



An impulse response formulation for small-sample learning and control of additive manufacturing quality

Qiang Huang 

Daniel J. Epstein Department of Industrial and Systems Engineering, University of Southern California, Los Angeles, CA, USA

ABSTRACT

Machine learning for additive manufacturing (ML4AM) has emerged as a viable strategy in recent years to enhance 3D printing performance. However, the amount of data required for model training and the lack of ability to infer AM process insights can be serious barriers for black-box learning methods. Due to the nature of low-volume fabrication of infinite product variety in AM, ML4AM also faces “small data, big tasks” challenges to learn heterogeneous point cloud data and control the quality of new designs. To address these challenges, this work establishes an impulse response formulation of layer-wise AM processes to relate design inputs with the deformed final products. To enable prescriptive learning from a small sample of printed parts with different 3D shapes, we develop a fabrication-aware input–output representation, where each product is constructed by a large amount of basic shape primitives. The impulse response model depicts how the 2D shape primitives (circular sectors, line segments, and corner segments) in each layer are stacked up to become final 3D shape primitives. A geometric quality of a new design can therefore be predicted through the construction of learned shape primitives. Essentially, the small-sample learning of printed products is transformed into a large-sample learning of printed shape primitives under the impulse response formulation of AM. This fabrication-aware formulation builds the foundation for applying well-established control theory to the intelligent quality control in AM. It not only provides theoretical underpinning and justification of our previous work, but also enable new opportunities in ML4AM. As an example, it leads to transfer function characterization of AM processes to uncover process insights. It also provides block-diagram representation of AM processes to design and optimize the control of AM quality.

ARTICLE HISTORY

Received 12 December 2021
Accepted 10 July 2022

KEYWORDS

Fabrication-aware representation and learning; impulse response modeling; transfer function; block-diagram; intelligent quality control

1. Introduction

Geometric and dimensional quality has long been essential to ensure product functionality and correct mechanical assemblies in complex engineering systems (Walker and Srinivasan, 1994; Srinivasan, 1999). As 3D printing (aka Additive Manufacturing; AM) technologies rapidly advance with broader applications in aerospace, automotive, medical industries, and beyond, intelligent Quality Control (QC) has become increasingly critical for cost-effective precision printing (Huang *et al.*, 2015; Colosimo *et al.*, 2018; Huang *et al.*, 2020).

However, AM poses unique challenges to QC (Colosimo *et al.*, 2018). Although it enables direct fabrication of products with complex geometries in a single production phase without tooling and fixturing, each phase actually consists of thousands of correlated steps forming and accumulating materials layer by layer. During printing, a multitude of factors such as materials, processing techniques, settings, and inter-layer interactions may cause printing defects. Physical modeling and simulation of complete layer-wise fabrication is still computationally prohibitive for timely prediction and QC operations

(Bourell *et al.*, 2009; King *et al.*, 2015). Multiple iterations or trial builds drive up printing costs significantly.

Quality verification of AM-built products is also a highly non-trivial procedure. Established primarily for multi-stage subtractive manufacturing, standard quality verification methods often depend on features fabricated in the previous stages as datum references to verify the quality of those made later (e.g., distance or parallelism of plane feature relative to a datum). Since AM builds complex geometries in a layer-wise fashion, representation and verification of complex freeform surfaces demand AM-specific specification standards and tolerancing methods. This has been widely recognized by various international organizations and engineering communities as one of the top priorities to reduce the risk and cost of adopting rapid-growing AM technologies (Ameta *et al.*, 2015; Morse *et al.*, 2018; Leach *et al.*, 2019).

Unlike scale-driven manufacturing, where operation expertise can be built up around a limited number of product families, AM aims at a versatile capability of making a theoretically infinite variety of products. Learning from similar cases or past experience finds difficulty in defining

similarity, particularly for low-volume AM with frequent design changes and heterogeneous AM processes. Current QC operations are heavily dependent on human expertise and intervention.

Lately, Machine Learning for AM (ML4AM) has emerged as a viable strategy to enhance 3D printing performance (Huang *et al.*, 2014; Huang *et al.*, 2015; Luan and Huang, 2017; Samie Tootooni *et al.*, 2017; Khanzadeh *et al.*, 2018; Sabbaghi and Huang, 2018; Sabbaghi *et al.*, 2018; Tsung *et al.*, 2018; de Souza Borges Ferreira *et al.*, 2020; Huang *et al.*, 2020; Wang *et al.*, 2022). General-purpose machine learning and analysis of 3D shapes have been extensively studied in computer vision for shape analysis. It focuses on shape classification, matching, deformation (e.g., facial expression change), and correspondence (Montagnat *et al.*, 2001; Zhang and Lu, 2004; Van Kaick *et al.*, 2011). However, purely geometric analysis without consideration of engineering mechanisms of shape generation limits the scope of applying general-purpose Machine Learning (ML) techniques to manufacturing. For example, deformation patterns of 3D printed products not only vary with shape, but also object size, due to thousands of correlated steps of layer formation and accumulation (Huang *et al.*, 2015; Jin *et al.*, 2016; Huang *et al.*, 2020). Currently, establishing fabrication-aware representation and learning of manifold-valued shape data are open issues (Bermano *et al.*, 2017). Progress in ML4AM has been made for specific tasks such as empirical and statistical modeling of AM processes (Zhou *et al.*, 2000; Tong *et al.*, 2003; Campanelli *et al.*, 2007; Tong *et al.*, 2008), sensing and inspection (Tapia and Elwany, 2014; Everton *et al.*, 2016; Samie Tootooni *et al.*, 2017; Khanzadeh *et al.*, 2018), statistical shape analysis (Colosimo *et al.*, 2008; del Castillo and Colosimo, 2011; del Castillo *et al.*, 2015), monitoring and detection (Grasso *et al.*, 2018).

Despite these advances in ML4AM, the large amount of data required for model training and the lack of model interpretability and scalability are serious barriers to the application of black-box learning methods to manufacturing. Product shape complexity, process complexity, and data heterogeneity in AM further complicate the QC efforts in AM. Recent exciting interactions between model-driven control theory and data-driven ML are motivated both by methodological development and practical applications such as self-driving cars and advanced robotics (Pillonetto *et al.*, 2014; Bensoussan *et al.*, 2020). Although this line of research provides a promising direction to achieve interpretability and principled design of ML methods, there is a lack of engineering-informed formulation of AM processes to enable model-driven ML4AM.

The current article reports on work to fill this gap by establishing an impulse response formulation of layer-wise AM processes. To enable prescriptive learning from a small sample of printed parts with different 3D shapes, we develop a fabrication-aware input-output representation, where each product is constructed by a large amount of basic shape primitives. The impulse response model depicts how the 2D shape primitives (circular sectors, line segments, and corner

segments) in each layer are stacked up to become the final 3D shape primitives. The geometric quality of a new design can therefore be predicted through the construction of learned shape primitives. Essentially, the small-sample learning of printed products is transformed into a large-sample learning of printed shape primitives under the impulse response formulation of AM.

This impulse response formulation of AM processes provides a control-theoretic justification of our convolution modeling framework for AM (Huang *et al.*, 2020), due to the convolution form being the solution of the impulse response system. By extending our prescriptive 2D freeform shape modeling approaches (Huang *et al.*, 2014; Huang *et al.*, 2015; Luan and Huang, 2017), this work provides, for the first time, a methodological framework suitable for 3D freeform shape quality prediction based on shape primitives generated under the impulse response formulation. Furthermore, this new control-theoretic formulation of AM builds the foundation for applying control theory to the intelligent QC. As an example, it leads to transfer function characterization of AM processes to uncover process insights. It also provides block-diagram representation of AM processes to design and optimize the control of AM quality.

Following the Introduction, Section 2 defines a set of open ML4AM problems for intelligent QC in AM. Section 3 establishes an impulse response formulation of layer-wise AM processes. Based on fabrication-aware representation of process input and output, impulse response functions are derived to characterize AM processes. Transfer functions and block diagrams are readily obtained to describe, design, and analyze AM process. Examples are presented to demonstrate the developed theories. Summary and conclusions are given in Section 4.

2. Problem definition for fabrication-aware ML and control in AM

2.1. Problem definition

To achieve intelligent QC for precision 3D printing, we define three key categories of open problems where control-theoretic formulation and learning has the potential to significantly advance both fundamental understanding and computational AM research. Note that the problem definition below is far from being comprehensive. Other critical categories of problems are discussed in the end.

Let input u to a 3D printing process be the designed shape of a 3D object. The set of design shapes is denoted as U . The output y represents the shape quality or the surface deformation/deviation of the actual printed product from its intended design u . The set of shape deviations of $u \in U$ is denoted as Y_U :

- **Learning Problem (LP):** The learning objective is to establish functional mapping $f: U \rightarrow Y_U$, that is, to generate a model $f(u)$ to predict y by learning from a small set of training data $D_S = \{(u, y_u) | (u_i, y_{u_i}), i = 1, 2, \dots, n\}$.

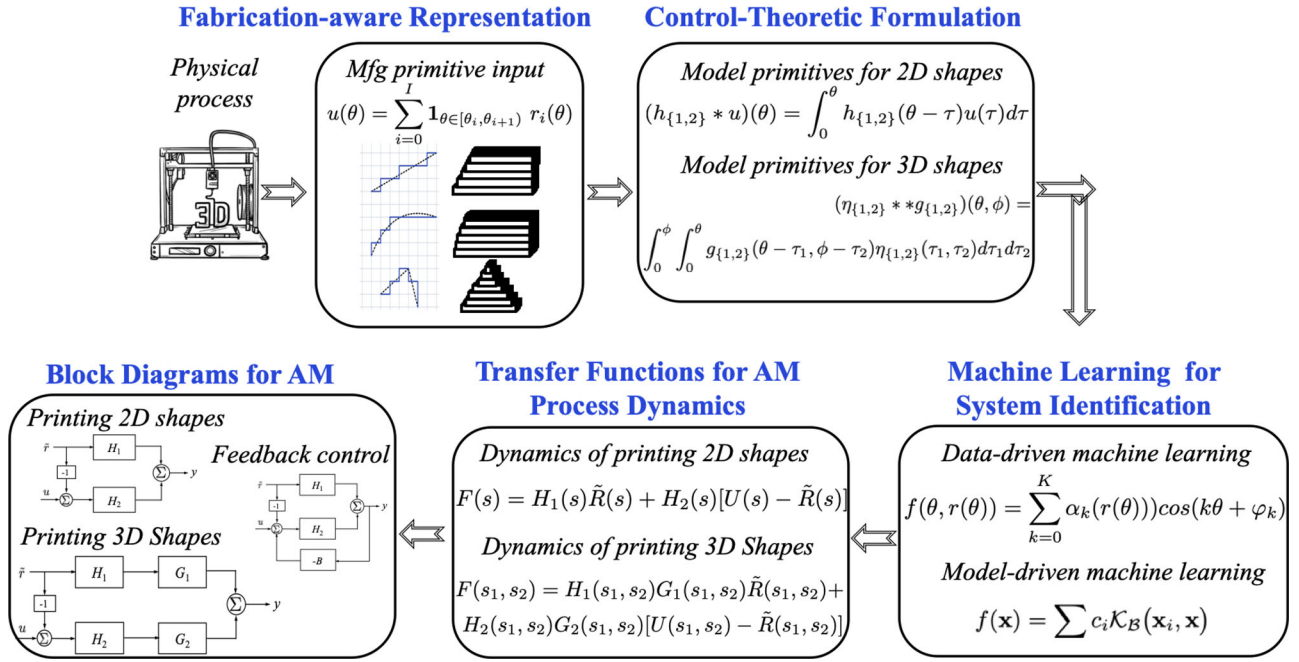


Figure 1. The control-theoretic foundation for computational AM.

