	=	autoencoder
ARIMAX	=	auto-regressive integrated moving average with eXogenous inputs
CAD	=	computer-aided design
Deep-DMD	=	deep learning-based dynamic mode decomposition
DETONATE	=	nonlinear dynamic evolution modeling of time-dependent three-dimensional point cloud profiles
DMD	=	dynamic mode decomposition
EDMD	=	extended dynamic mode decomposition
MAPE	=	mean average percentage error
MPC	=	model predictive control
PLC	=	programmable logic controller
RETROFIT	=	real-time control of time-dependent 3D point cloud profiles
SVD	=	singular value decomposition
WAAM	=	wire arc additive manufacturing
ε^t	=	random noise
$\mathcal{X}_S^t$	=	state space at time t
$\mathcal{S}$	=	manifold
$\mathcal{K}$	=	Koopman operator
φ	=	observable function
$\mathcal{U}$	=	control input space
$\mathcal{M}'$	=	function space
λ	=	tuning parameters
$\mathcal{L}_{\text{rec}}$	=	reconstruction loss
$\mathcal{L}_{\text{fwd}}$	=	forward prediction loss
$\mathcal{L}_h$	=	heterogeneous data loss
$\hat{\mathcal{X}}_S^t$	=	predicted system state at time t
ψ_d	=	decoder
ψ_e	=	encoder
κ_{embed}	=	feedback controller in embedded space
κ	=	feedback controller in original space
ϕ	=	embedding functions
γ, η	=	bounding constants
$\mathbb{X}$	=	state space
$\mathbb{U}$	=	action space

Appendix A: Proof of Proposition 1

Proof: The stability proof comprises two main components: the stability analysis of the nominal system and the assessment of the stability of the error system, which quantifies the deviation between the actual system and the nominal one. Note that for the theoretical analysis, we assume that the Koopman matrix A_t and control matrix B_t are time-invariant (i.e., A and B). This assumption is suitable for the parts manufactured in the WAAM case study, but for other systems or more complex parts, a time-varying Koopman matrix A_t and control matrix B_t may be necessary. However, it is worth mentioning that extending the derived propositions to accommodate time-varying matrices needs further study.

First, We Prove the Stability of the Nominal System (Eq. (1)). By solving Eqs. (16) and (17), the optimal control sequence

$$\{u^*_{tt|}, u^*_{t+1|t}, \ldots, u^*_{t+N_{\text{Pr}}-1|t}\} \tag{A1}$$

and the resulting optimal state trajectory

$$\{\hat{x}^*_{S_{t|t}}, \hat{x}^*_{S_{t+1|t}}, \ldots, \hat{x}^*_{S_{t+N_{\text{Pr}}-1|t}}\} \tag{A2}$$

at instant t are obtained. By appending the control signal produced by the feedback controller $\kappa_{\hat{\mathcal{X}}^*_{S_{t+N_{\text{Pr}}|t}}}$ to Eq. (A2), a sub-optimal solution at next time-step $t+1$ is given by

$$\{u^*_{t|t}, u^*_{t+1|t}, \ldots, u^*_{t+N_{\text{Pr}}-1|t}, \kappa_{\hat{x}_S t+N_{\text{Pr}}|t^*}\} \tag{A3}$$

and

$$\{\hat{x}^*_{S_{t|t}}, \hat{x}^*_{S_{t+1|t}}, \ldots, \hat{x}^*_{S_{t+N_{\text{Pr}}-1|t}}, A_{cl}\hat{x}^*_{S_{t+N_{\text{Pr}}|t}}\} \tag{A4}$$

where $A_{cl} = A + B\kappa$ denotes the closed-loop transition matrix. Based on this sub-optimal solution, we can now prove that the optimal value function $J^*\left(\hat{x}^*_{S_{t+1}}\right)$ decreases along this trajectory. Due to the suboptimality of Eqs. (A3) and (A4), we have