The function f is a curve for a 2D shape or a surface for a 3D shape. A product u to be predicted belongs to either source data D_S or a validation set D_V , i.e., $u \in D_S \cup D_V$ and $D_S \cap D_V = \emptyset$.

- **Control Problem (CP):** The control objective is to find input adjustment δ_u for design $u \in D_S \cup D_V$ so that the modified design input $u + \delta(u)$ is expected to minimize the shape deformation $y_{u+\delta(u)}$ in reference to the intended design u .
- **Generalization Problem (GP):** The generalization objective is to discover the structures of $f(u)$ and $\delta(u)$ and physical underpinnings so that the generation of $f(u)$ and $\delta(u)$, that is, solving LP and CP problems for new product designs under new AM processes can be guided by principles.

2.2. Sub-problem definition with AM-specific constraints

The descriptions for LP, CP, and GP problems are relatively generic. As discussed in the Introduction, QC for AM faces issues of shape complexity, process complexity, data heterogeneity, and small data. To tackle LP, CP, and GP problems, we further define a set of sub-problems to accommodate AM-specific constraints:

- **Input and output representation (LP.1):** A proper representation of (u, y_u) should enable small-sample learning. Training data D_S contains n products (e.g., $n < 10$) in different shapes and sizes. In general, the 3D shape u of AM built products has infinite variety and the shape space is a manifold without natural linear structure. The output y_u is a deviation of a product shape, not the shape itself. The pattern of y_u depends both on u and specific AM processes.

- **Model $f(u)$ generation and “1-to- ∞ ” learning (LP.2):** Model $f(u)$ is expected to predict product quality of infinite varieties by learning from small data D_S . This “1-to- ∞ ” learning problem prefers $f(u)$ to be constructed with a limited number of building blocks, primitives or basis functions for robustness and flexibility. Model interpretability is another key consideration to ensure better understanding of physical systems.
- **Optimal design compensation and optimality definition (CP.1):** Optimal criteria for minimizing the shape deformation have to be defined with consideration of tolerance design and verification in AM. Optimal design compensation is to derive δ_u^* such that $y_{u+\delta_u^*}$ is minimized in reference to the nominal design u for a given criterion.
- **Transfer function and system dynamics identification through learning (GP.1):** Deriving the transfer function from $f(u)$ will provide a compact description of AM processes and facilitate the understanding of physical process insights. Note that obtaining transfer functions independently for individual products will only result in a projected view of transfer functions in subspaces.

In the next section, we present a theoretical framework to address some critical issues related to these problems.

3. Impulse response formulation and learning for geometric quality prediction and control in AM

To enable model-driven learning and control for AM, we propose a control-theoretic framework which consists of: (i) fabrication-aware input and output representation; (ii) impulse response formulation and modeling of AM; (iii) ML for impulse response estimation; (iv) transfer function characterization of AM process dynamics; and (v) block

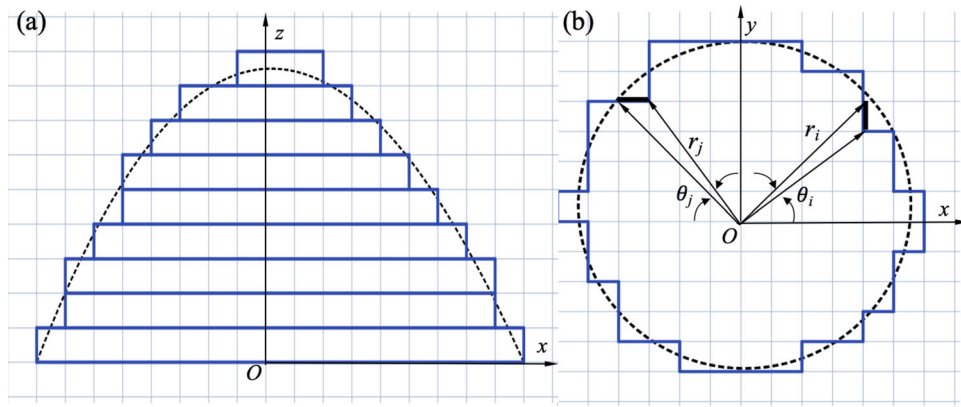


Figure 2. Primitive inputs to fabrication: (a) discretized layers through slicing 3D shapes along the z -direction; and (b) piecewise linear approximation of layer boundaries in the $x - y$ build plane.

diagram algebra for principled design and analysis of AM. The schematic diagram of this new control-theoretic foundation is illustrated in Figure 1.

3.1. Input and output representation (LP.1)

3.1.1. Review of shape and shape quality representation methods

Representation of design input u involves mathematical description of 3D shapes. Depending on applications, various methods have been developed to describe 3D objects, for example, parametric surfaces, solids, or constructive solid geometry representations (Bloomenthal, 1988; Snyder, 1992; Alliez *et al.*, 2005; Bermano *et al.*, 2017; Achlioptas *et al.*, 2018) finite element meshes (Ho-Le, 1988; Mori *et al.*, 1996; Pal *et al.*, 2014; King *et al.*, 2015) for multi-physics analysis, slices in STL format and landmarks (del Castillo and Colosimo, 2011; Alshraideh and Del Castillo, 2013; Dryden and Mardia, 2016; Khanzadeh *et al.*, 2018), point clouds (Besl and McKay, 1992; Tam *et al.*, 2013; Huang *et al.*, 2015; Yang *et al.*, 2016; Samie Tootooni *et al.*, 2017; Wang *et al.*, 2017; Xu *et al.*, 2017; Huang *et al.*, 2020; McConaha and Anand, 2020), or meshes (Besl and McKay, 1992; Tam *et al.*, 2013; Decker *et al.*, 2021) for shape registration, inspection, and distortion control.

Representation of shape quality y_u involves the description and measure of the deviations between a build product and its nominal design. Generally there are two strategies to represent output quality y_u :

- *Extracting characteristic features or descriptors* from a shape and measuring their deviations: The common features or descriptors include geometric and dimensional measures related to angular or distance metrics (e.g., parallelism, Euclidean, Hausdorff or geodesic distances) (Ameta *et al.*, 2015); differential measures such as curvatures and surface roughness (Savio *et al.*, 2007); and integral descriptors such as areas and volumes (e.g., volume shrinkage factor (Hilton and Jacobs, 2000).
- *Describing full shape deviations/deformation* for complex freeform surfaces: A finite number of descriptors is mathematically inadequate to represent complex shapes

obtained by topological optimization in AM. Complete description of local surface deviations everywhere along product boundaries not only provides a comprehensive representation solution, but also enables full control access to any region on a product surface (Huang *et al.*, 2015; Huang, 2016; Huang *et al.*, 2020). Naturally this study adopts this strategy to represent complete shape deviations. Note that this strategy also facilitates comprehensive feature extraction and evaluation afterwards.

Though each representation method is powerful in its own right, little work has concurrently considered the proper representation of design input u and quality output y_u for the purpose of small-sample learning, prediction, and control. Some limited attempts have been made towards this direction (Huang *et al.*, 2014; Huang *et al.*, 2015; Jin *et al.*, 2016; Luan and Huang, 2017; Wang *et al.*, 2017; Cheng *et al.*, 2018; Sabbaghi *et al.*, 2018; Sabbaghi and Huang, 2018; Jin *et al.*, 2019; de Souza Borges Ferreira *et al.*, 2020; Huang *et al.*, 2020). Using input u , materials and process information, finite element representation and modeling in theory is able to predict quality y_u with small sample data for parameter tuning and model validation. However, it is computationally costly (Bugeda Miguel Cervera and Lombra, 1999; King *et al.*, 2015) and it has been difficult to generalize the knowledge to new part geometries without extensive rounds of new simulations or new test builds.

To meet the goals defined in Section 2, we present input and output representation methods in the following two sub-sections.

3.1.2. Fabrication-aware input representation based on constructive shape primitives

To enable engineering-informed small-sample learning, we propose an input representation method based on the shape primitive concept, and establish a fabrication-aware formulation. Essentially, the small-sample learning of printed products is transformed into a large-sample learning of printed shape primitives.