$$\begin{aligned}J^*(\hat{x}^*_{S_{t+1}}) &\le \sum_{k=1}^{N_{\mathrm{Pr}}-1} q(\hat{x}^*_{S_{t+k|t}}, u^*_{t+k|t}) + q(\hat{x}^*_{S_{t+N_{\mathrm{Pr}}|t}}, u^*_{t+N_{\mathrm{Pr}}|t}) + p(A_{cl}\hat{x}^*_{S_{t+N_{\mathrm{Pr}}|t}}) \\ &= J^*(\hat{x}^*_{S_t}) + q(\hat{x}^*_{S_{t+N_{Pr}|t}}, \kappa\, u^*_{t+N_{Pr}|t}) + p(A_{cl}\hat{x}^*_{S_{t+N_{Pr}|t}}) - q(\hat{x}^*_{S_{t|t}}, u^*_{t|t})\end{aligned} \tag{A5}$$

where $q(\hat{x}_{S_{t|t}}, u_{t|t}) = \|x_S^{,t} Q_t x_{SQ}^{,t2}\| + \|u_{t_{R^2}}\|$ denotes the cost at a given time-step and $p(\hat{x}_{S_{t|t}}) = \|x_S^{,N_{\mathrm{Pr}}} Q_{N_{\mathrm{Pr}}} x_S^{,N_{\mathrm{Pr}}}\|_Q^2$ denotes the terminal cost. Hence it follows that

$$J^*(\hat{x}^*_{S_{t+1}}) - J^*(\hat{x}^*_{S_t}) \le -q(\hat{x}^*_{S_{|t}}, u^*_{t|t}) \tag{A6}$$

and the optimal value function $J^*(.)$ is a valid Lyapunov function. Therefore, the expectation of the nominal state $\mathbb{E}\hat{x}_{S,t}$ converges to zero as $t \to \infty$, that is the nominal state is mean square stable.

Second, Let Us Quantify the Deviation Bound of the Dynamics of the Error System $e^t = \hat{x}_S^t - x_S^t$. Substituting the controller (Eq. (18)) into the error system, it follows that

$$e^{t+1} = (\,A \;+\; B\kappa)e^t + \varepsilon^t \tag{A7}$$

Iterating the dynamics of the error system (Eq. (A7)) from the initial time instance 1 to t with $e^{t+1} = A_{cl}\varepsilon^t + A_{cl}^2\varepsilon^{t-1} + \ldots + A_{cl}^t\varepsilon^1 + A_{cl}^t e_1$. According to Algorithm 1, e_1 equals to zero as $x_S^0 = \psi_e(\mathcal{X}_S^0)$. Then the L_2 norm of the error state is given by

$$\begin{aligned}\|e^{t+1}\| &= \|A_{cl}\varepsilon^t + A_{cl}^2\varepsilon^{t-1} + \ldots + A_{cl}^t\varepsilon^1\| \\ &\le \|A_{cl}\varepsilon^t\| + \|A_{cl}^2\varepsilon^{t-1}\| + \ldots + \|A_{cl}^t\varepsilon^1\| \\ &\le \beta\|\varepsilon^t\| + \beta^2\|\varepsilon^{t-1}\| + \ldots + \beta^t\|\varepsilon^1\|\end{aligned} \tag{A8}$$

where β denotes the maximum eigenvalue of the closed-loop transition matrix A_{cl}.

Taking the expectation over the random noise ε^t, and using the fact that the random noise at different time instances is independently distributed, it follows that

$$\mathbb{E}\|e^{t+1}\| \le \beta\mathbb{E}_{\varepsilon^t}\|\varepsilon^t\| + \beta^2\mathbb{E}_{\varepsilon^{t-1}}\|\varepsilon^{t-1}\| + \ldots + \beta^t\mathbb{E}_{\varepsilon^1}\|\varepsilon^1\| \tag{A9}$$

According to Assumption 1, we can further infer that

$$\begin{aligned}\mathbb{E}\|e^{t+1}\| &\le \beta L\mathbb{E}_{\varepsilon^t}\|\varepsilon^t\| + \beta^2 L\mathbb{E}_{\varepsilon^{t-1}}\|\varepsilon^{t-1}\| + \ldots + \beta^t L\mathbb{E}_{\varepsilon^1}\|\varepsilon^1\| \\ &\le \beta Lb + \beta^2 Lb + \ldots + \beta^t Lb \\ &= \frac{(\beta - \beta^t)Lb}{1-\beta}\end{aligned} \tag{A10}$$

where L is the Lipschitz constant defined in Assumption 1. The second inequality is a direct result of Assumption 2. As $t \to \infty$, the expectation of the error state norm is bounded by $\frac{\beta Lb}{1-\beta}$. The state of the original system is given by $\mathcal{X}_S^t = \psi_d(x_s^t + \mathbb{E}(e^t))$, and thus the effect of the error state upon the original state is bounded by $\frac{\beta\sigma Lb}{1-\beta}$, where $\sigma := \|\psi_d\|$. Because the nominal state is mean square stable and the error between the actual and nominal state is bounded, the system (Eq. (1)) is proven to be uniformly ultimately bounded. ■

Appendix B: Proof of Proposition 2

Proof: Since the nominal system remains the same as in Proposition 1, the proof for the mean square stability of the nominal system is identical as well. We focus on proving the boundedness of the error system. In the presence of approximation residuals, the dynamic of the error system $e^t = \hat{x}_S^t - x_S^t$ is given as follows:

$$e^{t+1} = (A \;+\; B\kappa)e^t + \varepsilon^t + d^t \tag{B1}$$

Iterating the above Eq. (B1) from the initial time instance 1 to t, we have

$$\begin{aligned}e^{t+1} &= A_{cl}^t e_1 + (A_{cl}d^t + A_{cl}^2 d^{t-1} + \ldots + A_{cl}^t d^1) \\ &\quad + (A_{cl}\varepsilon^t + A_{cl}^2\varepsilon^{t-1} + \ldots + A_{cl}^t\varepsilon^1) \\ &= A_{cl}^t e_1 + \sum_{k=1}^{t} A_{cl}^k d^k + \sum_{k=1}^{t} A_{cl}^k\varepsilon^k\end{aligned} \tag{B2}$$

According to Algorithm 1, e_1 equals to zero as $x_S^0 = \psi_e(\mathcal{X}_S^0)$. Then the L_2 norm of the error state is given by

$$\begin{aligned}\|e^{t+1}\| &= \left\|\sum_{k=1}^{t} A_{cl}^k d^k + \sum_{k=1}^{t} A_{cl}^k\varepsilon^k\right\| \\ &\le \sum_{k=1}^{t}\|A_{cl}^k d^k\| + \sum_{k=1}^{t}\|A_{cl}^k\varepsilon^k\| \\ &\le \sum_{k=1}^{t}\beta^k\|d^k\| + \sum_{k=1}^{t}\beta^k\|\varepsilon^k\|\end{aligned} \tag{B3}$$

Taking the expectation over the random noise ε^t it follows that

$$\begin{aligned}\mathbb{E}\|e_{t+1}\| &\le \sum_{k=1}^{t}\beta^k\|d^k\| + \sum_{k=1}^{t}\beta^k\|\varepsilon^k\| \\ &\le \sum_{k=1}^{t}\beta^k\gamma + \sum_{k=1}^{t}\beta^k Lb \\ &= \frac{(\beta - \beta^t)(Lb + \gamma)}{1-\beta}\end{aligned} \tag{B4}$$

where the second inequality is a direct result of Assumptions 2 and 3. As $t \to \infty$, the expectation of the error state norm is bounded by $\frac{\beta Lb + \gamma}{1-\beta} + \eta$. Because the nominal state is mean square stable and the error between the actual and nominal state is bounded by a constant $\frac{\beta\sigma(Lb + \gamma)}{1-\beta} + \eta$, the system is proven to be uniformly ultimately bounded. ■

Appendix C: Proof of Proposition 4

Proof: To analyze the local and global controllability of the LTI system, we can consider the accessibility distribution matrix Q based on the Lie brackets of the vector fields associated with the system. For the LTI system, we have the vector fields $f_d = Ax_S$ and $\theta(x_S) = B$. The Lie bracket among those vector fields is computed as $[f_d,\ \theta] = -AB$. We find that the higher-order lie brackets between these vector fields satisfy $(ad^k f_d,\ \theta) = (-1)^k A^k B$ for $k > 1$. Using the Lie brackets, and the Cayley–Hamilton theorem, we can compute the accessibility matrix Q as follows:

$$Q = [B \quad -AB \quad (-1)^2A^2B \;\ldots\; (-1)^{n-1}A^{n-1}B] \tag{C1}$$

Hence, the LTI system is locally controllable at $x_S^0 \in \mathbb{R}^{d_m}$ if the rank of Q at x_S^0 is d_m and from Proposition 3 and Definition 3, it is globally controllable if $\mathrm{rank}(Q) = d_m$ for all $x_s^t \in \mathbb{R}^{d_m}$. ■

Appendix D: Proof of Proposition 5

Proof: We establish Proposition 5 through a contrapositive argument. We begin by assuming that for the nonlinear system described by the equation $x_S^{t+1} = Ax_S^t + Bu^t$ and $\hat{\mathcal{X}}_S^{t+1} = \psi_d(\,x_S^{t+1})$, there exist initial state x_S^0 and target state x_S^T such that there is no control input u^t that can drive the system trajectory from x_S^0 to x_S^T within the given time duration T.