Geometric (shape) primitives have been developed in computer graphics and CAD systems to construct 3D-shaped objects (Bermano *et al.*, 2017). Geometric

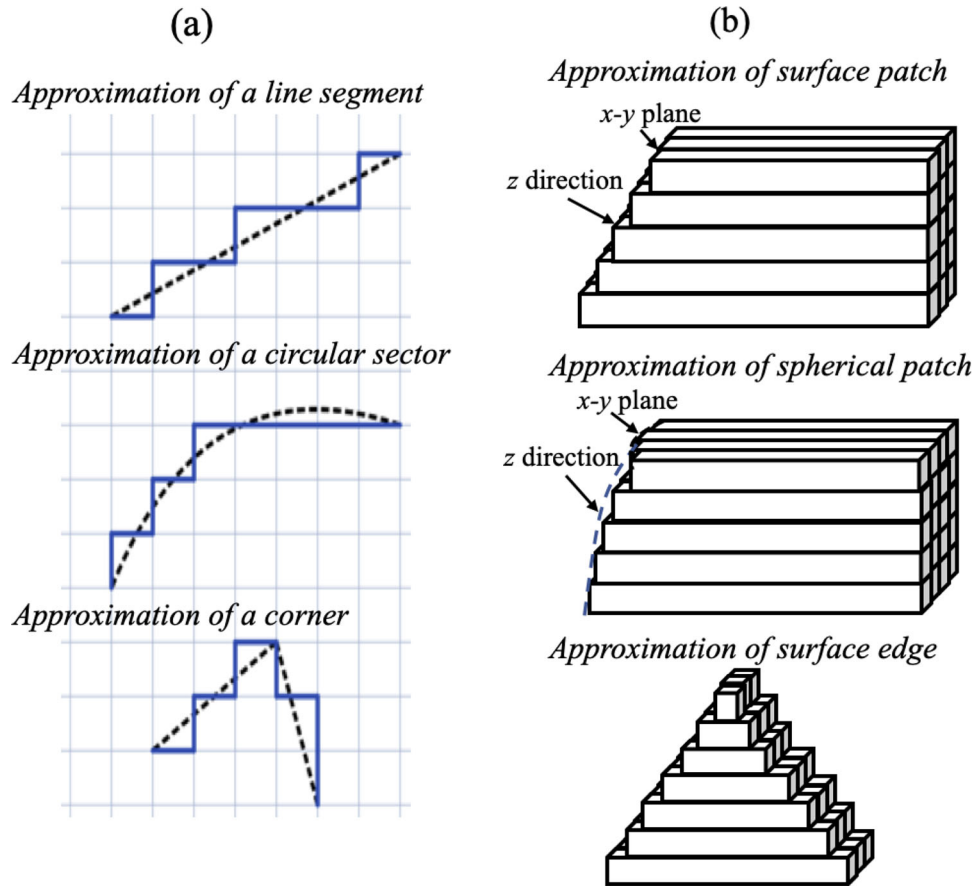


Figure 3. Geometric shape primitives: (a) 2D shape primitives for layer boundary approximation; and (b) 3D shape primitives for surface approximation.

primitives are simple shaped such as sphere, cube, cylinder, or surface patches. In 2D computer graphics, primitives include segments of straight lines, circles and more complicated curves. The true input u to the AM processes, however, is not the smooth shape defined analytically, for example, a dome in dash line illustrated in Figure 2(a), but rather discretized or sliced layers (Figure 2(a)) and piecewise linear approximation of layer boundaries (Figure 2(b)). Accuracy of the approximation is not only determined by computational modeling, but also by materials and processes. For example, resolution along the build direction (z) is determined by the layer thickness in Figure 2(a), whereas the resolution in the build plane ($x - y$ plane in Figure 2(b)) or the minimum feature size is limited by materials properties and technologies such as the laser spot size or the pixel size of a digital light projector.

The input representation therefore entails the *representation of individual layers and layer stackup*. Borrowing geometric primitive concepts from computer graphics, we propose to use line segments, circular sectors, and corners as primitive shapes to construct individual layer boundaries or 2D shapes (Figure 3(a)). Notice that the three 2D shape primitives have curvatures of zero, constant, and infinity, respectively. Comparing to line segments on the grids in Figure 2(b), the choice of primitives will reduce computational load of approximation and assist in the interpretation of the deviation patterns. We therefore state the following accepted truth as the basis for the subsequent derivation:

Definition 1 (Primitive manufacturing input). A 3D object built in AM is based on primitive manufacturing inputs, that is, sliced layers to be stacked up and individual layers with boundaries composed by 2D shape primitives.

Essentially, the design input to AM machines is the sliced and discretized model (e.g., in STL format defined by the unit normal and vertices), as opposed to smooth CAD models. Also note that even though the sliced layers can be geometrically stacked up in the design input, the final print outcome is nonlinear, due to physical layer interactions and accumulation. Considering layer stackup in input/output representation is therefore critical to capture underlying physical mechanisms.

To conveniently represent shape deformation (Huang *et al.*, 2015; Huang, 2016), we represent layers or 2D shapes in the Polar Coordinate System (PCS) and 3D shapes in the Spherical Coordinate System (SCS). The layer boundary is represented as piecewise segments through a circular approximation with selective cornering strategy developed in Luan and Huang (2017):

$$u(\theta) = \sum_{i=0}^I \mathbf{1}_{\theta \in [\theta_i, \theta_{i+1})} r_i(\theta) \quad (1)$$

where in the interval $[\theta_i, \theta_{i+1})$, $r_i(\theta)$ is either a line segment, circular sector, or a corner segment. Each segment consists of a set of points describing the primitive.

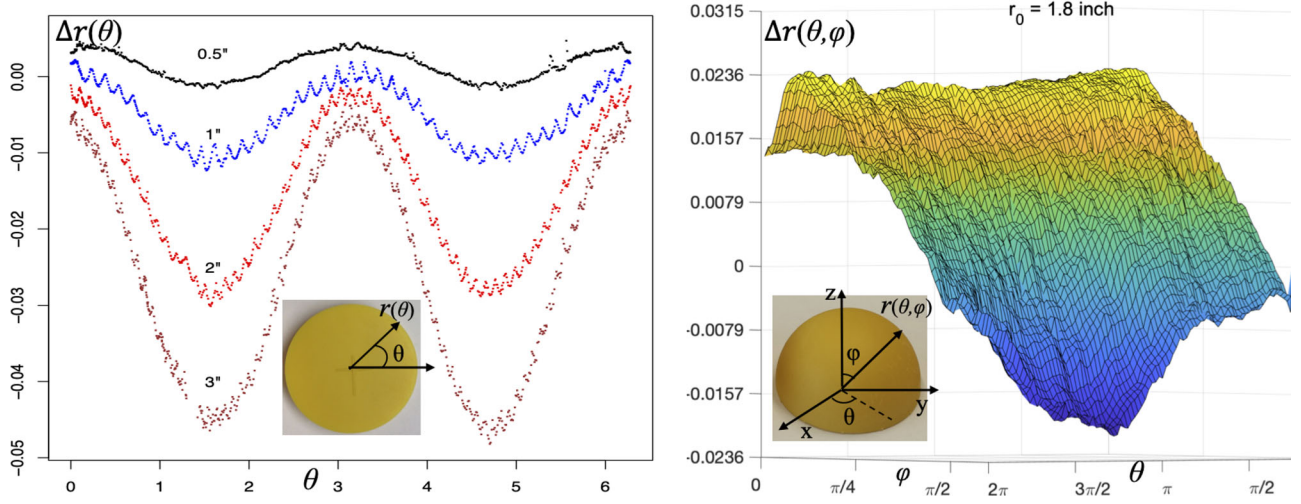


Figure 4. Deviation profiles of disks (Huang et al., 2015) (left) and deviation surface of a dome (Huang et al., 2020) (right).

When considering a layer indexed by j with height z_j , input u is represented as

$$u(\theta, \phi) = \sum_{j=0}^J \sum_{i=0}^I \mathbf{1}_{\theta \in [\theta_i, \theta_{i+1})} \mathbf{1}_{\{\phi: r(\theta, \phi) \cos \phi = z_j\}} r_i(\theta, \phi) \sin \phi \quad (2)$$

Purely from the geometric point of view, it is more convenient to adopt plane surface patches, spherical patches, and surface edges as 3D shape primitives to represent and construct 3D shapes. However, it is still an open issue to segment a 3D shape into 3D shape primitives for AM (Wang et al., 2022). In particular, prediction of the shape deformation in AM relies on a description of how products are fabricated. Since 3D shape primitives are constituents of a 3D shape, they are the results of primitive manufacturing inputs. We hence define 3D shape primitives for AM with the schematic plot shown in Figure 3(b).

Definition 2 (3D shape primitive). *The 3D shape primitives, that is, plane surface patches, spherical patches, and surface edge, are the result of layer stackup with layer boundaries composed by 2D shape primitives.*

In essence the definitions of primitive manufacturing input and 3D shape primitive intends to establish a fabrication-aware formulation that enables understanding of the deformation of 3D shape primitives. As such, deformation prediction of a freeform shape product can be achieved by predicting the deformation of the shape constructed by 3D shape primitives. A freeform 3D shape can be constructed as:

$$u(\theta, \phi) = \sum_{j=0}^J \sum_{i=0}^I \mathbf{1}_{\{\theta \in [\theta_i, \theta_{i+1}), \phi \in [\phi_j, \phi_{j+1})\}} r_{ij}(\theta, \phi) \quad (3)$$

where $r_{ij}(\theta, \phi)$ is either a plane surface patch, spherical patch, or a surface edges.

3.1.3. Output representation based on transformation of point cloud data

By representing design shapes in the SCS (Huang et al., 2015; Huang, 2016), the output y_u is the difference between

the actual printed product $r(\theta, \phi)$ and its intended design $r^0(\theta, \phi)$:

$$y_u = \Delta r(\theta, \phi) = r(\theta, \phi) - r^0(\theta, \phi) \quad (4)$$

And for individual layers or 2D shapes, y_u is simply $\Delta r(\theta)$. Also note that quality y_u is benchmarked by the nominal design, not the input $u(\theta, \phi)$ to the fabrication process, though $u(\theta, \phi)$ is a close approximation of design $r^0(\theta, \phi)$.

Representation of y_u in the SCS or the PCS intends to decouple the shape complexity from deviation modeling and to identify systematic shape deformation patterns. Figure 4 illustrate the deviation curves of four flat disks with different sizes (left) (Huang et al., 2015) and a deviation surface of a dome shape (right) (Huang et al., 2020). Figure 5 shows deviation curves of three square plates, two pentagon plates with different sizes (top panel) (Huang et al., 2014), and four thin walls in half-disk shapes. All products were printed in a stereolithography process (SLA) (similar patterns have been observed in other AM processes (Wang et al., 2017; Cheng et al., 2018; Luan et al., 2019)).

Remark. The proposed input-output representation approaches reveal the following information:

- Comparing with the scanned data in point cloud format, the output representation in the SCS and PCS uncovers deformation patterns for both 2D and 3D shapes.
- Deformation patterns vary with shapes and sizes (pls refer to additional example in (Jin et al., 2016, 2020)). Learning deformation patterns based on shapes is not suitable and efficient for small-sample learning.
- Being as constructive as shape primitives to shapes, the deformation of the proposed three shape primitives are distinctive, and can be utilized to learn and construct the deformation patterns of many different shapes with small training data.
- The deformation pattern of a 3D shape is not a simple linear summation or extension of the deformation patterns of its constructive layers. For example, the pattern of the dome in Figure 4 is not composed by simply

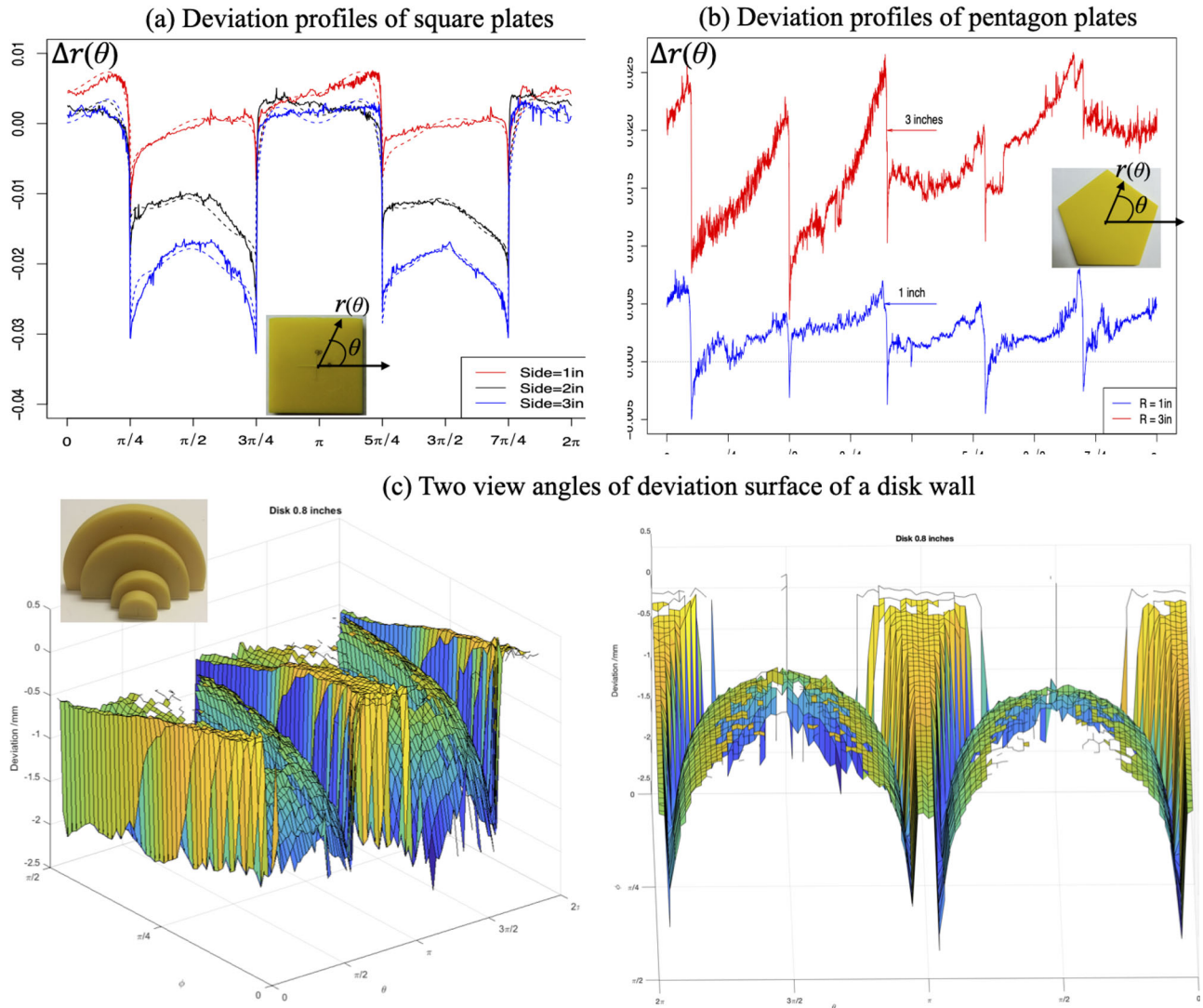


Figure 5. Deviation profiles of polygon plates (a)(b) (Huang *et al.*, 2014) and (c) thin walls in half-disk shapes.

stacking up the patterns of disks along the z-direction. But the deviation profile of a 3D shape at a given height ϕ (i.e., a horizontal section) partially resembles the one of its 2D counterpart. It is particularly remarkable to see in Figure 5 that the horizontal section of the thin walls in half-disk shapes has a rectangular shape. At a given ϕ (Figure 5(c), right panel), the deformation pattern resembles the ones for square plates to a large degree (Figure 5(a)).

- The connection between 2D and 3D shape deformation patterns implies that deformation models for 2D shapes should be special cases of those for 3D shapes. The 2D deviation models can be viewed as a projection of its 3D counterpart onto a subspace. The physical underpinning of projection relation is the nonlinear stackup of 2D layers into 3D shapes.

For complicated geometries, shape segmentation may be necessary to accurately represent shape deviations with deviation primitives. Shape segmentation has long been applied in 3D object recognition (Besl and Jain, 1985; Besl and Jain, 1988), reverse engineering (Várady *et al.*, 1997), and free-form surface metrology (Savio *et al.*, 2007). It involves the

process of dividing the original point set into subsets corresponding to natural surfaces. Segmentation for the purpose of deformation identification and modeling is worthy of further investigation. Little research has been reported.

3.2. Model $f(u)$ generation and “1-to- ∞ ” learning (LP.2) – an impulse response formulation and modeling approach

The proposed input–output representation enables the modeling of a *model primitive*, that is, to construct $f(u)$ with a small set of basis functions that characterize deformation patterns of shape primitives. Small-sample “1-to- ∞ ” learning therefore becomes feasible.

Definition 3 (2D model primitive). A 2D model primitive is defined as a functional model that predicts the deviation of a 2D shape primitive, which is either a circular sector, a line segment, or a corner.

Remark. Since a line segment can be viewed as a special case of a corner with one edge being zero, we only need two

types of 2D model primitives for circular sectors and corners, denoted as function $\eta_1(\cdot)$ and $\eta_2(\cdot)$, respectively.

Definition 4 (3D model primitive). A 3D model primitive is defined as a functional model that predicts the deviation of a 3D shape primitive, which is either spherical patch or a surface edge with the plane surface patches being a special case of the surface edge.

2D model primitives are connected with 3D model primitives through the definition of 3D shape primitives. The strategy to establish 3D model primitives is to model primitive manufacturing inputs, that is, 2D shape primitives that form the boundaries of layers and layer stackup. Therefore, 2D model primitives have to be established first.

3.2.1. Impulse response modeling for 2D model primitives

Our previous work (Huang *et al.*, 2014; Luan and Huang, 2017) innovated the shape deviation primitive concept for data-driven modeling of 2D shape deviation. The input representation, particularly, primitive inputs to the fabrication process shown in Figure 2 inspires the impulse response modeling developed in this work.

Theorem 1 (Impulse response characterization of 2D model primitives). The 2D model primitive $\eta_{\{1,2\}}(\theta)$ for a 2D primitive shape $u(\theta)$ can be characterized by an impulse response function $h_{\{1,2\}}(\theta)$ through a convolution formulation:

$$\eta_{\{1,2\}}(\theta) = (h_{\{1,2\}} * u)(\theta) = \int_0^\theta h_{\{1,2\}}(\theta - \tau)u(\tau)d\tau \quad (5)$$

Proof. As shown in Figure 3(a), a 2D shape primitive is a piecewise constant input, which can be represented as a sum of 1D step signals represented in the PCS with θ being equivalent to the time variable t . Once this critical connection is established, the proof follows an excellent control reference along Figure 6 [in chapter 6 of Åström and Murray (2021)].

Let the input $u(t)$ in Figure 6 represent a primitive angular input with $t = \theta$. The model primitive $\eta_{\{1,2\}}(\theta)$ represents the output or the deviation of the primitive shape. Let $H_{\{1,2\}}(\theta)$ be the response to a unit step applied at $\theta=0$ and assume $H_{\{1,2\}}(0) = 0$. The responses to a series of step inputs are $H_{\{1,2\}}(\theta - \theta_0)u(\theta_0)$, $H_{\{1,2\}}(\theta - \theta_1)(u(\theta_1) - u(\theta_0))$, and so on. The output $\eta_{\{1,2\}}(\theta)$ is the sum of individual responses (Åström and Murray, 2021):

$$\begin{aligned} \eta_{\{1,2\}}(\theta) &= H_{\{1,2\}}(\theta - \theta_0)u(\theta_0) + H_{\{1,2\}}(\theta - \theta_1)(u(\theta_1) - u(\theta_0)) + \dots \\ &= (H_{\{1,2\}}(\theta - \theta_0) - H_{\{1,2\}}(\theta - \theta_1))u(\theta_0) + (H_{\{1,2\}}(\theta - \theta_1) - H_{\{1,2\}}(\theta - \theta_2))u(\theta_1) + \dots \\ &= \lim_{n \rightarrow \infty} \sum_{k=1}^n (H_{\{1,2\}}(\theta - \theta_{k-1}) - H_{\{1,2\}}(\theta - \theta_k))u(\theta_{k-1}) + H_{\{1,2\}}(\theta - \theta_n)u(\theta_n) \\ &= \lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{H_{\{1,2\}}(\theta - \theta_{k-1}) - H_{\{1,2\}}(\theta - \theta_k)}{\theta_k - \theta_{k-1}} u(\theta_{k-1}) * (\theta_k - \theta_{k-1}) + H_{\{1,2\}}(\theta - \theta_n)u(\theta_n) \\ &= \int_0^\theta H_{\{1,2\}}'(\theta - \tau)u(\tau)d\tau = \int_0^\theta h_{\{1,2\}}(\theta - \tau)u(\tau)d\tau = (h_{\{1,2\}} * u)(\theta) \end{aligned}$$

where $h_{\{1,2\}}(\theta) = \frac{dH_{\{1,2\}}(\theta)}{d\theta}$, the derivative of the step response, is commonly known as the Impulse Response Function (IRF) in control theory. $\square$

Based on the definition of 2D model primitives, we have IRFs $h_1(\theta)$ and $h_2(\theta)$ corresponding to $\eta_1(\theta)$ and $\eta_2(\theta)$ for circular sectors and corners, respectively.

3.2.2. Impulse response modeling for 3D model primitives

We first postulated a convolution formulation of 3D shape generation in AM without proof (Huang *et al.*, 2020). Based on the definition of the 3D shape primitive and Theorem 1, this work provides a theoretical justification of the shape deviation generator model established in Huang *et al.*, (2020).