Let $T(x_S^0, \boldsymbol{u})$ denote the set of all possible control trajectories starting from the initial state x_S^0. We assert that $(x_S^0, \boldsymbol{u}) \cap x_S^T = \{\phi\}$, where $\{\phi\}$ denotes the empty set, implying that no trajectory from x_S^0 intersects with the target state x_S^T within the specified time frame.

Now, we consider the mapping ψ_d, which represents the original state space associated with the system. Given that ψ_d is an injective mapping, it ensures that distinct states in the state space are associated with distinct observables.

Therefore, in the lifted space, we can conclude that $\Phi(T(x_S^0, \boldsymbol{u})) \cap \Phi(x_S^T) = \{\phi\}$. This implies that the images of the set of control trajectories $\Phi(T(x_S^0, \boldsymbol{u}))$ and the observable associated with the target state $\Phi(x_S^T)$ are disjoint, having no common elements.

Hence, if the embedded Koopman system is deemed uncontrollable in the sense that there exist x_S^0 and x_S^T for which no control input u^t can achieve the desired trajectory within time T, then it follows that the original system is also uncontrollable. In other words, if the lifted Koopman bilinear system is shown to be controllable, then this implies that the original nonlinear system, described by $x_S^{t+1} = Ax_S^t + Bu^t$ and $\hat{\mathcal{X}}_S^{t+1} = \psi_d(\,x_S^{t+1})$, is indeed controllable. ■

References

[1] Biehler, M., Lin, D., and Shi, J., 2023, "DETONATE: Nonlinear Dynamic Evolution Modeling of Time-Dependent 3-Dimensional Point Cloud Profiles," IISE Trans., **56**(5), pp. 541–558.

[2] Zhen, Z., Paynabar, K., and Shi, J., 2023, "Image-Based Feedback Control Using Tensor Analysis," Technometrics, **65**(3), pp. 305–314.

[3] Rawlings, J. B., Mayne, D. Q., and Diehl, M., 2017, *Model Predictive Control: Theory, Computation, and Design*, Nob Hill Publishing, Madison, WI.

[4] Jin, J., and Shi, J., 1999, "State Space Modeling of Sheet Metal Assembly for Dimensional Control," ASME J. Manuf. Sci. Eng., **121**(4), pp. 756–762.

[5] Williams, R. L., and Lawrence, D. A., 2007, *Linear State-Space Control Systems*, John Wiley & Sons, Hoboken, NJ.

[6] Berber, R., and Kravaris, C., 2012, *Nonlinear Model Based Process Control*, Springer Science & Business Media, Berlin, Germany.

[7] Landers, R. G., Barton, K., Devasia, S., Kurfess, T., Pagilla, P., and Tomizuka, M., 2020, "A Review of Manufacturing Process Control," ASME J. Manuf. Sci. Eng., **142**(11), p. 110814.

[8] Sikder, S., Mukherjee, I., and Panja, S. C., 2020, "A Synergistic Mahalanobis–Taguchi System and Support Vector Regression Based Predictive Multivariate Manufacturing Process Quality Control Approach," J. Manuf. Syst., **57**, pp. 323–337.

[9] Babu, A., Yurtdas, K. Y., Koch, C. E. S., and Yüksel, M., 2019, "Trajectory Following Using Nonlinear Model Predictive Control and 3D Point-Cloud-Based Localization for Autonomous Driving," 2019 European Conference on Mobile Robots (ECMR), Prague, Czech Republic, Sept. 4–6, IEEE, pp. 1–6.