Theorem 2 (Impulse response characterization of 3D model primitives). The 3D model primitive for a 3D primitive shape $u(\theta, \phi)$ can be characterized by a 2D impulse response function $g(\theta, \phi)$ through a 2D convolution formulation:

$$(\eta_{\{1,2\}} * g_{\{1,2\}})(\theta, \phi) = \int_0^\phi \int_0^\theta g_{\{1,2\}}(\theta - \tau_1, \phi - \tau_2)\eta_{\{1,2\}}(\tau_1, \tau_2)d\tau_1d\tau_2 \quad (6)$$

Proof. As shown in Figure 3(b) and Figure 2(a), a 3D shape primitive can be viewed as an extension of a 1D piecewise constant input to the 2D case. In addition to the time variable $\tau_1 = \theta$ in the $x - y$ plan for 2D shape primitives, there is an additional time variable z or equivalently $\tau_2 = \phi$ along the z -direction to stack up 2D shape primitives at time τ_2 . Since a 3D shape primitive is the result or output of stacking up 2D shape primitives, the true step input to a 3D primitive at time τ_2 is therefore $\eta_{\{1,2\}}(\theta, \tau_2)$, that is, the deviation of a 2D shape primitive (here we remove the nominal input of 2D shape primitives for simplification of notation).

Let $G_{\{1,2\}}(\theta, \tau_2)$ be the response to the 2D unit step applied at $\tau_2 = 0$ and further assume $G_{\{1,2\}}(\theta, 0) = 0$. The responses to a series of 2D step inputs are $G_{\{1,2\}}(\theta - \theta_0, \phi - \phi_0)\eta_{\{1,2\}}(\theta_0, \phi_0)$, $G_{\{1,2\}}(\theta - \theta_1, \phi - \phi_1)(\eta_{\{1,2\}}(\theta_1, \phi_1) - \eta_{\{1,2\}}(\theta_0, \phi_0))$, and so on. Following the same procedure, but working in the 2D time space τ_1, τ_2 , we will have the output or the 3D model primitive as

$$\int_0^\phi \int_0^\theta \frac{\partial^2 G_{\{1,2\}}}{\partial \theta \partial \phi}(\theta - \tau_1, \phi - \tau_2)\eta_{\{1,2\}}(\tau_1, \tau_2)d\tau_1d\tau_2$$

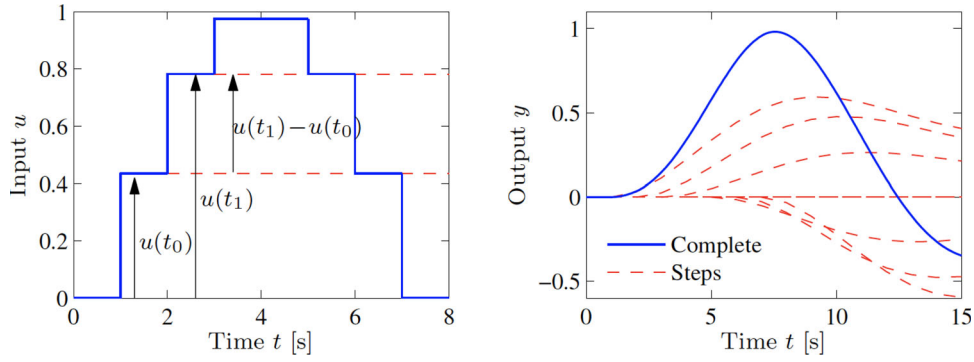


Figure 6. Response to piecewise constant input (a) piecewise constant input as a summation of step inputs and (b) output as the sum of individual output (Åström and Murray, 2021), [copyright] Princeton University Press.

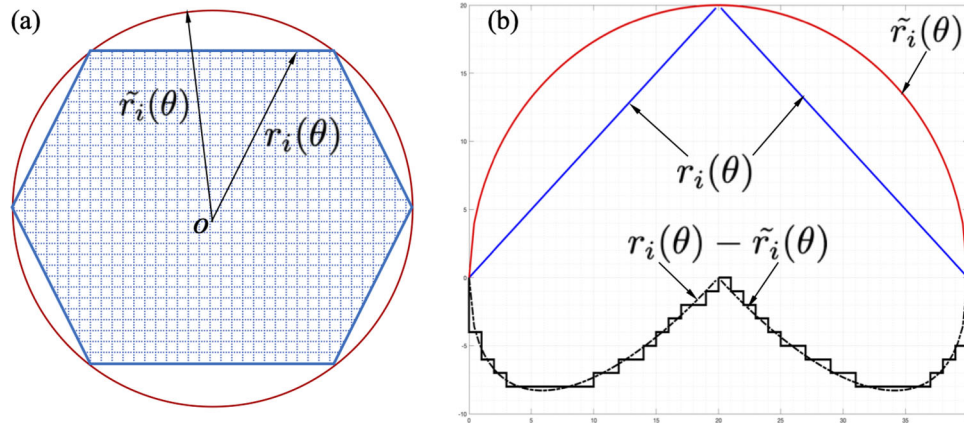


Figure 7. Simplification of deformation model (7) for 2D freeform shapes.

And let $g_{\{1,2\}}(\theta, \phi) = \frac{\partial^2 G_{\{1,2\}}}{\partial \theta \partial \phi}$, the partial derivative of the 2D step response, or the 2D IRF. Denoting this 2D convolution integral as $(\eta_{\{1,2\}} * g_{\{1,2\}})(\theta, \phi)$, we have the proof for [Theorem 2](#). $\square$

Remark. As we pointed out in [Huang et al. \(2020\)](#), if we only print a single layer of the 3D primitive shape, the IRF $g_{\{1,2\}}(\theta, \phi)$ is degenerated to a Dirac's delta function $\delta(\theta, z \approx 0)$. Then

$$(\eta_{\{1,2\}} * \delta)(\theta, z \approx 0) = \eta_{\{1,2\}}(\theta, z \approx 0) = \eta_{\{1,2\}}(\theta)$$

which is the 2D model primitive in (5). Clearly, [Theorem 2](#) is an extension of [Theorem 1](#).

3.2.3. Model construction and “1-to- ∞ ” learning based on model primitives

Theorems 1 and 2 establish a fabrication-aware, control-theoretic formulation of model building blocks that can be learned from a small training sample D_S . Furthermore, the two theorems enable model construction to predict the quality of freeform shapes in theory. In this work we use simple examples to show the feasibility and potential of applying the two theorems. Certainly, significant efforts are needed to achieve ultimate “1-to- ∞ ” learning for complex 3D freeform shapes.

a. **“1-to- ∞ ” learning to predict deformation of 2D freeform shapes:** We have achieved data-driven “1-to- ∞ ”

learning to predict deformation of 2D freeform shapes ([Huang et al., 2014](#); [Huang et al., 2015](#); [Luan and Huang, 2017](#); [Wang et al., 2017](#); [de Souza Borges Ferreira et al., 2020](#)). By taking a fresh look under the control-theoretic formulation, we intend to discover not only the process dynamics, but also principles for generalization.

Corollary 2.1 (Deformation model for 2D freeform shapes). *Based on [Theorem 1](#), (5), and (1), the deformation of a freeform 2D shape or layer can be derived as $y(u(\theta)) = f(u(\theta)) + \epsilon$ with $f(u(\theta))$:*

$$\begin{aligned} f(u(\theta)) &= \sum_{i=0}^I \mathbf{1}_{\theta \in [\theta_i, \theta_{i+1})} [\mathbf{1}_c(r_i(\theta))(h_1 * r_i)(\theta) + (1 - \mathbf{1}_c(r_i(\theta)))(h_2 * r_i)(\theta)] \\ &= \sum_{i=0}^I [\mathbf{1}_c(r_i(\theta))\eta_1(\theta, r_i(\theta)) + (1 - \mathbf{1}_c(r_i(\theta)))\eta_2(\theta, r_i(\theta))] \end{aligned} \quad (7)$$

where $\mathbf{1}_c(r_i(\theta)) = 1$ if $r_i(\theta)$ is a circular sector and zero otherwise. Notation $\eta_{\{1,2\}}(\theta, r_i(\theta))$ specifies that the model primitive $\eta_{\{1,2\}}(\theta)$ has shape primitive $r_i(\theta)$ as input. The error term ϵ can be noise or can impose a correlation structure to capture the interaction among neighboring shape primitives, depending on AM processes.

However, modeling and computation can be complicated and inefficient when the number of shape primitives I is large. One improvement of model (7) is provided.

Corollary 2.2 (Cookie-cutter modeling for 2D freeform shapes). The deformation model $f(u(\theta))$ for 2D freeform shapes in (7) can be well approximated as

$$f(u(\theta)) = \sum_{i=0}^I \eta_1(\theta, \tilde{r}_i(\theta)) + \sum_{i=0}^I \eta_2(\theta, r_i(\theta) - \tilde{r}_i(\theta)) \quad (8)$$

where $\tilde{r}_i(\theta)$ is the smallest circular sector that covers a line segment or a corner (Figure 7(a)).

Proof. We can find a smallest circular sector $\tilde{r}_i(\theta)$ to approximate each shape primitive $r_i(\theta)$. Apparently, $r_i(\theta) - \tilde{r}_i(\theta) = 0$ if $r_i(\theta)$ is a circular sector. Then (7) can be rewritten as:

$$f(u(\theta)) = \sum_{i=0}^I \eta_1(\theta, \tilde{r}_i(\theta)) + (1 - \mathbf{1}_c(r_i(\theta))) \sum_{i=0}^I [\eta_2(\theta, r_i(\theta)) - \eta_1(\theta, \tilde{r}_i(\theta))]$$

When $r_i(\theta) - \tilde{r}_i(\theta) \neq 0$, as shown in Figure 7(b), the shape of $r_i(\theta) - \tilde{r}_i(\theta)$ can still be approximated by a corner or a line segment with a higher-order approximation error. Therefore, $\eta_2(\theta, r_i(\theta)) - \eta_1(\theta, \tilde{r}_i(\theta)) \approx \eta_2(\theta, r_i(\theta) - \tilde{r}_i(\theta))$. When $r_i(\theta) - \tilde{r}_i(\theta) = 0$, clearly $\eta_2(\theta, r_i(\theta) - \tilde{r}_i(\theta)) = 0$, so we can drop the indicator $(1 - \mathbf{1}_c(r_i(\theta)))$ and obtain (8). $\square$

Remark. This revised model (8) essentially treats a 2D shape as being carved out from a circular disk or segments of circular disks using cookie-cutters. In Huang *et al.* (2014) we empirically established the so-called “cookie-cutter” modeling framework. Corollary 2.2 provides a theoretical justification and opportunity to further enhance the model identification.

Learning 2D model primitives can now be achieved by a small training sample consisting of disks and polygon shapes with different sizes. From our study of different AM processes (Huang *et al.*, 2014; Huang *et al.*, 2015; Wang *et al.*, 2017; Cheng *et al.*, 2018; Luan *et al.*, 2019), $\eta_1(\theta, \tilde{r}_i(\theta))$ for a circular sector can be well approximated by low-order Fourier bases:

$$\eta_1(\theta, \tilde{r}_i(\theta)) = \sum_{k=0}^K \alpha_k(\eta_1(\theta, \tilde{r}_i(\theta))) \cos(k\theta + \varphi_k) \quad (9)$$

For example, four disks (Figure 4, left panel) printed in a SLA process can be well modeled by $\eta_1(\theta, r) = -0.0134(r + 0.0088)^{0.86} + 0.0057(r + 0.0088)^{1.13} \cos(2\theta)$ with r being the disk radius (Huang *et al.*, 2015). This data-driven model provides a complete description of circular sectors at different location θ .

Candidate choice of $\eta_2(\theta, r_i(\theta) - \tilde{r}_i(\theta))$ is proposed in Huang *et al.* (2014), for example, for three square plates (Figure 5(a)), which can be a square wave function:

$$\eta_2(\theta, \tilde{r}(\theta)) = \beta(\tilde{r}(\theta)) \text{sign}[\cos(n(\theta - \phi_0)/2)]$$

or a sawtooth wave function:

$$\eta_2(\theta, \tilde{r}(\theta)) = \beta(\tilde{r}(\theta)) \{ \text{sign}[\sin(n(\theta - \phi_0)/2)] + 1 \} * [(\theta - \phi_0) \bmod (2\pi/n)]$$

where n represents the number of polygon sides, the MOD function obtains remainders, and ϕ_0 is the phase variable.

Apparently if a segment of a 2D shape boundary can be approximated by a circular sector, or a corner of an n -sided polygon (including a line segment), the corresponding model primitive can be utilized for prediction. This deviation primitive concept was first postulated in Luan and Huang (2017) through a unified data-driven model to predict deformation of 2D freeform shapes. Now model (8) essentially provides a theoretical justification and model-driven formulation.

a. **“1-to- ∞ ” learning to predict deformation of 3D shapes:** “1-to- ∞ ” learning for 3D freeform shapes remains an open issue, due to the lack of fabrication-aware modeling and learning theories (Huang *et al.*, 2020). Theorem 2 has shown that the deviation of a spherical patch on a 3D shape can be modeled as $(\eta_1 * g_1)(\theta, \phi)$ and the deviation of a surface edge (including the plane surface patch) can be modeled as $(\eta_2 * g_2)(\theta, \phi)$. With a 3D shape defined in (3), the deformation of a freeform 3D shape is provided by the following corollary.

Corollary 2.3 (Cookie-cutter modeling for 3D freeform shapes).

$$y(u(\theta, \phi)) = \sum_{j=0}^J \sum_{i=0}^I \mathbf{1}_{\{\theta \in [\theta_i, \theta_{i+1}), \phi \in [\phi_j, \phi_{j+1})\}} [\mathbf{1}_s(r_{ij}(\theta, \phi))(\eta_1 * g_1)(\theta, \phi, r_{ij}(\theta, \phi)) + (1 - \mathbf{1}_s(r_{ij}(\theta, \phi)))(\eta_2 * g_2)(\theta, \phi, r_{ij}(\theta, \phi))] + \psi(\theta, \phi, r_{ij}(\theta, \phi)) + \epsilon \quad (10)$$

where $\mathbf{1}_s(r_{ij}(\theta, \phi)) = 1$ if $r_{ij}(\theta, \phi)$ is a spherical patch, and zero otherwise. $\psi(\theta, \phi, r_{ij}(\theta, \phi))$ models the spatial correlations among shape primitives, and ϵ is the noise term.

If we denote $y(u(\theta, \phi)) = f(u(\theta, \phi)) + \psi(\theta, \phi, r_{ij}(\theta, \phi)) + \epsilon$. Model $f(u(\theta, \phi))$ in (10) can be further improved through approximation as

$$f(u(\theta, \phi)) = \sum_{j=0}^J \sum_{i=0}^I (\eta_1 * g_1)(\theta, \phi, \tilde{r}_{ij}(\theta, \phi)) + \sum_{j=0}^J \sum_{i=0}^I (\eta_2 * g_2)(\theta, \phi, r_{ij}(\theta, \phi) - \tilde{r}_{ij}(\theta, \phi)) \quad (11)$$

where $\tilde{r}_i(\theta, \phi)$ is the smallest spherical patch that covers a plane surface patch or a surface edge.

The proof of this corollary is omitted because it is extension from the 2D case presented in Corollary 2.1 and 2.2.

In Huang *et al.* (2020) we studied the dome shapes (Figure 4, right panel) and first proposed a convolution formulation of AM processes. The model takes the form of:

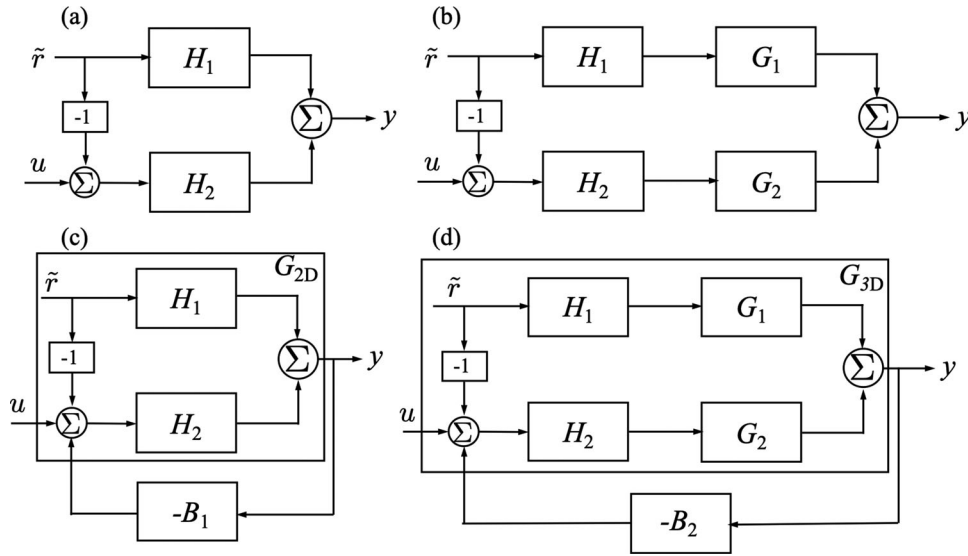


Figure 8. Block diagrams of AM processes (a) printing 2D shapes; (b) printing 3D shapes; (c) feedback control for printing 2D shapes; and (d) feedback control for printing 3D shapes.

$$y(\theta, \phi, r) = \alpha(r) \int_0^{\pi/2} \int_0^{2\pi} g_1(\theta - \tau_1, \phi - \tau_2) * \eta_1(\tau_1, \tau_2) d\tau_1 d\tau_2 + \beta(r) + \psi(\theta, \phi, r) + \epsilon \quad (12)$$

where $\eta_1(\theta, \phi) = \cos(2\theta)\sin(\phi)$ and $g_1(\theta, \phi) = \cos(n_1\phi)[1 + \cos(n_2\theta + \phi_0)]$ are found suitable for the studied SLA process (pls refer to Huang *et al.* (2020) for detailed modeling and model estimation).

Apparently the model for dome shapes in (12) is a special case of the model in (11). It provides meaningful understanding of the deformation of spherical patches. Further modeling and “1-to- ∞ ” learning based on the generalized model (11) will make the quality prediction of 3D freeform shapes feasible with small training data and cost-effective experimentation.

Remark. Our initial data-driven “1-to- ∞ ” modeling and learning (Huang *et al.*, 2014; Huang *et al.*, 2015; Luan and Huang, 2017; Wang *et al.*, 2017; de Souza Borges Ferreira *et al.*, 2020; Huang *et al.*, 2020) is consistent with the theoretic framework established in this study. The control-theoretic framework enables the exciting and more powerful model-based ML for example, impulse response estimation through kernel methods in system identification and machine learning (Pillonetto *et al.*, 2014).

3.3. AM process representation and dynamics analysis using transfer functions and block diagrams (GP.1)

Transfer functions and block diagrams are powerful tools in control theory to design, learn, identify, and analyze system dynamics. Comparing with data-driven models (Huang *et al.*, 2014; Huang *et al.*, 2015; Luan and Huang, 2017; Wang *et al.*, 2017; de Souza Borges Ferreira *et al.*, 2020; Huang *et al.*, 2020), the control-theoretic framework established in the study allows the utilization of transfer functions to find new interpretation of AM processes.

3.3.1. Transfer function characterization of AM process dynamics

A Laplace transform maps function $f: \mathbb{R}^+ \rightarrow \mathbb{R}$ to function $F = \mathcal{L}_1[f]: \mathbb{C} \rightarrow \mathbb{C}$ of a complex variable s . The double Laplace transform of a 2D f is denoted as $\mathcal{L}_2[f]$. By the property of the Laplace transform of a convolution, i.e., $\mathcal{L}[f * g] = F(s)G(s)$, the prediction model in (8) for 2D shapes and prediction model in (11) for 3D shapes can be transformed as

$$F(s) = H_1(s)\tilde{R}(s) + H_2(s)[U(s) - \tilde{R}(s)] \quad (13)$$

$$F(s_1, s_2) = H_1(s_1, s_2)G_1(s_1, s_2)\tilde{R}(s_1, s_2) + H_2(s_1, s_2)G_2(s_1, s_2)[U(s_1, s_2) - \tilde{R}(s_1, s_2)] \quad (14)$$

where $\hat{Y}$ represents the prediction without considering the error terms.

Remark. Clearly, the input/output dynamics of the AM processes can be captured by transfer functions $H_1(s)$ and $H_2(s)$ for printing 2D shapes (plates), and by transfer functions $H_1(s_1, s_2)$, $H_2(s_1, s_2)$, $G_1(s_1, s_2)$, and $G_2(s_1, s_2)$ for printing 3D shapes. Identification of transfer functions for AM processes and systems is therefore a critical area that can enable principled design, control, and generalization of process knowledge.