[10] Ruiz, C., Bhatt, P. M., Gupta, S. K., and Huang, Q., 2023, "Process-Informed Segmentation of Dense Point Clouds for Layer Quality Assessment in Large-Scale Metal Additive Manufacturing," 2023 IEEE 19th International Conference on Automation Science and Engineering (CASE), Auckland, New Zealand, Aug. 26–30, IEEE, pp. 1–6.

[11] Shen, H., Jin, J., Liu, B., and Zhou, Z., 2021, "Measurement and Evaluation of Laser-Scanned 3D Profiles in Wire Arc Hybrid Manufacturing Processes," Measurement, **176**, p. 109089.

[12] Tang, S., Wang, G., and Zhang, H., 2019, "In Situ 3D Monitoring and Control of Geometric Signatures in Wire and Arc Additive Manufacturing," Surf. Topogr.: Metrol. Prop., **7**(2), p. 025013.

[13] Wu, R., Yu, Z., Ding, D., Lu, Q., Pan, Z., and Li, H., 2021, "OICP: An Online Fast Registration Algorithm Based on Rigid Translation Applied to Wire Arc Additive Manufacturing of Mold Repair," Materials, **14**(6), p. 1563.

[14] Biehler, M., Yan, H., and Shi, J., "ANTLER: Bayesian Nonlinear Tensor Learning and Modeler for Unstructured, Varying-Size Point Cloud Data," IEEE Trans. Autom. Sci. Eng., **21**(1), pp. 402–415.

[15] Proctor, J. L., Brunton, S. L., and Kutz, J. N., 2016, "Dynamic Mode Decomposition With Control," SIAM J. Appl. Dyn. Syst., **15**(1), pp. 142–161.

[16] Williams, M. O., Kevrekidis, I. G., and Rowley, C. W., 2015, "A Data-Driven Approximation of the Koopman Operator: Extending Dynamic Mode Decomposition," J. Nonlinear Sci., **25**(6), pp. 1307–1346.

[17] Yeung, E., Kundu, S., and Hodas, N., 2019, "Learning Deep Neural Network Representations for Koopman Operators of Nonlinear Dynamical Systems," 2019 American Control Conference (ACC), IEEE, pp. 4832–4839.

[18] Schmid, P. J., 2022, "Dynamic Mode Decomposition and Its Variants," Annu. Rev. Fluid Mech., **54**(1), pp. 225–254.

[19] Lusch, B., Kutz, J. N., and Brunton, S. L., 2018, "Deep Learning for Universal Linear Embeddings of Nonlinear Dynamics," Nat. Commun., **9**(1), p. 4950.

[20] Morton, J., Jameson, A., Kochenderfer, M. J., and Witherden, F., 2018, "Deep Dynamical Modeling and Control of Unsteady Fluid Flows," Thirty-Second Annual Conference on Neural Information Processing Systems (NIPS), Montreal, Canada, Dec. 2–8.

[21] Otto, S. E., and Rowley, C. W., 2019, "Linearly Recurrent Autoencoder Networks for Learning Dynamics," SIAM J. Appl. Dyn. Syst., **18**(1), pp. 558–593.

[22] Achlioptas, P., Diamanti, O., Mitliagkas, I., and Guibas, L., 2018, "Learning Representations and Generative Models for 3D Point Clouds," International Conference on Machine Learning, Stockholm, Sweden, July 20–25, PMLR, pp. 40–49

[23] Brunton, S. L., Budišić, M., Kaiser, E., and Kutz, J. N., 2022, "Modern Koopman Theory for Dynamical Systems," SIAM Rev., **64**(2), pp. 229–340.

[24] Korda, M., and Mezić, I., 2018, "Linear Predictors for Nonlinear Dynamical Systems: Koopman Operator Meets Model Predictive Control," Automatica, **93**, pp. 149–160.

[25] Proctor, J. L., Brunton, S. L., and Kutz, J. N., 2018, "Generalizing Koopman Theory to Allow for Inputs and Control," SIAM J. Appl. Dyn. Syst., **17**(1), pp. 909–930.

[26] Williams, M. O., Hemati, M. S., Dawson, S. T., Kevrekidis, I. G., and Rowley, C. W., 2016, "Extending Data-Driven Koopman Analysis to Actuated Systems," IFAC-PapersOnLine, **49**(18), pp. 704–709.