There are generally two strategies to identify transfer functions. Data-driven system identification and model-based machine learning (Pillonetto *et al.*, 2014). There is no reported research for AM quality control in this regard.

Here we present an example of a data-driven method to identify transfer functions when printing 2D shapes. For the SLA process that printed the four disks (Figure 4, left panel) we obtained $\eta_1(\theta, r) = -0.0134(r + 0.0088)^{0.86} + 0.0057(r + 0.0088)^{1.13} \cos(2\theta)$ (Huang *et al.*, 2015). By the Laplace transform of (5), we have $\mathcal{L}_1[\eta_{\{1,2\}}] = H_{\{1,2\}}(s)U(s)$. As to the true input $u(\theta)$ to the SLA process of study, as we pointed out in Huang *et al.* (2015), it is not the nominal

design r , but rather $u(\theta) = (r + 0.0088)$ due to the effect of over-exposure.

Since the input to printing disks is r , then $H_1(s)$ can be obtained as

$$\begin{aligned} H_1(s) &= \frac{\mathcal{L}_1[\eta_1(\theta, r)]}{\mathcal{L}_1[r + 0.0088]} = -\frac{0.0134(r + 0.0088)^{0.86}}{r + 0.0088} \\ &\quad + \frac{0.0057(r + 0.0088)^{1.13}}{r + 0.0088} \frac{s^2}{s^2 + 4} \\ &= -0.0134(r + 0.0088)^{-0.14} + 0.0057(r + 0.0088)^{0.13} \frac{s^2}{s^2 + 4} \end{aligned}$$

Apparently, the radius/size of the disk has an impact on the process dynamics that causes the deformation of the disk. The transfer function also indicates that the SLA process needs improvement in two aspects: tuning the process to be independent of size (dimensionless), and to be close to zero gain.

Following the same procedure, we can also obtain $H_2(s)$ by Laplace transform of the square wave function or sawtooth function. For example, by expressing the square wave function in term of Heaviside's function, the $H_2(s)$ for the SLA process of study is

$$H_2(s) = \frac{\beta(\tilde{r})}{s} \tanh \frac{\pi}{n} s$$

and $H_2(s)$ is

$$H_2(s) = \beta(\tilde{r}) \left[\frac{n}{2\pi s^2} - \frac{1}{se^{\frac{2\pi}{n}s} - s} \right]$$

for the sawtooth wave function.

3.3.2. Block diagram representation of AM processes

The transfer function expression of AM processes in (13) and (14) can be conveniently represented in block diagrams. Figure 8(a) and (b) shows the block diagrams for AM processes to print 2D and 3D shapes, respectively. With the obtained transfer functions, the system dynamics can be analyzed and simulated with, e.g., Matlab Simulink.

Another important application of the block diagram algebra for AM is to represent, design, and analyze feedback control systems (Figure 8(c) and (d)). Since most AM machines adopt standard settings for different materials and designs, a common feedback control strategy is to change the design if errors occur in the manufacturing (Huang *et al.*, 2015; Huang, 2016), as opposed to changing process settings. Following the block diagram algebra for feedback control, we first define the equivalent transfer functions for 2D and 3D printing in Figure 8(a) and (b) as:

$$G_{2D} = [H_1 - H_2] \frac{\tilde{R}}{U} + H_2 \quad (15)$$

$$G_{3D} = [H_1 G_1 - H_2 G_2] \frac{\tilde{R}}{U} + H_2 G_2 \quad (16)$$

Let $\delta(s)$ represent the feedback adjustment. From the block diagram algebra, we know $\delta(s) = -Y(s)B_{\{1,2\}}(s)$. Our

previous work (Huang, 2016) has shown that

$$\delta^*(u) = -\frac{f(\theta, \phi, r(\theta, \phi))}{1 + \frac{df(\theta, \phi, r(\theta, \phi))}{dr(\theta, \phi)}} = -\frac{f(u)}{1 + f'(u)}$$

where $f(u)$ is given in either (8) or (11). (Examples of $\delta^*(u)$ can be found in Huang (2016).

Through the Laplace transform, we have $\delta^*(s) = \mathcal{L}[\delta^*(u)]$. For system robustness, we replace $Y(s)$ with $F(s)$. Then the transfer function for feedback adjustment is

$$B_{\{1,2\}}(s) = \mathcal{L} \left[\frac{f(u)}{1 + f'(u)} \right] / F(s) \quad (17)$$

With this design, the transfer function of the feedback control system is

$$G_{fb} = \frac{G_{\{2D, 3D\}}}{1 + G_{\{2D, 3D\}} B_{\{1,2\}}} \quad (18)$$

The predicted process response after feedback control is $\mathcal{L}^{-1}[G_{fb}u]$. System robustness and stability can be analyzed through powerful tools in control theory (e.g., Nyquist's criterion).

4. Summary and conclusion

By establishing a fabrication-aware, impulse response formulation and modeling framework, this work attempts to bridge exciting research at the intersection of ML, control theory, and 3D printing. Though data-driven ML models have achieved notable progress in AM, this establish theoretical foundation enables model-driven machine learning which provides not only guided model formulation, but also new perspectives and insights into process dynamics. The well-established control theory can thus be readily applied to AM and advance intelligent QC in AM.

This control-theoretic framework is built upon the primitives AM inputs, that is, sliced layers to be stacked up and individual layers with boundaries composed by 2D shape primitives. Two theorems are developed to characterize AM processes using impulse response functions. AM process models in convolution formulation are then derived based on the theorems. Transfer function theory and block diagram algebra are then applied to AM process representation, design, and analysis. Examples from SLA processes are given to demonstrate the proposed control-theoretic framework, including design and analysis of feedback control of AM processes.

There are many other critical categories of ML4AM problems. These open problems include, but not limited to verification problem (VP) that aims to to define, measure, characterize, evaluate, and verify the quality of 3D printed products; detection and diagnosis problem (DDP) with goals to detect process changes and identify root causes in AM; model adaptation and transfer learning (DTL) problem that aims to quickly generate new models for new AM processes or process conditions; and optimal design for system identification (ODSI) that aims to design an optimal set of training data set that maximize the system identification, etc.

Further research is imperative to advance the control-theoretic foundation and computational research for intelligent AM.

Funding

This work is supported by National Science Foundation with Grant# NSF CMMI-1901514. We appreciate all collaborators in our previous publications.

Notes on contributor

Dr. Qiang Huang is currently a professor at the Daniel J. Epstein Department of Industrial and Systems Engineering, University of Southern California (USC), Los Angeles. His research focuses on AI and Machine Learning for Manufacturing, in particular, Machine Learning for Additive Manufacturing (ML4AM). He was the holder of Gordon S. Marshall Early Career Chair in Engineering at USC from 2012 to 2016. He received IISE Fellow Award, ASME Fellow Award, NSF CAREER award, and 2021 IEEE CASE Best Conference Paper Award, 2013 IEEE Transactions on Automation Science and Engineering Best Paper Award, among others. He holds five patents on ML4AM. He is a Department Editor for *IISE Transactions* and an Associate Editor for *ASME Transactions*, *Journal of Manufacturing Science and Engineering*.

ORCID

Qiang Huang  <http://orcid.org/0000-0001-7826-4792>

References

- Achlioptas, P., Diamanti, O., Mitliagkas, I. and Guibas, L. (2018) Learning representations and generative models for 3D point clouds, in *Proceedings of the 35th International Conference on Machine Learning, Proceedings of Machine Learning Research*, **80**, 40–49.
- Alliez, P., Cohen-Steiner, D., Yvinec, M. and Desbrun, M. (2005) Variational tetrahedral meshing, in *ACM SIGGRAPH 2005 Papers*, pp. 617–625.
- Alshraideh, H. and Del Castillo, E. (2013) Statistical performance of tests for factor effects on the shape of objects with application in manufacturing. *IIE Transactions*, **45**(2), 121–131.
- Ameta, G., Lipman, R., Moylan, S. and Witherell, P. (2015) Investigating the role of geometric dimensioning and tolerancing in additive manufacturing. *Journal of Mechanical Design*, **137**(11), 111401.
- Åström, K.J. and Murray, R.M. (2021) *Feedback Systems: An Introduction for Scientists and Engineers* (second edition). Princeton University Press, Princeton, NJ.
- Bensoussan, A., Li, Y., Nguyen, D.P.C., Tran, M.-B., Yam, S.C.P. and Zhou, X. (2020) Machine learning and control theory. *arXiv: 2006.05604 [cs.LG]*.
- Bermano, A.H., Funkhouser, T. and Rusinkiewicz, S. (2017) State of the art in methods and representations for fabrication-aware design, in *Computer Graphics Forum*, pp. 509–535.
- Besl, P.J. and Jain, R.C. (1985) Three-dimensional object recognition. *ACM Computing Surveys*, **17**(1), 75–145.
- Besl, P.J. and Jain, R.C. (1988) Segmentation through variable-order surface fitting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **10**(2), 167–192.
- Besl, P.J. and McKay, N.D. (1992) A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **14**(2), 239–256.
- Bloomenthal, J. (1988) Polygonization of implicit surfaces. *Computer Aided Geometric Design*, **5**(4), 341–355.
- Bourell, D.L., Leu, M.C. and Rosen, D.W. (2009) Roadmap for additive manufacturing: Identifying the future of freeform processing. Technical report, Sponsored by National Science Foundation and the Office of Naval Research, Washington, DC.
- Bugeda Miguel Cervera, G. and Lombera, G. (1999) Numerical prediction of temperature and density distributions in selective laser sintering processes. *Rapid Prototyping Journal*, **5**(1), 21–26.
- Campanelli, S., Cardano, G., Giannoccaro, R., Ludovico, A. and Bohez, E.L. (2007) Statistical analysis of the stereolithographic process to improve the accuracy. *Computer-Aided Design*, **39**(1), 80–86.
- Cheng, L., Wang, A. and Tsung, F. (2018) A prediction and compensation scheme for in-plane shape deviation of additive manufacturing with information on process parameters. *IIE Transactions*, **50**(5), 394–406.
- Colosimo, B., Huang, Q., Dasgupta, T., and Tsung, F. (2018) Opportunities and challenges of quality engineering for additive manufacturing. *Journal of Quality Technology* **50**(3), 233–252.
- Colosimo, B., Pacella, M. and Semerado, Q. (2008) Statistical process control for geometric specifications: on the monitoring of roundness profiles. *Journal of Quality Technology*, **40**(1), 1–18.
- de Souza Borges Ferreira, R., Sabbaghi, A. and Huang, Q. (2020) Automated geometric shape deviation modeling for additive manufacturing systems via Bayesian neural networks. *IEEE Transactions on Automation Science and Engineering*, **17**(2), 584–598.
- Decker, N., Lyu, M., Wang, Y. and Huang, Q. (2021) Geometric accuracy prediction and improvement for additive manufacturing using triangular mesh shape data. *Journal of Manufacturing Science and Engineering*, **143**(6), 061006.
- del Castillo, E. and Colosimo, B.M. (2011) Statistical shape analysis of experiments for manufacturing processes. *Technometrics*, **53**(1), 1–15.
- del Castillo, E., Colosimo, B.M. and Tajbakhsh, S. (2015) Geodesic Gaussian processes for the reconstruction of a 3D free-form surface. *Technometrics*, **57**(1), 87–99.
- Dryden, I.L. and Mardia, K.V. (2016) *Statistical Shape Analysis: With Applications in R* (second edition). Hoboken, NJ: John Wiley & Sons.
- Everton, S.K., Hirsch, M., Stravroulakis, P., Leach, R.K. and Clare, A.T. (2016) Review of in-situ process monitoring and in-situ metrology for metal additive manufacturing. *Materials & Design*, **95**, 431–445.
- Grasso, M., Demir, A., Previtali, B. and Colosimo, B. (2018) In situ monitoring of selective laser melting of zinc powder via infrared imaging of the process plume. *Robotics and Computer-Integrated Manufacturing*, **49**, 229–239.
- Hilton, P. and Jacobs, P. (2000) *Rapid Tooling: Technologies and Industrial Applications*. New York: CRC.
- Ho-Le, K. (1988) Finite element mesh generation methods: A review and classification. *Computer-Aided Design*, **20**(1), 27–38.
- Huang, Q. (2016) An analytical foundation for optimal compensation of three-dimensional shape deformation in additive manufacturing. *ASME Transactions, Journal of Manufacturing Science and Engineering*, **138**(6), 061010.
- Huang, Q., Nouri, H., Xu, K., Chen, Y., Sosina, S. and Dasgupta, T. (2014) Statistical predictive modeling and compensation of geometric deviations of three-dimensional printed products. *ASME Transactions, Journal of Manufacturing Science and Engineering*, **136**(6), 061008.
- Huang, Q., Wang, Y., Lyu, M. and Lin, W. (2020) Shape deviation generator (sdg) – a convolution framework for learning and predicting 3D printing shape accuracy. *IEEE Transactions on Automation Science and Engineering*, **17**(3), 1486–1500.
- Huang, Q., Zhang, J., Sabbaghi, A. and Dasgupta, T. (2015) Optimal offline compensation of shape shrinkage for 3d printing processes. *IIE Transactions on Quality and Reliability*, **47**(5), 431–441.
- Jin, Y., Pierson, H. and Liao, H. (2019) Scale and pose-invariant feature quality inspection for freeform geometries in additive manufacturing. *Journal of Manufacturing Science and Engineering*, **141**(12), 1–15.
- Jin, Y., Qin, S. and Huang, Q. (2016) Offline predictive control of out-of-plane geometric errors for additive manufacturing. *ASME*

- Transactions on Manufacturing Science and Engineering*, **138**(12), 121005.
- Jin, Y., Qin, S. and Huang, Q. (2020) Modeling inter-layer interactions for out-of-plane shape deviation reduction in additive manufacturing. *IISE Transactions on Design and Manufacturing*, **52**(7), 721–731.
- Khanzadeh, M., Rao, P., Jafari-Marandi, R., Smith, B.K., Tschoop, M.A. and Bian, L. (2018) Quantifying geometric accuracy with unsupervised machine learning: Using self-organizing map on fused filament fabrication additive manufacturing parts. *Journal of Manufacturing Science and Engineering*, **140**(3), 031011.
- King, W.E., Anderson, A.T., Ferencz, R.M., Hodge, N.E., Kamath, C., Khairallah, S.A. and Rubenchik, A.M. (2015) Laser powder bed fusion additive manufacturing of metals: physics, computational, and materials challenges. *Applied Physics Reviews*, **2**(4), 041304.
- Leach, R.K., Bourell, D., Carmignato, S., Donmez, A., Senin, N. and Dewulf, W. (2019) Geometrical metrology for metal additive manufacturing. *CIRP Annals*, **68**(2), 677–700.
- Luan, H., Grasso, M., Colosimo, B. and Huang, Q. (2019) Prescriptive data-analytical modeling of laser powder bed fusion processes for accuracy improvement. *Journal of Manufacturing Science and Engineering, Transactions of the ASME*, **141**(1), 011008.
- Luan, H. and Huang, Q. (2017) Prescriptive modeling and compensation of in-plane shape deformation for 3-d printed freeform products. *IEEE Transactions on Automation Science and Engineering*, **14**(1), 73–82.
- McConaha, M. and Anand, S. (2020) Additive manufacturing distortion compensation based on scan data of built geometry. *Journal of Manufacturing Science and Engineering*, **142**(6), 061001.
- Montagnat, J., Delingette, H. and Ayache, N. (2001) A review of deformable surfaces: topology, geometry and deformation. *Image and Vision Computing*, **19**(14), 1023–1040.
- Mori, K., Osakada, K. and Takaoka, S. (1996) Simplified three-dimensional simulation of non-isothermal filling in metal injection moulding by the finite element method. *Engineering Computations*, **13**(2), 111–121.
- Morse, E., Dantan, J.Y., Anwer, N., Söderberg, R., Moroni, G., Qureshi, A., Jiang, X. and Mathieu, L. (2018) Tolerancing: Managing uncertainty from conceptual design to final product. *CIRP Annals*, **67**(2), 695–717.
- Pal, D., Patil, N., Zeng, K. and Stucker, B. (2014) An integrated approach to additive manufacturing simulations using physics based, coupled multiscale process modeling. *ASME Transactions, Journal of Manufacturing Science and Engineering*, **136**(6), 061022.
- Pillonetto, G., Dinuzzo, F., Chen, T., De Nicolao, G. and Ljung, L. (2014) Kernel methods in system identification, machine learning and function estimation: A survey. *Automatica*, **50**(3), 657–682.
- Sabbaghi, A. and Huang, Q. (2018) Model transfer across additive manufacturing processes via mean effect equivalence of lurking variables. *Annals of Applied Statistics*, **12**(4), 2409–2429.
- Sabbaghi, A., Huang, Q. and Dasgupta, T. (2018) Bayesian model building from small samples of disparate data for capturing in-plane deviation in additive manufacturing. *Technometrics*, **60**(4), 532–544.
- Samie Tootooni, M., Dsouza, A., Donovan, R., Rao, P.K., Kong, Z.J. and Borgesen, P. (2017) Classifying the dimensional variation in additive manufactured parts from laser-scanned three-dimensional point cloud data using machine learning approaches. *Journal of Manufacturing Science and Engineering*, **139**(9), 091005.
- Savio, E., De Chiffre, L. and Schmitt, R. (2007) Metrology of freeform shaped parts. *CIRP Annals - Manufacturing Technology*, **56**(2), 810–835.
- Snyder, J. (1992) *Generative Modeling for Computer Graphics and CAD*, Academic Press, Boston, MA.
- Srinivasan, V. (1999) A geometrical product specification language based on a classification of symmetry groups. *Computer Aided Design*, **31**(1), 659–668.
- Tam, G.K., Cheng, Z.Q., Lai, Y.K., Langbein, F.C., Liu, Y., Marshall, D., Martin, R.R., Sun, X.F. and Rosin, P.L. (2013) Registration of 3d point clouds and meshes: A survey from rigid to nonrigid. *IEEE Transactions on Visualization and Computer Graphics*, **19**(7), 1199–1217.
- Tapia, G. and Elwany, A. (2014) A review on process monitoring and control in metal-based additive manufacturing. *ASME Transactions, Journal of Manufacturing Science and Engineering*, **136**(6), 060801–060810.
- Tong, K., Joshi, S. and Lehtihet, E. (2008) Error compensation for fused deposition modeling (fdm) machine by correcting slice files. *Rapid Prototyping Journal*, **14**(1), 4–14.
- Tong, K., Lehtihet, E. and Joshi, S. (2003) Parametric error modeling and software error compensation for rapid prototyping. *Rapid Prototyping Journal*, **9**(5), 301–313.
- Tsung, F., Zhang, K., Cheng, L. and Song, Z. (2018) Statistical transfer learning: A review and some extensions to statistical process control. *Quality Engineering*, **30**(1), 115–128.
- Van Kaick, O., Zhang, H., Hamarneh, G. and Cohen-Or, D. (2011) A survey on shape correspondence. *Computer Graphics Forum*, **30**, 1681–1707.
- Várady, T., Martin, R.R. and Cox, J. (1997) Reverse engineering of geometric models - an introduction. *CAD Computer Aided Design*, **29**(4), 255–268.
- Walker, R.K. and Srinivasan, V. (1994) Creation and evolution of the ASME Y14. 5.1 M standard. *Manufacturing Review*, **7**(1), 16–23.
- Wang, A., Song, S., Huang, Q. and Tsung, F. (2017) In-plane shape-deviation modeling and compensation for fused deposition modeling processes. *IEEE Transactions on Automation Science and Engineering*, **17**(2), 968–976.
- Wang, Y., Ruiz, C. and Huang, Q. (2022) Learning and predicting shape deviations of smooth and non-smooth 3d geometries through mathematical decomposition of additive manufacturing. *IEEE Transactions on Automation Science and Engineering*. <https://doi.org/10.1109/TASE.2022.3174228>
- Xu, K., Kwok, T.-H., Zhao, Z. and Chen, Y. (2017) A reverse compensation framework for shape deformation control in additive manufacturing. *Journal of Computing and Information Science in Engineering*, **17**(2), 021012.
- Yang, J., Li, H., Campbell, D., and Jia, Y. (2016) Go-ICP: A globally optimal solution to 3D ICP point-set registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **38**(11), 2241–2254.
- Zhang, D. and Lu, G. (2004) Review of shape representation and description techniques. *Pattern Recognition*, **37**(1), 1–19.
- Zhou, J., Herscovici, D. and Chen, C. (2000) Parametric process optimization to improve the accuracy of rapid prototyped stereolithography parts. *International Journal of Machine Tools and Manufacture*, **40**(3), 363–379.