[27] Bartlett, P. L., Foster, D. J., and Telgarsky, M. J., 2017, "Spectrally-Normalized Margin Bounds for Neural Networks," The Thirty-First Annual Conference on Neural Information Processing Systems (NIPS), Long Beach, CA, Dec. 4–9.

[28] Mezić, I., 2022, "On Numerical Approximations of the Koopman Operator," Mathematics, **10**(7), p. 1180.

[29] Virmaux, A., and Scaman, K., 2018, "Lipschitz Regularity of Deep Neural Networks: Analysis and Efficient Estimation," Thirty-Second Annual Conference on Neural Information Processing Systems (NIPS), Montreal, Canada, Dec. 2–8.

[30] Bullo, F., and Lewis, A. D., 2019, *Geometric Control of Mechanical Systems: Modeling, Analysis, and Design for Simple Mechanical Control Systems*, Springer, New York.

[31] Heinrich, L., Feldhausen, T., Saleeby, K. S., Saldana, C., and Kurfess, T. R., 2022, "Analysis of Conduction Cooling Strategies for Wire Arc Additive Manufacturing," International Manufacturing Science and Engineering Conference, Vol. 85819, American Society of Mechanical Engineers, p. V002T05A005.

[32] Huang, Q., Wang, Y., Lyu, M., and Lin, W., 2020, "Shape Deviation Generator – A Convolution Framework for Learning and Predicting 3-D Printing Shape Accuracy," IEEE Trans. Autom. Sci. Eng., **17**(3), pp. 1486–1500.

[33] Ruiz, C., Jafari, D., Venkata Subramanian, V., Vaneker, T. H., Ya, W., and Huang, Q., 2022, "Prediction and Control of Product Shape Quality for Wire and Arc Additive Manufacturing," ASME J. Manuf. Sci. Eng., **144**(11), p. 111005.

[34] Lam, T. F., Xiong, Y., Dharmawan, A. G., Foong, S., and Soh, G. S., 2020, "Adaptive Process Control Implementation of Wire arc Additive Manufacturing for Thin-Walled Components With Overhang Features," Int. J. Adv. Manuf. Technol., **108**, pp. 1061–1071.

[35] Dharmawan, A. G., Xiong, Y., Foong, S., and Soh, G. S., 2020, "A Model-Based Reinforcement Learning and Correction Framework for Process Control of Robotic Wire Arc Additive Manufacturing," 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, May 31–Aug. 31, IEEE, pp. 4030–4036.

[36] Shi, J., 2022, "In-Process Quality Improvement: Concepts, Methodologies, and Applications," IISE Trans., **55**(1), pp. 2–21.

[37] Xia, C., Pan, Z., Polden, J., Li, H., Xu, Y., Chen, S., and Zhang, Y., 2020, "A Review on Wire arc Additive Manufacturing: Monitoring, Control and a Framework of Automated System," J. Manuf. Syst., **57**, pp. 31–45.

[38] Xiong, J., and Zhang, G., 2014, "Adaptive Control of Deposited Height in GMAW-Based Layer Additive Manufacturing," J. Mater. Process. Technol., **214**(4), pp. 962–968.

[39] Doumanidis, C., and Kwak, Y.-M., 2001, "Geometry Modeling and Control by Infrared and Laser Sensing in Thermal Manufacturing With Material Deposition," ASME J. Manuf. Sci. Eng., **123**(1), pp. 45–52.

[40] Xiong, J., Yin, Z., and Zhang, W., 2016, "Closed-Loop Control of Variable Layer Width for Thin-Walled Parts in Wire and Arc Additive Manufacturing," J. Mater. Process. Technol., **233**, pp. 100–106.

[41] Bonaccorso, F., Cantelli, L., and Muscato, G., 2011, "An Arc Welding Robot Control for a Shaped Metal Deposition Plant: Modular Software Interface and Sensors," IEEE Trans. Ind. Electron., **58**(8), pp. 3126–3132.

[42] Yang, J., Nguyen, M. N., San, P. P., Li, X. L., and Krishnaswamy, S., 2015, "Deep Convolutional Neural Networks on Multichannel Time Series for Human Activity Recognition," Twenty-Fourth International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina, July 25–31, pp. 3995–4001.

[43] Valdarrama, S. Convolutional Autoencoder for Image Denoising. https://keras.io/examples/vision/autoencoder/, Accessed February 2, 2024.